МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования «Самарский национальный исследовательский
университет имени академика С.П. Королева»
(Самарский университет)

На правах рукописи

Мясников Евгений Валерьевич

ФОРМИРОВАНИЕ ИНФОРМАТИВНЫХ ПРИЗНАКОВ ГИПЕРСПЕКТРАЛЬНЫХ ИЗОБРАЖЕНИЙ НА ОСНОВЕ ФУНКЦИЙ РАССТОЯНИЯ

2.3.8. Информатика и информационные процессы

Диссертация на соискание ученой степени
доктора технических наук

Научный консультант:
Сергеев Владислав Викторович,
доктор технических наук, профессор

Самара - 2026

\

Содержание

Введение

Диссертация посвящена разработке методов, алгоритмов и программных средств формирования признаков цифровых гиперспектральных изображений, обеспечивающих повышенное качество решения прикладных задач.

Актуальность темы.

Гиперспектральные изображения (ГСИ) представляют собой трехмерные массивы данных, содержащие в себе как пространственную информацию о наблюдаемой сцене, так и детальную спектральную информацию, соответствующую каждой паре дискретных пространственных координат. В привычных для нас цветных изображениях каждый пиксель содержит спектральную информацию, зарегистрированную в трех широких спектральных полосах, соответствующих красному, зеленому и синему каналам. Широко используемые в дистанционном зондировании Земли (ДЗЗ) мультиспектральные изображения содержат большее количество каналов (как правило, от 4 до 16) и могут включать как видимый, так и невидимый для человеческого глаза диапазоны (поддиапазоны инфракрасного, редко ультрафиолетовый диапазон). В отличие от цветных и мультиспектральных изображений, ГСИ регистрируются с гораздо более высоким спектральным разрешением и содержат сотни спектральных каналов.

На сегодняшний день гиперспектральная съемка применяется в самых различных областях человеческой деятельности. Используемая изначально для целей геологии (разведка полезных ископаемых), гиперспектральная съемка успешно применяется в сельском хозяйстве (мониторинг состояния посевов), минералогии (поиск и идентификация минералов), пищевой индустрии (оценка качества продуктов, поиск инородных включений и т.д.), а также химии, медицине, экологии, астрономии и других областях.

Постепенное снижение стоимости и повышение качества гиперспектральных камер, а также неуклонное развитие аппаратных средств, используемых для обработки результатов гиперспектральной съемки, приводят как к росту объемов гиперспектральных данных, так и расширению спектра решаемых с их помощью прикладных задач. Так, например, уже появились публикации [16, 126], в которых

гиперспектральная съемка используется для повышения качества распознавания сцены при автономной навигации.

Развитию аппаратных средств, используемых для регистрации гиперспектральных данных, сопутствует и развитие алгоритмических средств, используемых для обработки и анализа гиперспектральных данных. Развитием этого научного направления занимались научные коллективы, как в России, так и за рубежом. Можно выделить целый ряд российских ученых, внесших вклад в развитие указанного научного направления: Зеленцов В.А. (Санкт-Петербургский Федеральный исследовательский центр Российской академии наук), Казанский Н.Л. (Институт систем обработки изображений Российской академии наук), Козодеров В.В. (Московский государственный университет имени М.В.Ломоносова), Лупян Е.А. (Институт Космических Исследований Российской академии наук), Мясников В.В. (Самарский университет), Никоноров А.В. (Самарский университет), Обухова Н.А. (Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова), Пестунов И.А. (Федеральный исследовательский центр информационных и вычислительных технологий, Новосибирск), Пожар В.Э. (Научно-технологический центр уникального приборостроения Российской академии наук), Потатуркин О.И. (Институт автоматики и электрометрии Сибирского отделения Российской академии наук), Пяткин В.П. (Институт вычислительной математики и математической геофизики Сибирского отделения Российской академии наук), Саворский В.П. (Московский государственный университет имени М.В.Ломоносова), Сергеев В.В. (Самарский университет), Сойфер В.А. (Самарский университет), Турлапов В.Е. (Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского), Фурсов В.А. (Самарский университет), Шилин Б.В. (Санкт-Петербургский Федеральный исследовательский центр Российской академии наук) и и др. Среди зарубежных ученых можно отметить Benediktsson J.A., Bioucas-Dias J.M., Boardman J.W., Bruzzone L., Camps-Valls G., Chang C.-I., Chanussot J., Fauvel M., Gamba P., Landgrebe D. Mustard J.F., Nascimento J.M.P., Plaza A. и др.

ГСИ имеют ряд особенностей, которые необходимо учитывать при их обработке и анализе. Во-первых, вследствие большого количества регистрируемых спектральных компонент, ГСИ занимают при равном пространственном разрешении

гораздо больший объем памяти, чем обычные цветные или мультиспектральные изображения, что обуславливает чрезвычайно высокую стоимость и временные затраты как на передачу и хранение, так и на обработку таких изображений. Во-вторых, *гиперспектральные данные высокоспецифичны*, т.к. различная регистрирующая аппаратура может иметь различное число спектральных каналов, снимать в различных спектральных диапазонах, иметь различное пространственное разрешение и т.п. В-третьих, классы изображенных на сцене объектов часто обладают большой вариативностью – зависимостью их спектральных характеристик от целого ряда факторов (*спектральная изменчивость*). Например, при анализе растительности влияние на регистрируемые спектральные характеристики оказывают условия освещения, атмосфера, угол съемки, фаза развития, условия роста растений, а в некоторых случаях время суток, погодные условия и т.д.

Другой важный аспект, который необходимо учитывать при анализе ГСИ, связан с семантической разметкой таких данных. Ручная разметка ГСИ, является, как правило, трудоемкой и дорогостоящей процедурой, так как связана с необходимостью осуществления наземных измерений (для изображений ДЗЗ), привлечения квалифицированных экспертов для анализа биоматериалов (в медицине) и т.п. Вследствие этого *формируемые обучающие выборки для процедур классификации, как правило, очень ограничены по объему*.

Высокое спектральное разрешение ГСИ приводит к тому, что соседние спектральные диапазоны оказываются очень узкими и сильно коррелированными, что обуславливает, во-первых, *избыточность ГСИ*, а, во-вторых, *высокую размерность пространства спектров пикселей гиперспектральных изображений* (далее - спектрального пространства). Последняя особенность, с учетом ограниченности объемов обучающих выборок, приводит к разреженности спектрального пространства [5] и создает предпосылки для проявления, так называемого, *феномена проклятья размерности* [56, 137], что может вызывать падение качества работы традиционных методов анализа и классификации, рост вычислительных затрат [166].

По перечисленным выше причинам при решении задач анализа и классификации ГСИ часто используются не исходные данные, а извлеченная из них признаковая информация. При формировании такой информации, с одной стороны, стараются уменьшить ее избыточность, что позволяет снизить расходы на передачу (в

том случае, когда формирование признаков производится до передачи) и хранение данных, а также производить обработку и анализ данных за меньшее время. С другой стороны, к формируемым признакам предъявляют требование информативности, то есть возможности решения прикладных задач с тем же или более высоким качеством по сравнению с исходными гиперспектральными данными. Успешное решение проблемы формирования информативных признаков позволит вместо исходного ГСИ работать со сформированным признаковым представлением, обеспечивающим наилучшие характеристики качества для решения различных прикладных задач обработки и анализа ГСИ (классификации, сегментации, обнаружения аномалий, визуализации и др.), обращаясь к исходным гиперспектральным данным лишь в особых случаях.

Отметим, что существуют различные трактовки информативности [343] (вероятностная, алгоритмическая, комбинаторная и др.), каждая из которых эффективна в своём классе задач. В рамках настоящей работы мы определяем информативность признакового описания непосредственно через эмпирическое качество решения задачи попиксельной классификации ГСИ, где в качестве критерия выступают стандартные показатели точности классификации, рассчитанные по результатам на контрольной выборке. Это позволяет учесть как специфику данных, так и практическую значимость получаемого решения.

Сегодня для анализа ГСИ используется широкий спектр «классических» признаков (спектральные, пространственные и спектрально-пространственные признаки), а также нейросетевые признаки. К сожалению, исходные спектральные признаки сильно избыточны, отдельные спектральные компоненты или их комбинации в виде индексов, эффективно используемые при решении одних задач, часто малоинформативны при решении других. Существенным недостатком спектральных признаков является их неспособность учитывать пространственную информацию ГСИ. Пространственные (текстурные) признаки, чаще всего извлекаются из одно- или трехкомпонентных представлений, полученных тем или иным образом из ГСИ. Вследствие этого, детальная спектральная информация в таких признаках оказывается утраченной.

Большей информативностью обладают спектрально-пространственные признаки. Часто такие признаки формируются путем объединения богатой

спектральной информации, содержащейся в пикселях ГСИ, и локальных пространственных признаков, либо путем извлечения трехмерных участков (патчей) ГСИ. Так как сформированные такими способами признаки имеют еще больший объем, чем сами ГСИ, от исходных ГСИ переходят к их представлениям в виде сегментов или суперпикселей, которые описывают усредненными спектральными и пространственными характеристиками. В таких описаниях, однако, оказываются утраченными отличия между пикселями внутри сегментов (суперпикселей).

По указанным причинам все большую популярность среди исследователей завоевывают признаки, извлекаемые из ГСИ глубокими нейросетями. В том случае, когда используемые нейросети тем или иным способом адаптированы под конкретный класс ГСИ, такой подход способен учитывать как пространственную, так и богатую спектральную информацию, содержащуюся в ГСИ в их исходном разрешении. Однако необходимость настройки большого количества параметров в нейросетях является фактором, сдерживающим их практическое применение для анализа ГСИ по причине недостаточного объема обучающих выборок. Возможности использования методов переноса обучения (трансферного обучения) в области ГСИ также ограничены из-за высокой специфичности гиперспектральных данных и решаемых задач (в отличие от распространенных задач компьютерного зрения), а также спектральной изменчивости данных. Указанные факторы в совокупности приводят к тому, что получение достаточно универсальных нейросетевых моделей, эффективно учитывающих природу гиперспектральных изображений, на сегодняшний день весьма затруднительно.

Эффективным подходом к формированию информативных признаков ГСИ может стать использование методов снижения размерности. Применение таких методов для формирования информативных признаков позволит устранить не только избыточность гиперспектральных данных, но и предпосылки проявления «проклятья размерности». В подавляющем большинстве случаев при анализе ГСИ используются линейные методы снижения размерности, однако, по мнению автора, более предпочтительным следовало бы считать использование нелинейных методов, так как именно они позволяют учесть нелинейную природу гиперспектральных данных (проявляющуюся вследствие многократных переотражений света в реальных сценах).

Сегодня для снижения размерности гиперспектральных данных, как правило, используются методы общего назначения [232, 114, 257, 290, 252, 18, 327]. Такие методы предназначены для работы с наборами данных произвольной природы и не учитывают специфику гиперспектральных изображений. В подавляющем большинстве случаев используются линейные методы снижения размерности [101, 295, 211*] (здесь и далее звездочкой помечены работы автора), такие как метод главных компонент [232, 114]. Однако во многих случаях использование линейных методов снижения размерности оказывается недостаточным из-за невысокого качества решения конечных прикладных задач. Причина этого может заключаться в том, что анализируемые данные плохо описываются линейными моделями. Было показано [23, 67], что во многих случаях широко используемая модель линейной смеси спектральных сигнатур [3, 269, 4, 270] не является точной и следует использовать нелинейные модели спектральной смеси [218, 268, 326, 272, 75, 97, 8, 44, 173, 98, 264, 177, 265, 107, 48]. Здесь под спектральной сигнатурой поверхности будем понимать зарегистрированное излучение, представленное в виде функции длины волны [275].

По указанной причине при снижении размерности предпочтение может быть отдано нелинейным методам, позволяющим учесть природу гиперспектральных данных. В разное время разработкой нелинейных методов снижения размерности занимались J. Sammon Jr. [257], T. Kohonen [144], M. Kramer [146], P. Demartines [63], J. Hérault, J. Tenenbaum [289, 290], K. Weinberger [309], L. Saul, F. Sha, S. Roweis [252], L. Saul, M. Belkin [18], P. Niyogi, Z. Zhang [327], H. Zha, G. Hinton [109], L. McInnes [170], J. Healy, J. Melville, А.Н. Горбань [90], А.Ю. Зиновьев и др.

Нелинейные методы снижения размерности, как правило, опираются в своей работе на использование евклидовой метрики, однако допускают и возможность использования других функций расстояния. Указанная возможность представляется существенной, так как использование именно неевклидовых расстояний (спектральный угол [147], дивергенция спектральной информации [39] и др.) представляется при анализе ГСИ более обоснованным с физической точки зрения (например, обеспечивается инвариантность к уровню освещенности), а также часто обеспечивает превосходство перед евклидовой метрикой с точки зрения качества решения прикладных задач.

К сожалению, известные вычислительные процедуры формирования признаков, основанные на нелинейных методах снижения размерности общего назначения, как правило, обладают неприемлемо большим временем работы и пригодны для обработки лишь небольших наборов данных. Кроме того, в них нет естественной возможности учета пространственной информации, содержащейся в изображениях, а также отсутствуют механизмы совмещения (слияния) извлекаемых из ГСИ признаков различной природы. Поэтому разработка методов формирования информативных признаков ГСИ на основе специализированных алгоритмов снижения размерности, позволяющих учесть как различные функции расстояния, используемые при анализе ГСИ, так и содержащуюся в ГСИ пространственную информацию, представляется *возможной и перспективной*.

Исходя из сказанного, **научно-техническая проблема** *вычислительно-эффективного* формирования *информативных признаков гиперспектральных изображений, позволяющих устранить избыточность (сократить размерность) гиперспектральных данных без потери существенной информации, содержащейся в исходных гиперспектральных изображениях,* представляется важной и актуальной. Решению указанной проблемы и посвящена настоящая диссертация.

Цель и задачи исследования. Целью исследования является разработка методов формирования *информативных признаков гиперспектральных изображений на основе функций расстояния* для повышения эффективности использования гиперспектральных изображений.

Для достижения поставленной цели в диссертации решаются следующие **задачи**:

1. Анализ существующих методов и алгоритмов формирования признаков гиперспектральных изображений.
2. Разработка алгоритмов формирования признаков ГСИ на основе различных функций расстояния, используемых при анализе ГСИ.
3. Разработка методов формирования признаков ГСИ, позволяющих учитывать пространственную информацию, содержащуюся в ГСИ, а также выполнять слияние разнородных признаков.

4. Разработка алгоритмических способов ускорения формирования признаков, а также разработка параллельных реализаций, позволяющих выполнять формирование признаков с использованием современных графических процессоров.

5. Оценка эффективности разработанных методов в прикладных задачах классификации, кластеризации, сегментации и визуализации ГСИ.

Объект исследования – цифровые гиперспектральные изображения.

Предмет исследования – методы формирования признаков, извлекаемых из гиперспектральных изображений.

Методы исследования. В диссертационной работе используются методы машинного обучения и анализа данных, теории вероятностей и математической статистики, методы оптимизации.

Новизна исследования.

1. Разработан комплекс алгоритмических средств формирования признаков ГСИ на основе сохранения попарных расстояний, а именно: спектральных углов, дивергенции спектральной информации, дивергенции Хеллингера, угла Бхаттачария, а также спектральной корреляции.

Предложенные алгоритмы отличаются от известных применением методов снижения размерности гиперспектральных данных с учетом заданных неевклидовых расстояний. Предложенные алгоритмы позволяют формировать на основе содержащейся в ГСИ спектральной информации компактные признаковые представления, обеспечивающие повышенную точность попиксельной классификации.

2. Разработан новый метод формирования признаков ГСИ с учетом локального пространственного контекста на основе порядковых статистик.

Предложенный метод отличается от известных использованием для учета пространственного контекста порядковых статистик, рассчитанных с использованием выбранных функций расстояния в спектральном пространстве пикселей ГСИ. Предложенный метод позволяет формировать на основе пространственно-спектральной информации, содержащейся в ГСИ, компактные признаковые представления, обеспечивающие повышенную точность попиксельной классификации.

3. Разработан новый метод слияния разнородных признаков на основе приведения исходных систем признаков к однородным промежуточным представлениям с последующим объединением этих представлений и снижением размерности; предложены модификации метода для случаев использования пространства с евклидовой метрикой и l_p метрикой для промежуточного представления.

Предложенный метод отличается от известных использованием оригинальной схемы слияния разнородных признаков на основе выбранных функций расстояния, как в пространстве исходных признаков, так и в формируемом пространстве. Метод позволяет формировать на основе заданных систем признаков компактные признаковые представления ГСИ, обеспечивающие повышенную точность попиксельной классификации.

4. Разработаны новые алгоритмы и численные методы, обеспечивающие ускорение формирования признаков ГСИ, а именно:

- предложены алгоритм построения списка опорных узлов и численный метод ускорения формирования признаков на основе опорных узлов, а также модификация метода, учитывающая характеристики узлов как во входном, так и в формируемом пространствах;

- предложены алгоритм построения очереди опорных узлов с приоритетом и соответствующая модификация метода ускорения формирования признаков на основе очередей с приоритетом;

- предложен комбинированный метод ускорения формирования признаков на основе стохастического градиентного спуска, очередей опорных узлов с приоритетом и алгоритмов многомерной интерполяции.

Предложенные методы отличаются от известных использованием бинарных деревьев для разделения входного и формируемых пространств при снижении размерности гиперспектральных данных, а также использованием очередей с приоритетом для управления вычислительной сложностью. Предложенные методы обеспечивают повышенную производительность формирования признаков.

5. Разработаны программные средства формирования признаков ГСИ с использованием параллельной многопоточной реализации для современных графических процессоров NVIDIA и AMD.

Предложенные алгоритмы и программная реализация отличаются от соответствующих последовательных реализаций повышенной производительностью за счет использования специфики аппаратных средств.

6. Разработаны новые методы и алгоритмы решения некоторых задач анализа ГСИ:

- предложен алгоритм кластеризации, основанный на использовании неевклидовых метрик, в частности, дивергенции Хеллингера и угла Бхаттачария;

- предложен новый метод сегментации ГСИ, основанный на оценке размерности в пространственных областях.

7. Представлены результаты экспериментальных исследований, демонстрирующих преимущества предложенных методов и алгоритмов.

Научное и прикладное значение.

Научная значимость работы состоит в развитии методов неуправляемого анализа многомерных данных, в частности, ГСИ. Предложенные методы и алгоритмы формирования признаков работоспособны при отсутствии априорной информации об информативности исходных признаков, не требуют предварительной разметки данных или обучающих множеств. Полученные в работе результаты в совокупности доказывают, что построение информативных признаков в принципе возможно в рамках неуправляемого подхода, если правильно определить функцию расстояний между экземплярами.

Предложенные во втором разделе диссертации алгоритмы формирования признаков и способ их построения, а также предложенный в третьем разделе метод слияния разнородных признаков в значительной степени универсальны – их можно распространить на данные других типов (модальностей) в тех случаях, когда различия между экземплярами данных можно измерить с использованием задаваемых пользователем функций расстояния. Предложенный в пятом разделе алгоритм кластеризации, основанный на использовании неевклидовых метрик, также может быть распространен на другие метрики или использоваться с многомерными данными других типов.

Разработанные в четвертом разделе численные методы, обеспечивающие ускорение формирования признаков, также могут рассматриваться в качестве методов

снижения размерности общего назначения с пониженной (управляемой) вычислительной сложностью и применяться в других областях.

В целом, предложенные методы и алгоритмы формирования признаков расширяют инструментарий анализа ГСИ.

Прикладное значение работы состоит в снижении избыточности исходных гиперспектральных данных с сохранением информативности и без потери пространственного разрешения. Разработанные в диссертации методы и алгоритмы позволяют перейти от исходных ГСИ, содержащих сотни спектральных каналов, к изображениям, содержащим до 20-30 компонент (признаков) и обеспечивающим ускорение обработки при аналогичных или лучших результатах распознавания.

Разработанные в четвертом разделе программные средства формирования признаков ГСИ с использованием параллельной многопоточной реализации позволяют значительно (до 300 раз) сократить время формирования признаков или снижения размерности по сравнению с однопоточной версией.

Предложенные в работе методы и алгоритмы могут применяться при решении задач классификации гиперспектральных данных дистанционного зондирования, полученных с космических и атмосферных летательных аппаратов, данных наземной гиперспектральной съемки, гиперспектральных данных дорожной обстановки, гиперспектральных медицинских изображений и др., что подтверждается представленными в работе примерами решения актуальных прикладных задач.

Область применения. Результаты исследования применимы в компьютерных системах обработки, анализа и распознавания ГСИ. Наибольший эффект может быть достигнут в тех областях, где уже сегодня широко используется гиперспектральная съемка (сельское хозяйство, дистанционный мониторинг чрезвычайных ситуаций, медицина, пищевая индустрия и др.).

Кроме того, результаты исследования могут быть применимы в научных исследованиях для визуализации многомерных признаковых пространств и формирования псевдоцветовых представлений мульти- и гиперспектральных изображений.

Реализация результатов работы. Результаты диссертации использованы при выполнении проектов РФФИ 18-07-01312-а «Методы нелинейного снижения размерности гиперспектральных изображений и их применение», 16-37-00202-мол_а «Разработка методов сегментации мульти- и гиперспектральных цифровых спутниковых снимков с использованием нелинейного отображения с пониженной вычислительной сложностью», 20-37-70053-ст «Методы и алгоритмы обеспечения пассивной защиты цифровых изображений и их последовательностей с использованием внутренней и внешней информации», 18-01-00748-а «Анализ и обработка цифровых изображений с использованием нелинейных отображений, конструируемых с использованием отношений», 16-29-09494-офи_м «Методы компьютерной обработки мультиспектральных данных дистанционного зондирования Земли для определения ареалов растений в специальных криминалистических экспертизах», 09-01-00434-а «Эффективные линейные локальные признаки цифровых сигналов и изображений», 07-07-97603-р_офи "Информационные технологии создания региональной инфраструктуры геопространственных данных", 07-07-97610-р_офи «Разработка фундаментальных принципов синтеза и оптимизации алгоритмов обработки многомерных сигналов/изображений», 07-01-12070-офи «Разработка новых информационных технологий компрессии многомерных сигналов», грантов Министерства образования и науки Самарской области «Анализ мульти- и гиперспектральных космических изображений с использованием нелинейных методов снижения размерности", "Развитие методов анализа гиперспектральных космических изображений с использованием нелинейных методов снижения размерности данных с пониженной вычислительной сложностью», научно-исследовательского проекта «Фундаментальные проблемы разработки аэрокосмических транспортных систем и управления в аэрокосмической технике для обеспечения связанности территории Российской Федерации» (Соглашение № 075-15-2024-558) в рамках государственной программы Российской Федерации «Научно технологическое развитие Российской Федерации», а также в ряде хоздоговорных НИР в Самарском университете, АО «Самара-Информспутник», Институте систем обработки изображений РАН — филиале ФНИЦ «Кристаллография и фотоника» РАН (наименования тем указаны в соответствующих актах использования результатов диссертации).

Апробация работы. Основные результаты диссертации были представлены на международных научных конференциях: Image Mining. Theory and Applications – IMTA, проводимая в рамках International Conference on Pattern Recognition - ICPR (Индия, Калькутта, 2024; Монреаль, Канада, 2022; Милан, Италия, 2021), International Workshop on Photogrammetric Data Analysis - ISPRS PDA24 (Москва, Россия, 2024), Международная конференция и молодежная школа «Информационные технологии и нанотехнологии» (Самара, Россия, 2016-2023), Symposium on Intelligent Systems and Telecommunication - IST в рамках SIBIRCON 2022 (2022), Workshop on Hyperspectral Image and Signal Processing, Evolution in Remote Sensing - WHISPERS 2021 (Амстердам, Нидерланды, 2021), Analysis of Images, Social Networks and Texts - AIST (Тбилиси, Грузия, 2021; Москва, Россия, 2017, 2018), International Multi-Conference on Industrial Engineering and Modern Technologies - FarEastCon, (Владивосток, Россия, 2020), International Conference on Machine Vision - ICMV (Амстердам, Нидерланды, 2019; Барселона, Испания, 2015; Ницца, Франция, 2016), International Multi-Conference on Engineering, Computer and Information Sciences – SIBIRCON (Екатеринбург, Россия, 2019), International Conference on Computer Vision and Graphics – ICCVG 2016 (Варшава, Польша, 2016, 2018), International Conference on Image Analysis and Processing - ICIAP 2017 (Катания, Италия, 2017), Computer Analysis of Images and Patterns – CAIP 2017 (Истад, Швеция, 2017), International Conference on Image Analysis and Recognition – ICIAR 2016 (Повоа де варзим, Португалия, 2016), 23rd International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision - WSCG 2015 (Пльзень, Чехия), 9-th Open German-Russian Workshop on Pattern Recognition and Image Understanding – OGRW-2014 (Кобленц, Германия, 2014), International Conference on Pattern Recognition and Image Analysis: New Information Technologies (Санкт-Петербург, Нижний Новгород, Йошкар-Ола, Самара, Россия, 2004-2013), 9-я международная конференция «Интеллектуализация обработки информации» (Будва, Черногория, 2012), 9-ая научная конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» - RCDL’2007, (Переславль-Залесский, Россия, 2007), Second IASTED International Multi-Conference Software Engineering (Новосибирск, Россия, 2005).

Публикации. По теме диссертационной работы автором опубликовано 45 работ, в т.ч. 39 работ в изданиях, индексируемых в базах данных Scopus и Web of Science (WoS), и глава в монографии. 36 работ, индексируемых в базах данных Scopus и Web of Science (WoS), выполнены без соавторов.

Соответствие паспорту специальности 2.3.8. Информатика и информационные процессы. Область исследования соответствует направлениям исследований паспорта научной специальности 2.3.8 в части пунктов: 3 - Разработка методов и алгоритмов кодирования, сжатия и размещения информации для повышения эффективности и надежности функционирования инфокоммуникационных систем при её хранении и передаче; 4 - Разработка методов и технологий цифровой обработки аудиовизуальной информации с целью обнаружения закономерностей в данных, включая обработку текстовых и иных изображений, видео контента; 7 - Разработка методов обработки, группировки и аннотирования информации, в том числе, извлеченной из сети интернет, для систем поддержки принятия решений, интеллектуального поиска, анализа; 12 - Разработка технологий извлечения и анализа информации в больших базах данных, в том числе, с использованием концепции многомерного представления (OLAP) и интеллектуального анализа данных (Data Mining) статического и в реальном масштабе времени, реализация моделей баз знаний; 13 - Разработка и применение методов распознавания образов, кластерного анализа, нейросетевых и нечетких технологий, решающих правил, мягких вычислений при анализе разнородной информации в базах данных.

Структура диссертации.

Диссертация состоит из введения, пяти глав, заключения, списка литературы (366 наименований) и четырех приложений. Работа изложена на 354 страницах, содержит 102 рисунка, 24 таблицы.

На защиту выносятся:

1. Комплекс алгоритмических средств формирования сжатого признакового представления ГСИ на основе сохранения попарных расстояний, а

именно: спектральных углов, дивергенции спектральной информации, дивергенции Хеллингера, угла Бхаттачария, а также спектральной корреляции для последующего хранения и анализа.

2. Метод формирования сжатого признакового представления ГСИ с учетом локального пространственного контекста на основе порядковых статистик для последующего хранения и анализа.

3. Метод слияния разнородных признаков на основе приведения исходных систем признаков к однородным промежуточным представлениям с последующим объединением этих представлений и снижением размерности; модификации метода для случаев использования пространства с евклидовой метрикой и l_p метрикой для промежуточного представления разнородной информации.

4. Комплекс алгоритмических средств (численные методы), обеспечивающих ускорение формирования сжатого признакового представления ГСИ и управление вычислительной сложностью, и построенных на основе стохастических алгоритмов, интерполяции и разделения входного и формируемого пространств с использованием специальных древовидных структур данных.

5. Комплекс алгоритмических средств (алгоритмы и параллельная программная реализация), выполняющих формирование сжатого признакового представления ГСИ с использованием графических процессоров NVIDIA и AMD.

6. Алгоритм кластеризации (группировки), основанный на использовании неевклидовых метрик, в частности, дивергенции Хеллингера и угла Бхаттачария. Метод сегментации ГСИ, основанный на оценке размерности в пространственных областях.

7. Результаты решения прикладных задач (результаты экспериментальных исследований) обработки и анализа гиперспектральных изображений, подтвердившие эффективность разработанных методов и алгоритмов, а именно:

- классификации (аннотирования) гиперспектральных данных дистанционного зондирования высокого разрешения, полученных с беспилотного летательного аппарата;

- классификации (аннотирования) гиперспектральных данных дорожной обстановки;

- классификации (аннотирования) гиперспектральных изображений мозга человека;

- классификации (аннотирования) растительности по данным наземной гиперспектральной съемки;

- визуализации признаковых пространств и формирования псевдоцветовых представлений гиперспектральных изображений.

1 Методы формирования признаков гиперспектральных изображений

1.1 Анализ современного состояния в области исследования

Исходные гиперспектральные изображения содержат большой объем спектрально-пространственный информации о наблюдаемой сцене. Для эффективного использования такой информации было предложено большое число различных признаков, которые можно разделить на несколько крупных категорий: спектральные, пространственные, спектрально-пространственные признаки, а также признаки формы.

Исходные *спектральные признаки* представляют собой значения яркости (отражающей способности) наблюдаемого элемента сцены в каждом из регистрируемых спектральных каналов. Хотя такие признаки хранят все богатство исходной спектральной информации и могут непосредственно использоваться для решения различных прикладных задач, они имеют высокую размерность, сильную корреляцию между соседними каналами. При работе с такими признаками проявляется феномен «проклятья размерности», о котором говорилось во введении.

В ряде приложений эффективное решение прикладных задач может быть достигнуто путем анализа небольшого выбранного подмножества каналов исходного ГСИ. Наиболее распространен такой подход в сельском хозяйстве, где для анализа можно довольствоваться использованием нескольких каналов в области красного (670 нм) и ближнего инфракрасного (ИК) диапазона (700…800 нм). Нескольких каналов в ближнем ИК диапазоне и в области зеленого (550 нм) цвета также достаточно для выделения водных объектов. Аналогично, нескольких каналов может быть достаточно для выделения некоторых материалов и минералов.

Широкое распространение получили комбинации нескольких спектральных каналов в виде различных индексов. Например, индекс NDVI (Normalized Difference Vegetation Index) используется для оценки состояния растительности, SAVI (Soil-Adjusted Vegetation Index) - для оценки состояния почвы, NDWI (Normalized Difference Water Index) – для выделения водных объектов. Подобные индексы, как и небольшие подмножества спектральных каналов, достаточно эффективны при решении некоторых прикладных задач, их небольшой объем позволяет выполнять

обработку с высокой скоростью и не предъявляет высоких требований к аппаратному обеспечению. Однако такие признаки часто оказываются малоинформативны или бесполезны при решении других прикладных задач.

Наряду с индексными признаками для анализа спектральной информации применяются также различные преобразования исходных спектральных каналов, такие как метод главных компонент (PCA) [232, 114], одномерные спектральные преобразования [338], декомпозиция на эмпирические моды (EMD) [229, 358] и др.

В целом же главным недостатком спектральных признаков является то, что каждый пиксель ГСИ рассматривается независимо, и пространственная информация, содержащаяся в ГСИ, никак не учитывается. Этот недостаток устраняется при использовании достаточно большого класса *пространственных признаков*, часто называемых также текстурными признаками. Чаще всего такие признаки рассчитываются для каждого пикселя изображения по его локальной окрестности путем обработки скользящим окном. В качестве признаковой информации могут использоваться:

- различные локальные статистические и структурные характеристики [363, 366], например, гистограммы яркости, моментные характеристики, корреляционные характеристики, признаки на основе матрицы совместной встречаемости пикселей [99], признаки Тамура [285], локальные бинарные шаблоны (LBP) [226] и т.д.;

- признаки на основе двумерных спектральных преобразований, извлеченные, например, с использованием Фурье или косинусного преобразования [338], фильтров Габора [92], вейвлет-преобразований [41], преобразования Фурье-Меллина [262, 351*] и др.

Для извлечения пространственных признаков чаще всего формируется одно- или трехкомпонентное представление исходного ГСИ, для чего используются выбранные каналы, либо применяется метод главных компонент. Вследствие этого, детальная спектральная информация, содержащаяся в ГСИ, оказывается в таких представлениях утраченной.

Наибольшей информативностью обладают *спектрально-пространственные* признаки, учитывающие как детальную спектральную, так и пространственную информацию, содержащуюся в ГСИ. Учет обоих типов информации реализуется, как правило, путем дополнения спектральной информации каждого пикселя признаковой информацией о его локальной окрестности. В качестве признаков могут выступать непосредственно компоненты исходного изображений, компоненты спектральных преобразований, результаты фильтрации с определенными параметрами, выбранные статистические или структурные характеристики и т.д., обеспечивающие наилучшие показатели качества решаемой задачи.

Так как общий объем признаковой информации при использовании описанного подхода превышает объем исходных гиперспектральных данных, а признаки оказываются сильно избыточными, после расчета признаковой информации может производиться отбор подмножества информативных признаков (см. рисунок 1.1) с использованием одного из методов управляемого отбора [346, 80, 32, 142, 324].

Рисунок 1.1 – Учет спектральных и пространственных признаков изображений

С целью снижения объема обрабатываемых данных может производиться переход от исходного ГСИ к его представлению в виде суперпикселей или сегментов. В этом случае необходимые признаки извлекаются не из патчей (гиперкубов) исходного ГСИ, соответствующих различным положениям скользящего окна, а из

трехмерных участков, соответствующих суперпикселям или сегментам. Спектральная же информация, как правило, усредняется внутри различных суперпикселей (сегментов), что ведет к утрате отличий между пикселями внутри суперпикселей (сегментов).

К отдельному классу признаков можно отнести *признаки формы*. Для использования таких признаков сначала выполняется сегментация исходного ГСИ, а уже затем выделенные на изображении сегменты описываются с использованием тех или иных признаков формы. В качестве признаков формы могут использоваться:

- геометрические признаки (площадь, периметр, компактность, эксцентриситет и т.д.);
- моментные характеристики (моменты Ху, Зернике и др.);
- дескрипторы контуров (Фурье, цепные коды и т.д.);
- морфологические признаки (простые – количество связных компонент, число Эйлера; морфологические профили);

В целом, признаки формы сложно отнести к базовой признаковой информации, т.к. для извлечения таких признаков требуется выполнить сегментацию или классификацию исходного ГСИ. Такие признаки могут использоваться на более поздних этапах семантического анализа, но их полезность сильно зависит от качества сегментации.

Особого внимания заслуживает активно развивающееся в последние годы направление искусственных нейронных сетей. В анализе ГСИ нейросети чаще всего применяются для решения задачи попиксельной классификации ГСИ. Наиболее широко исследовались нейросетевые архитектуры, основанные на сверточных нейронных сетях (convolutional neural networks, CNN) [150]. В [117] была предложена модель на основе одномерной CNN (1D-CNN) для извлечения спектральной информации. Однако больший интерес представляют нейросетевые модели, позволяющие учитывать как спектральные, так и пространственные характеристики, например, описанные в [43]. Такой учет может быть реализован за счет повышения размерности сверточных сетей. Такая модель 3D-CNN была предложена в работе [159]. Другим способом является слияние информации, реализованное, например, в

виде двухканального слияния спектральной и пространственной информации (Two-CNN) [318] или многоканальной сверточной нейронной сети (MCCNN) [42]. Помимо моделей, работающих с исходной спектрально-пространственной информацией, были предложены модели, позволяющие использовать в качестве входных данных также производные признаки [85].

Кроме того, были предложены методы, в которых извлечение признаков выполняется с использованием ансамблей сверточных нейронных сетей одного типа (однородных), работающих, например, на случайных подпространствах исходного спектрального пространства [45, 104], либо с использованием ансамблей нейросетей различных типов (гетерогенных) [217].

Исследовались многомасштабные методы, в которых спектрально-пространственные признаки рассчитываются нейросетями для различных масштабов (размеров окна) с последующим объединением и классификацией [254, 178, 134].

Другой класс нейросетевых архитектур, исследуемых применительно к гиперспектральным изображениям, основан на графовых нейросетях (graph neural network, GNN). В таких сетях изображение рассматривается как плоский граф, где вершины представляют собой некоторые пространственные области изображения (в частном случае, непосредственно пиксели изображения), а ребра соединяют смежные области. В области анализа гиперспектральных изображений было предложено рассматривать различные способы построения графов. В работе [181] в качестве вершин рассматривались отдельные пиксели изображения. В работе [242] использовались суперпиксели (однородные компактные группы пикселей). В работе [112] использовались отобранные фрагменты изображения.

Помимо описанных выше нейросетей для анализа гиперспектральных изображений используются также и другие нейросетевые архитектуры, такие как сети с механизмами внимания [158, 238], рекуррентные нейронные сети (RNN) [160], сети глубокого доверия (DBN) [155], сети Колмогорова-Арнольда (KAN) [79*] и др.

Хотя описанные классы нейросетей нацелены на обучение и последующее решение конкретных прикладных задач (чаще всего, задачи попиксельной классификации), такие сети могут быть использованы для извлечения признаковой информации. Для этого из обученной для решения прикладной задачи сети отбрасывают классифицирующий блок (как правило, один или несколько последних

полносвязных слоев сети) и используют выход такой усеченной сети для генерации признаков.

С точки зрения классификации методов формирования признаков к классу спектральных признаков можно отнести признаки, извлекаемые одномерными сверточными нейросетями (1D-CNN); к классу пространственных признаков – признаки, извлекаемые двумерными сверточными нейросетями (2D-CNN); к классу спектрально-пространственных признаков - признаки, извлекаемые с использованием трехмерных сверточных моделей (3D-CNN), нейросетей на основе двух- и многоканальных архитектур (Two-CNN, MCCNN), трансформерных архитектур и т.д.

Необходимость настройки большого количества параметров в нейросетях является фактором, сдерживающим практическое применение нейросетей при анализе гиперспектральных данных. Одна из причин этого заключается в ограниченности обучающих данных, возникающей вследствие сложности разметки таких данных в различных областях. Например, в области дистанционного зондирования Земли получение таких данных связано с выездом и проведением измерений на местности; в медицине – с проведением гистологических исследований и т.п. Таким образом, получение обучающих данных является достаточно трудоемкой и дорогостоящей процедурой, а обучающие выборки весьма ограничены по объему.

По этой причине для обучения нейронных сетей, в особенности нейросетей глубокого обучения, исследователями применялись различные способы, позволяющие снизить требования к объему обучающего материала, например, трансферное обучение [318, 104, 317], метаобучение [84] и т. д.

К сожалению, даже при использовании указанных способов, остается ряд ограничивающих факторов. Один из этих факторов заключается в высокой специфичности гиперспектральных данных. На рынке существует широкий ассортимент гиперспектральных камер, построенных по различным схемам. Некоторые камеры представляют собой опытные образцы или существуют в единичных экземплярах. Соответственно, гиперспектральные снимки, получаемые с различных камер, обладают своими особенностями: различное число спектральных каналов, различный спектральной диапазон, различное пространственное разрешение и т.п.

Другой фактор заключается в специфичности решаемых задач. Как правило, распространенные модели нейросетей обучены на огромных массивах данных решению достаточно распространенных задач компьютерного зрения, например, классификации широкого класса часто встречающихся объектов: людей, автомобилей, кошек, собак, велосипедов и т.п. Задачи, решаемые при использовании гиперспектральных данных, напротив, высокоспецифичны. Например, требуется выполнить классификацию определенных сортов растений, находящихся на некотором этапе развития, или определить степень поражения ткани определенным заболеванием.

Кроме того, при гиперспектральной съемке проявляется также проблема спектральной изменчивости – зависимость получаемых спектральных характеристик от условий. Для сельскохозяйственных задач такими условиями могут быть: освещение, влажность, фаза развития растений, условия роста растений, для некоторых растений время суток, погодные условия и т.д.

Указанные факторы в совокупности приводят к тому, что существующие наборы размеченных гиперспектральных данных имеют небольшой объем, высокоспецифичны и предназначены для решения узкого круга задач. Такие наборы данных не могут использоваться для получения достаточно универсальных нейросетевых моделей, имеющих архитектуры, эффективно учитывающие природу гиперспектральных изображений (например, указанные выше архитектуры, извлекающие спектрально-пространственные признаки).

По указанным причинам для генерации информативных признаков ГСИ важной представляется возможность использования готовых моделей, предназначенных для работы с цветными трехканальными изображениями, и обученных на большом объеме таких изображений. При наличии такой возможности получение компактного описания гиперспектрального изображения с использованием методов снижения размерности и извлекаемых нейросетью признаков может быть полностью выполнено без обучающей разметки поступающих гиперспектральных данных.

Классификация описанных в настоящем разделе признаков гиперспектральных изображений представлена на рисунке 1.2. Основные категории приведенных рисунке

признаков также представлены в таблице 1.1 , где перечислены их достоинства и недостатки.

Рисунок 1.2 – Методы формирования признаков гиперспектральных изображений

Таблица 1.1 - Сравнение методов формирования признаков гиперспектральных изображений

Класс методов	**Достоинства**	**Недостатки**
Спектральные признаки	Простота реализации, интерпретируемость	Не учитывают пространственную информацию, содержащуюся в ГСИ
- исходные	Содержат всю исходную спектральную информацию	Большой объем и время обработки

Класс методов	**Достоинства**	**Недостатки**
- подмножества каналов и индексы	Малый объем, высокая скорость обработки, эффективность при решении частных задач	Малая информативность при решении других задач
- на основе 1D преобразований	Универсальность, компактное представление исходных данных	Утрата изначальной интерпретируемости признаков
Пространственные признаки	Содержат локальную пространственную информацию	Утрата богатой спектральной информации
Спектрально-пространственные признаки	Учет как спектральной, так и пространственной информации	
- трехмерные патчи/гиперкубы	То же	Избыточность
- признаки гиперпикселей /сегментов	То же	Утрата локальных спектральных отличий
Признаки формы	Признаки более высокого уровня для решения задач семантического анализа	Зависимость от качества семантической сегментации

Класс методов	Достоинства	Недостатки
Нейросетевые признаки	Учет как спектральной, так и пространственной информации	Необходимость наличия размеченных наборов данных значительного объема. Высокая трудоемкость или невозможность формирования таких выборок. Высокие вычислительные затраты при обучении. Не выполняют формирование компактного представления, восстановление исходной спектральной информации невозможно (за исключением автоэнкодеров).

Исходя из сказанного, наибольшей информативностью обладают спектрально-пространственные признаки. Именно признаки этого класса наиболее полно описывают гиперспектральные данные и позволяют достигать наилучших показателей качества решения прикладных задач. Однако признаки указанного класса, помимо исходных спектральных данных, несут в себе и извлеченную пространственную информацию, что еще больше увеличивает объем обрабатываемых и хранимых данных.

Таким образом, возникает необходимость устранения избыточности формируемой признаковой информации при сохранении ее информативности. Именно этому вопросу посвящен следующий пункт исследования.

1.2 Устранение избыточности гиперспектральных изображений

Высокая спектральная размерность гиперспектральных изображений обуславливает значительный объем таких изображений по сравнению с цветными или полутоновыми изображениями. Проблема особенно остро проявляется в

приложениях дистанционного зондирования Земли по причине значительных пространственных размеров регистрируемых изображений.

Как уже указывалось, значительный объем гиперспектральных данных обуславливает чрезвычайно высокую стоимость и временные затраты как на передачу и хранение, так и на обработку таких данных. При этом гиперспектральные изображения, так же как полутоновые и цветные изображения, обладают высокой степенью пространственной корреляции, а вследствие высокой спектральной размерности, обладают, кроме того, высокой степенью корреляции спектральных компонент. По указанным причинам гиперспектральные данные обладают значительной избыточностью.

Естественным способом решения проблемы значительного объема гиперспектральных изображений является **сжатие (компрессия) гиперспектральных изображений**. Решению проблемы сжатия посвящено множество статей и монографий [180, 7, 73]. Различают сжатие без потерь и сжатие с потерями [347]. В первом случае гиперспектральное изображение восстанавливается точно, во втором – с некоторой ошибкой.

Сжатие без потерь, к сожалению, не позволяет существенно снизить объем данных. Даже при учете межканальной корреляции сжатие без потерь по различным оценкам обеспечивает коэффициент сжатия порядка 3-5 [341, 132].

По этой причине большое практическое значение имеют методы сжатия с потерями. Последние, в зависимости от используемых принципов, в основном можно разделить на два больших класса: методы на основе кодирования с преобразованием и методы на основе предсказания (или дифференциального кодирования).

В методах на основе кодирования с преобразованием данные изображения преобразуются в пространство с другим базисом, выбираемым так, чтобы за большую часть информации в изображении отвечало наименьшее число коэффициентов преобразования, после чего выполняется квантование и энтропийное кодирование коэффициентов. К указанной группе методов относятся: методы на основе главных компонент или преобразования Карунена-Лоэва [347, 70], методы на основе трехмерных вейвлет-преобразований [135, 234] и ICER 3D [124], методы на основе трехмерного косинусного преобразования [237], метод на основе построения системы

«хорошо приспособленных» базисных функций [336], метод на основе расчета матриц квантования и оценки порядка сканирования квантованных коэффициентов для каждого изображения с предварительным удалением низкочастотных компонент [325] и др.

В методах на основе дифференциального кодирования [359, 176, 303] выполняется предсказание значений пикселей изображения по соседним пикселям с последующим квантованием и кодированием ошибки предсказания. Среди подобных подходов интерес представляет метод сжатия [9, 86, 337], основанный на иерархической сеточной интерполяции (HGI). В этом методе формируется неизбыточное иерархическое представление исходного изображения таким образом, что при обработке каждого уровня (кроме первого) значения пикселей интерполируются по пикселям предыдущего уровня, а наборы интерполяционных ошибок (остатков) квантуются и сжимаются энтропийным кодировщиком.

Каким бы ни был метод сжатия, для решения прикладных задач с использованием сжатых гиперспектральных изображений необходимо предварительно восстановить сжатые данные, что сопряжено с некоторыми накладными расходами. Вместе с тем эффективность решения прикладных задач остается неизменной при сжатии без потерь, в то время как при сжатии с потерями эффективность решения прикладных задач может существенно снизиться [332, 356].

При этом хотелось бы получить **такое компактное представление гиперспектральных изображений, которое дало бы возможность, с одной стороны, снизить объем передаваемых (в том случае, когда снижение размерности данных производится до их передачи) или хранимых данных, а с другой стороны, позволило бы производить обработку и анализ такого компактного представления (без восстановления полного изображения) за меньшее время без потери качества.**

Для формирования указанного представления можно использовать некоторую модель гиперспектрального изображения и вместо непосредственно гиперспектрального изображения работать в пространстве параметров этой модели.

Существуют различные **модели гиперспектральных изображений** [67, 107]. Во многих из них предполагается, что спектр любого пикселя может быть построен из некоторого числа спектров, соответствующих чистым компонентам (материалам), присутствующим на сцене.

Под спектром здесь понимается вектор числовых значений, характеризующих испускаемый или отраженный свет, где каждая компонента вектора соответствует определенному диапазону длин волн.

Наиболее простой и широко распространенной является линейная модель спектральной смеси (см., например, [3, 269, 4, 270]), в которой спектр каждого пикселя представляет собой линейную комбинацию элементарных спектров с аддитивным вектором шума. Вес каждого элементарного спектра определяется долей площади соответствующего компонента (материала), содержащегося на соответствующем пикселю участке сцены. Такая модель корректно описывает сцены, в которых падающие лучи света взаимодействуют только с одним компонентом (материалом), прежде чем попасть на регистрирующий излучение датчик.

Во многих случаях отраженный луч света до попадания на регистрирующий датчик претерпевает множественные отражения, часто от различных материалов, и линейная модель не является адекватной. По этой причине были предложены и применяются более сложные модели, чем линейная.

Наиболее простым является учет только вторичных отражений. Такие модели, в которых предполагается, что луч света претерпел не более двух отражений, называют билинейными. Можно выделить билинейные модели с двумя элементарными спектрами [218, 268], в том числе, модель с учетом спектра пропускания [326], а также билинейные модели с множеством конечных элементарных спектров. Последние различаются между собой вариантами коэффициентов при билинейных компонентах и накладываемыми на коэффициенты ограничениями. В частности, был предложен ряд моделей, в которых не учитывались билинейные самовзаимодействия: модель Насименто [272], Фэна [75], обобщенная билинейная модель (GBM) [97], а также билинейные модели, учитывающие билинейные самовзаимодействия: полиномиальная постнелинейная модель (PPNM) [8], модель Чена [44], линейноквадратичная модель Меганем [173].

Другим классом исследуемых моделей гиперспектральных изображений являются модели перемешанных смесей твердых частиц или капель в смесях или эмульсиях. В таких смесях свет многократно взаимодействует с частицами, прежде чем достичь наблюдателя гиперспектральных изображений. Предполагается, что при каждом взаимодействии фотон может быть поглощен, либо рассеян в случайном направлении. Начиная с начала 30-х гг. XX века было предложено множество моделей, однако наибольшую популярность приобрела модель Хапке [98]. Существует множество модификаций и улучшений этой модели (описание таких модификаций можно найти, например, в [264]), а также ряд альтернативных подходов, основанных на уравнении переноса излучения, описывающем распространение излучения через среду (см., например, [177, 265]).

Перспективным направлением исследований стало совмещение линейной модели спектральной смеси с моделью Хапке. Так, например, в [107, 48] предполагается, что в реальных сценах присутствуют отделенные друг от друга участки, представляющие собой участки хорошо перемешанных частиц.

Как видно даже из приведенного краткого обзора (не претендующего на полноту), на сегодняшний день предложено большое количество моделей спектрального смешивания. Выбор модели для произвольной реальной сцены может стать отдельной непростой задачей; каждая из моделей вводилась, исходя из некоторых ограничений и допущений, которые, в особенности для более простых моделей, могут легко нарушаться в реальных сценах.

Определение параметров для выбранной модели спектрального смешивания также представляет в общем случае серьезную проблему по причине отсутствия достоверной априорной информации о качественных характеристиках сцены (приходится решать задачу слепого оценивания компонент спектральной смеси и их количества), большого количества свободных параметров, чувствительности процедур оценивания к шумам [106]. Дополнительную сложность создает отсутствие открытых тестовых наборов гиперспектральных данных в объемах, достаточных для настройки и оценивания таких методов.

Перечисленные причины приводят к практической невозможности использования модельного подхода для устранения избыточности и формирования

компактного представления гиперспектральных изображений, о котором шла речь выше.

Однако в литературе были представлены еще два способа устранения избыточности и построения компактного представления гиперспектральных изображений: управляемый отбор спектральных компонент [279] и неуправляемое снижение размерности гиперспектральных данных [139, 274]. Первый из указанных способов формирует некоторое подмножество спектральных компонент исходного гиперспектрального изображения, второй – выполняет преобразование данных гиперспектрального изображения, так что спектральная размерность изображения снижается. Оба указанных способа (так же как и сжатие с потерями) неизбежно приводят к некоторой потере спектральной информации при сохранении пространственного разрешения изображений. Однако, как показывает анализ литературы [279, 274], исследователям удалось достичь достаточно высокого качества классификации гиперспектральных изображений при использовании обоих из них.

Управляемый отбор спектральных компонент может быть осуществлен с использованием одного из методов отбора признаков. При этом в качестве признаков здесь рассматриваются непосредственно спектральные компоненты (каналы) гиперспектрального изображения. Термин «управляемый» («supervised») здесь означает то, что для работы таких методов используется информация о распределении пикселей по классам. Такая информация может использоваться для оценки отдельных признаков (спектральных компонент) или их комбинаций; оцениваться может непосредственно качество решения какой-то задачи (чаще всего попиксельной классификации) или некоторые характеристики самого признака.

Наиболее очевидным решением является полный перебор всевозможных комбинаций спектральных компонент. Однако такое решение практически нереализуемо, поскольку требует непомерно большого количества времени. Используемые на практике методы отбора можно разделить на несколько классов: методы поиска субоптимального набора признаков; методы, основанные на фильтрации признаков и комбинированные методы.

В методах поиска субоптимального набора признаков выполняется построение, оценка и поиск субоптимальных комбинаций признаков. К методам этого класса относятся методы последовательного присоединения и отбрасывания признаков [346, 339*], рекурсивные методы отбора признаков, а также метаэвристические методы отбора признаков, основанные на генетических алгоритмах [266], алгоритмах на основе роя частиц [314], колонии муравьев [6], поиска кукушки [235] и др. Во всех случаях подобные методы требуют значительных вычислительных затрат из-за многократно повторяющихся этапов обучения и оценки качества, но при этом обеспечивают нахождение субоптимального набора признаков.

Более эффективны в вычислительном плане методы фильтрации признаков. Решение о включении признака в набор в таких методах принимается исходя из важности (информативности) отдельных признаков, оцениваемой на основе некоторых статистических характеристик. В качестве таких характеристик используются: дисперсия (отбрасываются признаки с низкой дисперсией), средняя абсолютная разница, отношения межклассовых и внутриклассовых расстояний [80], важность перестановочного признака [32] (характеризует, насколько качество решения падает при случайной перестановке признака), оценка признаков рельефа [142] (оценка признака увеличивается, если разница в значениях признака наблюдается для соседних экземпляров из разных классов, но снижается, если разница наблюдается для пары соседей того же класса), коэффициент корреляции (выбираются признаки с высокой корреляцией с целевым показателем и низкой корреляцией с другими признаками), симметрическая неопределенность (основана на энтропии Шеннона и используется в алгоритме быстрой фильтрации на основе корреляции [324]) и др.

Возможно использование комбинированного подхода, когда на начальном этапе признаки выбираются с использованием фильтрации, а затем выполняется поиск субоптимального набора признаков с использованием одного из методов первого класса. В некоторых случаях отбор признаков интегрирован с методом обучения. Примерами таких методов являются случайный лес [32] и дерево решений [260].

По ряду объективных причин, таких как большие вычислительные затраты, малая устойчивость к изменениям сцены, необходимость наличия

классифицированного множества, методы отбора спектральных компонент при анализе гиперспектральных изображений уступают методам снижения размерности (см. ниже). И хотя при использовании методов снижения размерности данных, исходная интерпретация данных (спектральных сигнатур) оказывается утраченной, последние чаще используются при работе с гиперспектральными изображениями.

Методы снижения размерности выполняют преобразование данных из многомерного пространства в пространство более низкой размерности, по возможности, сохраняя значимые свойства исходных данных.

При применении таких методов к гиперспектральному изображению I_X последнее рассматривается как набор пикселей $X=\{x_i\}$, $i=1..N$ - векторов в многомерном спектральном пространстве. Применение к набору пикселей некоторого метода снижения размерности позволяет получить соответствующий набор векторов $Y=\{y_i\}$, $i=1..N$ в редуцированном пространстве меньшей размерности, который можно трактовать, как изображение I_Y, представляющее собой более компактное представление изображения I_X.

Как правило, при выполнении преобразования пространственное положение пикселей не учитывается, и пиксели преобразуются в этом смысле независимо друг от друга. При этом сформированное таким образом изображение I_Y сохраняет те же пространственные отношения между пикселями, которые имели место в I_X, но пиксели полученного изображения находятся в ином редуцированном пространстве, в общем случае не являющимся подпространством исходного спектрального пространства. Таким образом, полученное изображение может рассматриваться как синтезированное многоканальное изображение, представляющее ту же сцену с тем же пространственным разрешением.

Хотя снижение размерности гиперспектрального изображения может, с некоторой точки зрения, рассматриваться как сжатие с потерями, мы будем отличать снижение размерности от сжатия (компрессии), так как полученное изображение I_Y, в отличие от результата сжатия, не требует последующего восстановления (декомпрессии) и может непосредственно использоваться для визуализации, классификации, сегментации или решения других прикладных задач.

Для целей снижения размерности гиперспектральных данных используются как линейные, так и нелинейные методы снижения размерности. Наиболее часто используются линейные методы, включая метод главных компонент (Principal Component Analysis - PCA) [232, 114], осуществляющий поиск линейной проекции в подпространство меньшей размерности, максимизирующей разброс данных. Примеры применения PCA к анализу гиперспектральных данных можно найти во многих работах, например, [243, 165]. Другим известным линейным методом является анализ независимых компонент (Independent Component Analysis - ICA) [49, 123], примеры его применения можно найти в публикациях [305, 163, 165].

Нелинейные методы снижения размерности до сих пор использовались реже, хотя, как указывалось выше, гиперспектральные изображения дистанционного зондирования подвержены нелинейным эффектам в связи с многократными переотражениями, рассеянием и другими причинами [11]. Нелинейные методы снижения размерности были основаны на нелинейном отображении [257], нелинейном методе главных компонент на основе ядра (KernelPCA) [259], анализе криволинейных компонент (Curvilinear component analysis – CCA [63], Curvilinear distance analysis – CDA) [152], изометрическом отображении (ISOMAP) [290], локально-линейном вложении (Locally-linear embedding - LLE) [252], методе лапласианских собственных карт (Laplacian eigenmaps - LE) [18], методе локальных касательных пространств (Local Tangent Space Alignment - LTSA) [327], стохастическом вложении соседей (Stochastic Neighbor Embedding, SNE) [298], методе аппроксимации и проекции однородного многообразия (Uniform Manifold Approximation and Projection, UMAP) [170] и некоторых других методах.

В последние несколько лет популярность нелинейных методов снижения размерности данных непрерывно растет. Было показано [179, 66, 188*, 156], что, в отличие от линейных методов снижения размерности, нелинейные методы позволяют, например, повысить точность классификации подстилающей поверхности и обнаружения объектов. В области визуализации гиперспектральных изображений эти методы также позволяют получить ложноцветовые (псевдоцветовые) представления с желаемыми свойствами.

На сегодняшний день можно указать целый ряд работ, в которых нелинейные методы снижения размерности применялись при обработке мульти- и

гиперспектральных изображений. Так, методы CCA и CDA использовались в [131, 154] для снижения размерности мультиспектральных изображений. Метод LLE использовался в [140] и позднее использовался вместе с методом LE в работе [261]. Более поздние примеры использования методов LLE, LE и LTSA для анализа гиперспектральных изображений можно найти в работах [315,281,113]. Метод ISOMAP применялся в работе [247] и позднее совместно с методом оптимизации SMACOFF [60] в работе [228]. Метод KernelPCA использовался в работе [165], метод t-SNE – в работе [64], метод UMAP – в работе автора [198*], а позднее - в работе [156].

В большинстве случаев указанные методы снижения размерности применялись как один из этапов решения задачи попиксельной классификации гиперспектральных изображений. Реже такие методы использовались при решении других задач, например, сегментации [179, 211*], визуализации [130, 54], определения компонент спектральной смеси [110, 286], обнаружения целей [127], криминалистике [64].

Во многих из указанных выше нелинейных методов требуется измерять рассогласование между пикселями гиперспектрального изображения. Наиболее часто используемой функцией расстояния для этой цели является евклидово расстояние. Реже в качестве функции расстояния использовался спектральный угол (Spectral Angle Mapper, SAM) [147].

В частности, можно указать ряд работ, в которых SAM использовался с нелинейными методами снижения размерности. Наиболее ранней работой, по всей видимости, следует считать [10], где SAM использовался вместе с евклидовым расстоянием на первом этапе метода ISOMAP для определения соседних пикселей в спектральном пространстве. В работе [316] авторы изучают эффективность SAM и евклидова расстояния для снижения размерности с использованием метода лапласианских собственных карт. В работе [315] SAM использовался для улучшения метода лапласианских собственных карт для снижения чувствительности к пропускам данных во временных последовательностях мультиспектральных спутниковых изображений.

Гораздо менее изученными в смысле совместного использования с нелинейными методами снижения размерности являются другие функции расстояния, например, дивергенция спектральной информации (Spectral Information Divergence,

SID) [40] и меры сходства, например, спектральная корреляция (Spectral Correlation Measure, SCM) [25]. При этом было показано [145], что такие функции расстояния имеют ряд преимуществ перед евклидовым расстоянием в области анализа гиперспектральных изображений. Известным автору примером применения SID с нелинейным методом снижения размерности (ISOMAP) для анализа гиперспектральных изображений служит работа [69].

Исходя из сказанного и учитывая практически полное отсутствие работ, посвященных анализу использования неевклидовых функций расстояния (за исключением SAM) при снижении размерности гиперспектральных данных, развитие этого направления исследований представляется целесообразным.

Следует отметить, что существенной проблемой нелинейных методов снижения размерности является их длительное время работы. По этой причине, с одной стороны, исследования часто проводятся на небольших фрагментах гиперспектральных изображений, а, с другой стороны, предпринимаются попытки ускорения работы таких методов.

Так как в методе ISOMAP существенное время занимает расчет геодезических расстояний на основе кратчайших путей, то ускорения можно добиться за счет использования так называемых ориентиров (landmarks), что реализовано в методе landmark ISOMAP [267]. Для локальных нелинейных методов, таких, как LLE, LE, LTSA одним из затратных по времени шагов является поиск ближайших соседей пикселей в исходном гиперспектральном пространстве, для ускорения которого применяются древовидные структуры данных или пространственное хеширование. В частности, kd-деревья применялись в [321] для ускорения LLE, а хэширование (locality-sensitive hashing, LSH) - в работе [316] для ускорения LE.

Другим подходом к ускорению является использование высокопроизводительных аппаратных средств, таких как программируемые пользователем вентильные матрицы – FPGA (один из типов программируемых логических интегральных схем - ПЛИС) или графические укорители (GPU). Так, FPGA применялись для ускорения UMAP [37] и линейного метода анализа независимых компонент ICA [68]. GPU, в свою очередь, применялись для методов ISOMAP [34], LLE [34, 321], LE [34], UMAP [288]. В целом, вопросам аппаратного

ускорения методов снижения размерности гиперспектральных изображений пока посвящены лишь единичные работы.

Важным фактом является то, что рассмотренные выше методы снижения размерности при их использовании для обработки гиперспектральных изображений действуют только в спектральном пространстве. Однако гиперспектральные изображения содержат, как спектральную, так и пространственную информацию. И последний тип информации остается неиспользуемым в традиционных методах снижения размерности. Несмотря на то, что можно указать на множество статей, в которых пространственная информация используется при анализе изображений, в частности, гиперспектральных (обзор по этой теме может быть найден в [306]), работ, посвященных проблеме использования пространственной информации гиперспектральных изображений в нелинейных методах снижения размерности, достаточно мало (см., например, [28, 280, 118]). По этой причине важным направлением развития нелинейных методов снижения размерности видится учет пространственной информации, содержащейся в гиперспектральных изображениях.

Наиболее активно развивающееся как в области компьютерного зрения, так и в целом искусственного интеллекта, направление **искусственных нейронных сетей** существенно повлияло и на область анализа гиперспектральных данных [230, 94].

Наиболее подходящим для решения задачи устранения избыточности представляется класс автоассоциативных нейронных сетей или автоэнкодеров [146]. Нейронные сети указанного класса позволяют формировать редуцированное представление входных данных на выходах нейронов промежуточного слоя. Такое представление может формироваться как на основе лишь спектральных характеристик (работы автора [185*, 191*], а также более поздние работы других исследователей, например [149, 165]), так и с учетом пространственного контекста [46, 320]. Следует отметить, что нейронные сети указанного класса часто относят к методам снижения размерности.

Другой тип нейросетей, который можно условно отнести к методам снижения размерности - самоорганизующиеся карты, также называемые нейросетями Кохонена [144]. Такие сети способны отобразить входные данные в дискретное пространство решетки, узлами которой являются нейроны сети, а также выполнить векторное

квантование входных данных. Этот тип сети применялся для снижения размерности гиперспектральных данных в работе [108].

Классификация рассмотренных выше подходов к устранению избыточности гиперспектральных изображений показана на рисунке 1.3.

Рисунок 1.3 – Подходы к устранению избыточности гиперспектральных изображений

Ниже в таблице 1.2 сведены основные достоинства и недостатки рассмотренных в настоящем разделе подходов к устранению избыточности гиперспектральных изображений.

Таблица 1.2 - Сравнение подходов к устранению избыточности гиперспектральных изображений

Класс методов	**Достоинства**	**Недостатки**
Методы сжатия:	Не требуют обучающего множества.	Требуют восстановления (декомпрессии) перед решением прикладных задач.
- без потерь,	Сохраняют исходную спектральную информацию.	Низкий коэффициент сжатия.
- с потерями.	Более высокий коэффициент сжатия.	Снижение качества решения прикладных задач.
Методы на основе моделей спектрального смешивания.	Учитывают особенности формирования гиперспектральных изображений.	Сложность выбора моделей и определения их параметров для реальных сцен. Недостаток открытых полноценных наборов данных для настройки и тестирования алгоритмов.
Методы отбора спектральных компонент.	Ориентация на качество решения заданных прикладных задач.	Необходимость наличия размеченных наборов данных. Невозможность восстановления исходной спектральной информации. Высокие вычислительные затраты.

Класс методов	Достоинства	Недостатки
Методы снижения размерности.	Не требуют обучающего множества. Возможность учета особенностей гиперспектральных изображений.	Невозможность восстановления исходной спектральной информации (для ряда методов). Высокие вычислительные затраты (для ряда методов).
Нейросетевые методы (на основе глубокого обучения).	Обеспечивают высокое качество решения заданных прикладных задач.	Необходимость наличия размеченных наборов данных значительного объема. Высокие вычислительные затраты при обучении. Не выполняют формирование компактного представления, восстановление исходной спектральной информации невозможно (за исключением автоэнкодеров).

Исходя из отмеченных достоинств и недостатков описанных выше подходов, представляется целесообразным выполнять построение компактных представлений гиперспектральных изображений на основе методов снижения размерности. В отличие от методов компрессии, такие методы позволяют формировать компактные представления гиперспектральных изображений, с которыми можно работать непосредственно, без необходимости восстановления (декомпрессии). В отличие от методов на основе глубокого обучения и методов отбора признаков, методы

снижения размерности являются неуправляемыми, а значит, не требуют разметки обучающих множеств, которая является особенно трудоемкой в задачах дистанционного зондирования Земли по причине необходимости выполнения наземных проб и измерений. В отличие от методов на основе моделей спектрального смешивания, нелинейные методы снижения размерности не требуют подбора конкретных моделей спектрального смешивания под регистрируемую сцену и проверки ее адекватности, а также решения сложной задачи «слепого» определения ее параметров.

Имеющиеся недостатки методов снижения размерности, по мнению автора, не должны стать причиной отказа от выбранного направления исследований. Невозможность восстановления исходной спектральной информации, присущая большинству нелинейных методов, может быть проигнорирована в том случае, если качество решения прикладных задач окажется сопоставимым с качеством решений, получаемым по исходным гиперспектральным данным. Проблема высоких вычислительных затрат может быть решена с использованием алгоритмических или аппаратных средств, что будет показано в настоящей работе.

1.3 Классификация методов снижения размерности общего назначения

На сегодняшний день разработано достаточно большое количество методов снижения размерности данных общего назначения, которые можно классифицировать следующим образом.

По типу преобразования: на линейные и нелинейные. К линейным методам относятся: метод главных компонент (PCA), метод случайных проекций, метод классического многомерного шкалирования (MDS), линейные автоассоциативные нейронные сети (AANN) и некоторые другие методы. К нелинейным методам относятся: отображение Сэммона, метрическое многомерное шкалирование, самоорганизующиеся карты Кохонена (SOM), анализ криволинейных компонент (CCA), Isomap, метод развертки максимальной дисперсии (MVU), метод локально-линейного вложения (LLE), метод лапласианских собственных карт (Laplacian eigenmaps), метод выравнивания локальных касательных подпространств (LTSA), нелинейные AANN, эластичные главные графы и некоторые другие методы.

Указанные методы сведены со ссылками и оригинальными названиями в таблицу 1.3. Краткое описание вынесено в указанный в таблице подраздел приложения А. Классификация методов показана на рисунке 1.4.

Таблица 1.3 – Методы снижения размерности общего назначения

Название	**Оригинальное название**	**Ссылки**	**Описание**
Линейные методы			
Метод главных компонент	Principal component analysis, PCA	[104, 105],	Приложение А.1
Метод случайных проекций	Random Projection	[107, 108],	Приложение А.2
Классическое многомерное шкалирование	Classical multidimensional scaling, MDS		Приложение А.3
Линейные автоассоциативные нейронные сети	Linear Autoassociative Neural Network, AANN	[139, 140],	Приложение А.15
Нелинейные методы			
Отображение Сэммона	Sammon’s mapping	[257]	Приложение А.4
Метрическое многомерное шкалирование	Metric Multidimensional Scaling, MDS		Приложение А.3
Самоорганизующиеся карты Кохонена	Self-organizing maps, SOM	[355]	Приложение А.5
Анализ криволинейных компонент	Curvilinear Component Analysis, CCA	[63]	Приложение А.6,
Isomap	Isomap	[290, 289]	Приложение А.7
Kernel PCA	Kernel PCA	[259]	Приложение А.8

Название	**Оригинальное название**	**Ссылки**	**Описание**
Метод развертки максимальной дисперсии	Maximum Variance Unfolding, MVU	[307, 308]	Приложение А.9
Метод локально-линейного вложения	Locally Linear Embedding, LLE	[252]	Приложение А.10
Метод лапласианских собственных карт	Laplacian Eigenmaps	[18]	Приложение А.11
Метод выравнивания локальных касательных подпространств	Local Tangent Space Alignment, LTSA	[327]	Приложение А.12
Стохастическое вложение соседей	Stochastic Neighbor Embedding, SNE	[109]	Приложение А.13
Метод аппроксимации и проекции однородного многообразия	Uniform Manifold Approximation and Projection, UMAP	[170]	Приложение А.14
Нелинейные автоассоциативные нейронные сети	Nonlinear Autoassociative Neural Network, AANN / Autoencoder	[13, 146]	Приложение А.15
Эластичные главные графы	Principal Elastic Graphs	[90]	Приложение А.16

Нелинейные методы могут быть разделены на глобальные (MDS, ISOMAP и т.д.) и локальные (LLE, Laplacian Eigenmaps и т.д.). Первые нацелены на сохранение свойств данных на всех масштабах, в то время как вторые – на сохранение локальных свойств. Локальные методы обладают большей «репрезентативной емкостью»,

однако глобальные методы формируют более точное представление структуры данных в целом [267].

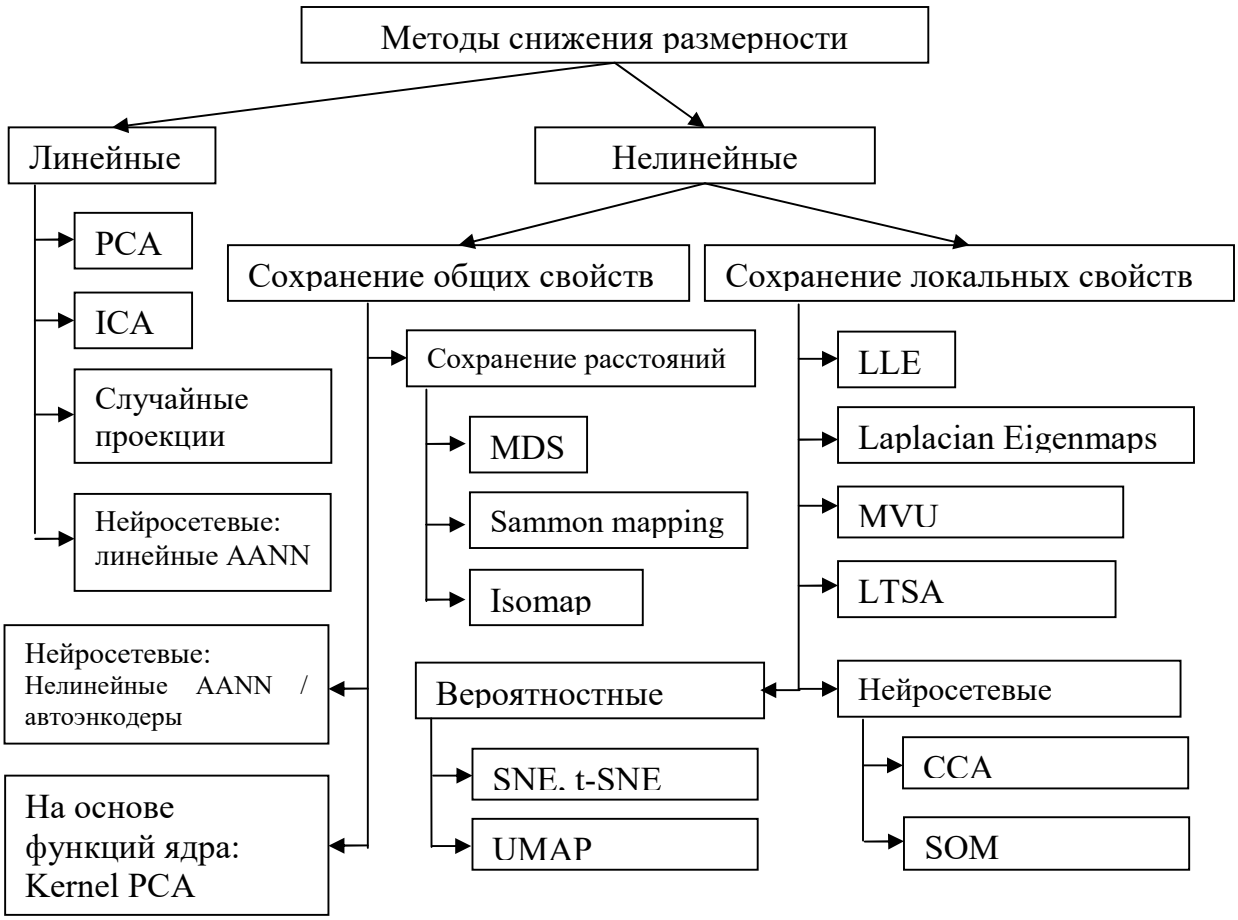


Рисунок 1.4 – Классификация методов снижения размерности

По используемым подходам к построению методов можно выделить: классические методы (PCA, случайные проекции, MDS, отображение Сэммона), методы, основанные на графовых моделях (Isomap, MVU, LLE, Laplacian eigenmaps, LTSA, эластичные главные графы), методы, основанные на нейросетях (SOM, CCA, AANN).

Говоря об особенностях методов снижения размерности, удобно рассматривать методы по группам. Описанным линейным методам присуща высокая вычислительная эффективность, однако они не способны обнаружить во входных данных нелинейные взаимосвязи. Нелинейные методы, напротив, более требовательны к вычислительным ресурсам, но могут быть способны более адекватно отразить структуру входных данных, когда данные находятся на некотором многообразии меньшей размерности.

Классические нелинейные методы стремятся отразить как глобальную, так и локальную структуру данных, используя для этого сразу весь набор входных данных, и по этой причине могут оказаться неспособны эффективно работать в случае входных данных сложной структуры. Предполагается, что методы, основанные на графовых моделях, наоборот будут способны работать в случаях, когда данные расположены на многообразиях со сложной геометрией. Однако использование жесткого соседства в таких методах требует тщательной настройки на набор данных, не гарантирует корректной работы методов, в тех случаях, когда набор данных имеет разномасштабную структуру. Отдельным вопросом является и разворачивание (англ., unfolding) такими методами имеющихся в данных глобальных структур, что ведет к значительной и необратимой потере информации о глобальной структуре, которая может быть весьма полезна при анализе.

Рассмотренные нейросетевые методы довольно сильно отличаются друг от друга. Так, SOM выполняет отображение входных данных в дискретное пространство своей кристаллической решетки и по этой причине его применение как самостоятельного метода снижения размерности ограничено. AANN, напротив, позволяют не только выполнить отображение входного набора данных в пространство сниженной размерности, но и являются единственным нелинейным методом из рассмотренных, позволяющим построить и прямое и обратное преобразование в явном виде (в виде архитектуры и настройки весов нейросети). Недостаток же AANN кроется в сложности интерпретации полученных в результате работы сети данных. Это является следствием того, что к редуцированным представлениям, порождаемым кодировщиком сети, не предъявляется никаких требований, кроме возможности восстановления исходного многомерного сигнала декодером. Другая проблема – подбор архитектуры сети, обеспечивающей одновременно и высокую точность восстановления и обобщающую способность, ведет к непомерным временным затратам по причине длительного обучения нейросети.

Таким образом, каждое из существующих решений обладает своими достоинствами и недостатками, а современное состояние исследований не позволяет говорить о том, что проблема снижения размерности получила удовлетворительное решение в общем случае.

Рассматривая проблему применения методов снижения размерности в контексте формирования информативных признаков гиперспектральных изображений можно констатировать факт отсутствия специализированных методов снижения размерности. В то же время методы снижения размерности общего назначения, преимущественно линейные, применяются при анализе таких изображений повсеместно.

В этой связи у приведенных выше методов снижения размерности общего назначения существует целый ряд недостатков. Во-первых, это отсутствие естественных механизмов учета содержащейся в изображениях пространственной информации, а также слияния (интеграции) спектральной информации с другими производными системами признаков, например, текстурными признаками или признаками, извлекаемыми с использованием нейросетей. Во-вторых, методы общего назначения имеют весьма ограниченные возможности работы с неевклидовыми функциями расстояния, что может делать их применение неэффективным с точки зрения качества решения некоторых прикладных задач. В-третьих, работа с гиперспектральными изображениями, в особенности высокого разрешения, характеризуется как исключительно большим количеством обрабатываемых объектов (пикселей, векторов), так и высокими размерностями входных пространств (спектральные пространства, системы с большим количеством признаков). Это делает высокую вычислительную сложность, присущую нелинейным методам снижения размерности общего назначения, непреодолимым препятствием для применения большинства таких методов.

По мнению автора настоящей диссертации, указанные недостатки могут быть преодолены путем создания специализированных методов и алгоритмов формирования информативных признаков ГСИ. Прежде чем перейти к принципам создания таких методов и алгоритмов, приведем постановку задачи формирования информативных признаков гиперспектральных изображений, принятую в настоящей диссертации.

1.4 Постановка задачи формирования информативных признаков гиперспектральных изображений

Пусть дано гиперспектральное изображение $I_X \in \mathrm{R}^{H \times W \times M}$, представляющее собой набор пикселей $X = \{x_1, x_2, \ldots x_N\}$, где H – высота, W – ширина ГСИ, $N=WH$ – количество пикселей, M – количество спектральных каналов (диапазонов) ГСИ, $M \geq 2$. Требуется сформировать изображение $I_Y \in \mathrm{R}^{H \times W \times m}$ того же пространственного разрешения $H \times W$ с пикселями $Y = \{y_1, y_2, \ldots y_N\}$, содержащими меньшее количество компонент $m < M$ (требование *компактности*), таким образом, чтобы качество решения задач анализа ГСИ по сформированному изображению I_Y было близко к качеству решения таких задач по исходному ГСИ I_X (требование *информативности* признакового представления) в смысле заданного критерия.

Например, для задачи попиксельной классификации задача формирования информативных признаков для изображения I_X сводится к поиску такого изображения I_Y (при $m<M$), которое максимизирует качество классификации $\mathrm{Q}(I_Y)$ при ограничении на размерность m, либо минимизирует m при условии $Q(I_Y) \geq \mathrm{Q}(I_X) - e$, где $e>0$ — допустимое снижение качества.

1.5 Принципы построения методов и алгоритмов формирования информативных признаков, принятые в настоящей работе

Предлагаемый в настоящей работе подход к построению методов и алгоритмов формирования информативных признаков основывается на введении в рассмотрение следующих базовых принципов, позволяющих преодолеть указанные выше недостатки:

1) *Сохранение выбранных расстояний* между отсчетами ГСИ. Значения выбранной функции расстояния δ между соответствующими пикселями исходного и формируемого изображений должны, по возможности, сохраняться: $\delta(x_i, x_j) \approx \delta(y_i, y_j)$, $i, j = 1 \ldots N$. Иными словами, близкие в спектральном пространстве в смысле выбранной функции расстояния отсчеты исходного ГСИ должны оставаться близкими в формируемом пространстве, а далекие отсчеты – далекими. Это позволит применять в сформированном признаковом пространстве существующие функции расстояния, эффективность которых доказана.

2) *Учет как спектральных, так и пространственных характеристик ГСИ*, а также *интеграция спектральной информации ГСИ с другими системами признаков* при формировании признаков. Если кроме спектральных признаков $I_X \in \mathrm{R}^{H \times W \times M}$ на вход процедуре формирования поданы также пространственные признаки $I_{X'} \in \mathrm{R}^{H \times W \times M'}$, выходное изображение $I_Y \in \mathrm{R}^{H \times W \times m}$, $m < M+M'$ должно формироваться на основе признаков обоих типов. Это позволит учитывать в формируемых представлениях пространственные взаимосвязи, содержащиеся в изображениях.

3) При работе с ГСИ, содержащими большое количество отсчетов, должна быть обеспечена возможность *разработки методов с пониженной вычислительной сложностью* и возможностью *управления вычислительной сложностью*, что позволит находить баланс между качеством получаемого решения и скоростью обработки данных.

4) Должна быть обеспечена возможность реализации *методов с использованием параллельных вычислительных сред*, что позволит эффективно использовать современные аппаратные средства для обеспечения высокой скорости обработки данных.

Поставленная описанным образом задача может решаться для любого отдельно взятого ГСИ, что обеспечивает *адаптивность* процедуры формирования к этому изображению.

Выводы и результаты по главе 1

В настоящей главе выполнен обзор и классификация методов формирования признаков ГСИ; показано, что наибольшей информативностью обладают спектрально-пространственные признаки. Проведен анализ подходов, используемых для устранения избыточности ГСИ. В частности, рассмотрены методы сжатия (компрессии данных), модели спектральной смеси, методы отбора спектральных компонент, методы снижения размерности, нейросетевые методы; обосновано применение подхода на основе методов снижения размерности данных.

Выполнен анализ существующих методов снижения размерности данных общего назначения, рассмотрены особенности существующих методов и выполнена их классификация. Приведены недостатки методов снижения размерности данных

общего назначения при работе с ГСИ. Установлен факт отсутствия специализированных методов, предназначенных для работы с ГСИ.

Приведена постановка задачи формирования информативных признаков ГСИ, принятая в настоящей работе. Предложены и обоснованы базовые принципы к построению методов и алгоритмов формирования информативных признаков ГСИ.

2 Алгоритмические средства формирования признаков ГСИ, основанные на сохранении попарных расстояний

В настоящей работе мы используем понятие **расстояния**, понимая под ним выраженное неотрицательной вещественнозначной скалярной величиной различие между двумя точками (векторами) в некотором пространстве. Под **функцией расстояния** $\delta: \mathrm{R}^M \times \mathrm{R}^M \to \mathrm{R}$ будем понимать математическую функцию двух аргументов x_1 и x_2, представляющих собой точки (вектора) из R^M, возвращающую расстояние от точки x_1 до точки x_2.

Определенная таким образом функция расстояния в общем случае не является **метрикой,** которая обязана удовлетворять аксиомам неотрицательности, тождества, симметрии и неравенства треугольника [334]. Отметим, что в одних работах (см., например, [334, 335]) понятие расстояния используется наравне с метрикой, в то время как в других понятие расстояния используется в более широком смысле, и от него не требуется выполнения всех указанных выше аксиом. Так, может не выполняться аксиома тождества (для полуметрики, псевдометрики), симметричности (для квазиметрики) или неравенства треугольника (для семиметрики). В настоящей работе термин метрика будет использоваться в тех случаях, когда речь будет идти об истинных метриках, а в остальных случаях будет использоваться понятие функции расстояния.

В области анализа изображений используется исключительно широкий набор различных функций расстояния. Особенно много функций расстояния предложено в области анализа гиперспектральных изображений. В частности, рассматривались [246, 245, 116] такие функции как евклидово расстояние, нормализованное евклидово расстояние (Normalized Euclidean Distance, NED), спектральный угол (Spectral Angle, SA, Spectral Angle Mapper, SAM), дивергенция спектральной информации (Spectral Information Divergence, SID), спектральная корреляция, угол спектральной корреляции (Spectral Correlation Angle, SCA), угол спектрального градиента (Spectral Gradient Angle, SGA), условная энтропия и т.д.

Как уже указывалось в разделе 1.1, исследователями использовались некоторые из неевклидовых функций расстояния (в основном, SAM; в единственной работе - SID) при снижении размерности гиперспектральных данных. Интеграция

этих расстояний при этом, как правило, сводилась к их использованию для измерения рассогласования между пикселями во входном пространстве, а сами методы снижения размерности не претерпевали каких-либо модификаций. Такой подход, в целом, соответствует идеям, принятым в многомерном шкалировании (multidimensional scaling). При таком подходе предпринимается попытка приближения (аппроксимации) различий, измеряемых неевклидовыми расстояниями во входном спектральном пространстве, евклидовыми расстояниями в выходном пространстве меньшей размерности.

Однако задача снижения размерности в контексте неевклидовых расстояний может быть поставлена иначе и состоять в сохранении, по возможности, различий, измеряемых неевклидовыми расстояниями, при вложении данных в пространства более низкой размерности. Идея здесь состоит в том, что структура данных, определяемая попарными различиями между пикселями (точками) в спектральных пространствах высокой размерности и обеспечивающая высокие качественные показатели при решении прикладных задач, должна быть, по возможности, сохранена и при вложении в пространства более низких размерностей.

Руководствуясь приведенной постановкой, в настоящей главе выполняется разработка алгоритмов формирования информативных признаков ГСИ для ряда неевклидовых функций расстояния, а также разрабатываются соответствующие алгоритмы формирования признаков, основанные на приближении неевклидовых расстояний евклидовыми расстояниями.

Раздел построен следующим образом. В разделе 2.1 приводится обзор функций расстояния, используемых в настоящей работе. В разделе 2.2 выполняется разработка алгоритмов формирования информативных признаков, основанных на сохранении заданных расстояний, а в разделе 2.3 – алгоритмов, основанных на приближении заданных расстояний евклидовыми расстояниями. Раздел 2.4 содержит результаты численных экспериментов.

2.1 Обзор функций расстояния, используемых при анализе гиперспектральных изображений

2.1.1 l_p расстояние

l_p расстояние (расстояние Минковского) [65] между двумя векторами x_i и x_j, соответствующими i-му и j-му пикселям гиперспектрального изображения, определяется с использованием l_p нормы $\|\cdot\|_p$ следующим образом:

$$d(x_i, x_j) = \|x_i - x_j\|_p = \left(\sum_{k=1}^{M} |x_{ik} - x_{jk}|^p \right)^{1/p}. \quad (2.1)$$

Это расстояние является истинной метрикой при значениях показателя $p \geq 1$.

Известными частными случаями являются: евклидово расстояние, определяемое на основе l_2 нормы; расстояние городских кварталов (расстояние Манхеттена), определяемое на основе абсолютной нормы l_1; а также расстояние Чебышева, определяемое на основе максимальной нормы l_∞.

$$\|x_i - x_j\|_\infty = \max_k |x_{ik} - x_{jk}|. \quad (2.2)$$

Самым часто используемым в анализе гиперспектральных изображений является евклидово расстояние.

2.1.2 Спектральный угол

Наиболее известной среди неевклидовых функций расстояния, используемых в анализе гиперспектральных изображений, является спектральный угол (Spectral Angle Mapper, SAM) [147]:

$$\theta(x_i, x_j) = \arccos \left(\frac{x_i^T x_j}{\|x_i\| \|x_j\|} \right). \quad (2.3)$$

Здесь x_i и x_j – два вектора, соответствующих i-му и j-му пикселям гиперспектрального изображения. SAM измеряет спектральное рассогласование как угол между двумя спектральными сигнатурами x_i и x_j независящим от амплитуды способом, что дает некоторую инвариантность к вариациям в освещении.

Отметим, что для указанной функции расстояния допускаются нулевые значения расстояния между различными точками, и спектральный угол является **полуметрикой** [334], или **псевдометрикой** [169]. Отметим также, что близкое по

смыслу и часто используемое в различных задачах косинусное расстояние метрикой не является, так как не удовлетворяет неравенству треугольника.

2.1.3 Спектральная корреляция

Спектральная корреляция [25] выражается следующим образом:

$$r(x_i, x_j) = \frac{K\sum_{k=1}^{K} x_{ik}x_{jk} - \sum_{k=1}^{K} x_{ik}\sum_{k=1}^{K} x_{jk}}{\sqrt{K\sum_{k=1}^{K} x_{ik}^2 - \left(\sum_{k=1}^{K} x_{ik}\right)^2}\sqrt{K\sum_{k=1}^{K} x_{jk}^2 - \left(\sum_{k=1}^{K} x_{jk}\right)^2}}. \quad (2.4)$$

Здесь x_i и x_j - пиксели гиперспектрального изображения, K – количество перекрывающихся спектральных диапазонов.

Эта мера часто записывается в другом виде, известном как коэффициент корреляции Пирсона:

$$r(x_i, x_j) = \frac{\sum_{k=1}^{K}\left(x_{ik} - \bar{x}_i\right)\left(x_{jk} - \bar{x}_j\right)}{\sqrt{\sum_{k=1}^{K}\left(x_{ik} - \bar{x}_i\right)^2}\sqrt{\sum_{k=1}^{K}\left(x_{jk} - \bar{x}_j\right)^2}}. \quad (2.5)$$

где $\bar{x}$ – среднее значение x.

В отличие от других рассмотренных в настоящем разделе функций, спектральная корреляция не является функцией расстояния, измеряющей рассогласование между точками (векторами), а является мерой подобия.

2.1.4 Дивергенция спектральной информации

Дивергенция спектральной информации (Spectral Information Divergence, SID) [40] является довольно популярной функцией расстояния в области анализа гиперспектральных изображений. Для пары векторов x_i и x_j эта функция задается следующим образом [39]:

$$\begin{gathered} SID\left(x_i, x_j\right) = D\left(x_i \| x_j\right) + D\left(x_j \| x_i\right), \\ D\left(x_i \| x_j\right) = \sum_{k=1}^{M} p_k(x_i)\log\left(p_k(x_i)/p_k(x_j)\right), \\ p_k(x_i) = \frac{x_{ik}}{\sum_{l=1}^{M} x_{il}}. \end{gathered} \quad (2.6)$$

Здесь $p(x) = (p_1(x),\ldots p_M(x))^T$ рассматривается как функция распределения вероятностей дискретной случайной величины, порождаемая вектором x, а $D(x_i||x_j)$ – усредненное по всем спектральным диапазонам различие собственной информации для вектора x_i по отношению к информации для вектора x_j (относительная энтропия).

Отметим, что дивергенция спектральной информации, называемая также дивергенцией Джефри, является симметричной функцией по построению.

2.1.5 Дивергенция Хеллингера

К сожалению, приведенная выше дивергенция спектральной информации не является истинной метрикой, так как для нее не выполняется неравенство треугольника [1] (при этом другим требованиям эта функция расстояния удовлетворяет). Нарушение неравенства треугольника имеет при практическом использовании существенное значение, так как делает невозможным применение индексных структур данных при выполнении точных запросов по ближайшему соседу (верные точки могут быть ошибочно пропущены во время поиска).

По указанной причине в настоящей работе рассматриваются две другие теоретико-информационные функции расстояния, являющиеся истинными метриками: дивергенция Хеллингера и угол Бхаттачария (расстояние Бхаттачария также не является истинной метрикой [133]). Автором было показано [192*], что указанные метрики обеспечивают сопоставимое с SID качество классификации гиперспектральных изображений.

Дивергенция Хеллингера или функция расстояния Хеллингера (Hellinger distance) [105] определяется выражением:

$$HD(x_i, x_j) = \sqrt{1 - \sum_k^M \sqrt{p_k(x_i) p_k(x_j)}} \,. \qquad (2.7)$$

2.1.6 Угол Бхаттачария

Угол Бхаттачария (Bhattacharyya angle) [22] задается следующим образом:

$$BCa(x_i, x_j) = \arccos\left(\sum_k^M \sqrt{p_k(x_i) p_k(x_j)}\right). \qquad (2.8)$$

где $p_k(x_i)$ определяется так же, как в формуле (2.6) при описании SID.

Угол Бхаттачария так же, как и дивергенция Хеллингера является истинной метрикой.

2.2 Метод построения алгоритмов формирования информативных признаков, основанный на сохранении заданных расстояний

Ниже предлагается метод построения алгоритмов формирования информативных признаков гиперспектральных изображений, основанных на принципе сохранения выбранных расстояний в редуцированном пространстве.

В соответствии с предлагаемым методом для построения алгоритма формирования информативных признаков, требуется выполнить следующие шаги:

1. Задать функционал ошибки ε представления гиперспектральных данных в формируемом пространстве признаков меньшей размерности. В общем случае такой функционал может строиться с использованием разных функций расстояния δ и ψ во входном и выходном (формируемом) пространствах, соответственно. В рассмотренных в настоящем исследовании методах в качестве такого функционала выступает рассчитанная по всевозможным парам пикселей сумма квадратов разности между попарными расстояниями между пикселями во входном и выходном пространствах:

$$\varepsilon = \mu \sum_{i,j=1\,(i<j)}^{N} \rho_{ij} \left(\delta(x_i, x_j) - \psi(y_i, y_j) \right)^2 . \qquad (2.9)$$

В этом выражении μ - постоянный масштабный коэффициент, ρ_{ij} - весовые множители, позволяющие адаптировать вид функционала к конкретной задаче.

В рамках настоящего подраздела 2.2 будем считать, что во входном и выходном пространствах используется одна и та же функция расстояния δ. Тогда функционал принимает следующий вид:

$$\varepsilon = \mu \sum_{i,j=1\,(i<j)}^{N} \rho_{ij} \left(\delta(x_i, x_j) - \delta(y_i, y_j) \right)^2 . \qquad (2.10)$$

2. Аналитически определить градиент $\nabla\varepsilon$ введенного функционала ошибки по координатам точек в формируемом пространстве признаков.

3. С использованием метода градиентного спуска реализовать численную процедуру оптимизации $\varepsilon \xrightarrow[Y]{} \min$ функционала ошибки:

$$Y^{[t+1]} = Y^{[t]} - \alpha \nabla \varepsilon (Y^{[t]}). \quad (2.11)$$

Здесь t – номер итерации, $Y^{[t]} = \left\{ y_1^{[t]}, y_2^{[t]}, ... y_N^{[t]} \right\}$ - искомые координаты точек в формируемом пространстве признаков меньшей размерности (см. п. 1.4) на шаге t, $\nabla \varepsilon$ - градиент целевой функции, α - коэффициент градиентного спуска.

4. Выбрать процедуру инициализации, задающую начальные значения координат $Y^{[0]}$ в редуцированном пространстве (управляемых переменных) перед началом процедуры оптимизации. В качестве такой процедуры может использоваться метод главных компонент, инициализация выбранными компонентами или другие подходы, задающие начальное приближение.

В целом разработанный подход агрегирует как хорошо известное отображение Сэммона [257], так и разработанные автором алгоритмы формирования информативных признаков ГСИ, основанные на сохранении спектральных углов, спектральной корреляции, дивергенции спектральной информации и дивергенции Хеллингера [196*, 195*, 193*, 183*]. Таким образом, предложенный подход позволяет работать с любой из наиболее часто используемых мер спектрального рассогласования.

Хотя предложенный подход и включает как частный случай алгоритм [257] 1969 г., автору не известны работы, в которых он использовался бы с неевклидовыми мерами спектрального рассогласования, как неизвестны методы и алгоритмы снижения размерности, работающие по принципу сохранения неевклидовых мер.

Альтернативой описанному является подход, при использовании которого значения выбранной функции расстояния в гиперспектральном пространстве приближаются (аппроксимируются) евклидовыми расстояниями в редуцированном пространстве. В отличие от описанного выше, такой подход использовался с альтернативными методами снижения размерности, хотя такое приближение и ограничивалось использованием спектральных углов. О реализации методов, работающих по принципу приближения дивергенции спектральной информации, дивергенции Хеллингера или угла Бхаттачария евклидовыми расстояниями в редуцированных пространствах, автору неизвестно.

2.2.1 Алгоритм формирования признаков ГСИ, основанный на сохранении значений l_p метрики

Так как класс l_p метрик (см. п. 2.1.1) включает самую широко распространенную метрику – евклидово расстояние, разработку алгоритмов формирования признаков ГСИ, действующих по принципу сохранения попарных расстояний, начнем именно с этого класса метрик. Отметим, что автору не известно об использовании l_p метрик при $p\neq 2$ при работе с гиперспектральными изображениями.

Функционал ошибки представления гиперспектральных данных в формируемом пространстве меньшей размерности при использовании l_p метрик определим следующим образом:

$$\varepsilon = \mu \cdot \sum_{i=1}^{N} \sum_{j=i+1}^{N} \rho_{ij} \left(\left\| x_i - x_j \right\|_p - \left\| y_i - y_j \right\|_p \right)^2 . \tag{2.12}$$

Согласно представленному в разделе 2.2 методу, вычислим (см. Приложение Б.1) компоненты градиента ошибки (2.12):

$$\frac{\partial \varepsilon}{\partial y_{ik}} = -2\mu \cdot \sum_{\substack{j=1, \\ j\neq i}}^{N} \rho_{ij} \cdot \frac{\left\| x_i - x_j \right\|_p - \left\| y_i - y_j \right\|_p}{\left\| y_i - y_j \right\|_p^{p-1}} \cdot \left| y_{ik} - y_{jk} \right|^{p-1} \operatorname{sgn}\left(y_{ik} - y_{jk} \right).$$

Численная процедура оптимизации в соответствии с (2.11) примет вид:

$$y_{ik}^{[t+1]} = y_{ik}^{[t]} + 2\alpha\mu \sum_{\substack{j=1, \\ j\neq i}}^{N} \rho_{ij} \cdot \frac{\left\| x_i - x_j \right\|_p - \left\| y_i^{[t]} - y_j^{[t]} \right\|_p}{\left\| y_i^{[t]} - y_j^{[t]} \right\|_p^{p-1}} \cdot \left| y_{ik}^{[t]} - y_{jk}^{[t]} \right|^{p-1} \operatorname{sgn}\left(y_{ik}^{[t]} - y_{jk}^{[t]} \right), \quad i = 1...N. \tag{2.13}$$

Отметим, что при использовании в качестве констант $p = 2, \quad \mu = 1$, функционал ошибки (2.12) аналогичен функционалу (А.1), используемому в многомерном шкалировании. Нормализованная ошибка (stress) строится в работе по многомерному шкалированию [148] с использованием значений:

$$\mu = \frac{1}{\sum_{i=1}^{N} \sum_{j=i+1}^{N} d^2(x_i, x_j)}, \quad \rho_{i,j} = 1 . \tag{2.14}$$

При использовании следующих значений

$$p = 2, \quad \mu = \frac{1}{\sum_{i=1}^{N}\sum_{j=i+1}^{N} d(x_i, x_j)}, \quad \rho_{i,j} = \frac{1}{d(x_i, x_j)} \tag{2.15}$$

функционал ошибки (2.12) аналогичен выражению (А.2), используемому в отображении Сэммона, а процедура оптимизации (2.13) принимает вид:

$$y_i^{[t+1]} = y_i^{[t]} + 2\alpha \frac{1}{\sum_{i=1}^{N}\sum_{j=i+1}^{N} d(x_i, x_j)} \sum_{j=1(j\neq i)}^{N} \frac{d(x_i, x_j) - d(y_i^{[t]}, y_j^{[t]})}{d(x_i, x_j) d(y_i^{[t]}, y_j^{[t]})} \left(y_i^{[t]} - y_j^{[t]}\right), \tag{2.16}$$

$$i = 1 \ldots N,$$

аналогичный (А.3) и (А.4), без учета производных второго порядка.

2.2.2 Алгоритм формирования признаков ГСИ, основанный на сохранении спектральных углов

Как указывалось ранее, спектральный угол (spectral angle mapper) [147] часто используется как функция для измерения расстояния между пикселями изображения в гиперспектральном пространстве. Автором настоящей работы были предложены несколько алгоритмов формирования признаков [196*], в которых для измерения расстояний между пикселями используется именно указанная функция.

2.2.2.1 Алгоритм формирования признаков ГСИ с сохранением спектральных углов в декартовых координатах

Следуя описанному в разделе 2.2 методу, введем ошибку представления данных с учетом спектральных углов, выражаемую в следующем виде:

$$\varepsilon_{SAM_C} = \mu \cdot \sum_{i=1}^{N}\sum_{j=i+1}^{N} \rho_{ij} \left(\theta\left(x_i, x_j\right) - \theta\left(y_i, y_j\right)\right)^2 . \tag{2.17}$$

Отметим, что в этом и последующих пунктах настоящего раздела конкретный вид множителя μ указываться не будет, так как он является константой и не влияет на процесс поиска решения.

Применим алгоритм градиентного спуска для получения субоптимального решения. Последнее потребует вычисления частных производных ошибки представления данных с учетом спектральных углов. Выполнив некоторые простые

преобразования (см. Приложение Б.2), получим следующее решение для частных производных ошибки:

$$\frac{\partial \varepsilon_{SAM_C}}{\partial y_{ik}} = 2\mu \sum_{\substack{j=1,\\ j\neq i}}^{N} \rho_{ij} \cdot \left(\theta\left(x_i, x_j\right) - \theta\left(y_i, y_j\right)\right) \cdot \frac{y_{jk} - y_{ik} \left\langle y_i, y_j \right\rangle \Big/ \left\| y_i \right\|^2}{\sqrt{\left\| y_i \right\|^2 \left\| y_j \right\|^2 - \left\langle y_i, y_j \right\rangle^2}}.$$

Наконец, итеративное выражение для координат точек данных в выходном пространстве примет следующий вид:

$$y_{ik}^{[t+1]} = y_{ik}^{[t]} - 2\alpha\mu \sum_{\substack{j=1,\\ j\neq i}}^{N} \rho_{ij} \left(\theta\left(x_i, x_j\right) - \theta\left(y_i^{[t]}, y_j^{[t]}\right)\right) \frac{y_{jk}^{[t]} - y_{ik}^{[t]} \left\langle y_i^{[t]}, y_j^{[t]} \right\rangle \Big/ \left\| y_i^{[t]} \right\|^2}{\sqrt{\left\| y_i^{[t]} \right\|^2 \left\| y_j^{[t]} \right\|^2 - \left\langle y_i^{[t]}, y_j^{[t]} \right\rangle^2}}, \quad (2.18)$$

$i = 1..N,$

$k = 1..m.$

Последнее выражение определяет алгоритм, выполняющий формирование признаков ГСИ с сохранением спектральных углов, полученный в декартовых координатах (Spectral Angle Preserving Mapping in Cortesian coordinates, SAPM-C) [196*].

Несложно показать, что вектор коррекции $y_i^{[t+1]} - y_i^{[t]}$ ортогонален к корректируемому вектору $y_i^{[t]}$, что влечет за собой рост векторов с количеством итераций. Чтобы избежать этого роста, можно контролировать, например, среднюю длину вектора или принудительно установить длины всех векторов быть равными единице путем нормализации векторов на каждом шаге. Последнее означает, что все точки данных отображаются на единичную гиперсферу. В этом случае приведенное выше выражение может быть записано следующим образом:

$$y_{ik}^{[t+1]} = y_{ik}^{[t]} - 2\alpha\mu \sum_{\substack{j=1,\\ j\neq i}}^{N} \rho_{ij} \cdot \left(\theta\left(x_i, x_j\right) - \theta\left(y_i^{[t]}, y_j^{[t]}\right)\right) \cdot \frac{y_{jk}^{[t]} - y_{ik}^{[t]} \left\langle y_i^{[t]}, y_j^{[t]} \right\rangle}{\sqrt{1 - \left\langle y_i^{[t]}, y_j^{[t]} \right\rangle^2}}. \quad (2.19)$$

Необходимо отметить, что нормализация не приводит к какой-либо потере информации о спектральных углах. Более того, так как в этой модификации длины векторов не используются, можно снизить размерность выходного пространства преобразуя выходные данные к гиперсферическим координатам (детали прямого и обратного преобразований показаны ниже) и отбрасывая координату радиуса, содержащую информацию о длине вектора. Таким образом, для рассматриваемой

модификации отображение на единичную гиперсферу в m – мерном пространстве может рассматриваться как отображение в $(m-1)$ - мерное выходное пространство.

2.2.2.2 Преобразование в гиперсферические и декартовы координаты

Введем сферическую систему координат в m-мерном евклидовом пространстве, задаваемую радиусом r и угловыми координатами $\varphi_1, \varphi_2, \ldots \varphi_{m-1}$. Преобразование в сферическую систему координат может быть вычислено с использованием следующих выражений [24, 225]:

$$r = \sqrt{\sum_{i=1}^{m} x_i^2}$$

$$\phi_k = \arccos\left(\frac{x_k}{\sqrt{\sum_{i=k}^{m} x_i^2}}\right), \quad k = 1..(m-2) \quad .$$

$$\phi_{m-1} = \begin{cases} \arccos\left(\dfrac{x_{m-1}}{\sqrt{x_{m-1}^2 + x_m^2}}\right), & x_m \geq 0, \\ 2\pi - \arccos\left(\dfrac{x_{m-1}}{\sqrt{x_{m-1}^2 + x_m^2}}\right), & x_m < 0. \end{cases} \tag{2.20}$$

Обратное преобразование может быть сделано следующим образом [24, 225]:

$$x_k = \begin{cases} r\prod_{i=1}^{k-1}\sin(\phi_i)\cos(\phi_k), & k = 1..(m-1), \\ r\prod_{i=1}^{m-1}\sin(\phi_i), & k = m. \end{cases} \tag{2.21}$$

2.2.2.3 Алгоритм формирования признаков ГСИ с сохранением спектральных углов в гиперсферических координатах

Как было сказано выше, формирование признаков с сохранением спектральных углов может рассматриваться как отображение на единичную гиперсферу. В этом подразделе мы разработаем аналог предложенного выше алгоритма, в гиперсферических координатах. Для этого мы зададим $r = 1$.

Обозначая

$$P_a^b(\phi_i, \phi_j) = \prod_{k=a}^{b} \sin(\phi_{ik})\cos(\phi_{jk}), \tag{2.22}$$

получаем функцию вычисления спектрального угла в гиперсферической системе координат:

$$\theta(\phi_i,\phi_j)=\arccos\left(\sum_{k=1}^{m-1}P_1^{k-1}(\phi_i,\phi_j)\cos(\phi_{ik})\cos(\phi_{jk})+P_1^{m-1}(\phi_i,\phi_j)\right) \quad (2.23)$$

Следуя базовому подходу, вводим ошибку представления данных с использованием введенной меры спектрального угла:

$$\varepsilon_{SAM_H}=\mu\cdot\sum_{i=1}^{N}\sum_{j=i+1}^{N}\rho_{ij}\left(\theta(\psi_i,\psi_j)-\theta(\phi_i,\phi_j)\right)^2 . \quad (2.24)$$

где ψ_i - гиперсферические координаты точек данных в многомерном пространстве, φ_i - (гипер-) сферические координаты соответствующих точек в формируемом пространстве меньшей размерности.

Далее применим алгоритм градиентного спуска для получения субоптимального решения. Это потребует вычисления частных производных ошибки представления данных:

$$\frac{\partial\varepsilon_{SAM_H}}{\partial\phi_{ik}}=\mu\sum_{j=1\,(j\neq i)}^{N}\left(-2\rho_{ij}\right)\cdot\left(\theta(\psi_i,\psi_j)-\theta(\phi_i,\phi_j)\right)\cdot\frac{\partial\theta(\phi_i,\phi_j)}{\partial\phi_{ik}}.$$

Здесь

$$\frac{\partial\theta(\phi_i,\phi_j)}{\partial\phi_{ik}}=-\left(1-\arg^2(\phi_i,\phi_j)\right)^{-1/2}\frac{\partial\arg(\phi_i,\phi_j)}{\partial\phi_{ik}}.$$

где *arg* – аргумент функции *arccos* в (2.23):

$$\arg(\phi_i,\phi_j)=\arccos\left(\sum_{k=1}^{m-1}P_1^{k-1}(\phi_i,\phi_j)\cos(\phi_{ik})\cos(\phi_{jk})+P_1^{m-1}(\phi_i,\phi_j)\right)$$

Частная производная *arg* записывается в виде выражения:

$$\frac{\partial\arg(\phi_i,\phi_j)}{\partial\phi_{ik}}=P_1^{k-1}(\phi_i,\phi_j)\cdot$$
$$\cdot\left(-\sin(\phi_{ik})\cos(\phi_{jk})+\cos(\phi_{ik})\sin(\phi_{jk})\sum_{l=k+1}^{m-1}P_{k+1}^{l-1}(\phi_i,\phi_j)\cos(\phi_{il})\cos(\phi_{jl})+P_{k+1}^{m-1}(\phi_i,\phi_j)\right).$$

Наконец, итеративное выражение для координат точек данных в выходном пространстве принимает следующий вид:

$$\phi_{ik}^{[t+1]}=\phi_{ik}^{[t]}+2\alpha\mu\sum_{j=1,\,j\neq i}^{N}\rho_{ij}\frac{\theta(\psi_i,\psi_j)-\theta(\phi_i^{[t]},\phi_j^{[t]})}{\sqrt{1-\arg^2(\phi_i^{[t]},\phi_j^{[t]})}}\frac{\partial\arg(\phi_i^{[t]},\phi_j^{[t]})}{\partial\phi_{ik}} \quad (2.25)$$
$$i=1\ldots N,$$
$$k=1\ldots(m-1).$$

Последнее выражение определяет алгоритм формирования признаков с сохранением спектральных углов в гиперсферической системе координат (Spectral Angle Preserving Mapping in Hyperspherical coordinates, SAPM-H) [196*].

К сожалению, описанный выше алгоритм оптимизации подвержен уходу в локальные минимумы, особенно в случае высокой выходной размерности. В частности, в экспериментах, описанный алгоритм обеспечивал более высокие значения ошибки по сравнению с другими рассмотренными методами, в том случае, когда в качестве метода оптимизации использовался простой алгоритм стохастического градиентного спуска. По этой причине полезно рассмотреть упрощенный подход при работе в гиперсферической системе координат. Этот подход [196*] основывается на модифицированной функции расстояния:

$$\theta^*\left(\phi_i, \phi_j\right) = \sum_{k=1}^{m} \sin^2 \frac{\phi_{ik} - \phi_{jk}}{2}. \tag{2.26}$$

Соответствующий алгоритм формирования признаков определяется с использованием следующего выражения (частные производные ошибки представления данных показаны в Приложении Б.3):

$$\begin{aligned} &\phi_{ik}^{[t+1]} = \phi_{ik}^{[t]} + \alpha\mu \sum_{j=1,\, j\neq i}^{N} \rho_{ij} \left(\theta^*(\psi_{i,}\psi_j) - \theta^*(\phi_i^{[t]}, \phi_j^{[t]})\right) \sin\left(\phi_{ik}^{[t]} - \phi_{jk}^{[t]}\right), \\ &i = 1...N, \\ &k = 1...(m-1). \end{aligned} \tag{2.27}$$

2.2.3 Алгоритм формирования признаков ГСИ, основанный на сохранении спектральной корреляции

В работе автора [195*] разработан алгоритм формирования признаков, основанный на принципе сохранения попарных мер спектральной корреляции между пикселями гиперспектрального изображения. Для разработки алгоритма вводится следующая ошибка представления данных в пространстве меньшей размерности, отражающая сохранение попарных мер спектральной корреляции:

$$\varepsilon_r = \mu \sum_{i=1}^{N} \sum_{j=i+1}^{N} \left(r\left(x_i, x_j\right) - r\left(y_i, y_j\right)\right)^2. \tag{2.28}$$

Здесь x_i – пиксели исходного гиперспектрального изображения, а y_i – соответствующие векторы в формируемом пространстве малой размерности.

Введенная мера качества показывает, насколько хорошо попарные значения спектральной корреляции сохраняются процедурой формирования признаков.

Как и ранее, для поиска оптимальных значений оптимизируемых параметров *Y*, будем использовать градиентный алгоритм. Определим частные производные ошибки представления данных:

$$\frac{\partial \varepsilon_r}{\partial y_{ik}} = \mu \sum_{j=1\,(j\neq i)}^{N} (-2)\cdot \left(r\left(x_i, x_j\right) - r\left(y_i, y_j\right)\right)\cdot \frac{\partial r\left(y_i, y_j\right)}{\partial y_{ik}}.$$

Здесь частные производные спектральной корреляции могут быть записаны следующим образом (подробнее см. Приложение Б.4):

$$\frac{\partial r\left(y_i, y_j\right)}{\partial y_{ik}} = \frac{1}{g_{ij}}\left(my_{jk} - \sum_{l=1}^{m} y_{jl} - \left(my_{ik} - \sum_{l=1}^{m} y_{il}\right) f_{ij}\right),$$

где

$$f_{ij} = \frac{m\sum_{k=1}^{m} y_{ik} y_{jk} - \sum_{k=1}^{m} y_{ik} \sum_{k=1}^{m} y_{jk}}{m\sum_{k=1}^{m} y_{ik}^2 - \left(\sum_{k=1}^{m} y_{ik}\right)^2},$$

$$g_{ij} = \sqrt{m\sum_{k=1}^{m} y_{ik}^2 - \left(\sum_{k=1}^{m} y_{ik}\right)^2}\sqrt{m\sum_{k=1}^{m} y_{jk}^2 - \left(\sum_{k=1}^{m} y_{jk}\right)^2}.$$

Наконец, рекуррентное выражение для координат выходных векторов в формируемом пространстве принимает следующий вид:

$$y_{ik}^{[t+1]} = y_{ik}^{[t]} - 2\alpha\mu \sum_{\substack{j=1,\\ j\neq i}}^{N} \rho_{ij} \frac{r\left(x_i, x_j\right) - r\left(y_i^{[t]}, y_j^{[t]}\right)}{g_{ij}^{[t]}}\left(my_{jk}^{[t]} - \sum_{l=1}^{m} y_{jl}^{[t]} - \left(my_{ik}^{[t]} - \sum_{l=1}^{m} y_{il}^{[t]}\right) f_{ij}^{[t]}\right), \tag{2.29}$$

$i = 1...N,$

$k = 1...m.$

Предложенный алгоритм формирует признаки ГСИ на основе принципа сохранения спектральной корреляции (Spectral Correlation Preserving Mapping, SCPM) [195*].

2.2.4 Алгоритм формирования признаков ГСИ, основанный на сохранении дивергенции спектральной информации

Предложенный автором в работе [193*] алгоритм формирования признаков ГСИ основан на принципе сохранения дивергенции спектральной информации между пикселями гиперспектрального изображения.

В работе вводится следующая ошибка представления гиперспектральных данных в пространстве меньшей размерности на основе дивергенции спектральной информации:

$$\varepsilon_{SID} = \mu \sum_{i=1}^{N} \sum_{j=i+1}^{N} \rho_{ij} \left(SID(x_i, x_j) - SID(y_i, y_j) \right)^2 , \qquad (2.30)$$

показывающая, насколько хорошо попарные меры SID между пикселями гиперспектрального изображения сохраняются при отображении данных в пространство меньшей размерности.

Решение задачи минимизации выполняется в соответствии с предложенным методом с использованием алгоритма градиентного спуска. Реализация схемы оптимизации требует нахождения частных производных (см. Приложение Б.5) введенного выше функционала ошибки (2.30):

$$\frac{\partial \varepsilon_{SID}}{\partial y_{ik}} = -2\mu \sum_{j=1\,(j\neq i)}^{N} \rho_{ij} \left(SID(x_i, x_j) - SID(y_{i,} y_j) \right) \frac{\partial}{\partial y_{ik}} SID\left(y_i, y_j \right)$$

и самой функции расстояния:

$$\frac{\partial}{\partial y_{ik}} SID\left(y_i, y_j \right) = \frac{1}{\sum_{l=1}^{m} y_{il}} \left(1 - \frac{p_k(y_j)}{p_k(y_i)} + \log \frac{p_k(y_i)}{p_k(y_j)} - D\left(y_i \| y_j \right) \right).$$

Окончательное рекуррентное соотношение, позволяющее уточнять координаты векторов y_i, i=1..N в формируемом редуцированном пространстве, выглядит следующим образом:

$$y_{ik}^{[t+1]} = y_{ik}^{[t]} + \\ + \frac{2\alpha\mu}{\sum_{l=1}^{m} y_{il}^{[t]}} \sum_{\substack{j=1, \\ j\neq i}}^{N} \rho_{ij} \left(SID(x_i, x_j) - SID(y_i^{[t]}, y_j^{[t]}) \right) \left(1 - \frac{p_k(y_j^{[t]})}{p_k(y_i^{[t]})} + \log \frac{p_k(y_i^{[t]})}{p_k(y_j^{[t]})} - D\left(y_i^{[t]} \| y_j^{[t]} \right) \right), \qquad (2.31)$$
$$i = 1...N,$$
$$k = 1..m.$$

Последнее выражение определяет алгоритм формирования признаков ГСИ на основе принципа сохранения дивергенции спектральной информации (Spectral information Divergence Preserving Mapping, SDPM) [193*].

Отметим, что при реализации градиентной процедуры оптимизации для функционала (2.30) хорошие результаты достигаются с использованием

экспоненциального градиентного спуска [143], обновление в котором носит мультипликативный характер:

$$y_{ik}^{[t+1]} = y_{ik}^{[t]} \cdot \exp\left(-\alpha \frac{\partial \varepsilon}{\partial y_{ik}}\right) \Big/ \sum_{l=1}^{m} y_{il}^{[t]} \cdot \exp\left(-\alpha \frac{\partial \varepsilon}{\partial y_{il}}\right).$$

Это естественным образом обеспечивает неотрицательность и нормированность оптимизируемых переменных, что критично при работе с такими функциями расстояния как SID.

2.2.5 Алгоритм формирования признаков ГСИ, основанный на сохранении дивергенции Хеллингера

Чтобы применить предложенный в настоящем разделе метод для дивергенции Хеллингера, введем соответствующую ошибку представления данных [183*]:

$$\varepsilon_{HD} = \mu \sum_{i=1}^{N} \sum_{j=i+1}^{N} \rho_{ij} \left(HD(x_i, x_j) - HD(y_i, y_j)\right)^2, \tag{2.32}$$

Для получения градиента целевой функции $\nabla\varepsilon$ вычислим частные производные дивергенции Хеллингера (см. Приложение Б.6):

$$\frac{\partial}{\partial y_{ik}} HD(y_i, y_j) = \frac{1}{4HD(y_i, y_j) \sum_{l=1}^{m} y_{il}} \left(\sum_{l=1}^{m} \sqrt{p_l(y_i) p_l(y_j)} - \sqrt{\frac{p_k(y_j)}{p_k(y_i)}} \right).$$

Для получения частных производных ошибки представления данных применим ряд преобразований:

$$\frac{\partial \varepsilon_{HD}}{\partial y_{ik}} = -\frac{\mu}{2\sum_{l=1}^{m} y_{il}} \sum_{\substack{j=1, \\ j \neq i}}^{N} \rho_{ij} \frac{HD(x_i, x_j) - HD(y_i, y_j)}{HD(y_i, y_j)} \left(\sum_{l=1}^{m} \sqrt{p_l(y_i) p_l(y_j)} - \sqrt{\frac{p_k(y_j)}{p_k(y_i)}} \right).$$

Теперь можно записать рекуррентное соотношение для оптимизируемых параметров Y, реализующее процедуру градиентного спуска:

$$y_{ik}^{[t+1]} = y_{ik}^{[t]} + \\ + \frac{\alpha\mu}{2\sum_{l=1}^{m} y_{il}^{[t]}} \sum_{\substack{j=1, \\ j \neq i}}^{N} \rho_{ij} \frac{HD(x_i, x_j) - HD(y_i^{[t]}, y_j^{[t]})}{HD(y_i^{[t]}, y_j^{[t]})} \left(\sum_{l=1}^{m} \sqrt{p_l(y_i^{[t]}) p_l(y_j^{[t]})} - \sqrt{\frac{p_k(y_j^{[t]})}{p_k(y_i^{[t]})}} \right), \tag{2.33}$$

$i = 1 \ldots N,$

$k = 1 \ldots m.$

Для инициализации оптимизируемых параметров $Y^{[0]}$ использовались равноотстоящие друг от друга компоненты исходного вектора, а сама оптимизация выполнялась с контролем ошибки. Последнее выражение определяет алгоритм формирования признаков, действующий по принципу сохранения дивергенции Хеллингера (Hellinger Divergence Preserving Mapping, HDPM) [183*].

2.2.6 Алгоритм формирования признаков ГСИ, основанный на сохранении углов Бхаттачария

Аналогичный результат можно получить и для угла Бхаттачария (см. п. 2.1.6). Для целевой функции, выраженной в виде:

$$\varepsilon_{BCa} = \mu \sum_{i=1}^{N} \sum_{j=i+1}^{N} \rho_{ij} \left(BCa(x_i, x_j) - BCa(y_i, y_j) \right)^2 , \tag{2.34}$$

частные производные имеют следующий вид (подробнее см. Приложение Б.7):

$$\frac{\partial \varepsilon_{BCa}}{\partial y_{ik}} = -\frac{\mu}{\sum_{l=1}^{m} y_{il}} \sum_{\substack{j=1, \\ j \neq i}}^{N} \rho_{ij} \frac{BCa(x_i, x_j) - BCa(y_{i,} y_j)}{\sqrt{1 - \left(\sum_{l=1}^{m} \sqrt{p_l(y_i) p_l(y_j)} \right)^2}} \left(\sum_{l=1}^{m} \sqrt{p_l(y_i) p_l(y_j)} - \sqrt{\frac{p_k(y_j)}{p_k(y_i)}} \right)$$

Рекуррентное соотношение для оптимизируемых параметров, реализующее процедуру градиентного спуска записывается следующим образом:

$$y_{ik}^{[t+1]} = y_{ik}^{[t]} + \\ + \frac{\alpha\mu}{\sum_{l=1}^{m} y_{il}^{[t]}} \sum_{\substack{j=1, \\ j \neq i}}^{N} \rho_{ij} \frac{BCa(x_i, x_j) - BCa(y_i^{[t]}, y_j^{[t]})}{\sqrt{1 - \left(\sum_{l=1}^{m} \sqrt{p_l(y_i^{[t]}) p_l(y_j^{[t]})} \right)^2}} \left(\sum_{l=1}^{m} \sqrt{p_l(y_i^{[t]}) p_l(y_j^{[t]})} - \sqrt{\frac{p_k(y_j^{[t]})}{p_k(y_i^{[t]})}} \right), \tag{2.35}$$

$i = 1...N,$

$k = 1..m.$

Следует отметить, что все разработанные алгоритм формирования признаков успешно минимизируют ошибку (2.10) представления гиперспектральных данных в формируемом пространстве признаков меньшей размерности. Это подтверждается рисунком 2.1, на котором показано, как изменяется ошибка (2.10) по мере работы предложенных алгоритмов. Как видно из рисунка, уже на первых 50 итерациях удается значительно снизить ошибку (по оси ординат использован логарифмический масштаб).

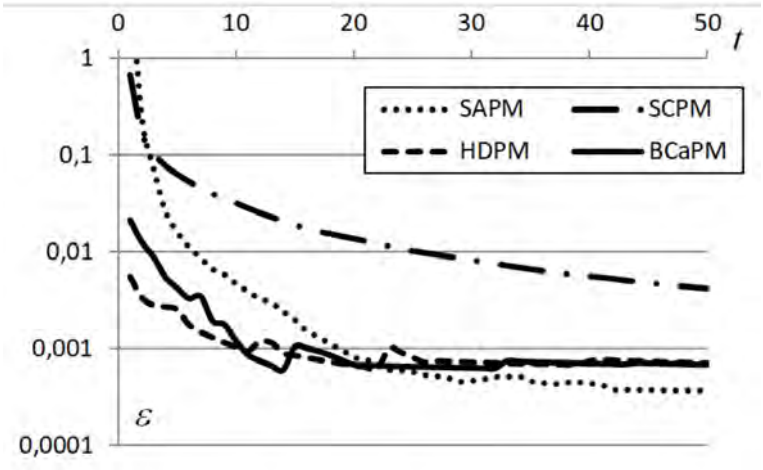


Рисунок 2.1 – Зависимость ошибки представления данных от номера итерации для алгоритмов формирования признаков, основанных на сохранении спектральных углов (SAPM), спектральной корреляции (SCPM), дивергенции Хеллингера (HDPM), углов Бхаттачария (BCaPM).

2.3 Формирование признаков ГСИ на основе аппроксимации заданных расстояний евклидовыми расстояниями

2.3.1 Алгоритм формирования признаков, основанный на аппроксимации значений выбранной функции расстояния евклидовыми расстояниями

Построение алгоритма формирования признаков, основанного на аппроксимации значений выбранной функции расстояния во входном пространстве евклидовыми расстояниями в выходном пространстве, не представляет сложностей. Функционал ошибки при этом будет иметь вид (2.9), где функцией расстояния в выходном пространстве будет являться евклидово расстояние: $\psi(y_i,y_j)\equiv d(y_i,y_j)$,

Реализация такого алгоритма снижения размерности вытекает из (2.13) при p=2 путем замены вычисления евклидовых расстояний в исходном пространстве вычислением выбранной функции расстояния δ:

$$y_i^{[t+1]} = y_i^{[t]} + 2\alpha\mu \sum_{j=1\,(j\neq i)}^{N} \rho_{ij} \frac{\delta(x_i,x_j) - d(y_i^{[t]},y_j^{[t]})}{d(y_i^{[t]},y_j^{[t]})} \cdot \left(y_i^{[t]} - y_j^{[t]}\right). \qquad (2.36)$$

Фактически здесь мы будем пытаться аппроксимировать, например, расстояния SAM или HD евклидовыми расстояниями в формируемом редуцированном пространстве. Соответствующий алгоритм, основанный на спектральных углах

(SAM), будем называть SAED, на дивергенции спектральной информации (SID) – SDED, на дивергенции Хеллингера (HD) - HDED, на углах Бхаттачария (BCa) - BCED и др.

Чтобы показать, что такой алгоритм действительно позволяет снизить ошибку представления данных в формируемом пространстве, приведем на рисунке 2.2 кривые зависимости ошибки (2.9) от номера итерации для различных функций расстояния, используемых в спектральном пространстве.

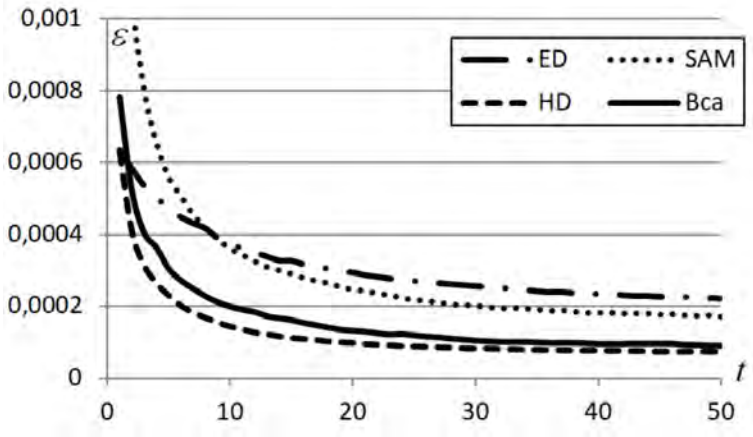


Рисунок 2.2 - Зависимость ошибки представления данных от номера итерации для алгоритмов формирования признаков, основанных на аппроксимации значений выбранной функции расстояния (ED, SAM, HD, BCa) евклидовыми расстояниями

2.3.2 Применение выбранных функций расстояния для формирования признаков с использованием альтернативных нелинейных методов снижения размерности

В этом подразделе описывается довольно простой подход к формированию признаков ГСИ с использованием выбранных функций расстояния с рядом нелинейных методов снижения размерности общего назначения, приведенных ранее в разделе 1.3 и описанных более подробно в приложении А.

В методе ISOMAP [290, 289] мы используем выбранные функции расстояния для нахождения соседних точек на шаге 1 метода (см. Приложение А.7) и для определения весов ребер (см. выражение (А.5)) на шаге 2.

В методе локально линейного вложения (LLE) [252] мы используем выбранные функции расстояния только для нахождения соседних точек на шаге 1 (см. Приложение А.10).

В методе лапласианских собственных карт (LE) [18] мы используем выбранные функции расстояния для нахождения соседних точек на шаге 1 (см. Приложение А.11) и для определения теплового ядра (см. выражение (А.6)) на шаге 2.

В методе UMAP [170] (см. Приложение А.14) неевклидовы функции расстояния используются на этапах 1-3. В частности, мы используем выбранные функции для формирования наборов соседних точек на шаге 1. Затем, на шаге 2, мы используем выбранные функции для нахождения ближайших точек, расстояний ρ_i и параметров σ_i. Позже такие функции влияют на построение графа на шаге 3, поскольку они участвуют в вычислении весов.

2.4 Экспериментальное исследование алгоритмов формирования признаков ГСИ, основанных на сохранении заданных расстояний

2.4.1 Используемые наборы данных

Для представленного здесь исследования мы использовали несколько хорошо известных тестовых гиперспектральных изображений [120, 17], предварительно размеченных, т.е. снабженных истинной попиксельной классификацией. Краткое описание используемых изображений приведено в таблице 2.1.

Таблица 2.1 - Тестовые гиперспектральные изображения.

Сцена	Сенсор	Размер (H×W)	Количество каналов*	Количество классов
Indian pines	AVIRIS	145×145	220 (200)	16
Salinas	AVIRIS	512×217	224 (204)	16
Botswana	Hyperion	1476×256	145	14
Kennedy Space Center	AVIRIS	512×614	176	13

*) В скобках указано количество каналов после отбрасывания каналов с высоким уровнем шума

Поскольку почти все представленные сцены содержат более 100000 пикселей, и для выполнения исследования необходимо было выполнить множество прогонов алгоритмов формирования признаков ГСИ, в некоторых случаях для экспериментов использовалась регулярная (в пространственной области) выборка из тестовых изображений, которая при необходимости маскировалась с использованием маски, получаемой по размеченным (истинным) данным. Использованные в работе тестовые гиперспектральные снимки приведены на рисунке 2.3.

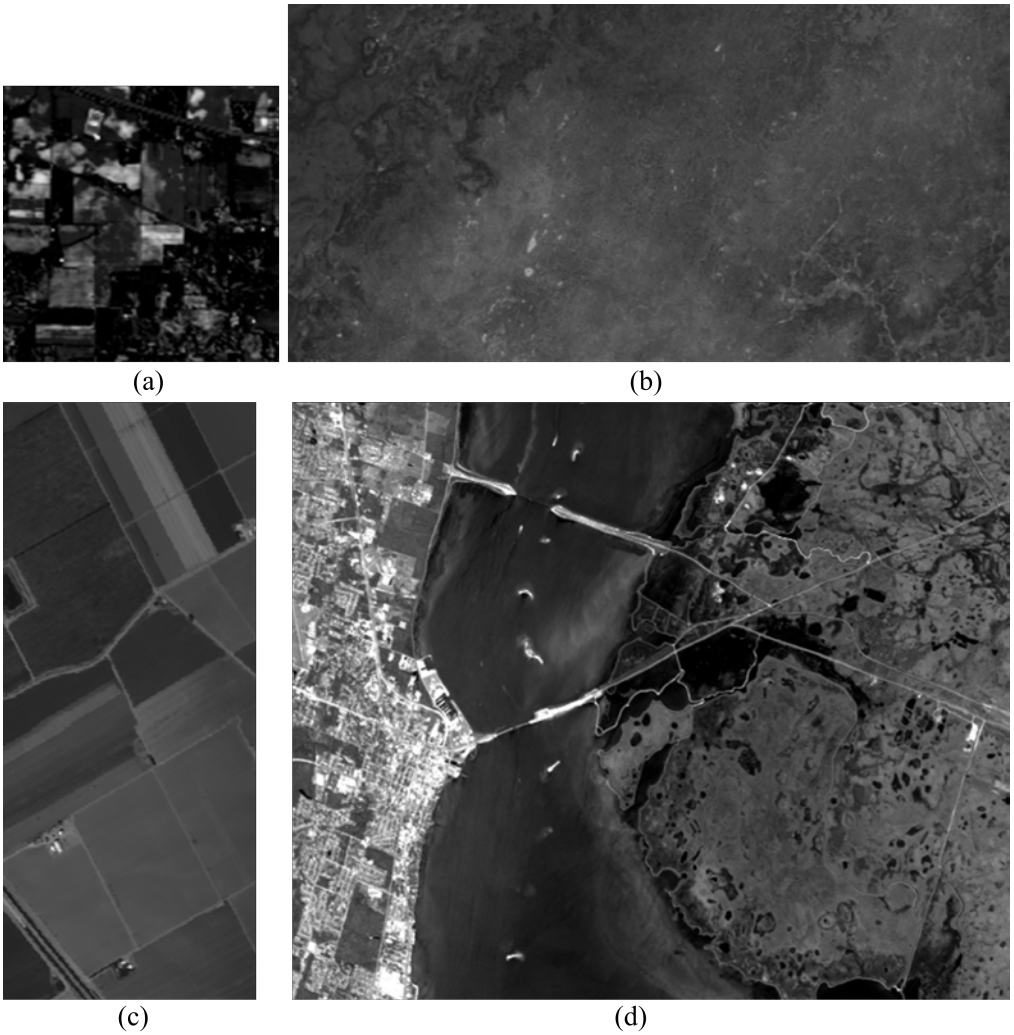


Рисунок 2.3 - Тестовые гиперспектральные изображения (представлены в виде полутоновых изображений): Indian Pines (а), Botswana, фрагмент (б), Salinas (в) и космический центр Кеннеди, KSC (г).

Для проведения экспериментов использовалась реализация PCA, поставляемая со средой Matlab [168]. Для методов LLE, лапласианских собственных карт и

ISOMAP использовался Matlab Toolbox для снижения размерности [297]. Также использовалась реализация UMAP [171], устанавливаемая с использованием PyPI [342]. Алгоритмы формирования признаков ГСИ, основанные на сохранении заданных расстояний, а также алгоритмы, основанные на аппроксимации заданных значений функций расстояния евклидовыми расстояниями, были реализованы самостоятельно на языке C++.

2.4.2 Оценка качества

Для оценки предложенных алгоритмов формирования признаков и сравнения с альтернативными методами использовался показатель качества решения задачи попиксельной классификации, выполняемой по сформированным признаковым представлениям ГСИ в сниженной размерности, полученным с использованием оцениваемых алгоритмов и методов. В качестве показателя качества классификации была выбрана общая точность классификации (Acc), определяемая как доля правильно классифицированных пикселей в тестовом наборе.

Использовались два типа классификаторов: классификатор по ближайшему соседу (NN-классификатор) и машина опорных векторов (SVM). Выбор классификатора по ближайшему соседу обусловлен следующими соображениями:

- этот классификатор не имеет параметров, что исключает возможную необъективность ввиду неоптимальности параметров классификатора для тех или иных сценариев;

- этот классификатор не требует настройки, что означает отсутствие необходимости обучения или настройки каких либо параметров всей цепочки, за исключением размерности формируемого пространства m.

Для получения обучающих и тестовых подмножеств множество пикселей, снабженных истинной классификацией, разделялось в пропорции 60:40.

2.4.3 Результаты экспериментов.

Для каждого отдельно взятого алгоритма формирования признаков (см. п. 2.2, п. 2.3.1) и альтернативного метода (см. п. 2.3.2) в сочетании с различными функциями расстояния был проведен ряд экспериментов, описанных в работах [188*, 205*, 187*]. В частности, в каждом эксперименте размерность формируемого пространства изменялась от 2 до 20, выполнялось формирование признаков в пространстве

выбранной размерности, выборка разбивалась на обучающее и тестовое подмножества, обучались указанные выше классификаторы, и оценивалась точность классификации.

Результаты экспериментального исследования, представленные на рисунках 2.4-2.7, позволяют сравнить некоторые из разработанных алгоритмов формирования признаков, в частности, EDPM (п. 2.2.1), SDPM (п. 2.2.4), SAED и SDED (п. 2.3.1), а также альтернативные методы Isomap, LLE, Laplacian Eigenmaps и UMAP (п. 2.3.2) при различных размерностях формируемого пространства.

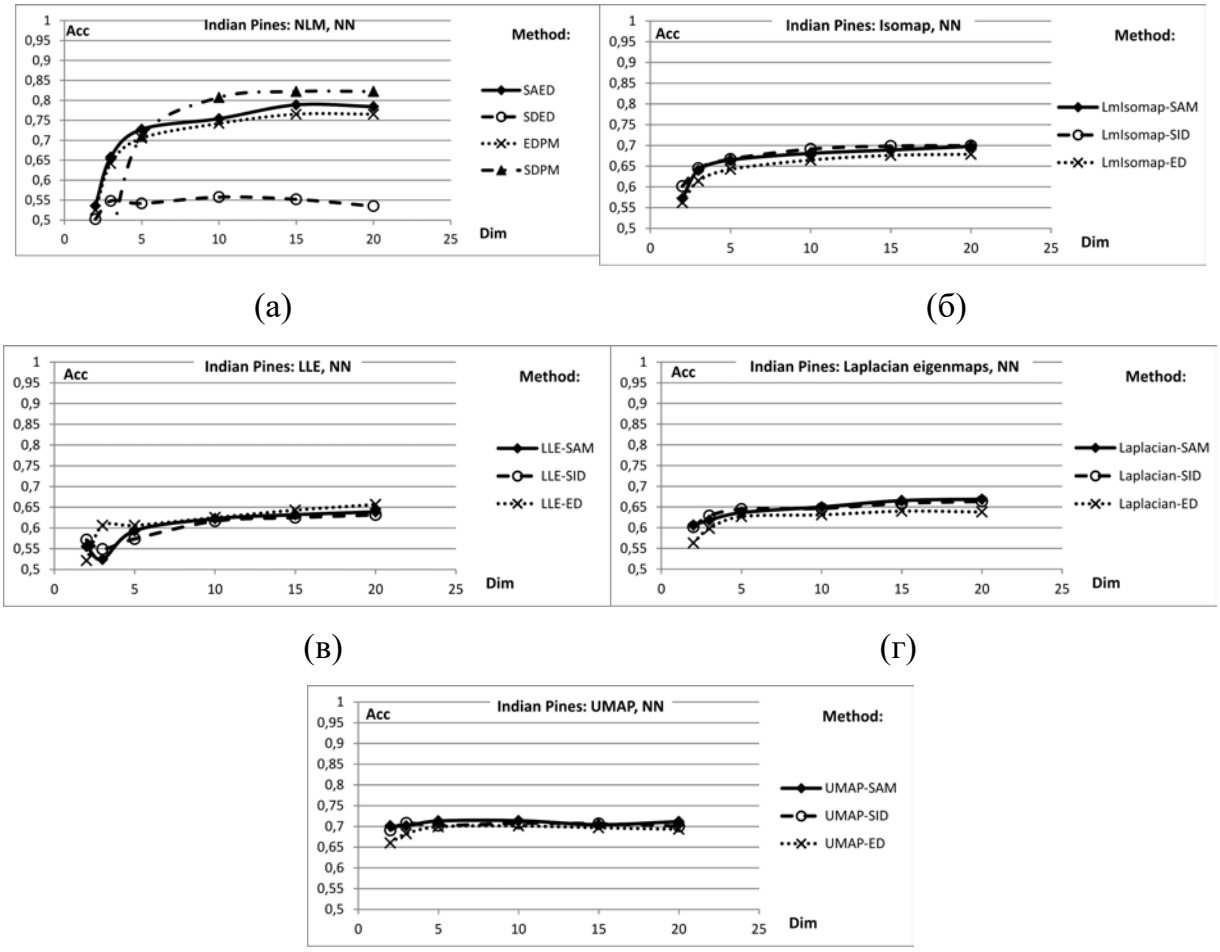


Рисунок 2.4 - Зависимость точности классификации Acc от размерности редуцированного пространства Dim для NN - классификатора для различных методов снижения размерности: сцена Indian Pines.

Каждый рисунок соответствует отдельному гиперспектральному изображению, а отдельные его части – конкретному алгоритму или методу формирования признаков в сочетании с различными функциями расстояния и классификаторами. Так, на рисунке 2.4 (а) представлены результаты экспериментов, полученные для изображения Indian Pines и алгоритмов формирования признаков, основанных на

сохранении евклидовых расстояний (EDPM, п. 2.2.1), аппроксимации спектральных углов евклидовыми расстояниями (SAED, п. 2.3.1), аппроксимации дивергенции спектральной информации евклидовыми расстояниями (SDED, п. 2.3.1), а также алгоритма на основе сохранения дивергенции спектральной информации (SDPM, см. п. 2.2.4). В качестве классификатора использовался классификатор по ближайшему соседу.

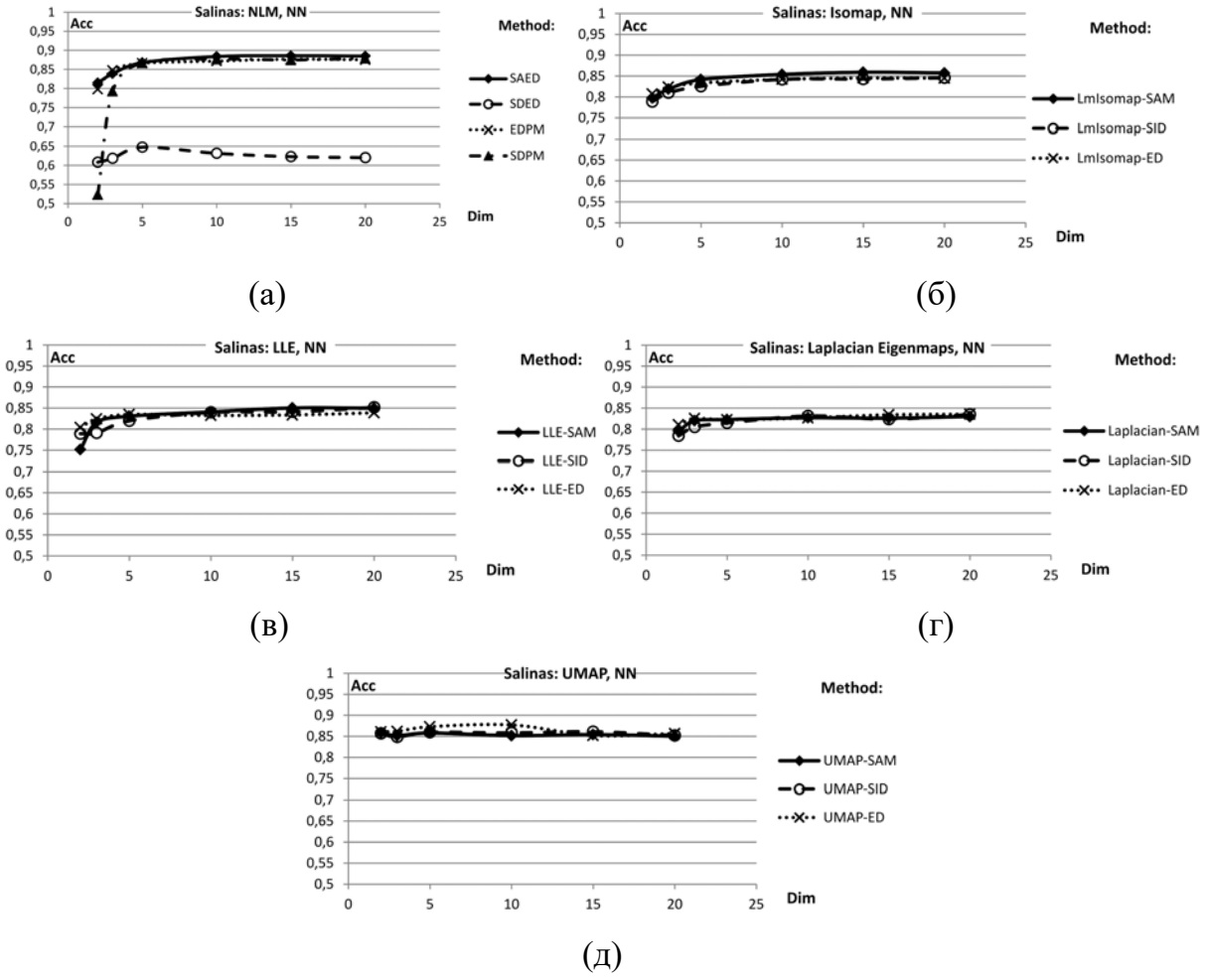


Рисунок 2.5 - Зависимость точности классификации Acc от размерности редуцированного пространства Dim для NN - классификатора и различных методов снижения размерности: сцена Salinas.

Таким образом, каждая часть рисунка содержит, по крайней мере, по три кривых, соответствующих трем различным функциям расстояния (ED, SAM и SID). Кривые обозначены на рисунках в формате <алгоритм>-<функция расстояния>. Исключение составляют рисунки под буквами (а), содержащие также четвертую кривую, отражающую результаты работы алгоритма SDPM, о чем дополнительно будет сказано ниже.

По оси абсцисс на каждом рисунке отложена размерность формируемого выходного пространства, по оси ординат – точность классификации по сформированным в пространстве указанной размерности признакам.

Как видно из представленных графиков, точность работы методов в целом возрастает с ростом размерности формируемого пространства, что является ожидаемым результатом.

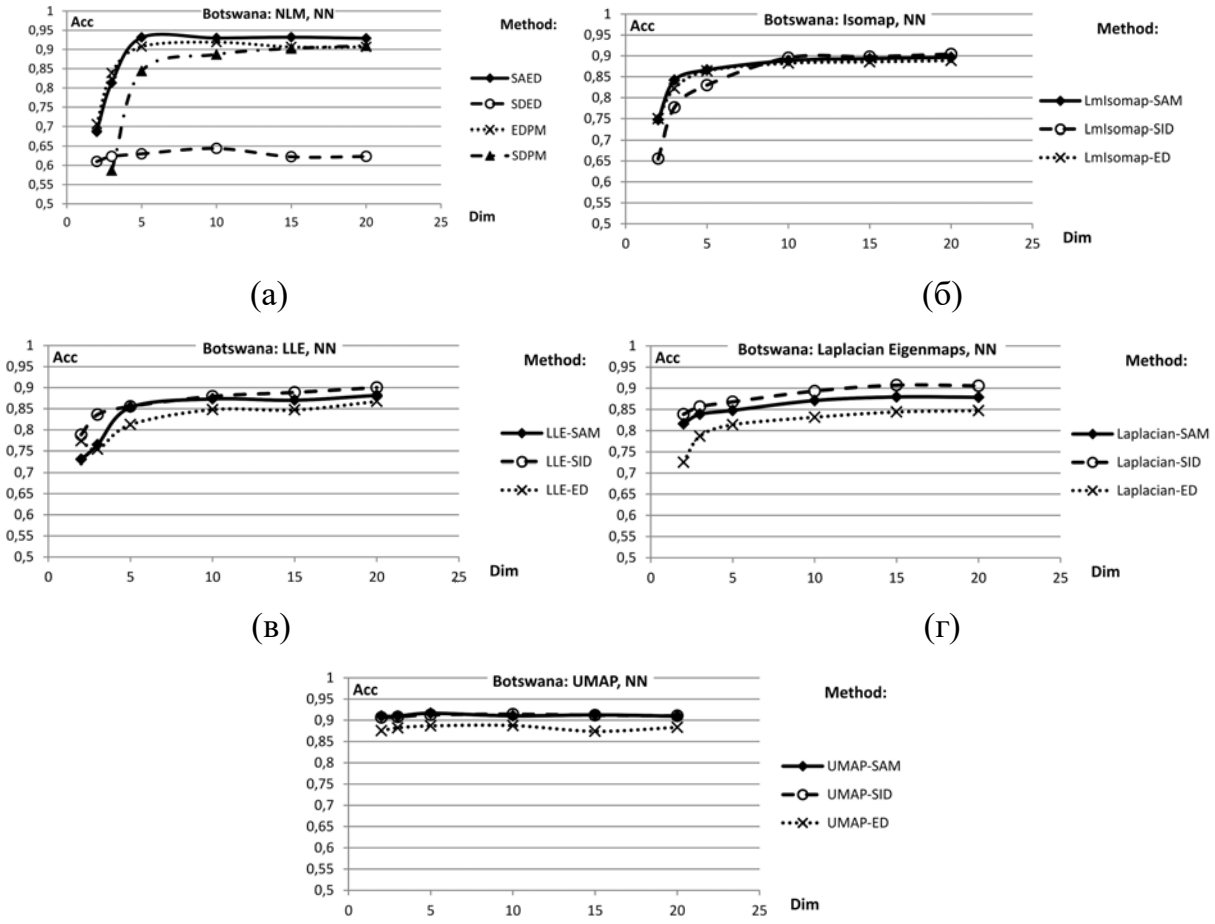


Рисунок 2.6 - Зависимость точности классификации Acc от размерности редуцированного пространства Dim для NN - классификатора для различных методов снижения размерности: сцена Botswana.

Выбор функции расстояния однозначно сделать невозможно. Однако можно отметить, что алгоритм формирования признаков, основанный на аппроксимации значений дивергенции спектральной информации евклидовыми расстояниями (SDED) давал заметно худшие результаты, чем при использовании других функций расстояния. По этой причине в графики, содержащие результаты для указанного алгоритма, включены также результаты для алгоритма SDPM [193*], работающего по принципу сохранения дивергенции спектральной информации. Результаты работы

этого алгоритма оказываются конкурентоспособными, а в ряде случаев и наилучшими.

Основное отличие алгоритма SDPM от SDED заключается в том, что SDPM пытается не аппроксимировать исходные попарные расстояния SID евклидовыми расстояниями в выходном пространстве, а найти такое отображение, при котором будут по возможности сохраняться сами расстояния SID. Это также означает, что при использовании NN-классификатора должно использоваться также расстояние SID вместо евклидового расстояния, что и было сделано в настоящем исследовании. SVM классификатор в исследовании не претерпевал изменений.

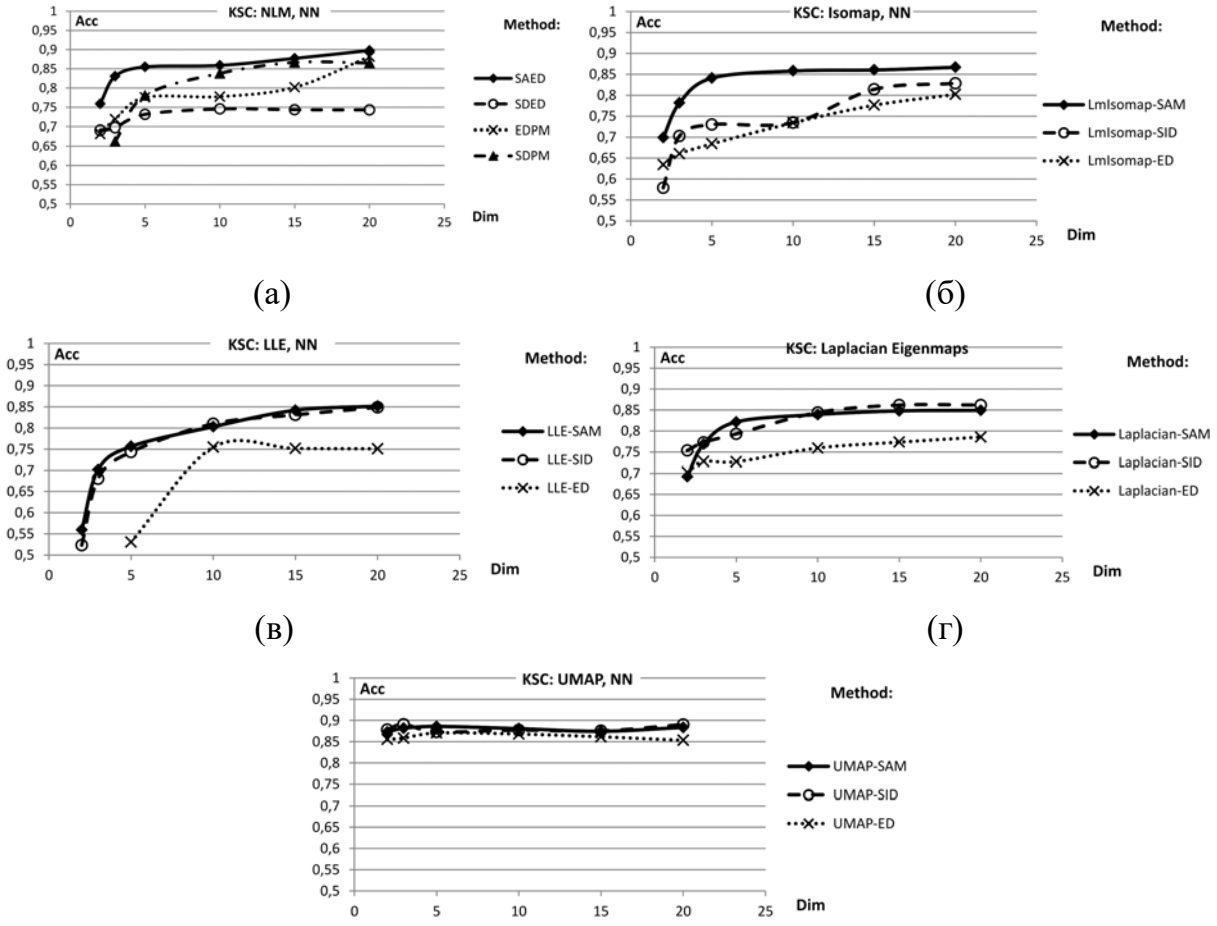


Рисунок 2.7 - Зависимость точности классификации Acc от размерности редуцированного пространства Dim для NN - классификатора для различных методов снижения размерности: сцена KSC.

Сравнение показывает, что для низких размерностей формируемого пространства (Dim ≤ 3) метод UMAP (см. рисунки 2.4(д) – 2.7(д), 2.8(д) - 2.11(д)) превосходит другие методы почти во всех рассмотренных случаях. С ростом выходной размерности для Dim = 10..20 преимущества UMAP нивелируются, и на

первое место выходят алгоритмы формирования признаков, основанные на сохранении выбранных расстояний и аппроксимации значений выбранной функции расстояния евклидовыми расстояниями (см. 2.2(а) – 2.5(а), 2.6(а) - 2.9(а)), которые представляются наиболее подходящим выбором для классификатора по ближайшему соседу.

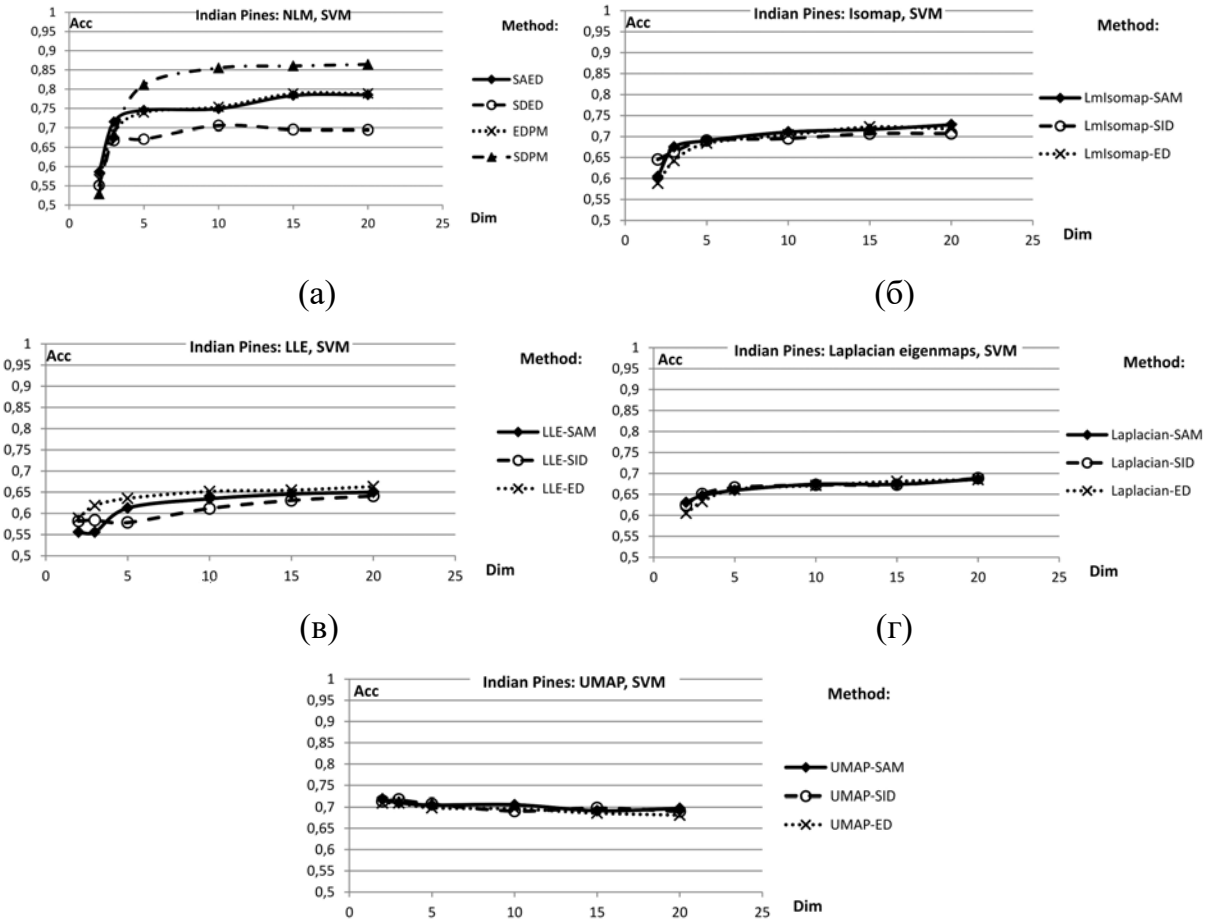


Рисунок 2.8 - Зависимость точности классификации Acc от размерности редуцированного пространства Dim для SVM- классификатора для различных методов снижения размерности: сцена Indian Pines.

С целью сравнения результатов, получаемых с использованием алгоритмов формирования признаков, основанных на нелинейных методах снижения размерности, с методом главных компонент (PCA), были проведены соответствующие эксперименты. На рисунках 2.12-2.13 наилучшие значения показателей качества, полученные с использованием нелинейных методов приведены в сравнении с методом PCA. На этих рисунках под отдельными буквами представлены результаты, соответствующие различным тестовым изображениям и классификаторам. Каждый рисунок содержит две кривых, одна из которых

соответствует методу PCA, а другая – наилучшим результатам среди рассмотренных нелинейных методов (NLDR).

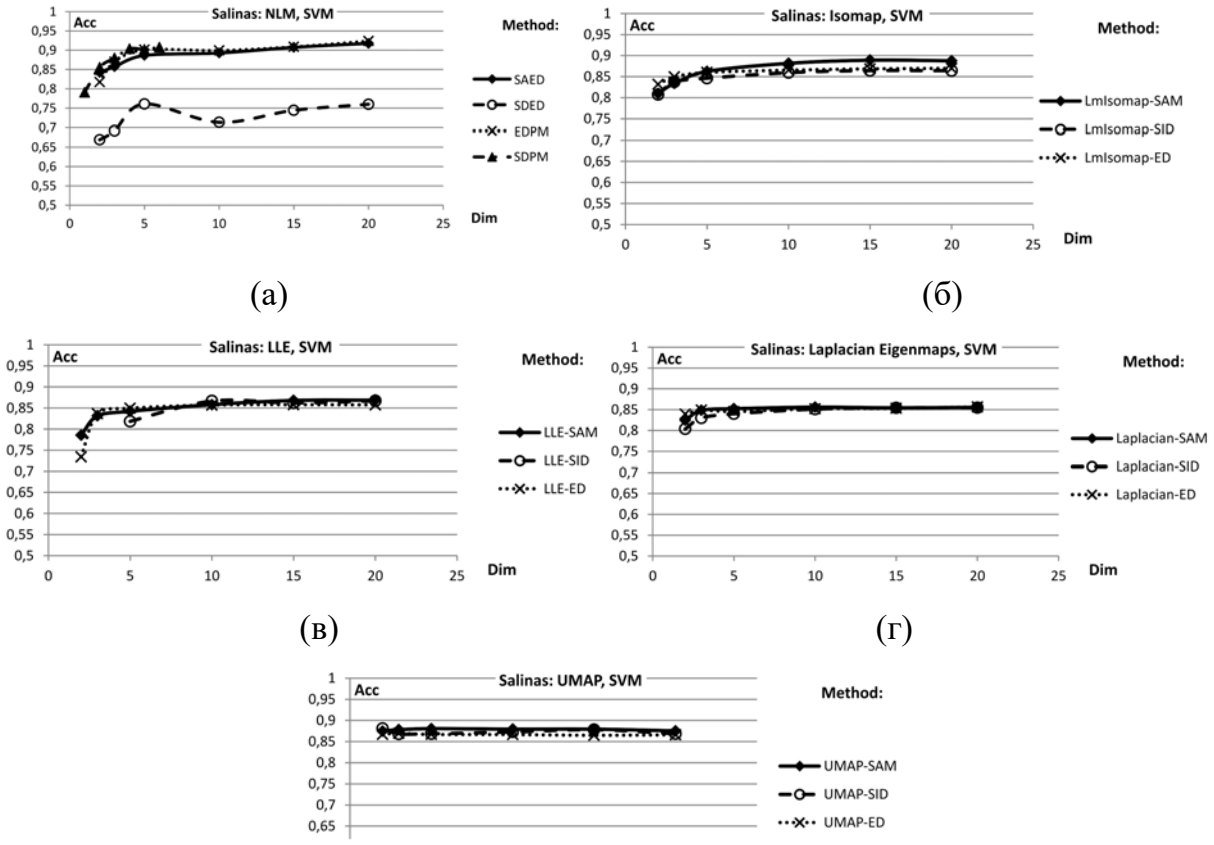


Рисунок 2.9 - Зависимость точности классификации Acc от размерности редуцированного пространства Dim для SVM- классификатора для различных методов снижения размерности: сцена Salinas.

Как видно из представленных рисунков 2.12-2.13, наименьший отрыв нелинейных методов от метода главных компонент обнаруживается на изображениях Salinas и Botswana, наибольший – на Indian Pines и KSC.

Стоит отметить, что изображение KSC оказалось самым сложным для линейного метода PCA. Точность классификации при снижении размерности с использованием этого метода составила около 40% для *Dim* = 2, 3, и даже для *Dim* = 20 результаты PCA оставались заметно ниже результатов любого нелинейного метода, что можно увидеть из сравнения рисунков 2.12 и 2.13.

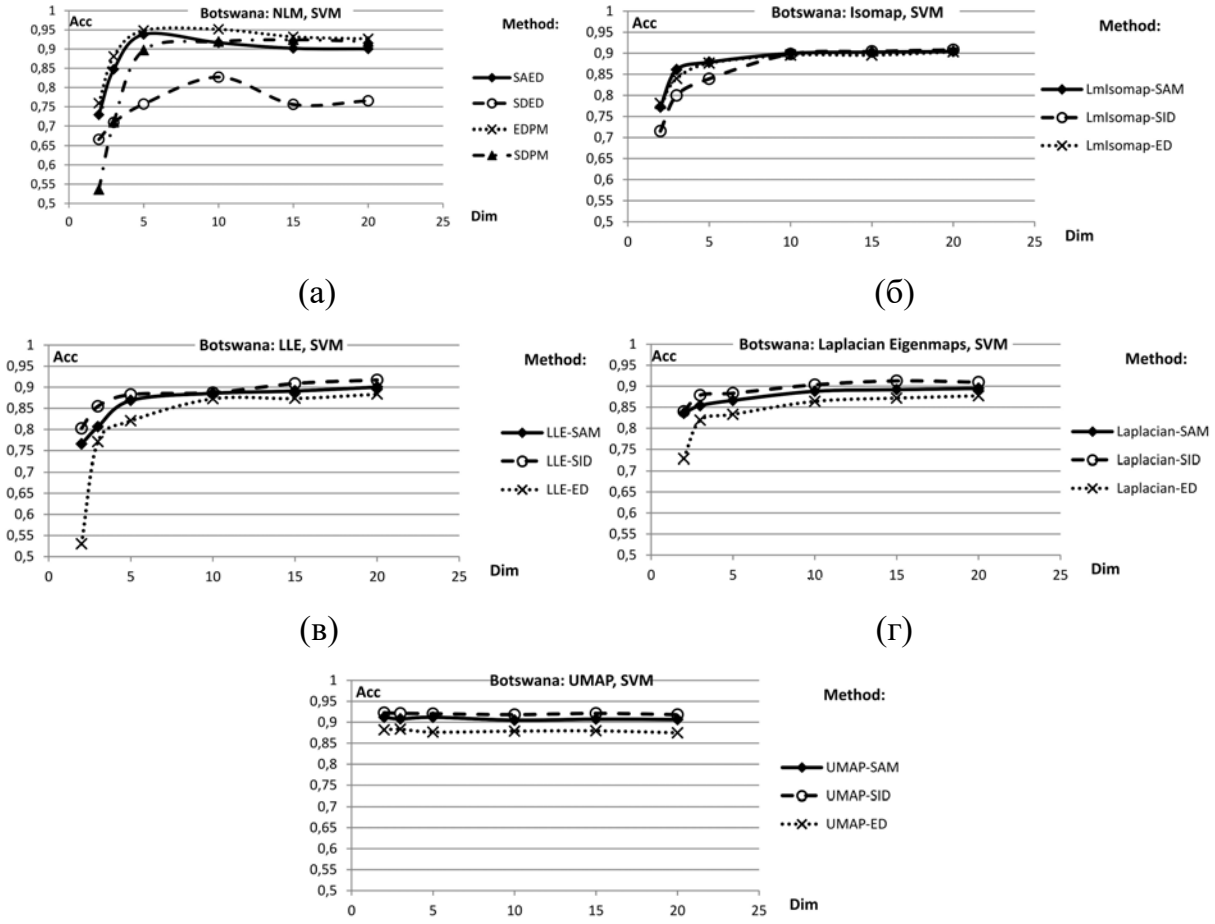


Рисунок 2.10 - Зависимость точности классификации Acc от размерности редуцированного пространства Dim для SVM- классификатора для различных методов снижения размерности: сцена Botswana.

В целом, эксперименты показали, что для малых выходных размерностей (2, 3) наилучшие результаты дает метод UMAP совместно с мерами SAM или SID. Для более высоких размерностей (10..20) разработанные алгоритмы формирования признаков совместно с неевклидовыми функциями расстояния являются более предпочтительным выбором для NN-классификатора. Следует отметить, что PCA оказался конкурентоспособен только на изображениях Botswana и Salinas при использовании классификатора SVM и размерностях больше пяти.

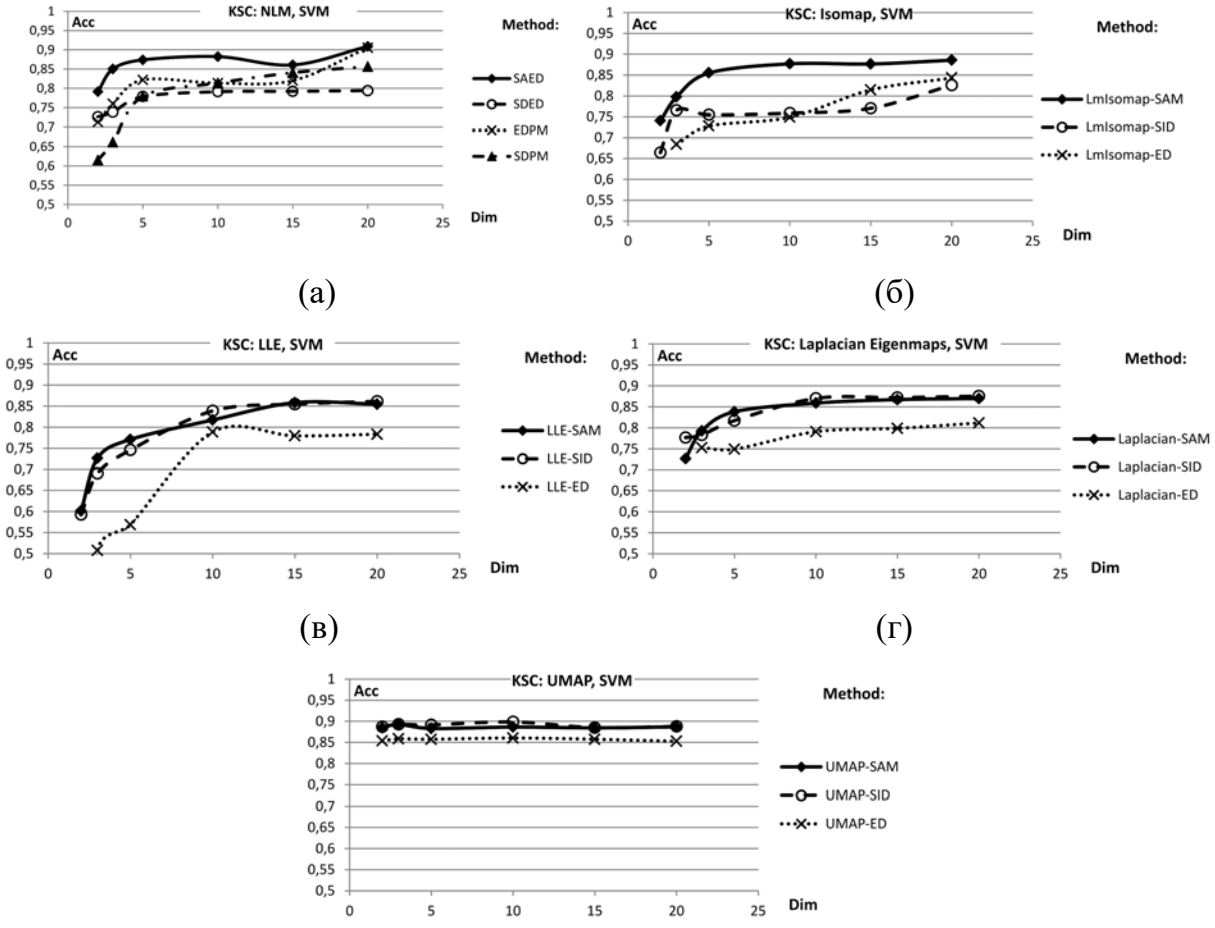


Рисунок 2.11 - Зависимость точности классификации Acc от размерности редуцированного пространства Dim для SVM- классификатора для различных методов снижения размерности: сцена KSC.

Дальнейшие эксперименты включали сравнение разработанных алгоритмов формирования признаков, основанных на сохранении заданных расстояний, с алгоритмами, основанными на аппроксимации заданных расстояний во входном пространстве евклидовыми расстояниями в формируемом пространстве и уже исследованными альтернативными методам. В частности, исследовались алгоритмы, основанные на сохранении заданных расстояний: EDPM (п. 2.2.1), SAPM (п. 2.2.2), SCPM (п. 2.2.3), SDPM (п. 2.2.4), HDPM (п. 2.2.5), BCPM (п. 2.2.6); алгоритмы, основанные на аппроксимации заданных расстояний во входном пространстве евклидовыми расстояниями: SAED, SDED, HDED и BCED (п. 2.3.1), а также альтернативные методы: Isomap, LLE, Laplacian Eigenmaps и UMAP совместно с различными функциями расстояния. Результаты сравнения для каждого тестового изображения и для различных размерностей формируемого пространства приведены в таблице 2.2.

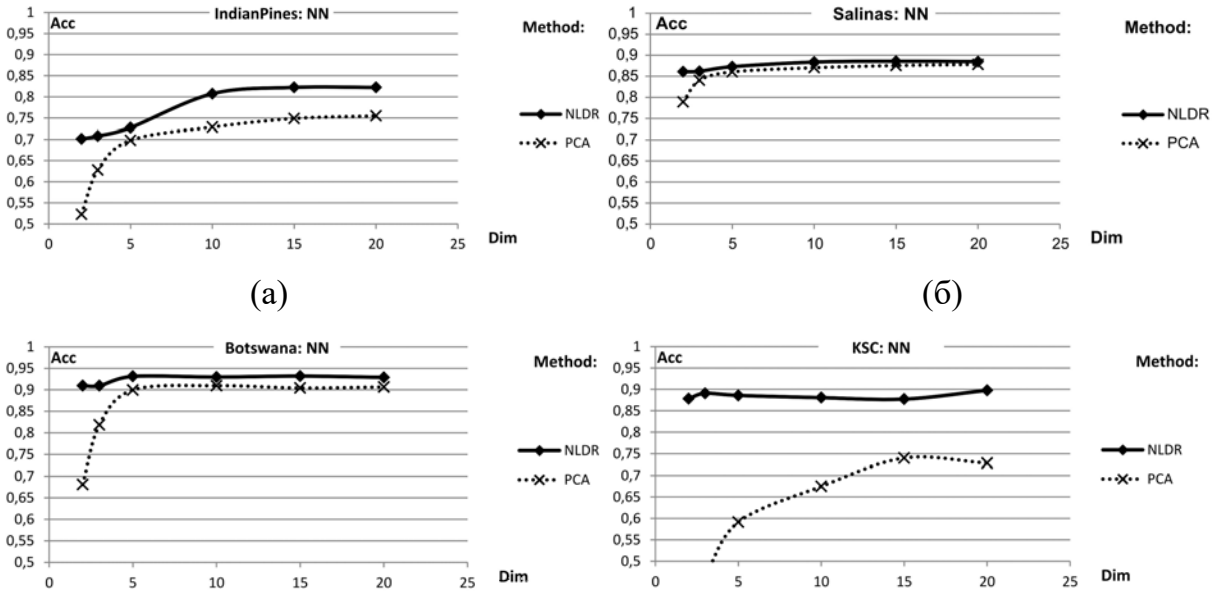


Рисунок 2.12 - Зависимость точности классификации Acc от размерности редуцированного пространства Dim для NN - классификатора для наилучших из нелинейных методов снижения размерности (NLDR) и метода главных компонент (PCA).

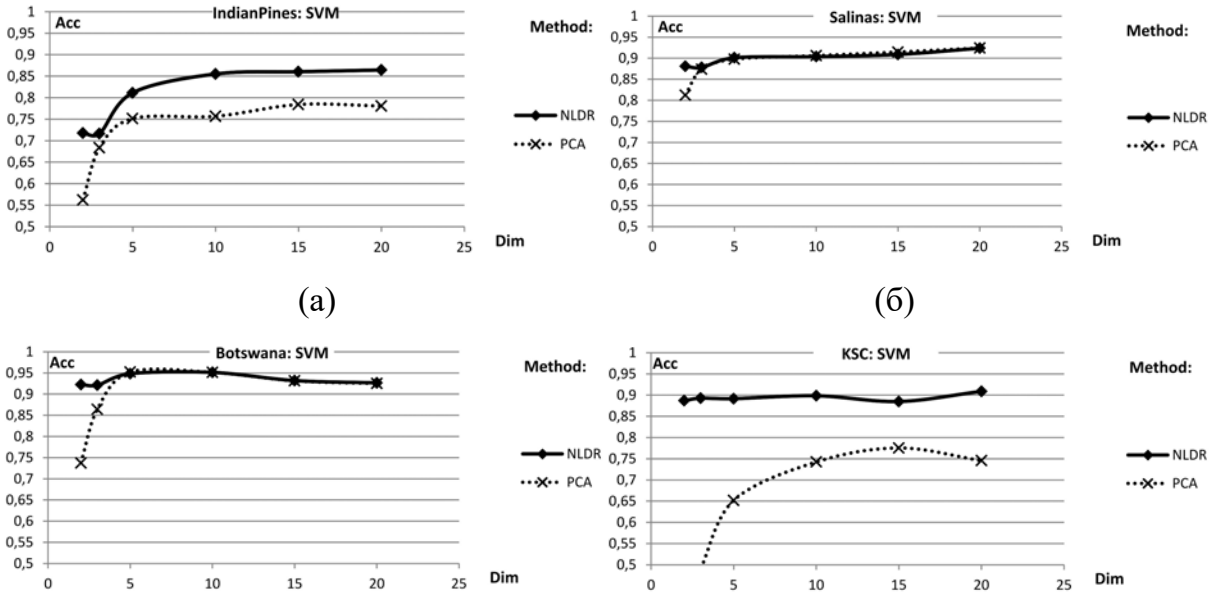


Рисунок 2.13 - Зависимость точности классификации Acc от размерности редуцированного пространства Dim для SVM - классификатора для наилучших из нелинейных методов снижения размерности (NLDR) и метода главных компонент (PCA).

В указанной таблице приведены наилучшие комбинации методов формирования признаков и расстояний для каждого тестового изображения и исследуемой размерности.

Как видно из таблицы, для малых размерностей $\mathrm{Dim} = 2, 3$ наилучшие результаты были получены с использованием метода UMAP в сочетании с различными функциями расстояния. Отметим, что этот метод предлагалось использовать при обработке ГСИ в работе автора [198*].

Для более высоких размерностей формируемых пространств $\mathrm{Dim} = 5…40$ лучшие результаты дают предложенные алгоритмы формирования, основанные на сохранении заданных расстояний (SDPM, HDPM, EDPM, SAPM), помеченные темно-серым цветом), и предложенные алгоритмы, основанные на аппроксимации заданных расстояний во входном пространстве евклидовыми расстояниями в формируемом пространстве (SAED, HDED, BCED), помеченные светло-серым цветом.

Отметим, что во многих случаях при размерности $m >= 10$ с использованием разработанных алгоритмов формирования признаков удалось превзойти качество классификации по исходным данным, независимо от используемой функции расстояния (соответствующие ячейки в таблице выделены жирным). Для сравнения в последней строке «Исходные» показано наивысшее качество классификации, полученное по исходным данным с использованием всех рассмотренных функций расстояния, а также количество каналов в соответствующих ГСИ. Таким образом, качество классификации по сформированным данным оказалось выше качества классификации по исходным данным в 25 из 35 случаев, что составляет более 70%. При этом повышения качества удалось добиться для каждого из рассмотренных изображений.

Что касается выбора функций расстояния, спектральный угол SAM и теоретико-информационные меры, такие как дивергенция спектральной информации (SID) и дивергенция Хеллингера (HD), оказывались более предпочтительными, чем евклидово расстояние, так как обеспечивали наилучшие показатели качества, за исключением сцены Pavia University. При этом прослеживались зависимости наилучшего расстояния от рассматриваемого тестового изображения.

Таблица 2.2 - Наилучшие комбинации мер и методов
(достигнутая точность классификации указана в скобках)

m	**Тестовое ГСИ**				
	Indian Pines	**Salinas**	**Botswana**	**Pavia Univ.**	**KSC**
2	UMAP-SAM (0,701)	UMAP-ED (0,861)	UMAP-SAM (0,909)	UMAP-ED (0,771)	UMAP-SAM (0,873)
3	UMAP-SID (0,708)	UMAP-ED (0,862)	UMAP-SAM (0,909)	UMAP-ED (0,774)	UMAP-SAM (0,882)
5	SAED (0,724)	HDED (0,872)	SAED (0,939)	EDPM (0,813)	UMAP-SAM (0,886)
10	**SDPM (0,801)**	**HDED (0,891)**	HDED (0,938)	**EDPM (0,850)**	UMAP-SAM (0,881)
15	**SDPM (0,821)**	**HDED (0,894)**	HDPM (0,939)	**EDPM (0,849)**	SAPM (0,887)
20	**SDPM (0,823)**	**BCED (0,893)**	HDPM (0,941)	**EDPM (0,894)**	SAED (0,898)
25	**SDPM (0,810)**	**HDED (0,892)**	BCED (0,943)	**EDPM (0,852)**	**SAED (0,920)**
30	**SDPM (0,802)**	**HDED (0,897)**	HDPM (0,938)	EDPM (0,842)	**SAED (0,909)**
35	**SDPM (0,825)**	**BCED (0,892)**	**BCED (0,946)**	EDPM (0,845)	**HDED(0,908)**
40	**SDPM (0,827)**	**BCPM (0,886)**	**HDPM (0,945)**	**EDPM (0,853)**	**SAPM (0,911)**
Исход-ные	0,775 (*M*=200)	0,886 (*M*=204)	0,944 (*M*=145)	0,846 (*M*=103)	0,903 (*M*=176)

Выводы и результаты по главе 2

Во второй главе разработан ряд алгоритмов формирования признаков ГСИ, основанных на сохранении заданных расстояний. В частности, предложены алгоритмы, работающие на основе принципа сохранения l_p метрики, спектральных углов (в декартовой и гиперсферической системах координат), спектральной корреляции, дивергенции спектральной информации, дивергенции Хеллингера, угла Бхаттачария. Необходимо отметить, что использование таких расстояний, как угол Бхаттачария и дивергенция Хеллингера, обладающих свойствами истинных метрик, открывает возможности для применения в сформированных пространствах методов ускорения поиска ближайших соседей, что положительно сказывается на эффективности решения прикладных задач.

Проведено исследование ряда методов снижения размерности общего назначения (Isomap, Locally Linear Embedding, Laplacian Eigenmaps, UMAP) при совместном использовании с известными функциями расстояния в спектральном пространстве, а также линейного метода главных компонент (PCA). Исследованы

разработанные в главе алгоритмы формирования признаков, основанные как на сохранении заданных расстояний, так и на аппроксимации заданных расстояний евклидовыми расстояниями.

Экспериментально показано, что при размерностях формируемого пространства 5…40 разработанные алгоритмы позволяют достичь более высокого качества формируемого признакового описания по сравнению с рассмотренными альтернативами (качество оценивалось как точность попиксельной классификации гиперспектральных изображений). Более того, качество классификации по сформированным данным оказалось выше качества классификации по исходным данным в 25 из 35 случаев, что составляет более 70%. При этом повышения качества удалось добиться для каждого из рассмотренных изображений. Стоит отметить, что для гиперспектральных сцен, содержащих зашумленные данные или существенные нелинейные эффекты, традиционный метод PCA демонстрировал неудовлетворительные результаты.

3 Учет пространственно-спектральных характеристик и слияние систем признаков при формировании информативных признаков

До настоящего момента все рассмотренные методы формирования информативных признаков работали только в одном входном пространстве - пространстве спектров пикселей гиперспектральных изображений. Решение прикладных задач часто можно улучшить, если дополнительно учитывать признаковую информацию различной природы, в частности, локальные пространственные или текстурные характеристики гиперспектральных изображений.

На сегодняшний день можно выделить несколько основных подходов, используемых для учета локальной пространственной информации гиперспектральных изображений. Один из возможных подходов заключается в объединении результатов попиксельной классификации, выполненной на основе спектральных признаков, с результатами неуправляемой сегментации (см., например, [287, 273]).

Другой подход состоит в использовании композитных функций ядра, учитывающих как спектральную, так и пространственную информацию при построении SVM классификаторов (см., например, [35, 76])

Еще один подход заключается в использовании функций расстояния, которые учитывают как пространственный контекст, так и спектральные особенности пикселей изображения. Этот подход может быть использован с алгоритмами формирования признаков вполне естественным образом. Самый простой способ состоит в аппроксимации значений таких функций расстояния евклидовыми расстояниями с использованием алгоритмов из раздела 2.3.

Для построения новых функций расстояния могут использоваться различные схемы встраивания пространственного контекста. Эти функции расстояния также могут быть основаны на различных мерах спектрального рассогласования, таких как евклидово расстояние или спектральный угол.

В этой главе разрабатываются две схемы встраивания пространственного контекста с использованием функций расстояния. Первая основана на использовании оконных функций, вторая - на использовании порядковых статистик. Обе схемы были

введены в работе автора [189*] для евклидова расстояния, выбранного в качестве базовой функции расстояния, и позднее расширены в [186*] для спектральных углов.

Кроме того, в настоящей главе разрабатывается подход к объединению систем признаков, основанный на приведении исходных систем признаков к однородным (в смысле свойств соответствующих пространств) промежуточным представлениям с последующим слиянием этих представлений и формированием признаков. Предложенный подход впервые описан автором в работе [200*] и позднее применялся в работах [182*, 209*] для совместного учета спектральной и пространственной информации.

3.1 Оконные функции

Первая схема внедрения пространственного контекста состоит в расширении исходного пространства признаков значениями соседних пикселей. Хорошо известная из обработки изображений реализация учета пространственного контекста в попиксельной классификации и сегментации изображений использует 4- или 8-пиксельное соседство. В более общем случае можно рассматривать пространственную пиксельную окрестность с некоторым радиусом R.

С одной стороны, в неуправляемых процедурах снижения размерности не используется никакой информации об истинной классификации, и отсутствует возможность построить какую-либо процедуру, которая присваивает больший вес более информативным пикселям. С другой стороны, интуитивно понятно, что более близкие пиксели должны оказывать большее влияние на различие, чем более дальние. Также ясно, что при отсутствии предварительной информации об изображаемой сцене эти веса не должны зависеть от пространственного направления. Эти соображения приводят нас к использованию двумерных радиально-симметричных оконных функций в качестве весовых функций.

Расширим пространство пикселей исходного гиперспектрального изображения путем конкатенации координат пикселей, входящих в локальную окрестность пикселей изображения. Для этого вместо пикселя исходного изображения x_i, i=1..N, представляющего собой вектор размерности M, будем рассматривать расширенный вектор

$$\widetilde{x}_i = \left(w_1 x_{i_1 1}, w_1 x_{i_1 2}, \ldots w_1 x_{i_1 M}, w_2 x_{i_2 1}, w_2 x_{i_2 2}, \ldots w_2 x_{i_2 M}, \ldots w_K x_{i_K 1}, w_K x_{i_K 2}, \ldots w_K x_{i_K M}\right), \quad (3.1)$$

образованный из всех компонент K пикселей $x_{i_1}, x_{i_2}, \ldots x_{i_K}$, входящих в локальную окрестность пикселя x_i, взятых с неотрицательными весами $w_1, w_2, \ldots w_K$. Здесь и далее будем считать, что пиксель x_i входит в собственную окрестность.

В этом случае функция расстояния $\tilde{d}$ между i-м и j-м пикселями изображения может быть задана как взвешенное евклидово расстояние в расширенном пространстве пикселей:

$$\tilde{d}(\tilde{x}_i, \tilde{x}_j) = \left(\sum_{k=1}^{K} w_k^2 \sum_{l=1}^{M} \left(x_{i_k l} - x_{j_k l} \right)^2 \right)^{1/2}.$$

Здесь каждый весовой коэффициент w_k определяется значением оконной функции $w(r)$, а ее аргумент r является евклидовым расстоянием на изображении между пикселем x_i и k-м пикселем из его окрестности:

$$w_k = w\left(r(x_i, x_{i_k}) \right).$$

В обработке сигналов было предложено и широко используется большое число оконных функций. Примерами таких функций являются следующие оконные функции [100]: прямоугольные, треугольные, Гаусса, Блэкмена, Хана, Хэмминга, Натала [222] и др. В работе автора [189*] использовались оконные функции на основе следующего выражения:

$$w(r) = \begin{cases} 1 - (r/R)^p, & \text{если } (r \in [0; R]) \text{ и } (p > 0), \\ (1 - r/R)^{-p}, & \text{если } (r \in [0; R]) \text{ и } (p < 0), \\ 0 & \text{иначе.} \end{cases} \qquad (3.2)$$

Для некоторых значений p это выражение приводит к некоторым специальным окнам: треугольному окну для $p = 1$ (или $p = -1$), прямоугольному окну (Rect) для $p = \infty$. При $p = -\infty$ оконная функция $w(r)$ соответствует функции

$$\delta(r) = \begin{cases} 1, & \text{если } (r = 0), \\ 0, & \text{иначе,} \end{cases} \qquad (3.3)$$

означающей, что пространственный контекст не используется.

Вид окон при различных значениях параметра p показан на рисунке 3.1.

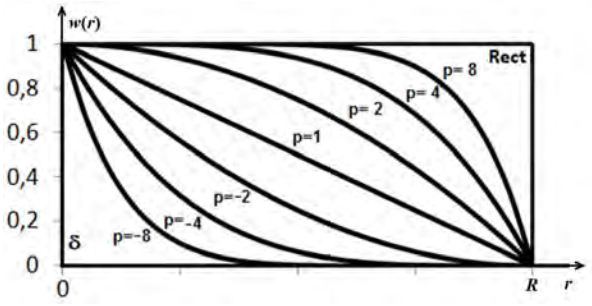


Рисунок 3.1 – Графики функции окна $w(r)$ при различных значениях параметра p.

3.1.1.1 Разделение выборки

Для представленных в настоящем разделе исследований использовался набор тестовых гиперспектральных изображений, описанный ранее в п. 2.4.1: Indian pines, Salinas, Pavia University, Kennedy Space Center.

Для решения задачи попиксельной классификации необходимо тем или иным образом разделить данные - сформировать обучающую и тестовую выборки. Описанные выше тестовые изображения не комплектуются стандартными разбиениями, поэтому чаще всего используется случайное разбиение. Именно такое разбиение использовалось ранее в Главе 2. Однако такой подход к разбиению может давать переоцененные показатели качества для методов, использующих помимо спектральной также пространственную информацию. Причина этого заключается в том, что при использовании пространственной информации признаковое пространство содержит информацию не только о самом пикселе, но и о его локальной окрестности. Например, для метода, изложенного в разделе 3.1, признаковое пространство содержит информацию о пикселях, расположенных на изображении в пределах радиуса R от рассматриваемого. При случайном разбиении в локальную окрестность пикселей из обучающего множества попадают пиксели не только из обучающего, но и из тестового множества. Аналогично, в локальную окрестность пикселя из тестового множества попадают пиксели не только из тестового, но и из обучающего множества. Таким образом, при использовании случайного разбиения существует взаимное проникновение информации из обучающего и тестового множеств, происходящее посредством пространственного контекста.

По этой причине в настоящем подразделе, помимо использованного нами ранее случайного разбиения, также использовался следующий способ разбиения (далее – фиксированное разбиение). Для каждой размеченной области изображения, принадлежащей одному классу, определялось, вдоль какой оси область является

более вытянутой, после чего область разбивалась на две равные части ортогонально этой оси. Примеры полученных разбиений показаны на рисунке 3.2.

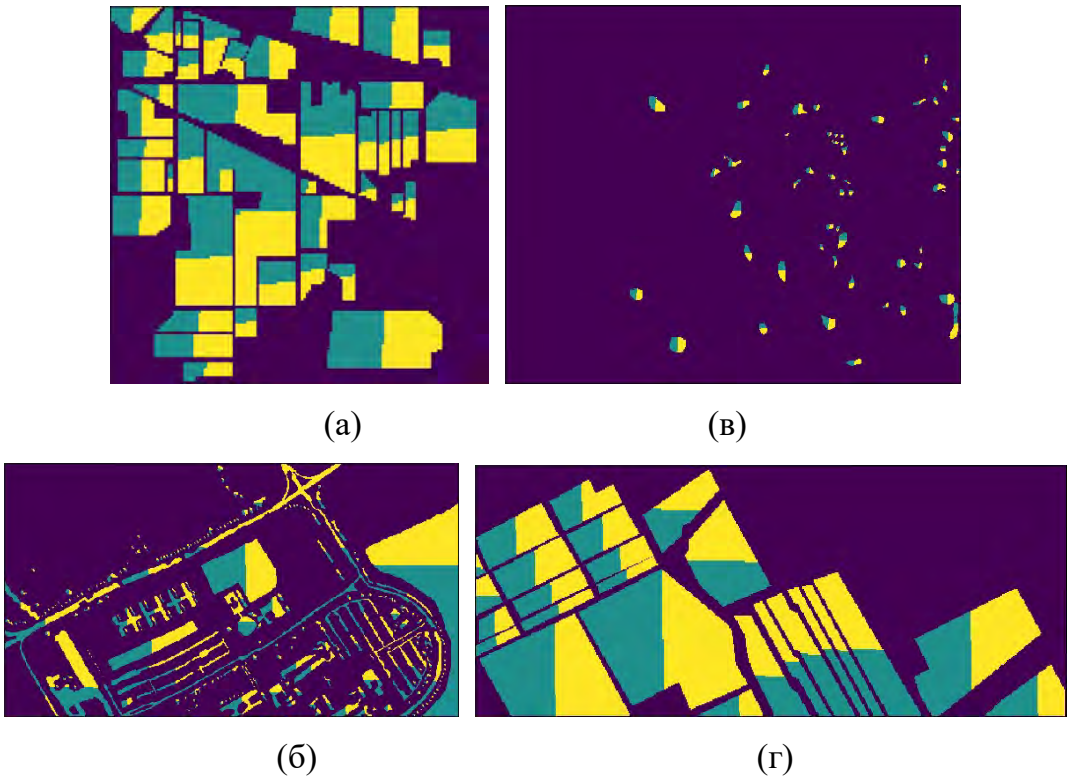


Рисунок 3.2 – Пример разбиения на обучающую и тестовую выборки для изображений Indian Pines (а), Pavia University (б), Kennedy Space Center (в), Salinas (г). Изображения масштабированы, (б, г) повернуты.

Одна из полученных половин использовалась для обучения, а вторая - для тестирования, после чего обучающая и тестовая выборки менялись местами, а результаты усреднялись.

3.1.1.2 Численные эксперименты

Приведем некоторые численные результаты экспериментов с использованием пространственного контекста на основе функций окна для решения формирования информативных признаков гиперспектральных данных с последующей классификацией. При проведении экспериментов использовалось гиперспектральное изображение IndianPines (см. п. 2.4.1). Для выполнения экспериментов все множество пикселей изображения с известной истинной классификацией разделялось на две равные части, одна из которых использовалась для обучения, а другая – для тестирования, после чего обучающая и тестовая выборки менялись местами, а

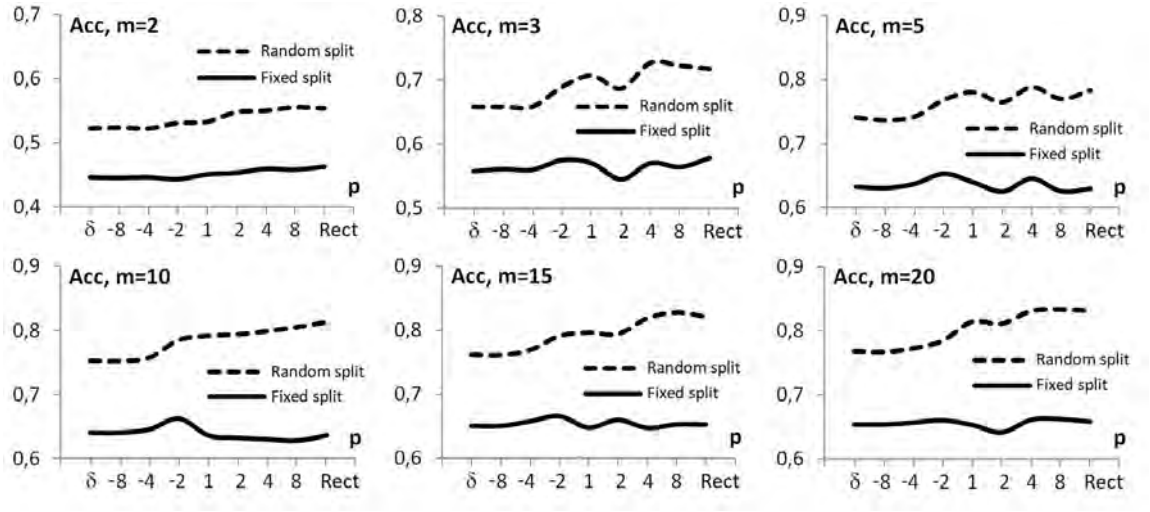
результаты усреднялись. Как было сказано выше, разбиение осуществлялось двумя способами: случайным и фиксированным. В качестве классификатора использовался классификатор по ближайшему соседу (NN).

В рамках экспериментов варьировался размер окрестности, задаваемый радиусом R, параметр p функции окна (3.2), а также размерность выходного пространства m. Некоторые результаты экспериментов показаны на рисунке 3.3.

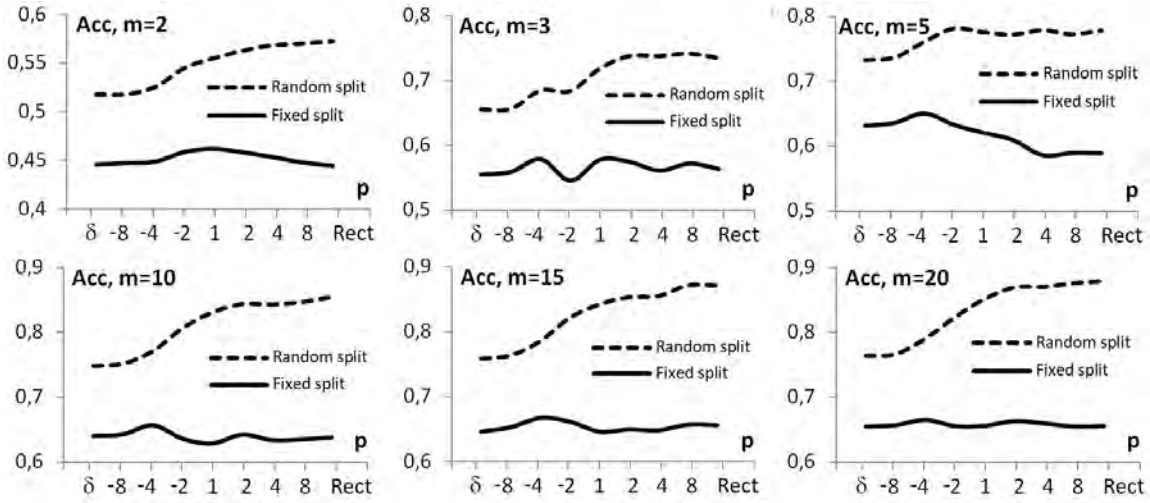
На представленном рисунке под буквой «а» показаны результаты при использовании окрестности радиусом R=1, под «б» - радиусом R=2, под «в» - радиусом R=3. Отдельные графики соответствуют различным значениям выходной размерности $m \in \{2, 3, 5, 10, 15, 20\}$. По оси абсцисс на каждом графике отложены различные значения параметра p оконной функции (3.2). При этом самые левые значения p соответствуют оконной функции δ, заданной выражением (3.3) и означающей, что пространственный контекст при снижении размерности не использовался. Самые правые значения соответствуют прямоугольной функции окна (Rect), которая равна 1 для всех пикселей окрестности заданного радиуса R. Промежуточные значения $p \in \{-8, -4, -2, 1, 2, 4, 8\}$ задают вид функции окна в соответствии с выражением (3.2). По оси ординат на графиках отложена достигнутая точность классификации Acc. На каждом графике показано по две кривые, соответствующие случайному (Random split) и фиксированному (Fixed split) разбиению.

Как можно видеть, во всех рассмотренных случаях при использовании случайного разбиения точность классификации была существенно повышена путем использования пространственного контекста с $p > 1$ по сравнению с точностью при δ (без пространственного контекста). При этом, как правило, большие значения p соответствовали большей точности классификации, и наилучшее качество классификации обеспечивалось функцией окна прямоугольной формы (Rect).

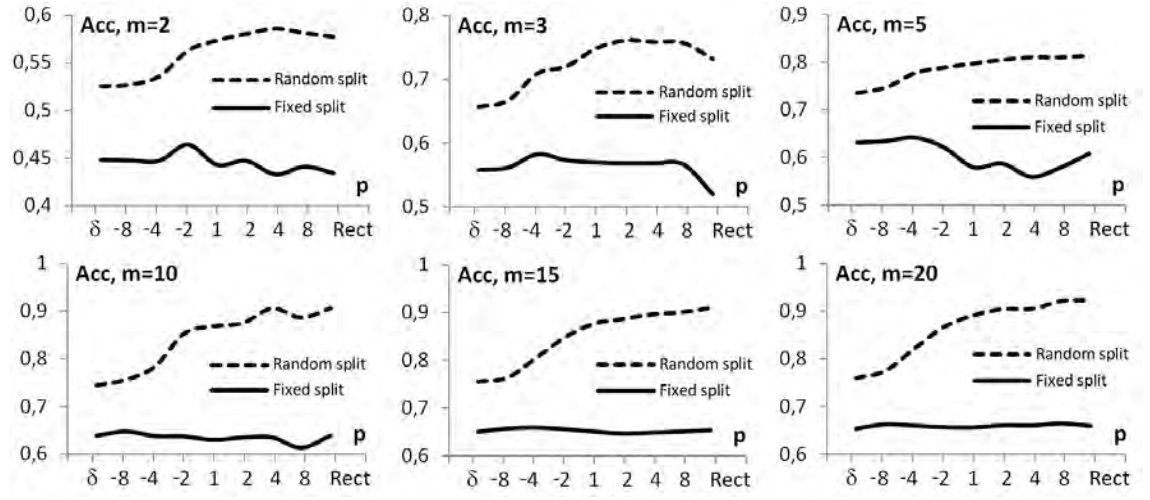
При использовании фиксированного разбиения, напротив, невозможно говорить о повышении точности классификации с ростом параметра p. Наибольшие значения точности были получены для функций окна вогнутой формы (при $p \leq -2$) и отличались от значений, полученных без пространственного контекста (при δ), не так значительно, как при случайном разбиении.



(а)



(б)



(в)

Рисунок 3.3 - Зависимость точности классификации (*Acc*) от параметра *p* функции окна (3.2) для указанной размерности *m* редуцированного пространства R^m, и различного радиуса соседства *R*: радиус *R* = 1 (а), радиус соседства *R* = 2 (б), радиус соседства *R* = 3 (в). Значения *δ* и *Rect* соответствуют частным случаям функции окна: функции (3.3) и прямоугольной функции окна.

Точность классификации для размерностей выходного пространства m=5..20 показана также в таблице 3.1.

Таблица 3.1 – Качество классификации по сформированным признакам, полученным с использованием функций окна.

Размер-ность, *m*	**Точность классификации,** *Acc*				**Прирост точности классификации**
	Без контекста (при δ**)**	**Треугольная функция (при** p=1**)**	**Прямоугольная функция** *(Rect)*	**Наилучшая точность**	
Случайное разбиение. Радиус окрестности R=1					
5	0,737	0,781	0,783	0,789	0,052
10	0,747	0,792	0,813	0,813	0,066
15	0,758	0,797	0,821	0,828	0,070
20	0,762	0,815	0,832	0,834	0,071
Случайное разбиение. Радиус окрестности R=2					
5	0,737	0,775	0,773	0,781	0,045
10	0,747	0,831	0,847	0,855	0,108
15	0,758	0,842	0,872	0,872	0,114
20	0,762	0,852	0,876	0,879	0,117
Случайное разбиение. Радиус окрестности R=3					
5	0,737	0,797	0,813	0,813	0,077
10	0,747	0,869	0,906	0,906	0,159
15	0,758	0,877	0,910	0,910	0,152
20	0,762	0,891	0,923	0,923	0,161
Фиксированное разбиение. Радиус окрестности R=1					
5	0,632	0,640	0,630	0,654	0,021
10	0,640	0,636	0,636	0,662	0,023
15	0,650	0,648	0,653	0,666	0,017
20	0,654	0,654	0,659	0,663	0,009
Фиксированное разбиение. Радиус окрестности R=2					
5	0,632	0,621	0,589	0,651	0,018
10	0,640	0,629	0,638	0,657	0,017
15	0,650	0,646	0,656	0,667	0,017
20	0,654	0,656	0,656	0,665	0,011
Фиксированное разбиение. Радиус окрестности R=3					
5	0,632	0,580	0,608	0,642	0,010
10	0,640	0,631	0,640	0,648	0,009
15	0,650	0,651	0,654	0,660	0,010
20	0,654	0,657	0,660	0,665	0,011

В этой таблице представлена точность классификации для признаков, сформированных без учета пространственного контекста, а также с использованием пространственного контекста на основе треугольной и прямоугольной функций окна. Кроме того, в таблице приводится наилучшая (максимальная) точность

классификации, достигнутая при использовании окон с различными параметрами $p \in \{-8, -4, -2, 1, 2, 4, 8\}$, а также прямоугольного окна. Также в последнем столбце показан прирост точности классификации по сравнению с результатами, полученными без использования пространственного контекста.

Из представленной таблицы можно сделать вывод о том, что использование пространственного контекста на основе оконных функций при случайном разбиении позволяет значительно улучшить качество классификации. При этом прирост качества составляет 0,045…0,161 или 4,5-16,1%. При использовании фиксированного разбиения качество классификации оказывается заметно ниже, а прирост качества менее значителен и не превышает 2,3%.

3.2 Порядковые статистики

Хотя описанная выше схема позволяет учитывать пространственный контекст пикселей изображения, она работает неадаптивно, назначая одинаковые веса одинаково удаленным пикселям в пространственной окрестности, независимо от их спектральных характеристик. Кроме того, описанная схема на основе оконных функций не обеспечивает инвариантности к повороту, так как положение пикселей из локальной окрестности в расширенном векторе (3.1) определяется только порядком обхода пикселей окрестности.

Чтобы преодолеть эти недостатки, имеет смысл использовать адаптивное взвешивание, основанное на сходстве пикселей в окрестности. Эта идея была реализована в работах автора [189*, 186*] в схеме с использованием порядковых статистик окрестности пикселя.

Эта схема состоит в сортировке всех пикселей

$$x_{i_1}, x_{i_2}, ... x_{i_K}$$

из окрестности пикселя x_i по близости к центральному пикселю x_i:

$$d(x_i, x_{(i_1)}) \le d(x_i, x_{(i_2)}) \le ... \le d(x_i, x_{(i_K)}).$$

Здесь переупорядоченный набор пикселей

$$x_{(i_1)}, x_{(i_2)}, ... x_{(i_K)}$$

состоит из порядковых статистик, а d является функцией расстояния между пикселями изображения в спектральном пространстве.

Теперь мы можем усечь этот набор и использовать некоторое число $S \leq K$ порядковых статистик в новой функции расстояния:

$$\widetilde{d}(x_i, x_j) = \eta \sum_{s=1}^{S} w_s \; d\left(x_{(i_s)}, x_{(j_s)}\right) \tag{3.4}$$

Для расчета весов w_s здесь будем использовать обратное взвешивание $w_s = 1/s$, S - количество порядковых статистик, используемых в качестве пространственного контекста.

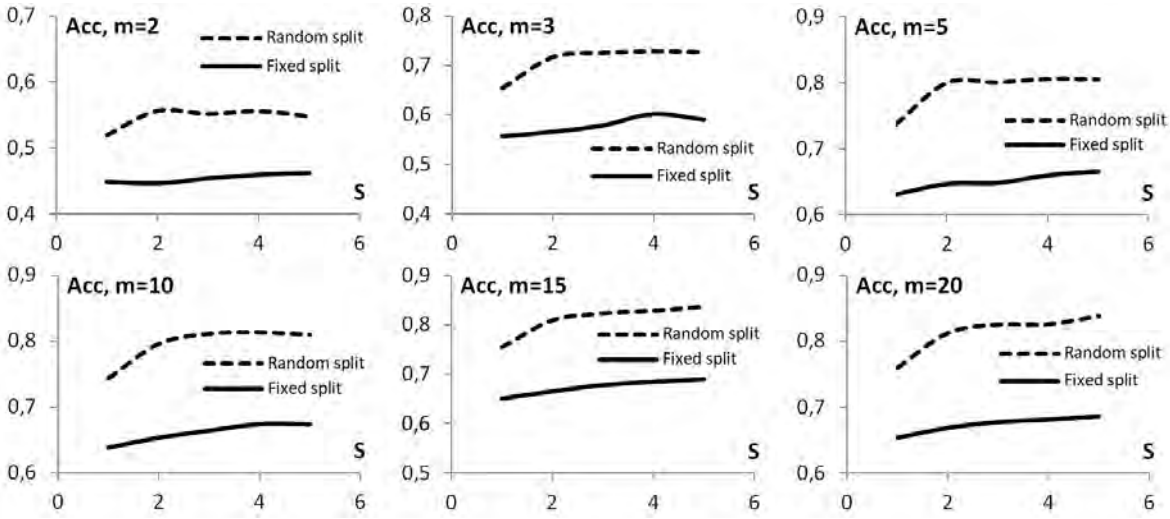
Выражение (3.4) готово к использованию с неевклидовыми функциями расстояния, поскольку любая неотрицательная мера может использоваться как мера d спектрального различия между пикселями изображения.

3.2.1.1 Численные эксперименты для случая Евклидовых расстояний

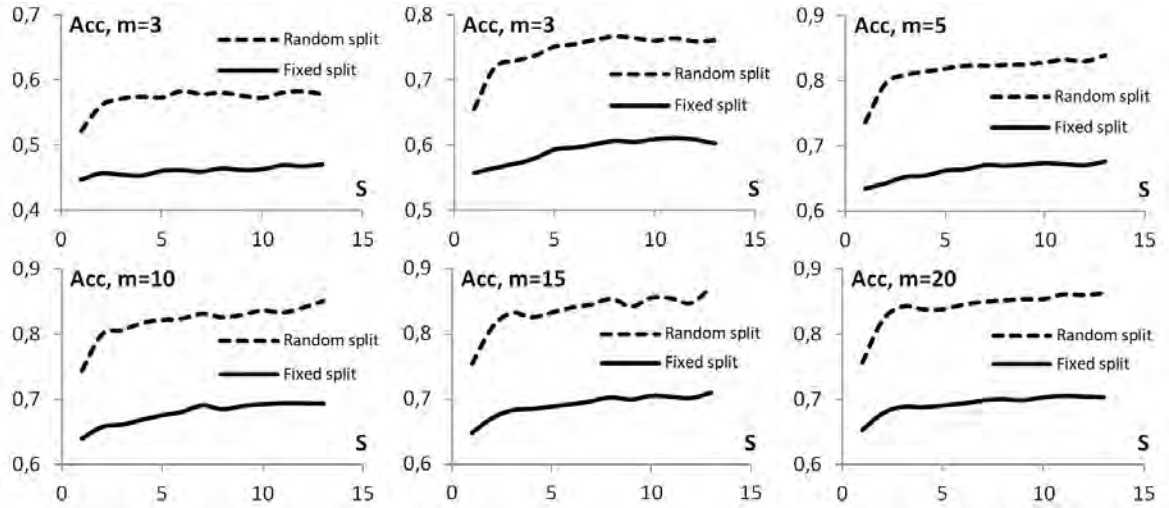
Приведем некоторые экспериментальные результаты для описанной схемы на основе порядковых статистик для случая, когда в качестве функции расстояния в спектральном пространстве d в выражении (3.4) используется Евклидово расстояние. В рамках экспериментов, проводимых на том же материале и по той же схеме, что и в п. 3.1, будем по-прежнему варьировать размер окрестности, задаваемой радиусом (R = 1, 2, 3), размерность формируемого пространства m = 2,3,5,10,15,20, а также количество S используемых порядковых статистик. Результаты приведены на рисунке 3.4.

Здесь самое левое значение S = 1 соответствует случаю, когда использовался только центральный пиксель пространственного соседства и пространственный контекст не использовался. Значение S = 2 означало, что первый ближайший (в спектральном пространстве) сосед из пространственного соседства использовался в качестве пространственного контекста, и т.д. Самые правые значения S соответствуют случаям, когда в вычислениях использовалось все пространственное соседство.

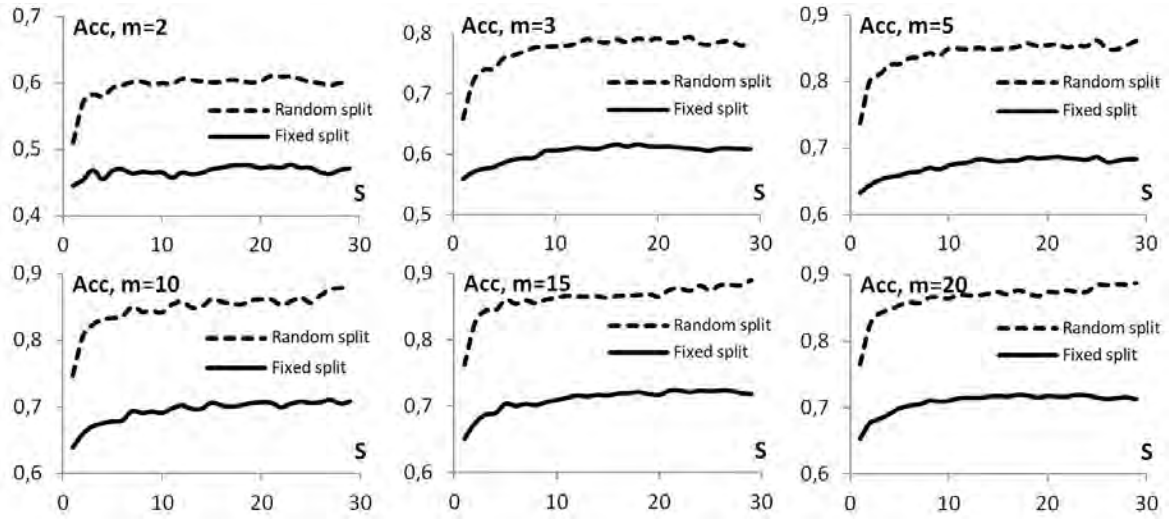
Как можно видеть из рисунка, во всех рассмотренных случаях точность классификации повышена путем использования пространственного контекста при формировании признаков.



(а)



(б)



(в)

Рисунок 3.4 - Зависимость точности классификации (*Acc*) от количества *S* порядковых статистик на основе евклидова расстояния для заданной размерности *m* формируемого пространства R^m, и различного радиуса соседства *R*: радиус *R* = 1 (а), радиус соседства *R* = 2 (б), радиус соседства *R* = 3 (в).

При этом независимо от того, использовалось ли случайное или фиксированное разбиение, качество классификации повышалось с ростом числа использованных порядковых статистик.

В таблице 3.2 для размерностей выходного пространства m=5..20 и радиусов окрестности R=1..3 представлена точность классификации для случаев формирования признаков без учета пространственного контекста, с использованием половины, а также с использованием всех порядковых статистик.

Таблица 3.2 – Качество классификации по сформированным признакам, полученным с использованием порядковых статистик и Евклидовых расстояний.

Размер-ность, m	**Точность классификации,** Acc				**Прирост точности классификации**
	Без контекста	**Половина статистик**	**Все статистики**	**Наилучшая**	
Случайное разбиение. Радиус окрестности R=1					
5	0,737	0,800	0,805	0,806	0,069
10	0,747	0,811	0,810	0,813	0,067
15	0,758	0,823	0,836	0,836	0,078
20	0,762	0,825	0,838	0,838	0,076
Случайное разбиение. Радиус окрестности R=2					
5	0,737	0,823	0,839	0,839	0,102
10	0,747	0,831	0,850	0,850	0,104
15	0,758	0,845	0,871	0,871	0,114
20	0,762	0,849	0,863	0,863	0,100
Случайное разбиение. Радиус окрестности R=3					
5	0,737	0,849	0,861	0,862	0,126
10	0,747	0,862	0,881	0,881	0,135
15	0,758	0,863	0,890	0,890	0,132
20	0,762	0,874	0,887	0,887	0,125
Фиксированное разбиение. Радиус окрестности R=1					
5	0,632	0,648	0,666	0,666	0,033
10	0,640	0,665	0,674	0,674	0,035
15	0,650	0,678	0,690	0,690	0,040
20	0,654	0,678	0,686	0,686	0,032
Фиксированное разбиение. Радиус окрестности R=2					
5	0,632	0,671	0,676	0,676	0,044
10	0,640	0,691	0,693	0,694	0,054
15	0,650	0,697	0,710	0,710	0,060
20	0,654	0,698	0,703	0,706	0,052
Фиксированное разбиение. Радиус окрестности R=3					
5	0,632	0,679	0,683	0,687	0,055
10	0,640	0,706	0,708	0,711	0,071
15	0,650	0,716	0,718	0,723	0,073
20	0,654	0,717	0,713	0,719	0,065

Как и в случае с оконными функциями, в таблице приводится наилучшая (максимальная) точность классификации, достигнутая при использовании порядковых статистик, а также прирост точности классификации по сравнению с результатами, полученными без использования пространственного контекста.

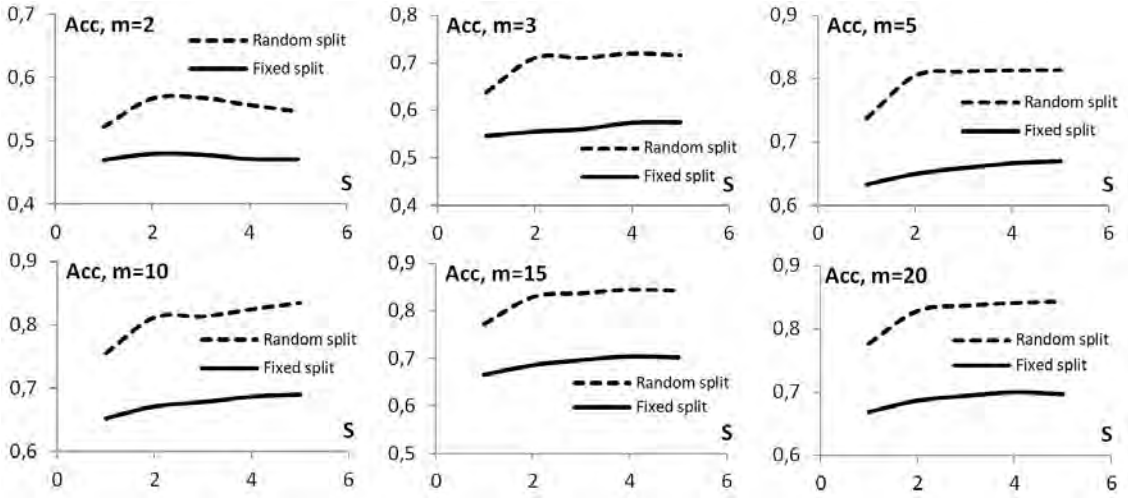
Как следует из представленной таблицы, использование пространственного контекста на основе порядковых статистик позволяет получить прирост качества на 0,067…0,135, т.е. на 6,7-13,5% при случайном разбиении и на 0,032…0,073, т.е. на 3,2-7,3% при фиксированном разбиении.

Сравнение метода на основе оконных функций и метода на основе порядковых статистик показывает преимущество последнего, что проявляется при использовании фиксированного разбиения. В зависимости от радиуса окрестности и размерности формируемого пространства прирост качества составляет 1,2% – 6,3%.

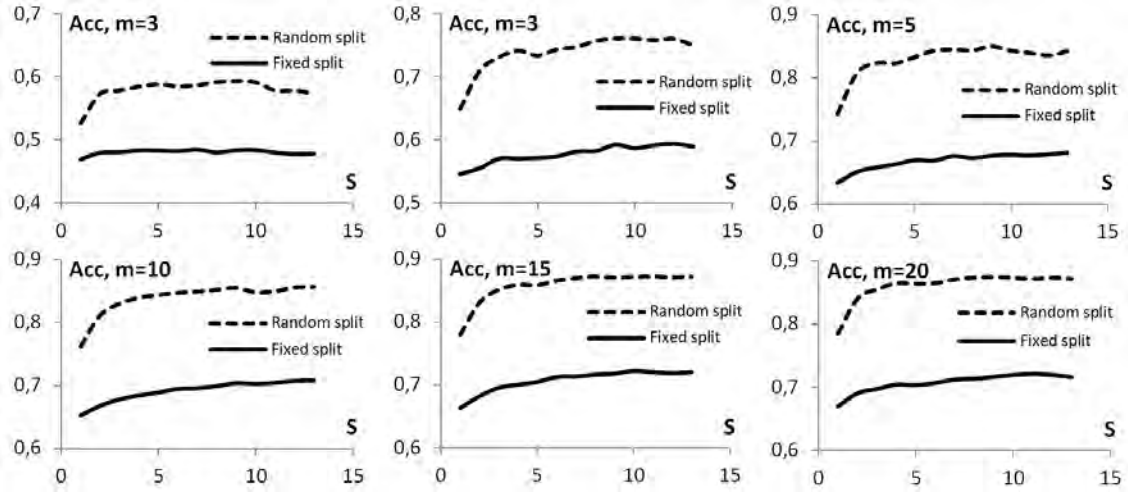
3.2.1.2 Численные эксперименты для случая спектральных углов

Обе рассмотренные в настоящей главе схемы использования пространственного контекста допускают использование неевклидовых функций расстояния. По этой причине, в следующем эксперименте, изложенном в работе автора [194*], исследуется схема на основе порядковых статистик с использованием спектральных углов. В рамках этого эксперимента радиус окрестности принимал значения R = 1,2,3 размерность формируемого пространства - значения m = 2,3,5,10,15,20. Кроме того, варьировалось количество S используемых порядковых статистик. Некоторые результаты этого эксперимента показаны на рисунке 3.5.

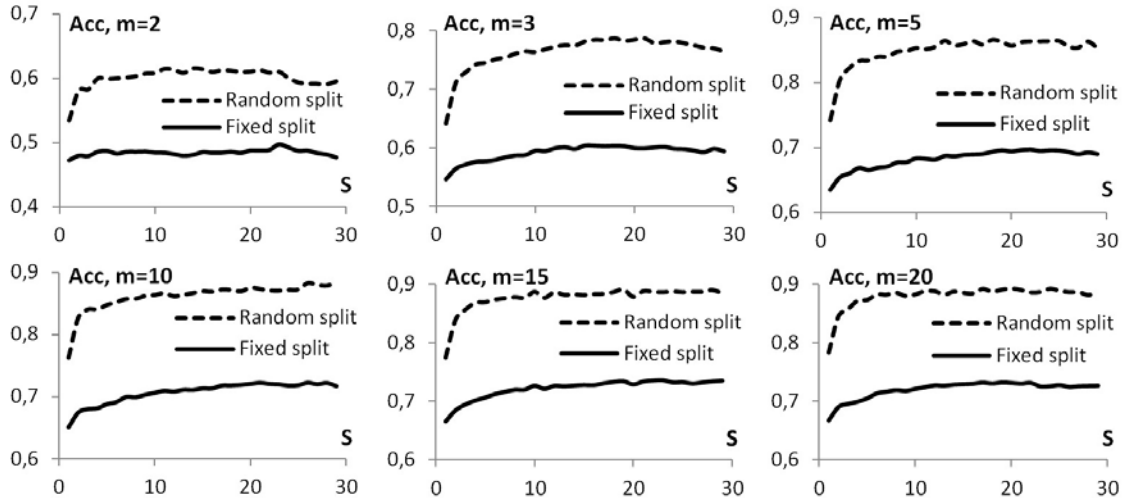
Как можно видеть, почти во всех рассмотренных случаях, чем больше порядковых статистик использовалась, тем выше было качество классификации.



(а)



(б)



(в)

Рисунок 3.5 - Зависимость точности классификации (*Acc*) от количества *S* порядковых статистик на основе спектрального угла для заданной размерности *m* формируемого пространства R^m, и различного радиуса соседства *R*: радиус *R* = 1 (а), радиус соседства *R* = 2 (б), радиус соседства *R* = 3 (в).

В таблице 3.3 для размерностей формируемого пространства m=5..20 и радиусов окрестности *R*=1..3 представлена точность классификации для случаев без учета пространственного контекста, с использованием половины, а также всех порядковых статистик. Как и в предыдущем случае, использование пространственного контекста на основе порядковых статистик позволило получить существенный прирост качества. Для случайного разбиения прирост составил 6,1%-12,5%, в то время как для фиксированного разбиения – 3.2%-7.1%.

Таблица 3.3 – Качество классификации по сформированным признакам, полученным с использованием порядковых статистик и спектральных углов.

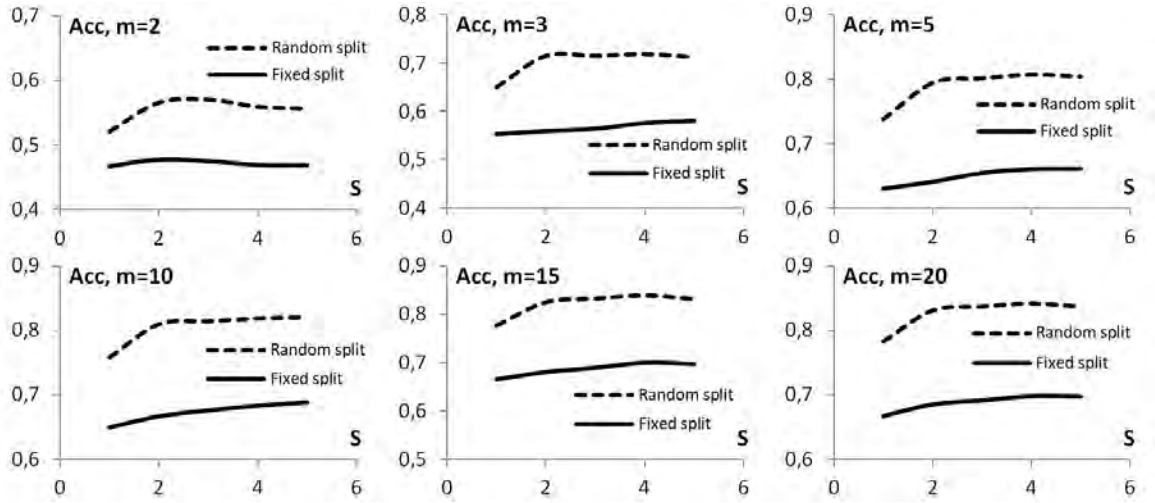
Размерность, *m*	**Точность классификации, *Acc***				**Прирост точности классификации**
	Без контекста	**Половина статистик**	**Все статистики**	**Наилучшая**	
Случайное разбиение. Радиус окрестности *R*=1					
5	0,741	0,811	0,814	0,814	0,073
10	0,760	0,813	0,835	0,835	0,075
15	0,776	0,838	0,843	0,845	0,069
20	0,782	0,837	0,843	0,843	0,061
Случайное разбиение. Радиус окрестности *R*=2					
5	0,741	0,845	0,844	0,850	0,109
10	0,760	0,849	0,857	0,857	0,097
15	0,776	0,869	0,872	0,872	0,096
20	0,782	0,871	0,872	0,875	0,093
Случайное разбиение. Радиус окрестности *R*=3					
5	0,741	0,859	0,853	0,866	0,125
10	0,760	0,870	0,884	0,884	0,124
15	0,776	0,882	0,885	0,891	0,116
20	0,782	0,885	0,887	0,893	0,111
Фиксированное разбиение. Радиус окрестности *R*=1					
5	0,634	0,659	0,670	0,670	0,036
10	0,652	0,678	0,690	0,690	0,038
15	0,665	0,697	0,702	0,705	0,039
20	0,669	0,694	0,697	0,701	0,032
Фиксированное разбиение. Радиус окрестности *R*=2					
5	0,634	0,676	0,682	0,682	0,048
10	0,652	0,696	0,708	0,708	0,056
15	0,665	0,713	0,720	0,722	0,057
20	0,669	0,712	0,716	0,721	0,052
Фиксированное разбиение. Радиус окрестности *R*=3					
5	0,634	0,689	0,691	0,697	0,063
10	0,652	0,714	0,717	0,723	0,071
15	0,665	0,728	0,735	0,736	0,071
20	0,669	0,729	0,727	0,732	0,064

3.2.1.3 Численные эксперименты для случая дивергенции Хеллингера

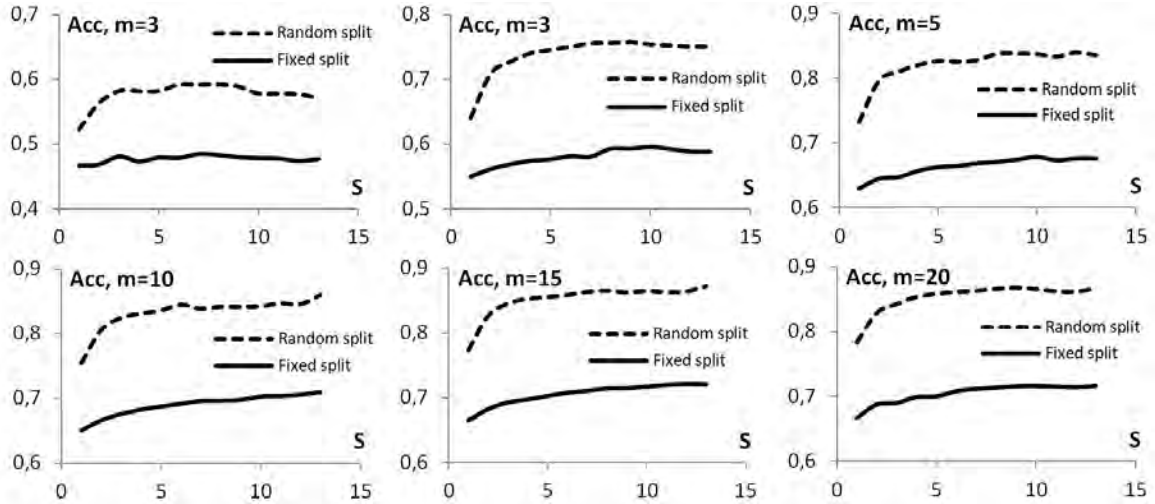
Эксперименты, аналогичные проведенным в предыдущих двух пунктах, были выполнены для случая использования теоретико-информационной функции расстояния - дивергенции Хеллингера. Результаты этих экспериментов представлены на рисунке 3.6 и в таблице 3.4.

Таблица 3.4 – Качество классификации по сформированным признакам, полученным с использованием порядковых статистик и дивергенции Хеллингера.

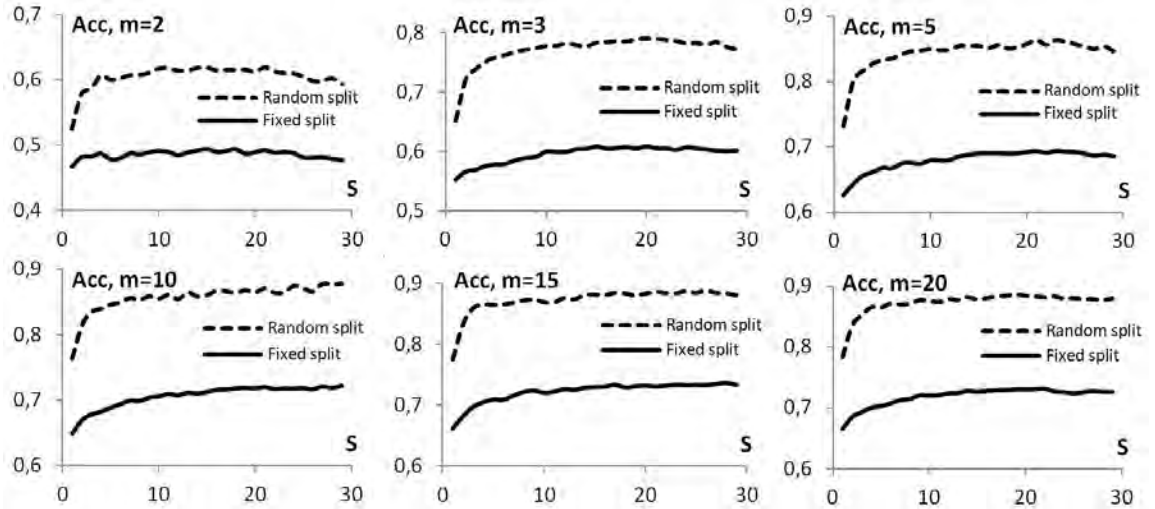
Размер-ность, m	**Точность классификации,** *Acc*				**Прирост точности классификации**
	Без контекста	**Половина статистик**	**Все статистики**	**Наилучшая**	
Случайное разбиение. Радиус окрестности R=1					
5	0,734	0,801	0,804	0,807	0,073
10	0,759	0,814	0,820	0,820	0,061
15	0,775	0,833	0,832	0,839	0,064
20	0,783	0,837	0,837	0,842	0,058
Случайное разбиение. Радиус окрестности R=2					
5	0,734	0,828	0,835	0,839	0,105
10	0,759	0,839	0,860	0,860	0,101
15	0,775	0,863	0,872	0,872	0,097
20	0,783	0,863	0,866	0,867	0,084
Случайное разбиение. Радиус окрестности R=3					
5	0,734	0,855	0,846	0,864	0,130
10	0,759	0,860	0,878	0,878	0,119
15	0,775	0,882	0,881	0,888	0,113
20	0,783	0,879	0,880	0,887	0,103
Фиксированное разбиение. Радиус окрестности R=1					
5	0,629	0,655	0,661	0,661	0,032
10	0,650	0,676	0,689	0,689	0,039
15	0,664	0,690	0,698	0,700	0,036
20	0,667	0,692	0,698	0,698	0,032
Фиксированное разбиение. Радиус окрестности R=2					
5	0,629	0,668	0,675	0,678	0,049
10	0,650	0,696	0,710	0,710	0,060
15	0,664	0,711	0,720	0,721	0,057
20	0,667	0,711	0,716	0,716	0,049
Фиксированное разбиение. Радиус окрестности R=3					
5	0,629	0,690	0,686	0,694	0,065
10	0,650	0,712	0,722	0,722	0,072
15	0,664	0,730	0,734	0,737	0,073
20	0,667	0,727	0,727	0,732	0,065



(а)



(б)



(в)

Рисунок 3.6 - Зависимость точности классификации (*Acc*) от количества *S* порядковых статистик на основе дивергенции Хелингера для заданной размерности *m* формируемого пространства R^m, и различного радиуса соседства *R*: радиус *R* = 1 (а), радиус соседства *R* = 2 (б), радиус соседства *R* = 3 (в).

В целом, сделанные ранее наблюдения для случая евклидовых расстояний и спектральных углов справедливы и для дивергенции Хеллингера. Поэтому приведем здесь лишь сведения о приросте качества за счет использования пространственного контекста. Такой прирост для случая случайного разбиения составил 5,8%-13,0%, а для случая фиксированного разбиения – 3,2%-7,3%.

3.3 Слияние признаков

3.3.1 Способы объединения систем признаков

Объединение систем признаков может быть не вполне очевидным по следующим причинам [249]:

- явная связь между системами признаков может быть неизвестна,

- системы признаков могут быть несовместимы (признаки переменной и фиксированной размерности),

- простая конкатенация признаков может приводить к пространству слишком высокой размерности, что будет приводить к неудовлетворительным результатам решения прикладных задач (например, при обучении по малому числу примеров).

Дополнительно причинами могут стать:

- необходимость использовать разные метрики или меры схожести для сопоставления объектов в тех или иных признаковых пространствах,

- существенно различная размерность систем признаков и др.

Некоторые из приведенных причин могут быть устранены путем нормализации и применения методов отбора признаков или снижения размерности. Пример схемы объединения, основанной на нормализации с последующим отбором признаков, показан на рисунке 3.7. В приведенной схеме производится объединение двух различных систем биометрических признаков: признаков лица и геометрии руки.

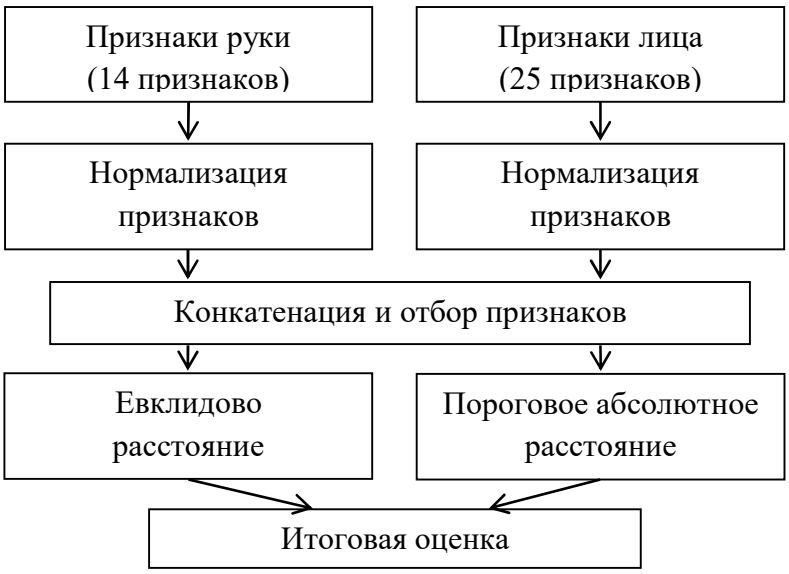


Рисунок 3.7 - Схема объединения признаков из работы [249]

На первом этапе выполняется нормализация признаков, которая может быть быть выполнена по минимаксной схеме [248]

$$x'_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}, \tag{3.5}$$

которая является очень чувствительной к выбросам в данных. По этой причине чаще применяется нормализация с приведением дисперсии к единичному значению.

$$x'_i = \frac{x_i - \bar{x}}{Dx}, \tag{3.6}$$

где $\bar{x}$ - выборочное среднее значение векторов x_i, Dx – оценка их дисперсии.

Также устойчивой к выбросам в данных является нормализация с использованием медианы [248]:

$$x'_i = \frac{x_i - med(x)}{med\left(\left|x - med(x)\right|\right)}. \tag{3.7}$$

Оценки медианы, также как оценки среднего, минимума или максимума в приведенных выше выражениях выполняются покомпонентно по имеющемуся набору данных.

На приведенной выше схеме после нормализации, применяемой к каждой из систем признаков независимо, выполняется этап объединения, который реализуется путем конкатенации с последующим отбором признаков.

Для объединенных признаков рассчитываются два различных типа расстояний: евклидово и пороговое абсолютное расстояние [248]:

$$d_{tad}(x_i, x_j) = \sum_{k=1}^{M} I(\left|x_{ik} - x_{jk}\right|, t), \quad I(a,t) = \begin{cases} 1, & \text{если } a < t, \\ 0, & \text{иначе.} \end{cases}$$

На заключительном этапе расстояния объединяются простым суммированием.

Так как объединение в приведенной схеме основывается на непосредственной конкатенации признаков, указанная схема не может быть применена для несовместимых признаков. Кроме того, эта схема не может применяться и для систем признаков, требующих применения различных расстояний, так как расчет расстояний производится уже после конкатенации признаков. Предлагаемый ниже подход свободен от указанных ограничений.

3.3.2 Предлагаемый подход

Описываемый ниже подход к объединению систем признаков, предложенный автором в работах [182*, 200*], основан на приведении исходных систем признаков к однородным (в смысле свойств соответствующих пространств) промежуточным представлениям с последующим слиянием этих представлений и формированием единого признакового пространства. Общая схема подхода показана на рисунке 3.8.

Как видно из приведенной схемы, на вход поступают два набора векторов признаков, представленных в двух пространствах (в общем случае разной размерности), на каждом из которых определена некоторая функция расстояния. Функции расстояния в общем случае также могут быть различными. Важным элементом в рассматриваемом подходе является выбор параметров внутренней формы представления, а именно размерности промежуточного пространства и определенной на нем метрики. Так как основной задачей внутреннего представления является согласование между собой исходных данных, представленных в различных пространствах, представляется целесообразным формирование пространств с одной метрикой. Учитывая, что важнейшим требованием к формируемым внутренним представлениям является сохранение попарных различий между векторами, размерность внутренних представлений может превышать размерность исходного пространства. Другим важным требованием к внутреннему представлению является

вычислительная эффективность процедуры формирования внутреннего представления.

Рисунок 3.8 - Схема предлагаемого подхода к объединению признаков

Отметим, что если масштаб формируемых представлений различен, то есть попарные расстояния в одной системе признаков существенно отличаются от расстояний в другой системе признаков, процедура формирования внутреннего представления должна предусматривать нормализацию формируемых представлений тем или иным образом, например, с использованием (3.5)-(3.7).

Собственно процедура слияния внутренних представлений, выполняемая с учетом согласования, может быть реализована, например, путем конкатенации признаковых представлений.

Размерность результата, формируемого в результате слияния промежуточных представлений, может оказаться избыточной, что приводит к необходимости снижения размерности при формировании итогового признакового представления. Здесь мы опять приходим к необходимости выбора размерности выходного пространства и метрики. Такой выбор целесообразно осуществлять, исходя из эффективности решения требуемых прикладных задач.

3.3.3 Использование евклидовой метрики для промежуточного представления данных

Так как наиболее часто используемой является евклидова метрика, рассмотрим этот частный случай. Пусть в исходных пространствах $X \subset R^M$ и $Y \subset R^K$ заданы некоторые функции расстояния $\varphi(x_i, x_j)$ и $\psi(y_i, y_j)$, где $x_i, x_j \in X$, $y_i, y_j \in Y$. Предположим, существует способ отображения исходных пространств X и Y в пространства $\widetilde{X} \subset R^{\widetilde{M}}$ и $\widetilde{Y} \subset R^{\widetilde{K}}$, размерностей $\widetilde{M}$ и $\widetilde{K}$ соответственно, обеспечивающий точное соответствие евклидовых расстояний попарным расстояниям в исходных пространствах:

$$d(\widetilde{x}_i, \widetilde{x}_j) = \varphi(x_i, x_j),$$

$$d(\widetilde{y}_i, \widetilde{y}_j) = \psi(y_i, y_j),$$

где $\widetilde{x}_i, \widetilde{x}_j \in \widetilde{X}$, $x_i, x_j \in X$, $\widetilde{y}_i, \widetilde{y}_j \in \widetilde{Y}$, $y_i, y_j \in Y$.

В том случае, когда этап нормализации реализуется линейным преобразованием с некоторыми масштабными коэффициентами λ_X и λ_Y, расстояния после нормализации будут равны $\widetilde{d}_{ij}^{X} = \lambda_X d(\widetilde{x}_i, \widetilde{x}_j)$ и $\widetilde{d}_{ij}^{Y} = \lambda_Y d(\widetilde{y}_i, \widetilde{y}_j)$ для пространств $\widetilde{X}$ и $\widetilde{Y}$ соответственно.

После слияния признаков в результате конкатенации расстояния будут рассчитываться следующим образом:

$$d(z_i, z_j) = \left(\sum_{k=1}^{\widetilde{M}+\widetilde{K}} \left(z_{ik} - z_{jk}\right)^2 \right)^{1/2} = \left(\sum_{k=1}^{\widetilde{M}} \left(z_{ik} - z_{jk}\right)^2 + \sum_{k=\widetilde{M}+1}^{\widetilde{M}+\widetilde{K}} \left(z_{ik} - z_{jk}\right)^2 \right)^{1/2} =$$

$$= \left(\left(\widetilde{d}_{ij}^{X}\right)^2 + \left(\widetilde{d}_{ij}^{Y}\right)^2 \right)^{1/2} = \left(\lambda_X^2 d^2(\widetilde{x}_i, \widetilde{x}_j) + \lambda_Y^2 d^2(\widetilde{y}_i, \widetilde{y}_j) \right)^{1/2} =$$

$$= \left(\lambda_X^2 \varphi^2(x_i, x_j) + \lambda_Y^2 \psi^2(y_i, y_j) \right)^{1/2}.$$

Таким образом, для случая промежуточных представлений с евклидовой метрикой расчет попарных расстояний на этапе снижения размерности может быть выполнен с использованием приведенного выше выражения, минуя явное преобразование исходных данных во внутреннюю форму представления, без связанных с этим вычислительных затрат и погрешностей. Кроме того, отсутствие необходимости в явном преобразовании во внутреннюю форму представления освобождает и от необходимости определения размерности такого представления.

Отметим, что аналогичный результат может быть легко получен и для более широкого класса l_p метрик.

$$d(z_i, z_j) = \left(\sum_{k=1}^{\tilde{M}+\tilde{K}} \left(z_{ik} - z_{jk} \right)^p \right)^{1/p} = \left(\sum_{k=1}^{\tilde{M}} \left(z_{ik} - z_{jk} \right)^p + \sum_{k=\tilde{M}+1}^{\tilde{M}+\tilde{K}} \left(z_{ik} - z_{jk} \right)^p \right)^{1/p} =$$

$$= \left(\left(\tilde{d}_{ij}^{\,X} \right)^p + \left(\tilde{d}_{ij}^{\,Y} \right)^p \right)^{1/p} = \left(\lambda_X^p d^p(\tilde{x}_i, \tilde{x}_j) + \lambda_Y^p d^p(\tilde{y}_i, \tilde{y}_j) \right)^{1/p} =$$

$$= \left(\lambda_X^p \varphi^p(x_i, x_j) + \lambda_Y^p \psi^p(y_i, y_j) \right)^{1/p}.$$

Для метрики l_∞ получим

$$d(z_i, z_j) = \max_{k=1}^{\tilde{M}+\tilde{K}} \left| z_{ik} - z_{jk} \right| = \max\left(\max_{k=1}^{\tilde{M}} \left| z_{ik} - z_{jk} \right|, \max_{k=\tilde{M}+1}^{\tilde{M}+\tilde{K}} \left| z_{ik} - z_{jk} \right| \right) =$$

$$= \max\left(\tilde{d}_{ij}^{\,X}, \tilde{d}_{ij}^{\,Y} \right) = \max\left(\lambda_X d(\tilde{x}_i, \tilde{x}_j), \lambda_Y d(\tilde{y}_i, \tilde{y}_j) \right) =$$

$$= \max\left(\lambda_X \varphi(x_i, x_j), \lambda_Y \psi(y_i, y_j) \right).$$

3.3.4 Демонстрация подхода на примере слияния спектральных и текстурных признаков

Для демонстрации предлагаемого подхода рассмотрим задачу формирования информативных признаков гиперспектральных изображений с последующей попиксельной классификацией.

В соответствии с предложенной схемой будем объединять спектральные и текстурные признаки изображений. Будем использовать пиксели изображения непосредственно в качестве спектральных признаков после шага нормализации. Для сравнения спектральных признаков будем использовать такие функции расстояния как спектральный угол и дивергенция Хеллингера. Автором было показано [192*],

что последняя функция расстояния позволяет эффективно решать задачи классификации, обладая при этом свойствами истинной метрики.

В качестве текстурных признаков будем использовать широко используемый набор признаков, рассчитываемых на основе матрицы совместной встречаемости пикселей [99]: контрастность, корреляцию, энергию и однородность. Матрицу совместной встречаемости будем рассчитывать с использованием одноканального изображения, полученного как проекция исходных гиперспектральных данных на первую главную компоненту. Матрицы совместной встречаемости будем формировать для четырех различных смещений ([0, 1], [-1, 1], [-1, 0], [-1, -1]) и 64-уровневого квантования. После вычисления усредненные текстурные признаки будут нормализоваться по стандартным отклонениям. В качестве функции расстояния для текстурных признаков будем использовать евклидово расстояние.

Выполним слияние признаков по схеме, приведенной в предыдущем разделе. Для шага формирования признаков будем использовать алгоритм формирования признаков ГСИ на основе аппроксимации заданных расстояний евклидовыми расстояниями (см. п. 2.3), реализованный с использованием процедуры стохастического градиентного спуска (см. подраздел 4.2).

Классификация в выходном пространстве осуществляется двумя способами: методом k ближайших соседей (k-NN) и методом опорных векторов (SVM) с радиальным ядром. В качестве меры качества используется общая точность классификации (*Acc*), рассчитанная как доля верно классифицированных пикселей тестового подмножества.

Общая схема для экспериментального исследования показана на рисунке 3.9.

Рисунок 3.9 - Схема вычислений для экспериментальных исследований.

В первом наборе экспериментов в качестве тестовой гиперспектральной сцены используется изображение Indian Pines (см. п. 2.4.1). Некоторые результаты, показывающие зависимость качества классификации от масштабного коэффициента, показаны на рисунке 3.10. Здесь масштабный коэффициент λ определяет влияние спектральных признаков, а (1-λ) - влияние текстурных признаков. Как видно из результатов, текстурные признаки играют важную роль в качестве классификации для рассматриваемой тестовой сцены. Сочетание как спектральных, так и текстурных признаков позволяет значительно повысить качество классификации.

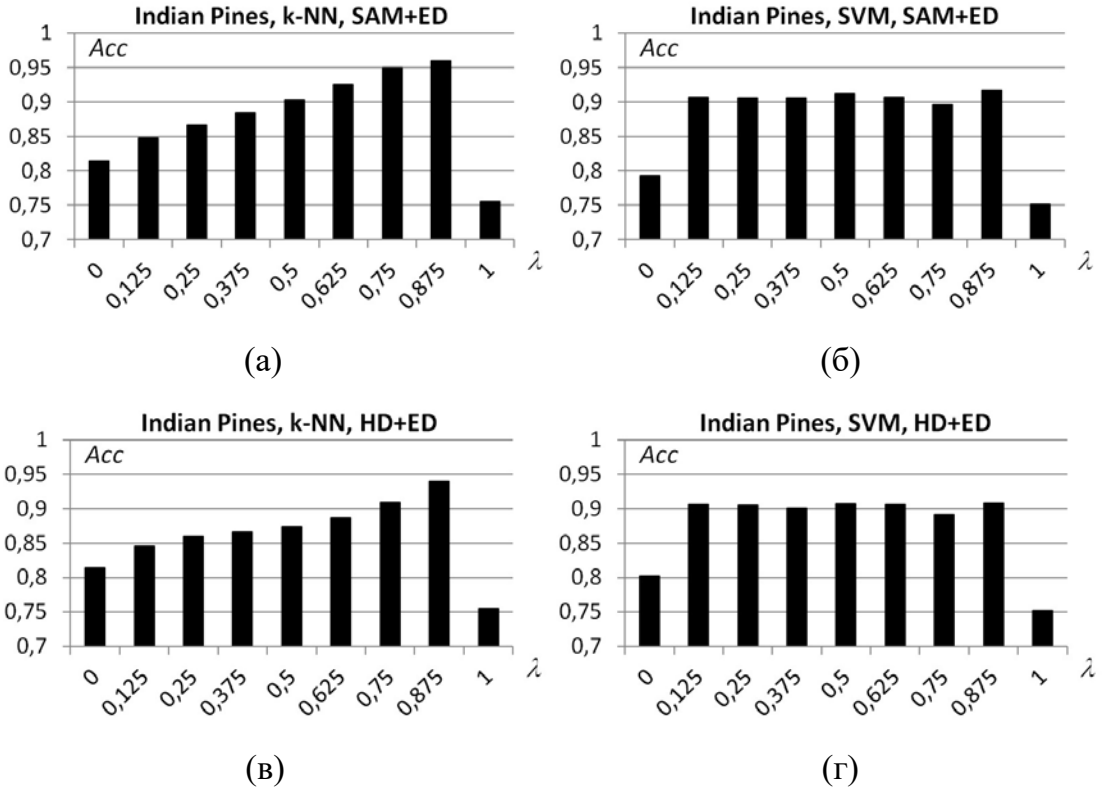


Рисунок 3.10 - Зависимость точности классификации *Acc* от масштабного коэффициента λ для размерности D=10 (сцена Indian Pines). В первой строке представлены результаты для спектрального угла в спектральном пространстве и евклидова расстояния в текстурном пространстве с классификатором k-ближайших соседей (а) и SVM классификатором (б). Во второй строке представлены результаты для дивергенции Хеллингера в спектральном пространстве и евклидова расстояния в текстурном пространстве с классификатором k-ближайших соседей (в) и SVM классификатором (г).

На рисунке 3.11 представлена зависимость качества классификации *Acc* от размерности формируемого выходного пространства *D*. Как видно из рисунка, объединение спектральных и текстурных признаков позволяет формировать пространства относительно небольшой размерности (D=10, 15, 20), качество классификации в котором значительно выше, чем для каждой из систем признаков в отдельности.

Аналогичные результаты для другого гиперспектрального изображения Salinas (см. п. 2.4.1) показаны на рисунках 3.12, 3.13. Для снижения объема вычислений это изображение было пространственно прорежено с шагом 3.

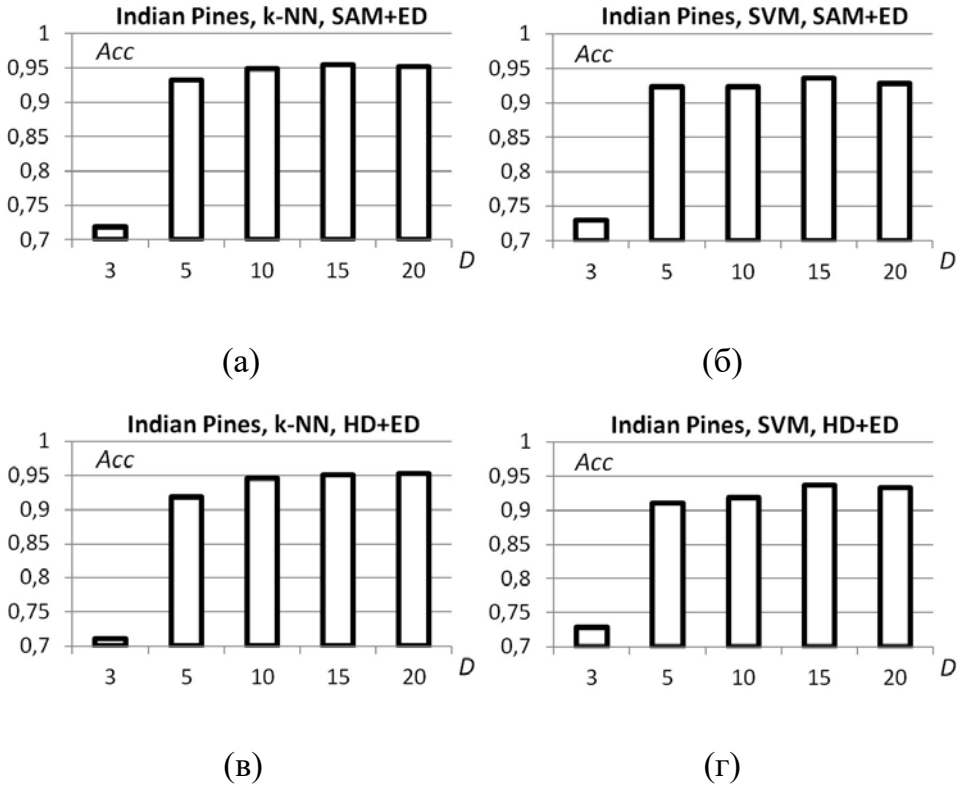


Рисунок 3.11 - Зависимость точности классификации *Acc*
от размерности *D* формируемого пространства (сцена Indian Pines).
В первой строке представлены результаты для спектрального угла в спектральном пространстве и евклидова расстояния в текстурном пространстве с классификатором k-ближайших соседей (а) и SVM классификатором (б).
Во второй строке представлены результаты для дивергенции Хеллингера в спектральном пространстве и евклидова расстояния в текстурном пространстве с классификатором k-ближайших соседей (в) и SVM классификатором (г).

На рисунке 3.12 показана зависимость качества классификации от масштабного коэффициента для изображения Salinas. На рисунке 3.13 показана зависимость качества классификации от размерности выходного пространства. В целом представленные результаты для сцены Salinas подтверждают описанные выше результаты, полученные для Indian Pines.

Стоит отметить, что представленные результаты, вероятно, могут быть дополнительно улучшены за счет более тщательного подбора параметров. Основная цель этих экспериментов — продемонстрировать возможность и целесообразность слияния разнородных систем признаков в рамках разработанной схемы.

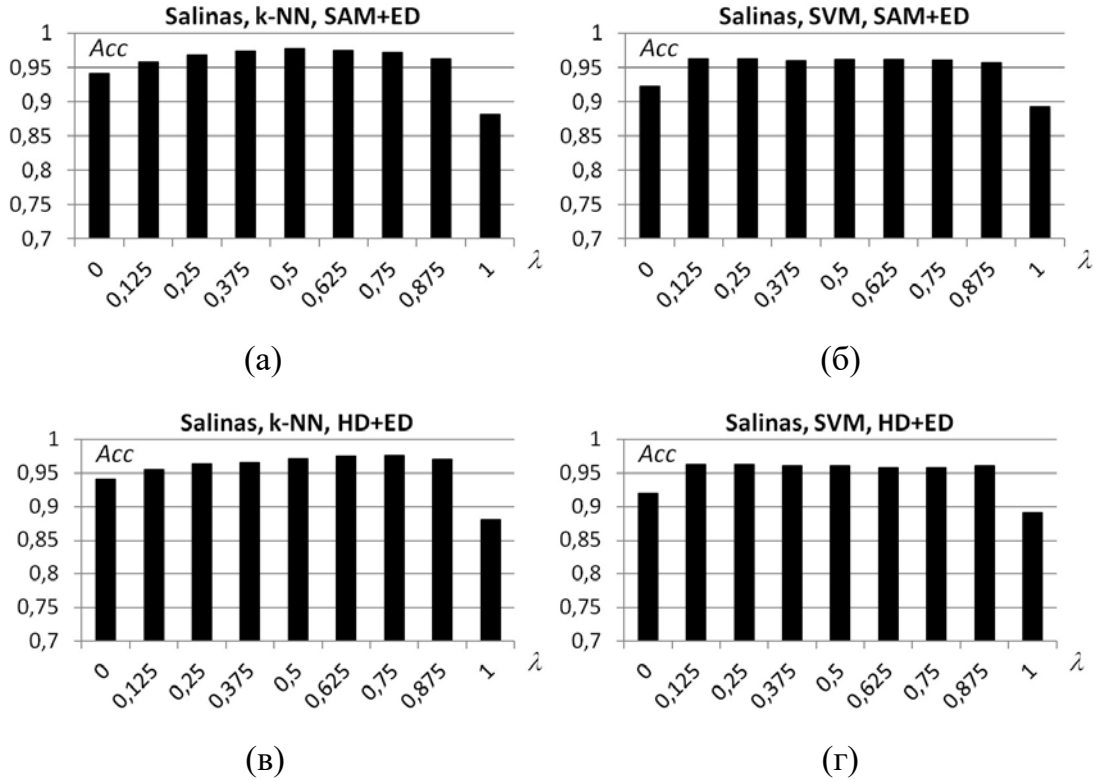


Рисунок 3.12 - Зависимость точности классификации *Acc* от масштабного коэффициента λ для размерности формируемого пространства *D*=10 (сцена Salinas). В первой строке представлены результаты для спектрального угла в спектральном пространстве и евклидова расстояния в текстурном пространстве с классификатором k-ближайших соседей (а) и SVM (б).
Во второй строке представлены результаты для дивергенции Хеллингера в спектральном пространстве и евклидова расстояния в текстурном пространстве с классификатором k-ближайших соседей (в) и SVM классификатором (г).

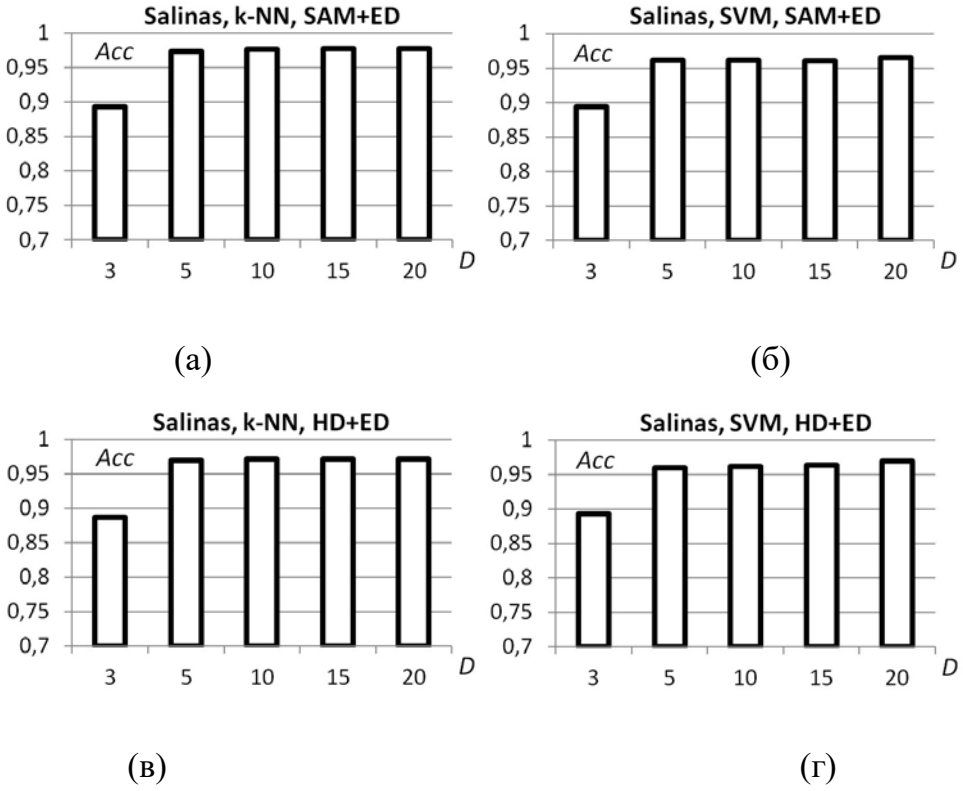


Рисунок 3.13 - Зависимость точности классификации *Acc* от размерности *D* формируемого пространства (сцена Salinas).
В первой строке представлены результаты для спектрального угла в спектральном пространстве и евклидова расстояния в текстурном пространстве с классификатором k-ближайших соседей (а) и SVM (б).
Во второй строке представлены результаты для дивергенции Хеллингера в спектральном пространстве и евклидова расстояния в текстурном пространстве с классификатором k-ближайших соседей (в) и SVM (г).

3.4 Слияние спектральных признаков с признаками, извлеченными из гиперспектральных изображений глубокими нейросетями

Как говорилось в п. 1.1, в последние годы в анализе изображений, в частности, гиперспектральных изображений дистанционного зондирования Земли, все чаще используются искусственные нейронные сети. Для обучения таких сетей, в особенности нейросетей глубокого обучения, требуется размеченная выборка достаточно большого объема. При этом ручная разметка нужного объема данных для обучения сетей требует значительных затрат времени и ресурсов, так как связана с осуществлением наземных измерений.

По этой причине привлекательной возможностью для анализа гиперспектральных изображений ДЗЗ может быть использование нейросетей, обученных на данных другого типа, например, на цветных или мультиспектральных данных ДЗЗ. В этом случае необходимо создание некоторого механизма, позволяющего, во-первых, использовать такие сети с гиперспектральными данными, а во-вторых, использовать преимущества высокого спектрального разрешения гиперспектральных изображений.

В настоящем подразделе описывается применение, изложенного выше в п. 3.3 подхода, к слиянию спектральных признаков с признаками, извлеченными из гиперспектральных изображений глубокими нейросетями. Описываемая в разделе реализация подхода представлена в работах автора [209*, 210*]. Основная идея заключается в слиянии расстояний между признаками, извлеченными с использованием глубоких нейросетей, с расстояниями между пикселями в гиперспектральном пространстве, рассчитываемыми с использованием выбранной функции расстояния. Полученные расстояния позволяют формировать новое признаковое пространство пониженной размерности, в котором могут решаться различные прикладные задачи, в том числе, задача попиксельной классификации.

Схема слияния подобна изложенной выше и показана на рисунке 3.14. На этом рисунке обработка гиперспектрального изображения выполняется по двум независимым ветвям. В левой ветви после выделения каналов, соответствующих натуральным цветам, с использованием одной из глубоких нейросетей (на рисунке Resnet18) выполняется извлечение признаков, часто называемых также вложениями (англ., embeddings). Такие признаки извлекаются для пространственных окрестностей каждого из анализируемых пикселей изображения. После этого между любыми двумя вложениями e_i и e_j, соответствующими окрестностям пикселей i и j может быть рассчитано евклидово расстояние $d(e_i, e_j)$.

Рисунок 3.14 - Схема слияния спектральных признаков с признаками, извлеченными глубокими нейросетями.

В правой ветви в качестве спектральных признаков используются непосредственно пиксели гиперспектральных изображений. Как и ранее, расстояние $\delta(s_i, s_j)$ в спектральном пространстве между двумя пикселями i и j со спектрами s_i и s_j мы будем рассчитывать тремя способами: как евклидово расстояние (ED), спектральный угол (SAM) и дивергенцию Хеллингера (HD).

Объединенное расстояние в формируемом Евклидовом пространстве Y между пикселями i и j будем вычислять по формуле:

$$d_{ij} = \left((1-\alpha)^2 d^2(e_i, e_j) + \alpha^2 \delta^2(s_i, s_j)\right)^{1/2},$$

где α - коэффициент, определяющий влияние глубоких или спектральных признаков. $0 \le \alpha \le 1$.

Как говорилось ранее, в настоящем разделе для построения вложений мы будем использовать полностью обученную нейронную сеть, предназначенную для анализа цветных (мультиспектральных) изображений дистанционного зондирования. В качестве такой сети была выбрана предобученная сеть ResNet18, входящая в состав пакета TorchGeo [276].

При классификации с использованием нейросети, для каждого анализируемого пикселя изображения на входное изображение накладывается прямоугольное окно с

центром в анализируемом пикселе, и содержимое окна подается на вход нейросети, принимающей решение о принадлежности к тому или иному классу подстилающей поверхности. При этом из-за разницы в размере окна и входной размерности нейросети может потребоваться интерполяция. Опираясь на рекомендации из работы [53], будем использовать билинейную интерполяцию.

В качестве модели использовалась ResNet18 SeCo, обученная с использованием метода Seasonal Contrast (SeCo) [164]. При предварительных исследованиях эта модель демонстрировала более высокую точность по сравнению с моделью ResNet18 MoCo, обученной с использованием метода Momentum contrast [103], также входящей в пакет TorchGeo [276].

3.4.1 Эксперименты

Основной целью экспериментов была проверка возможности использования нейросетей, обученных с использованием цветных изображений, для анализа гиперспектральных данных с использованием предложенного подхода к слиянию со спектральными признаками.

Для оценки формируемых с использованием предложенного подхода признаков, как и ранее, будем оценивать качество решения задачи попиксельной мультиклассовой классификации с использованием формируемых признаков. Для оценки качества, будем по–прежнему использовать общую точность классификации (*Acc*), а в качестве классификатора - классификатор по ближайшему соседу (NN). Таким образом, исследуемыми параметрами будут: размер окна нейронной сети w, параметр α и размерность формируемого пространства m.

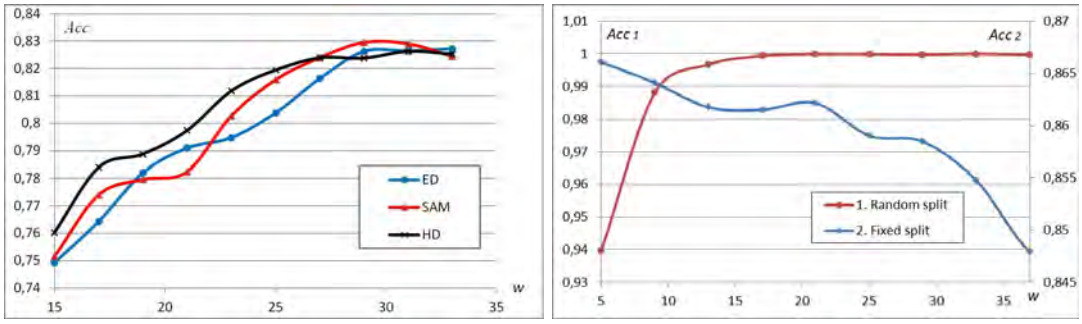
3.4.1.1 Выбор размера окна и размерности формируемого пространства

Эксперименты были разделены на несколько этапов [204*, 210*].

На первом этапе исследовалось, как размер пространственного окна при извлечении «глубоких» признаков (то есть признаков, рассчитанных с использованием глубокой нейросети) влияет на качество классификации. Мы варьировали линейный размер окна w от 15 до 33 пикселей и измеряли качество классификации. Для каждого размера окна мы извлекали «глубокие» признаки и объединяли их со спектральными признаками по предложенной схеме. Для каждой из

функций расстояния (ED, SAM, HD) были использованы значения коэффициента слияния α, близкие к оптимальным. Выходная размерность для формируемого пространства была выбрана m=20. На рисунке 3.15(а) показаны результаты эксперимента для изображения Indian Pines.

Как видно из рисунка, качество классификации ожидаемо растет по мере роста размера окна для всех функций расстояния. Значения w=29..31 представляются оптимальными в выбранном диапазоне. Подобные графики были получены для изображений Pavia University и Kennedy Space Center.



(а) (б)

Рисунок 3.15 – Зависимость точности классификации *Acc* от размера пространственного окна *w* при извлечении «глубоких» признаков для изображений Indian Pines (а) и для сцены Salinas (б).

Интересно отметить, что для изображения Salinas при случайном разбиении выборки, качество классификации росло с размером окна (рисунок 3.15(б), красная линия “1”). При фиксированном разбиении, напротив, рост размера окна приводил к снижению точности классификации (рисунок 3.15(б), синяя линия «2»). По этой причине для изображения Salinas далее использовался размер окна 5.

3.4.1.2 Зависимость результатов от способа разбиения обучающей и тестовой выборки

Интересна оказалась зависимость точности классификации с использованием глубоких признаков (при α=0) от способа разбиения обучающей и тестовой выборки. В таблице 3.5 представлены результаты, полученные для фиксированного разбиения и случайного разбиения, сделанного в том же соотношении 50:50 при размерности выходного пространства m=30.

Как видно из приведенной таблицы, при случайном разбиении использование только нейросетевых признаков позволяет добиться практически безошибочной классификации. При сравнении с фиксированным разбиением, использованным в настоящем подразделе, такой результат представляется сильно переоцененным. Хотя нейронная сеть является полностью предобученной и не знакома с обучающей выборкой (обучения нейронной сети в рамках экспериментов настоящего подраздела не осуществлялось), генерируемые ею признаки для пикселей одного класса оказываются близки друг другу по причине практически совпадающего пространственного контекста.

Использование фиксированного разбиения снижает возможность совпадения пространственного контекста, что отражается на точности классификации и делает результаты более адекватными. Обратим внимание на результаты для изображения Kennedy Space Center. Здесь даже при использовании фиксированного разбиения, точность классификации только глубокими признаками достаточно высока. Можно предположить, что причина этого заключается в достаточно мелких областях с истинной классификацией (рисунок 3.2(в)). Нейросетевые признаки с достаточным размером окна для таких областей хорошо описывают пространственный контекст и показывают высокие результаты.

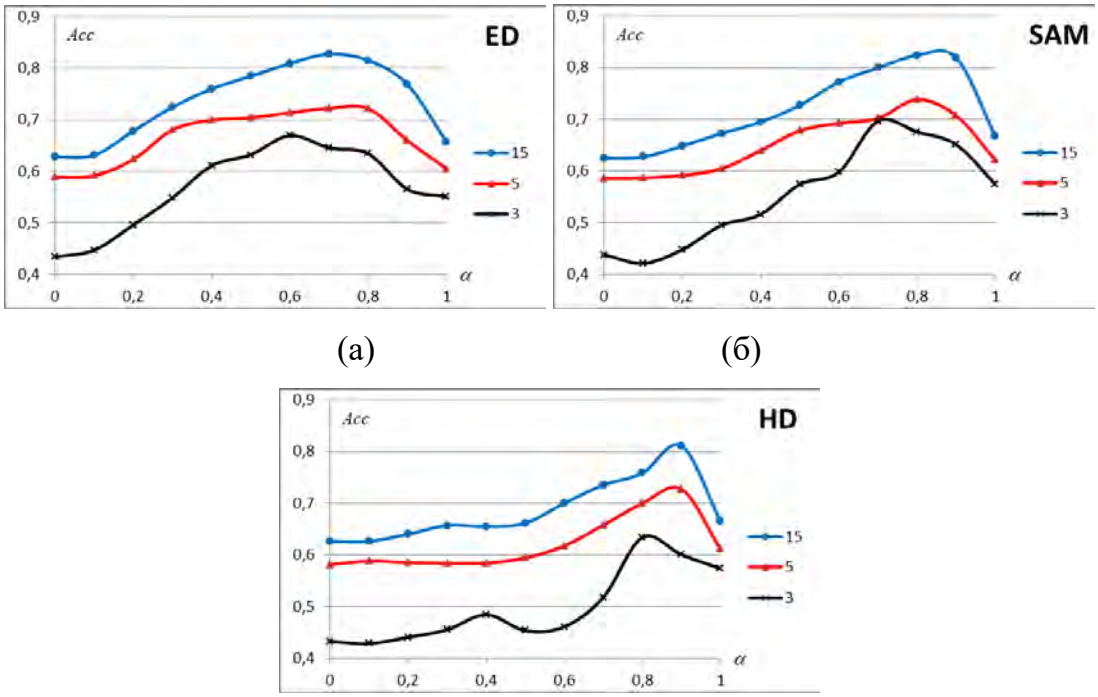
Таблица 3.5 - Сравнение разбиений на обучающую и тестовую выборки для признаков, извлеченных глубокой нейросетью.

Изображение	Фиксированное разбиение	Случайное разбиение
Indian pines	0,635	0,994
Salinas	0,792	0,999
Pavia University	0,766	0,999
Kennedy Space Center	0,965	1,00

3.4.1.3 Выбор коэффициента слияния

На втором этапе экспериментов, исследовалось влияние коэффициента слияния α глубоких и спектральных признаков на качество классификации. Здесь для четырех

выбранных изображений коэффициент слияния варьировался α от 0 до 1 с шагом 0.1, измерялось качество классификации при различных размерностях m формируемого пространства. Результаты экспериментального исследования представлены на рисунках 3.16-3.19.



(а) (б)

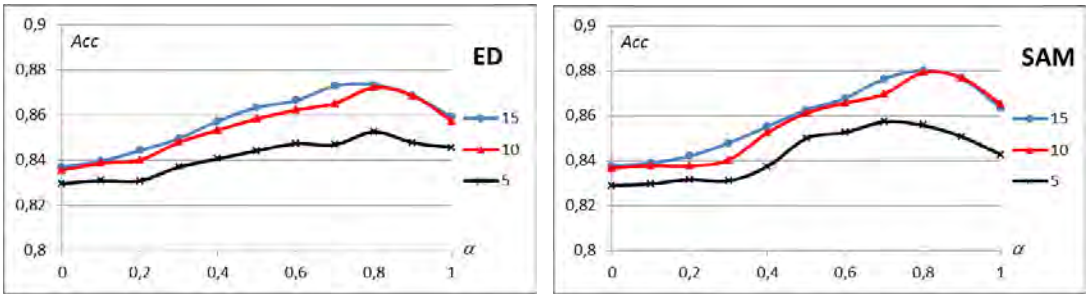
(в)

Рисунок 3.16 – Зависимость точности классификации *Acc* от коэффициента слияния α для изображения Indian Pines (фиксированное разбиение).

Каждый рисунок соответствует отдельному гиперспектральному изображению, а отдельные его части – различным функциям расстояния в спектральном пространстве (ED, SAM или HD). Каждая часть рисунка содержит три кривых, соответствующих трем различным размерностям формируемого пространства. Например, на рисунке 3.16 размерность m принимает значения 3, 5 и 15. По горизонтальной оси на каждом рисунке отложен коэффициент слияния α, а по вертикальной – точность классификации *Acc*. Рассмотрим результаты подробнее.

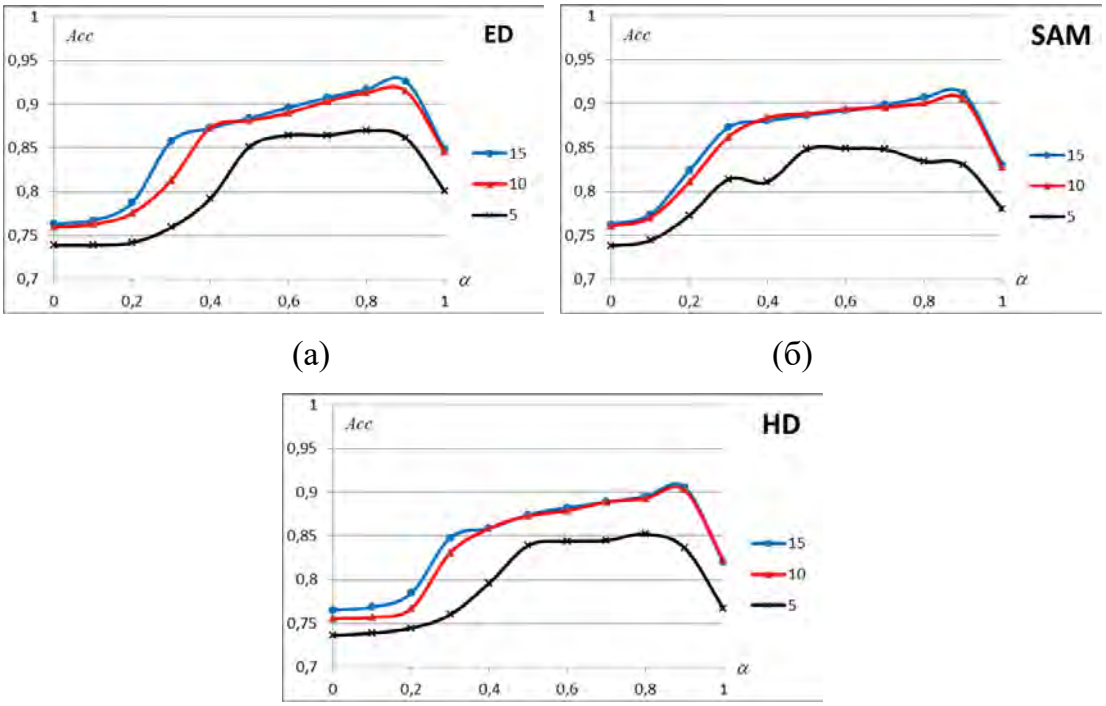
Для изображения Indian Pines качество классификации с использованием чисто спектральных признаков для каждой из размерностей оказывается несколько выше, чем для чисто глубоких признаков. Например, для размерности m=30 (не показана на рисунке) качество классификации с использованием глубоких признаков составило 63,5% против 64,7%, 65,8% и 65,8% для ED, SAM и HD, соответственно.

Слияние по предложенной схеме привело к ожидаемому существенному повышению качества классификации. Наилучшие достигнутые значения для размерности m=30 составили 82,8%, 83,5%, 83,2% для ED, SAM и HD, соответственно, что на 17,4-18,1% выше результатов, полученных при использовании только спектральных признаков.



(а) (б)

Рисунок 3.17 – Зависимость точности классификации *Acc* от коэффициента слияния α для изображения Salinas (фиксированное разбиение).



(а) (б)

(в)

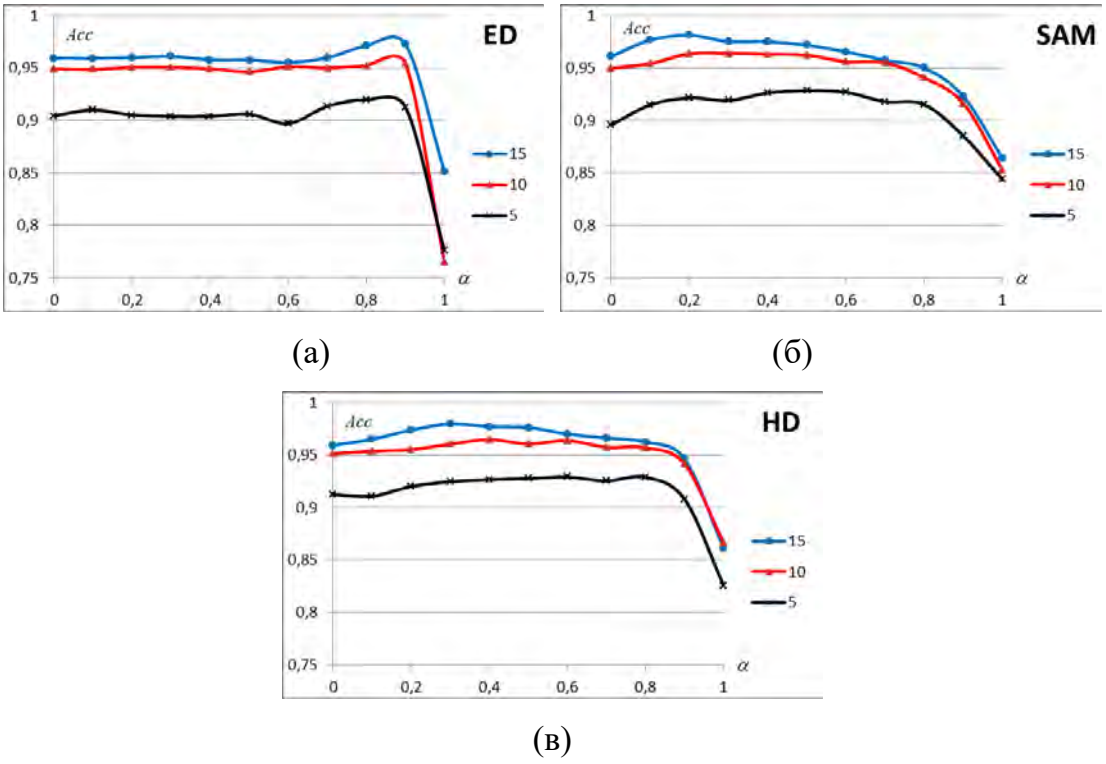
Рисунок 3.18 – Зависимость точности классификации *Acc* от коэффициента слияния α для изображения Pavia University (фиксированное разбиение).

Оптимальные значения коэффициента слияния α_{opt} оказались разными для разных функций расстояния в спектральном пространстве. Более детальное

исследование с шагом α в 0.01 для m=10..30 позволило выделить следующие диапазоны: для ED $\alpha_{opt}\in$[0.77, 0.79], для SAM $\alpha_{opt}\in$[0.8, 0.83], для HD $\alpha_{opt}\in$[0.92, 0.94].

Результаты для Pavia University, приведенные на рисунке 3.18, в целом подтверждают сказанное выше.

Несколько отличен вид графиков для сцены Kennedy Space Center (см. рисунок 3.19). Здесь, во-первых, качество классификации с использованием глубоких признаков (при α=0) оказывается существенно выше, чем при использовании чисто спектральных признаков (при α=1).



(а) (б)

(в)

Рисунок 3.19 – Зависимость точности классификации *Acc* от коэффициента слияния α для изображения Kennedy Space Center (фиксированное разбиение).

Так при m=30 (на рисунке не приводится) качество классификации с использованием глубоких признаков составило 96,5% против 86,6%, 87,9% и 88,3% для ED, SAM и HD, соответственно. Наилучшие значения для размерности m=30, достигнутые при слиянии глубоких и спектральных признаков, составили 98,0%, 98,2%, 98,2% для ED, SAM и HD, соответственно, что на 1.5-1,7% выше результатов, полученных при использовании только глубоких признаков.

Оптимальные значения коэффициента слияния α_{opt} существенно различались для разных мер спектральных различий. Более детальное исследование с шагом α в 0.01 для m=10..30 позволило выделить следующие диапазоны: для ED $\alpha_{opt}\in$[0.92, 0.96], для SAM $\alpha_{opt}\in$[0.18, 0.2], для HD $\alpha_{opt}\in$[0.28, 0.36].

Результаты экспериментов, позволяющие сравнить качество классификации для «глубоких», спектральных и сформированных в результате слияния признаков при размерности m=30, приведены в таблице 3.6 для всех четырех рассмотренных изображений. Из приведенных результатов видно, что для всех рассмотренных изображений применение разработанного метода позволило получить прирост качества классификации как относительно спектральных, так и относительно «глубоких» признаков. Наибольший прирост (до 20%) был получен на изображении Indian Pines, качество классификации на котором до слияния не превышало 65,8.

Таблица 3.6 - Точность классификации ГСИ по сформированным признакам, полученным в результате слияния «глубоких» и спектральных признаков (фиксированное разбиение).

ГСИ	Глубокие признаки	Спектральные признаки			Слияние признаков			Прирост относительно «глубоких» / спектральных признаков
		ED	SAM	HD	ED	SAM	HD	
Indian Pines	63,5%	64,7 %	65,8%	65,8%	82,8%	83,5%	83,2%	19,3-20% / 17,4-18,1%
Salinas	84,0%	85,9 %	86,4%	87,0%	87,3%	88,0%	88,2%	3,3-4,2% / 1,2-1,5
Pavia university	76,7%	84,7 %	82,8%	81,3%	92,7%	91,3%	90,8%	14,2-16,1% / 7,9-9,5%
Kennedy Space Center	96,5%	86,6 %	87,9%	88,3%	98,0%	98,2%	98,2%	1,5-1,7% / 9,9-11,4%

Выводы и результаты по главе 3

В настоящей главе предложены два метода формирования признаков ГСИ с учетом локального пространственного контекста. Первый метод основан на расширении исходного признакового пространства за счет включения пикселей из локальной окрестности и использовании функций окна в пространственной области изображения, второй – на использовании порядковых статистик.

Проведены экспериментальные исследования, показавшие важность процедуры разбиения на тестовую и обучающую выборки для методов, учитывающих пространственный контекст, так как случайное разбиение может давать переоцененные результаты. Было показано, что при использовании оконных функций для учета пространственного контекста удалось получить прирост качества классификации на 4,5-16,1% по сравнению с версией без использования пространственного контекста при использовании случайного разбиения на обучающие и тестовые выборки, в то время как при использовании фиксированного разбиения прирост качества оказался менее значительным и стабильным – до 2.2%.

Использование метода на основе порядковых статистик обеспечило не только прирост качества по сравнению с базовой версией без использования пространственного контекста (на 3,3-9,1%), но и преимущество по сравнению с методом на основе оконных функций при использовании фиксированного разбиения (на 1,2% – 6,3%). Кроме того, метод на основе порядковых статистик позволил использовать неевклидовы функции расстояния. В рамках раздела были рассмотрены две такие функции – спектральный угол и дивергенция Хеллингера. При использовании спектральных узлов прирост качества составил 6,1%-12,5% для случайного разбиения и 3.2%-7.1% для фиксированного разбиения. При использовании дивергенции Хеллингера - 5,8-13,0% и 3,2-7,3% для случайного и фиксированного разбиений, соответственно.

Предложен новый подход к объединению систем признаков. Предложенный подход основан на переходе к промежуточному представлению данных и позволяет объединять разнородные признаки в тех случаях, когда для измерения рассогласования между точками в различных признаковых пространствах используются различные функции расстояния. Рассмотрены частные случаи, когда

для промежуточного представления используются пространства с евклидовой метрикой и l_p метрикой.

Рассмотрено применение предложенного подхода к задаче попиксельной классификации гиперспектральных изображений. Для измерения расстояний в спектральном пространстве использовались спектральный угол и дивергенция Хеллингера, а в качестве текстурных признаков использовались признаки, основанные на матрице совместной встречаемости. Результаты исследования показали, что предложенный метод позволяет формировать пространства сравнительно небольшой размерности, качество классификации в которых значительно превосходит качество классификации для базовых систем признаков.

В рамках настоящей главы было предложено объединять глубокие признаки, рассчитанные с использованием предобученных на цветных изображениях сверточных нейронных сетей со спектральными признаками гиперспектральных изображений. Такая схема не требует какого-либо обучения, кроме подбора размера пространственного окна, размерности формируемого пространства и коэффициента слияния. Используя слияние по предложенной схеме, был получен прирост качества классификации, составляющий в зависимости от тестового ГСИ и используемой функции расстояния 1,2-20,0%.

4 Алгоритмическое и аппаратное ускорение методов формирования информативных признаков

К сожалению, возможность практического применения методов, описанных в предыдущих главах, ограничивает высокая вычислительная сложность, обусловленная структурой лежащей в их основе оптимизационной процедуры.

При использовании рассмотренных в главе 2 алгоритмов, основанных на процедуре градиентного спуска, рекуррентное соотношение для координат в выходном пространстве низкой размерности можно записать в обобщенном виде:

$$y_i(t+1) = y_i(t) + 2\alpha\mu\sum_{j=1}^{N}\varsigma(x_i, x_j, y_i(t), y_j(t)) \qquad (4.1)$$

где ς – функция расчета частичных приращений координат точки i в выходном пространстве при взаимодействии с точкой j, вид которой определяется конкретным функционалом ошибки.

Структура базового алгоритма оптимизации в этом случае может быть представлена так, как показано на рисунке 4.1.

В приведенной схеме на каждой итерации внешнего цикла, выполняемого до достижения некоторого критерия остановки алгоритма, и для каждой точки данных $i = 1..N$ рассчитывается влияние всех остальных точек $j = 1..N$ ($i \neq j$). Таким образом, вычислительная сложность алгоритма может быть оценена как $O(IN^2U)$, где I – число итераций, N – число точек, а сложность U вычисления приращений в общем случае зависит от выбранных функций расстояния и приведена в таблице 4.1.

В указанной таблице M - размерность входного пространства, m - размерность формируемого выходного пространства, K – размер пространственного контекста, U_φ и U_ψ - сложность вычисления функций расстояния φ и ψ в пространствах R^{M_1} и R^{M_2}, соответственно. Для рассмотренных в работе функций расстояния $O(U_\varphi+U_\psi+m) = O(M+m)$, где $M = M_1+M_2$.

Как видно из приведенной таблицы, во всех случаях можно принять оценку общей вычислительной сложности $O(IN^2(KM+m))$, где $K = 1$, если пространственный контекст не учитывается. В дальнейшем при оценке вычислительной сложности будем считать размерности входного и выходного пространств постоянными и опускать соответствующие величины K, M и m, если явно не будет указано иное.

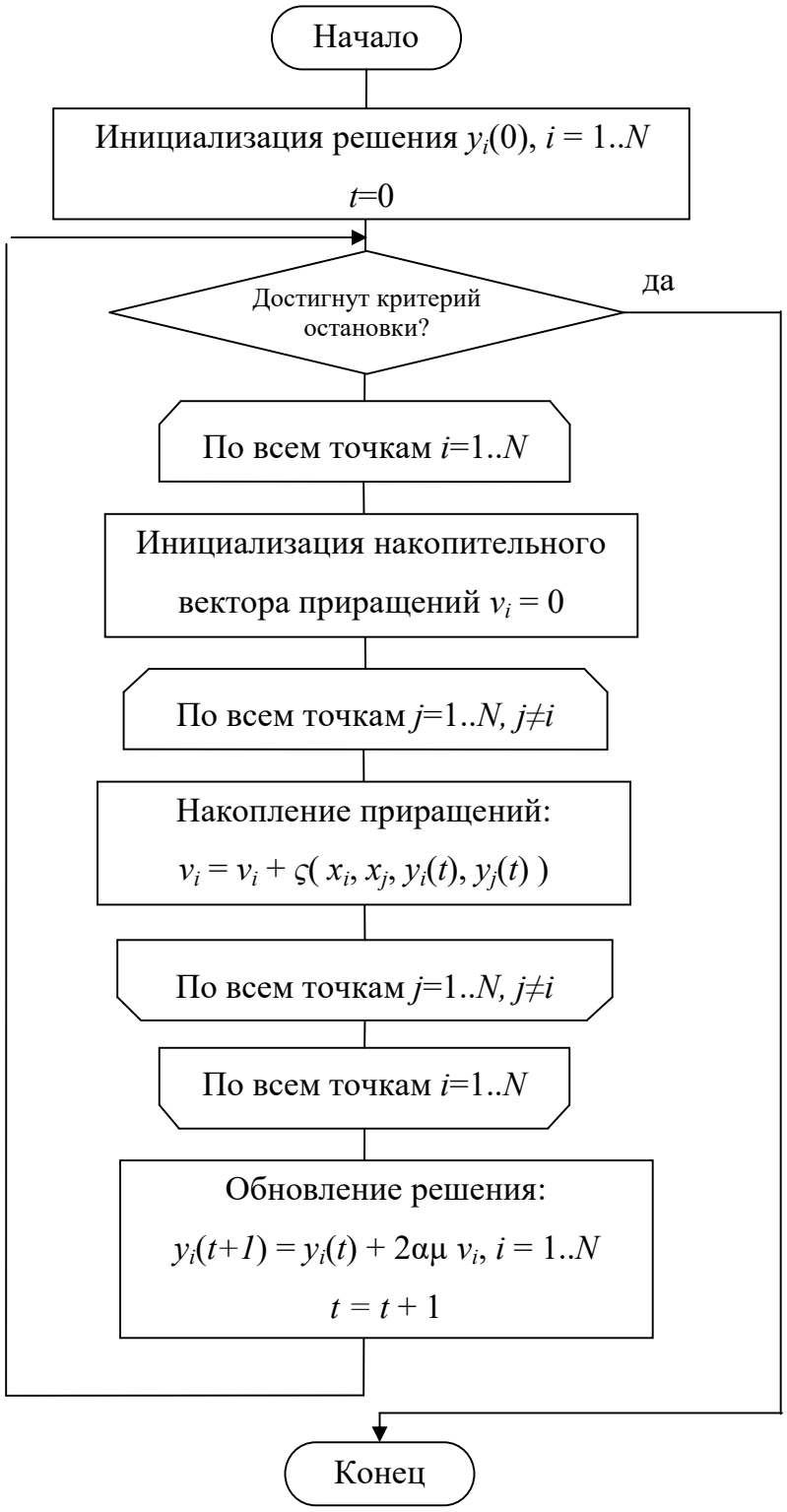


Рисунок 4.1 - Схема базового алгоритма оптимизации.

Таблица 4.1 - Оценка вычислительной сложности расчета приращений

Функция расстояния	**Функция приращения ς**	**Оценка** U
1. Евклидово расстояние $d(x_i,x_j)=\sqrt{\sum_{k=1}^{M}(x_{ik}-x_{jk})^2}$	$\rho_{ij}\cdot\frac{d(x_i,x_j)-d(y_i,y_j)}{d(y_i,y_j)}\cdot(y_i-y_j)$	O(M+m)
2. Спектральный угол $\theta(x_i,x_j)=\arccos\left(\frac{x_i^T x_j}{\|x_i\|\|x_j\|}\right)$	$\rho_{ij}\cdot(\theta(x_i,x_j)-\theta(y_i,y_j))\cdot\frac{y_{jk}-y_{ik}\langle y_i,y_j\rangle}{\sqrt{1-\langle y_i,y_j\rangle^2}}$	O(M+m)
3. Дивергенция спектральной информации $SID(x_i,x_j)=D(x_i\|x_j)+D(x_j\|x_i)$ $D(x_i\|x_j)=\sum_{k=1}^{M}p_k(x_i)\log(p_k(x_i)/p_k(x_j)$ $p_k(x_i)=\frac{x_{ik}}{\sum_{l=1}^{M}x_{il}}$	$2\rho_{ij}\frac{(SID(x_i,x_j)-SID(y_i^{[t]},y_j^{[t]}))}{\sum_{l=1}^{m}y_{il}^{[t]}}\left(1-\frac{p_k(y_j^{[t]})}{p_k(y_i^{[t]})}+\log\frac{p_k(y_i^{[t]})}{p_k(y_j^{[t]})}-D(y_i^{[t]}\|y_j^{[t]})\right)$	O(M+m)
4. Спектральная корреляция $\frac{\sum_{k=1}^{K}(x_{ik}-\bar{x_i})(x_{jk}-\bar{x_j})}{\sqrt{\sum_{k=1}^{K}(x_{ik}-\bar{x_i})^2}\sqrt{\sum_{k=1}^{K}(x_{jk}-\bar{x_j})^2}}$	$\rho_{ij}\frac{r(x_i,x_j)-r(y_i,y_j)}{g_{ij}}\left(my_{jk}-\sum_{l=1}^{m}y_{jl}-\left(my_{ik}-\sum_{l=1}^{m}y_{il}\right)f_{ij}\right)$ $f_{ij}=\frac{m\sum_{k=1}^{m}y_{ik}y_{jk}-\sum_{k=1}^{m}y_{ik}\sum_{k=1}^{m}y_{jk}}{m\sum_{k=1}^{m}y_{ik}^2-\left(\sum_{k=1}^{m}y_{ik}\right)^2}$ $g_{ij}=\sqrt{m\sum_{k=1}^{m}y_{ik}^2-\left(\sum_{k=1}^{m}y_{ik}\right)^2}\sqrt{m\sum_{k=1}^{m}y_{jk}^2-\left(\sum_{k=1}^{m}y_{jk}\right)^2}$	O(M+m)
5. Дивергенция Хеллингера $HD(x_i,x_j)=$ $\sqrt{1-\sum_{k}^{M}\sqrt{p_k(x_i)p_k(x_j)}}$	$\rho_{ij}\frac{HD(x_i,x_j)-HD(y_i,y_j)}{2HD(y_i,y_j)\sum_{l=1}^{m}y_{il}}\left(\sum_{l=1}^{m}\sqrt{p_l(y_i)p_l(y_j)}-\sqrt{\frac{p_k(y_j)}{p_k(y_i)}}\right)$	O(M+m)
6. Угол Бхаттачария $BCa(x_i,x_j)=$ $\arccos\left(\sum_{k}^{M}\sqrt{p_k(x_i)p_k(x_j)}\right)$	$\rho_{ij}\frac{BCa(x_i,x_j)-BCa(y_i,y_j)}{\sqrt{1-\left(\sum_{k=1}^{M}\sqrt{p_k(y_i)p_k(y_j)}\right)^2}\sum_{l=1}^{M}y_{il}}\left(\sum_{k=1}^{M}\sqrt{p_k(y_i)p_k(y_j)}-\sqrt{\frac{p_n(y_j)}{p_n(y_i)}}\right)$	O(M+m)

Функция расстояния	**Функция приращения ς**	**Оценка** U
7. Учет пространственного контекста $\delta(x_i, x_j) = \left(\sum_{k=1}^{K} w_k \sum_{l=1}^{M} \left(x_{i_k l} - x_{j_k l}\right)^2\right)^{1/2}$	$\rho_{ij} \frac{\delta(x_i, x_j) - d(y_i, y_j)}{d(y_i, y_j)} \cdot \left(y_i - y_j\right)$	O(KM+ +m)
8. Слияние признаков $d(z_i, z_j) = \left(\lambda_X^p \varphi^p(x_i, x_j) + + \lambda_Y^p \psi^p(y_i, y_j)\right)^{1/p}$	$\rho_{ij} \cdot \frac{d(z_i, z_j) - d(y_i, y_j)}{d(y_i, y_j)} \cdot \left(y_i - y_j\right)$	O(U_φ+U_ψ+ m)

Поскольку процедура оптимизации является итеративной, а общее число итераций зависит от многих факторов, включая критерий остановки, размер шага α (или адаптивную схему его изменения) и т.д., имеет смысл оценивать сложность вычислений (или время выполнения) на одну итерацию процедуры оптимизации: O(N^2).

Поскольку расстояния $d(x_i, x_j)$ в многомерном пространстве остаются постоянными в процессе оптимизации, представляется рациональным рассчитать их заранее. Но для хранения предварительно рассчитанных расстояний требуется дополнительная память объемом $O(N^2)$, а для вычисления такой матрицы расстояний требуются $O(N^2)$ операций. К сожалению, для больших наборов данных это практически невозможно. Например, для хранения половины (симметричной) матрицы расстояний для изображения, содержащего 256 x 256 пикселей, требуется приблизительно 16 ГБ оперативной памяти (при использовании 8-байтовых вещественных чисел с двойной точностью).

По указанным причинам алгоритмы формирования признаков, разработанные в разделах 2 и 3, не могут быть непосредственно использованы для обработки больших изображений. Чтобы преодолеть эти недостатки, в этой главе описываются предложенные автором подходы для ускорения расчетов.

4.1 Существующие методы ускорения

Ближайшим аналогом среди методов снижения размерности является нелинейное отображение [257]. Существующие модификации, направленные на снижение его вычислительной сложности, базируются на аппарате триангуляции [153] и линейном преобразовании [233].

Принимая во внимание, что нелинейное отображение [257] может также быть отнесено к техникам многомерного шкалирования, стоит отметить методы, основанные на стохастической оптимизации [38].

Идея иерархического разбиения пространства для снижения вычислительной сложности была заимствована из физики, где она широко используется при моделировании систем, состоящих из большого количества объектов (n-body simulations).

Разбиение пространства применялось при отображении графов (силовой укладке графов на плоскость), например, в [81] (регулярное разбиение) и [239] (иерархическое разбиение) для аппроксимации сил, действующих на вершины графа.

В области снижения размерности иерархическое разбиение входного многомерного пространства впервые использовалось в работах автора [348*, 353*]; алгоритмы, подобные алгоритму Барнеса-Хата в физике [14] применялись в работе автора [202*] и последующих работах.

Иерархическое разбиение выходного пространства малой размерности с использованием алгоритма Барнеса-Хата применялось в [296, 319] для ускорения алгоритма стохастической укладки (stochastic neighbor embedding, t-SNE). В работе [302] другой широко известный в физике быстрый метод мультиполей [93] был применен для ускорения алгоритма эластичной укладки (elastic embedding).

4.2 Стохастический градиентный спуск

Одним из самых простых и эффективных способов ускорения градиентного алгоритма оптимизации, лежащего в основе разработанных в главе 2 методов, является алгоритм стохастического градиентного спуска. Этот алгоритм широко используется в случае решения крупномасштабных задач. Простая версия этого алгоритма использует один случайный экземпляр (точку) из набора данных для аппроксимации градиента на каждой итерации:

$$y_i^{[t+1]} = y_i^{[t]} + 2\alpha\mu\varsigma(x_i, x_j, y_i^{[t]}, y_j^{[t]})$$

где j – индекс случайно выбранного экземпляра.

Широко используемая модификация этого алгоритма использует небольшие подмножества случайно выбранных точек (мини-выборок), которые обеспечивают более гладкую сходимость.

В нашем случае мы можем приблизить градиент с использованием множества r, содержащего индексы r_j

$$0 \le r_j < N, j = 1..R$$

случайно выбранных точек на каждой итерации. Таким образом, для стохастического градиентного спуска на основе мини-выборок, выражение принимает вид [207*]:

$$y_i^{[t+1]} = y_i^{[t]} + 2\alpha\mu \sum_{j=1,\, r_j \neq i}^{R} \varsigma(x_i, x_{r_j}, y_i^{[t]}, y_{r_j}^{[t]}) \tag{4.2}$$

Здесь r - случайно выбранная подвыборка, использованная для аппроксимации градиента на итерации t оптимизационного процесса, r_j - j-ый элемент этой подвыборки, R - мощность множества r.

В описанном подходе размер подвыборки определяет вычислительную сложность алгоритма на одну итерацию. Здесь вычислительная сложность сводится к $O(NR)$, и обычно в этом алгоритме $R << N$.

Описанный алгоритм дает возможность обрабатывать небольшие фрагменты гиперспектральных изображений и дает неплохие результаты в наших экспериментах. В работе автора [207*] рассматривалось несколько дальнейших модификаций алгоритма стохастического градиентного спуска, а именно: метод моментов и алгоритм адаптивного градиента.

4.2.1 Метод моментов

Метод моментов [253] является дальнейшим улучшением метода стохастического градиентного спуска. Главная идея метода основывается на сохранении предыдущего значения градиента. Оно позволяет до некоторой степени сохранить принятое ранее направление поиска решения.

$$Y^{[t+1]} = Y^{[t]} + \beta\Delta Y^{[t-1]} - \alpha\nabla\varepsilon^{[t]} \tag{4.3}$$

Здесь β определяет вклад предыдущего шага в новое значение, $\Delta Y^{[t-1]}$ - вектор приращений параметров, полученных на предыдущем шаге.

4.2.2 Метод адаптивного градиента

Метод адаптивного градиента ADAGRAD [71] включает в себя изменение параметров в соответствии с выражением:

$$Y^{[t+1]} = Y^{[t]} - \alpha\left(\sum_{\tau=1}^{t} diag^2\left(\nabla\varepsilon^{[\tau]}\right)\right)^{-\frac{1}{2}}\nabla\varepsilon^{[t]} \tag{4.4}$$

Здесь $diag^2\left(\nabla\varepsilon^{[\tau]}\right)$ - диагональная матрица, построенная из квадратов элементов градиента на шаге τ. В этом алгоритме вектор градиента $\nabla\varepsilon^{[t]}$ на шаге t масштабируется по каждой k-й координате с коэффициентом, обратным длине вектора, составленного из t элементов, представляющих собой k-е компоненты векторов градиента прошедших t итераций.

4.2.3 Результаты экспериментов

Для оценки алгоритмов были взяты несколько гиперспектральных изображений [120]: Cuprite, Indian pines и др. (см. п. 2.4.1). Для того чтобы эксперименты могли быть выполнены за разумное время, из исходных изображений формировались подвыборки, содержащие от 5000 до 10000 пикселей. Затем в ходе предварительных экспериментов оценивались параметры стохастических алгоритмов. При сравнении рассматривалась задача построения двух- и трехмерных признаковых пространств. Выбор такой задачи обусловлен необходимостью как выполнять визуальный анализ признаковых пространств, так и формировать псевдоцветовые представления гиперспектральных изображений в реальном масштабе времени (более подробно - см. подраздел 5.7).

Во всех случаях инициализация в пространстве малой размерности выполнялась проецированием точек данных из гиперспектрального пространства на первые две (или три) главные компоненты. На рисунке 4.2 показаны результаты для различных методов и различных значений размера шага градиентного спуска.

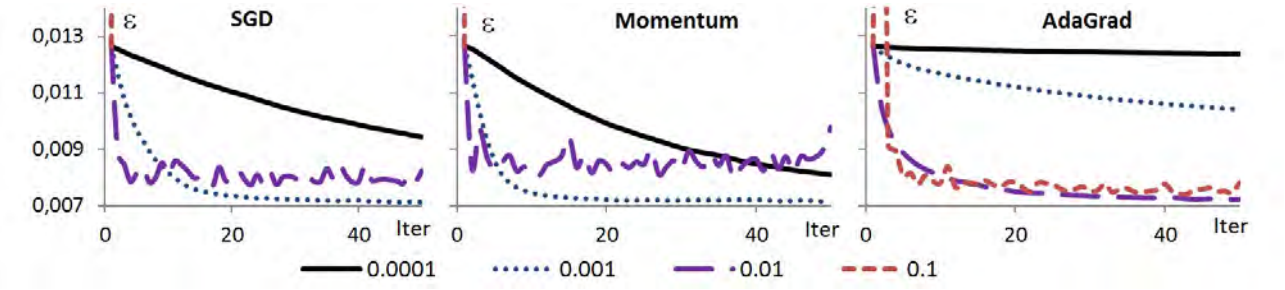


Рисунок 4.2 - Зависимость ошибки представления данных ε от числа итераций *Iter* для различных методов стохастического градиентного спуска (SGD, Momentum, AdaGrad) и различных значений параметра α (0,0001; 0,001; 0,01; 0,1).

На рисунке 4.3 показаны результаты для различных методов стохастического градиентного спуска для пары тестовых изображений Cuprite и Indian Pines, а также двух- и трехмерных (2D и 3D) выходных пространств. Также приведено время выполнения одной итерации для процессора Intel Core i7 6500U с тактовой частотой 2.5GHz.

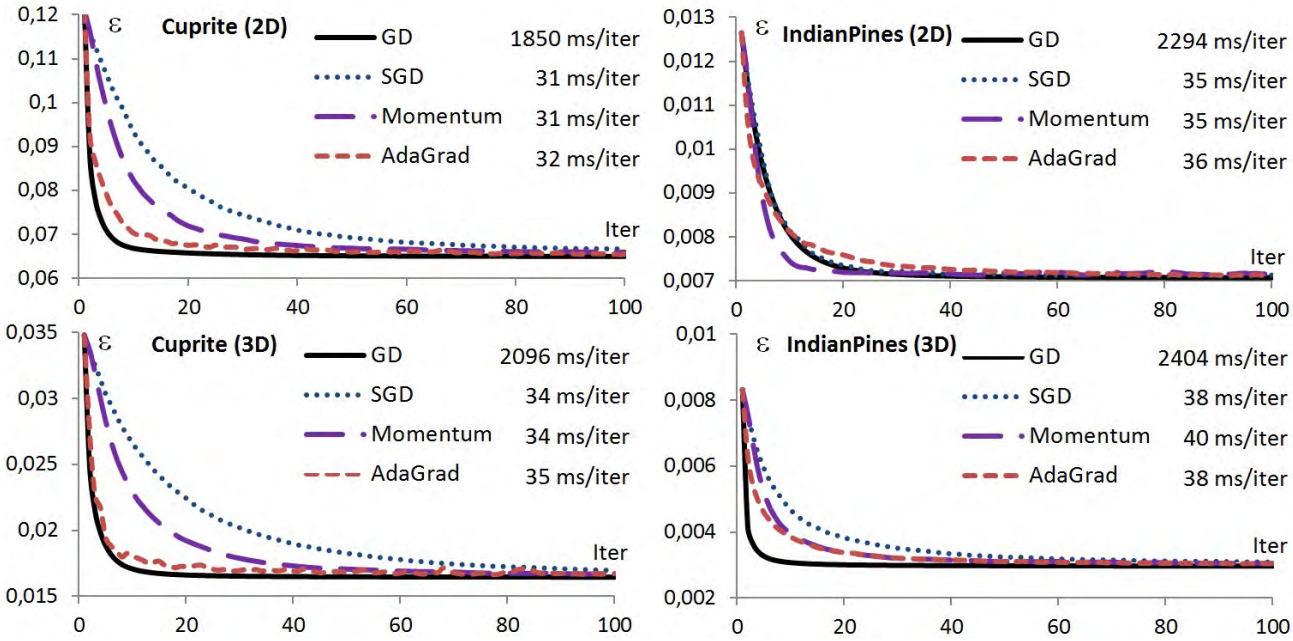


Рисунок 4.3 - Сравнение методов градиентного спуска. Зависимость ошибки представления данных ε от числа итераций *Iter* для различных методов оптимизации.

Как можно видеть, все описанные выше методы обеспечивают удовлетворительное решение задачи. Во всех случаях ошибка снижается до достаточно малых величин. В большинстве случаев метод адаптивного градиента (AdaGrad) демонстрирует лучшие результаты среди методов стохастического градиентного спуска, метод моментов (Momentum) показывает устойчивые

результаты во всех случаях. Следует отметить, что базовый алгоритм градиентного спуска (GD) для вычисления градиента на каждой итерации использует все множество точек, и имеет длительное время выполнения (на итерацию). В то же время другие алгоритмы используют только небольшие подмножества (minibatches), состоящие из пятидесяти точек. Можно отметить, что приемлемое качество решения задачи с использованием любого из рассмотренных стохастических методов может быть получено за время, сравнимое со временем выполнения всего лишь одной итерации базового метода.

4.2.3.1 Использование производных второго порядка

В литературе предложено множество модификаций метода градиентного спуска. Наиболее хорошо известный алгоритм – метод Ньютона [357], учитывающий производные второго порядка. В этом случае, обновление координат выполняется в соответствии со следующим выражением

$$Y^{[t+1]} = Y^{[t]} - H^{-1}\nabla\varepsilon \tag{4.5}$$

Здесь H - матрица Гессе на шаге t, $H = \nabla^2\varepsilon$.

К сожалению, метод Ньютона в представленном виде неприменим к минимизации функций, содержащих большое количество переменных. По этой причине для решения масштабных задач используются методы, учитывающие только производные первого порядка или использующие информацию об аппроксимации производных второго порядка.

4.3 Разбиение пространства

Другая идея для ускорения расчетов в контексте рассматриваемой задачи была заимствована из физики. Согласно этому подходу, мы выполняем разбиение пространства на ячейки и при вычислении приращений рассматриваем взаимодействия «точка-ячейка», а не взаимодействия «точка-точка», когда это возможно [14]. Эта идея ускорения широко используется в моделировании для решения задачи n тел (см., например, [310, 83, 111]). В нашем случае идея аппроксимировать взаимодействия «точка-точка» взаимодействиями «точка-ячейка» позволяет значительно ускорить процесс итеративной оптимизации.

Для ускорения вычислений с использованием методов разбиения пространства всё множество точек (векторов) X разбивается на некоторые подмножества c_1, c_2,…, c_C так, чтобы все точки $x_i \in c_j$, $i=1..|c_j|$ из некоторого подмножества c_j обладали схожими характеристиками. То есть, чтобы все точки из некоторого подмножества c_j находились близко друг к другу в признаковом пространстве. В этом случае при определенных условиях точки подмножества c_j могут рассматриваться не по отдельности, а в совокупности, как единый объект.

Дальнейшие рассуждения будем проводить для случая входного и выходного пространств с евклидовой метрикой. Предположим, что подмножество c находится «достаточно далеко» от точки x_i. В этом случае прямая сумма

$$\sum_{x_j \in c} \rho_{ij} \frac{d(x_i, x_j) - d(y_i, y_j)}{d(y_i, y_j)} \cdot (y_i - y_j) \tag{4.6}$$

может быть аппроксимирована следующим образом

$$|c| \rho_{ic} \frac{d(x_i, x_c) - d(y_i, y_c)}{d(y_i, y_c)} \cdot (y_i - y_c) \tag{4.7}$$

где $|c|$ - количество точек в рассматриваемом подмножестве c, $d(x_i, x_c)$ - расстояние от точки x_i до центра подмножества c во входном пространстве, $d(y_i, y_c)$ - расстояние от точки y_i до центра подмножества в выходном пространстве, x_c, y_c - координаты центра множества точек c во входном и выходном пространствах, соответственно, ρ_{ic} – константы, определяемые конкретным функционалом ошибки (2.12), например, $\rho_{ic}=1$ или $\rho_{ic} = 1 / d(x_i, x_c)$.

Теперь, если все точки исходного множества X разделены на C непересекающихся подмножеств $c_j \subset X$, таких что

$$\bigcup_{j=1}^{C} c_j = X,$$
$$c_i \cap c_j = \varnothing, i, j = 1..C$$

то (4.1) принимает вид

$$y_i^{[t+1]} = y_i^{[t]} + 2\alpha\mu \cdot \sum_{j=1}^{C} \tilde{\varsigma}(x_i, y_i^{[t]}, c_j), \tag{4.8}$$

где, например, для случая евклидовых расстояний

$$\tilde{\varsigma}(x_i, y_i, c_j) = \left|c_j\right| \cdot \rho_{i,c_j} \frac{d(x_i, x_{c_j}) - d(y_i, y_{c_j})}{d(y_i, y_{c_j})} \cdot \left(y_i - y_{c_j}\right). \quad (4.9)$$

Очевидно, что выражение (4.8) позволяет аппроксимировать (4.1) с некоторой точностью, которая зависит от того, насколько хорошо точки разделены на подмножества. По этой причине возникает вопрос о том, как лучше выполнять подобную декомпозицию.

4.3.1 Разбиение входного пространства. Использование опорных узлов

Базовая схема описываемого ниже метода опубликована автором в [202*]. В целом, метод состоит из следующих четырех шагов:

1. Разбиение входного пространства.
2. Построение списков опорных узлов.
3. Инициализация в выходном пространстве.
4. Итеративная оптимизация.

Приведенные этапы описаны далее более детально.

4.3.1.1 Этап 1. Разбиение входного пространства

Предположим, что с использованием некоторого метода разбиения пространства мы получили древоподобную структуру, содержащую в терминальных узлах (листьях) точки исходного множества. Корнем структуры является узел верхнего уровня c^0, а каждая вершина-узел c_i^l на уровне $l > 0$ содержит либо дочерние узлы $\left\{c_{i_1}^{l+1}, c_{i_2}^{l+1}, ..., c_{i_{\left|c_i^l\right|}}^{l+1}\right\}$, либо непосредственно точки $\left\{x_{i_1}, x_{i_2}, ..., x_{i_{\left|c_i^l\right|}}\right\}$ исходного множества X в качестве терминальных узлов.

Для построения такого дерева могут быть применены как алгоритмы иерархической агломеративной кластеризации [220], так и неиерархические алгоритмы кластеризации, запускаемые в рекурсивном порядке (например, алгоритм k-внутригрупповых средних (см. п. 5.5.1), нейросетевой алгоритм WTA (см. Приложение А.5) и т.д.), либо алгоритмы разбиения пространства. Более подробно такие алгоритмы будут рассмотрены в п. 4.3.2.

В частном случае при использовании алгоритмов, разбивающих на каждом шаге узлы иерархии на два потомка, формируемое дерево является бинарным, как показано на рисунке 4.4.

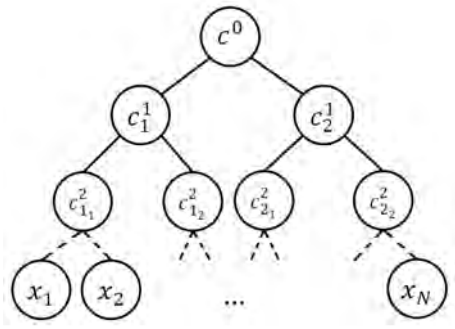


Рисунок 4.4 - Структура бинарного дерева для декомпозиции пространства

Дерево, созданное на первом шаге алгоритма, может быть непосредственно использовано в оптимизационном процессе следующим образом. Мы итеративно обновляем в пространстве малой размерности координаты точек y_i в соответствии с (4.8), используя улы дерева верхнего уровня, если такие узлы находятся «достаточно далеко» от точки x_i, и узлы дерева нижнего уровня или непосредственно точки исходного множества X в противном случае.

В предлагаемом методе построенное дерево не используется напрямую в оптимизационном процессе. Вместо этого в оптимизации используется специальная структура данных, описываемая ниже.

4.3.1.2 Этап 2. Построение списков опорных узлов

Так как разбиение происходит в исходном пространстве высокой размерности, то дерево декомпозиции остается постоянным на протяжении всего процесса оптимизации. По этой причине в работах автора [202*, 212*] было предложено предварительно вычислять и хранить множество узлов и точек, используемых в оптимизационном процессе (так называемые списки опорных узлов), с которыми будет взаимодействовать каждая из точек $x_i \in X$ исходного множества. Использование списков опорных узлов требует дополнительной памяти, но позволяет избежать повторных вычислений критерия разбиения во время оптимизационного процесса и позволяет сохранять предварительно вычисленные расстояния до опорных узлов, что позволяет сделать оптимизационный процесс независимым от размерности входного пространства.

Формально под опорным узлом точки x в настоящей работе будем понимать некоторую точку $x_i \neq x$ или множество точек $\{x_i\}$, обладающих в исходном многомерном пространстве близкими характеристиками и рассматриваемых как единое целое. Под списком опорных узлов Λ точки x будем понимать упорядоченное множество опорных узлов точки x.

При построении списка опорных узлов для каждой из точек примем во внимание тот факт, что при построении аппроксимации приращений координат точки, расположенные вблизи рассматриваемой, следует учитывать по отдельности, а множества точек, расположенных в удалении от рассматриваемой, возможно учитывать совместно.

Таким образом, если при построении аппроксимации какой-либо узел иерархии находится в удалении от точки, он может рассматриваться как единое целое. В противном случае необходимо произвести декомпозицию узла на узлы более низкого уровня. В том же случае, когда узел состоит непосредственно из точек и не содержит дочерних узлов, декомпозиция узла приводит к добавлению в список опорных узлов всех содержащихся в нем точек.

Формирование списка опорных узлов выполняется для каждой из точек исходного множества X один раз до запуска процесса оптимизации. Ниже приводится более формальное описание предлагаемого алгоритма формирования списка опорных узлов.

Пусть x- некоторая точка, для которой требуется сформировать список Λ опорных узлов, c_i^l - произвольный узел иерархии. Тогда рассмотренные выше принципы могут быть реализованы в виде следующего рекурсивного алгоритма.

1. Если узел c_i^l удовлетворяет критерию декомпозиции по отношению к рассматриваемой точке $\theta(c_i^l, x) = True$, и в c_i^l содержатся дочерние узлы $\left\{ c_{i_1}^{l+1}, c_{i_2}^{l+1}, ..., c_{i_{\left|c_i^l\right|}}^{l+1} \right\}$, то применить этот рекурсивной алгоритм для каждого из дочерних узлов $c_{i_1}^{l+1}, c_{i_2}^{l+1}, ..., c_{i_{\left|c_i^l\right|}}^{l+1}$.

2. Если узел c_i^l удовлетворяет критерию декомпозиции по отношению к рассматриваемой точке $\theta(c_i^l, x) = True$, и в c_i^l содержатся точки $\left\{x_{i_1}, x_{i_2}, ..., x_{i_{\left|c_i^l\right|}}\right\}$, то добавить в список опорных узлов Λ содержащиеся в узле точки: $\Lambda = \Lambda \cup \left\{x_{i_1}, x_{i_2}, ..., x_{i_{\left|c_i^l\right|}}\right\}$.

3. Если узел c_i^l не удовлетворяет критерию декомпозиции по отношению к рассматриваемой точке $\theta(c_i^l, x) = False$, то добавить его в список опорных узлов: $\Lambda = \Lambda \cup \{c\}$.

Рассмотренный алгоритм задан с точностью до критерия декомпозиции. В литературе были описаны различные критерии подобной декомпозиции (см., например, [255]). Автором настоящей диссертации применялись два таких критерия.

Первый критерий декомпозиции был построен на сопоставлении расстояния от рассматриваемой точки *x* до узла *c* с радиусом узла *R*(*c*). Декомпозиция узла дерева осуществлялась при выполнении следующего условия [202*]:

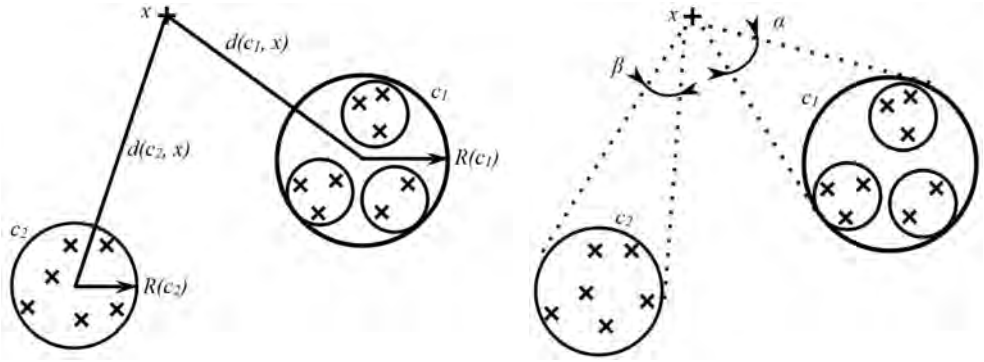
$$\theta(c, x) = \left(\frac{d(c, x)}{R(c)} \le T\right). \tag{4.10}$$

где T – предопределенный пороговый параметр. В противном случае узел *c* дерева добавляется к списку опорных узлов. Этот процесс проиллюстрирован на рисунке 4.5(а): узел c_1 будет разделен на три дочерних узла, если $d(c_1, x) / R(c_1) \le T$; узел c_2 будет рассматриваться как единое целое, если $d(c_2, x) / R(c_2) > T$. Здесь радиус узла *c* может быть оценен следующим образом: $R(c) = \max_{x_i \in c}\{d(x_c, x_i)\}$.

В качестве альтернативного критерия рассматривался критерий разбиения на основе значения угла, под которым узел *c* наблюдается из точки *x* в исходном пространстве:

$$\theta(c, x) = \left(2 \cdot \arcsin\left(\frac{R(c)}{d(c, x)}\right) \ge \delta\right). \tag{4.11}$$

Здесь с использованием радиуса $R(c)$ оценивается угол, под которым описывающая узел c гиперсфера наблюдается из точки x в исходном пространстве. На рисунке 4.5(б) узел, наблюдаемый из точки x под углом α, будет разбит на три дочерних узла, если $\alpha \geq \delta$; узел, наблюдаемый из o под углом β, будет рассматриваться как единое целое при $\beta < \delta$.



(а) (б)

Рисунок 4.5 - Иллюстрация создания списка опорных узлов: критерий разбиения на основе расстояний (а); критерий разбиения на основе углов (б).

Простой алгоритм создания списков опорных узлов [206*] является рекурсивным и описывается приведенным ниже псевдокодом.

```
Procedure CreateRefList( Point x, Node c, ReferenceNodeList Λ)
begin
        if θ(c,x)=False then
                add c to Λ
        else if c contains Points then
                for each Point p in c
                        add p to Λ
        else
                for each Node n in c
                        CreateRefList( x, n, Λ )
end
```

Процедура CreateRefList принимает на вход точку x, для которой осуществляется построение списка, узел c построенной структуры разбиения пространства и заполняет список опорных узлов Λ. При первом вызове процедуры узел c представляет собой корень дерева разбиения пространства, а список Λ пуст. В процессе работы решение о дальнейшей декомпозиции принимается на основании критерия $\theta(c, x)$. Если критерий не выполняется, то узел c считается опорным и добавляется к списку опорных узлов. В противном случае рассматриваются потомки узла c.

Примечание - как было сказано выше, при использовании на этапе разбиения входного пространства алгоритмов, разбивающих на каждом шаге узлы иерархии на два потомка, формируемое дерево является бинарным, и при последующей декомпозиции узла такого дерева в список опорных узлов добавляется ровно два узла. При использовании деревьев более высокой степени узлов может быть больше, что может приводить к значительному увеличению длины списка опорных узлов для расчета приращений. Чтобы избежать этого, в работе автора [202*] предлагалось вместо декомпозиции узла на отдельные дочерние элементы выделять из него те элементы, которые близки к рассматриваемой точке во входном пространстве, а остальные дочерние элементы узла объединять во множество, называемое далее неполным узлом, которое учитывать при расчете приращений как единое целое.

Использование неполных узлов позволяет немного ускорить вычислительный процесс за счет объединения некоторого количества дочерних элементов декомпозированных узлов дерева, что требует дополнительной памяти для хранения информации об агрегированных элементах.

Следует отметить, что количество узлов в списке опорных узлов зависит от распределения данных. Полагая, что списки опорных узлов содержат в среднем L узлов, можно оценить объем памяти, требуемый для хранения списков, как O[LN].

4.3.1.3 Этап 3. Инициализация в выходном пространстве

На третьем этапе метода осуществляется инициализация координат точек в целевом пространстве малой размерности. Такая инициализация (начальное приближение) может быть выполнена различными способами. В частности, в литературе встречаются способы инициализации случайными значениями и результатами работы метода PCA. В работе автора [348*] отмечено, что инициализация с использованием метода PCA позволяет обеспечить в среднем снижение размерности с более высоким качеством.

При использовании метода PCA для инициализации в выходном пространстве рассчитываются только m главных компонент, а затем координаты входного многомерного пространства проецируются в подпространство, образуемое этими компонентами. Этот подход позволяет сократить временные затраты по сравнению со

случайной инициализацией на следующих шагах итеративного оптимизационного процесса, так как процесс начинает выполняться с лучшими начальными условиями.

Хотя время поиска главных компонент по ковариационной матрице зависит от размерности входного пространства M и может оцениваться как $O(M^2)$ на итерацию, весь процесс может быть медленным, так как нахождение самой матрицы ковариации выполняется за время $O(NM^2)$ и зависит не только от размерности, но также от количества точек N. Для сокращения времени вычислений в работах автора [213*, 208*, 214*] используется только небольшое подмножество случайно выбранных точек исходного множества X, что делает общую вычислительную сложность независимой от количества точек. Существуют также другие решения, например, использование нейросетевого подхода на основе обобщенного обучающего правила Хэбба (Сангера) [258], использованного в работе автора [202*].

4.3.1.4 Этап 4. Итеративная оптимизация

После инициализации координат точек в формируемом пространстве малой размерности для нахождения субоптимального решения выполняется процедура оптимизации. Как говорилось выше, оптимизация строится на основе аппроксимации вычисления рекуррентного соотношения (4.1).

Наибольший интерес представляет приближенное вычисление для произвольной точки x_i суммы $\sum\varsigma(...)$ в (4.1), так как расчет множителя $2\alpha\mu$ может быть сделан заранее и не представляет затруднений. С использованием сформированного на предыдущем шаге работы метода списка опорных узлов Λ_i для точки x_i выражение может быть записано в виде

$$y_i^{[t+1]} = y_i^{[t]} + 2\alpha\mu \sum_{n\in\Lambda_i} \tilde{\varsigma}(x_i, y_i^{[t]}, n), \tag{4.12}$$

где функция $\tilde{\varsigma}(x_i, y_i(t), n)$ принимает различный вид в зависимости от того, является ли элемент списка опорных узлов n точкой или узлом дерева.

1. Если n является точкой x_j, то

$$\tilde{\varsigma}(x_i, y_i^{[t]}, n) = \varsigma(x_i, x_j, y_i^{[t]}, y_j^{[t]}), \tag{4.13}$$

или, для пространств с евклидовой метрикой

$$\tilde{\varsigma}(x_i, y_i^{[t]}, n) = \rho_{ij} \frac{d(x_i, x_j) - d(y_i^{[t]}, y_j^{[t]})}{d(y_i^{[t]}, y_j^{[t]})} \cdot \left(y_i^{[t]} - y_j^{[t]}\right). \quad (4.14)$$

2. Если *n* является узлом c_j, то

$$\tilde{\varsigma}(x_i, y_i^{[t]}, n) = \left|c_j\right| \varsigma(x_i, x_{c_j}, y_i^{[t]}, y_{c_j}^{[t]}), \quad (4.15)$$

или, для пространств с евклидовой метрикой

$$\tilde{\varsigma}(x_i, y_i^{[t]}, n) = \left|c_j\right| \rho_{i,c_j} \frac{d(x_i, x_{c_j}) - d(y_i^{[t]}, y_{c_j}^{[t]})}{d(y_i^{[t]}, y_{c_j}^{[t]})} \cdot \left(y_i^{[t]} - y_{c_j}^{[t]}\right), \quad (4.16)$$

где $d(x_i, x_{c_j})$ – расстояние от точки до центра узла в исходном пространстве, $d(y_i, y_{c_j})$ – расстояние от точки до центра узла в результирующем пространстве, y_{c_j} – координаты центра узла в результирующем пространстве.

Следует отметить, что центры узлов x_{c_j} в многомерном пространстве могут быть рассчитаны заранее. Также заранее могут быть рассчитаны расстояния $d(x_i, x_{c_j})$, что избавит от необходимости пересчета расстояний в многомерном пространстве и сделает вычислительную сложность этапа итеративной оптимизации независящей от размерности входного пространства. Центры узлов y_{c_j}, как и расстояния $d(y_i, y_{c_j})$ в результирующем пространстве, должны уточняться в процессе оптимизации.

Таким образом, вычислительная сложность этапа оптимизации может в среднем оцениваться как O(*LN*) на одну итерацию, где *L* – средняя длина списка опорных узлов:

$$L = \frac{1}{N} \sum_{i=1}^{N} \left|\Lambda_i\right|. \quad (4.17)$$

Примечание - при использовании неполных узлов (модификация, представленная в работе автора [202*]) центры неполных узлов y_{v_j} $\boldsymbol{y_{vj}}$ могут быть рассчитаны через центры соответствующих полных узлов y_{c_j} и позиции исключенных точек:

$$y_{V_j} = \frac{1}{|c_j|} \cdot \left(|c_j| \cdot y_{c_j} - \sum_{x_k \in c_j \setminus v_j} y_k \right),\ y_{C_j} = \frac{1}{|c_j|} \cdot \sum_{x_k \in c_j} y_k \ .$$

Сам расчет при этом выполняется аналогично выражениям (4.15) и (4.16).

На практике по ходу оптимизационного процесса целесообразно контролировать ошибку представления данных, рост которой может указать на расходимость процесса, а снижение ее ниже заданного порога или малая величина ее изменения может использоваться как критерий остановки. Вычислительная сложность вычисления ошибки представления данных в формируемом пространстве меньшей размерности (2.9) примерно соответствует вычислительной сложности одного шага базовой версии оптимизационного процесса (O(N^2)). С использованием списков опорных узлов ошибка представления данных в формируемом пространстве может быть оценена в среднем за O(LN) в соответствии с выражением:

$$\widetilde{\varepsilon} = \frac{1}{\mu} \sum_{i=1}^{N} \sum_{n \in \Lambda_i} \widetilde{\zeta}(x_i, y_i, n), \tag{4.18}$$

где

$$\widetilde{\zeta}(x_i, y_i, n) = \begin{cases} \rho_{ij} \left(\varphi(x_i, x_j) - \psi(y_i, y_j) \right)^2, & \text{если } n \text{ является точкой}, n \equiv x_j \\ |c_j| \rho_{i,c_j} \left(\varphi(x_i, x_{c_j}) - \psi(y_i, y_{c_j}) \right)^2, & \text{если } n \text{ является узлом}, n \equiv c_j \end{cases}, \tag{4.19}$$

а φ и ψ - функции расстояния во входном и выходном пространствах, соответственно.

Следует отметить, что выражение (4.18) оценивает ошибку (2.9) представления данных в формируемом пространстве меньшей размерности с некоторой погрешностью. В настоящей работе эта оценка использовалась для контроля оптимизационного процесса. При повышении значения оценки ошибки коэффициент градиентного спуска α понижался до тех пор, пока процесс не становился сходящимся. Также эта оценка использовалась для остановки процесса оптимизации, когда относительное уменьшение оценки ошибки за заданное число итераций не превышало предопределенного значения.

4.3.2 Выбор структуры для разбиения пространства

Использование неадаптивных равномерных сеток для разбиения многомерных пространств неэффективно вследствие быстрого роста числа ячеек такого разбиения с размерностью пространства. Так, при разбиении пространства размерностью M по каждой координате на K частей образуется K^M ячеек.

По этой же причине при работе в пространстве отсчетов гиперспектральных изображений, размерность которого исчисляется сотнями компонент, невозможно использовать многомерные обобщения квадро-, окто- и гексадекадеревьев [78, 172]. Хотя в таких структурах применяется адаптивное разбиение, каждый узел структуры при разбиении будет порождать 2^M ячеек, что делает их применение с данными указанной размерности невозможным. Поэтому хотелось бы иметь некоторую адаптивную процедуру, которая на первом этапе описанного выше метода могла бы выполнять иерархическое разбиение большого набора гиперспектральных данных и не порождала бы неприемлемо большого количества узлов.

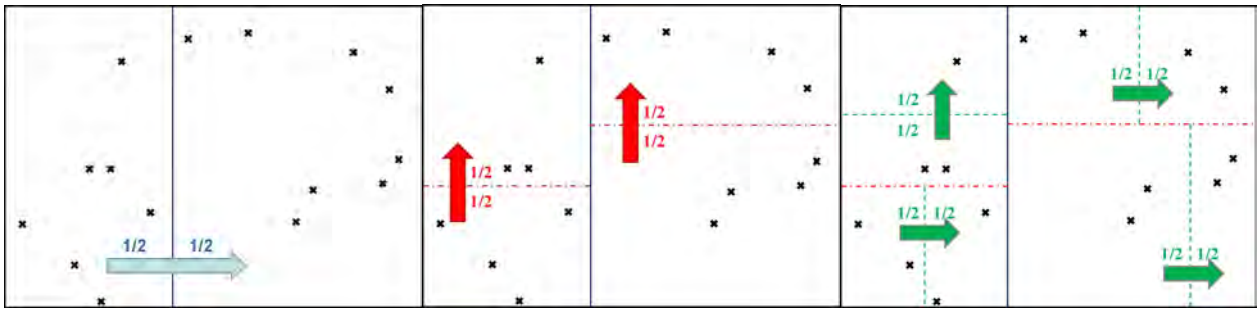
Такая процедура может быть построена на основе кластеризации, являющейся наиболее естественным способом группировки точек в многомерном пространстве по их характеристикам. Однако сложность вычислений даже в случае эффективной агломеративной кластеризации составляет $O(N^2)$. В случае дивизивных алгоритмов, основанных, например, на использовании нейросети (WTA), примененной в [202*], время работы иерархической кластеризации сильно зависит от параметров алгоритма.

В то же время существует большое количество деревьев, применяемых для декомпозиции пространства [256], которые могут быть построены за меньшее время и широко используются в мультимедийных базах данных, геоинформационных системах, информационном поиске, компьютерной графике и т.д.

В работе автора [206*] для разбиения входного пространства использовались и сравнивались несколько бинарных деревьев для декомпозиции пространства: kd-дерево [19], метрическое дерево (vp-дерево) [292, 322], и бинарное дерево, построенное на упрощении минимаксного алгоритма кластеризации [361] (дерево кластеров).

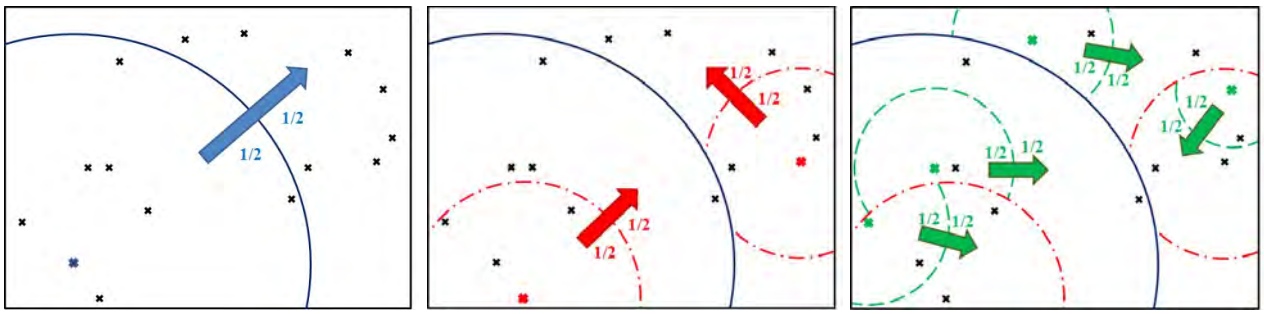
Такой выбор позволяет оценить разные по своим свойствам структуры. KD-дерево является одной из давно и хорошо известных структур, основанной на разбиении пространства гиперплоскостями, ортогональными к осям координат (см.

рисунок 4.6). VP-дерево - другая хорошо известная структура для декомпозиции, основанная на разбиении пространства гиперсферами (см. рисунок 4.7). Дерево кластеров – структура, разделяющая пространство гиперплоскостями, расположенными между парами выбранных удаленных точек (см. рисунок 4.8).



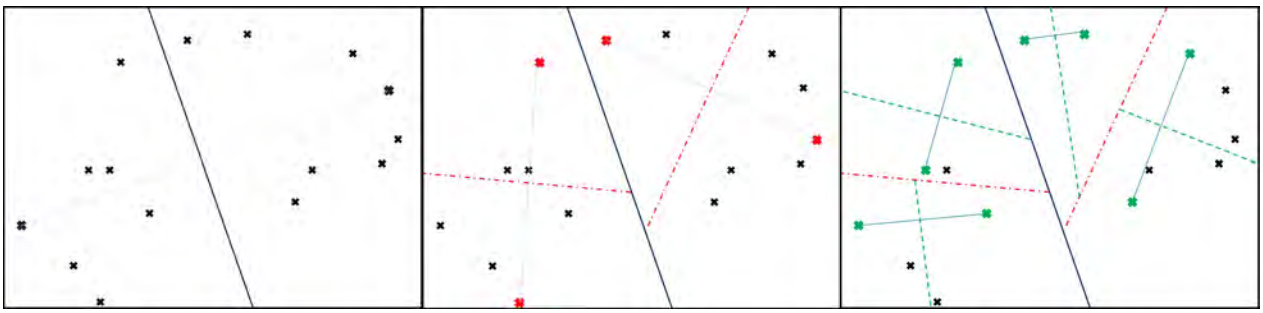
(а) (б) (в)

Рисунок 4.6 - Иллюстрация декомпозиции пространства с использованием KD-дерева. Различные уровни декомпозиции показаны различными линиями: первый уровень показан сплошной линией, второй уровень показан штрих-пунктирной линией, третий – штриховой линией.



(а) (б) (в)

Рисунок 4.7 - Иллюстрация декомпозиции пространства с использованием VP-дерева. Различные уровни декомпозиции показаны различными линиями: первый уровень показан сплошной линией, второй уровень показан штрих-пунктирной линией, третий – штриховой линией.



(а) (б) (в)

Рисунок 4.8 - Иллюстрация декомпозиции пространства с использованием дерева кластеров. Различные уровни декомпозиции показаны различными линиями: первый уровень показан сплошной линией, второй уровень показан штрих-пунктирной линией, третий – штриховой линией.

Все упомянутые структуры могут быть построены за O($N \log N$) операций (в случае сбалансированности дерева), что представляется наиболее подходящим для рассматриваемой задачи. Так как алгоритмы построения для первых двух структур хорошо известны, ниже в виде псевдокода приводится только алгоритм [206*] для последней структуры.

```
Function CreateTree( SetOfPoints c ) returns Node
begin
        create new Node n
        if Number of points in c is less than Threshold then
        begin
                n.Children = c
                return n
        end
        find mean vector value in c
        find Point p1 farthest from mean
        find Point p2 farthest from p1
        create new SetOfPoints c1
        create new SetOfPoints c2
        for each Point p in c begin
                if ( d(p1, p) < d(p2, p) )
                        add p to c1
                else
                        add p to c2
        end
        Node n1 = CreateTree( c1 )
        Node n2 = CreateTree( c2 )
        add n1 to n.Children
        add n2 to n.Children
        return n
end
```

Приведенный алгоритм принимает на вход множество точек c и возвращает корневой узел построенного дерева. На первом шаге создается пустой узел n, который впоследствии и будет возвращен в качестве корневого. Далее если мощность множества c менее заданного порога (параметр алгоритма), все точки множества c помещаются в узел n в качестве дочерних, и этот узел возвращается в качестве результата.

В противном случае по множеству c вычисляется средний вектор *mean*, ищется наиболее удаленная от среднего точка p_1, а затем самая удаленная от точки p_1 точка p_2. Затем точки множества c разделяются на два подмножества c_1 и c_2, так, что в c_1 попадают точки, расстояние от которых до точки p_1 меньше, чем до точки p_2, а в подмножество c_2 – остальные точки множества. Далее из сформированных подмножеств c_1 и c_2 рекурсивным образом создаются узлы n_1 и n_2, которые помещаются в узел n в качестве поддеревьев. На последнем шаге узел n возвращается в качестве результата.

4.3.2.1 Экспериментальное исследование

В исследовании, описанном в работе автора [206*], были использованы два хорошо известных набора данных. Первый из них – база данных MNIST рукописных цифр [151]. Второй – набор данных Corel Image Features Data Set [52].

Первая база данных содержит полутоновые изображения рукописных цифр. База данных разделена на два множества: обучающее множество, содержащее 60000 изображений, и тестовое множество, содержащее 10000 изображений. В экспериментах использовались изображения обучающего множества размером 28x28 пикселей, трактуемые как вектора в пространстве размерностью 784.

Второй набор данных содержит признаки, вычисленные по цифровым изображениям из коллекции Corel image collection [51]. Набор Corel Image Features Data Set рассчитан по 68040 изображениям. При проведении экспериментов использовались следующие признаки:

Набор 1 - цветовые моментные характеристики [277] из набора данных Corel Image Features Data Set [52], рассчитанные по цифровым изображениям из коллекции [51]. Рассчитывалось по три характеристики для каждой из цветовых

компонент: среднее, СКО и коэффициент ассиметрии [345]. Размерность пространства признаков - 9.

Набор 2 - текстурные признаки на основе матриц совместной встречаемости [99] из набора данных Corel Image Features Data Set [52]. Рассчитывались четыре характеристики (второй угловой момент, контраст, обратный момент разности, энтропия) по четырем направлениям (горизонтальному, вертикальному и двум диагональным) в плоскости изображения. Размерность пространства признаков - 16.

Набор 3 - цветовые гистограммы [284], построенные в цветовом пространстве HSV. Цветовое пространство разбивалось на 8 частей по H и на 4 части по S компонентам. Размерность пространства признаков - 32.

Для оценки эффективности методов измерялись и рассчитывались следующие характеристики:

- время построения дерева для декомпозиции пространства,
- время построения списков опорных узлов,
- длина списков опорных узлов,
- время инициализации координат в пространстве малой размерности,
- время выполнения одной итерации оптимизационного процесса,
- ошибка представления многомерных данных.

Все описанные методы были реализованы на C++. Исследования с использованием набора данных MNIST выполнялись на ПК на базе Intel Core i5-3470 CPU с тактовой частотой 3.2 GHz. Исследования с использованием набора данных COREL выполнялись на ноутбуке на базе Intel Core i3 M370 CPU с тактовой частотой 2.4 GHz.

Некоторые результаты показаны на рисунках 4.9-.

Работа методов останавливалась при замедлении скорости снижения ошибки (если относительное снижение ошибки за 10 итераций не превышало 0.01). Во всех случаях размерность формируемого пространства была выбрана равной двум (решалась задача двумерного отображения).

На рисунке 4.9 показана зависимость количественных и качественных характеристик от порогового параметра T в выражении (4.10), при достижении

которого алгоритм переходит к дочерним узлам соответствующей структуры декомпозиции (метрического (vp-) дерева, KD-дерева или дерева кластеров).

Как видно из результатов, малые значения T (особенно $T<1$) приводят к ожидаемому снижению качества отображения по причине более грубой аппроксимации, что отражается в больших значениях ошибки представления многомерных данных ε (см. рисунок 4.9(г)). Время, затрачиваемое на одну итерацию, возрастает с ростом T (см. рисунок 4.9(в)) в связи с большим количеством обрабатываемых опорных узлов соответствующей иерархической структуры (см. рисунок 4.9(б)).

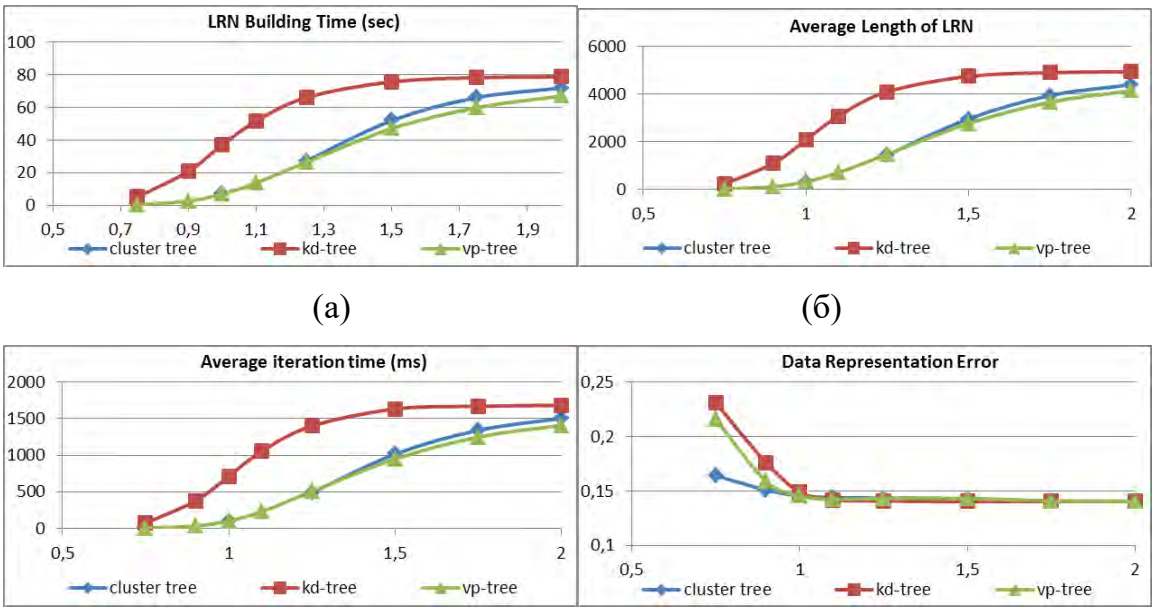
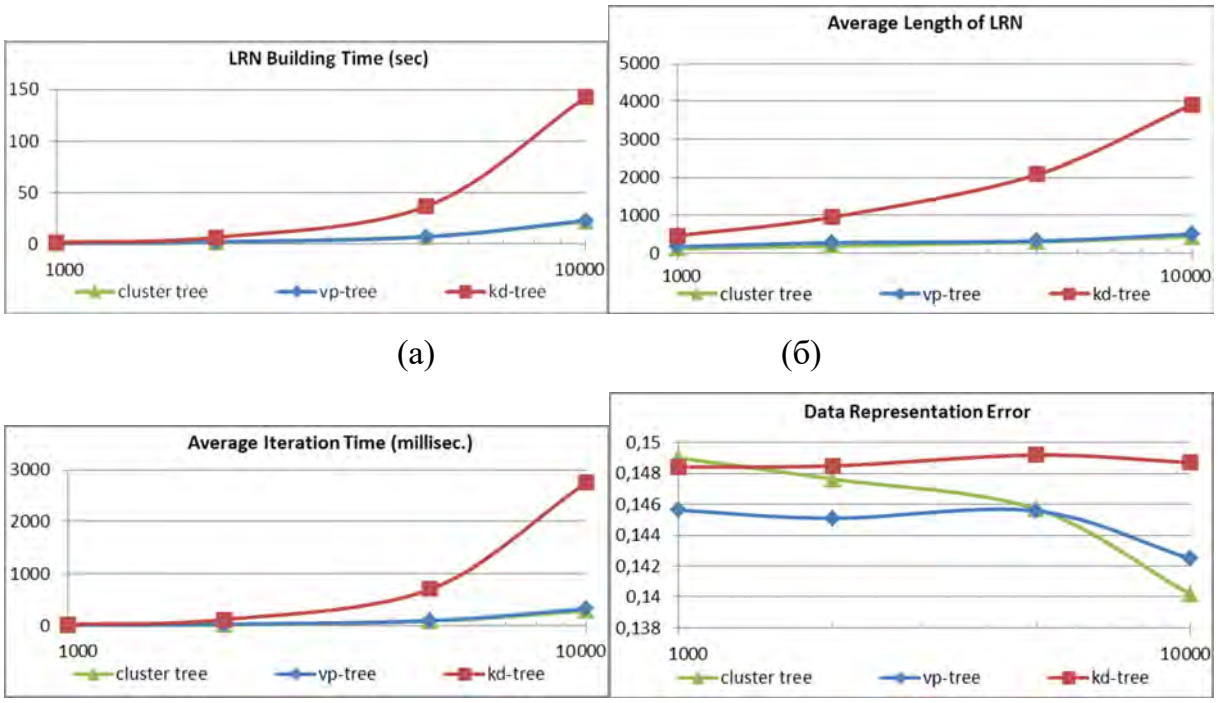


Рисунок 4.9 - Зависимости от порога T (набор данных MNIST):
(а) среднее время построения списков опорных узлов (в секундах);
(б) средняя длина списка опорных узлов;
(в) среднее время итерации оптимизационного процесса (в миллисекундах);
(г) ошибка представления многомерных данных ε

Зависимости качественных и временных характеристик от числа обрабатываемых точек показаны на рисунках 4.10 и 4.11. Качество отображения, измеряемое ошибкой представления многомерных данных ε (см. рисунок 4.10(г), 4.11(г)), слабо зависит от типа разбивающей пространство структуры. В то же время средняя длина списка опорных узлов существенно выше при использовании kd-

дерева (см. рисунки 4.10(б), 4.11(б)). Это подтверждает тот факт, что kd-деревья плохо подходят для обработки многомерных данных.

Следует отметить, что некоторый выигрыш в ошибке представления многомерных данных на 4.11(г) объясняется длиной списка опорных узлов (для kd-деревьев он в несколько раз больше, чем для других структур).



(а) (б)

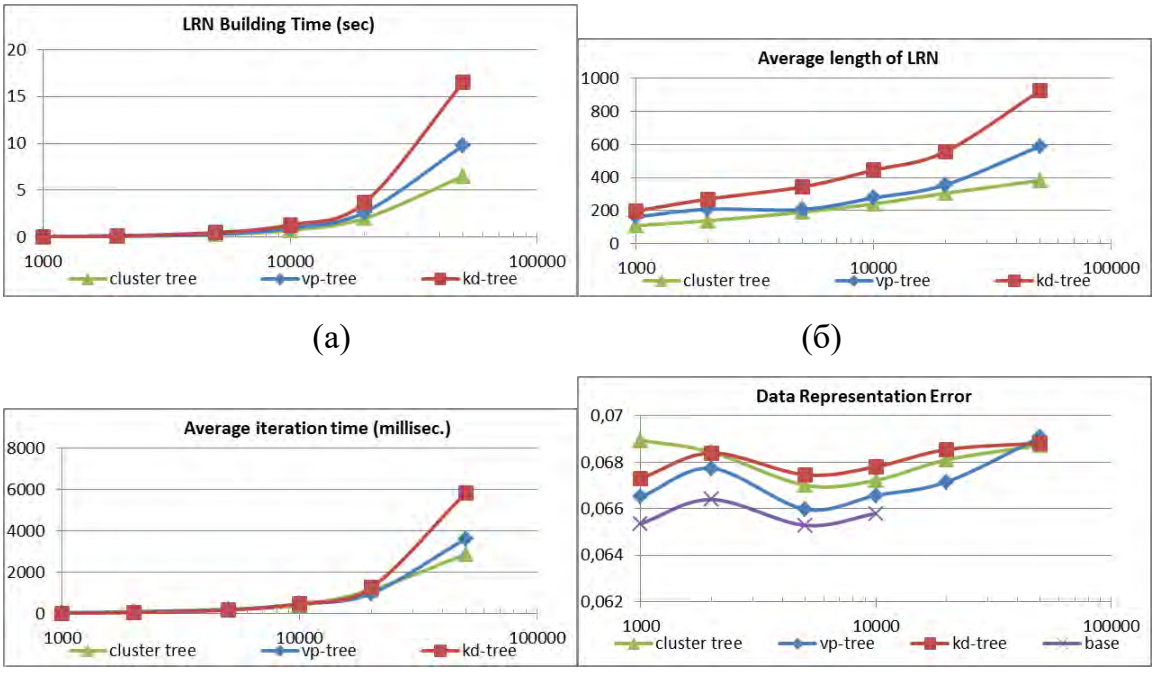
(в) (г)

Рисунок 4.10 - Зависимости от размера выборки (набор данных MNIST):
(а) среднее время построения списков опорных узлов (в секундах);
(б) средняя длина списка опорных узлов;
(в) среднее время итерации оптимизационного процесса (в миллисекундах);
(г) ошибка ε представления многомерных данных
в пространстве меньшей размерности

Некоторые оценки времени работы, полученные для базового метода снижения размерности, показаны на рисунке 4.12 (логарифмический масштаб). Оценки времени для случая предварительно вычисленных расстояний не проводились для больших объемов выборки (более 10000 точек) в связи с ограничениями оперативной памяти. Ошибка представления многомерных данных базового метода показана для сравнения на рисунке 4.11 (г). Как можно видеть, оценки значения ошибки для базового метода лишь немного лучше, чем для исследуемых методов. При этом

оценки качества для выборки более, чем 10000 точек, для базового метода не представлены в связи с неприемлемыми временными затратами на их получение.

Эксперименты, проведенные на других наборах данных, описанных выше, показывают аналогичные результаты.



(а) (б)

(в) (г)

Рисунок 4.11 - Зависимости от размера выборки (набор данных COREL):
(а) среднее время построения списков опорных узлов;
(б) средняя длина списка опорных узлов;
(в) среднее время итерации оптимизационного процесса;
(г) ошибка представления многомерных данных ε

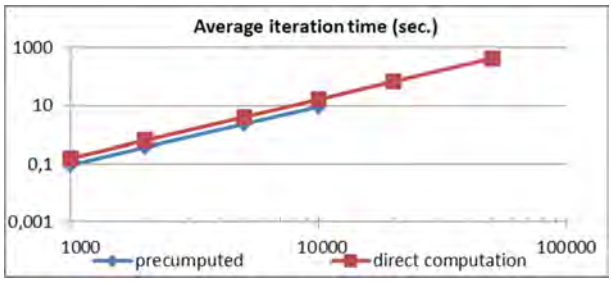


Рисунок 4.12 - Зависимость среднего времени выполнения одной итерации от размера выборки для базового алгоритма (набор данных COREL)

4.3.3 Разбиение выходного пространства

При решении задачи формирования признаков вышеупомянутые подмножества могут быть сформированы на основе характеристик точек данных в двух пространствах: во входном многомерном пространстве и в формируемом пространстве низкой размерности. Исследования по этому вопросу были проведены в работе автора [215*], где наряду с описанным выше методом, основанным на разбиении исходного пространства, исследовался подход, основанный на разбиении выходного пространства.

В отличие от рассмотренного в п. 4.3.1 метода, выполнение разбиения в выходном пространстве неизбежно ведет к необходимости периодического обновления или перестройки той структуры, с использованием которой выполняется разбиение. Причина этого заключается в изменении координат точек в выходном пространстве по мере выполнения оптимизационного процесса. По этой же причине при аппроксимации на каждой итерации оптимизационного процесса будут использоваться обновленные или перестроенные узлы структуры разбиения пространства, и построение структур, подобных спискам опорных узлов, теряет смысл.

В работе автора [215*] для разбиения выходного пространства малой размерности были использованы KD-деревья [19]. Их построение выполняется перед каждой итерацией оптимизационного процесса, а сам метод формирования признаков состоит из следующих шагов.

1. Инициализация положения точек в выходном пространстве.
2. Построение KD-дерева для разбиения выходного пространства малой размерности.
3. Построенное дерево используется для выполнения очередной итерации оптимизационного процесса. При этом для каждой точки исходного множества построенное дерево обходится, начиная с корневого узла, и вычисляется сумма, аналогичная выражению (4.12). Те узлы дерева, которые не удовлетворяют критерию декомпозиции (хорошо отделены от рассматриваемой точки) в выходном пространстве (см. рисунок 4.13), рассматриваются в совокупности как один объект, и выполняется аппроксимация в соответствии с выражением (4.13). Для узлов, удовлетворяющих критерию декомпозиции и содержащих

непосредственно точки исходного множества, выполняется расчет приращений аналогично выражению (4.15). Для узлов с истинным значением критерия декомпозиции, содержащих дочерние узлы, выполняется переход к дочерним узлам, и аналогичным образом обходятся оба поддерева рассматриваемого узла.

4. Если положение точек в выходном пространстве стабилизировалось, то алгоритм завершает работу, иначе – переход к шагу 2.

Следует отметить, что указанные на шаге 3 действия выполняются на каждой итерации для каждой точки, поэтому для повышения вычислительной эффективности построенное дерево обходится не рекурсивно, а итеративно с использованием специальных указателей.

Условие декомпозиции, использованное при обходе дерева, аналогично рассмотренному ранее и ограничивает оценку угла, под которым минимальный ограничивающий узел прямоугольник наблюдается из рассматриваемой точки в целевом пространстве (см. рисунок 4.13).

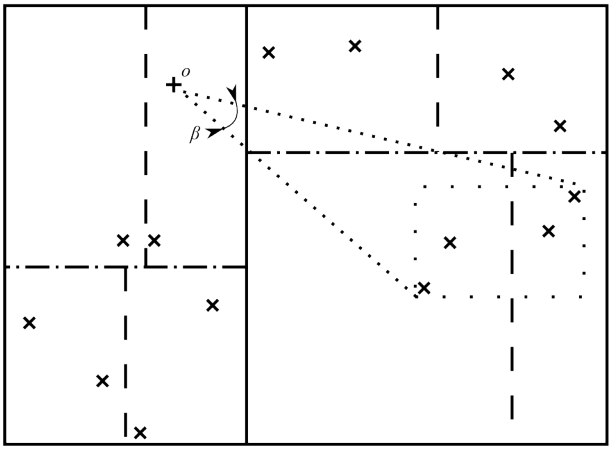


Рисунок 4.13 - Иллюстрация декомпозиции в целевом пространстве с использованием KD-дерева. Различные уровни декомпозиции показаны различными линиями: первый уровень показан сплошной линией, второй уровень показан штрих-пунктирной линией, третий – штриховой линией. Группа из четырех точек (минимальный ограничивающий прямоугольник показан точками) будет разбита на две, если угол β, под которым минимальный ограничивающий прямоугольник наблюдается из точки o больше угла декомпозици δ.

Построение дерева выполняется перед каждой итерацией оптимизационного процесса с использованием следующей рекурсивной процедуры:

1. Вычисление характеристик текущего узла дерева (средние координаты в исходном и целевом пространстве, границы узла в целевом пространстве).

2. В том случае, если в узле содержится лишь одна точка, то - выход.

3. Разбиение узла на два дочерних перпендикулярно оси координат, вдоль которой разброс данных максимален, так, чтобы количество точек в дочерних узлах отличалось не более чем на единицу.

4. Выполнение этой процедуры для вновь образованных дочерних узлов.

4.3.4 Учет характеристик узлов во входном и выходном пространствах

Можно предположить, что для повышения точности (снижения ошибки представления многомерных данных (2.9)) при формировании признаков, необходимо при выполнении декомпозиции учитывать характеристики узлов как во входном, так и в выходном пространствах. С одной стороны это позволит контролировать ошибку аппроксимации, поскольку при расчетах аппроксимируются расстояния в обоих пространствах. С другой стороны, мы имеем дело с алгоритмами, действующими по принципу сохранения попарных рассогласований между точками. Следовательно, если некоторое подмножество точек достаточно компактно во входном многомерном пространстве, то можно рассчитывать, что выходные точки данных низкой размерности также будут располагаться достаточно компактно, и центр подмножества в формируемом пространстве также будет хорошим представителем содержащиеся в нем точек данных.

Так как при построении алгоритма нам бы хотелось избежать повторного перестроения структуры разбиения на каждом шаге, в описываемом ниже методе в качестве основы будет использоваться структура, построение которой производится во входном многомерном пространстве. Так же, как и ранее (см. п. 4.3.1), по полученной структуре производится построение списков, содержащих ссылки на узлы структуры разбиения, удовлетворяющие критерию декомпозиции во входном пространстве. Однако, в отличие от представленного в п. 4.3.1 метода, элементы списка используются непосредственно, только тогда, когда для соответствующих узлов критерий декомпозиции не выполняется также и в выходном пространстве. В

случае же, если критерий декомпозиции в выходном пространстве выполняется, проверяются на соответствие критерию уже дочерние узлы.

Этапы 1-3. Этапы 1-3 метода, описанного в работах автора [203*, 201*], полностью аналогичны соответствующим этапам метода, изложенного в п. 4.3.1. Отличия заключаются лишь в четвертом этапе, более подробное описание которого приводится ниже.

Этап 4. Итеративная процедура оптимизации.

Последний этап работы метода связан с уточнением положения точек в выходном пространстве малой размерности с использованием рекуррентного выражения (4.12). Вычисление слагаемых $\widetilde{\varsigma}(x_i, y_i^{[t]}, n)$ в выражении (4.12) может быть связано с дальнейшей декомпозицией элементов $n \in \Lambda_i$ из списка опорных узлов. Алгоритм вычисления слагаемого ς для некоторой точки x по опорному узлу n, записанный в виде псевдокода, выглядит следующим образом.

```
Procedure Sum( Point x, Point y, Node n, Vector ς)
begin
    if ϑ(n, y) = False then
        ς = ς + ς̃(x, y, n)
    else if n contains Points then
        for each Point p in n
            ς = ς + ς(x, x_p, y, y_p)
    else
        for each node c in n
            Sum(x, y, c, ς)
end
```

В приведенном алгоритме опорный узел учитывается целиком, если критерий декомпозиции не выполняется $\vartheta(n, y) = False$. Формально здесь используется критерий, отличный от того, что использовался при формировании списков опорных узлов: критерий $\vartheta(n, y)$ работает в формируемом пространстве малой размерности, в то время как $\theta(n, x)$ работает в исходном многомерном пространстве.

Эксперименты

Как было отмечено выше, выражение (4.8) аппроксимирует (4.1) лишь с некоторой точностью. Ошибка аппроксимации для некоторой точки x_i по множеству узлов Λ_i на шаге t может быть записана следующим образом:

$$\xi_i = \left\| \sum_{j=1}^{N} \varsigma(x_i, x_j, y_i^{[t]}, y_j^{[t]}) - \sum_{n \in \Lambda_i} \widetilde{\varsigma}(x_i, y_i^{[t]}, n) \right\| \tag{4.20}$$

Тогда средняя ошибка аппроксимации на шаге t будет иметь вид.

$$\xi = \frac{1}{N} \sum_{i=1}^{N} \xi_i = \frac{1}{N} \sum_{i=1}^{N} \left\| \sum_{j=1,\ j \neq i}^{N} \varsigma(x_i, x_j, y_i^{[t]}, y_j^{[t]}) - \sum_{n \in \Lambda_i} \widetilde{\varsigma}(x_i, y_i^{[t]}, n) \right\| \tag{4.21}$$

При проведении экспериментов будем считать, что именно ошибка аппроксимации (4.21) отражает качество аппроксимации с использованием описанного выше метода.

Временные характеристики разработанного метода зависят, в основном, от времени работы итеративной процедуры оптимизации. Время работы указанной процедуры, в свою очередь, напрямую зависит как от количества итераций, так и от количества узлов, обрабатываемых на каждом шаге итеративной процедуры, то есть от количества слагаемых в (4.8). Отметим, что количество итераций оптимизационной процедуры может сильно варьироваться в зависимости от выбранных параметров метода (начального значения коэффициента α, стратегии изменения коэффициента, критерия остановки итерационной процедуры и т.д.). Учитывая, что временные характеристики, полученные на разных ЭВМ, также будут различны, при проведении экспериментов представляется целесообразным оценивать среднее количество обрабатываемых узлов L на одну точку исходного множества (4.17).

Таким образом, для исследования предложенных методов фиксировались ошибка (2.10) представления данных в пространстве меньшей размерности, ошибка аппроксимации (4.21) и среднее количество обрабатываемых узлов L.

При проведении экспериментов были использованы наборы данных, извлеченные из цифровых изображений и значительно отличающиеся друг от друга по составу признаковой информации. Использовались наборы данных Набор 1 (цветовые моментные характеристики) и Набор 2 (текстурные признаки), описанные

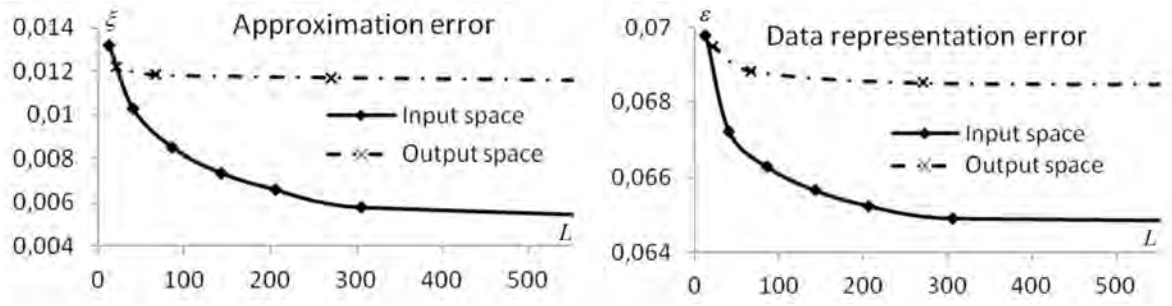
выше в п. 4.3.2. Кроме того использовался Набор 3, представляющий собой пиксели гиперспектрального спутникового снимка Indian pines [17] (см. п. 2.4.1).

Ввиду того, что построение приведенных в настоящем разделе графиков требует многократного прямого расчета сумм в соответствии с (4.21), приведенные результаты получены для случайных подвыборок размером 5000 векторов (точек) из указанных наборов данных.

При реализации метода иерархическое разбиение исходного множества точек на подмножества (первый этап метода, изложенного в настоящем п. 4.3.4), осуществлялось с использованием метрических деревьев (vp-деревьев). Как уже говорилось ранее, такие деревья могут быть построены за $O(N \log N)$ операций (N – количество точек), что является приемлемым для рассматриваемой задачи. Так как алгоритм построения этой структуры хорошо известен, он здесь не приводится.

Критерий декомпозиции как во входном, так и в выходном пространстве задавался выражением (4.10) с настраиваемыми порогами T_1 и T_2, для входного и выходного пространств, соответственно.

На рисунке 4.14 приведены результаты сравнения методов снижения размерности, основанных на разбиении входного (обозначен «Input space», см.п. 4.3.1) и выходного (обозначен «Output space», см.п. 4.3.3) пространств. Как видно из приведенных результатов, разбиение входного многомерного пространства является гораздо более предпочтительным, чем разбиение выходного пространства малой размерности как с точки зрения ошибки аппроксимации (4.21), так и с точки зрения ошибки представления многомерных данных (2.10). Приведенные на рисунке результаты относятся к набору данных 1. Результаты аналогичных экспериментов для других наборов данных можно найти в [203*].

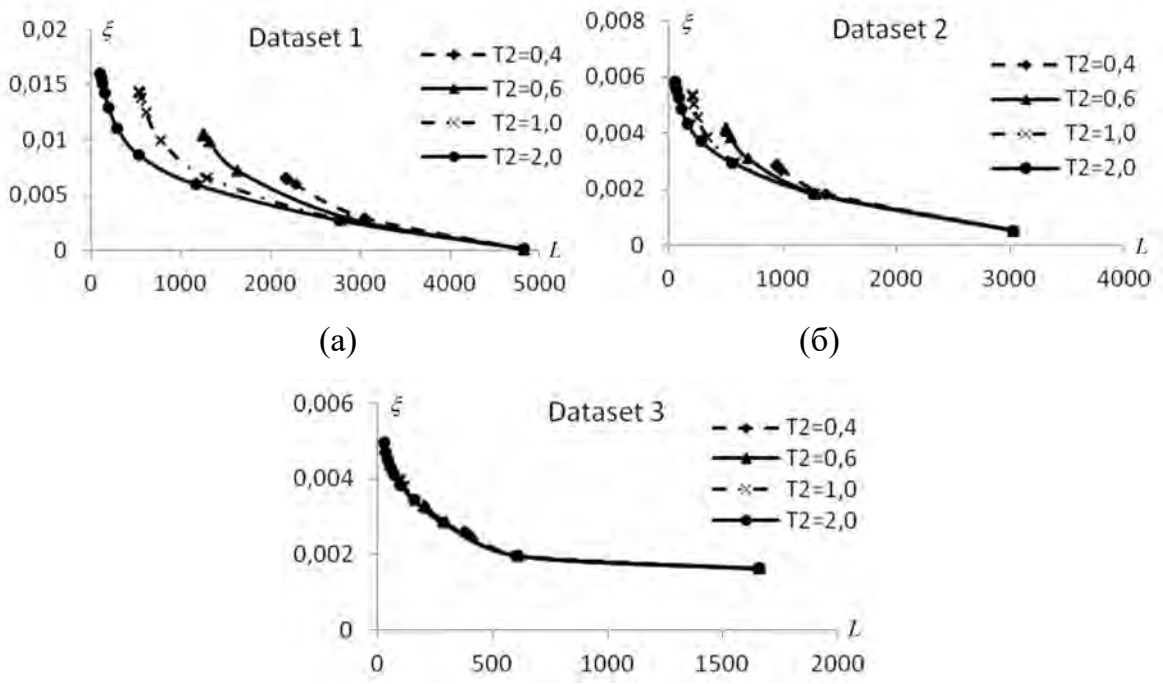


(а) (б)

Рисунок 4.14 - Сравнение методов снижения размерности, основанных на разбиении входного и выходного пространств для набора данных 1: зависимость ошибки аппроксимации ξ от количества узлов L, участвующих в аппроксимации (а), зависимость ошибки представления данных ε от количества узлов L, участвующих в аппроксимации (б)

На рисунке 4.15 показаны семейства кривых, отображающих зависимость ошибки аппроксимации (4.21) от среднего количества узлов, участвующих в аппроксимации, для метода, изложенного в настоящем п. 4.3.4, учитывающего характеристики как во входном, так и в выходном пространствах. Линиями разных типов показаны кривые, полученные для различных пороговых значений T_2. Для построения каждой кривой значение порога T_1 варьировалось в диапазоне [0,2;2,0].

Как видно из приведенных рисунков, для всех рассмотренных наборов данных выявлены схожие зависимости, показывающие, что с уменьшением пороговых значений T_1 и T_2 количество участвующих в аппроксимации узлов растет, что приводит к ожидаемому снижению ошибки аппроксимации.



(а) (б)

(в)

Рисунок 4.15 - Зависимости ошибки аппроксимации ξ от количества узлов L, участвующих в аппроксимации:
для набора данных 1 (а), для набора данных 2 (б), для набора данных 3 (в)

Так как выполнять аппроксимацию желательно с минимальными вычислительными затратами, на указанных графиках нас более всего интересует часть, соответствующая небольшому (до нескольких сотен) количеству узлов. В особенности нас интересует вопрос, возможно ли за счет контроля конфигурации узлов в выходном пространстве выполнить аппроксимацию с более высоким качеством или меньшими вычислительными затратами, чем без такого контроля (с использованием метода из раздела 4.3.1).

На рисунке 4.16 показаны графики зависимости ошибки аппроксимации (4.21) от среднего количества узлов, участвующих в аппроксимации, для небольшого количества узлов (до 500 штук). Кроме кривых для метода с контролем положения узлов в выходном пространстве, изложенного в настоящем п. 4.3.4, на рисунке сплошной линией также показаны графики зависимостей для базового метода *Base* без контроля в выходном пространстве (п. 4.3.1).

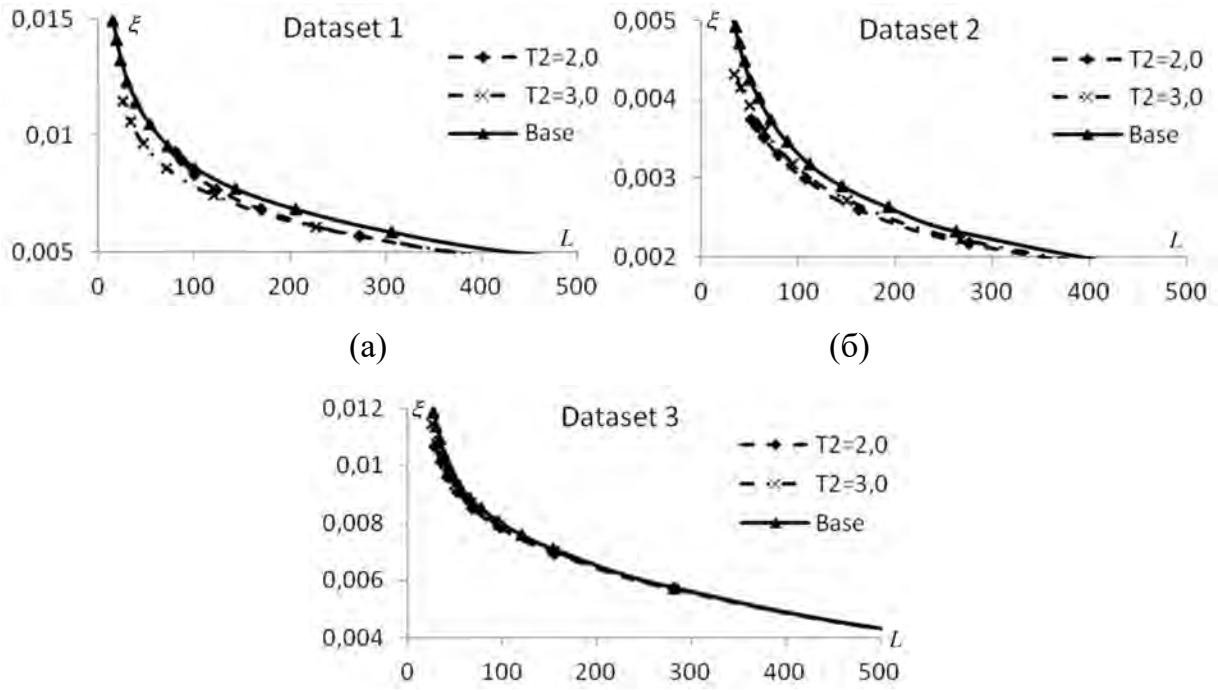
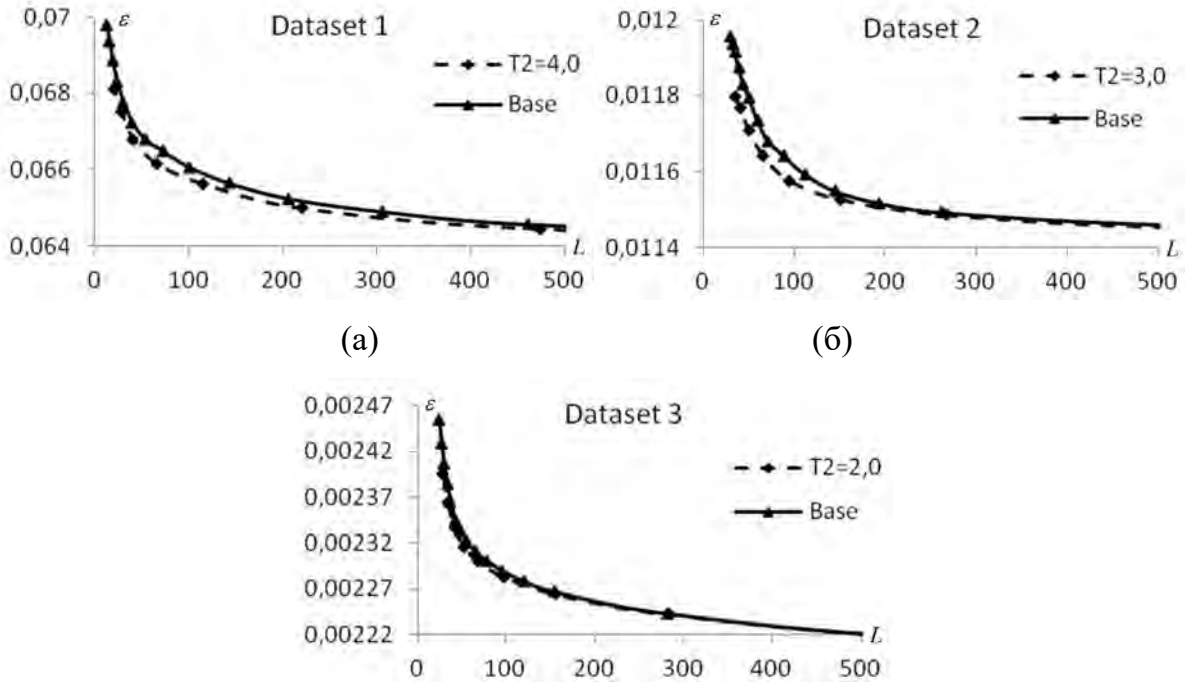


Рисунок 4.16 - Зависимости ошибки аппроксимации ξ от количества узлов L, участвующих в аппроксимации:
для набора данных 1 (а), для набора данных 2 (б), для набора данных 3 (в)

Как видно из приведенного рисунка, при сравнительно слабых ограничениях на положение узлов в выходном пространстве (при значениях T_2 = 2,3) графики для разработанного метода лежат ниже базового метода без ограничений в выходном пространстве. Фактически это означает, что при наложении сравнительно слабых ограничений в выходном пространстве можно получить более точную аппроксимацию с использованием того же количества узлов, то есть за то же время. Либо получить аппроксимацию того же качества с использованием меньшего количества узлов, то есть за меньшее время. Как и ранее, для набора данных 3 заметного выигрыша за счет учета ограничений в выходном пространстве получить не удается.

На рисунке 4.17 показана зависимость ошибки представления многомерных данных (2.10) от количества узлов, участвующих в аппроксимации, после выполнения 50 итераций. Как видно из приведенных графиков, для указанных наборов данных снижение ошибки аппроксимации приводит также к снижению ошибки представления многомерных данных (2.10). Отметим, что по мере выполнения

итераций и уточнения положения точек в выходном пространстве ошибка аппроксимации снижается.



(а) (б)

(в)

Рисунок 4.17 - Зависимость ошибки представления данных ε от количества узлов L, участвующих в аппроксимации:
для набора данных 1 (а), для набора данных 2 (б), для набора данных 3 (в)

На рисунке 4.18 показана зависимость ошибки аппроксимации от числа итераций оптимизационного процесса для некоторых параметров предложенного метода, а также для базового метода (из п. 4.3.1). Параметры подбирались таким образом, чтобы количество узлов, участвующих в аппроксимации, было примерно одинаковым для всех методов, а коэффициент α оставался неизменным. В таблице 4.2 показаны окончательные результаты работы метода при тех же настройках.

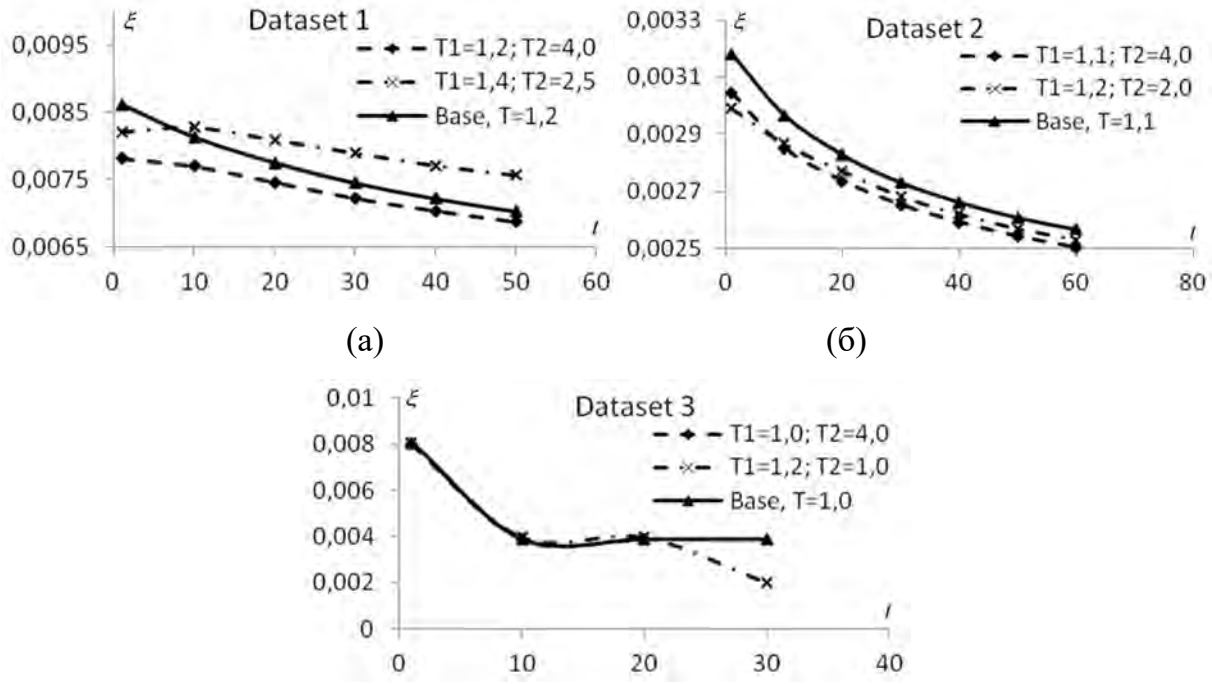


Рисунок 4.18 - Зависимость ошибки аппроксимации ξ от количества итераций *t*: для набора данных 1 (а), для набора данных 2 (б), для набора данных 3 (в)

Таблица 4.2 – Конечная ошибка представления многомерных данных в пространстве меньшей размерности

Набор данных	Метод	T_1	T_2	L	ε
Набор данных 1	п. 4.3.4	1,2	4,0	110,9	**0,06689**
	п. 4.3.4	1,4	2,5	92,5	0,06704
	Базовый (п. 4.3.1)	1,2	-	100,7	0,06727
Набор данных 2	п. 4.3.4	1,1	4,0	113,9	**0,01172**
	п. 4.3.4	1,2	2,0	107,7	**0,01172**
	Базовый (п. 4.3.1)	1,1	-	111,5	0,01174
Набор данных 3	п. 4.3.4	1,0	4,0	95,7	0,002289
	п. 4.3.4	1,2	1,0	95,5	**0,002287**
	Базовый (п. 4.3.1)	1,0	-	95,3	0,002291

Как видно из приведенных результатов, влияние пороговых значений (для учета характеристик в выходном пространстве) на конечную ошибку представления данных (2.10) для рассмотренных наборов данных оказывается не столь значительным. Отметим также, что, с учетом снижения ошибки аппроксимации, с некоторого момента накладные расходы, связанные с наложением ограничений в

выходном пространстве, могут стать неоправданными. В таком случае имеет смысл отказаться от таких ограничений после выполнения определенного количества итераций.

4.3.5 Управление вычислительной сложностью

Следует отметить, что длина сформированных при выполнении метода (п. 4.3.1) списков опорных узлов определяет вычислительную сложность процесса оптимизации. Сама длина зависит от нескольких факторов: распределения точек, способа разбиения пространства, критерия декомпозиции и т.д. В результате трудно предсказать длину и, следовательно, регулировать вычислительную сложность процесса оптимизации. Поэтому для выбора параметров критерия декомпозиции и получения желаемой средней длины списков требовалось проведение отдельных экспериментов.

В то же время, описанный выше (в подразделе 4.2) алгоритм стохастического градиентного спуска позволяет регулировать сложность вычислений напрямую, варьируя количество элементов в мини-подвыборках. Чтобы придать аналогичные полезные свойства и методам, основанным на разбиении пространства, в [213*] был предложен подход, основанный на очередях с приоритетом. Основная идея этого подхода состоит в том, чтобы вместо списков использовать очереди с приоритетами с длиной, ограниченной заранее заданным значением.

Для этого определим некоторую меру отделимости, которая показывает, насколько хорошо рассматриваемый узел дерева разбиения отделен от рассматриваемой точки. Эту меру отделимости будем использовать в качестве приоритета в очереди, так чтобы наиболее плохо отделенные узлы были первыми, а за ними следовали узлы, которые отделены лучше.

Алгоритм построения очереди начинается с единственного элемента, представляющего корень дерева разбиения пространства, после чего следует декомпозиция первого элемента очереди (наиболее плохо отделенного) до тех пор, пока длина очереди не достигнет предварительно определенного значения L.

Псевдокод, показанный ниже, описывает алгоритм, который строит приоритетную очередь для точки данных p, используя корень *root* двоичного дерева разбиения пространства. Длина построенной очереди не превышает *maxlen*.

```
function ConstructPriorityQueue( DataPoint p, Node root, int L ) returns PriorityQueue
begin
        create new PriorityQueue queue
        queue.insert( INFINITY, root )
        while (  ( queue.length()  < L ) and (queue.max_priority() > 0)  )  begin
                Node s = queue.pull()
                if s contains Nodes
                        for each Node n contained in s do   queue.insert( Priority(n, p), n )
                else  /* s contains Points */
                        for each Point a contained in s do   queue.insert( 0, a )
        end
        return queue
end
```

Здесь мы предполагаем, что структура данных *PriorityQueue* поддерживает следующие операции:

- вставка *insert*(*Priority, Node*) для добавления в очередь нового элемента Node с приоритетом Priority;

- извлечение *pull*() для удаления элемента с наивысшим приоритетом из очереди и возврата его вызывающей стороне;

- *max_priority*() для возврата приоритета элемента очереди с наивысшим приоритетом;

- *length*() для возврата текущей длины (размера) очереди.

Приоритет элемента показывает, насколько груба аппроксимация соответствующего узла. Элемент имеет низкое значение приоритета, если он находится далеко (хорошо отделен) от рассматриваемой точки данных, и имеет высокое значение приоритета, если он находится близко (плохо отделен) от рассматриваемой точки. Таким образом, приоритетные элементы декомпозируются в первую очередь.

Эксперименты

При исследовании измерялась ошибка представления многомерных данных и среднее время выполнения одной итерации для предложенного метода на основе очередей с приоритетами и метода на основе стохастического градиентного спуска (подраздел 4.2).

Оба описанных метода были реализованы на C++ и проверены на тестовых гиперспектральных изображениях. Результаты для гиперспектрального изображения Indian Pines [17] (см.п. 2.4.1) показаны на рисунке Рисунок 4.19.

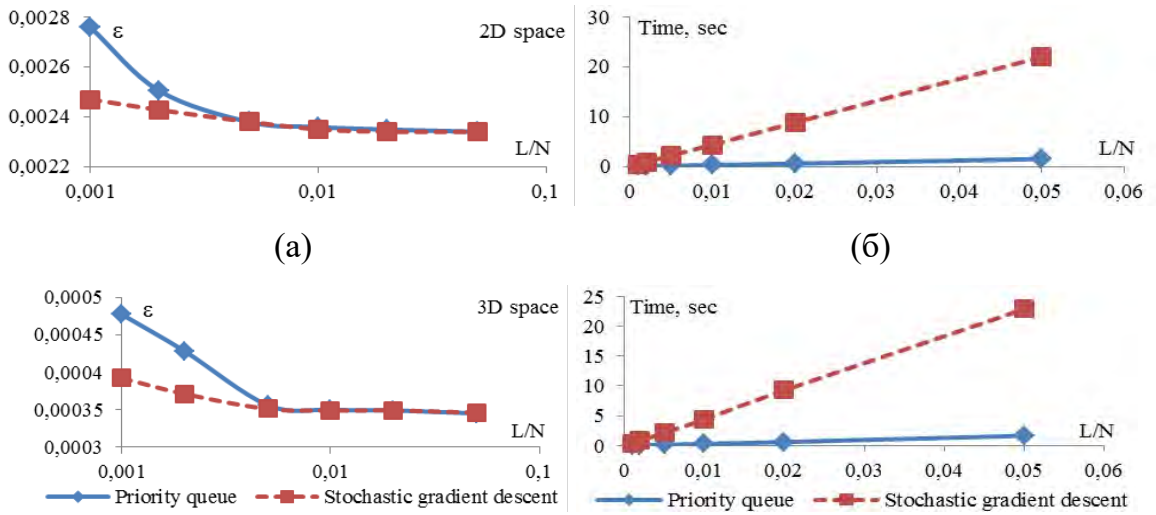


Рисунок 4.19 - Результаты экспериментов для формирования 2D и 3D пространства признаков. Зависимость ошибки представления многомерных данных ε от доли *L/N* элементов, используемых в аппроксимации (логарифмическая шкала):
(а) для двумерного пространства, (в) для трехмерного пространства.
Зависимость среднего времени на одну итерацию процесса оптимизации (в секундах) от доли *L/N* элементов, используемых в аппроксимации:
(б) для двумерного пространства, (г) для трехмерного пространства.

Как видно, оба исследованных метода дают практически одинаковые значения погрешности для доли элементов L/N > 0,005 (>0,5%). Среднее время итерации почти линейно в обоих случаях, что подтверждает тот факт, что вычислительная сложность линейно зависит от количества элементов, используемых в аппроксимации.

При этом предложенная в настоящем п. 4.3.5 схема позволяет не только уменьшить количество слагаемых путем аппроксимации подмножеств их центрами, но и сделать процесс оптимизации независимым от размерности входного пространства. Последнее достигается путем предварительного вычисления и сохранения входных расстояний от точек до центров подмножеств, используемых в приближении (опорных узлов). Это требует O(*LN*) памяти, но позволяет значительно сократить время процесса оптимизации (см. рисунок 4.19(б, г)) даже по сравнению с алгоритмом стохастического градиентного спуска.

4.4 Интерполяция

Другим способом решения проблемы высокой вычислительной сложности для рассматриваемой задачи является использование алгоритмов интерполяции. При использовании этих алгоритмов K точек сначала отображается из многомерного пространства в формируемое пространство меньшей размерности, используя один из рассмотренных выше методов формирования признаков, а затем оставшиеся (N-K) точек последовательно добавляется в формируемое пространство, используя один из алгоритмов многомерной интерполяции. В работе автора [216*] исследован ряд алгоритмов многомерной интерполяции, включая интерполяцию ближайшего соседа, взвешивание обратных расстояний, радиальные базисные функции и интерполяцию путем минимизации ошибки представления данных. Опишем эти алгоритмы и полученные результаты их исследования более подробно.

Предположим, что заданное множество известных точек x_i, i=1..K в многомерном пространстве R^M успешно отображено в соответствующие точки y_i в пространстве малой размерности R^L. Требуется получить интерполированное значение y прежде неизвестной точки x.

4.4.1 Интерполяция по ближайшему соседу

Интерполяция по ближайшему соседу (Nearest Neighbor, NN) - наиболее простой и быстрый алгоритм, учитывающий только ближайшую точку к интерполируемой:

$$y=y_k, \text{ такое что } k = \arg\min_i \left(d(x,x_i)\right). \tag{4.22}$$

4.4.2 Обратное взвешенное расстояние

В интерполяции по обратному взвешенному расстоянию (Inverse Distance Weighting, IDW, интерполяция Шепарда) [263] для получения интерполированного значения вычисляется взвешенная сумма значений в известных точках

$$y = \frac{\sum_{i=1}^{K} w_i(x) y_i}{\sum_{i=1}^{K} w_i(x)}. \tag{4.23}$$

Здесь веса $w_i(x)$ определяются функцией обратного взвешенного расстояния $w_i(x)=1/d(x,x_i)^p$, где p – параметр (степень) алгоритма.

4.4.3 Радиально-базисные функции

Интерполяция радиально-базисными функциями (Radial Basis Functions, RBF) основана на линейной комбинации радиальных ядер, помещенных в известные точки x_i [33]:

$$y = \sum_{i=1}^{K} w_i \varphi(\|x - x_i\|). \quad (4.24)$$

Здесь ядро $\varphi(.)$ может быть любой заданной радиально-базисной функцией, а w_i, i=1..K – веществennозначные коэффициенты. В качестве нормы ||.|| используется Евклидово расстояние $d(.)$, как наиболее распространенное.

Некоторые примеры ядер включают [251]:

- мультиквадратичное (multiquadric) $\varphi_m(r) = (r^2 + r_0^2)^{1/2}$,
- обратное мультиквадратичное (inverse multiquadric) $\varphi_{im}(r) = (r^2 + r_0^2)^{-1/2}$,
- сплайн «тонкая пластина» (thin plate spline) $\varphi_{tps}(r) = r^2 \log(r/r_0)$ (частный случай полигармонического сплайна (poliharmonic spline) и т.д.

Здесь r_0 – настраиваемый параметр.

Для вычисления весов формируется следующая система линейных уравнений:

$$\sum_{i=1}^{K} w_i \varphi\left(d\left(x_i, x_j\right)\right) - y_j = 0, \;\; j = 1...K. \quad (4.25)$$

Решение дает известные значения y_i в известных точках x_i и позволяет получать интерполированные значения в соответствии с приведенным выше выражением в других точках.

4.4.4 Интерполяция путем минимизации ошибки представления данных в пространстве меньшей размерности

Описанные выше алгоритмы интерполяции являются известными и общими для многомерной интерполяции. Другой подход состоит в минимизации заданного функционала ошибки для нахождения интерполированного значения. В контексте рассматриваемой задачи алгоритм итеративной интерполяции для выходного пространства с евклидовой метрикой принимает форму [216*]:

$$y^{[t+1]} = y^{[t]} + 2\alpha\mu \sum_{i=1}^{K} \rho_{i,idx(x)} \left(\frac{d(x, x_i)}{d\left(y^{[t]}, y_i\right)} - 1 \right) \left(y^{[t]} - y_i \right). \quad (4.26)$$

Здесь мы полагаем, что позиции y_i известных точек x_i остаются постоянными, а уточняется только позиция интерполируемой точки y.

4.4.5 Вычислительная сложность

Вычислительная сложность интерполяции NN и IDW может быть оценена как O(K) на одну интерполируемую точку. Для алгоритма RBF одна интерполяция требует O(K) операций, но кроме того, она требует O(K^3) операций для решения системы линейных уравнений. Интерполяция путем итеративной минимизации ошибки представления данных в пространстве меньшей размерности требует O(IK) операций на одну интерполируемую точку, где I – число итераций.

Таким образом, последний алгоритм наследует итеративную природу алгоритма градиентного спуска и требует O(IK(N-K)) операций для отображения оставшихся (N-K) точек данных, что сопоставимо со стохастическим градиентным спуском. Другие рассмотренные алгоритмы требуют O(K(N-K)) операций для отображения оставшихся точек, и, кроме того, интерполяция радиальными базисными функциями требует однократно O(K^3) операций.

4.4.6 Эксперименты

Для оценки описанных выше алгоритмов использовалось несколько хорошо известных наборов данных:

При проведении экспериментов были использованы наборы данных, извлеченные из цифровых изображений и значительно отличающиеся друг от друга по составу признаковой информации. Использовались наборы данных Набор 1 (цветовые моментные характеристики, CM), Набор 2 (текстурные признаки, CT), описанные выше в п. 4.3.2 и Набор 3, представляющий собой пиксели гиперспектрального снимка Indian pines (IP) [17] (см. п. 2.4.1).

На первой стадии экспериментов из приведенных выше наборов данных создавались выборки, содержащие N=5000 точек. После этого из выборок размера N отбирались подмножества x_i, i=1..K, содержащие K=100, 500, 1000 точек. Затем для формирования признаков y_i, i=1..K точек данных в выходном пространстве малой размерности L=2 (двумерное отображение) применялся базовый алгоритм формирования признаков, основанный на градиентном спуске. После этого многомерные координаты x_i и соответствующие координаты y_i в формируемом

пространстве низкой размерности, i=1..K использовались для получения решения для оставшихся (N-K) точек данных с использованием рассмотренных выше методов интерполяции.

Результаты эксперимента представлены в таблице 4.3.

Таблица 4.3 - Результаты эксперимента

K	Base	NN	IDW	RBF	ErM	SGD
Набор признаков 1: CT, M=16						
100	0,009802	0,04389	0,04202	0,01387	0,01681	0,01059
500	0,01005	0,02314	0,02239	0,01221	0,01035	0,01059
1000	0,01062	0,01834	0,01779	0,01119	0,01020	0,01060
Набор признаков 2: CM, M=9						
100	0,05733	0,1265	0,1253	0,078423	0,08077	0,06251
500	0,06281	0,09463	0,09388	0,073737	0,06416	0,06250
1000	0,06334	0,08388	0,08327	0,068599	0,06337	0,06250
Набор признаков 3: IP, M=220						
100	0,001883	0,02039	0,01947	0,002819	0,005982	0,002197
500	0,002409	0,008402	0,007992	0,002817	0,002302	0,002198
1000	0,002338	0,005975	0,005710	0,002585	0,002230	0,002198

Для оценки качества использовалась ошибка представления данных в пространстве меньшей размерности (2.12) для евклидовых расстояний (p=2), нормализованная в соответствии с (2.14). В столбце Base показана ошибка представления данных, полученная после применения базового алгоритма градиентного спуска для подмножеств, содержащих K точек данных. Ошибка представления данных для выборки, содержащей N=5000 после интерполяции по ближайшему соседу (NN), Шепарда (IDW), радиально-базисными функциями (RBF) с мультиквадратичным (multiquadric) ядром и путем минимизации ошибки представления данных (ErM) показаны в соответствующих столбцах. Для сравнения столбец SGD показывает ошибку представления данных после применения алгоритма стохастического градиентного спуска (п. 4.2) для выборок размером N=5000 точек данных без интерполяции.

В первой серии экспериментов были определены наилучшие параметры рассмотренных методов. После этого эксперименты выполнялись на выборках, содержащих N=10000 точек данных. Результаты показаны на рисунках 4.20-4.22 со временем работы. Следует отметить, что время показано в логарифмической шкале.

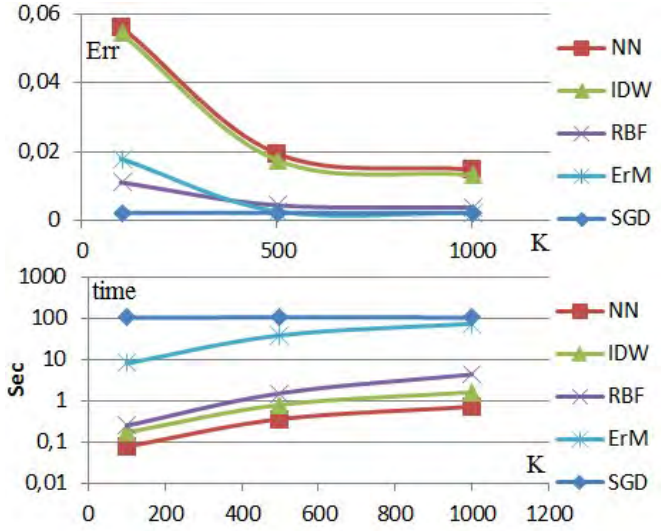


Рисунок 4.20 - Результаты экспериментов для текстурных признаков, основанных на матрице совместной встречаемости (набор признаков 1). Ошибка представления данных в пространстве меньшей размерности (сверху), и время (внизу)

Как можно видеть из результатов экспериментов, интерполяция по ближайшему соседу и обратного взвешенного расстояния дают большие значения ошибки, но работают быстрее, чем другие алгоритмы.

Радиально-базисные функции и интерполяция путем минимизации ошибки обеспечивают достаточно малые значения ошибки по сравнению с алгоритмом ближайшего соседа и интерполяцией Шепарда. В некоторых случаях при малом количестве базовых точек (K=100) RBF интерполяция превосходит интерполяцию путем минимизации ошибки. При больших значениях K предпочтение следует отдавать интерполяции путем минимизации ошибки.

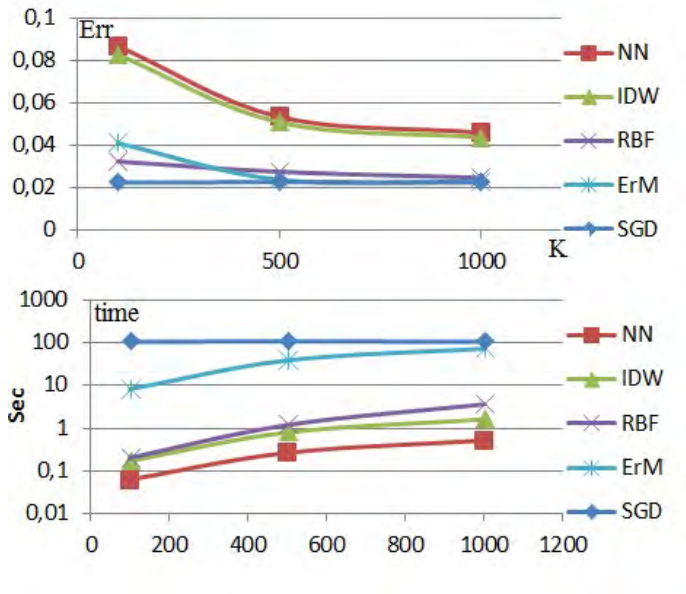


Рисунок 4.21 - Результаты экспериментов для цветовых моментных характеристик (набор признаков 2). Ошибка представления данных в пространстве меньшей размерности (сверху), и время (внизу)

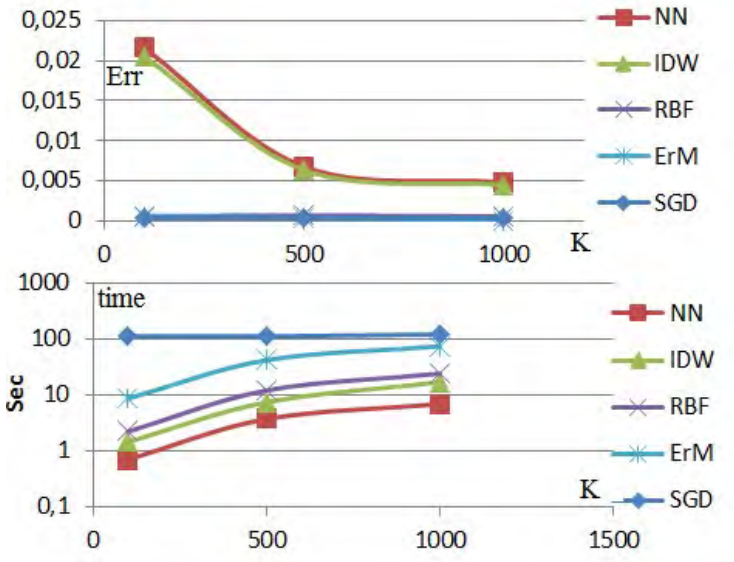


Рисунок 4.22 - Результаты экспериментов
для гиперспектрального спутникового изображения Indian Pines (набор признаков 3). Ошибка представления данных в пространстве меньшей размерности (сверху), и время (внизу)

В то же время последний метод был существенно более медленным в связи с его итеративной природой. Так, когда количество базовых и добавляемых точек было большим, полное время работы было почти таким же, как для алгоритма стохастического градиентного спуска (без интерполяции). Интерполяция радиально-базисными функциями производилась существенно быстрее.

Отметим, что интерполяция с использованием радиально-базисных функций может быть ускорена с использованием библиотек cuSOLVER [57] и cuBLAS [55], позволяющих решать системы линейных уравнений с использованием графических процессоров и включенных в состав инструментария CUDA [223].

4.5 Комбинирование методов повышения производительности

На рисунке 4.23 показана общая схема обработки, объединяющая описанные выше в настоящей главе методы и алгоритмы. Необязательные элементы схемы показаны пунктирной линией, альтернативные шаги показаны штрихпунктирной линией.

В приведенной схеме входной набор данных опционально разделяется на две части, первая из которых (основная) подвергается обработке в соответствии с методами, изложенными в подразделах 4.2 или 4.3 настоящей главы. Обработка второй (оставшейся) части набора данных выполняется с использованием методов интерполяции, описанных в подразделе 4.4. При этом точки из первой части набора рассматриваются как базовые, и результат их обработки используется при выполнении интерполяции точек из второй части набора данных.

Рисунок 4.23 - Общая схема обработки

Формирование признаков для первой части набора выполняется либо с использованием алгоритма стохастического градиентного спуска (см. подраздел 4.2), либо с использованием метода снижения размерности на основе очередей с

приоритетами. В последнем случае последовательно выполняются этапы построения бинарного дерева разбиения пространства (см. п. 4.3.2), построения очередей с приоритетами (см. п. 4.3.5) с последующей оптимизацией на основе аппроксимации градиента по опорным узлам (см. п. 4.3.1).

Результаты экспериментов для различных возможных конфигураций в соответствии с приведенной выше схемой представлены в таблице 4.3. Эксперименты были выполнены на гиперспектральном изображении Indian Pines, 220 диапазонов [17] (см. п. 2.4.1). Размерность формируемого пространства признаков была установлена равной 3 (3D отображение). Для экспериментов использовался ПК средней ценовой категории со следующим CPU: Intel Core i7-6500U с тактовой частотой 2.5 GHz.

Таблица 4.4 - Результаты экспериментов

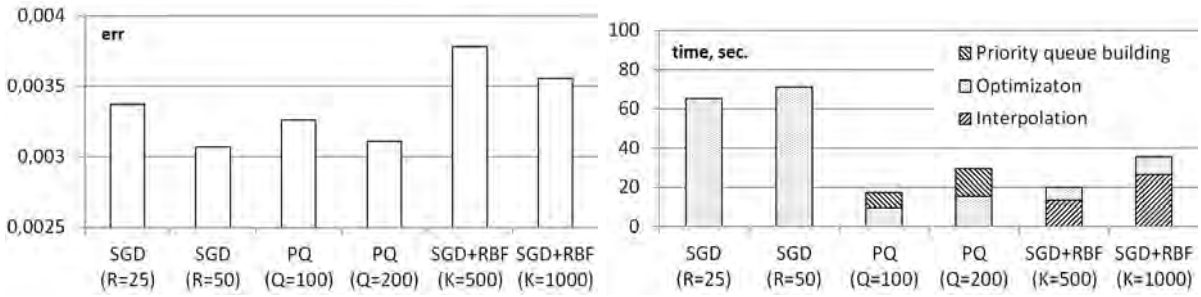
Конфигурация: метод отображения (K), метод интерполяции	Время построения BSPT и PQ, сек.	Время одной итерации, сек.	Размер подвыборки/ Длина очереди	Время интерполяции, сек.	Общее время, сек.	Конечная ошибка (после макс.100 итераций)
SGD (все множество)	-	0,60	25	-	65,5	0,00337
SGD (все множество)	-	0,83	50	-	71,4	0,00307
SGD (все множество)	-	1,26	100	-	119	0,00308
PQ (все множество)	7,9	0,19	100	-	17,8	0,00326
PQ (все множество)	14,0	0,40	200	-	29,6	0,00311
PQ (все множество)	24,6	0,81	400	-	50,6	0,00306
SGD(500 точек), NN	-	0,022	50	3,6	11,0	0,0107
SGD(1000 точек), NN	-	0,041	50	7,2	16,2	0,0087
SGD(2000 точек), NN	-	0,079	50	15,4	28,5	0,00719
SGD(500 точек), RBF	-	0,022	50	13,6	20,3	0,00378
SGD(1000 точек), RBF	-	0,042	50	26,7	35,8	0,00356
SGD(2000 точек), RBF	-	0,079	50	60,8	72,4	0,00342

Первый столбец таблицы 4.4 описывает конфигурацию, использованную при проведении каждого отдельного эксперимента. Он показывает, использовался ли для выполнения снижения размерности стохастический градиентный спуск (SGD) или

метод, основанный на разбиении пространства и очередях с приоритетами (PQ). В случаях, когда использовалась интерполяция, первый столбец содержит информацию о количестве (*K*) точек, отображенных с использованием метода SGD, и конкретном методе интерполяции, использованном для отображения оставшихся (*N*-*K*) точек: интерполяции по ближайшему соседу (NN) или радиально базисных функциях (RBF). В случаях, когда интерполяция не использовалась, а алгоритмом SGD или методом PQ отображалось все множество, вместо количества точек указывалось «все множество».

Во втором столбце показано время, затраченное на построение необходимых структур данных для метода PQ. В качестве структуры для разбиения пространства использовались метрические (vp-) деревья. В третьем столбце показано среднее время, затрачиваемое градиентной оптимизационной процедурой на одну итерацию для метода SGD или PQ. В четвертом столбце показано количество точек (узлов) использованных в аппроксимации на каждой итерации процедуры оптимизации. В следующих столбцах показано время, затрачиваемое на интерполяцию и общее затраченное время. В последнем столбце показана оценка качества (ошибка представления многомерных данных в пространстве меньшей размерности). В таблице не приводится время, затрачиваемое на инициализацию точек в пространстве малой размерности, так как оно относительно мало (0,3..0,4 сек. на инициализацию всего набора данных).

Некоторые результаты для наглядности также приведены на рисунке 4.24. Для стохастического градиентного спуска (SGD) показаны результаты для случайных подвыборок по R = 25 и R = 50 элементов. Для метода на основе очередей с приоритетом показаны результаты для 100 и 200 опорных узлов в очереди. Кроме того показаны результаты для алгоритма интерполяции с использованием RBF функций, где интерполяция выполнялась по K=500 и K=1000 точкам, отображенным с использованием SGD.



(а) (б)

Рисунок 4.24 - Экспериментальные результаты для различных способов ускорения: ошибка представления данных в пространстве меньшей размерности (а) и время обработки (б) для гиперспектрального изображения Indian Pines.

Как можно видеть из таблицы и рисунка, конфигурации без использования алгоритмов интерполяции обеспечивают лучшее качество отображения (см. Конечная ошибка). Хотя метод PQ требует построения дополнительных структур, он работает быстрее, чем SGD (см. Время одной итерации и Общее время), и позволяет включить больше узлов (см. Размер подвыборки/ Длина очереди) в аппроксимацию благодаря возможности кэширования предварительно рассчитанных расстояний. В то же время метод PQ и алгоритм SGD обеспечивают сравнимые результаты отображения (см. Конечная ошибка).

Конфигурации с алгоритмами интерполяции работают хуже. Более точная RBF интерполяция позволяет достичь приемлемого качества, но ее вычислительная сложность ограничивает возможности улучшения качества. В то же время методы интерполяции могут быть полезны при обработке больших наборов данных. Кроме того, такие конфигурации позволяют добавлять новые точки данных в существующее решение.

Заметим, что все результаты в таблице относятся к реализации на центральном процессоре в однопоточном режиме для обеспечения возможности сравнения.

4.6 Реализация разработанных алгоритмов формирования информативных признаков на графических процессорах

4.6.1 Параллельная реализация на графических процессорах NVIDIA

К сожалению, формирование информативных признаков для ГСИ большого объема с использованием предложенных в главах 2 и 3 настоящей работы методов и алгоритмов может оказаться медленным даже при использовании описанных выше методов ускорения. В этом случае скорость формирования может быть повышена путем реализации алгоритмов в параллельной вычислительной среде. В этом разделе для ускорения вычислительного процесса используются графические процессоры.

Ввиду того, что детальное рассмотрение архитектурных особенностей современных графических процессоров и аспектов программной реализации выходит за рамки основной линии изложения и носит вспомогательный характер, указанные материалы вынесены в приложение В. В частности, в разделе В.1 представлены общие сведения о технологии CUDA, позволяющие лучше понять содержание настоящего раздела. В разделе В.2 описывается программная реализация алгоритмов формирования информативных признаков с использованием технологии CUDA. При этом рассматривается как общая схема вычислений, так и ключевые аспекты реализации наиболее существенных компонентов. Отдельный подраздел В.3 посвящен вопросам взаимодействия с памятью графического процессора.

На основе анализа, проведённого в п. В.3.2, установлено, что низкая эффективность базовой реализации методов формирования признаков на GPU обусловлена преимущественно неоптимальным паттерном доступа к глобальной памяти при вычислении попарных расстояний в многомерном пространстве. Ключевым фактором снижения производительности является рассредоточенность запрашиваемых компонент векторов в адресном пространстве, что препятствует объединению (coalescing) транзакций памяти и снижает эффективность использования кэш-памяти.

Для устранения указанных ограничений предложен комплекс оптимизационных мер. Комбинированная блочно-столбцовая организация памяти (см. п. В.3.3) предполагает группировку данных по блокам, соответствующим варпу,

с постолбцовым размещением координат внутри блока, что обеспечивает компактное расположение одновременно запрашиваемых компонент и позволяет достичь объединения запросов к глобальной памяти. Дополнительным положительным эффектом выступает пространственная локальность, благодаря которой в кэш загружаются и соседние компоненты.

Асимметричная реализация функции расстояния с раздельным хранением первого аргумента по блочно-столбцовой схеме, а второго — по классической построчной (см. п. В.3.4), оптимизирует доступ к векторам случайной подвыборки за счёт того, что при обращении к конкретной компоненте вектора в кэш попадают смежные компоненты той же точки, а не соседних точек. Однако данный подход требует дополнительного копирования данных на каждой итерации.

Использование разделяемой памяти (см. п. В.3.5) для размещения векторов градиента позволяет сократить число обращений к медленной глобальной памяти устройства. При этом данные организуются таким образом, что каждый поток работает со своим уникальным сегментом массива, что исключает конфликты доступа и необходимость синхронизации. Поскольку размерность выходного пространства существенно ниже входной, вклад данной модификации в общий прирост производительности оказывается умеренным.

Развертка циклов (см. п. В.3.6) сокращает накладные расходы на управление итерациями, позволяет раскрыть константные индексы и смещения на этапе компиляции, способствует более эффективному использованию регистрового файла и упрощает планирование команд, что в совокупности повышает пропускную способность вычислительных конвейеров.

Чтобы оценить влияние предложенных мер на вычислительную эффективность нелинейного метода снижения размерности, был проведен ряд экспериментов с наборами данных, содержащими различное количество точек в пространствах различной размерности. Эксперименты производились над синтетическими данными, объем выборки варьировался от 10 тыс. до 2 млн. точек, размерность менялась от 50 до 1000.

Для каждой конкретной комбинации объема выборки и размерности запускалось несколько модификаций метода формирования признаков. Для каждого запуска выполнялось по 100 итераций соответствующего выбранной модификации

алгоритма градиентного спуска и фиксировалось усредненное время выполнения одной итерации.

Исследовались следующие модификации.

1. Базовая реализация алгоритма, описанная в разделе В.2 и основнная на классической построчной схеме организации памяти для входного массива данных высокой размерности (NORMAL).

2. Модификация с комбинированной блочно-столбцовой схемой организации памяти для входного массива данных высокой размерности, описанная в п. В.3.3 (COMB).

3. Модификация, реализующая комбинированную блочно-столбцовую схему распределения памяти (см. п. В.3.3) также и для выходного массива данных сниженной размерности (COMBY).

4. Модификация, реализующая копирование входных многомерных данных в разделяемую (shared) память (SHARED_X).

5. Модификация, реализующая работу с вектором градиентов в разделяемой памяти, описанная в п. В.3.5 (SHARED_Y).

6. Модификация с комбинированной блочно-столбцовой схемой распределения памяти для входных данных, копированием случайной выборки с построчным распределением данных и несимметричной реализацией функции расстояния, описанная в п. В.3.4 (ASYNC).

7. Модификация, реализованная с применением блочно-столбцовой схемой организации памяти, размещением вектора градиента в разделяемой памяти и разверткой циклов, описанной в п. В.3.6 (SHARED_Y UNWRAP).

Эксперименты проводились на трех различных графических процессорах: NVIDIA GTX 1070, RTX 2070 и RTX 3090. Основные характеристики указанных устройств приведены в таблице В.1 приложения В. Результаты экспериментов показаны на рисунках 4.25-4.27.

Как видно из представленных результатов, все предложенные модификации показали лучшие результаты по сравнению с базовым методом. Наибольший эффект наблюдался от комбинированной блочно-столбцовой схемы организации входного массива (COMB), которая легла в основу и всех остальных исследованных модификаций.

Несколько особняком здесь стоит модификация на основе сохранения входных данных в разделяемую память (SHARED_X). Результаты использования этой модификации представлены лишь на рисунке 4.25. Как видно из рисунка, эта модификация не использовалась при размерностях 5000 и 1000. Причиной этого является зависящее от размерности входного пространства потребление разделяемой памяти. Исходя из объема разделяемой памяти 48кб и 32 потоков на блок, максимальная входная размерность, для которой может быть применена схема с копированием входных данных, составляет (48*1024)/(32*sizeof(FTYPE)) = 384. Учитывая такое ограничение, а также дополнительные временные затраты вследствие необходимости копирования данных из глобальной в разделяемую память, приводящие к достаточно средним показателям, такая модификация в экспериментах на других GPU не исследовалась.

Сравнивая результаты для модификации с комбинированной схемой организации данных в выходном пространстве (COMBY) и модификации с размещением вектора градиента в разделяемой памяти (SHARED_Y), можно отдать предпочтение последней. Модификация SHARED_Y всюду обеспечивает сравнимые или немного лучшие результаты, чем COMBY.

Преимущества модификации с комбинированной блочно-столбцовой схемой организации памяти для входных данных, копированием случайной выборки с построчным размещением данных и несимметричной реализацией функции расстояния (ASYNC) проявляются лишь на высоких размерностях (500, 1000) при работе с более слабыми процессорами (GTX 1070, RTX 2070). Причиной этого, по всей видимости, являются высокие накладные расходы на копирование случайной подвыборки многомерных данных в глобальную память на каждой итерации алгоритма (в остальных модификациях копируются только индексы, а не сами данные).

По этой причине последняя из предложенных модификаций (SHARED_Y UNROLL), основанная на развертывании циклов, построена на базе комбинированной схемы организации данных во входном пространстве и размещении вектора градиента в разделяемой памяти.

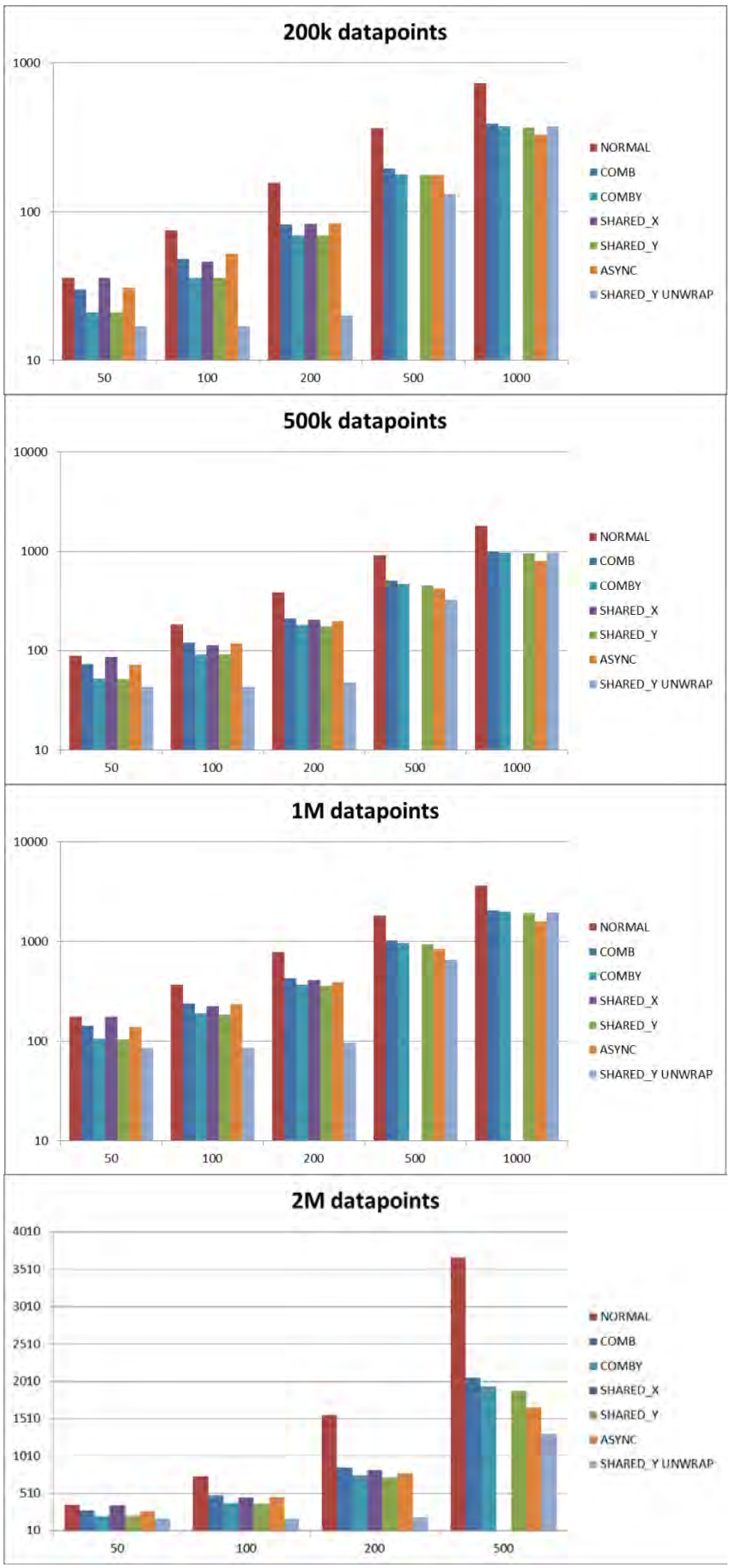


Рисунок 4.25 – Зависимость времени выполнения одной итерации (в миллисекундах) от размерности входного пространства для NVIDIA RTX 2070

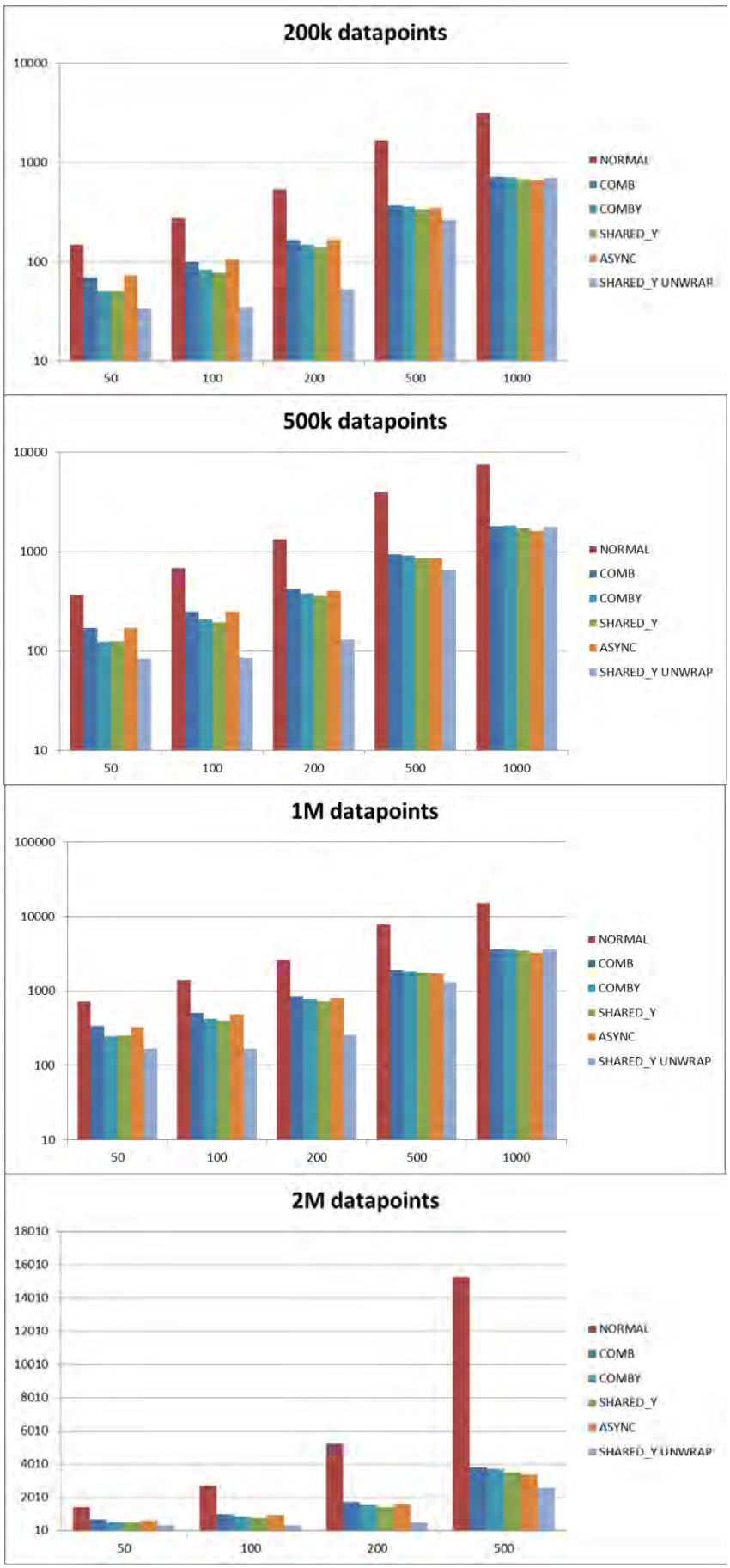


Рисунок 4.26 - Зависимость времени выполнения одной итерации (в миллисекундах) от размерности входного пространства для NVIDIA GTX 1070

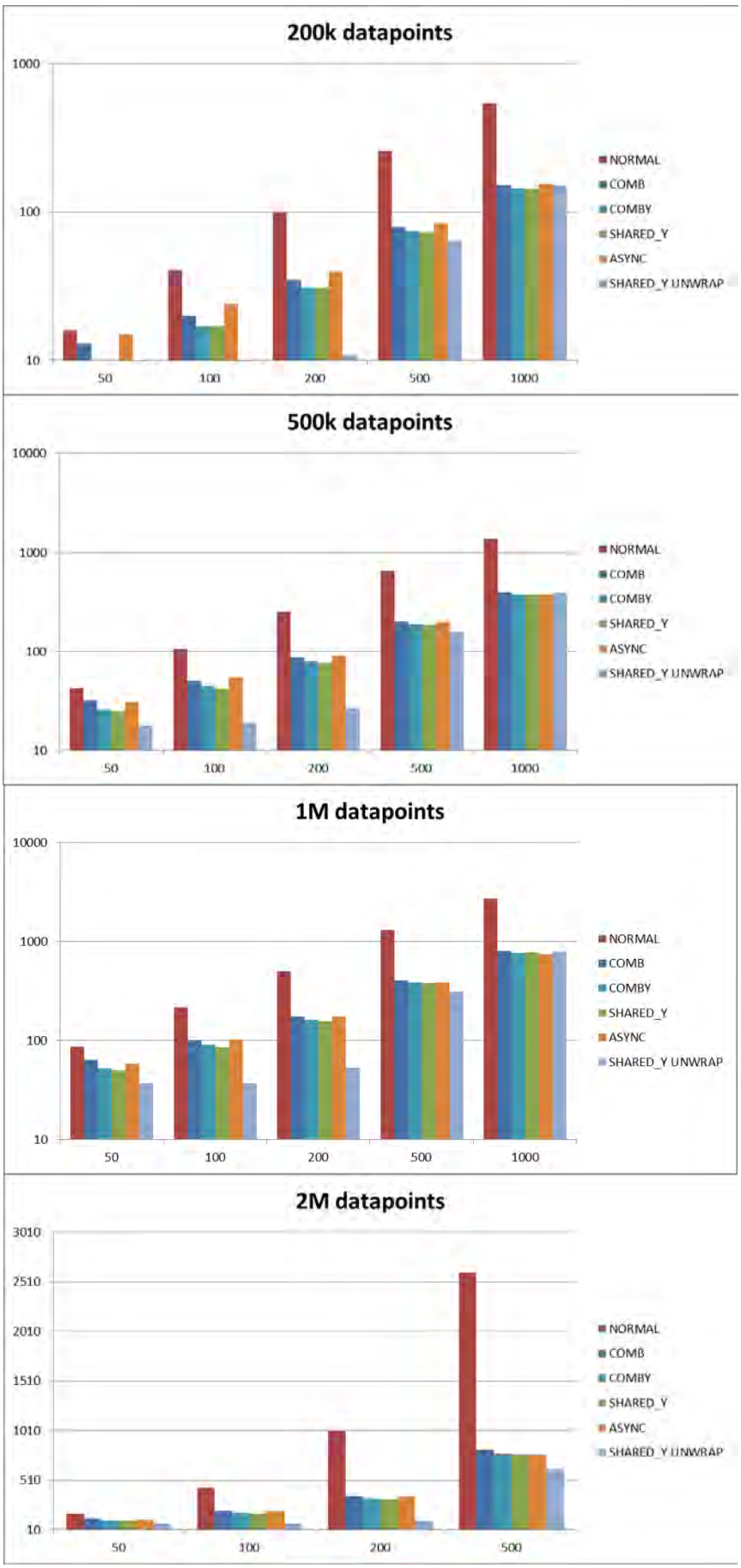


Рисунок 4.27 - Зависимость времени выполнения одной итерации (в миллисекундах) от размерности входного пространства для NVIDIA RTX 3090

Последняя модификация показывает наибольший прирост производительности при размерностях входного пространства 50...500. Следует учитывать, что развертывание циклов может и не быть выполнено компилятором, например, при большом количестве итераций (вследствие высокой размерности входного пространства). Автору не удалось найти конкретных указаний на такие ограничения в спецификациях. В худшем случае, когда циклы не будут развернуты, производительность будет примерно соответствовать модификации SHARED_Y.

В таблице 4.5 показан коэффициент ускорения при использовании GPU, определяемый как отношение времени выполнения одной итерации на графическом процессоре ко времени выполнения итерации на CPU, работающем в однопоточном режиме.

Таблица 4.5 – Коэффициент ускорения при использовании GPU (NVIDIA GTX 2070 / RTX 3090) относительно CPU (Ryzen 5 5600) в однопоточном режиме (в скобках показано время выполнения одной итерации на CPU в сек.).

Число точек	**Размерность**				
	1000	**500**	**200**	**100**	**50**
50000	68,1/144,9 (5,7)	82,2/169,3 (2,9)	243,6/304,5 (1,2)	134,4/336 (0,7)	79,4/198,5 (0,4)
100000	63,4/152,6 (11,3)	95/196 (6,3)	243,1/486,2 (2,4)	149,2/335,8 (1,3)	87,8/197,5 (0,8)
200000	65,8/163,3 (24,7)	94,6/192,2 (12,5)	264,1/480,1 (5,3)	158,1/383,9 (2,7)	93/225,9 (1,6)
500000	63,5/158 (61,6)	95,3/199,8 (31,4)	275,9/490,4 (13,2)	168,6/381,6 (7,3)	91,9/219,6 (4,0)
1000000	62,8/155,5 (123,2)	95,8/201 (62,7)	279,6/506,5 (26,8)	168,3/391,2 (14,5)	99,5/228,7 (8,5)
2000000	-	109,1/229,4 (142,9)	276,3/512,3 (52,8)	170,6/391,9 (29,0)	99,6/231,9 (16,9)
5000000	-	-	310,7/596,7 (149,8)	171,1/406,3 (72,7)	99,6/235,9 (42,2)

В качестве GPU использовались NVIDIA GTX 2070 и RTX 3090 с модификацией алгоритма с применением блочно-столбцовой схемы организации памяти, размещением вектора градиента в разделяемой памяти и разверткой циклов (SHARED_Y UNWRAP). Соответствующие коэффициенты записаны в таблице через слэш. Кроме того, в таблице в скобках приведено время выполения одной итерации на центральном процессоре в секундах.

Как видно из таблицы, использование GPU позволило заметно ускорить вычисления. При использовании NVIDIA GTX 2070 коэффициент ускорения находился приблизительно в диапазоне от 63 до 311, а при использовании RTX 3090 – в диапазоне от 145 до 597. Таким образом, применение предложеной параллельной программной реализации позволило значительно ускорить процесс формирования признаков.

4.6.2 Параллельная реализация на альтернативных архитектурах

Помимо описанной в настоящем разделе реализации алгоритмов с использованием технологии CUDA для графических процессоров NVIDIA, была также выполнена [29*, 30*, 333*] реализация для графических процессоров AMD, а также многопоточная реализация для центральных процессоров.

Реализация методов снижения размерности для графических процессоров AMD выполнялась с использованием HIP [282] (Heterogeneous-Compute Interface for Portability, англ. «Гетерогенный вычислительный интерфейс для переносимости») – диалекта языка C++ разработанного для написания программ для графических процессоров. Помимо собственно языка, HIP включает средства миграции программ, написанных для CUDA, что позволяет создавать исполняемый код как для графических процессоров AMD, так и NVIDIA. По сути, при программировании для GPU AMD, HIP предоставляет разработчику слой абстракции, который позволяет использовать библиотеки ROCm более низкого уровня и выполнять компиляцию с использованием компилятора HCC.

Учитывая, что программная модель и терминология, используемая HIP, в значительной степени схожа с таковой для NVIDIA CUDA, нет необходимости детально описывать здесь полученное для GPU AMD решение.

Многопоточная реализация для центрального процессора была выполнена на языке C++ с использованием OpenMP. При этом компилятором выполнялась автоматическая векторизация циклов, и использовались инструкции AVX.

Некоторые результаты сравнения реализаций с использованием HIP и OpenMP приведены на рисунке 4.28. На графике приведены результаты для четырех вариантов параллельной реализации с использованием HIP (V1…V4), каждая из которых запускалась с тремя различными конфигурациями сетки потоков: 16, 32 и 64 потоков на блок. Сами реализации различались, во-первых, организацией матрицы координат по строкам (варианты V1 и V2) и по столбцам (варианты V3 и V4), а во-вторых, учетом аксиомы симметричности ($d(x,y)=d(y,x)$) при использовании метрик в качестве функций расстояния. При этом в вариантах V1 и V3 симметричность не учитывалась, в вариантах V2 и V4 – учитывалась.

Вариант CPU соответствует многопоточной реализации на центральном процессоре для 16 потоков.

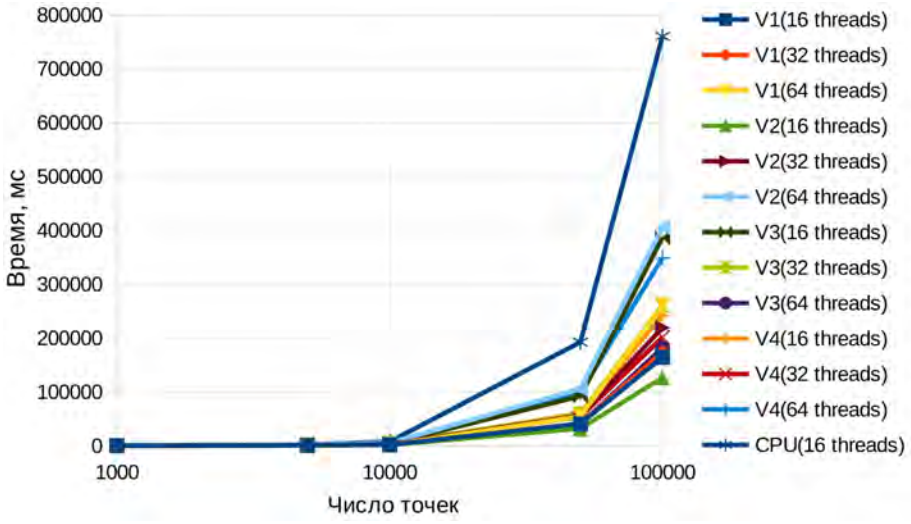


Рисунок 4.28 - Результаты исследования реализации на HIP для графических процессоров AMD и реализации с использованием OpenMP для CPU.

Эксперименты выполнялись на графическом процессоре AMD Radeon RX Vega 56, содержащем 3584 потоковых процессора и 8 Гб оперативной памяти. В качестве центрального процессора использовался 16 поточный, 8 ядерный процессор AMD Ryzen 7 3700X с базовой частотой 3,6 ГГц. В качестве операционной системы использовалась Ubuntu 18.04 LTS.

Как видно из представленных результатов, использование графического процессора позволяет существенно ускорить решение по сравнению с многопоточной реализацией для центрального процессора. В частности, версия V2 для GPU оказалась более чем в 6 раз быстрее версии для CPU, работающей в 16 поточном режиме.

Необходимо отметить, что представленные в настоящем пункте численные результаты получены при использовании классического алгоритма градиентного спуска, в то время как результаты, описанные ранее в п. 4.6.1, получены для стохастической версии алгоритма и не могут сравниваться напрямую.

Выводы и результаты по главе 4

В четвертой главе выполнен анализ различных подходов к ускорению предложенных в предыдущих главах алгоритмов и методов формирования информативных признаков гиперспектральных изображений. Рассмотрен ряд алгоритмов стохастического градиентного спуска, включая метод моментов и метод адаптивного градиента (AdaGrad).

Исследован подход к ускорению вычислений на основе иерархического разбиения многомерного пространства с последующей аппроксимацией градиента, выполняемой по узлам структуры разбиения. В частности, предложены:

- алгоритм построения списка опорных узлов и численный метод ускорения формирования признаков на основе опорных узлов, в котором построение структуры разбиения осуществляется во входном многомерном пространстве;

- численный метод ускорения формирования признаков на основе иерархического разбиения формируемого (выходного) пространства малой размерности;

- модификация метода на основе опорных узлов, учитывающая характеристики узлов как во входном, так и в формируемом пространствах;

- алгоритм построения очереди опорных узлов с приоритетом и соответствующая модификация метода ускорения формирования признаков на основе очередей с приоритетом.

Проведено исследование нескольких структур для декомпозиции пространства, а именно: метрических деревьев (vp-деревьев), kd-деревьев и деревьев кластеров. Показано, что для представленных наборов данных метрические деревья (vp-деревья)

и деревья кластеров могут эффективно применяться для разбиения входного пространства.

Показано, что разбиение входного пространства более предпочтительно, чем разбиение выходного пространства. Модификация метода на основе опорных узлов, учитывающая характеристики узлов как во входном, так и в формируемом пространствах, позволяет достигать более точных приближений с более высокими вычислительными затратами, но требует тщательного выбора параметров. Влияние этого метода на окончательную ошибку представления данных незначительно.

Выполнен анализ различных алгоритмов многомерной интерполяции, включая интерполяцию по ближайшему соседу, алгоритм обратных взвешенных расстояний, интерполяцию с использованием радиально-базисных функций и интерполяцию путем минимизации ошибки представления данных в пространстве меньшей размерности. Показано, что интерполяция радиальными базисными функциями и интерполяция путем минимизации ошибки представления данных в пространстве меньшей размерности обеспечивают лучшие качественные результаты.

Предложен комбинированный метод ускорения формирования признаков на основе стохастического градиентного спуска, очередей опорных узлов с приоритетом и алгоритмов многомерной интерполяции. Даны рекомендации по применению метода. Использование разработанных методов в целом сделало возможным формирование признаков относительно больших наборов данных за приемлемое время даже без применения параллельных вычислительных сред.

Разработаны программные средства формирования признаков ГСИ с использованием параллельной многопоточной реализации для современных графических процессоров NVIDIA и AMD.

Предложены и исследованы несколько модификаций, основанных на комбинированной блочно-столбцовой схеме организации памяти для хранения на GPU исходных данных; копировании случайной подвыборки с построчным распределением данных на GPU и несимметричной реализации функции расстояния; использовании разделяемой памяти для хранения исходных и формируемых данных на GPU и развертывании циклов.

Показано, что наибольший эффект наблюдался от применения комбинированной блочно-столбцовой схемы организации памяти, а наибольшую производительность при размерностях входного пространства 50...500 показала модификация, построенная на базе комбинированной схемы организации памяти, размещении вектора градиента в разделяемой памяти GPU и использующая развертывание циклов. Использование разработанных параллельных реализаций позволило достигнуть ускорения от 145 до 597 раз по сравнению с однопоточной версией, работающей на ЦП.

Выполнена альтернативная реализация с использованием HIP для графических процессоров AMD и многопоточная реализация для ЦП с использованием OpenMP. Показано, что реализация для GPU AMD оказалась более чем в 6 раз быстрее версии для ЦП, работающей в 16 поточном режиме.

5 Примеры использования методов формирования информативных признаков гиперспектральных изображений

5.1 Классификация гиперспектральных данных дистанционного зондирования высокого разрешения, полученных с беспилотного летательного аппарата

Наиболее широко гиперспектральная съемка используется на сегодняшний день в приложениях дистанционного зондирования Земли (ДЗЗ). В настоящей диссертационной работе ранее уже описывалась и решалась важнейшая задача тематической классификации данных ДЗЗ. При решении этой задачи использовались данные низкого разрешения (см. п. 2.4.1).

Однако развитие как беспилотных летательных аппаратов, так и компактных средств регистрации гиперспектральных данных, позволяет на сегодняшний день получать гиперспектральные данные значительно более высокого пространственного разрешения. В представленном здесь исследовании информативные признаки ГСИ высокого пространственного разрешения формировались в соответствии с предложенной в разделе 3.4 схемой слияния признаков, использующей как спектральные признаки, так и локальные пространственные признаки, вычисленные с помощью сверточной нейронной сети.

При проведении экспериментов использовался новый открытый набор ГСИ высокого разрешения Ухань WHU-Hi [330], полученных с помощью беспилотного летательного аппарата. Он содержит три гиперспектральных изображения, а также соответствующие маски истинной классификации. Изображения были получены с помощью датчика Nano-Hyperspec, имеют высокое пространственное и спектральное разрешение. Краткое описание сцен приведено в таблице 5.1.

В первом эксперименте исследовалось, как на качество классификации влияет размерность *dim* формируемого пространства признаков, а также используемая в спектральном пространстве функция расстояния и коэффициент слияния α локальной пространственной и спектральной информации. Функция расстояния в пространстве сформированных нейросетью признаков не менялась; в качестве такой функции

использовалось евклидово расстояние. Конфигурация метода при этом обозначалась как Ed+Ed, Ed+Sam или Ed+Hlg в зависимости от используемой в спектральном пространстве функции расстояния (евклидово расстояние, спектральный угол или расстояние Хеллингера).

В этом эксперименте использовался классификатор по ближайшему соседу (NN), поскольку он не имеет настраиваемых параметров и позволяет проводить объективное сравнение различных конфигураций. В качестве показателя качества использовалась общая точность классификации *Acc*. Во всех экспериментах использовалось стандартное разбиение набора данных; в качестве обучающего набора бралось по 100 точек на класс. Графики зависимости качества классификации от указанных параметров показаны на рисунках 5.1 и 5.2 для всех трех сцен используемого набора данных.

Таблица 5.1 – Набор данных WHU-Hi

Сцена	**Размер (HxW)**	**Каналы**	**Пространственное разрешение**	**Классы**
WHU-Hi-LongKou	550×400	270	0,463 m	9
WHU-Hi-HanChuan	1217×303	274	0,109 m	16
WHU-Hi-HongHu	940×475	270	0,043 m	22

На рисунке 5.1 показано, как качество классификации меняется с ростом размерности выходных данных *dim*. Мы видим ожидаемый рост качества классификации с ростом размерности. Однако после значения 30 кривые стабилизируются, и качество меняется очень незначительно. По этой причине дальнейшие исследования проводились при значении размерности *dim*=30.

Как видно из рисунка 5.2, пространственные признаки играют важную роль, так как качество классификации с их использованием (при коэффициенте α, равном нулю) существенно выше, чем для спектральных признаков (при коэффициенте α, равном единице). Слияние по предложенной в разделе 3.4 схеме позволило существенно улучшить качество классификации для всех трех сцен. Среди функций

расстояния при использовании классификатора ближайшего соседа предпочтение можно отдать дивергенции Хеллингера.

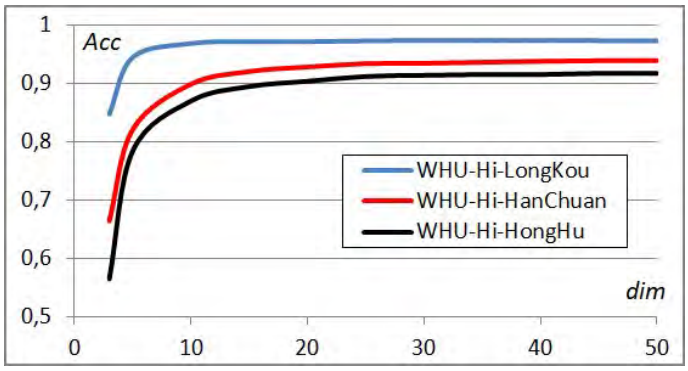
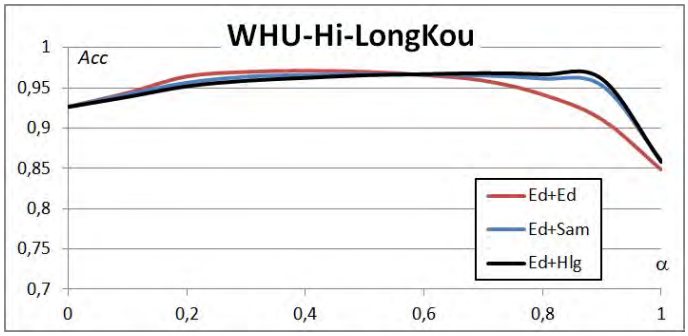
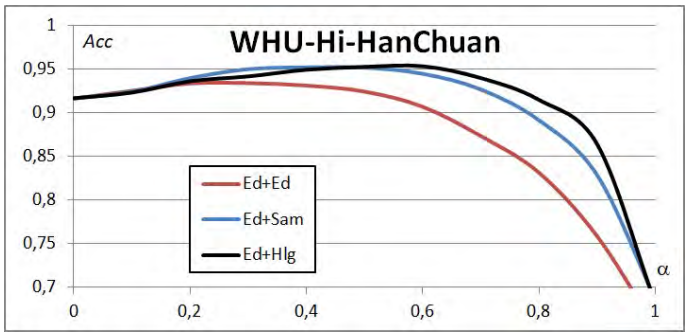


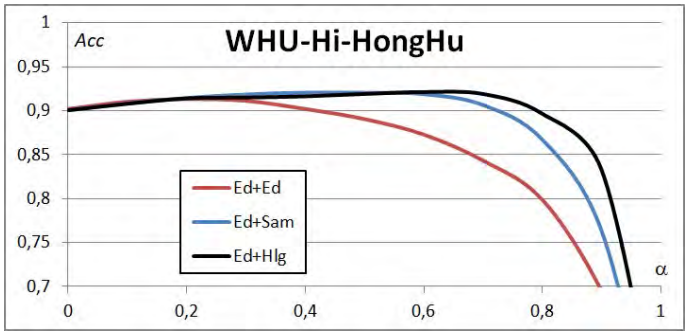
Рисунок 5.1 - Зависимость точности классификации *Acc* от размерности *dim*



(а)



(б)



(в)

Рисунок 5.2 - Зависимость точности классификации *Acc* от коэффициента слияния α для трех сцен набора данных Ухань WHU-Hi: LongKou (а), HanChuan (б), HongHu (в)

Сцена WHU-Hi-LongKou

Для улучшения качества классификации с использованием сгенерированного представления рассматривался вопрос использования многослойного персептрона вместо простого NN-классификатора. В частности, рассматривался персептрон с 1, 2 и 3 скрытыми слоями; ширина слоя варьировалась от 10 до 1000 нейронов. Сеть обучалась на протяжении максимум 200 эпох с использованием оптимизатора Adam. В качестве функция активации использовалась ReLU. Входными данными сети служили результаты предложенного в разделе 3.4 метода слияния признаков. При этом спектральное рассогласование измерялось с использованием дивергенции Хеллингера, а размерность формируемого пространства признаков была равна 30. Коэффициент слияния был установлен в предыдущем эксперименте. Результаты экспериментального исследования показаны в Таблице 5.2.

Как можно видеть, увеличение количества нейронов в слое, как правило, приводит к повышению качества классификации, и выбор двух скрытых слоев представляется достаточным. Таким образом, с учетом принятых в экспериментах ограничений, наилучшим представляется использование двух скрытых слоев по 1000 нейронов каждый.

Таблица 5.2 – Точность классификации (%) для многослойного персептрона на основе признаков, сформированных предложенным в разделе 3.4 методом в конфигурации Ed+Hlg (сцена WHU-Hi-LongKou).

Скрытых слоев	**Ширина слоя**						
	10	**20**	**50**	**100**	**200**	**500**	**1000**
1	84,8	88,5	95,5	97,1	97,6	98,2	98,4
2	82,7	96,5	97,6	98,3	98,4	98,5	**98,6**
3	92,7	95,6	97,2	98,1	98,5	98,4	98,2

В таблице 5.3 качество классификации, полученное с использованием предложенного метода, сравнивается с результатами работы других методов классификации гиперспектральных данных. Значения точности классификации для альтернативных методов взяты из [330] и были рассчитаны с использованием точно такого же набора данных с тем же разделением обучающей и тестовой выборок.

Альтернативными методами являются [330]: спектрально-пространственная сеть внимания (spectral-spatial attention network - SSAN) [174], машина опорных векторов (Support Vector Machine - SVM), спектральная пространственная полностью сверточная сеть (Spectral Spatial Fully Convolutional Network - SSFCN) [313], классификатор на основе условных случайных полей опорных векторов с ограничением расстояния Махаланобиса (Support Vector conditional Random Fields classifier with a Mahalanobis distance boundary Constraint - SVRFMC) [329], подход на основе эволюции фрактальной объектно-ориентированной сети (Object Oriented Fractal Net Evolution Approach - FNEA-OO), глубокие пирамидальные остаточные сети (deep Pyramidal Residual Networks - pResNet) [231], фреймворк классификации CNN и CRF (CNN and CRF classification framework - CNNCRF) [330], спектрально-пространственная остаточная сеть (spectral-spatial residual network - SSRN) [331] и фреймворк быстрого глобального обучения без патчей (Fast Patch-Free Global Learning Framework - FPGA) [328].

Как видно из представленных результатов, сформированные с использованием предложенного в разделе 3.4 метода информативные признаки позволили получить хорошее качество классификации, сопоставимое с текущим уровнем результатов и уступающее лучшему результату всего на 0,6%.

Таблица 5.3 – Сравнение с альтернативными методами классификации (сцена WHU-Hi-LongKou). Методы расположены в порядке возрастания качества.

Метод	**SSAN**	**SSFCN**	**SVM**	**SVRFMC**	**FNEA-OO**
Точность, %	94,4	94,6	95,0	98,4	98,6

Метод	***Предложенный***	**pResNet**	**CNNCRF**	**SSRN**	**FPGA**
Точность, %	*98,6*	98,7	98,9	99,0	99,2

Сцена WHU-Hi HanChuan

Аналогичные эксперименты были проведены для других двух сцен из набора данных. Результаты исследования MLP для сцены WHU-Hi- HanChuan показаны в Таблице 5.4. В данном случае для измерения рассогласования в спектральном пространстве использовался спектральный угол. Как видно из приведенной таблицы,

наилучшие результаты достигались, как и на предыдущей сцене, двумя скрытыми слоями по 1000 нейронов.

Таблица 5.4 – Точность классификации (%) для многослойного персептрона на основе признаков, сформированных предложенным в разделе 3.4 методом в конфигурации Ed+Sam (сцена WHU-Hi- HanChuan).

Скрытых слоев	**Ширина слоя**						
	10	**20**	**50**	**100**	**200**	**500**	**1000**
1	76,9	80,5	88,4	90,8	92,5	93,7	94,5
2	76,1	87,0	92,4	93,5	94,3	95,0	**95,4**
3	86,5	92,3	93,5	94,7	94,7	94,8	86,5

Сравнение полученных результатов с результатами альтернативных методов из работы [330] приведено в Таблице 5.5. Как следует из приведенной таблицы, предложенный в разделе 3.4 метод оказался вторым по качеству методом среди десяти рассмотренных, уступив лишь методу FPGA.

Таблица 5.5 – Сравнение с альтернативными методами классификации (сцена WHU-Hi- HanChuan). Методы расположены в порядке возрастания качества

Метод	**SVM**	**FNEA-OO**	**SVRFMC**	**SSAN**	**SSFCN**
Точность, %	77,6	85,6	86,5	88,6	89,8

Метод	**SSRN**	**pResNet**	**CNNCRF**	***Предложенный***	**FPGA**
Точность, %	89,8	93,3	93,9	*95,4*	97,8

Сцена WHU-Hi HongHu

При исследовании сцены WHU-Hi- HongHu с использованием MLP в качестве функции расстояния в спектральном пространстве также использовался спектральный угол. На этой сцене наилучшее качество было получено с использованием той же конфигурации сети, что и в предыдущих случаях (см. Таблицу 5.6). Само значение качества оказывается при этом ниже, что говорит о сложности классификации на указанной сцене.

Таблица 5.6 – Точность классификации (%) для многослойного персептрона на основе признаков, сформированных предложенным в разделе 3.4 методом в конфигурации Ed+Sam (сцена WHU-Hi- HongHu)

Скрытых слоев	**Ширина слоя**						
	10	**20**	**50**	**100**	**200**	**500**	**1000**
1	79,5	87,0	90,2	91,2	91,7	91,9	91,8
2	71,3	86,5	90,6	91,6	92,2	92,4	**92,5**
3	79,1	85,1	89,6	90,6	91,4	91,6	89,2

При сравнении с результатами других современных методов, метод показывает сопоставимое качество, занимая среднюю позицию (см. Таблицу 5.7). В частности, метод уступает четырем методам (CNNCRF, SSFCN, pResNet и FPGA), опережая по качеству пять других альтернативных методов.

Таблица 5.7 – Сравнение с альтернативными методами классификации (сцена WHU-Hi- HongHu). Методы расположены в порядке возрастания качества

Метод	**SVM**	**SSAN**	**FNEA-OO**	**SVRFMC**	**SSRN**
Точность, %	73,6	87,3	88,8	89,9	91,3

Метод	***Предложенный***	**CNNCRF**	**SSFCN**	**pResNet**	**FPGA**
Точность, %	92,5	93,7	94,3	95,3	97,5

Стоит отметить, что увеличение количества нейронов ведет к дальнейшему повышению качества. Так, двуслойная сеть с шириной 5000 достигает 92,8%, а аналогичная сеть с шириной 10000 нейронов – 93,1%. Таким образом, потенциал сформированных информативных признаков оказывается неисчерпанным, несмотря на малую размерность (30) формируемого предложенным методом пространства по сравнению с размерностью исходных пространств (270 + 512).

В заключение отметим, что для формирования информативных признаков не потребовалось производить обучение или донастройку нейронной сети. Достаточно выбрать размерность формируемого пространства признаков и коэффициент слияния.

Достигаемое при этом с помощью обычного MLP классификатора качество вполне соответствует современному уровню исследований в рассматриваемой области.

5.2 Классификация гиперспектральных данных дорожной обстановки

Создание средств автономной навигации в условиях сложной дорожной обстановки, а также плохо структурируемой среды и бездорожья является актуальной научно-технической проблемой. Использование только данных видеокамер, регистрирующих цветной видеопоток, может быть недостаточно. По этой причине в системы автономной навигации могут включаться датчики иных типов.

Одним из возможных перспективных направлений развития таких систем регистрации является использование гиперспектральных видеокамер. Использование таких камер позволит с большей точностью определять растительный покров, дорожные покрытия, классифицировать наблюдаемые объекты по сравнению с обычными видеокамерами.

Для подобных целей необходимо использовать видеокамеры, позволяющие регистрировать гиперспектральные изображения с относительно высокой скоростью.

Попытка создания такой системы регистрация предпринята в работе [311]. В рамках этого проекта на транспортное средство были установлены две гиперспектральные камеры, выполняющие съемку в видимом и инфракрасном диапазонах.

В качестве гиперспектрального сенсора видимого диапазона использовалась версия камеры MQ022HG-IM-SM4X4-VIS [122] производства фирмы Ximea. Камера оборудована чипом IMEC [87] с использованием мозаичного фильтра кадров. Активный диапазон в видимой области спектра составляет: 470–630 нм, пространственное разрешение: 512 × 272 пикселей, количество спектральных диапазонов: 16.

Для гиперспектральной съемки в ближнем инфракрасном диапазоне использовалась версия камеры MQ022HG-IM-SM5X5-NIR [121] того же производителя со следующими характеристиками: активный диапазон: 600–975 нм с разрешением 409 × 217 пикселей, количество спектральных диапазонов: 25.

Обе гиперспектральные камеры построены на основе сенсора с мозаичной архитектурой (SSM). Оптическая система обеих камер включала объективы LM5JC10M от Kowa с фокусным расстоянием 5.0 мм, обладающий оптическим разрешением не менее 160 пар линий/мм, с покрытием, обеспечивающим пропускание широкого диапазона длин волн 380–1000 нм.

Гиперспектральные данные извлекались с камер каждые четыре секунды движения. Полученные с гиперспектральных камер данные проходили предварительную обработку и отбор. Отобранные данные были аннотированы вручную.

Данные разбиты на два набора, соответствующих различным датам съемки. В более ранней версии набора данных Hyko-1 [311] представлено по 234 гиперспектральных кадра для каждого из диапазонов. В более поздней Hyko-2 представлено 162 аннотированных гиперспектральных кадра, зарегистрированных в видимом диапазоне, и 77 - в ближнем инфракрасном диапазоне. Примеры кадров, зарегистрированных в видимом диапазоне, представлены на рисунке 5.3.

В целях представленного в настоящем подразделе исследования использовалась аннотация на основе следующих пяти семантических классов: Undefined (класс не определен), Drivable (асфальтовые и грунтовые дороги, тротуары, дорожная разметка), Not Drivable (растительность), Obstacle (знаки, светофоры, здания, транспортные средства, пешеходы).

Рисунок 5.3 - Примеры кадров дорожной обстановки из набора данных [311] (представлена первая главная компонента, изображения контрастированы): версия набора Hyko-1 (вверху), Hyko2 (внизу), видимый диапазон.

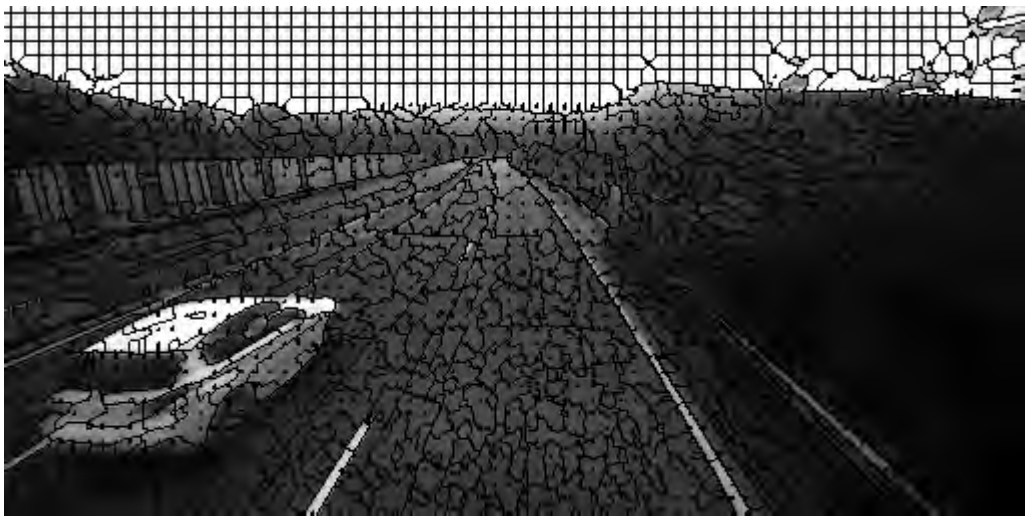
5.2.1 Анализ гиперспектральных данных видимого диапазона

Ввиду того, что обработка гиперспектральных видеоданных является значительной нагрузкой на вычислительные средства, важным этапом является снижение объема обрабатываемых данных. Одним из способов снижения объема, примененных в работе [312], является переход от исходного гиперспектрального изображения к набору мелких, компактных в пространственной области сегментов — суперпикселей.

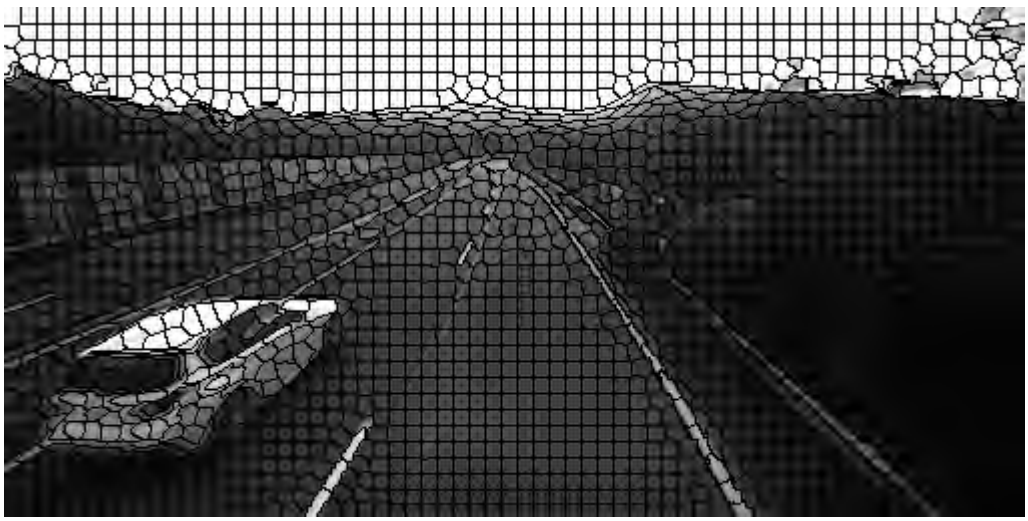
Для получения набора суперпикселей в настоящей работе испытывались как методы сегментации на основе многомерного или векторного градиента [287], так и методы, предназначенные для работы с цветными изображениями, например, [77, 300, 2].

Приведем несколько примеров такой сегментации. На рисунке 5.4 приведен пример цветного морфологического градиента; сегментированное изображение, полученное с использованием преобразования водораздела над указанным градиентом; сегментация на суперпиксели с использованием метода SLIC.

(а)



(б)



(в)

Рисунок 5.4 - Примеры сегментации на суперпиксели:
изображение цветного морфологического градиента (а);
результат преобразования водораздела над изображением градиента (б);
результаты сегментации с использованием метода SLIC (в)

По результатам предварительных испытаний, был выбран метод SLIC [2]. Отметим, что такой же выбор был сделан в работе [312] .

После выполнения сегментации для каждого выделенного сегмента осуществляется усреднение спектральных характеристик по всем входящим в сегмент пикселям изображения. Кроме того, рассчитываются текстурные признаки (далее называемые «Признаки 1») — усредненные по различным масштабам средние значения яркости, локального градиента и собственные значения матрицы Гессиана. В качестве альтернативы использовались текстурные признаки, основанные на матрице совместной встречаемости пикселей (co-occurence matrix), именуемые далее «Признаки 2». Для обоих вариантов текстурных признаков выполнялась нормализация по стандартным отклонениям.

Основной целью исследования стало определение наилучшей функции расстояния между суперпикселями кадров гиперспектральной съемки и определение параметров метода слияния спектральных и текстурных признаков, изложенного в подразделе 3.3.

В первой серии экспериментов решалась задача подбора функции расстояния. Для этого по методу подраздела 3.3 выполнялось слияние спектральных признаков суперпикселей с текстурными признаками (Признаки 1). Для промежуточного представления данных выбиралось пространство с евклидовой метрикой. Размерность формируемого пространства варьировалась от 2 до 20. Для каждой выходной размерности осуществлялось обучение и тестирование NN-классификатора на основе евклидова расстояния. Для обучения использовалось 30% однородных с точки зрения семантических классов суперпикселей. Тестирование выполнялось на оставшихся 70% выборки.

Результаты первой серии экспериментов показаны на рисунке 5.5. Различные диаграммы соответствуют разным выходным размерностям: 3, 5, 10, 15. Дальнейшее повышение размерности не оказывало заметного влияния на характер диаграмм. По этой причине диаграммы для больших размерностей не приводятся.

В каждой диаграмме различными цветами показаны результаты, соответствующие трем различным функциям расстояния в спектральном пространстве: евклидову расстоянию (Ed), спектральному углу (Sam), дивергенции

Хеллингера (Hlg). Для измерения рассогласования текстурных признаков во всех случаях использовалось евклидово расстояние. Коэффициент слияния признаков варьировался в диапазоне 0...1 с шагом 0,1; выполнялось обучение и тестирование классификатора. Для оценки качества использовалась общая точность классификации, рассчитываемая как доля верно классифицированных суперпикселей из общего объема тестовой выборки.

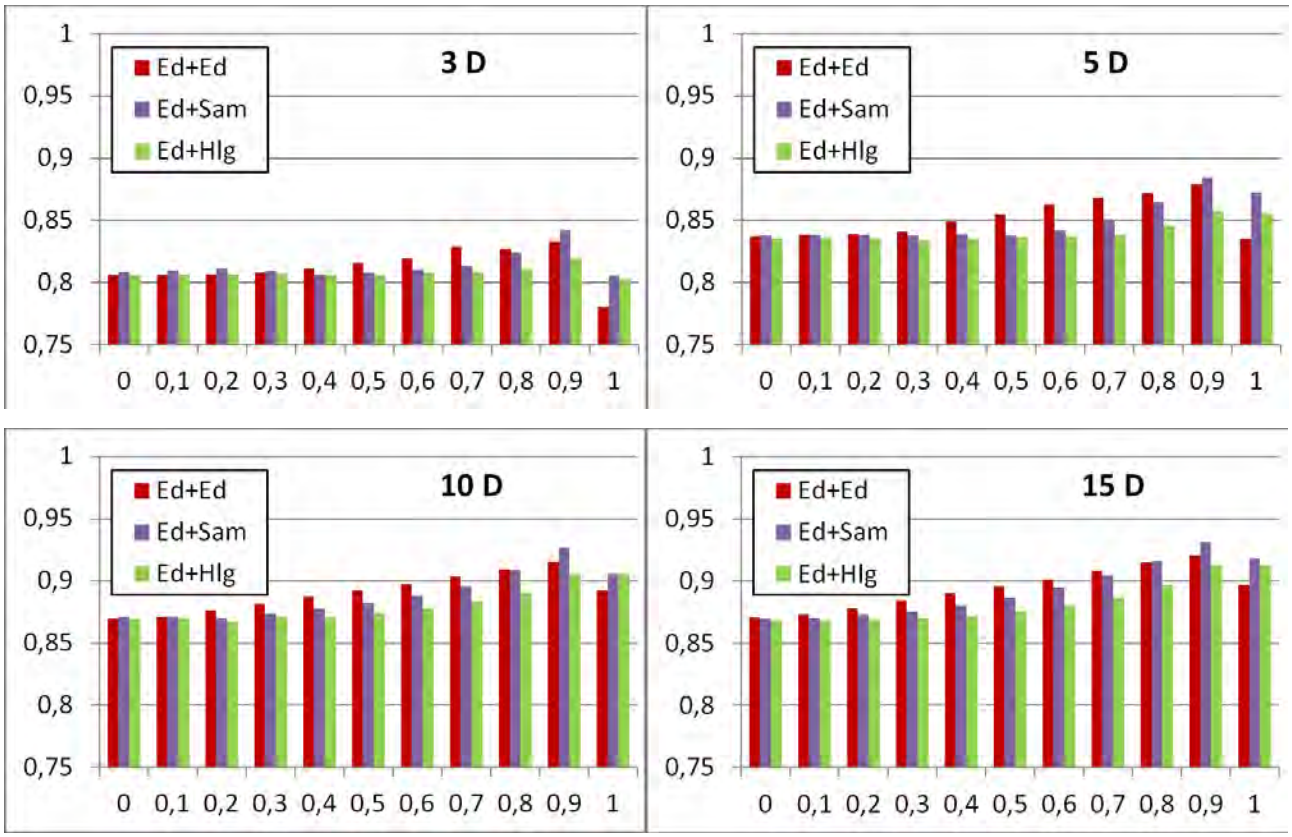


Рисунок 5.5 - Зависимость точности *Acc* классификации суперпикселей от коэффициента λ слияния текстурных и спектральных признаков для различных функций расстояния и размерностей выходного пространства (набор данных Hyko-2, видимый диапазон).

Как видно из проведенных экспериментов, во всех случаях слияние признаков позволяло улучшить показатель качества классификации. Далее рост размерности выходного пространства оказывал заметное положительное влияние вплоть до D=15, причем более существенный рост качества наблюдался для спектральных признаков (при коэффициенте слияния, равном 1). Наилучшие результаты для всех рассмотренных выходных размерностей были достигнуты при использовании

спектрального угла (Sam) в качестве функции расстояния в спектральном пространстве.

Во второй серии экспериментов выполнялось сравнение двух различных систем текстурных признаков (Признаки 1 и Признаки 2) при фиксированных функциях расстояния в пространстве признаков текстур (Ed) и спектральном пространстве (Sam). Результаты сравнения показаны на рисунке 5.6.

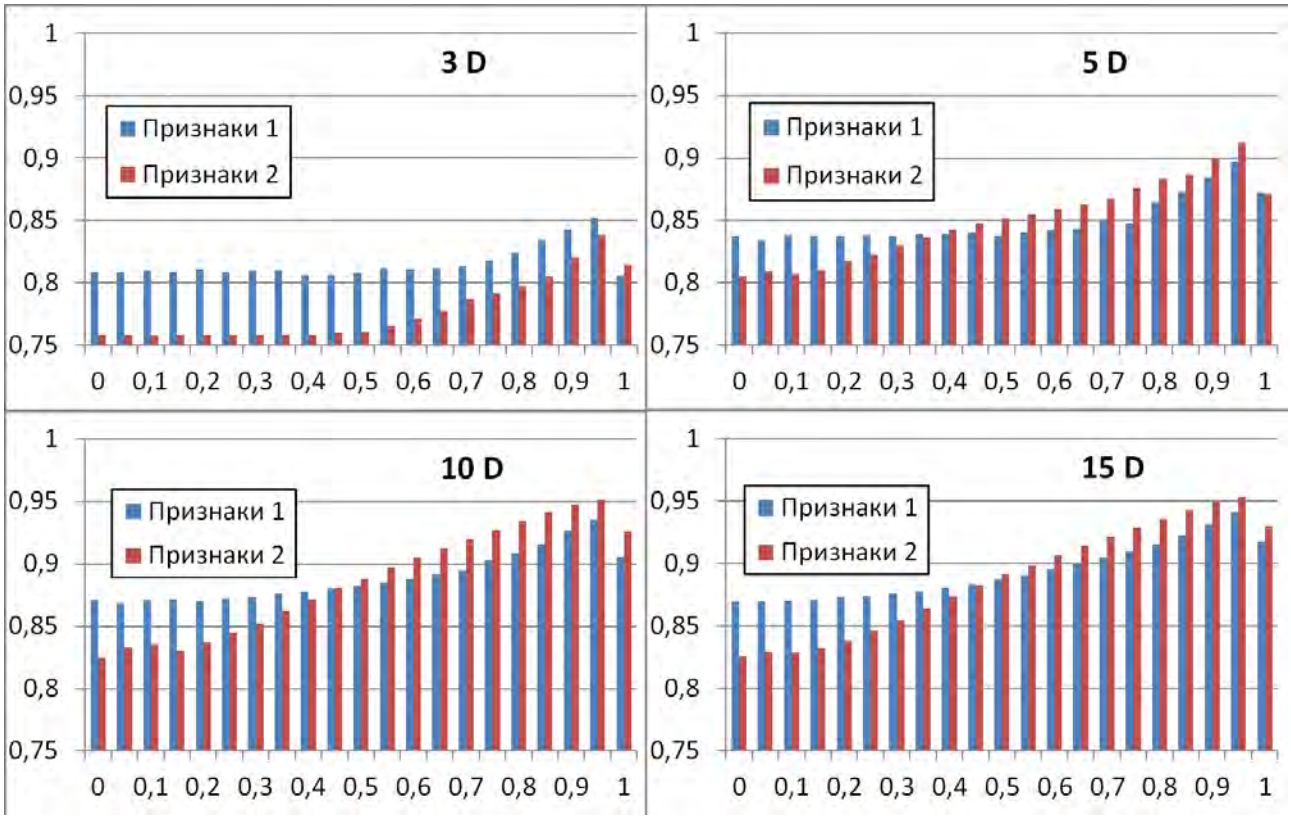


Рисунок 5.6 - Зависимость точности классификации суперпикселей *Acc* от коэффициента слияния λ текстурных и спектральных признаков для различных текстурных признаков и размерностей формируемого пространства (набор данных Hyko-2, видимый диапазон).

Как видно из приведенных диаграмм, несмотря на то, что «Признаки 1» при коэффициенте слияния, равном нулю (унимодальном использовании), оказывались более информативными, с ростом этого коэффициента они начинали проигрывать (при D>=5) альтернативным текстурным признакам. Наилучшие значения достигались при использовании признаков «Признаки 2» и коэффициенте слияния, равном 0.95 для размерностей D=10, 15.

Кроме описанных экспериментов, были выполнены аналогичные эксперименты для первого варианта исследуемого набора данных Hyko-1. В этом

эксперименте в качестве текстурных признаков использовались только текстурные признаки на основе матриц совместной встречаемости пикселей «Признаки 2» и спектральные признаки. Результаты эксперимента показаны на рисунке 5.7.

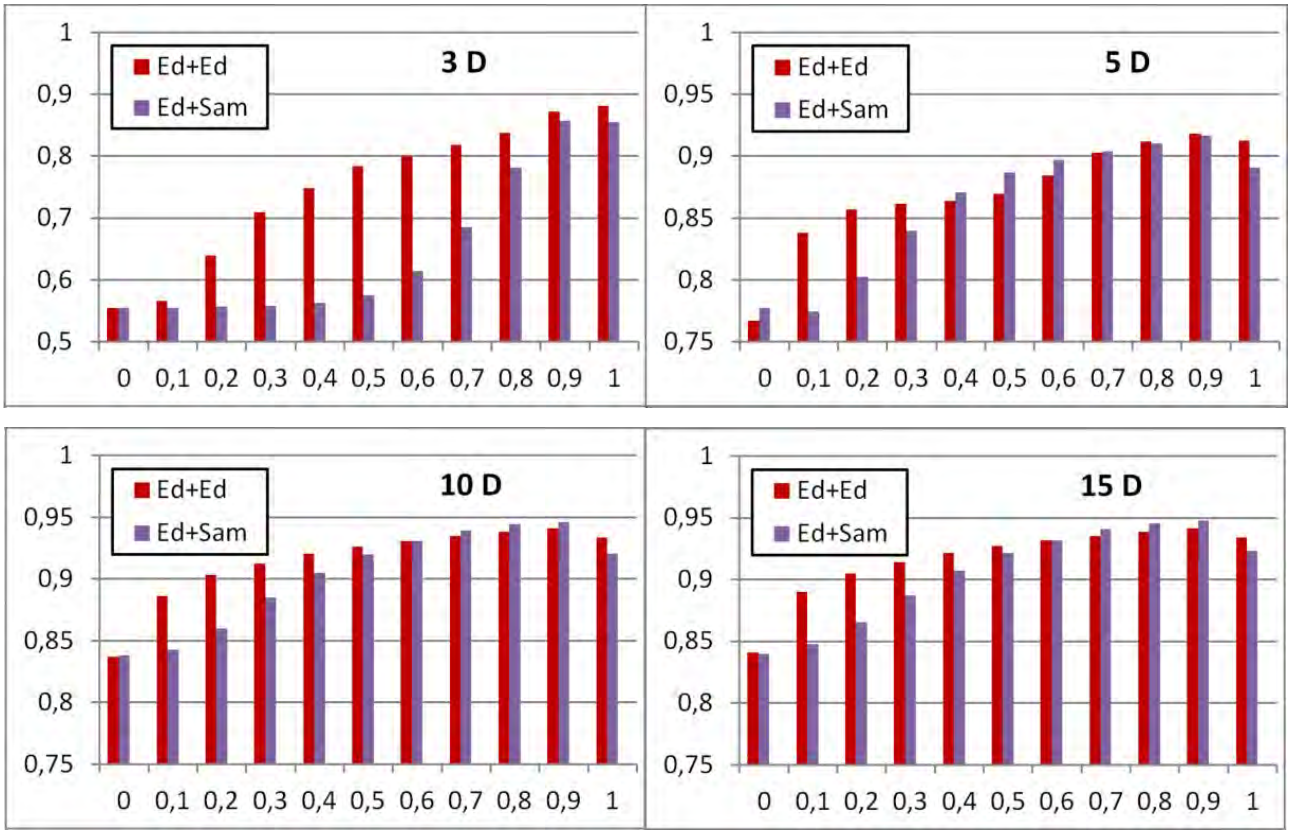


Рисунок 5.7 - Зависимость точности классификации суперпикселей от коэффициента слияния текстурных и спектральных признаков для различных функций расстояния и размерностей выходного пространства (набор данных Hyko-1, видимый диапазон).

Как видно из приведенных результатов, сделанные выше выводы подтверждаются и для этого набора данных. В частности, наилучшее значение качества достигается при D=15, коэффициенте слияния равном 0.9 (коэффициент менялся с шагом 0.1) и использовании спектрального угла в качестве функции расстояния во входном спектральном пространстве.

5.2.2 Анализ гиперспектральных данных ближнего инфракрасного диапазона

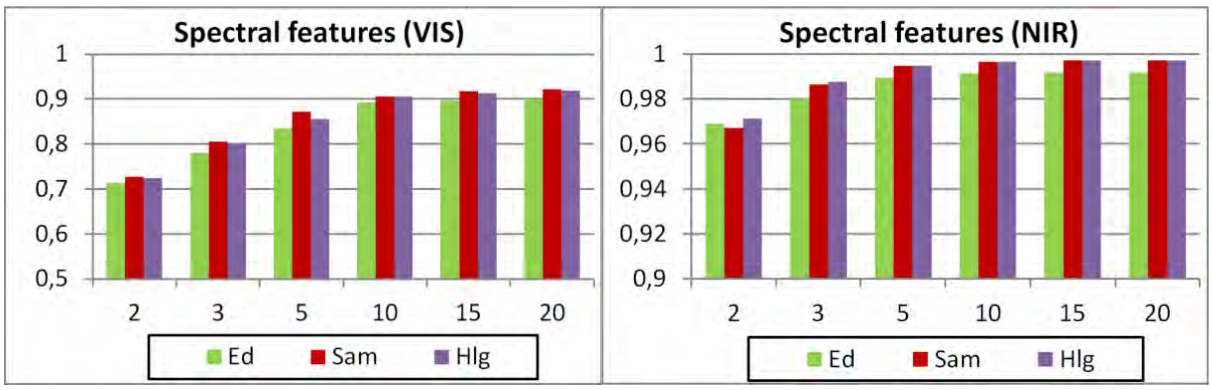
Исследование части набора данных, содержащей съемку ближнего инфракрасного диапазона, проводилось по аналогичной схеме. Существенное

отличие, помимо собственно диапазона регистрации, заключалось в сужении наблюдаемой сцены (см. рисунок 5.8).

Рисунок 5.8 - Примеры кадров дорожной обстановки из набора данных [311] (представлена первая главная компонента, изображения контрастированы): версия набора Hyko-1 (вверху), Hyko2 (внизу), ближний инфракрасный диапазон.

Исследование поведения классификаторов на признаках, формируемых с использованием различных функций расстояния в спектральном пространстве (с использованием методов, изложенных в разделе 2), показало, что достигнуть высокой точности классификации в NIR диапазоне было существенно проще, чем для случая видимого диапазона даже при малых размерностях (см. рисунок 5.9). Следует учесть, что представленные на рисунке результаты достигаются с использованием лишь спектральных признаков.

Учитывая высокое качество классификации с использованием спектральных признаков в NIR диапазоне, при проведении описываемых далее экспериментов объем обучающей выборки составлял всего 5% данных (суперпикселей).



(а) (б)

Рисунок 5.9 - Зависимость точности классификации суперпикселей для различных функций расстояния и размерностей выходного пространства: набор данных Hyko-1, видимый (а) и ближний инфракрасный (б) диапазон.

Текстурные признаки при этом также обеспечивали достаточно высокое качество классификации, что показал эксперимент, в котором осуществлялось слияние текстурных и спектральных признаков по схеме, изложенной в подразделе 3.3.

Из результата, представленного на рисунке 5.10 для набора данных Hyko-2, видно, что текстурные признаки (при коэффициенте слияния равном 0) давали достаточно высокое качество классификации. Использование чисто спектральных признаков (при коэффициенте слияния равном 1) давало несколько более высокое качество классификации. При этом слияние текстурных и спектральных признаков не позволило получить видимых преимуществ по сравнению с использованием только спектральных признаков.

Аналогичные исследования, выполненные для варианта набора данных Hyko-1, результат которых представлен на рисунке 5.11, показали, что такое слияние имеет смысл, что хорошо видно при использовании малых выходных размерностей. Наилучший результат (99,4%) был достигнут при размерности 15, слиянии с коэффициентом 0.9 и использовании спектрального угла в качестве функции расстояния в спектральном пространстве.

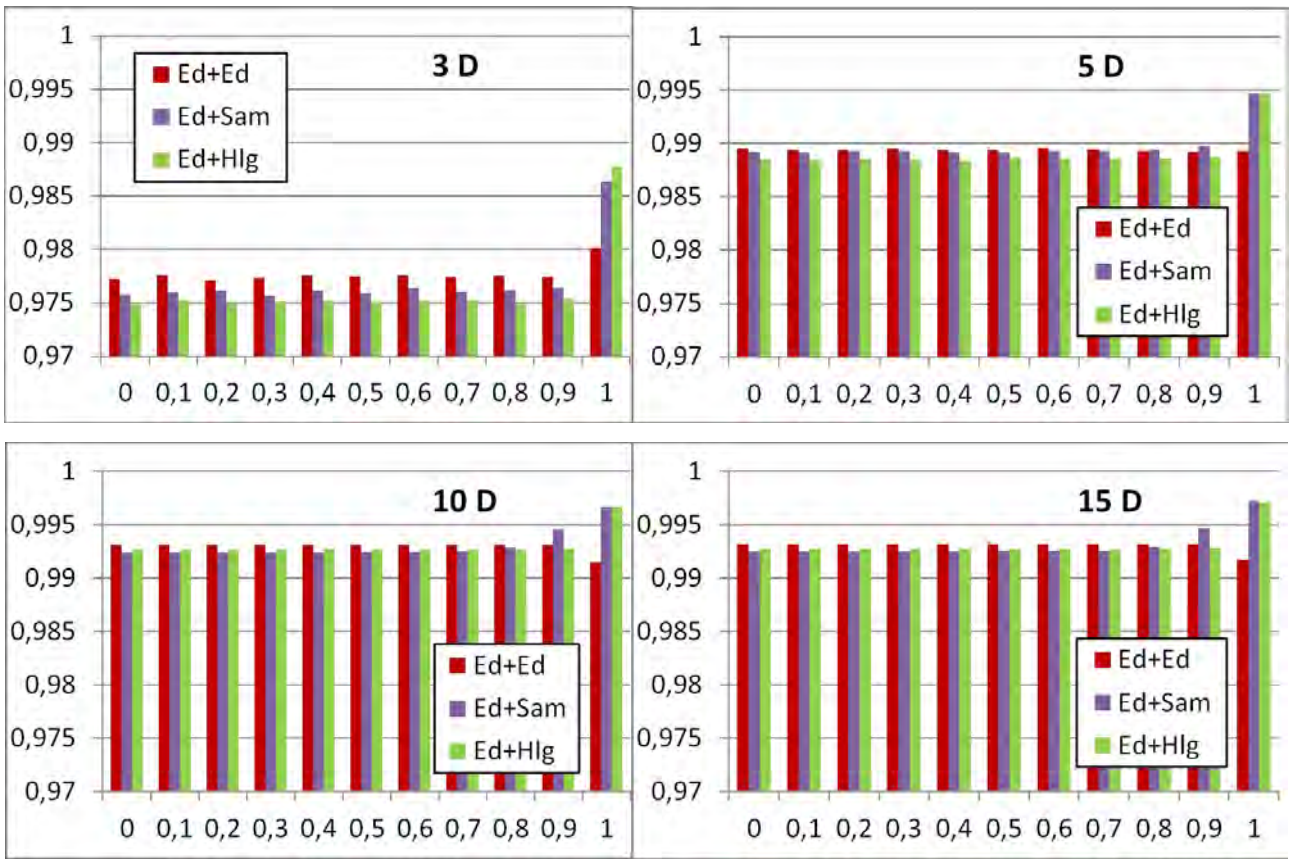


Рисунок 5.10 - Зависимость точности классификации суперпикселей от коэффициента слияния текстурных и спектральных признаков для различных мер рассогласования и размерностей выходного пространства (набор данных Hyko-2, ближний инфракрасный диапазон).

5.3 Классификация гиперспектральных изображений мозга человека

Гиперспектральная съемка может использоваться не только в дистанционном зондировании или навигации, но также предлагает многообещающие результаты для обнаружения и диагностики различных типов заболеваний в медицине. В частности, гиперспектральная съемка может использоваться в анализе и диагностике такого опасного заболевания как рак. Поскольку рак представляет собой клеточную патологию, приводящую, в том числе, к изменениям в химическом составе, пораженные ткани могут быть распознаны по изменениям в их спектральных сигнатурах [162].

Особенно важной и в то же время перспективной представляется возможность использования гиперспектральной съемки при выполнении операций по удалению опухолей головного мозга.

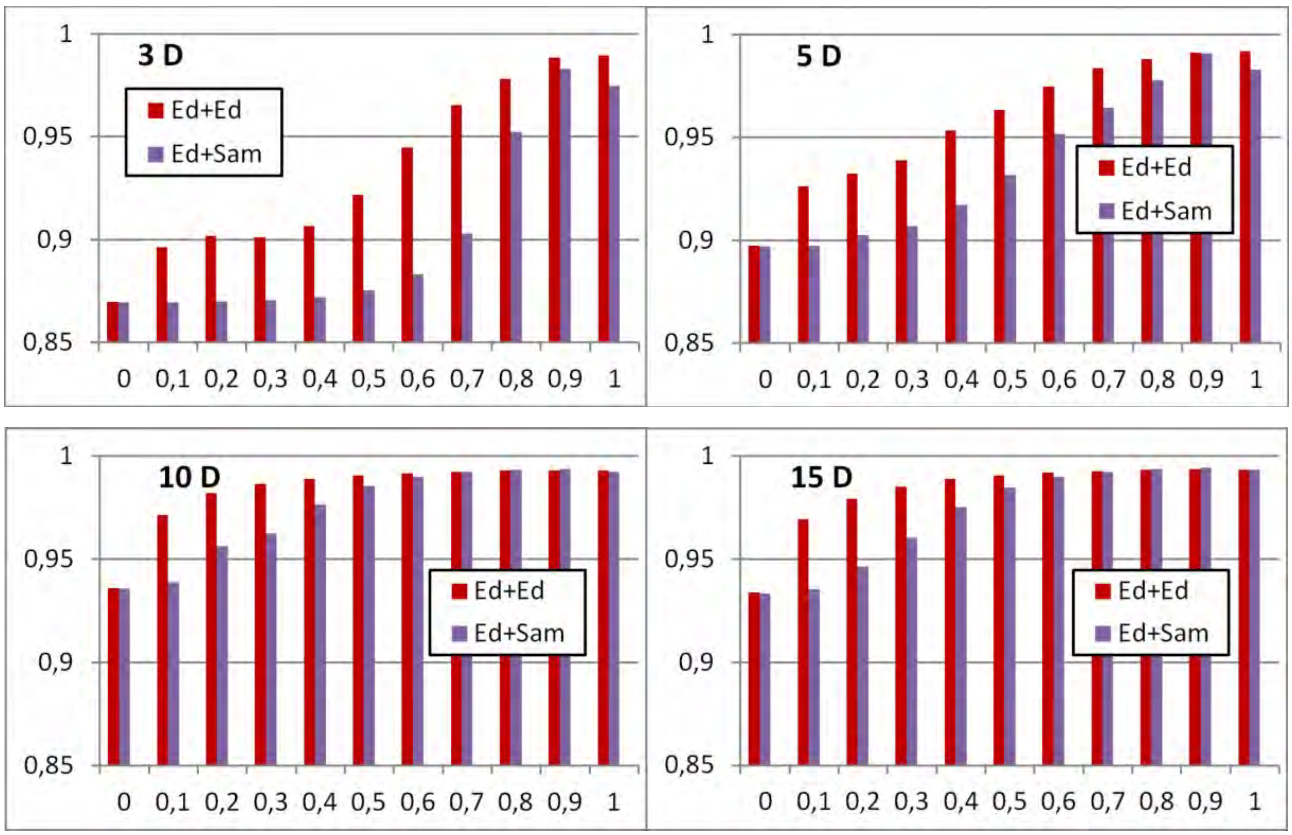


Рисунок 5.11 - Зависимость точности классификации суперпикселей от коэффициента слияния текстурных и спектральных признаков для различных функций расстояния и размерностей выходного пространства (набор данных Hyko-1, ближний инфракрасный диапазон).

В отличие от других видов опухолей, опухоль головного мозга инфильтрирует окружающую нормальную ткань головного мозга, поэтому ее границы нечеткие, хирургу трудно их идентифицировать невооруженным глазом. При этом окружающие опухоль нормальные ткани головного мозга могут иметь решающее значение для качества последующей жизни пациента. При выполнении таких операций критически важно точно идентифицировать границы опухоли, чтобы окружающие здоровые ткани оказались затронуты в минимальной степени. На сегодняшний день гиперспектральная съемка, являясь бесконтактным, безвредным для пациента методом получения диагностической информации, становится перспективным инструментом оказания помощи при нейрохирургических операциях по резекции опухоли.

Для изучения возможности и эффективности применения гиперспектральной съемки для детектирования и идентификации опухолей головного мозга во время проведения хирургических операций в рамках проекта HELICoID (HypEr-spectraL

Imaging Cancer Detection) был создан “набор данных гиперспектральных изображений мозга человека In-Vivo для обнаружения рака головного мозга” (In-Vivo Hyperspectral Human Brain Image Database for Brain Cancer Detection) [74].

Изображения для указанного набора получены с использованием специально разработанной системы сбора гиперспектральных данных, разработанной в рамках проекта HELICoiD [74]. Система сбора данных построена на базе маятникового (pushbroom) видеоспектрометра (Hyperspec® VNIR A-Series [119]). Сканер регистрирует излучение в диапазоне 400 - 1000 nm (видимый и ближний инфракрасный диапазон) с высоким спектральным разрешением, что позволяет получать 826 канальные гиперспектральные изображения.

Набор данных состоит из 36 изображений поверхности головного мозга 22 различных пациентов. Указанный набор данных снабжен истинной классификацией, включающей четыре тематических класса: нормальная ткань, опухолевая ткань, кровеносный сосуд и фон. Общее количество аннотированных пикселей составляет 387387 штук. Для аннотации использовалась автоматизированная система тематической разметки.

Примеры изображений из набора данных и соответствующая им истинная классификация показаны на рисунке 5.12.

5.3.1 Проведение экспериментов

Для описываемых далее исследований необходимо было выполнить подготовку данных из указанного выше набора. Подготовка данных включала в себя чтение исходных изображений в двоичном формате ENVI (включая HDR и RAW файлы), а также выполнение калибровки с применением темных и белых эталонных файлов. Калибровка выполнялась с использованием простого линейного преобразования с усечением значений, выходящих за границы [0..1]. Ввиду высокой спектральной размерности и зашумленности входных данных, выполнялась медианная фильтрация сигнатур скользящим окном в спектральной области с последующим сокращением размерности путем усреднения до размерности 165.

Для сокращения вычислительной нагрузки в процессе проведения предварительных экспериментов использовалось лишь 10% случайно выбранных аннотированных пикселей.

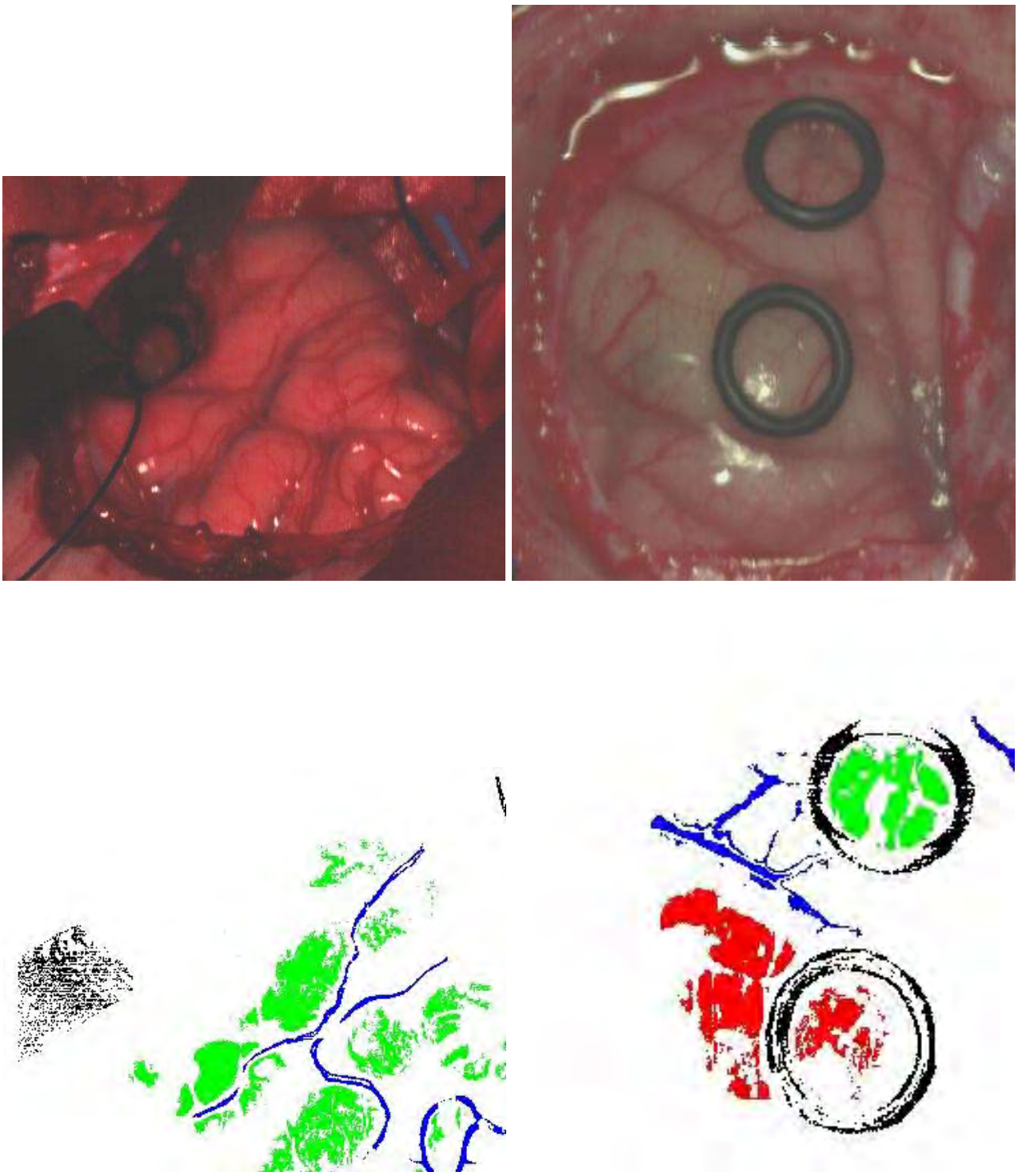
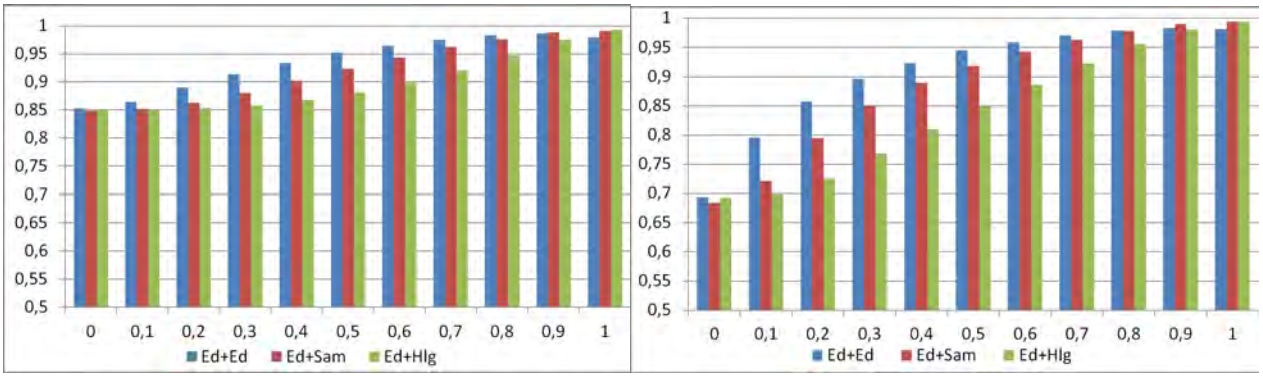


Рисунок 5.12 - Примеры изображений из набора данных [74] (синтезированные изображения в натуральных цветах, изображения контрастированы).

В первом эксперименте исследовалась целесообразность слияния текстурных и спектральных признаков по схеме, описанной в подразделе 3.3. Для этого осуществлялось слияние спектральных и двух описанных в подразделе 5.2 систем текстурных признаков с формированием признакового пространства размерностью до 20. Затем данные разделялись случайным образом на два равных подмножества, на

одном из которых выполнялось построение NN-классификатора, а второе использовалось для тестирования, после чего подмножества менялись местами, а результаты усреднялись. На рисунке 5.13 показаны результаты эксперимента.



(а) (б)

Рисунок 5.13 - Зависимость точности классификации *Acc* от коэффициента слияния λ признаков для различных типов текстурных признаков:
Признаки-1 (а); Признаки-2 (б)

Как видно из приведенных результатов, добавление текстурных признаков не оказывает заметного положительного влияния на качество классификации. Использование только спектральных признаков совместно со спектральным углом или дивергенцией Хеллингера в качестве функции расстояния позволило достичь точности классификации *Acc*=99.4%.

Применение же схемы с учетом пространственного контекста на основе порядковых статистик, изложенной в подразделе 3.2, позволило при использовании спектрального угла в качестве меры рассогласования, добиться точности *Acc*=99.5%. При этом качество классификации с увеличением количества использованных порядковых статистик повышалось достаточно медленно (см. рисунок 5.14).

Следует отметить, что таких показателей качества не удалось достичь при использовании стандартного метода снижения размерности PCA совместно с целым рядом методов классификации. В рамках этого эксперимента размерность спектрального пространства подготовленных данных снижалась с использованием PCA до значений 5,10,20,50,100, после чего данные разделялись на обучающую и тестовую выборки в соотношении 50:50, осуществлялось обучение и тестирование классификаторов. Результаты оценки качества показаны на рисунке 5.15. Отметим,

что при выполнении эксперимента использовались реализации классификаторов, включенные в пакет sklearn [283] с их стандартными настройками.

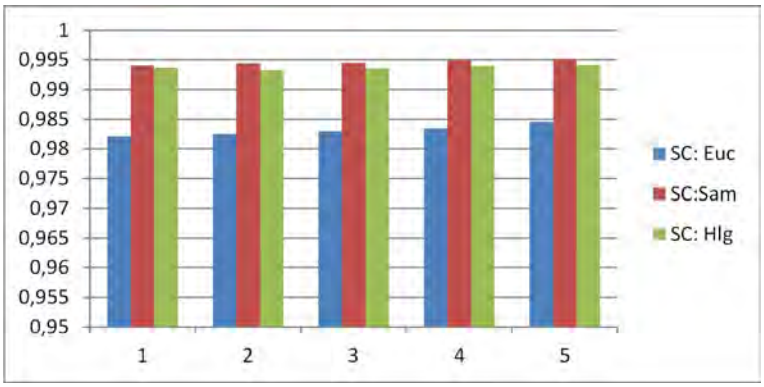


Рисунок 5.14 - Зависимость точности классификации *Acc* от количества порядковых статистик (размера пространственного контекста) для различных функций расстояния в спектральном пространстве.

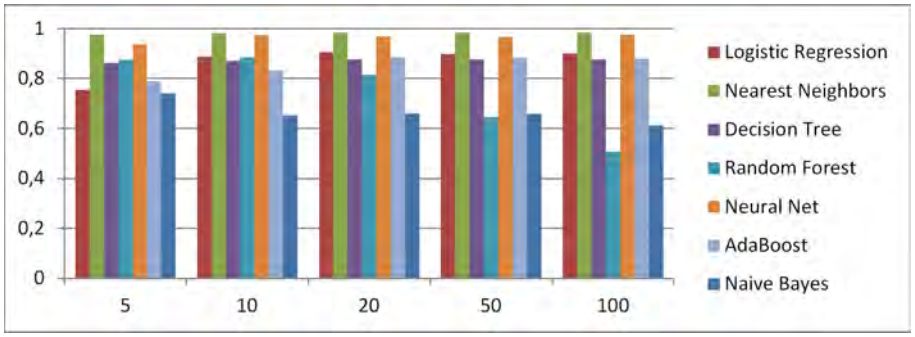


Рисунок 5.15 – Зависимость точности классификации от размерности выходного пространства при использовании метода главных компонент совместно с различными классификаторами

Наилучшее качество на всех значениях выходных размерностей показал классификатор по ближайшему соседу (обозначен на рисунке как Nearest Neighbor, 98.4%). Немного уступил ему по качеству многослойный персептрон (обозначен на рисунке как Neural Net, 97.6%). Остальные классификаторы показали меньшую точность.

Таким образом, на сокращенной выборке предложенные в подразделах 3.2 и 3.3 методы позволили повысить точность на 1.4% относительно стандартного подхода.

При трехкратной перекрестной проверке (кросс-валидации) на полном множестве данных метод на основе порядковых статистик показал точность классификации *Acc*=99.8%. Аналогичных результатов удалось достичь и при точном

подборе коэффициента слияния (λ=0.96) текстурных (Признаки-2) и спектральных признаков совместно с использованием спектрального угла в качестве функции расстояния во входном спектральном пространстве.

Таким образом, применение подхода на основе слияния спектральных и текстурных признаков и k-NN классификатора позволило достичь практически безошибочной классификации на исследуемом наборе данных.

5.4 Классификация типов растительности по данным наземной гиперспектральной съемки

В настоящем подразделе рассматривается важный класс гиперспектральных изображений – данные, получаемые по результатам наземной гиперспектральной съемки растительности. Такие данные используются, прежде всего, при решении сельскохозяйственных задач.

Данные, на основе которых проводилось исследование, были получены с использованием щелевого гиперспектрометра, построенного по схеме Оффнера [344]. Достаточно подробное описание условий съемки и набора данных приведено в [362].

Сформированный набор данных содержит как предварительно подготовленные гиперспектральные снимки растительности с пространственным разрешением 976x3000 пикселей и спектральным разрешением 250 каналов в диапазоне длин волн 420 – 980 нм, так и соответствующие данные истинной классификации. Данные истинной разметки включали маски двух зерновых сельхозкультур (овес и кукуруза), маски сорной травы, а также неклассифицированные области.

Гиперспектральное изображение из исследуемого набора данных и соответствующая маска истинной классификации показаны на рисунке 5.16.

При проведении экспериментов, как и в предыдущих подразделах, исследовалась целесообразность слияния текстурных и спектральных признаков по схеме, описанной в подразделе 3.3. В качестве размерности формируемого пространства использовались значения 10 и 20. В качестве функций расстояния в спектральном пространстве, как и ранее, использовались спектральный угол и дивергенция Хеллингера. Результаты, представленные на рисунке Рисунок 5.17, показывают возможность достаточно хорошего различения классов с использованием

указанных функций расстояния. При этом добавление простых текстурных признаков не дает положительного эффекта. Рост размерности формируемого пространства (с 10 до 20), напротив, дает возможность улучшить показатели классификации на 0,5-0,7%.

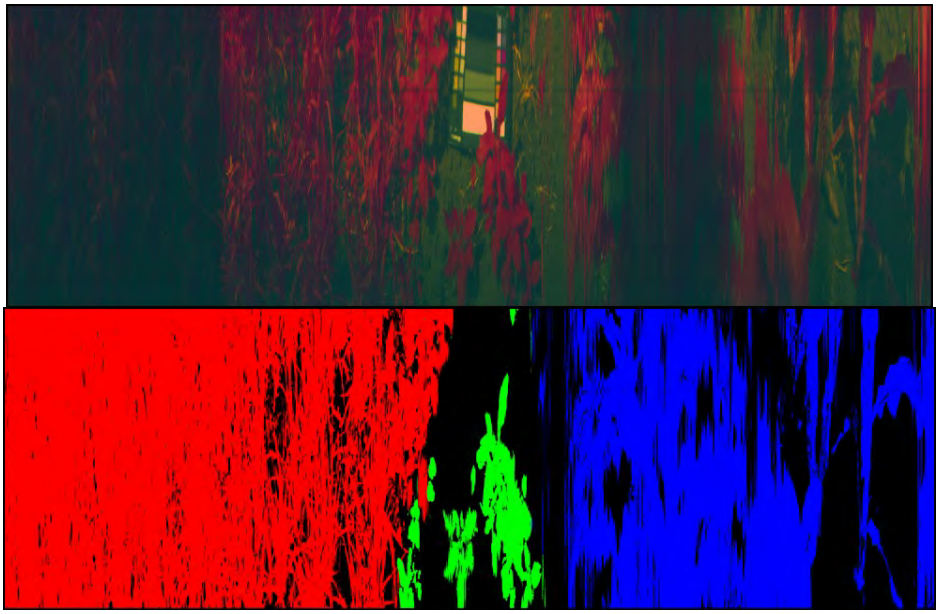
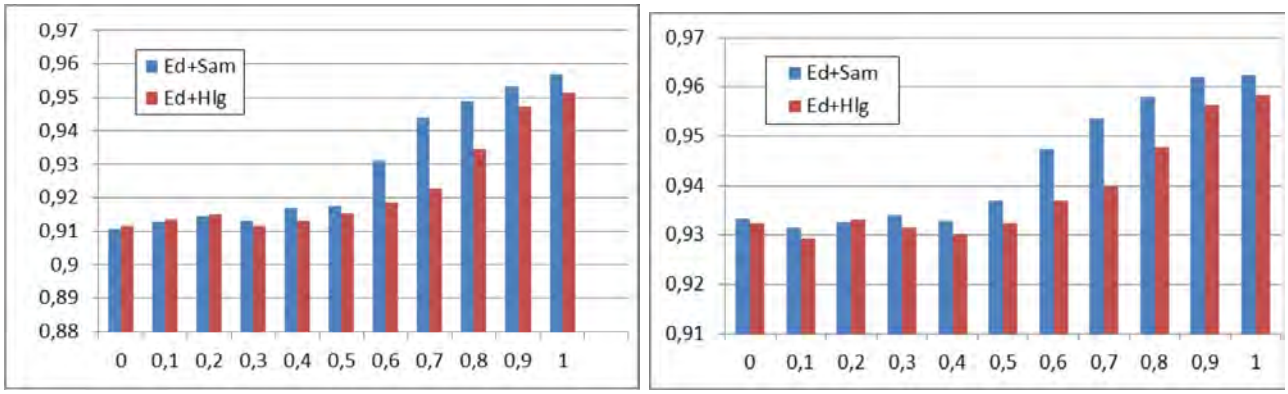


Рисунок 5.16 – Пример гиперспектрального изображения из набора данных наземной гиперспектральной съемки растительности: первые три главные компоненты показаны псевдоцветами (вверху), истинная классификация изображения (внизу): овес-красный, кукуруза – синий, сорняк – зеленый.



(а) (б)

Рисунок 5.17 - Зависимость точности классификации *Acc* от коэффициента слияния λ спектральных и текстурных признаков для размерности формируемого пространства 10 (а) и 20 (б)

Следует отметить, что с использованием стандартного подхода на основе метода главных компонент и классификатора на основе евклидовых расстояний

удается достичь лишь 91,7% точности в пространстве размерности 10 и 94,1% точности в пространстве размерности 20. Для сравнения, использование алгоритма формирования признаков ГСИ на основе сохранения спектральных углов (см. раздел 2.2) позволило получить точность классификации 95,3% и 96,2% для размерности 10 и 20, что выше на 4% и 2,1%, соответственно.

5.5 Алгоритмы кластеризации для неевклидовых метрик

Кластеризация гиперспектральных данных является одной из важнейших задач анализа гиперспектральных данных, в частности, изображений. Кластеризация может выполняться как один из этапов сегментации гиперспектральных изображений (сегментация через кластеризацию), использоваться для подготовки гиперспектральных изображений к полуавтоматической разметке, использоваться для решения прикладных задач анализа. Хотя материал настоящего подраздела перекликается с теоретическими основами, представленного в разделе 2 комплекса алгоритмических средств формирования признаков ГСИ, основанных на сохранении попарных расстояний, речь здесь идет не о формировании признаков, а о кластеризации гиперспектральных данных. Изложенные в настоящем подразделе решения используют теоретические результаты в части построения градиентных процедур, изложенных в подразделе 2.2.

Разрабатываемые алгоритмы кластеризации строятся по аналогии с известным алгоритмом k-внутригрупповых средних, который, к сожалению, некорректно использовать в тех случаях, когда расстояние между векторами (точками данных) рассчитывается с использованием неевклидовых расстояний.

Материал в настоящем подразделе организован следующим образом. В п. 5.5.1 кратко изложены основы алгоритма k-внутригрупповых средних. В пункте 5.5.2 рассматривается известный альтернативный алгоритм, допускающий использование при кластеризации неевклидовых метрик. Далее предлагается общая схема, и разрабатываются алгоритмы кластеризации для неевклидовых метрик. В частности, в пп. 5.5.3 и 5.5.4 реализованы алгоритмы для дивергении Хеллингера и угла Бхаттачария, соответственно.

5.5.1 Базовый алгоритм K внутригрупповых средних

В классическом алгоритме k-внутригрупповых средних минимизируется суммарное квадратическое отклонение содержащихся в кластерах точек от центроидов кластеров:

$$E_{ED} = \sum_{k=1}^{K} \sum_{x \in C_k} \left\| x - c_k \right\|^2 ,$$

где C_k – множество точек, входящих в k-ый кластер, c_k – центроид k-го кластера.
Наиболее распространенным методом решения, приводящим к локальному минимуму указанного выше функционала ошибки, является алгоритм Ллойда [161].

Алгоритм начинается с инициализации центроидов c_k, за которой следует последовательность итераций, на каждой из которых выполняется следующее.
Шаг 1. Все множество точек разбивается на k непересекающихся подмножеств C_k в соответствии с близостью центроидам:

$$C_k = \left\{ x \,\middle|\, k = \arg\min_j \left\| x - c_j \right\| \right\}. \quad .$$

Шаг 2. Обновляются центроиды кластеров:

$$c_k = \frac{1}{|C_k|} \sum_{x \in C_k} x .$$

Алгоритм завершает свою работу, когда кластеры перестают изменяться.

Как можно видеть, приведенный алгоритм построен для работы с евклидовой метрикой, и простая замена евклидова расстояния на другую метрику при разбиении точек на кластеры не совсем корректна.

5.5.2 Алгоритм k медоидов

В алгоритме k- медоидов данные обрабатываются аналогичным образом, однако вместо центроидов в качестве репрезентативных объектов используются сами элементы выборки (медоиды) [136]. В ряде задач это более естественно. Кроме того, этот алгоритм позволяет использовать произвольные расстояния и не накладывает ограничений на исходное пространство.

В качестве целевой функции алгоритм использует абсолютную ошибку:

$$e = \sum_{k=1}^{K} \sum_{x \in C_k} d(x, c_k) \qquad (5.1)$$

Здесь медоид кластера c_k определяется как элемент, который меньше всего отличается от других элементов кластера C_k в смысле среднего рассогласования:

$$c_k = \arg\min_{c \in C_k} \sum_{x \in C_k} d(x, c) \quad .$$

Хотя сама задача k-медоидов является NP-сложной, наиболее распространенный итерационный алгоритм Вороного [102] похож на описанный выше алгоритм Ллойда для k-внутригрупповых средних. Существенной проблемой алгоритма k- медоидов является фактический выбор медоида для каждого из кластеров. В отличие от вычисления центроида в алгоритме k- внутригрупповых средних, который выполняется за линейное по отношению к количеству элементов время, для непосредственного выбора медоида требуется квадратичное время.

По этой причине было разработано несколько алгоритмов для ускорения вычислений, например, TOPRANK, TOPRANK2 [227], TRIMED [219]. В настоящем исследовании последний из перечисленных алгоритмов [219] используется для вычисления медоида за субквадратичное время, но он накладывает ограничения на пространство: оно должно быть метрическим.

5.5.3 Модифицированный алгоритм K-средних для дивергенции Хеллингера

Разработку алгоритма кластеризации для неевклидовых метрик будем проводить на примере дивергенции Хеллингера. Такой выбор обуславливается следующими соображениями. В быстрой версии алгоритма k-медоидов требуется, чтобы мера рассогласования была метрикой. Для обеспечения возможности прямого сравнения с быстрой версией k-медоидов, создаваемый алгоритм кластеризации должен основываться на такой же метрике. Кроме того, для измерения расстояния между точками при кластеризации особый интерес представляет использование теоретико-информационных расстояний, так как такие расстояния обеспечивали высокое качество решения смежной задачи классификации.

При этом дивергенция спектральной информации, хорошо зарекомендовавшая себя в классификации гиперспектральных изображений, не удовлетворяет неравенству треугольника [1] и, следовательно, не является истинной метрикой. Между тем, как уже упоминалось в подразделе 2.1, дивергенция Хеллингера не только удовлетворяет необходимым требованиям (положительность, тождество,

симметрия и неравенство треугольника), но и демонстрирует практически неотличимое качество классификации по сравнению с дивергенцией спектральной информации [192*].

Эту метрику можно сразу использовать в алгоритме k-медоидов. Прежде чем перейти к описанию разработанного алгоритма кластеризации на основе дивергенции Хеллингера, напомним, что алгоритм Ллойда состоит из двух итеративно повторяемых шагов: разбиение всех точек на непересекающиеся подмножества в соответствии с близостью к центроидам и обновление центроидов.

Предложенный автором алгоритм кластеризации изложен в работе [190*] и состоит в следующем.

Шаг 1. Модификация первого шага очевидна: мы используем расстояние Хеллингера в качестве метрики для разделения точек на кластеры:

$$C_k = \left\{ x \,\middle|\, k = \arg\min_j HD(x, c_j) \right\}$$

Шаг 2. На втором шаге необходимо реализовать подходящую процедуру обновления центроидов кластеров.

Чтобы построить процедуру обновления, определяем центроид c кластера C как точку в пространстве, для которой общее расстояние от остальных точек кластера минимально в смысле выбранной метрики:

$$c = \arg\min_{c^*} \sum_{x \in C} HD^2(x, c^*)$$

Далее, чтобы найти центроид c кластера C, применяем метод градиентного спуска, состоящий из следующих шагов:

Шаг 2.a. Инициализация центроида значением $c(0)$.

Шаг 2.b. Итеративное обновление центроида c:

$$c(t) = c(t-1) - \alpha \nabla \sum_{x \in C} HD^2(x, c) .$$

Здесь, как и ранее, t - номер итерации, ∇ - градиент, α - коэффициент градиентного спуска.

Частная производная дивергенции Хеллингера по координатам центроида записывается следующим образом (см. п. 2.2.5 или Приложение Б.6):

$$\frac{\partial}{\partial c_n} HD(c,x) = \frac{1}{4\, HD(c,x) \sum_{l=1}^{M} c_l} \left(\sum_{k=1}^{M} \sqrt{p_k(c) p_k(x)} - \sqrt{\frac{p_n(x)}{p_n(c)}} \right).$$

Это дает следующее рекуррентное соотношение для нахождения центроида *c*:

$$c(\tau) = c(\tau - 1) - \frac{\alpha}{2 \sum_{l=1}^{M} c_l(\tau - 1)} \sum_{x \in C} \left(\sum_{k=1}^{M} \sqrt{p_k(c(\tau-1)) p_k(x)} - \sqrt{\frac{p_n(x)}{p_n(c(\tau-1))}} \right).$$

В случае использования абсолютной ошибки, определяем центроид как $c = \arg\min_{c^*} \sum_{x \in C} HD(x, c^*)$, а соответствующее решение записывается в виде:

$$c(\tau) = c(\tau - 1) - \frac{\alpha}{4 \sum_{l=1}^{M} c_l(\tau - 1)} \sum_{x \in C} \frac{1}{HD(c(\tau-1), x)} \left(\sum_{k=1}^{M} \sqrt{p_k(c(\tau-1)) p_k(x)} - \sqrt{\frac{p_n(x)}{p_n(c(\tau-1))}} \right).$$

Стоит отметить, что для инициализации центроидов в начале модифицированного алгоритма используется тот же подход, что и в алгоритме kmeans ++ [9].

В реализации предложенного метода использовался алгоритм градиентного спуска с контролем ошибок, в котором коэффициент α уменьшался в 2 раза в случае повышения ошибки. В качестве критерия остановки использовалась абсолютная разность ошибки на двух последовательных шагах процедуры: оптимизация останавливалась, когда эта разность падала ниже небольшого предопределенного значения (10^{-8}). Кроме того, максимальное количество итераций ограничивалось значением *MaxIter*, которое было параметром алгоритма.

5.5.4 Модифицированный алгоритм К-средних для угла Бхаттачария

Аналогичным образом можно построить процедуру оптимизации и для случая использования угла Бхаттачария. Частная производная угла Бхаттачария по координатам центроида записывается в виде (см. Приложение Б.7):

$$\frac{\partial}{\partial c_n} BCa(c,x) = \frac{1}{2 \sqrt{1 - \left(\sum_{k=1}^{M} \sqrt{p_k(c) p_k(x)} \right)^2} \sum_{l=1}^{M} c_l} \left(\sum_{k=1}^{M} \sqrt{p_k(c) p_k(x)} - \sqrt{\frac{p_n(x)}{p_n(c)}} \right).$$

Поиск решения $c = \arg\min_{c^*} \sum_{x \in C} BCa^2(x, c^*)$ при использовании квадратичной ошибки дает следующее рекуррентное соотношение для центроида *c*:

$$c(\tau) = c(\tau - 1)$$
$$- \frac{\alpha}{\sum_{l=1}^{M} c_l(\tau - 1)} \sum_{x \in C} \frac{BCa(c(\tau - 1), x)}{\sqrt{1 - \left(\sum_{k=1}^{M} \sqrt{p_k(c(\tau - 1)) p_k(x)} \right)^2}} \left(\sum_{k=1}^{M} \sqrt{p_k(c(\tau - 1)) p_k(x)} - \sqrt{\frac{p_n(x)}{p_n(c(\tau - 1))}} \right).$$

При использовании абсолютной ошибки и минимизации $c = \arg\min_{c^*} \sum_{x \in C} BCa(x, c^*)$, получаем:

$$c(\tau) = c(\tau - 1)$$
$$- \frac{\alpha}{2\sum_{l=1}^{M} c_l(\tau - 1)} \sum_{x \in C} \frac{1}{\sqrt{1 - \left(\sum_{k=1}^{M} \sqrt{p_k(c(\tau - 1)) p_k(x)} \right)^2}} \left(\sum_{k=1}^{M} \sqrt{p_k(c(\tau - 1)) p_k(x)} - \sqrt{\frac{p_n(x)}{p_n(c(\tau - 1))}} \right).$$

5.5.5 Эксперименты

В экспериментах предложенный алгоритм для дивергенции Хеллингера сравнивался с альтернативным алгоритмом k-медоидов по критерию ошибки кластеризации (5.2) и временных характеристик.

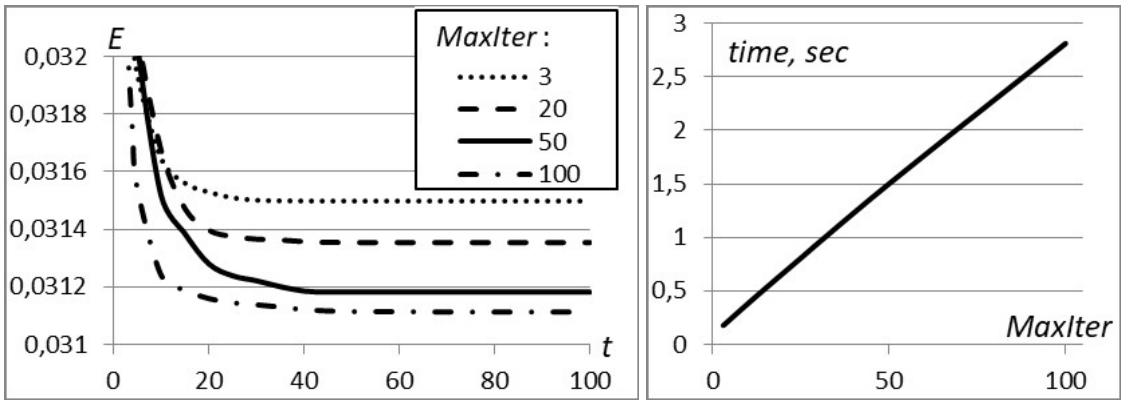
$$E = \sum_{k=1}^{K} \sum_{x \in C_k} HD^2(X, c_k). \tag{5.2}$$

Для оценки алгоритмов кластеризации, использовалось известное гиперспектральное изображение Indian Pines [17]. Все алгоритмы были реализованы на языке C++. Исследование проводилось на ноутбуке на базе процессора Intel Core i7-6500U @ 2,3 ГГц.

В первом эксперименте исследовалось, как максимальное количество итераций *MaxIter* в алгоритме градиентного спуска влияет на окончательное качество кластеризации (5.2). Для этого изменялось максимальное количество итераций в алгоритме обновления центроидов (шаг 2.b алгоритма) и измерялось качество кластеризации после обновления всех центроидов. Результаты показаны на рисунке 5.18.

Каждый график на рисунке 5.18(a) соответствует определенному значению параметра *MaxIter* (здесь мы представляем только результаты для 3, 20, 50 и 100

итераций алгоритма градиентного спуска) и показывает зависимость качества кластеризации *E* от количества *t* итераций в модифицированном алгоритме Ллойда, описанном в п. 5.5.3. Как можно видеть, более высокие значения *MaxIter*, приводящие к более точным обновлениям центроидов, обеспечивают меньшее количество ошибок, а усредненное время *time* одной итерации модифицированного алгоритма Ллойда растет с увеличением *MaxIter* почти линейно (см. рисунок 5.18(б)).

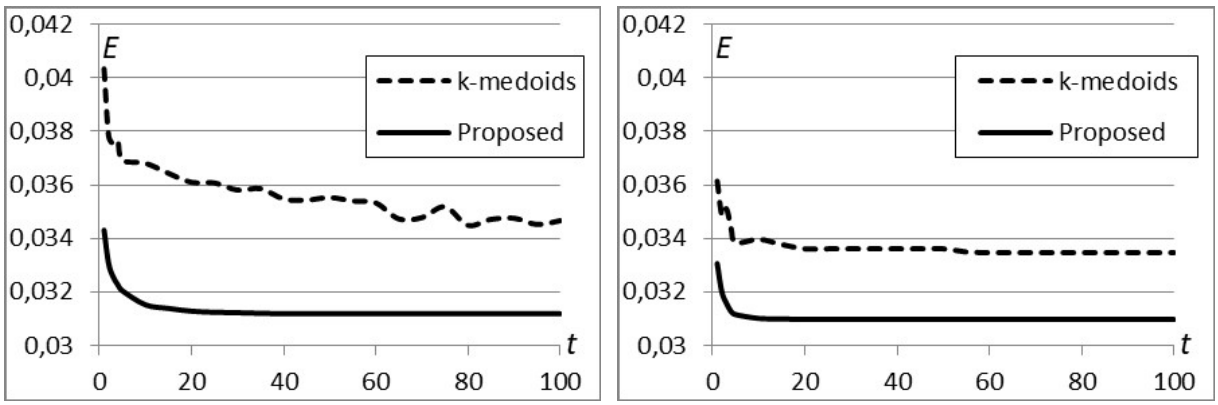


(а) (б)

Рисунок 5.18 - Влияние параметра *MaxIter* на качество кластеризации *E* (а) и время *time* выполнения одной итерации алгоритма Ллойда (б) для предлагаемого метода (усреднено по 20 запускам).

Во втором эксперименте предложенный алгоритм кластеризации сравнивался с алгоритмом k-медоидов, основанным на алгоритме TRIMED и работающим с дивергенцией Хеллингера (см. Раздел 2.2). На рисунке 5.19 показана зависимость качества кластеризации *E* от количества *t* итераций в алгоритме Ллойда как для предложенного метода, так и для метода k-медоидов. На этом рисунке левая часть показывает значение ошибки, усредненное по 30 запускам алгоритмов, а правая часть показывает лучшие (минимальные) значения по 30 запускам. В обоих случаях мы видим, что предложенный подход дает лучшие результаты независимо от количества итераций.

Что касается времени, TRIMED-реализация алгоритма k-медоидов потребовала около 0,55 секунды на итерацию, что примерно соответствует настройке MaxIter = 20 для предложенного подхода (0,67 сек). При этой настройке предлагаемый алгоритм не обеспечивает наилучшего качества, но по-прежнему превосходит алгоритм k-медоидов.



(а) (б)

Рисунок 5.19 - Зависимость качества кластеризации для k-медоидов (TRIMED) и предложенного метода от количества итераций:
усредненная ошибка (а) и минимальная ошибка (б) за 30 прогонов.

Стоит отметить, что алгоритм k-медоидов использует целевую функцию (5.1) (с дивергенцией Хеллингера в качестве расстояния), отличную от (5.2), но всё вышесказанное верно и в отношении ошибки (5.1). В частности, после 100 итераций показатель (5.1) был равен e_{k_med} = 0,0015 и e_{prop} = 0,0012 для k-medoids и предложенного алгоритма, соответственно, что соответствует снижению ошибки на 25%.

Во всех случаях представлены результаты для кластеризации на 10 кластеров. В качестве примера приведем результаты (ошибки (5.2) и (5.1)) для кластеризации на 20 кластеров после 100 итераций: E_{k_med} = 0,0331 и e_{k_med} = 0,00137 для k-медоидов против E_{prop} = 0,0295 и e_{prop} = 0,00105 для предложенного алгоритма, что на 12% и 30% ниже, соответственно. Таким образом, предложенный алгоритм превзошел алгоритм k-медоидов во всех рассмотренных случаях.

5.6 Сегментация гиперспектральных изображений на основе оценки размерности в пространственных областях

Автоматическая сегментация гиперспектральных изображений, наряду с управляемой классификацией, кластеризацией и спектральным разделением, является одной из основных задач их анализа. Под сегментацией изображения обычно понимается разбиение изображения на области, соответствующие различным

объектам или их частям, обладающие свойствами однородности в смысле определенного критерия.

Для сегментации изображений используется множество различных методов, которые можно классифицировать следующим образом [82]:

- методы, основанные на характеристиках, разделяющие все пиксели изображения на подмножества на основе значений пикселей или производных свойств;

- методы, основанные на областях, использующие некоторый критерий однородности для обнаружения областей на изображении;

- методы, основанные на границах, использующие свойства разрывности для обнаружения границ, разделяющих изображение на области.

При сегментации гиперспектральных изображений популярные методы кластеризации [20, 36] являются примерами первого класса; методы, основанные на выращивании областей и преобразовании водораздела [221, 287], представляют второй класс методов.

К сожалению, стандартные подходы к сегментации гиперспектральных изображений [211*], основанные, например, на кластеризации или преобразовании водораздела, часто не обеспечивают желаемого качества сегментации. По этой причине важным представляется создание методов сегментации, основанных на свойствах гиперспектральных изображений. Таким методом стал разработанный автором [184*, 197*, 354*] метод сегментации, основанный на разделении изображения на области, однородные с точки зрения размерности спектральных данных, содержащихся в областях.

5.6.1 Описание метода

Предположим, что гиперспектральное изображение содержит представление конечного числа объектов (или материалов), описываемых уникальными спектральными сигнатурами. Из-за низкого пространственного разрешения изображения отдельный пиксель этого изображения может содержать информацию о нескольких спектральных сигнатурах.

Предполагается, что результирующий спектр x_i пикселя с индексом i, $i = 1…N$ может быть описан как линейная комбинация [138] некоторого числа K элементарных сигнатур $\{e_k\}_{k=1…K}$, соответствующих спектрам чистых компонентов (материалов),

присутствующих на сцене, с точностью до шумовой составляющей, а сами сигнатуры линейно независимы:

$$x_i = \sum_{k=1}^{K} a_{ik} e_k + \eta_i \,. \tag{5.3}$$

Здесь η_i является аддитивным шумовым вектором, а каждый из весов a_{ik} определяется долей площади компонента (материала) с элементарным спектром e_k, содержащегося на соответствующем пикселю участке сцены.

Содержание a_{ij} элементарных сигнатур e_j в каждом пикселе x_i подчиняется естественным ограничениям:

$$a_{ik} \geq 0, \;\; i = 1..N, k = 1..K,$$
$$\sum_{k=1}^{K} a_{ik} = 1, \;\; i = 1..N.$$

Рассмотрим некоторую пространственную область ГСИ, однородную по спектральному составу (см. область окрестности N_i пикселя x_i на рисунке 5.20). Внутренняя размерность d спектральных данных, содержащихся во всех пикселях такой области, будет определяться количеством различных спектральных сигнатур, содержащихся в пикселях области (предположим, что в этой области имеется достаточное количество пикселей для такой оценки). Добавление пикселей, содержащих другие сигнатуры (например, красный пиксель на рисунке 5.20) в однородную область, неизбежно приведет к увеличению ее внутренней размерности и нарушению однородности.

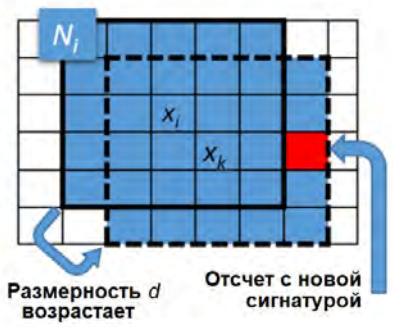


Рисунок 5.20 - Сдвиг окна в пространственной области вызывает изменение внутренней размерности.

Основываясь на этой идее, можно выполнять обработку изображения скользящим окном (рисунок 5.20). Для каждого пространственного положения скользящего окна, соответствующего пикселю x_i, оценивается внутренняя

размерность d_i соответствующей окрестности N_i. Чтобы получить такую оценку, выполняется анализ главных компонент [129] с последующим поиском минимальной размерности линейного подпространства, проекция в которое позволяет сохранить не менее заранее определенной доли общей дисперсии данных.

Пусть X – центрированная матрица данных, составленная из пикселей (векторов) некоторой пространственной области N. Выполним решение задачи анализа главных компонент (см. Приложение А.1) и запишем собственные значения $\{\lambda_i\}_{i=1..D}$ ковариационной матрицы $C = X^T X$ в порядке возрастания $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_D$. Теперь искомую внутреннюю размерность d пикселей пространственной области N можно оценить следующим образом:

$$\begin{aligned} & d \longrightarrow \min \\ & \textit{при условии}: \quad \left(\sum_{i=1}^{d} \lambda_i \Big/ \sum_{i=1}^{D} \lambda_i \right) > \tau, \\ & d \in \{0,1,\ldots D\}. \end{aligned} \tag{5.4}$$

Здесь порог τ - настраиваемый параметр.

Собственно метод сегментации состоит в следующем.

С использованием (5.4) выполняется агрегация оценок по окрестностям всех пикселей изображения и формируется матрица оценок размерности.

На следующем шаге указанная выше матрица оценок используется для поиска связных областей пикселей с одинаковой размерностью. Результат этого шага можно рассматривать как предварительную сегментацию, но она все еще избыточна.

По этой причине для каждой связанной области r выполняется пересчет собственных значений $\{\lambda_{r,i}\}_{i=1..D}$ и собственных векторов $\{v_{r,i}\}_{i=1..D}$ с оценкой внутренней размерности d_r.

Далее производится попытка объединения соседних пикселей или областей, если это не увеличивает внутреннюю размерность (5.4) и не ведет к достаточно большим ошибкам реконструкции при проекции присоединенных пикселей в линейное подпространство меньшей размерности. Последнее условие для некоторой области r и пикселя x_i может быть представлено в следующем виде:

$$\left\| x_i \left[0 \ldots \quad 0 \quad v_{r,d_r+1} \ldots \quad v_{r,D} \right] \right\| < w\sigma_r ,$$

Здесь $v_{r,d_r+1},\ldots v_{r,D}$ - собственные вектора, соответствующие наименьшим собственным значениям $\lambda_{r,d_r+1},\ldots \lambda_{r,D}$, значение $w\sigma_r$ — порог, оцениваемый с помощью ошибки реконструкции σ_r, усредненной по всем пикселям области r, изначально содержавшимся в этой области

$$\sigma_r = \frac{1}{|r|}\sum_{x_i \in r} \left\| x_i \begin{bmatrix} 0\ldots & 0 & v_{r,d_r+1}\ldots & v_{r,D} \end{bmatrix} \right\|,$$

w — постоянный множитель, являющийся настройкой метода.

Выполнение описанного метода может занимать много времени, особенно для больших изображений, содержащих сотни спектральных компонентов. По этой причине на этапе предварительной обработки может быть выполнено снижение размерности, чтобы уменьшить вычислительную нагрузку.

Общая схема описываемого метода представлена на рисунке 5.21.

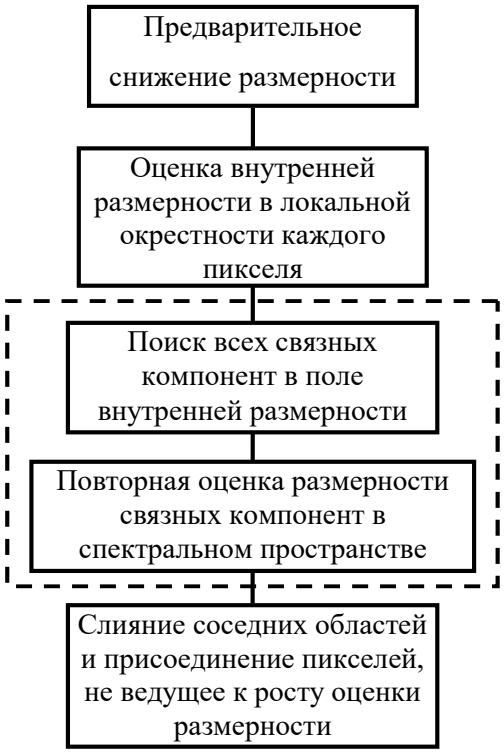


Рисунок 5.21 - Схема предлагаемого метода.

Рассматриваемый метод впервые был описан в [184*] как часть метода адаптивного снижения размерности ГСИ.

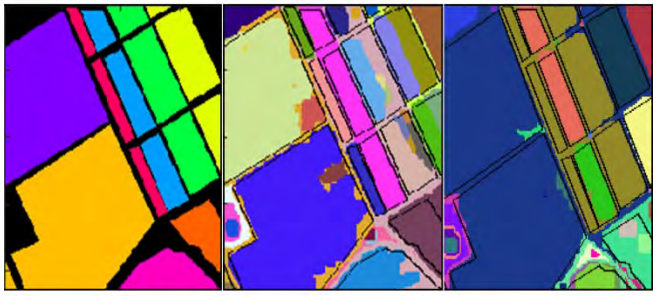
Стоит отметить, что здесь используется прямоугольное окно для выполнения начальной оценки внутренней размерности. Геометрия и размер окна существенно влияют на результаты сегментации, поскольку окно определяет минимальную область, которая может быть изначально извлечена алгоритмом. С одной стороны,

если окно достаточно велико и не может быть полностью помещено в однородную область из-за его размера или геометрических свойств (например, узкая длинная область), такая область может быть пропущена на ранних этапах метода, даже если площадь области достаточно большая. С другой стороны, мы не можем использовать слишком маленькие окна, потому что это приводит к вырожденным оценкам размерности. В целом, можно рассматривать использование прямоугольных окон как ограничение текущей реализации.

5.6.2 Эксперименты

Для проведения исследования использовались те же тестовые гиперспектральные изображения, что и ранее (см. п. 2.4.1). Приведем результаты для изображений Indian Pines [120, 17] (224 спектральных полосы были сокращены до 200 полос за счет удаления полос из-за шума и / или поглощения воды) и Salinas (224 спектральных полосы были сокращены до 204 полос по тем же причинам).

Поскольку оба изображения имеют данные истинной классификации, можно использовать ее для оценки качества сегментации. На рисунке 5.22(а) показан фрагмент истинной классификации изображения Salinas, в котором различными цветами показаны области, соответствующие различным классам растительности. На рисунке 5.22(б, в) показаны примеры сегментированных изображений, полученных для различных параметров метода. На этих изображениях различные выделенные области показаны случайными цветами.



(а) (б) (в)

Рисунок 5.22 - Информация об истинной классификации изображения Salinas (а) и соответствующие сегментированные фрагменты для различных параметров (б, в).

К сожалению, истинная классификация не является полной как для сцены Salinas, так и для сцены Indian Pines (неклассифицированные области показаны

черным цветом на рисунке Рисунок 5.22(а)), поэтому для оценки качества сегментации использовалась только имеющаяся информация.

Для оценки качества сегментации использовались такие меры, как индекс Рэнда и Глобальная ошибка согласованности.

Индекс Рэнда (RI) [240] описывается следующим выражением.

$$RI(S_1,S_2)=\frac{1}{\binom{N}{2}}\sum_{\substack{i,j\\ i\neq j}}\left(I(l_i^1=l_j^1\wedge l_i^2=l_j^2)+I(l_i^1\neq l_j^1\wedge l_i^2\neq l_j^2)\right).$$

Здесь $I()$ - функция, принимающая значение 1, если ее аргумент истинен, и 0 в противном случае, l_i^k - метка (сегмент) i-го пикселя на k-й сегментации, а знаменатель $\binom{N}{2}$ - количество всех возможных уникальных пар из N пикселей.

Глобальная ошибка согласованности (GCE) [293] определяется формулами:

$$GCE(S_1,S_2)=\frac{1}{n}\min\left\{\sum_i\varepsilon_i(S_1,S_2),\sum_i\varepsilon_i(S_2,S_1)\right\},$$

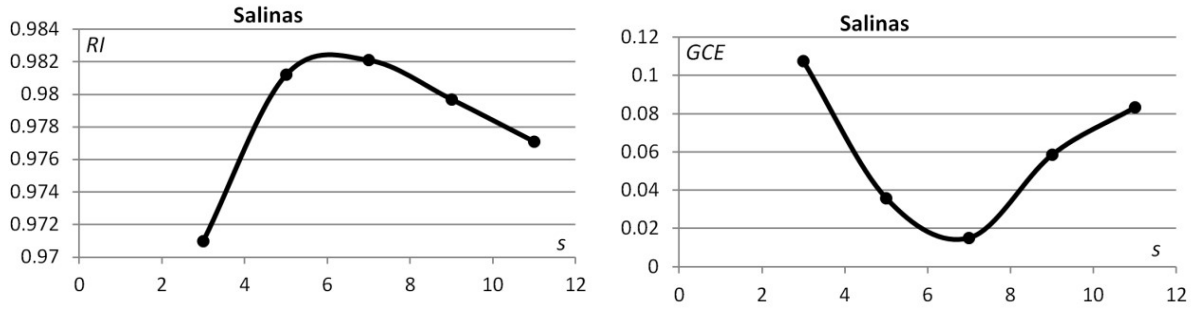
где S_1 и S_2 - два сравниваемых результата сегментации, а

$$\varepsilon_i(S_1,S_2)=\frac{|R_{1i}\setminus R_{2i}|}{|R_{1i}|}$$

- мера ошибки для i-го пикселя. В последнем выражении R_{1i} - это область, содержащая i-й пиксель в сегментации S_1, а R_{2i} - область, содержащая i-й пиксель в сегментации S_2.

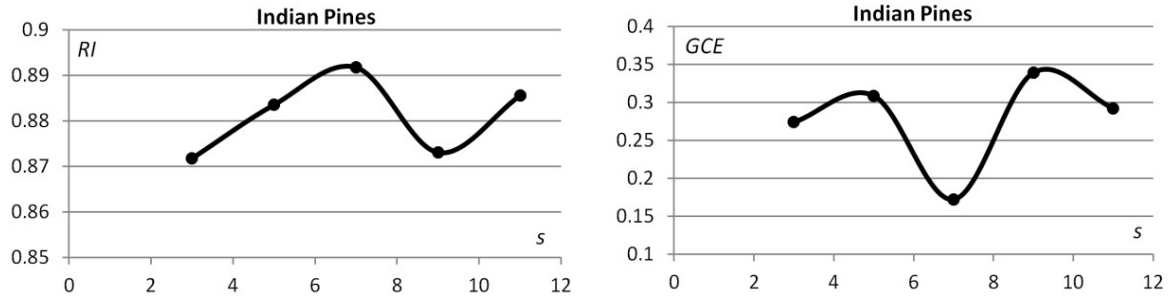
Целью первого эксперимента было изучение того, как размер соседства влияет на качество сегментации. Результаты первого эксперимента показаны на рисунке 5.23 для сцены Salinas и на рисунке 5.24 для сцены Indian Pines.

Представленные зависимости показывают поведение показателей *RI* (большие значения - лучше) и *GCE* (меньшие значения - лучше) в зависимости от линейного размера *s* (стороны квадрата пространственного окна обработки). Как видно, индикаторы демонстрируют практически противоположное поведение, что ожидаемо. Оптимальное значение размера окна в обоих случаях равно 7, что дает окрестность 49 пикселей.



(а) (б)

Рисунок 5.23 - Зависимость качества сегментации от размера пространственной окрестности s для сцены Salinas: индикаторы RI (а) и GCE (б).



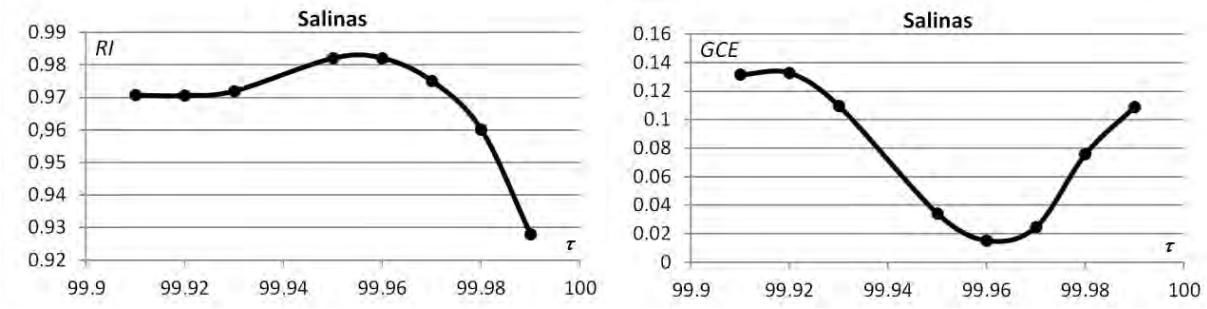
(а) (б)

Рисунок 5.24 - Зависимость качества сегментации от размера пространственной окрестности s для сцены Indian Pines: индикаторы *RI* (а) и *GCE* (б).

Целью следующего эксперимента было изучение зависимости качества сегментации от порогового параметра τ (в процентах), используемого при оценке размерности. Результаты этого эксперимента показаны на рисунках 5.25 и 5.26.

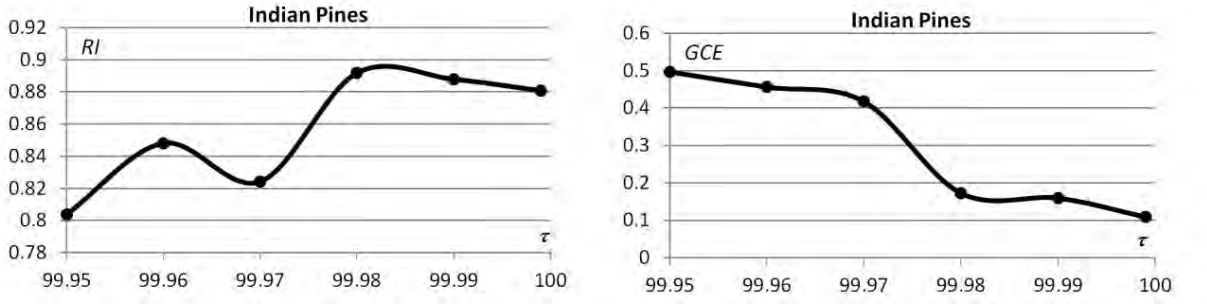
Как видно, исследуемый параметр существенно влияет на качество сегментации. Для изображения Salinas оптимальное значение находится около 99,96%. Для изображения Indian Pines трудно сделать однозначный выбор значений выше 99,98%, так как с ростом порога оба показателя уменьшаются.

Для сравнения изучаемого метода с альтернативными методами обратимся к работе [211*], в которой использовались те же показатели качества. По сравнению с классическими методами, описанными в [211*], для сцены Salinas исследуемый метод показывает преимущество по *RI*: 0,9821 против 0,942, 0,982 и 0,9275 для методов [82], основанных на кластеризации, наращивании областей и преобразовании водораздела, соответственно, с ограничением *GCE* <0,15.



(а) (б)

Рисунок 5.25 - Зависимость качества сегментации от порога τ (%), используемого для оценки размерности сцены Salinas: показатели *RI* (а) и *GCE* (б).



(а) (б)

Рисунок 5.26 - Зависимость качества сегментации от порога τ (%), используемого для оценки размерности сцены Indian Pines: индикаторы *RI* (а) и *GCE* (б).

Таким образом, в настоящем подразделе представлен и исследован метод сегментации гиперспектральных изображений, основанный на разбиении изображения на области, однородные с точки зрения размерности спектральных данных. Определены лучшие параметры исследуемого метода; показаны преимущества перед классическими методами сегментации гиперспектральных изображений.

5.7 Визуализация признаковых пространств на основе спектральных и текстурных признаков

Визуализация признаковых пространств позволяет исследователю получить представление о характере распределения данных, выбрать более эффективные методы решения прикладных задач и выявить потенциальные проблемы. Задача визуализации данных приобретает особую актуальность при анализе гиперспектральных изображений в силу как высокой размерности признаковых

пространств, так и использования различных модальностей, например, спектральных и текстурных характеристик изображений.

Распространенные методы визуализации многомерных данных не позволяют выполнять визуализацию с учетом разнородных систем признаков, используя, например, одновременно признаки переменной и фиксированной размерности, комбинировать признаки с различными расстояниями и т. д. Однако предложенная в подразделе 3.3 схема слияния признаков позволяет использовать разнородные системы признаков для решения различных задач.

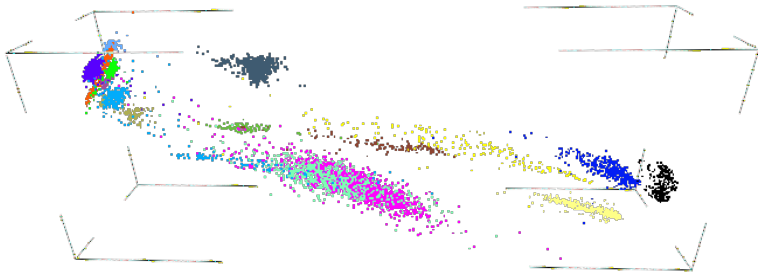
В настоящем подразделе указанная схема слияния используется для решения задачи визуализации пространства признаков гиперспектральных данных на основе спектральных и текстурных характеристик, различие для которых может измеряться с использованием различных функций расстояния. Приведенные результаты были описаны в работах автора [199*, 350*].

Напомним, что предложенная в подразделе 3.3 схема слияния признаков [200*] основана на переходе от базовых признаков к промежуточным представлениям, однородным с точки зрения используемых метрик. Как и ранее, для формирования промежуточного представления используется неявный переход в евклидово пространство. После слияния промежуточных представлений формируется выходное евклидово пространство с размерностью 2 или 3, что позволяет осуществлять интерактивную визуализацию на плоскости или в трехмерном пространстве. После отображения данных в пространство низкой размерности запускается интерактивная визуализация данных [47], позволяющая выполнять масштабирование, поворот и смещение точки наблюдения, а также цветовое кодирование данных при наличии тематической классификации данных.

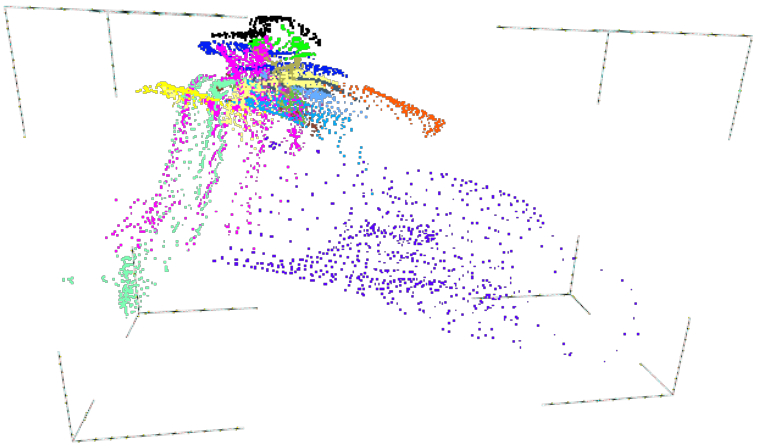
В настоящем разделе в качестве исходных систем признаков приняты спектральные и текстурные признаки. Спектральные признаки используются со следующими мерами рассогласования: спектральный угол (SAM) [147] и дивергенция Хеллингера (HD) [105]. Текстурные признаки (контрастность, корреляция, энергия и однородность) основаны на основе матрицы совместной встречаемости [99] (описаны в п. 3.3.4) и используются с евклидовым расстоянием.

На следующих рисунках 5.27 и 5.28 мы приводим несколько примеров визуализации пространства признаков в трехмерном пространстве для изображений

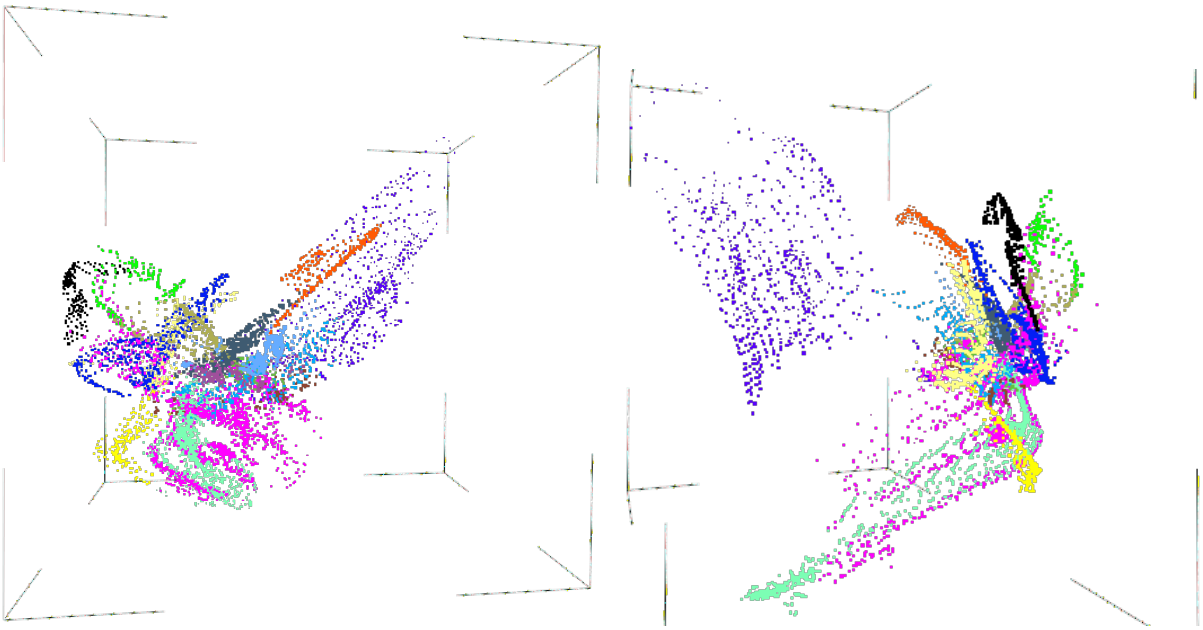
Salinas и Indian Pines [17, 120]. В частности, мы приводим примеры спектральных признаков (рисунки 5.27(а), 5.28(а)), текстурных признаков (рисунки 5.27(б), 5.28 (б)) и объединенных признаков (рисунки 5.27 (в, г), 5.28 (в, г)).



(а)



(б)

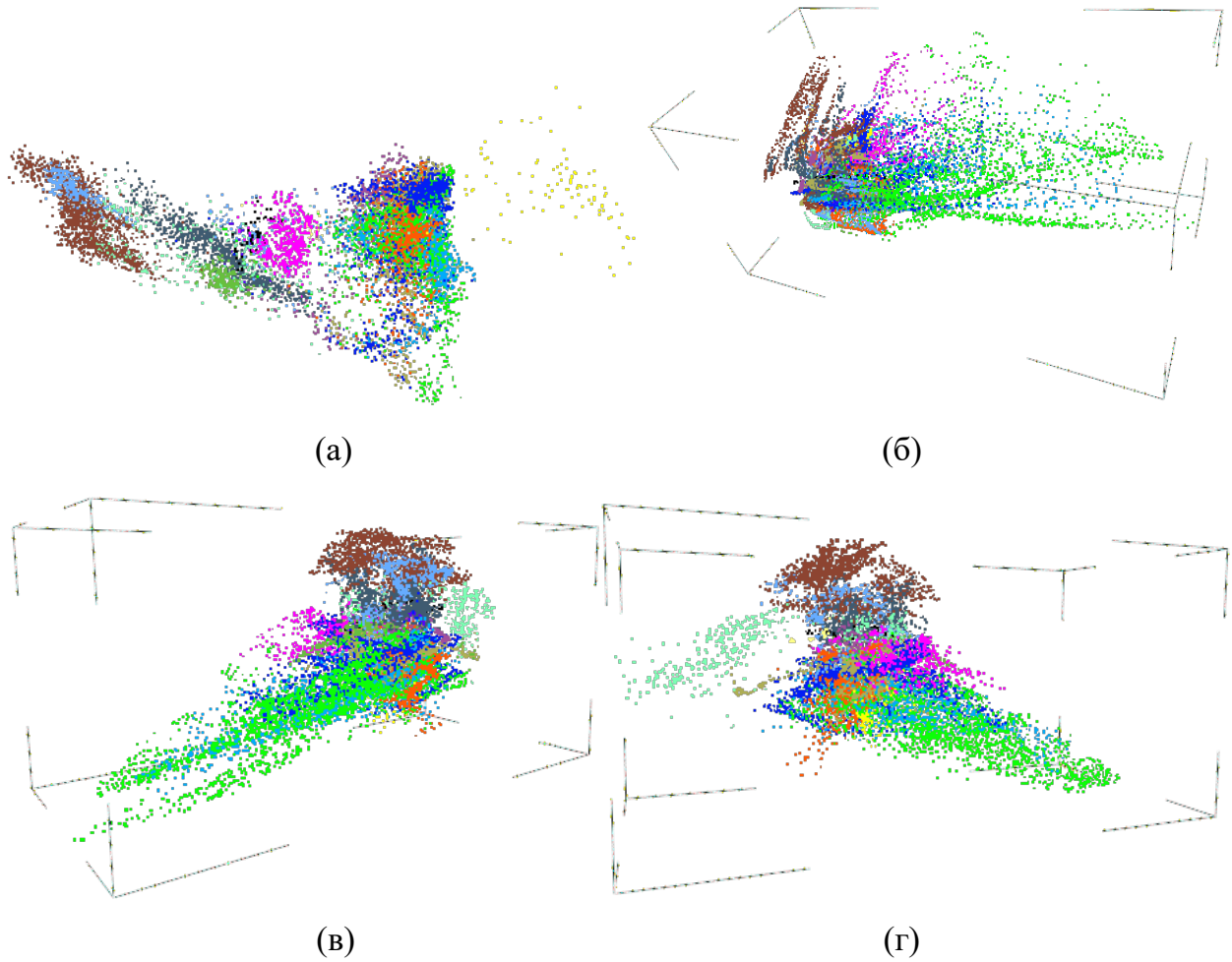


(в) (г)

Рисунок 5.27 - Пример 3D-визуализации пространства признаков для изображения Salinas: спектральные признаки λ=1 (а), текстурные признаки λ=0 (б), объединенные признаки λ=0,75 (в, г).

На приведенных рисунках λ показывает вклад спектральных и текстурных признаков в решение. При $\lambda = 1$ учитываются только спектральные признаки, при $\lambda = 0$ решение строится на основе текстурных признаков, а при промежуточных значениях на решение влияют обе системы признаков. Пиксели на рисунках показаны в трехмерном пространстве разными цветами, соответствующими разным тематическим классам истинной классификации.

Как видно из приведенных рисунков, предложенный метод визуализации дает нам представление о сложной структуре данных гиперспектрального изображения. Мы наблюдаем классы, которые кажутся трудноразделимыми, особенно на двумерных проекциях сформированных 3D представлений. Представляется важной доступная в [47] возможность масштабирования и изменения точки обзора и угла обзора.



(а) (б)

(в) (г)

Рисунок 5.28 - Пример визуализации пространства признаков для изображения Indian Pines: спектральные признаки λ=1 (а), текстурные признаки λ=0 (б), объединенные признаки λ=0,75 (в, г)

5.8 Формирование псевдоцветовых представлений и гиперспектральных изображений

По причине высокой размерности спектрального пространства гиперспектральных спутниковых снимков непосредственная визуализация таких изображений невозможна. Поэтому визуализацию гиперспектральных снимков выполняют, выбирая для отображения отдельные каналы таких изображений (см., например, среды ENVI и MultiSpec), используя линейную комбинацию каналов (см., например, среду HyperCube) или путем отображения исходных данных в пространства малой размерности с использованием метода главных компонент [54, 291].

Хотя метод главных компонент для решения такой задачи используется чаще других, тот факт, что метод является линейным, ограничивает его возможности учесть нелинейные характеристики гиперспектральных изображений. По этой причине исследователями предпринимаются попытки применения нелинейных методов для решения задачи визуализации, но возможности их применения для интерактивной визуализации ограничивает низкая скорость работы и высокие требования к объему памяти таких методов [54].

В отличие от альтернативных подходов, описываемый в настоящем подразделе метод визуализации мульти- и гиперспектральных изображений [352*] основан на оригинальных подходах к формированию признаков ГСИ, изложенных в предыдущих разделах работы. Отличительной особенностью метода является то, что расстояния между пикселями визуализируемого изображения в цветовом пространстве аппроксимируют рассогласования между пикселями гиперспектрального изображения в исходном спектральном пространстве [208*]. Указанная особенность метода является следствием того, что процедура формирования, лежащая в основе метода, действует по принципу сохранения попарных расстояний. В контексте решаемой задачи подразумевается сохранение расстояний между пикселями изображения в исходном и формируемом трехмерном пространстве, отображаемом на цветовое пространство визуализации.

Общая схема метода показана на рисунке 5.29.

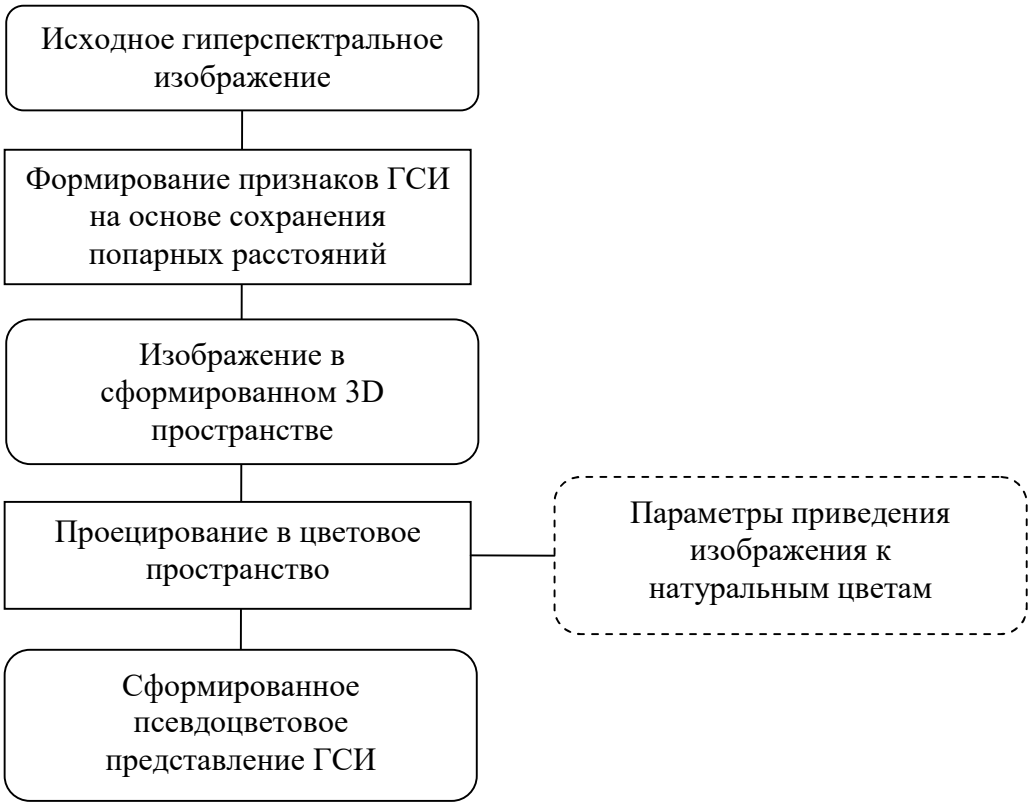
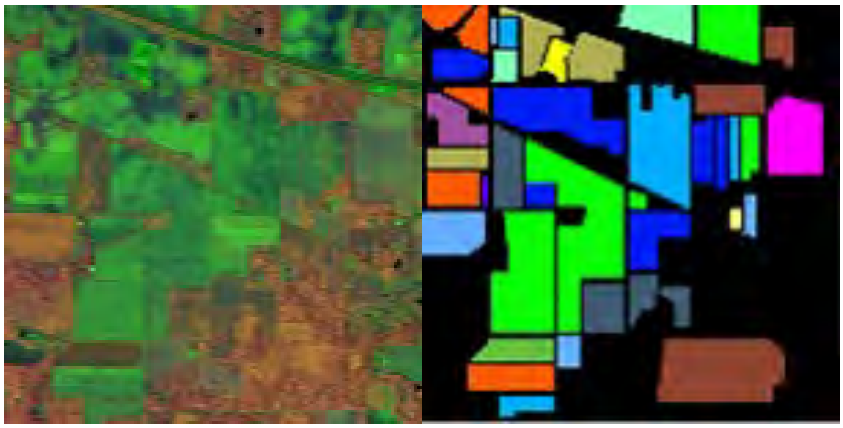


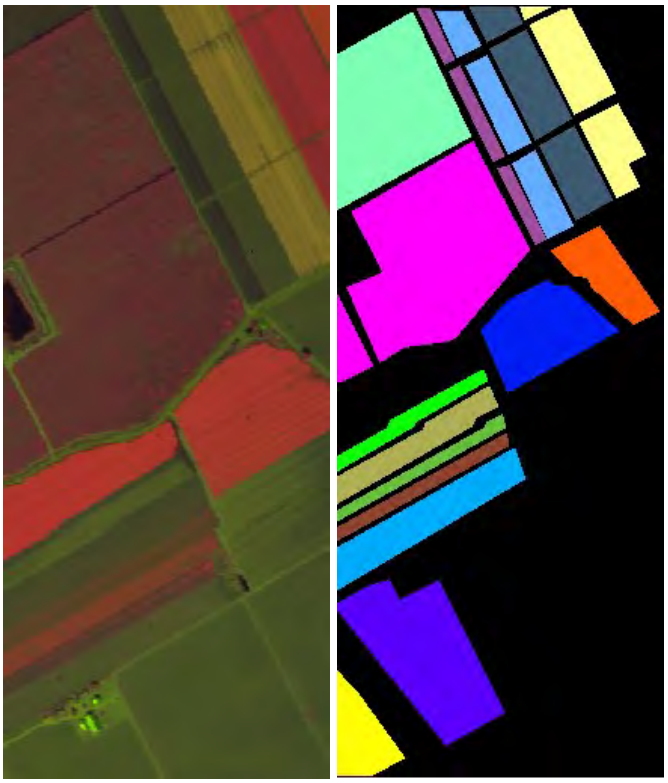
Рисунок 5.29 - Схема метода формирования псевдоцветовых представлений гиперспектральных изображений

Примеры сформированных в результате работы метода изображений показаны на рисунках 5.30, Рисунок 5.31.



(а) (б)

Рисунок 5.30 - Визуализация гиперспектрального изображения Indian pines: (а) представление ГСИ в псевдоцветах; (б) соответствующее изображение, содержащее истинную классификацию: разные классы подстилающей поверхности показаны разными цветами, неклассифицированные области - черным цветом.



(в) (г)

Рисунок 5.31 - Визуализация гиперспектрального изображения Salinas:
(а) представление ГСИ в псевдоцветах; (б) соответствующее изображение, содержащее истинную классификацию: разные классы подстилающей поверхности показаны разными цветами, неклассифицированные области - черным цветом.

Выводы и результаты по главе 5

В главе рассмотрено несколько практических приложений, демонстрирующих возможности изложенных в работе методов и алгоритмов. В частности, представлено решение задач классификации гиперспектральных данных дистанционного зондирования высокого разрешения, полученных с беспилотного летательного аппарата, классификации гиперспектральных данных дорожной обстановки, классификации медицинских изображений тканей головного мозга, классификации растительности по данным наземной гиперспектральной съемки, а также визуализации признаковых пространств и генерации псевдоцветовых представлений гиперспектральных изображений. Показано, что для многих из представленных задач удалось повысить эффективность их решения.

Разработаны алгоритмы кластеризации данных, основанные на использовании неевклидовых метрик, в частности, дивергенции Хеллингера и угла Бхаттачария.

Показано преимущество предложенного алгоритма перед альтернативным алгоритмом k-медоидов.

Разработан метод сегментации ГСИ, основанный на оценке размерности в пространственных областях. Показано его преимущество перед классическими методами сегментации.

Заключение

В диссертационной работе разработаны методы, алгоритмы и программные средства вычислительно эффективного адаптивного формирования информативных признаков цифровых гиперспектральных изображений на основе функций расстояния, обеспечивающих повышенное качество решения прикладных задач.

В работе были получены следующие основные результаты:

1. Разработан комплекс алгоритмических средств формирования информативных признаков ГСИ, основанных на принципе сохранения попарных расстояний, а именно: спектральных углов, дивергенции спектральной информации, дивергенции Хеллингера, угла Бхаттачария, а также спектральной корреляции. Предложенные методы позволили существенно сократить размерность гиперспектральных данных (до размерности 10…40), обеспечив лучшее качество классификации по сравнению с альтернативными методами. Кроме того, качество классификации по сформированным данным оказалось выше качества классификации по исходным данным в более чем 70% случаев.

2. Разработан метод формирования информативных признаков ГСИ с учетом локального пространственного контекста на основе порядковых статистик. Предложенный метод обеспечил прирост качества на 3,3-9,1% по сравнению с базовой версией без пространственного контекста, а также на 1,2% – 6,3% по сравнению с методом на основе оконных функций (при использовании фиксированного разбиения). Кроме того, метод позволил использовать неевклидовы расстояния, обеспечив дополнительный прирост качества на 3,2-7,3%.

3. Разработан метод слияния разнородных признаков на основе приведения исходных систем признаков к однородным промежуточным представлениям с последующим объединением этих представлений и снижением размерности. Предложенный метод позволил получить прирост качества классификации на 1.2-20% по сравнению с «глубокими» и спектральными признаками.

4. Разработаны новые алгоритмы и численные методы, обеспечивающие ускорение формирования признаков ГСИ, а именно: алгоритм построения списка опорных узлов и численный метод ускорения формирования признаков на основе опорных узлов, а также модификация метода, учитывающая характеристики узлов

как во входном, так и в формируемом пространствах; алгоритм построения очереди опорных узлов с приоритетом и соответствующая модификация метода ускорения формирования признаков на основе очередей с приоритетом.

Предложен комбинированный метод ускорения формирования признаков на основе стохастического градиентного спуска, очередей опорных узлов с приоритетом и алгоритмов многомерной интерполяции.

Предложенные методы отличаются от известных повышенной производительностью. В частности, предложенный метод на основе очередей с приоритетом позволяет пользователю выбирать между качеством и скоростью обработки. Показано значительное (более чем на 50%) сокращение времени обработки по сравнению с алгоритмом стохастического градиентного спуска без потери качества.

5. Разработаны программные средства формирования признаков ГСИ с использованием параллельной многопоточной реализации для современных графических процессоров NVIDIA и AMD. При использовании ГП NVIDIA RTX 3090 достигнуто ускорение от 145 до 597 раз по сравнению с однопоточной версией, работающей на ЦП.

6. Разработан алгоритм кластеризации гиперспектральных данных при использовании неевклидовых метрик, а также метод сегментации гиперспектральных изображений на основе оценки размерности в пространственных областях.

Получены решения следующих прикладных задач анализа ГСИ: классификация гиперспектральных данных дистанционного зондирования высокого разрешения, полученных с беспилотного летательного аппарата; классификация гиперспектральных данных дорожной обстановки; классификация гиперспектральных изображений мозга человека; визуализация признаковых пространств; формирование псевдоцветовых представлений ГСИ.

По мнению автора, полученные результаты в совокупности представляют собой решение научной проблемы, являющееся вкладом в развитие информационно-технологической отрасли России. Предложенное в диссертации целостное решение для адаптивного признакового описания гиперспектральных данных позволяет не только устранить избыточность таких данных, но и повысить точность решения прикладных задач, а также существенно сократить вычислительные затраты.

Центральным результатом работы является доказательство эффективности перехода от стандартных евклидовых метрик и глобальных линейных преобразований к неевклидовым функциям расстояния в сочетании с локальным пространственным контекстом. Это обеспечивает стабильный прирост качества на широком спектре прикладных задач: от анализа данных дистанционного зондирования до обработки медицинских гиперспектральных изображений. Практическая значимость подтверждена разработанным программным обеспечением, готовым к интеграции в существующие геоинформационные и медицинские диагностические системы.

Результаты работы можно рекомендовать к использованию следующим образом:

1. Использовать разработанный комплекс алгоритмических средств в задачах, где критична высокая точность анализа, и требуется интерпретируемость признаков, при этом обучающие выборки ограничены по объему.

2. Использовать метод на основе очередей с приоритетом в случаях, когда необходимо достижение компромисса между точностью и скоростью обработки данных, а также при невозможности использования средств аппаратного ускорения.

3. Внедрить параллельные реализации для GPU в центрах обработки, работающих с большими массивами гиперспектральных данных, так как достигаемое ускорение делает обработку оперативной даже для больших наборов данных.

Дальнейшие исследования целесообразно развивать в следующих направлениях:

1. Адаптация и интеграция разработанных решений для работы с данными других модальностей, в частности, с данными, получаемыми радарными и лидарными системами.

2. Разработка гибридных схем, объединяющих предложенные решения с современными нейросетевыми архитектурами на основе использования полученных признаков в качестве входных каналов для трансформерных, графовых, KAN-архитектур и др. с целью дальнейшего повышения семантической точности.

3. Распространение разработанных решений для обработки данных гиперспектральной видеосъемки в реальном масштабе времени.

4. Создание для научного сообщества специализированных библиотек с открытым исходным кодом, реализующих разработанные методы и алгоритмы.

Решение перечисленных задач позволит перейти от статического анализа гиперспектральных изображений к интеллектуальным системам непрерывного мониторинга с адаптивной перестройкой признакового пространства в реальном масштабе времени.

Список литературы

1. Abou-Moustafa, K.T. Note on Metric Properties for Some Divergence Measures: The Gaussian Case / K.T. Abou-Moustafa, F.P. Ferrie // JMLR: Workshop and Conference Proceedings. – 2012. - Vol. 25. - P. 1-15.
2. Achanta, R. SLIC Superpixels Compared to State-of-the-art Superpixel Methods / R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Süsstrunk // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2012. – Vol. 34(11). – P. 2274 – 2282. DOI: 10.1109/TPAMI.2012.120
3. Adams, J. B. Viking Lander I: a New Map of Rock and Soil Types / J.B. Adams, M.O. Smith, P.E. Johnson // Lunar and planetary science. – 1985. – Vol. XVI - P. 3–4.
4. Adams, J.B. Simple models for complex natural surfaces: a strategy for the hyperspectral era of remote sensing / J.B. Adams, M.O. Smith, A.R. Gillespie // 12th Canadian Symposium on Remote Sensing Geoscience and Remote Sensing Symposium. – 1989. - P. 16-21.
5. Agrawal, N. Dimensionality Reduction on Hyperspectral Data Set / N. Agrawal, K. Verma // 2020 First International Conference on Power, Control and Computing Technologies (ICPC2T). – 2020. - P. 139-142. DOI: 10.1109/ICPC2T48082.2020.9071461.
6. Al-Ani, A. Feature Subset Selection Using Ant Colony Optimization / A. Al-Ani // International Journal of Computational Intelligence. - 2005. - Vol. 2(1). - P. 53-58.
7. Alomari, A. Hyperspectral Data Compression and Red Edge Indices / A. Alomari. – London: LAP Lambert Academic Publishing, 2011. - 88 p. - ISBN: 978-3-8454-7222-5.
8. Altmann, Y. Supervised nonlinear spectral unmixing using a post-nonlinear mixing model for hyperspectral imagery / Y. Altmann, A. Halimi, N. Dobigeon, J.-Y. Tourneret // IEEE Transactions on Image Processing. – 2012. - Vol. 21(6). - P. 3017-3025.
9. Arthur, D. K-means++: the advantages of careful seeding / D. Arthur, S. Vassilvitskii // Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'07). – 2007. – P. 1027-1035. DOI: 10.1145/1283383.1283494.
10. Bachmann, C.M. Exploiting manifold geometry in hyperspectral imagery / C.M. Bachmann, T.L. Ainsworth, R.A. Fusina // IEEE Transactions on Geoscience and Remote Sensing. – 2005. – Vol. 43(3). – P. 441–454.

11. Bachmann, C.M. Improved Manifold Coordinate Representations of Large-Scale Hyperspectral Scenes / C.M. Bachmann, T.L. Ainsworth, R.A. Fusina // IEEE Transactions on Geoscience and Remote Sensing. - 2006. – Vol. 44 (10). – Vol. 2786-2803.
12. Baldi, P. Neural networks and principal component analysis: Learning from examples without local minima / P. Baldi, K. Hornik // Neural Networks. – 1989. - Vol. 2(1). - P. 53-58,
13. Ballard, D.H. Modular Learning in Neural Networks / D.H. Ballard // Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87). - 1987. - Vol. 1. – 279-284.
14. Barnes, J. A hierarchical O(N log N) force-calculation algorithm / J. Barnes, P. Hut // Nature. – 1986. – Vol. 324(4). – P. 446–449.
15. Barnes, R. Priority-Flood: An Optimal Depression-Filling and Watershed-Labeling Algorithm for Digital Elevation Models / R. Barnes, C. Lehman, D. Mulla // Computers & Geosciences. – 2014. – Vol. 62. – P. 117-127.
16. Basterretxea, K. HSI-Drive: A Dataset for the Research of Hyperspectral Image Processing Applied to Autonomous Driving Systems / K. Basterretxea, V. Martínez, J. Echanobe, J. Gutiérrez–Zaballa and I. Del Campo // 2021 IEEE Intelligent Vehicles Symposium (IV). – 2021. - P. 866-873. DOI: 10.1109/IV48863.2021.9575298.
17. Baumgardner, M.F. 220 Band AVIRIS Hyperspectral Image Data Set: June 12, 1992 Indian Pine Test Site 3 / M.F. Baumgardner, L.L. Biehl, D.A. Landgrebe // Purdue University: сайт. – URL: https://purr.purdue.edu/publications/1947/1 (дата обращения: 06.10.2024).
18. Belkin, M. Laplacian eigenmaps and spectral techniques for embedding and clustering / M. Belkin, P. Niyogi // Advances in Neural Information Processing Systems. – 2001. – Vol. 14. – P. 585-591.
19. Bentley, J.L. Multidimensional binary search trees used for associative searching / J.L. Bentley // Communications of the ACM. - 1975. – Vol. 18(9). – P. 509-517.
20. Berthier, M. Binary codes k-modes clustering for HSI segmentation / M. Berthier, S. El Asmar, C. Frélicot // Proceedings of the 2016 IEEE 12th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP). – 2016. – P. 1-5.

21. Beucher, S. Use of watersheds in contour detection / S. Beucher, C. Lantuejoul // Proceedings of the International Workshop Image Processing, Real-Time Edge and Motion Detection / Estimation.– 1979. – P. 1-12.
22. Bhattacharyya, A.K. On a measure of divergence between two statistical populations defined by their probability distributions / A.K. Bhattacharyya // Bulletin of the Calcutta Mathematical Society. – 1943. - Vol. 35. - P. 99-109,
23. Bioucas-Dias, J.M. Hyperspectral Unmixing Overview: Geometrical, Statistical, and Sparse Regression-Based Approaches / J.M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, J. Chanussot // IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. – 2012. - Vol. 5(2). - P. 354-379.
24. Blumenson, L.E. A derivation of n-dimensional spherical coordinates / L.E. Blumenson // American Mathematical Monthly. - 1960. – Vol. 67(1). – P. 63–66.
25. Blumenson, L.E. CCSM: cross correlogram spectral matching / L.E. Blumenson, W. Bakker // International Journal of Remote Sensing. – 1997. – Vol. 18(5). – P. 1197-1201.
26. Borchers, B. CSDP, a C library for semidefinite programming / B. Borchers // Optimization Methods and Software. – 1999. – Vol. 11(1). – P. 613–623.
27. Borg, I. Modern Multidimensional Scaling: Theory and Applications / I. Borg, P. Groenen - New York: Springer, 2005. – 614 p.
28. Borhani, M. Kernel Multivariate Spectral-Spatial Analysis of Hyperspectral Data / M. Borhani, H. Ghassemian // IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. – 2015. – Vol. 8(6). – N. 7056457. - P. 2418-2426.
29. Borisov, A. Performance Evaluation of Gradient-based Dimensionality Reduction Methods on Different Devices / A. Borisov, **E. Myasnikov** // 2020 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT 2020). IEEE Xplore Digital Library. — 2020. — P. 428-431.
30. Borisov, A., Dimensionality reduction using GPU-accelerated gradient descent / A. Borisov, **E. Myasnikov** // CEUR Workshop Proceedings. – 2020. - Vol. 2667. - P. 248-250.
31. Bourlard, H. Auto-association by multilayer perceptrons and singular value decomposition / H. Bourlard, Y. Kamp // Biological Cybernetics. – 1988. - Vol. 59(4-5). - P. 291-294.

32. Breiman, L. Random Forests / L. Breiman // Machine Learning. – 2001. - Vol. 45(1). – P. 5-32.
33. Broomhead, D.H. Multivariable Functional Interpolation and Adaptive Networks / D.H. Broomhead, D. Lowe // Complex Systems. – 1988. – Vol. 2. – P. 321-355.
34. Campana-Olivo, R. Parallel implementation of nonlinear dimensionality reduction methods applied in object segmentation using CUDA in GPU / R. Campana-Olivo, V. Manian // Proceedings of SPIE - The International Society for Optical Engineering. - 2011. – Vol. 8048. – P. 80480R.
35. Camps-Valls, G. Composite kernels for hyperspectral image classification / G. Camps-Valls, L. Gomez-Chova, J. Munoz-Mari, J. Vila-Frances, J. Calpe-Maravilla // IEEE Geoscience and Remote Sensing Letters. – 2006. - Vol. 3(1). - P. 93-97. DOI: 10.1109/LGRS.2005.857031.
36. Cariou, C. Unsupervised nearest neighbors clustering with application to hyperspectral images / C. Cariou, K. Chehdi // IEEE Journal of Selected Topics in Signal Processing. – 2015. – Vol. 9(6). – P. 1105-1116.
37. Carminati , L. UmapPizza. FPGA implementation of the computationally-intesive part of UMAP / L. Carminati, A. Cettolin, G. Colomban, M.D. Santambrogio // GitHub: сайт. – URL: https://github.com/Giocol/UMAPPizza (дата обращения: 06.10.2024).
38. Chalmers, M. A linear iteration time layout algorithm for visualising high-dimensional data / M. Chalmers // Proceedings of IEEE Visualization. – 1996. – pp. 127–132.
39. Chang, C.-I. Hyperspectral Data Processing: Algorithm Design and Analysis / C.-I. Chang. – Berlin: John Wiley & Sons, 2013. - 1168 p. - ISBN: 0471690562.
40. Chang, C.I. Spectral information divergence for hyperspectral image analysis / C.I. Chang // Proceedings of the IEEE 1999 International Geoscience and Remote Sensing Symposium, IGARSS'99. – 1999. – Vol. 1. - P. 509-511.
41. Chang, T. Texture analysis and classification with tree-structured wavelet transform / T. Chang, C.-C.J. Kuo // IEEE Transactions on Image Processing. – 1993. - Vol. 2(4). - P. 429-441. DOI: 10.1109/83.242353.
42. Chen, C. Classification of hyperspectral data using a multi-channel convolutional neural network / C. Chen, J.J. Zhang, C.H. Zheng, Q. Yan, L.N. Xun // Proceedings of the 14th International Conference on Intelligent Computing (ICIC). – 2018. - P. 81–92.

43. Chen, C. Hyperspectral classification based on spectral–spatial convolutional neural networks / C. Chen, F. Jiang, C. Yang, S. Rho, W. Shen, S. Liu, Z. Liu // Engineering Applications of Artificial Intelligence. – 2018. - Vol. 68. - P. 165–171. DOI: 10.1016/j.engappai.2017.10.015.
44. Chen, X. Spectral mixture analysis of hyperspectral data acquired using a tethered balloon / X. Chen, L. Vierling // Remote Sensing of Environment. – 2006. - Vol. 103. - P. 338–350.
45. Chen, Y. Deep Learning Ensemble for Hyperspectral Image Classification / Y. Chen, Y. Wang, Y. Gu, X. He, P. Ghamisi, X. Jia // IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. – 2019. - Vol. 12(6). - P. 1882-1897. DOI: 10.1109/JSTARS.2019.2915259.
46. Chen, Y. Deep learning-based classification of hyperspectral data / Y. Chen, Z. Lin, X. Zhao, G. Wang, Y. Gu, // International Journal of Applied Earth Observation and Geoinformation. – 2014. – Vol. 7. – P. 2094–2107.
47. Cignoni, P. MeshLab: an Open-Source Mesh Processing Tool / P. Cignoni, M. Calleri, M. Corsini, M. Dellepiane, F. Ganovelli, G.Ranzuglia // Sixth Eurographics Italian Chapter Conference. – 2008. - P. 129-136.
48. Close, R. Hyperspectral unmixing using macroscopic and microscopic mixture models / R. Close, P. Gader, J. Wilson, A. Zare // Journal of Applied Remote Sensing. – 2014. – Vol. 8. - P. 083642.
49. Comon, P. Independent component analysis, a new concept? / P. Comon // Signal processing. – 1994. – Vol. 36(3). - P. 287–314.
50. Cook, J. Visualizing Similarity Data with a Mixture of Maps / J. Cook, I. Sutskever, A. Mnih, G. Hinton // Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics, PMLR. – 2007. - Vol. 2. – P. 67-74.
51. Corel image dataset // digitalriver: сайт. – URL: http://corel.digitalriver.com (дата обращения: 06.10.2024).
52. Corel image features // UC Irvine Machine Learning Repository: сайт. – URL: https://archive.ics.uci.edu/dataset/119/corel+image+features (дата обращения: 06.10.2024).
53. Corley, I. Revisiting pre-trained remote sensing model benchmarks: resizing and normalization matters / I. Corley, C. Robinson, R. Dodhia, J.M. Lavista Ferres, P.

Najafirad // arXiv. - 2023. – URL: https://arxiv.org/abs/2305.13456 (дата обращения: 06.10.2024). DOI: 10.48550/arXiv.2305.13456.

54. Cui, M. Interactive hyperspectral image visualization using convex optimization / M. Cui, A. Razdan, J. Hu, P. Wonka // IEEE Transactions on Geoscience and Remote Sensing. - 2009. – Vol. 47(6). – P. 1673-1684.
55. cuBLAS // NVIDIA Documentation Hub: сайт. – URL: https://docs.nvidia.com/cuda/cublas/index.html (дата обращения: 06.10.2024).
56. Curse of dimension // Encyclopedia of Mathematics: сайт. – URL: http://encyclopediaofmath.org/index.php?title=Curse_of_dimension&oldid=54772 (дата обращения: 06.10.2024).
57. cuSOLVER API Reference // NVIDIA Documentation Hub: сайт. – URL: https://docs.nvidia.com/cuda/cusolver/index.html (дата обращения: 06.10.2024).
58. Cybenko, G. Approximation by Superpositions of a Sigmoidal Function / G. Cybenko // Mathematics of Control, Signals, and Systems. – 1989. - Vol. 2. - P. 303-314.
59. Dasgupta, S. An elementary proof of a theorem of Johnson and Lindenstrauss / S. Dasgupta, A. Gupta // Random Structures and Algorithms. – 1999. – Vol. 22(1). - P. 60-65.
60. De Leeuw, J. Applications of convex analysis to multidimensional scaling / J. de Leeuw // Recent Developments in Statistics. – Amsterdam: North Holland Publishing Company, 1977. - P. 133–145.
61. De Leeuw, J. Convergence of the Majorization Method for Multidimensional Scaling / J. de Leeuw // Journal of Classification. – 1988. – Vol. 5. – P. 163-180.
62. De Leeuw, J. Multidimensional Scaling Using Majorization: SMACOF in R / J. de Leeuw, P. Mair // UCLA: Department of Statistics: сайт. – URL: https://escholarship.org/uc/item/9z64v481 (дата обращения: 06.10.2024).
63. Demartines, P. Curvilinear Component Analysis: A Self-Organizing Neural Network for Nonlinear Mapping of Data Sets / P. Demartines, J. Hérault // IEEE Transactions on Neural Networks. - 1997. – Vol. 8(1). – P. 148-154. DOI: 10.1109/72.554199.
64. Devassy, B.M. Dimensionality reduction and visualisation of hyperspectral ink data using t-SNE / B. M. Devassy, S. George // Forensic Science International. - 2020. – Vol. 311. – 110194. DOI: 10.1016/j.forsciint.2020.110194.

65. Deza, M. Encyclopedia of Distances / M. Deza, E. Deza. — Heidelberg: Springer, 2016. – 756 p.
66. Ding, S. Dimension Reduction of Local Manifold Learning Algorithm for Hyperspectral Image Classification / S. Ding, L. Chen, J. Li // Intelligent Image and Video Interpretation: Algorithms and Applications. - 2013. - P. 217-231. - DOI: 10.4018/978-1-4666-3958-4.ch009.
67. Dobigeon, N. Nonlinear Unmixing of Hyperspectral Images: Models and Algorithms / N. Dobigeon, J.-Y. Tourneret, C. Richard, J.C.M. Bermudez, S. McLaughlin, A.O. Hero // IEEE Signal Processing Magazine. – 2014. - Vol. 31(1). - P. 82-94. DOI: 10.1109/MSP.2013.2279274.
68. Du, H. An FPGA implementation of parallel ICA for dimensionality reduction in hyperspectral images / H. Du, H. Qi // International Geoscience and Remote Sensing Symposium (IGARSS). – 2004. – Vol. 5. - P. 3257-3260.
69. Du, P. Dimensionality reduction and feature extraction from hyperspectral remote sensing imagery based on manifold learning / P. Du, X. Wang, K. Tan, J. Xia // Wuhan Daxue Xuebao (Xinxi Kexue Ban)/Geomatics and Information Science of Wuhan University. – 2011. – Vol. 36(2). - P. 148-152.
70. Du, Q. Hyperspectral Image Compression Using JPEG2000 and Principal Component Analysis / Q. Du, J.E. Fowler // IEEE Geoscience and Remote Sensing Letters. – 2007. - Vol. 4(2). - P. 201-205. DOI: 10.1109/LGRS.2006.888109.
71. Duchi, J. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization / J. Duchi, E. Hazan, Y. Singer // The Journal of Machine Learning Research. – 2011. - Vol. 12. - P. 2121–2159.
72. Ella, B. Random projection in dimensionality reduction: Applications to image and text data / B. Ella, M. Heikki // Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. - 2001. - P. 245–250. DOI: 10.1145/502512.502546.
73. Estep, L. Band-Moment Compression of Aviris Hyperspectral Data and Its Use in the Detection of Vegetation Stress / L. Estep, B. Davis // NASA Technical Reports Server. 2001. – URL: https://ntrs.nasa.gov/citations/20040058117 (дата обращения: 06.10.2024).

74. Fabelo, H. In-Vivo Hyperspectral Human Brain Image Database for Brain Cancer Detection / H. Fabelo, S. Ortega, A. Szolna, D. Bulters, J.F. Piñeiro, S. Kabwama // IEEE Access. – 2019. - Vol. 7. P. 39098-39116. DOI: 10.1109/ACCESS.2019.2904788.
75. Fan, W. Comparative study between a new nonlinear model and common linear model for analysing laboratory simulated forest hyperspectral data / W. Fan, B. Hu, J. Miller, M. Li // International Journal of Remote Sensing. – 2009. - Vol. 30(11). - P. 2951-2962.
76. Fauvel, M. A spatial–spectral kernel-based approach for the classification of remote-sensing images / M. Fauvel, J. Chanussot, J.A. Benediktsson // Pattern Recognition. - 2012. - Vol. 45(1). - P. 381-392. DOI: 10.1016/j.patcog.2011.03.035.
77. Felzenszwalb, P.F. Efficient graph-based image segmentation / P.F. Felzenszwalb, D.P. Huttenlocher // International Journal of Computer Vision. - 2004. - Vol. 59. - P. 167-181.
78. Finkel, R.A. Quad trees a data structure for retrieval on composite keys / R.A. Finkel, J.L. Bentley // Acta Informatica. - 1974. - Vol. 4(1). - P. 1-9. DOI: 10.1007/BF00288933.
79. Firsov, N. HyperKAN: Kolmogorov-Arnold Networks make Hyperspectral Image Classifiers Smarter / N. Firsov, **E. Myasnikov**, V. Lobanov [и др.] // Sensors. – 2024. – Vol. 24(23). - 7683.
80. Fisher, R.A. The Use of Multiple Measurements in Taxonomic Problems / R.A. Fisher // Annals of Eugenics. – 1936. – Vol. 7(2). P. 179-188.
81. Fruchterman, T. Graph Drawing by Force-directed Placement. / T. Fruchterman, E. Reingold // Software – Practice and Experience. - 1991. - Vol. 21(11). - P. 1129-1164.
82. Fu, K.S. A survey on image segmentation / K.S. Fu, J.K. Mui // Pattern Recognition. – 1981. – Vol. 13(1). – P. 2-16.
83. Gan, Z. Efficient implementation of the Barnes-Hut octree algorithm for Monte Carlo simulations of charged systems / Z. Gan, Z. Xu // Science China Mathematics. - 2014. - Vol. 57. - P. 1331–1340. DOI: 10.1007/s11425-014-4783-5.
84. Gao, K. Unsupervised meta learning with multiview constraints for hyperspectral image small sample set classification / K. Gao, B. Liu, X. Yu, A. Yu // IEEE Transactions on Image Processing. – 2022. – Vol. 31. – P. 3449-3462.

85. Gao, Q. Hyperspectral Image Classification Using Convolutional Neural Networks and Multiple Feature Learning / Q. Gao, S. Lim, X. Jia // Remote Sensing. – 2018. – Vol. 10. – P. 299.
86. Gashnikov, M.V. Compression Method for Real-Time Systems of Remote Sensing / M.V. Gashnikov, N.I. Glumov, V.V.Sergeyev // Proceedings 15th International Conference on Pattern Recognition. ICPR-2000. – 2000. - Vol. 3. - P. 228-231. DOI: 10.1109/ICPR.2000.903527.
87. Geelen, B. A compact snapshot multispectral imager with a monolithically integrated per-pixel filter mosaic / B. Geelen, N. Tack, A. Lambrechts // Proceedings of SPIE. Advanced Fabrication Technologies for Micro/Nano Optics and Photonics VII. – 2014. – Vol. 8974. – P. 89740L. DOI: 10.1117/12.2037607.
88. GeForce GTX 10 // NVIDIA Corporation: сайт. – URL: https://www.nvidia.com/ru-ru/geforce/10-series/ (дата обращения: 06.10.2024).
89. GeForce GTX 1070 vs RTX 3090 // Technical city. Сравнения, спецификации и рейтинги видеокарт и процессоров : сайт. – URL: https://technical.city/ru/video/GeForce-GTX-1070-protiv-GeForce-RTX-3090 (дата обращения: 06.10.2024).
90. Gorban, A.N. Beyond The Concept of Manifolds: Principal Trees, Metro Maps, and Elastic Cubic Complexes / A.N. Gorban, N.R. Sumner, A.Y. Zinovyev // Principal Manifolds for Data Visualization and Dimension Reduction. Lecture Notes in Computational Science and Enginee. – 2008. - Vol 58. – P. 219-237. DOI: 10.1007/978-3-540-73750-6_9 .
91. Goretta, N. An iterative hyperspectral image segmentation method using a cross analysis of spectral and spatial information / N. Goretta, G. Rabatel, C. Fiorio, C. Lelong, J.M. Roger // Chemometrics and Intelligent Laboratory systems. – 2012. – Vol. 117(1). – P. 213-223.
92. Granlund, G.H. In Search of a General Picture Processing Operator / G.H. Granlund // Computer Graphics and Image Processing. - 1978. – Vol. 8(2). – P. 155–173. DOI: 10.1016/0146-664X(78)90047-3.
93. Greengard, L. A fast algorithm for particle simulations / L. Greengard, V. Rokhlin // Journal of Computational Physics. – 1987. – Vol. 73. - P. 325-348.

94. Guerri, M.F. Deep learning techniques for hyperspectral image analysis in agriculture: A review, / M.F. Guerri, C. Distante, P. Spagnolo, F. Bougourzi, A. Taleb-Ahmed // ISPRS Open Journal of Photogrammetry and Remote Sensing. - 2024. - Vol. 12. - P. 100062. DOI: 10.1016/j.ophoto.2024.100062.
95. Gusev, A. Finite element mapping for spring network representations of the mechanics of solids / A. Gusev // Physical Review Letters. – 2004. – Vol. 93(2). – P. 034302.
96. Guttman, L. A General Nonmetric Technique for Fitting the Smallest Coordinate Space for a Configuration of Points / L. Guttman // Psychometrika. - 1968. – Vol. 33. - P. 469-506.
97. Halimi, A. Nonlinear unmixing of hyperspectral images using a generalized bilinear model / A. Halimi, Y. Altmann, N. Dobigeon, J.-Y. Tourneret // IEEE Transactions on Geoscience and remote sensing. – 2011. - Vol. 49(11). - P. 4153-4162.
98. Hapke, B. Bidirectional reflectance spectroscopy. 1. Theory / B. Hapke // Journal of Geophysical Research. – 1981. - Vol. 86. - P. 3039–3054.
99. Haralick, R.M. Textural Features for Image Classification / R.M. Haralick, K. Shanmugan, I. Dinstein // IEEE Transactions on Systems, Man, and Cybernetics, 1973. – Vol. SMC-3(6). – P. 610-621. DOI: 10.1109/TSMC.1973.4309314.
100. Harris, F.J. On the use of windows for harmonic analysis with the discrete Fourier transform / F.J. Harris // Proceedings of the IEEE. - 1978. - Vol. 66(1). - 51–83.
101. Harsanyi, J.C. Hyperspectral image classification and dimensionality reduction: an orthogonal subspace projection approach / J.C. Harsanyi, C.-I. Chang // IEEE Transactions on Geoscience and Remote Sensing. – 1994. - Vol. 32(4). - P. 779-785. DOI: 10.1109/36.298007.
102. Hastie, T.J. The elements of statistical learning: data mining, inference, and prediction / T.J. Hastie, R.J. Tibshirani, J.H. Friedman. - New York: Springer, 2009. - 767 p.
103. He, K. Momentum contrast for unsupervised visual representation learning / K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. - 2020. – P. 9729-9738. DOI: 10.1109/CVPR42600.2020.00975.

104. He, X. Transferring CNN Ensemble for Hyperspectral Image Classification / X. He, Y. Chen // IEEE Geoscience and Remote Sensing Letters. – 2021. - Vol. 18(5). - P. 876-880. DOI: 10.1109/LGRS.2020.2988494.

105. Hellinger, E. Neue Begrundung der Theorie Quadratischer Formen von Unendlichvielen Veranderlichen / E. Hellinger // Journal fur diereine und angewandte Mathematik. – 1909. - Vol. 136. - P. 210-271. (in German)

106. Heylen, R. A Review of Nonlinear Hyperspectral Unmixing Methods / R. Heylen, M. Parente, P. Gader // IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. – 2014. - Vol. 7(6). - P. 1844-1868. DOI: 10.1109/JSTARS.2014.2320576.

107. Heylen, R. Nonlinear spectral unmixing with a linear mixture of intimate mixtures model / R. Heylen, P. Gader // IEEE Geoscience and Remote Sensing Letters. - 2014. – Vol. 11(7). – P. 1195-1199.

108. Hidalgo, D.R. Dimensionality reduction of hyperspectral images of vegetation and crops based on self-organized maps / D.R. Hidalgo, B.B. Cortés, E.C. Bravo // Information Processing in Agriculture. – 2021. – Vol. 8(2). – P. 310-327. DOI: 10.1016/j.inpa.2020.07.002.

109. Hinton, G.E. Stochastic Neighbor Embedding / G.E. Hinton, S.T. Roweis // Advances in Neural Information. Processing Systems. – 2002. – Vol. 15. – P. 833-840.

110. Hoang Nguyen, N. Hyperspectral image unmixing using manifold learning methods derivations and comparative tests / N. Hoang Nguyen, C. Richard, P. Honeine, C. Theys // 2012 IEEE International Geoscience and Remote Sensing Symposium. – 2012. P. 3086-3089. DOI: 10.1109/IGARSS.2012.6350773.

111. Hoffmann, R. DEM Simulations of Toner Particles with an O (N log N) Hierarchical Tree Code Algorithm / R. Hoffmann // Granular Matter. - 2006. - Vol. 8. P. 151–157. DOI: 10.1007/s10035-006-0002-6.

112. Hong, D. Graph convolutional networks for hyperspectral image classification / D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, J. Chanussot // IEEE Transactions on Geoscience and Remote Sensing. – 2020. – Vol. 59. – P. 5966-5978.

113. Hong, D.F. Local manifold learning with robust neighbors selection for hyperspectral dimensionality reduction / D.F. Hong, N. Yokoya, X.X. Zhu // International Geoscience and Remote Sensing Symposium (IGARSS). – 2016. - 7729001. – P. 40-43.

114. Hotelling, H. Analysis of a complex of statistical variables into principal components / H. Hotelling // Journal of Educational Psychology. – 1933. – Vol. 24. – P. 417-441. DOI: 10.1037/h0071325.

115. How to Access Global Memory Efficiently in CUDA C/C++ Kernels // NVIDIA developer: сайт. – URL: https://developer.nvidia.com/blog/how-access-global-memory-efficiently-cuda-c-kernels/ (дата обращения: 06.10.2024).

116. Hu, B. Vegetation species identification using hyperspectral imagery / B. Hu, J. Lévesque, J.-P. Ardouin // International Geoscience and Remote Sensing Symposium (IGARSS). - 2008. – Vol. 2(1). - P. 299-302.

117. Hu, W. Deep convolutional neural networks for hyperspectral image classification / W. Hu, Y. Huang, L. Wei, F. Zhang, H. Li, // Journal of Sensors. – 2015. - 258619. – P. 1-12. DOI: 10.1155/2015/258619.

118. Huang, H. Dimensionality Reduction of Hyperspectral Imagery Based on Spatial–Spectral Manifold Learning / H. Huang, G. Shi, H. He, Y. Duan, F. Luo // IEEE Transactions on Cybernetics. – 2020. - Vol. 50(6). - P. 2604-2616. DOI: 10.1109/TCYB.2019.2905793.

119. Hyperspec VNIR Hyperspectral Imaging Sensor. Product data sheet // Polytec: сайт. – URL: https://www.polytec.com/fileadmin/website/optical-systems/spezialkameras/pdf/PH_HWP_Hyperspec_VNIR_0118.pdf (дата обращения: 06.10.2024).

120. Hyperspectral Remote Sensing Scenes // Grupo de Inteligencia Computacional: сайт. – URL: http:// www.ehu.eus/ccwintco/index.php? title= Hyperspectral_Remote_Sensing_Scenes (дата обращения: 06.10.2024).

121. Hyperspectral Snapshot USB3 camera 24 bands 665-960nm // XIMEA: сайт. – URL: https://www.ximea.com/en/products/hyperspectral-cameras-based-on-usb3-xispec/mq022hg-im-sm5x5-nir (дата обращения: 06.10.2024).

122. Hyperspectral Snapshot USB3 camera 16 bands 460-600nm // XIMEA: сайт. – URL: https://www.ximea.com/en/products/hyperspectral-cameras-based-on-usb3-xispec/mq022hg-im-sm4x4-vis (дата обращения: 06.10.2024).

123. Hyvärinen, A. Independent component analysis: algorithms and applications / A. Hyvärinen, E. Oja // Neural networks. – 2000. – Vol. 13(4-5). – P. 411-430.

124. ICER-3D Hyperspectral Image Compression Software // NASA Tech Briefs, February 2010: сайт. – URL: https://ntrs.nasa.gov/api/citations/20100005261/downloads/ 20100005261.pdf (дата обращения: 06.10.2024).

125. Indyk, P. Approximate nearest neighbors: towards removing the curse of dimensionality / P. Indyk, R. Motwani // Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing. – 1998. - Vol. 13. - P. 604-613.

126. Jakubczyk, K. Hyperspectral Imaging for Mobile Robot Navigation / K. Jakubczyk, B. Siemiatkowska, R. Wieckowski, J. Rapcewicz // Sensors. – 2023. – Vol. 23(1). – P. 383. DOI: 10.3390/s23010383.

127. Jiang, Y. Hyperspectral Image Dimension Reduction and Target Detection Based on Weighted Mean Filter and Manifold Learning / Y. Jiang , T. Wang, H. Chang, Y. Su // Journal of Physics: Conference Series. – 2020. - Vol. 1693. – P. 012182.

128. Johnson, W.B. Extensions of Lipschitz mappings into a Hilbert space / W.B. Johnson, J. Lindenstrauss // AMS Contemporary Mathematics. – 1984. - Vol. 26. – P. 189-206.

129. Jollife, I.T. Principal Component Analysis / I.T. Jollife. - New York: Springer-Verlag, 1986. – 487 p.

130. Jordan, J. Hyperspectral image visualization with a 3-D self-organizing map / J. Jordan, E. Angelopoulou //5th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS). – 2013. - P. 1-4. DOI: 10.1109/WHISPERS.2013.8080607.

131. Journaux, L. Nonlinear reduction of multispectral images by curvilinear component analysis: application and optimization / L. Journaux, I. Foucherot, P. Gouton // International Conference on CSIMTA 2004. - 2004.

132. Kaarna, A. Compression of Spectral Images / A. Kaarna // Vision Systems: Segmentation and Pattern Recognition. – Vienna: I-Tech, 2007. – P. 269-298. DOI: 10.5772/4964.

133. Kailath, T. The divergence and Bhattacharyya distance measures in signal selection / T. Kailath // IEEE Transactions on Communication Technology. – 1967. - Vol. 15(1). - P. 52-60. DOI: 10.1109/TCOM.1967.1089532.

134. Kanthi, M. Multi-scale 3D-convolutional neural network for hyperspectral image classification / M. Kanthi, T. Sarma, S. Chigarapalle, // Indonesian Journal of Electrical Engineering and Computer Science. - 2022. - Vol. 25(1). - P. 307-316. DOI: 10.11591/ijeecs.v25.i1.pp307-316.

135. Karami, A. Compression of hyperspectral images using discrete wavelet transform and tucker decomposition / A. Karami, M. Yazdi, G. Mercier // IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. - 2012 – Vol. 5(2). – P. 444-450. DOI: 10.1109/JSTARS.2012.2189200.

136. Kaufman, L. Partitioning Around Medoids (Program PAM) / L. Kaufman, P.J. Rousseeuw // Finding Groups in Data: An Introduction to Cluster Analysis. - Hoboken: John Wiley & Sons, 1990. - P. 68-125. DOI: https://doi.org/10.1002/9780470316801.ch2.

137. Keogh, E. Curse of Dimensionality / E. Keogh, A. Mueen, // Encyclopedia of Machine Learning. - Boston: Springer, 2011. – P. 257-258. DOI: 10.1007/978-0-387-30164-8_192.

138. Keshava, N. Spectral unmixing / N. Keshava, J.F. Mustard // IEEE Signal Processing Magazine. -2002. - Vol. 19(1). - P. 44-57. DOI: 10.1109/79.974727.

139. Khodr, J. Dimensionality reduction on hyperspectral images: A comparative review based on artificial datas / J. Khodr, R. Younes // 4th International Congress on Image and Signal Processing. - 2011. - P. 1875-1883. DOI: 10.1109/CISP.2011.6100531.

140. Kim, D.H. Hyperspectral image processing using locally linear embedding / D.H. Kim, L.H. Finkel // First International IEEE EMBS Conference on Neural Engineering. – 2003. - P. 316-319.

141. Kingma, D. Adam: Adam: A Method for Stochastic Optimization / D. Kingma, J. Ba // International Conference on Learning Representations (ICLR). - 2015. DOI: 10.48550/arXiv.1412.6980.

142. Kira, K. A Practical Approach to Feature Selection / K. Kira, L. Rendell, // Proceedings of the Ninth International Workshop on Machine Learning. – 1992. – P. 249-256.

143. Kivinen J. Exponentiated gradient versus gradient descent for linear predictors / J. Kivinen, M.K. Warmuth // Information and Computation. – 1997. – Vol. 132(1). – P. 1–63.

144. Kohonen, T. Self-organizing Maps / T. Kohonen. - Heidelberg: Springer, 2000. – 502 p. – ISBN 978-3-642-56927-2.

145. Kong, X. The Research on Effectiveness of Spectral Similarity Measures for Hyperspectral Image / X. Kong, N. Shu, W. Huang, J. Fu // 3rd International IEEE Congress on Image and Signal Processing (CISP2010). – 2010. – P. 2269-2273.

146. Kramer, M.A. Nonlinear principal component analysis using autoassociative neural networks / M.A. Kramer // AIChE Journal. – 1991. - Vol. 37. - P. 233–243.

147. Kruse, F.A. The Spectral Image Processing System (SIPS) – Interactive Visualization and Analysis of Imaging Spectrometer Data / F.A. Kruse, J.W. Boardman, A.B. Lefkoff, K.B. Heidebrecht, A.T. Shapiro, P.J. Barloon, A.F.H. Goetz // Remote Sensing of Environment. – 1993. – Vol. 44.- P. 145-163.

148. Kruskal, J.B. Multidimensional scaling by optimizing goodness of fit to a non-metric hypothesis / J.B. Kruskal // Psychometrika. – 1964. – Vol. 29. - P. 1-27.

149. Kuester, J. 1D-Convolutional Autoencoder Based Hyperspectral Data Compression / J. Kuester, W. Gross, W. Middelmann // International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. – 2021. - Vol. XLIII-B1-2021. – P. 15-21. DOI: 10.5194/isprs-archives-XLIII-B1-2021-15-2021.

150. LeCun, Y. Backpropagation Applied to Handwritten Zip Code Recognition / Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel // Neural Computation. – 1989. – Vol. 1(4). – P. 541-551.

151. LeCun, Y. The MNIST database of handwritten digits / Y. LeCun, C. Cortes, C.J.C. Burges // Yann LeCun: сайт. – URL: http://yann.lecun.com/exdb/mnist/ (дата обращения: 06.10.2024).

152. Lee, J.A. A Robust Nonlinear Projection Method / J.A. Lee, A. Lendasse, N. Donckers, M. Verleysen // 8th European Symposium on Artificial Neural Networks (ESANN’2000). - 2000. - P. 13–20.

153. Lee, R.C.T. A Triangulation Method for the Sequential Mapping of Points from N-Space to Two-Space / R.C.T. Lee, J.R. Slagle, H.A. Blum // IEEE Transactions on Computers. – 1977. - Vol. 26(3). - P. 288-292.

154. Lennon, M. Curvilinear component analysis for nonlinear dimensionality reduction of hyperspectral images / M. Lennon, G. Mercier, M. Mouchot, L. Hubert-Moy // Image

and signal processing for remote sensing VII. Proceedings of SPIE. - 2002. – Vol. 4541 - P. 157–168.

155. Li, C. Deep Belief Network for Spectral–Spatial Classification of Hyperspectral Remote Sensor Data / C. Li, Y. Wang, X. Zhang, H. Gao, Y. Yang, J. Wang // Sensors. – 2019. Vol. 19(1). – P. 204. DOI: 10.3390/s19010204.

156. Li, H. Dimensionality Reduction and Classification of Hyperspectral Remote Sensing Image Feature Extraction / H. Li, J. Cui, X. Zhang, Y. Han, L. Cao // Remote Sensing. – 2022. – Vol. 14(18). – P. 4579. DOI: 10.3390/rs14184579.

157. Li, Q. Ensemble EMD-Based Spectral-Spatial Feature Extraction for Hyperspectral Image Classification / Q. Li, B. Zheng, B. Tu, J. Wang, C. Zhou // IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. – 2020. - Vol. 13. - P. 5134-5148. DOI: 10.1109/JSTARS.2020.301871.

158. Li, R. Classification of hyperspectral image based on double-branch dual-attention mechanism network / R. Li, S. Zheng, C. Duan, Y. Yang, X. Wang // Remote Sensing. – 2020. – Vol. 12. – P. 582.

159. Li, Y. Spectral-spatial classification of hyperspectral imagery with 3D convolutional neural network / Y. Li, H. Zhang, Q. Shen // Remote Sensing. – 2017. – Vol. 9. - P. 67.

160. Liang, L. Multiscale DenseNet meets with bi-RNN for hyperspectral image classification / L. Liang, S. Zhang, J. Li // International Journal of Applied Earth Observation and Geoinformation. – 2022. – Vol. 15. – P. 5401-5415.

161. Lloyd, S.P. Least squares quantization in PCM / S.P. Lloyd // IEEE Transactions on Information Theory. – 1982. - Vol. 28(2). – P. 129–137. DOI: 10.1109/TIT.1982.1056489.

162. Lu, G Medical hyperspectral imaging: a review / G. Lu, B. Fei // Journal of Biomedical Optics. – 2014. – Vol. 19(1). – P. 10901. DOI: 10.1117/1.JBO.19.1.010901.

163. Lupu, D. Dimensionality reduction of hyperspectral images using an ICA-based stochastic second-order optimization algorithm / D. Lupu, I. Necoara, T.C. Ionescu, L. Ghinea, J. Garrett, T.A. Johansen // 2023 European Control Conference (ECC). – 2023. - P. 1-6. DOI: 10.23919/ECC57647.2023.10178317.

164. Manas, O. Seasonal contrast: Unsupervised pre-training from uncurated remote sensing data / O. Manas, A. Lacoste, X. Giró-i Nieto, D. Vazquez, A. Rodriguez //

Proceedings of the IEEE/CVF International Conference on Computer Vision. – 2021. - P. 9414–9423.

165. Mantripragada, K. The effects of spectral dimensionality reduction on hyperspectral pixel classification: A case study / K. Mantripragada, P.D. Dao, Y. He, F.Z. Qureshi // PLoS ONE. – 2022. - Vol. 17(7). - P. e0269174. DOI: 10.1371/journal.pone.0269174.

166. Marimont, R.B. Nearest Neighbour Searches and the Curse of Dimensionality / R.B. Marimont, M.B. Shapiro // IMA Journal of Applied Mathematics. – 1979. – Vol. 24(1). – P. 59–70. DOI: 10.1093/imamat/24.1.59.

167. Martinetz, T. A neural gas network learns topologies / T. Martinetz, K. Schulten // IEEE International Conference on Neural Networks. – 1991. - Vol. 1. - P. 397-407.

168. Matlab // The MathWorks, Inc.: сайт. – URL: https://www.mathworks.com/products/matlab.html (дата обращения: 06.10.2024).

169. Matousek, J. Lecture notes on metric embeddings / J. Matousek. – Praha: Charles University, 2013. – 125 p.

170. McInnes, L. Umap: Uniform manifold approximation and projection for dimension reduction / L. McInnes, J. Healy, J. Melville // arXiv: 1802.03426. – 2018. – URL: https://arxiv.org/abs/1802.03426 (дата обращения: 06.10.2024).

171. McInnes, L. UMAP / L. McInnes // Read the Docs: сайт. – URL: https://umap-learn.readthedocs.io/en/latest/index.html (дата обращения: 06.10.2024).

172. Meagher, D. Octree Generation, Analysis and Manipulation / D. Meagher // Rensselaer Polytechnic Institute. Image Processing Laboratory. Technical Report. – 1982. - IPL-TR-027. - P. 1-144.

173. Meganem, I. Linear-quadratic mixing model for reflectances in urban environments / I. Meganem, P. D´eliot, X. Briottet, Y. Deville, S. Hosseini // IEEE Transactions on Geoscience and Remote Sensing. – 2014. - Vol. 52(1). - P. 544–558.

174. Mei, X. Spectral-Spatial Attention Networks for Hyperspectral Image Classification / X. Mei, E. Pan, Y. Ma, X. Dai, J. Huang, F. Fan, Q. Du, H. Zheng, J. Ma // Remote Sensing. – 2019. - Vol. 11(8). - P. 963.

175. Mercer, J. Functions of positive and negative type and their connection with the theory of integral equations / J. Mercer // Philosophical Transactions of the Royal Society A. – 1909. - |Vol. 209. – P. 415–446.

176. Mielikainen, J. Improved Vector Quantization for Lossless Compression of AVIRIS Images / J. Mielikainen, P. Toivanen // Proceedings of the 11th European Signal Processing Conference (EUSIPCO-2002). – 2002. – P. 1-3.

177. Mishchenko, M.I. Bidirectional reflectance of flat, optically thick particulate layers: An efficient radiative transfer solution and applications to snow and soil surfaces / M.I. Mishchenko, J.M. Dlugach, E.G. Yanovitskij, N.T. Zakharova // Journal of Quantitative Spectroscopy and Radiative Transfer. – 1999. - Vol. 63. - P. 409-432.

178. Mohan, A. Hybrid CNN based hyperspectral image classification using multiscale spatiospectral features / A. Mohan, M. Venkatesan // Infrared Physics & Technology. – 2020. - Vol. 108. – P. 103326. DOI: 10.1016/j.infrared.2020.103326.

179. Mohan, A. Spatially Coherent Nonlinear Dimensionality Reduction and Segmentation of Hyperspectral Images / A. Mohan, G. Sapiro, E. Bosch // IEEE Geoscience and Remote Sensing Letters. – 2007. - Vol. 4(2). - P. 206-210. DOI: 10.1109/LGRS.2006.888105.

180. Motta , G. Hyperspectral Data Compression / G. Motta, F. Rizzo, J.A. Storer. - N.Y.: Springer Science and Business Media, 2006. - 421 p.

181. Mou, L. Nonlocal graph convolutional networks for hyperspectral image classification / L. Mou, X. Lu, X. Li, X.X. Zhu // Transactions on Geoscience and Remote Sensing. – 2020. – Vol. 58. – P. 8246–8257.

182. **Myasnikov, E.** A Framework for Feature Fusion with Application to Hyperspectral Images / E. Myasnikov // Pattern Recognition, Computer Vision, and Image Processing. ICPR 2022 International Workshops and Challenges. Lecture Notes in Computer Science. – 2023. - Vol. 13644. - P. 509–518.

183. **Myasnikov, E.** Data Dimensionality Reduction Technique Based on Preservation of Hellinger Divergence / E. Myasnikov // Analysis of Images, Social Networks and Texts. AIST 2021. Lecture Notes in Computer Science. – 2022. - Vol. 13217. - P. 190-198.

184. **Myasnikov, E.** Dimensionality Reduction of Hyperspectral Images Based on the Linear Mixture Model and Dimensionality Estimation / E. Myasnikov // Proceedings of SPIE, Twelfth International Conference on Machine Vision (ICMV 2019). — 2020. - Vol. 11433. — P. 114331L.

185. **Myasnikov, E.** Dimensionality reduction of hyperspectral images using autoassociative neural networks / E. Myasnikov // SIBIRCON 2019 - International

Multi-Conference on Engineering, Computer and Information Sciences. — 2019. — P. 591-595.

186. **Myasnikov, E.** Embedding spatial context into spectral angle based nonlinear mapping for hyperspectral image analysis / E. Myasnikov // Computer Vision and Graphics. Lecture Notes in Computer Science. — 2018. — Vol. 11114. — P. 263-274.

187. **Myasnikov, E.** Evaluation of nonlinear dimensionality reduction techniques for classification of hyperspectral images / E. Myasnikov // CEUR Workshop Proceedings. — 2018. — Vol. 2268. — P. 147-154.

188. **Myasnikov, E.** Evaluation of spectral similarity measures and dimensionality reduction techniques for hyperspectral images / E. Myasnikov // Pattern Recognition. ICPR International Workshops and Challenges. ICPR 2021. Lecture Notes in Computer Science. – 2021. Vol. 12665. –pp. 289–300.

189. **Myasnikov, E.** Exploiting Spatial Context in Nonlinear Mapping of Hyperspectral Image Data / E. Myasnikov // Image Analysis and Processing. Lecture Notes in Computer Science. — 2017. — Vol. 10485. — P. 180-190.

190. **Myasnikov, E.** Hyperspectral Data Clustering Using Hellinger Divergence / E. Myasnikov // Journal of Physics: Conference Series. – 2021. - Vol. 2096(1). – P. 012170.

191. **Myasnikov, E.** Hyperspectral data dimensionality reduction using nonlinear autoencoders / E. Myasnikov // CEUR Workshop Proceedings. — 2020 — Vol. 2665. — P. 33-36.

192. **Myasnikov, E.** Nearest Neighbor Search in Hyperspectral Data Using Binary Space Partitioning Trees / E. Myasnikov // IEEE Xplore Digital Library. Workshop on Hyperspectral Image and Signal Processing, Evolution in Remote Sensing (WHISPERS). – 2021. – P. 9484041.

193. **Myasnikov, E.** Nonlinear dimensionality reduction of hyperspectral data based on spectral information divergence preserving principle / E. Myasnikov // Journal of Physics: Conference Series. - 2019. – Vol. 1368(3). - P. 032030.

194. **Myasnikov, E.** Nonlinear dimensionality reduction of hyperspectral images based on spectral angles and exploiting the spatial context / E. Myasnikov // Journal of Physics: Conference Series. — 2018. — Vol. 1096(1). – P. 012037. DOI: 10.1088/1742-6596/1096/1/012037.

195. **Myasnikov, E.** Nonlinear dimensionality reduction of hyperspectral data using spectral correlation as a similarity measure / E. Myasnikov // Analysis of Images, Social Networks and Texts. AIST 2017. Lecture Notes in Computer Science. — 2018. — Vol. 10716. — P. 237-244. DOI: 10.1007/978-3-319-73013-4_22.

196. **Myasnikov, E.** Nonlinear mapping based on spectral angle preserving principle for hyperspectral image analysis / E. Myasnikov // Computer Analysis of Images and Patterns. Lecture Notes in Computer Science. — 2017. — Vol. 10425. — P. 416-427.

197. **Myasnikov, E.** Segmentation of hyperspectral images based on dimensionality estimation in spatial regions / E. Myasnikov // IEEE Xplore Digital Library, 2021 International Conference on Information Technology and Nanotechnology (ITNT). – 2021. - P. 1-4. doi: 10.1109/ITNT52450.2021.9649299.

198. **Myasnikov, E.** Using UMAP for Dimensionality Reduction of Hyperspectral Data / E. Myasnikov // IEEE Xplore Digital Library, 2020 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon). – 2020. - p. 1-5. DOI: 10.1109/FarEastCon50210.2020.9271656.

199. **Myasnikov, E.** Visualization of feature spaces based on spectral and texture characteristics / E. Myasnikov // IEEE Xplore Digital Library, 9th IEEE International Conference on Information Technology and Nanotechnology. - 2023. - P. 1-5. DOI: 10.1109/ITNT57377.2023.10139108. .

200. **Myasnikov, E.V.** A feature fusion technique for dimensionality reduction / E.V. Myasnikov // Pattern Recognition and Image Analysis. - 2022. - Vol. 32(3). - P. 607-610.

201. **Myasnikov, E.V.** A Nonlinear Dimensionality Reduction Using Combined Approach to Feature Space Decomposition / E.V. Myasnikov // CEUR Workshop Proceedings. – 2015. - Vol. 1452. - P. 85-95.

202. **Myasnikov, E.V.** A Nonlinear Method for Dimensionality Reduction of Data Using Reference Nodes / E.V. Myasnikov // Pattern Recognition and Image Analysis. – 2012. - Vol. 22(2). - P. 337–345.

203. **Myasnikov, E.V.** Analysis of Approaches to Feature Space Partitioning for Nonlinear Dimensionality Reduction / E.V. Myasnikov // Pattern Recognition and Image Analysis. – 2016. – Vol. 26(3). - P. 474-482.

204. **Myasnikov, E.V.** Compact representations of hyperspectral images based on the fusion of neural network and spectral features / E.V. Myasnikov // Pattern Recognition and Image Analysis. – 2024. – Vol. 34(4). – pp. 1021-1025.

205. **Myasnikov, E.V.** Comparison of Spectral Dissimilarity Measures and Dimension Reduction Techniques for Hyperspectral Images / E. Myasnikov // Pattern Recognition and Image Analysis. - 2021. –Vol. 31(3). - P. 453–464.

206. **Myasnikov, E.V.** Evaluation of Space Partitioning Data Structures for Nonlinear Mapping / E.V. Myasnikov // 23rd International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG 2015) - Full Papers Proceedings. - 2015. - P. 109-118.

207. **Myasnikov, E.V.** Evaluation of stochastic gradient descent methods for nonlinear mapping of hyperspectral data / E. Myasnikov // Image Analysis and Recognition. Lecture Notes in Computer Science. – 2016. – Vol. 9730. - P. 276-283.

208. **Myasnikov, E.V.** Fast Techniques for Nonlinear Mapping of Hyperspectral Data / E.V. Myasnikov // Proc. of SPIE, Ninth International Conference on Machine Vision (ICMV 2016). - Vol. 10341. – P. 103411D.

209. **Myasnikov, E.V.** Fusion of Deep Learning-based and Spectral Features for Hyperspectral Image Analysis / E.V. Myasnikov // International workshop on Photogrammetric Data Analysis - ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences. – 2024. – Vol. X-2/W1-2024. – P. 31–38.

210. **Myasnikov, E.V.** Generation of Compact Representations of Hyperspectral Images Using Neural Network and Spectral Features / E. Myasnikov // ICPR 2024 International Workshops and Challenges. Lecture Notes in Computer Science. - 2025. – Vol. 15616. – P. 274–284.

211. **Myasnikov, E.V.** Hyperspectral image segmentation using dimensionality reduction and classical segmentation approaches / E.V. Myasnikov // Computer Optics. – 2017.– V. 41(4).– P. 564-572.

212. **Myasnikov, E.V.** Nonlinear dimensionality reduction method with the use of reference nodes for analysis of digital image collections / E.V. Myasnikov // Pattern Recognition and Image Analysis. – 2011. - Vol. 21(2). - P. 312-315.

213. **Myasnikov, E.V.** Nonlinear mapping methods with adjustable computational complexity for hyperspectral image analysis / E.V. Myasnikov // Proceedings of SPIE -

The International Society for Optical Engineering. Eighth International Conference on Machine Vision (ICMV 2015). - 2015. -Vol. 9875. – P. 987508.

214. **Myasnikov, E.V.** Nonlinear Mapping of Hyperspectral Data / E.V. Myasnikov // Image Analysis and Pattern Recognition. State of the Art in the Russian Federation. – Singapore: World Scientific Publishing Company, 2024. ISBN: 9811267200.

215. **Myasnikov, E.V.** The Choice of a Method for Feature Space Decomposition for Non-Linear Dimensionality Reduction / E.V. Myasnikov // Computer optics. – 2014. - Vol. 38(4). - P. 790-797.

216. **Myasnikov, E.V.** The use of Interpolation Methods for Nonlinear Mapping / E. Myasnikov // Computer Vision and Graphics. Lecture Notes in Computer Science. – 2016. - Vol. 9972. - P. 649-655.

217. Nalepa, J. Deep Ensembles for Hyperspectral Image Data Classification and Unmixing / J. Nalepa, M. Myller, L. Tulczyjew, M. Kawulok // Remote Sensing. – 2021. – Vol. 13(20). – P. 4133. DOI: 10.3390/rs13204133.

218. Nascimento, J.M.P. Nonlinear mixture model for hyperspectral unmixing / J.M.P. Nascimento, J.M. Bioucas-Dias // Image and Signal Processing for Remote Sensing XV. Proceedings of SPIE. – 2009. - Vol. 7477. - P. 74770I-74770I-8.

219. Newling, J. A Sub-Quadratic Exact Medoid Algorithm / J. Newling, F. Fleuret // Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research. – 2017. – Vol. 54. – P. 185-193.

220. Nielsen, F. Hierarchical Clustering / F. Nielsen. - Introduction to HPC with MPI for Data Science. - Cham: Springer, 2016. - P. 195-211. DOI: 10.1007/978-3-319-21903-5_8.

221. Noyel, G. Morphological segmentation of hyperspectral images / G. Noyel, J. Angulo, D. Jeulin // Image Analysis and Stereology.– 2007. - Vol. 26(3). – P. 101-109. DOI: 10.5566/ias.v26.p101-109.

222. Nuttall, A.H. Some windows with very good sidelobe be-havior / A.H. Nuttall // IEEE Transactions on Acoustics, Speech and SignalProcessing. - 1981. - Vol. 29(1). - P. 84–91.

223. NVIDIA CUDA // NVIDIA Documentation Hub: сайт. – URL: https://docs.nvidia.com/cuda/doc/index.html (дата обращения: 06.10.2024).

224. NVIDIA H100 Tensor Core GPU // NVIDIA Cloud & Data Center: сайт. – URL: https://www.nvidia.com/en-us/data-center/h100/ (дата обращения: 06.10.2024).

225. n-sphere // Python Package Index: сайт. – URL: https://pypi.org/project/n-sphere/ (дата обращения: 06.10.2024).

226. Ojala , T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns / T. Ojala, M. Pietikäinen, T. Mäenpää // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2002. – Vol. 24(7). – P. 971–987 DOI: 10.1109/TPAMI.2002.1017623.

227. Okamoto, K. Ranking of closeness centrality for large-scale social networks / K. Okamoto, W. Chen, X.-Y. Li // Proceedings of the 2nd Annual International Workshop on Frontiers in Algorithmics (FAW'08). – 2008. – P. 186-195.

228. Orts Gómez, F.J. Hyperspectral Image Classification Using Isomap with SMACOF / F.J. Orts Gómez, G. Ortega López, E. Filatovas, O. Kurasova, G.E.M. Garzón // Informatica. - 2019. – Vol. 30(2). – P. 349-365. DOI: 10.15388/Informatica.2019.209.

229. Pahomov, P. Investigation of correlation of empirical modes and low-frequency residues of hyperspectral image signatures / P. Pahomov, V. Turlapov // 2020 International Conference on Information Technology and Nanotechnology (ITNT). – 2020. - P.1-5. DOI: 10.1109/ITNT49337.2020.9253179.

230. Paoletti, M.E. Deep learning classifiers for hyperspectral imaging: A review / M.E. Paoletti, J.M. Haut, J. Plaza, A. Plaza // ISPRS Journal of Photogrammetry and Remote Sensing. - 2019. - Vol. 158. - P. 279-317. DOI: 10.1016/j.isprsjprs.2019.09.006.

231. Paoletti, M.E. Deep pyramidal residual networks for spectral–spatial hyperspectral image classification / M.E. Paoletti, J.M. Haut, R. Fernandez-Beltran, J. Plaza, A.J. Plaza, F. Pla // IEEE Transactions on Geoscience and Remote Sensing. – 2018. - Vol. 57(2). - P. 740-754.

232. Pearson, K. On Lines and Planes of Closest Fit to Systems of Points in Space / K. Pearson // Philosophical Magazine. - 1901. – Vol. 2(11). – P. 559–572.

233. Pekalska, E. A new method of generalizing Sammon mapping with application to algorithm speed-up / E. Pekalska, D. de Ridder, R.P.W. Duin, M.A. Kraaijveld // 5th Annual Conference of the Advanced School for Computing and Imaging, ASCI'99. - 1999. - P. 221-228.

234. Penna, B. Progressive 3-D Coding of Hyperspectral Images Based on JPEG 2000 / B. Penna, T. Tillo, E. Magli, G. Olmo // IEEE Geoscience and Remote Sensing Letters. - 2006. - Vol. 3(1). - P. 125-129. DOI: 10.1109/ LGRS.2005.859942.

235. Pereira, L.A.M. A Binary Cuckoo Search and Its Application for Feature Selection / L.A.M. Pereira, D.Rodrigues, T.N.S. Almeida, C.C.O. Ramos, A.N. Souza, X.-S. Yang, J.P. Papa // Cuckoo Search and Firefly Algorithm. Studies in Computational Intelligence. - 2013. - Vol 516. - P.141–154. DOI: 10.1007/978-3-319-02141-6_7.

236. Plaut, E. From Principal Subspaces to Principal Components with Linear Autoencoders / E. Plaut // arXiv: 1804.10253v3. - 2018. – URL: https://arxiv.org/abs/1804.10253 (дата обращения: 22.11.2016).

237. Qiao, T. Effective compression of hyperspectral imagery using an improved 3D DCT approach for land-cover analysis in remote-sensing applications / T. Qiao, J. Ren, M. Sun, J. Zheng, S. Marshall // International Journal of Remote Sensing. – 2014. – Vol. 35(20). - P. 7316-7337. DOI: 10.1080/01431161.2014.968682.

238. Qu, L. Triple-Attention-Based Parallel Network for Hyperspectral Image Classification / L. Qu, X. Zhu, J. Zheng, L. Zou // Remote Sensing. – 2021. – Vol. 13. – P. 324.

239. Quigley, A. FADE: Graph Drawing, Clustering, and Visual Abstraction / A. Quigley, P. Eades // Proceedings of the 8th International Symposium on Graph Drawing. - 2001. - P. 197–210.

240. Rand, W.M. Objective criteria for the evaluation of clustering methods / W.M. Rand // Journal of the American Statistical Association. – 1971. – Vol. 66(336). – P. 846-850.

241. Reed, M. Functional Analysis / M. Reed, B. Simon. – Cambridge: Academic Press, 1980. - 416 p.

242. Ren, S. Semi-supervised classification for PolSAR data with multi-scale evolving weighted graph convolutional network / S. Ren, F. Zhou // International Journal of Applied Earth Observation and Geoinformation. – 2021. – Vol. 14. – P. 2911–2927.

243. Richards, J.A. Remote Sensing Digital Image Analysis: An Introduction / J.A. Richards, X. Jia , D.E. Ricken, W. Gessner. - New York: Springer-Verlag, 1999. – 363 p.

244. Ritter, H. On the stationary state of the Kohonen self-organizing sensory mapping / H. Ritter, K. Schulten // Biological Cybernetics. - 1986. – Vol. 54. – P. 234-249.

245. Robila, S. New developments in target detection in hyperspectral imagery using spectral metrics and spectra extraction / S. Robila // American Society for Photogrammetry and Remote Sensing - ASPRS Annual Conference 2007: Identifying Geospatial Solutions. – 2007. - Vol. 2. - P. 604-614.

246. Robila, S.A. Spectral matching accuracy in processing hyperspectral data / S.A. Robila, A. Gershman // ISSCS 2005: International Symposium on Signals, Circuits and Systems. – 2005. - Vol. 1. - P. 163-166.

247. Rosalind Wang, X. Applying ISOMAP to the learning of hyperspectral image / X. Rosalind Wang, S. Kumar, T. Kaupp, B. Upcroft, H. Durrant-Whyte // Proceedings of the 2005 Australasian Conference on Robotics and Automation. – 2005. - P. 1-8.

248. Ross, A. Feature Level fusion using hand and face biometrics / A. Ross, R. Govindarajan // Proceedings of SPIE Conference on Biometric Technology for Human Identification II. – 2005. - Vol. 5779. - P. 196–204.

249. Ross, A. Fusion, Feature-Level / A. Ross // Encyclopedia of Biometrics. – Boston: Springer, 2009. - P. 597-602. DOI: 10.1007/978-0-387-73003-5_157.

250. Rousseeuw, P.J. Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis / P.J. Rousseeuw // Computational and Applied Mathematics. – 1987. - Vol. 20. - P. 53–65,

251. Roussos, G. Rapid evaluation of radial basis functions / G. Roussos, B.J.C. Baxter // Journal of Computational and Applied Mathematics. - 2005. - Vol. 180. - P. 51 – 70.

252. Roweis, S.T. Nonlinear Dimensionality Reduction by Locally Linear Embedding / S.T. Roweis, L.K. Saul // Science. - 2000. – Vol. 290. – P. 2323-2326.

253. Rumelhart, D.E. Learning representations by back-propagating errors / D.E. Rumelhart, G.E. Hinton, R.J. Williams, // Nature. -1986. – Vol. 323(6088). – P. 533-536.

254. Safari, K. A Multiscale Deep Learning Approach for High-Resolution Hyperspectral Image Classification / K. Safari, S. Prasad, D. Labate // IEEE Geoscience and Remote Sensing Letters. – 2021. - Vol. 18(1). - P. 167-171. DOI: 10.1109/LGRS.2020.2966987.

255. Salmon, J.K. Skeletons from the Treecode Closet / J.K. Salmon, M.S. Warren // Journal of Computational Physics. - 1994. – Vol. 111 – P. 136-155.

256. Samet, H. Foundations of multidimensional and metric data structures / H. Samet. - Morgan Kaufmann, 2006. – 1024 p.

257. Sammon, J.W. A nonlinear mapping for data structure analysis / J.W. Sammon Jr. // IEEE Transactions on Computers. – 1969. - Vol. C–18(5). - P. 401-409.

258. Sanger, T.D. Optimal unsupervised learning in a single-layer linear feedforward neural network / T.D. Sanger // Neural Networks. – 1989. – Vol. 2(6). - P. 459-473.

259. Schölkopf, B. Kernel principal component analysis / B. Schölkopf, A. Smola, K.-R. Müller // International conference on artificial neural networks, ICANN ’97. Lecture Notes in Computer Science. – 1997. - Vol. 1327. – P. 583-588.

260. Shalev-Shwartz, S. Decision Trees / S. Shalev-Shwartz, S. Ben-David // Understanding Machine Learning. From Theory to Algorithms. – Cambridge: Cambridge University Press, 2014. – P. 212-218. – DOI: 10.1017/CBO9781107298019.019.

261. Shen-En, Q. A new nonlinear dimensionality reduction method with application to hyperspectral image analysis / Q. Shen-En, C. Guangyi // IEEE International Geoscience and Remote Sensing Symposium. – 2007. - P. 270-273. .

262. Sheng, Y. Experiments on pattern recognition using invariant Fourier–Mellin descriptors / Y. Sheng, H.H. Arsenault // Journal of the Optical Society of America A. – 1986. – Vol. 3. – P. 771-776.

263. Shepard, D. A two-dimensional interpolation function for irregularly-spaced data / D. Shepard // Proceedings of the 1968 ACM National Conference. – 1968. – P. 517–524.

264. Shkuratov, Y. A critical assessment of the Hapke photometric model / Y. Shkuratov, V. Kaydash, V. Korokhin, Y. Velikodsky, D. Petrov, E. Zubko, D. Stankevich, G. Videen // Journal of Quantitative Spectroscopy and Radiative Transfer. - 2012. - Vol. 113(18). - P. 2431-2456.

265. Shkuratov, Y.G. A model of spectral albedo of particulate surfaces: Implications for optical properties of the moon / Y.G. Shkuratov, L. Starukhina, H. Hoffmann, G. Arnold // Icarus. – 1999. - Vol. 137. - P. 235-246.

266. Siedlecki, W. A note on genetic algorithms for large-scale feature selection / W. Siedlecki, J. Sklansky // Pattern Recognition Letters. - 1989. - Vol. 10(5). - P. 335-347. DOI: 10.1016/0167-8655(89)90037-8.

267. Silva, V. Global Versus Local Methods in Nonlinear Dimensionality Reduction / V. Silva, J. Tenenbaum // NIPS'02: Proceedings of the 15th International Conference on Neural Information Processing Systems. – 2002. – P. 721-728.

268. Singer, R.B. Mars: Large scale mixing of bright and dark surface materials and implications for analysis of spectral reflectance / R.B. Singer, T.B. McCord // 10th Lunar and Planetary Science Conference. – 1979. - P. 1835–1848.

269. Smith, M.O. Interpretation of AIS images of Cuprite, Nevada using constraints of spectral mixtures / M.O. Smith, J.B. Adams // Airborne Imaging Spectrometer Data Analysis Workshop, NASA JPL Publication. - 1985. - P. 62-67.

270. Smith, M.O. Vegetation in deserts: I. A regional measure of abundance from multispectral images / M.O. Smith, S.L. Ustin, J.B. Adams, A.R. Gillespie // Remote Sensing of Environment. – 1990. – Vol. 31. – P. 1-26.

271. Smith, R. The NVIDIA GeForce GTX 1080 & GTX 1070 Founders Editions Review: Kicking Off the FinFET Generation / R. Smith // AnandTech: сайт. – URL: https://www.anandtech.com/show/10325/the-nvidia-geforce-gtx-1080-and-1070-founders-edition-review/4 (дата обращения: 06.10.2024).

272. Somers, B. Nonlinear hyperspectral image analysis for tree cover estimates in orchards / B. Somers, K. Cools, S. Delalieux, J. Stuckens, D. Van der Zande, W.W. Verstraeten, P. Coppin // Remote Sensing of Environment. – 2009. - Vol. 113. - P. 1183-1193.

273. Song, H. A spectral-spatial classification of hyperspectral images based on the algebraic multigrid method and hierarchical segmentation algorithm / H. Song, Y. Wang // Remote Sensing. - 2016. - Vol. 8(4). - P. 296. DOI: 10.3390/rs8040296.

274. Song, W. A Study on Dimensionality Reduction and Parameters for Hyperspectral Imagery Based on Manifold Learning / W. Song, X. Zhang, G. Yang, Y. Chen, L. Wang, H. Xu // Sensors. - 2024. - Vol. 24. - P. 2089. DOI: 10.3390/s24072089.

275. Spectral signatures // ESA-Eduspace. European Space Agency: сайт. – URL: https://www.esa.int/SPECIALS/Eduspace_EN/SEMPNQ3Z2OF_0.html (дата обращения: 06.10.2024)

276. Stewart, A.J. TorchGeo: deep learning with geospatial data / A.J. Stewart, C. Robinson, I.A. Corley, A. Ortiz, J.M. Lavista Ferres, A. Banerjee // 30th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '22). - 2022. – Vol. 19. - P. 1–12. DOI: 10.1145/3557915.3560953.

277. Stricker, M. Similarity of color images / M. Stricker, M. Orengo // Proceedings of SPIE, Storage and Retrieval for Image and Video Databases III. – 1995. – Vol. 2420. DOI: 10.1117/12.205308.

278. Sun, G. Deep fusion of localized spectral features and multi-scale spatial features for effective classification of hyperspectral images / G. Sun, X. Zhang, X. Jia, J. Ren, A. Zhang, Y. Yao, H. Zhao // International Journal of Applied Earth Observation and Geoinformation. – 2020. – Vol. 91. – P. 102157.

279. Sun, W. Hyperspectral Band Selection: A Review / W. Sun; Q. Du // IEEE Geoscience and Remote Sensing Magazine. – 2019. - Vol. 7(2). - P. 118 – 139. DOI: 10.1109/MGRS.2019.2911100.

280. Sun, W. Hyperspectral imagery classification using the combination of improved laplacian eigenmaps and improved k-nearest neighbor classifier / W. Sun, C. Liu, W. Li // Wuhan Daxue Xuebao (Xinxi Kexue Ban)/Geomatics and Information Science of Wuhan University. – 2015. – Vol. 40(9). - P. 1151-1156.

281. Sun, W. Nonlinear dimensionality reduction via the ENH-LTSA method for hyperspectral image classification / W. Sun, A. Halevy, J.J. Benedetto, W. Czaja, W. Li, C. Liu // IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. – 2014. – Vol. 7(2). - P. 6419851. - P. 375-388.

282. Sun, Y. Evaluating Performance Tradeos on the Radeon Open Compute Platform / Y. Sun, S. Mukherjee, T. Baruah, S. Dong, J. Gutierrez, P. Mohan, D. Kaeli // 2018 IEEE International Symposium on Performance Analysis of Systems and Software. - 2018. - P. 209-218.

283. Supervised learning // Scikit-learn. Machine Learning in Python: сайт. – URL: https://scikit-learn.org/stable/supervised_learning.html (дата обращения: 06.10.2024).

284. Swain, M. Color indexing / M. Swain, D. Ballard // International Journal of Computer Vision. - 1991. - Vol. 7(1). – P. 11-32.

285. Tamura H. Textural features corresponding to visual perception / H. Tamura, S. Mori, T. Yamawaki // IEEE Transactions on Systems, Man, and Cybernetics. – 1978. – Vol. 8(6). – P. 460–473. DOI: 10.1109/TSMC.1978.4309999.

286. Tang, X. Hyperspectral unmixing based on ISOMAP and spatial information / X. Tang, K. Gao, H. Cheng, G. Ni // Optik. – 2014. – Vol. 125(16). – P. 4283-4287. DOI: 10.1016/j.ijleo.2014.03.038.

287. Tarabalka, Y. Segmentation and classification of hyperspectral images using watershed transformation / Y. Tarabalka, J. Chanussot, J.A. Benediktsson. // Pattern recognition. – 2010. – Vol. 43(7). – P. 2367-2379.

288. Tegegne, Y. Parallel Nonlinear Dimensionality Reduction Using GPU Acceleration / Y. Tegegne, Z. Qu, Y.Qian, Q.V. Nguyen // Data Mining. AusDM 2021. Communications in Computer and Information Science. – 2021. – Vol. 1504. – P. 3-15. https://doi.org/10.1007/978-981-16-8531-6_1.

289. Tenenbaum, J. Mapping a manifold of perceptual observations / J. Tenenbaum // Advances in Neural Information Processing. – 1998. - Vol. 10. - P. 682-688.

290. Tenenbaum, J.B. A Global Geometric Framework for Nonlinear Dimensionality Reduction / J.B. Tenenbaum, V. de Silva, J.C. Langford // Science. - 2000. – Vol. 290. – P. 2319-2323.

291. Tyo, J.S. Principal-components-based display strategy for spectral imagery / J.S. Tyo, A. Konsolakis, D.I. Diersen, R.C. Olsen // IEEE Transactions on Geoscience and Remote Sensing. – 2003. – Vol. 41(3). – P. 708-718.

292. Uhlmann, J. Satisfying General Proximity/Similarity Queries with Metric Trees / J. Uhlmann // Information Processing Letters. – 1991. – Vol. 40(4). – P. 175-179.

293. Unnikrishnan, R.A. Measure for Objective Evaluation of Image Segmentation Algorithms / R.A. Unnikrishnan, C. Pantofaru, M. Hebert // IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). Workshops. – 2005. - P. 34-34.

294. Using Shared Memory in CUDA C/C++ // NVIDIA developer: сайт. – URL: https://developer.nvidia.com/blog/using-shared-memory-cuda-cc/ (дата обращения: 06.10.2024).

295. Vaddi, R., Strategies for dimensionality reduction in hyperspectral remote sensing: A comprehensive overview / R. Vaddi, B.L.N. Phaneendra Kumar, P. Manoharan, L. Agilandeeswari, V. Sangeetha // The Egyptian Journal of Remote Sensing and Space Sciences. – 2024. – Vol. 27(1). – P. 82-92. DOI: 10.1016/j.ejrs.2024.01.005.

296. Van der Maaten, L.J.P. Barnes-Hut-SNE / L.J.P. van der Maaten // Proceedings First 2013 International Conference on Learning Representations. – 2013. – P. 1-11.

297. Van der Maaten, L.J.P. Dimensionality Reduction: A Comparative Review / L.J.P. van der Maaten, E.O. Postma, H.J. van den Herik // Tilburg University Technical Report. - 2009. – N. TiCC TR 2009–005. - P. 1-35.

298. Van der Maaten, L.J.P. Visualizing High-Dimensional Data Using t-SNE / L.J.P. van der Maaten, G.E. Hinton // Journal of Machine Learning Research. – 2008. – Vol. 9. – P. 2579-2605.

299. Vandenberghe, L. Semidefinite programming / L. Vandenberghe, S.P. Boyd // SIAM Review. - 1996. – Vol. 38(1). – P. 49–95.

300. Vedaldi, A. Quick shift and kernel methods for mode seeking / A. Vedaldi, S. Soatto // European Conference on Computer Vision. ECCV 2008. Lecture Notes in Computer Science. - 2008. – Vol. 5305. - P. 705-718.

301. Visual Studio // Microsoft: сайт. – URL: https://visualstudio.microsoft.com/ (дата обращения: 06.10.2024).

302. Vladymyrov, M. Linear-time training of nonlinear low-dimensional embeddings / M. Vladymyrov, M.Á. Carreira-Perpiñán // 17th International Conference on Artificial Intelligence and Statistics (AISTATS 2014). – 2014. - P. 968-977.

303. Wang, H. Lossless Hyperspectral-Image Compression Using Context-Based Conditional Average / H. Wang, S. D. Babacan, K. Sayood // IEEE Transactions on Geoscience and Remote Sensing. – 2007. - Vol. 45(12). - P. 4187-4193. DOI: 10.1109/TGRS.2007.906085.

304. Wang, J. Geometric Structure of High-Dimensional Data and Dimensionality Reduction / J. Wang. - New York: Springer-Verlag, 2012. - 362 p. - ISBN 978-7-04-031704-6.

305. Wang, J. Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis / J. Wang, C.-I. Chang // IEEE Transactions on Geoscience and Remote Sensing. – 2006. – Vol. 44(6). - P. 1586-1600.

306. Wang, L. A survey of methods incorporating spatial information in image classification and spectral unmixing / L. Wang, C. Shi, C. Diao, W. Ji, D. Yin // International Journal of Remote Sensing. – 2016. – Vol. 37(16). – P. 3870-3910.

307. Weinberger, K.Q. Learning a kernel matrix for nonlinear dimensionality reduction / K.Q. Weinberger, F. Sha, L.K. Saul // Proceedings of the Twenty First International

Conference on Machine Learning (ICML 2004). - 2004. – P. 106. DOI: 10.1145/1015330.1015345.

308. Weinberger, K.Q. Unsupervised learning of image manifolds by semidefinite programming / K.Q. Weinberger, L.K.Saul // 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.- 2004. – P. 988-995. DOI: 10.1109/CVPR.2004.1315272.
309. Weinberger, K.Q. Unsupervised learning of image manifolds by semidefinite programming / K.Q. Weinberger, L.K.Saul // International Journal of Computer Vision. - 2006. – Vol. 70. – P. 77-90. DOI: 10.1007/s11263-005-4939-z.
310. Winkel, M. A massively parallel, multi-disciplinary Barnes–Hut tree code for extreme-scale N-body simulations / M. Winkel, R.Speck, H.Hübner, L.Arnold, R.Krause, P.Gibbon // Computer Physics Communications. - 2012. - Vol. 183(4). - P. 880-889. DOI: 10.1016/j.cpc.2011.12.013.
311. Winkens, C. HyKo: A Spectral Dataset for Scene Understanding / C. Winkens, F. Sattler, V. Adams, D. Paulus // IEEE International Conference on Computer Vision (ICCV). – 2017. - P. 254-261.
312. Winkens, C. Robust Features for Snapshot Hyperspectral Terrain-Classification / C. Winkens, V. Kobelt, D. Paulus // Computer Analysis of Images and Patterns. Lecture Notes in Computer Science. - 2017. – Vol. 10424. - P. 16-27. DOI: 10.1007/978-3-319-64689-3_2.
313. Xu, Y. Spectral-spatial unified networks for hyperspectral image classification / Y. Xu, L. Zhang, B. Du, F. Zhang // IEEE Transactions on Geoscience and Remote Sensing. – 2018. – Vol. 56(10). - P. 5893-5909.
314. Xue, B. Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms / B. Xue, M. Zhang, W.N. Browne // Applied Soft Computing. - 2014. - Vol. 18. - P. 261-276. DOI: 10.1016/j.asoc.2013.09.018.
315. Yan, L. Improved time series land cover classification by missing observation-adaptive nonlinear dimensionality reduction / L. Yan, D.P. Roy // Remote Sensing of Environment. – 2015. - Vol. 158. - P. 478-491.
316. Yan, L. Spectral-angle-based Laplacian Eigenmaps for nonlinear dimensionality reduction of hyperspectral imagery / L. Yan, X. Niu // Photogrammetric Engineering & Remote Sensing. – 2014. - Vol. 80(9). – P. 849–861.

317. Yang, B. Multisource domain transfer learning based on spectral projections for hyperspectral image classification / B. Yang, S. Hu, Q. Guo, D. Hong // International Journal of Applied Earth Observation and Geoinformation. – 2022. - Vol. 15. – P. 3730-3739.

318. Yang, J. Learning and transferring deep joint spectral–spatial features for hyperspectral classification / J. Yang, Y.Q. Zhao, J.C.W. Chan // IEEE Transactions on Geoscience and Remote Sensing. – 2017. - Vol. 55(8). – P. 4729–4742. DOI: 10.1109/TGRS.2017.2698503.

319. Yang, Z. Scalable optimization of neighbor embedding for visualization / Z. Yang, J. Peltonen, S. Kaski // 30th International Conference on Machine Learning, PMLR. – 2013. – Vol. 28(2). – P. 127-135.

320. Yao, C. A Collaborative Superpixelwise Autoencoder for Unsupervised Dimension Reduction in Hyperspectral Images / C. Yao, L. Zheng, L. Feng, F. Yang, Z. Guo, M. Ma // Remote Sensing. – 2023. – Vol. 15(17). – P. 4211. DOI: 10.3390/rs15174211.

321. Yeh, T.T. Efficient Parallel Algorithm for Nonlinear Dimensionality Reduction on GPU / T.T. Yeh, T.-Y. Chen, Y.-C. Chen, W.-K. Shih // 2010 IEEE International Conference on Granular Computing. – 2010. - P. 592-597. DOI: 10.1109/GrC.2010.145.

322. Yianilos, P.N. Data structures and algorithms for nearest neighbor search in general metric spaces / P.N. Yianilos // Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms. – 1993. - P. 311-321.

323. Yu, G. Image compression systems on board satellites / G. Yu, T. Vladimirova, M. Sweeting // Acta Astronautica. - 2009. - Vol. 64. - P. 988-1005. DOI: 10.1016/j.actaastro.2008.12.006.

324. Yu, L. Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution / L. Yu, H. Liu // Twentieth International Conference on Machine Learning (ICML-2003). – 2003. – P. 856-863.

325. Yuzkiv, R. Transform-based coding method for remote sensing hyperspectral data compression / R. Yuzkiv, V. Sergeev // Procedia Engineering. – 2017. – Vol. 201. – P. 249-257. DOI: 10.1016/j.proeng.2017.09.608.

326. Zhang, L. Study of the spectral mixture model of soil and vegetation in PoYang Lake area, China / L. Zhang, D. Li, Q. Tong, L. Zheng // International Journal of Remote Sensing. – 1998. - Vol. 19(11). - P. 2077-2084.

327. Zhang, Z. Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment / Z. Zhang, H. Zha // SIAM Journal on Scientific Computing. - 2005. – Vol. 26(1). – P. 313-338. DOI: 10.1137/s1064827502419154.

328. Zheng, Z. FPGA: Fast PatchFree Global Learning Framework for Fully End-to-End Hyperspectral Image Classification / Z. Zheng, Y. Zhong, A. Ma, L. Zhang // IEEE Transactions on Geoscience and Remote Sensing. - 2020. - Vol. 58(8). - P. 5612-5626. DOI: 10.1109/TGRS.2020.2967821.

329. Zhong, Y. A support vector conditional random fields classifier with a Mahalanobis distance boundary constraint for high spatial resolution remote sensing imagery / Y. Zhong, X. Lin, L. Zhang // IEEE Journal Of Selected Topics In Applied Earth Observations And Remote Sensing. – 2014. - Vol. 7(4). - P. 1314-1330.

330. Zhong, Y. WHU-Hi: UAV-borne hyperspectral with high spatial resolution (H2) benchmark datasets and classifier for precise crop identification based on deep convolutional neural network with CRF / Y. Zhong, X. Hu, C. Luo, X. Wang, J. Zhao, L. Zhang // Remote Sensing of Environment. - 2020. - Vol. 250. – P. 112012. DOI: 10.1016/j.rse.2020.112012.

331. Zhong, Z. Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework / Z. Zhong, J. Li, Z. Luo, M. Chapman // IEEE Transactions on Geoscience and Remote Sensing. – 2018. - Vol. 56(2). - P. 847-858.

332. Бабкин, В.Ф. Сжатие многоспектральных изображений для задач дистанционного зондирования земли из космоса / В.Ф. Бабкин, И.М. Книжный, К.Е. Хрекин // Современные проблемы дистанционного зондирования Земли из космоса. - 2004. - Т.1(1). - С. 330-332.

333. Борисов, А.Н., Понижение размерности методом градиентного спуска с использованием графических ускорителей / А.Н. Борисов, Е.В. Мясников // VI Международная конференция и молодёжная школа «Информационные технологии и нанотехнологии» (ИТНТ-2020). — 2020. — Т. 4. — С. 1047-1054.

334. Бураго, Д.Ю. Курс метрической геометрии. / Д.Ю. Бураго, Ю.Д. Бураго, С.В. Иванов. — Ижевск: Институт компьютерных исследований, 2004. – 512 с. - ISBN 978-5-93972-300-7.

335. Васильев, Н. Метрические пространства / Н. Васильев // Квант. — 1990. — N 1. – С. 17-23.

336. Васин, Д.Ю. Устранение информационной избыточности гиперспектральных растровых изображений методом «хорошо приспособленного» базиса / Д.Ю. Васин, В.П. Громов, П.А. Пахомов // V Международная конференция и молодёжная школа «Информационные технологии и нанотехнологии» (ИТНТ-2019). – 2019. - Т. 2 – С. 223-231.

337. Гашников М.В. Hierarchical grid interpolation for hyperspectral image compression / М.В. Гашников, Н.И Глумов // Компьютерная оптика. – 2014. - Том 38(1). – С. 87-93. ISSN 0134-2452.

338. Гашников, М. В. Методы компьютерной обработки изображений / М.В. Гашников, Н.И. Глумов, Н.Ю. Ильясова, В.В. Мясников, С.Б. Попов, В.В. Сергеев, В.А. Сойфер, А.Г. Храмов, А.В. Чернов, В.М. Чернов, М.А. Чичева, В.А. Фурсов; под ред. В. А. Сойфера. - 2-е изд., испр. - Москва: ФИЗМАТЛИТ, 2003. - 784 с. - ISBN 5-9221-0270-2.

339. Глумов, Н.И. Метод отбора информативных признаков на цифровых изображениях / Н.И. Глумов, **Е.В. Мясников** // Компьютерная оптика. – 2007. - Т. 31(3). - С. 73-76.

340. Егорова, А.А. Система признаков для расширенного суперпиксельного представления изображений / А.А. Егорова, В.В. Сергеев // Компьютерная оптика. – 2021. – Т. 45(4). – С. 562-574. – DOI: 10.18287/2412-6179-CO-876.

341. Замятин, А.В. Алгоритм сжатия гиперспектральных аэрокосмических изображений с учетом междиапазонной корреляции / А.В. Замятин, А.Ж. Саринова // Прикладная информатика. – 2013. – Т. 47(5). – С. 35-42.

342. Индекс пакетов Python (PyPI) // Python Software Foundation: сайт. – URL: https://pypi.org/ (дата обращения: 06.10.2024).

343. Каркищенко, А.Н. Применение понятия информативности в распознавании образов / А.Н. Каркищенко, А.Е. Лепский // Научно-техническая конференция «Многопроцессорные вычислительные и управляющие системы». – 2007. – С. 1-6.

344. Карпеев, С.В. Юстировка и исследование макетного образца гиперспектрометра по схеме Оффнера / С.В. Карпеев, С.Н. Хонина, А.Р. Мурдагулов, М.В. Петров // Вестник Самарского университета. Аэрокосмическая техника, технологии и машиностроение. – 2016. – Т. 15(1). – С. 197-206. DOI: 10.18287/2412-7329-2016-15-1-197-206.

345. Кремер, Н.Ш. Теория вероятностей и математическая статистика: учебник для студентов вузов / Н.Ш. Кремер. - 3-е изд., перераб. и доп. - М.: ЮНИТИ-ДАНА, 2010. - 551 с.

346. Кутин, Г.И. Методы ранжировки комплексов признаков. Обзор. / Г.И. Кутин // Зарубежная радиоэлектроника. – 1981. – Т. 9. - С. 54-70.

347. Минкин, А.С. Сжатие гиперспектральных данных методом главных компонент / А.С. Минкин, О.В. Николаева, А.А. Русских // Компьютерная оптика. – 2021. – Т. 45(2). – С. 235-244. DOI: 10.18287/2412-6179-CO-806.

348. **Мясников, Е. В.** Навигация по коллекциям цифровых изображений на основе методов автоматической классификации / Е. В. Мясников // Интернет-математика 2007: сборник работ участников конкурса научных проектов по информационному поиску. - 2007. — С. 144-152.

349. **Мясников, Е.В.** Автоматизированная интеллектуальная система поиска изображений в базе данных / Е.В. Мясников // Пояснительная записка к дипломной работе. - Самара, 2004. – 227 с.

350. **Мясников, Е.В.** Визуализация признаковых пространств на основе спектральных и текстурных характеристик / Е.В. Мясников // Информационные технологии и нанотехнологии (ИТНТ-2023): сборник трудов по материалам IX Международной конференции и молодежной школы. – 2023. – Т. 3. – С. 32572.

351. **Мясников, Е.В.** Определение параметров геометрических трансформаций для совмещения портретных изображений / Е.В. Мясников // Компьютерная оптика. – 2007. – Т. 31(3). – С. 77–82.

352. **Мясников, Е.В.** Развитие методов анализа гиперспектральных космических изображений с использованием нелинейных методов снижения размерности данных с пониженной вычислительной сложностью: отчет о НИР / Е.В. Мясников. — Самара, 2016.

353. **Мясников, Е.В.** Разработка метода навигации по коллекциям цифровых изображений / Е.В. Мясников // Труды 9-ой Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» - RCDL'2007. - 2007. - С. 185-194.

354. **Мясников, Е.В.** Сегментация гиперспектральных изображений на основе оценки размерности данных в пространственных областях . / Е. В. Мясников // Информационные технологии и нанотехнологии (ИТНТ-2021): сборник трудов по материалам VII Международной конференции и молодежной школы. - 2021. - Т. 2. - С. 024162.

355. Осовский, С. Нейронные сети для обработки информации / С. Осовский [Пер. с польского И.Д. Рудинского] . – М.:Финансы и статистика, 2002. - 344 с.

356. Печкурова, В.Р. Обзор существующих алгоритмов сжатия данных, получаемых от многозональных сканирующих устройств / В.Р. Печкурова // Современные проблемы дистанционного зондирования Земли из космоса. - 2021. - Т. 18(2). - С. 9-17.

357. Поляк, Б.Т. Метод Ньютона и его роль в оптимизации и вычислительной математике / Б.Т. Поляк // Труды ИСА РАН. - 2006. - Т. 28. - С.48-66.

358. Пухкий, К.К. Классификация гиперспектральных изображений ДЗЗ с помощью высокоуровневых признаков на основе эмпирических мод / К.К. Пухкий, В.Е. Турлапов // Труды Международной конференции по компьютерной графике и зрению "Графикон". - 2023. - С. 743-756.

359. Сергеев, В.В. Информационная технология компрессии изображений в системах оперативного дистанционного зондирования / М.В. Гашников, Н.И. Глумов, В.В. Сергеев // Известия Самарского научного центра Российской академии наук. – 1999. – Т. 1. – С. 99-107.

360. Толстова, Ю. Н. Основы многомерного шкалирования: учебное пособие / Ю. Н. Толстова. — М.: КДУ, 2006. — 160 с.

361. Ту, Дж. Принципы распознавания образов / Дж. Ту, Р. Гонсалес; перев. с англ. И. Б. Гуревича; под ред. Ю. И. Журавлева. - М.: Мир, 1978 . - 412 с.

362. Фирсов, Н.А. Нейросетевая классификация гиперспектральных изображений растительности с формированием обучающей выборки на основе адаптивного вегетационного индекса / Н.А. Фирсов, В.В. Подлипнов, Н.А. Ивлиев, П.П.

Николаев, С.В. Машков, П.А. Ишкин, Р.В. Скиданов, А.В. Никоноров // Компьютерная оптика. – 2021. – Т. 45(6). – С. 887-896. DOI: 10.18287/2412-6179-CO-1038.

363. Фу, К. Структурные методы в распознавании образов / К. Фу. — М.: Мир, 1977. — 320 с.

364. Фукунага, К. Введение в статистическую теорию распознавания образов / К. Фукунага. - М.: Наука, 1979. — 368 с.

365. Хорн, Р. Матричный анализ: Пер. с англ. / Р. Хорн, Ч. Джонсон. – М.: Мир, 1989. – 655 с.

366. Шапиро, Л. Компьютерное зрение / Л. Шапиро, Дж. Стокман; под ред. С.М. Соколова [пер. с англ. А.А. Богуславского]. — М.: БИНОМ. Лаборатория знаний, 2006. — 752 с.

Приложение А. Обзор существующих методов снижения размерности

А.1. Анализ главных компонент

Метод анализа главных компонент (англ., Principal component analysis, PCA) [232, 114] является наиболее известным линейным методом снижения размерности, используемым в широком диапазоне приложений. Этот метод также входит в состав ряда нелинейных методов снижения размерности.

PCA выполняет поиск такой линейной проекции исходных данных в подпространство меньшей размерности, которая максимизирует дисперсию данных.

Общая идея PCA состоит в решении задачи о собственных значениях:

$$\mathbf{C} = \mathbf{W}^T \mathbf{\Lambda} \mathbf{W}$$

где **C** - ковариационная матрица $\mathbf{C} = \mathbf{X}^{\mathrm{T}}\mathbf{X}$,

X – матрица, составленная из исходных векторов данных $\mathbf{X} = [x_1, x_2, \ldots x_N]$, предполагается, что вектора исходных данных центрированы относительно начала координат $\left(\sum_{i=1}^{N} x_i = 0\right)$,

W - матрица собственных векторов,

Λ - диагональная матрица собственных значений λ_k ковариационной матрицы.

После последующей сортировки собственных значений в порядке убывания и соответствующей перестановки столбцов в матрице собственных векторов **W** линейную проекцию набора данных **X** можно получить с помощью усечения матрицы **W**:

$$\mathbf{Y} = \mathbf{X}\mathbf{W}_m,$$

где $\mathbf{W}_m$ - усеченная матрица главных компонент размером M x m.

А.2. Случайные проекции

Случайная проекция представляет собой линейное отображение данных в случайное подпространство [72, 304].

Пусть задан набор из m случайных векторов $\{v_1, v_2,.. v_m\}$ в пространстве R^M, $\mathbf{V}=[v_1, v_2,.. v_m]$ – случайная матрица, составленная из этих векторов. Случайной

проекцией назовем отображение множества векторов X из исходного пространства R^M во множество векторов Y пространства меньшей размерности R^m:

$$X \to Y,\ X \subset R^{\mathrm{m}},\ Y \subset R^{\mathrm{m}},$$

рассчитываемых как $\mathbf{Y}=\mathbf{V}^{\mathrm{T}}\mathbf{X}$, где $\mathbf{X}=[x_1,\ x_2,..\ x_m]$ и $\mathbf{Y}=[y_1,\ y_2,..\ y_m]$ – матрицы, составленные из векторов множеств X и Y, соответственно.

Использование случайных проекций основывается на лемме Джонсона-Линденштрауса (Johnson-Lindenstrauss) [128]. Метод случайных проекций, называемый также методом вложений Липшица (Lipschitz embedding) или Джонсона-Линденштрауса (JL embedding) впервые применялся в работе [125] и впоследствии многократно использовался.

Общая схема использования случайных проекций состоит в генерации матриц случайной проекции с последующим отображением многомерных данных в пространство сниженной размерности. Случайные проекции могут строиться на основе различных распределений . В частности, для гауссовских распределений верно следующее утверждение [59, 304]:

Пусть **X -** матрица, составленная из векторов x_i, i=1..N в исходном пространстве R^M, **V** - случайная матрица размером kxM, элементы которой v_{ij}~N(0,1) распределены по гауссовскому закону, ε - число в диапазоне $0<\varepsilon<1$, k – целое число, удовлетворяющее соотношению

$$k \geq \frac{4\ln(N)}{\varepsilon^2/2-\varepsilon^3/3},$$

а **Y**-матрица, полученная в результате преобразования

$$\mathbf{Y}=\frac{1}{\sqrt{k}}\mathbf{V}\mathbf{X}$$

и трактуемая как набор векторов $\mathbf{Y}=[y_1, y_2,\ldots y_N]$. Тогда для любой пары векторов x_i, x_j и соответствующих им векторов y_i, y_j с вероятностью равной, по крайней мере, $1\text{-}2/N^2$ выполняется следующее соотношение:

$$(1-\varepsilon)\left\|x_i-x_j\right\|^2 \leq \left\|y_i-y_j\right\|^2 \leq (1+\varepsilon)\left\|x_i-x_j\right\|^2.$$

Приведенная теорема устанавливает, что при случайной проекции набора данных, выполненной с использованием указанного распределения, евклидовы расстояния между точками набора данных отклонятся от исходных значений не более, чем на ε с заданной вероятностью. Подобные теоремы существуют и для

других распределений [304]. Наиболее широкое использование случайные проекции получили при создании методов приближенного поиска ближайшего соседа.

А.3. Многомерное шкалирование

Существует целая группа методов многомерного шкалирования (англ., multidimensional scaling). Изначально, методы многомерного шкалирования применялись в психологии и социологии для поиска пространства восприятия. Сегодня эти инструменты достаточно широко используются в анализе данных.

Термин «шкалирование» означает частный случай измерения [360], а само многомерное шкалирование представляет собой поиск значений переменных, соответствующих положению измеряемых объектов в пространстве восприятия. Исходными данными при многомерном шкалировании выступает матрица близости, схожести (англ., similarity, proximity) или, наоборот, матрица различий, расстояний (англ., dissimilarity, distance). В зависимости от того, как происходило формирование указанных матриц, различают метрическое и неметрическое многомерное шкалирование (MDS). В случае метрического MDS матрицы близости или различий получаются в результате оценки по интервальным шкалам (упорядоченные шкалы с равными интервалами), в случае неметрического – по порядковым. Исходя из решаемой в работе задачи, нас будет интересовать метрический MDS.

На вход MDS принимает симметричную, неотрицательную матрицу рассогласований $\mathbf{\Delta} = [\delta_{ij}]$ размером NxN, содержащую нули на главной диагонали (в нашем случае можно задать δ_{ij} равными расстояниям между точками во входном пространстве). Задача состоит в нахождении N точек $y_i, i=1..N$ в евклидовом пространстве малой размерности R^m таким образом, чтобы евклидовы расстояния между этими точками аппроксимировали данные рассогласования $d(y_i,y_j) \approx \delta_{ij}$. Для оценки качества приближения вводится понятие стресса (stress):

$$\varepsilon_{MDS} = \sum_{i=1}^{N}\sum_{j=1}^{N} \rho_{i,j} \left(\delta_{ij} - d(y_i,y_j)\right)^2 \qquad \text{(A.1)}$$

Матрица весов $\mathbf{R} = [\rho_{ij}]$ также симметричная, неотрицательная и содержит нули на главной диагонали. Веса могут быть использованы для работы с пропущенными измерениями. Когда рассогласования δ_{ij} отсутствуют, соответствующие веса ρ_{ij} могут

быть заданы нулями, в то время как веса, соответствующие известным рассогласованиям – единицами.

SMACOF

Одним из эффективных способов решения рассматриваемой задачи MDS является алгоритм SMACOF, изложенный в настоящем подразделе по работе [62].

Положим (без ограничения общности), что имеют место ограничения

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\rho_{i,j}\ \delta_{ij}^{\ 2}=1,$$

и матрица весов **R** неприводимая [60] так, что задача минимизации стресса не разделяется на множество независимых подзадач.

Приведенный выше стресс может быть приведен в виду:

$$\varepsilon_{MDS}=\sum_{i=1}^{N}\sum_{j=1}^{N}\rho_{i,j}\delta_{ij}^{\ 2}+\sum_{i=1}^{N}\sum_{j=1}^{N}\rho_{i,j}d^2(y_i,y_j)-2\sum_{i=1}^{N}\sum_{j=1}^{N}\rho_{i,j}\delta_{ij}d(y_i,y_j)=s_1+s_2-2s_3$$

Из введенных ограничений следует, что первое слагаемое s_1 равно единице. Второе слагаемое s_2 представляет собой взвешенную сумму квадратов расстояний и, следовательно, является выпуклой квадратичной функцией. Третье слагаемое $-2s_3$ представляет собой вогнутую функцию - взятую с минусом взвешенную сумму расстояний.

Пусть e_i – единичный вектор, i-ый компонент которого равен единице, а остальные – нулю. Определим матрицу $\mathbf{A}_{ij}=(e_i-e_j)(e_i-e_j)^{\mathrm{T}}$, представляющую собой разреженную матрицу с отличными от нуля элементами $a_{ii}=a_{jj}=1$ и $a_{ij}=a_{ji}=-1$.

Далее определим

$$\mathbf{V}_2=\sum_{i=1}^{N}\sum_{j=1}^{N}\rho_{i,j}\ \mathbf{A}_{ij}\ .$$

Теперь второе слагаемое в выражении стресса $s_2=\mathbf{tr}\ \mathbf{Y}^{\mathrm{T}}\mathbf{V}_2\mathbf{Y}$.

Определим также

$$\mathbf{V}(\mathbf{Y})=\sum_{i=1}^{N}\sum_{j=1}^{N}\rho_{ij}\tilde{\delta}_{ij}(\mathbf{Y})\mathbf{A}_{ij}\ ,$$

где $\tilde{\delta}_{ij}(\mathbf{Y})=\begin{cases}\delta_{ij}-d(y_i,y_j) & if\ d(y_i,y_j)>0\\ 0 & if\ d(y_i,y_j)=0\end{cases}.$

Теперь третье слагаемое $s_3=\mathbf{tr}\ \mathbf{Y}^{\mathrm{T}}\mathbf{V}(\mathbf{Y})\mathbf{Y}$.

Введем опорную конфигурацию $\mathbf{P}=[p_{ij}]_{i=1..N, j=1..p}$ и определим $V(\mathbf{P})$:

$$V(\mathbf{P}) = \sum_{i=1}^{N}\sum_{j=1}^{N} \rho_{ij}\tilde{\delta}_{ij}(\mathbf{P})\mathbf{A}_{ij}\ ,\ \tilde{\delta}_{ij}(\mathbf{P}) = \begin{cases} \delta_{ij} - d(p_i, p_j) & if\ d(p_i, p_j) > 0 \\ 0 & if\ d(p_i, p_j) = 0 \end{cases}.$$

Из неравенства Коши-Шварца следует, что для любых **Y** и **P**:

$$s_3 = \mathbf{tr}\ \mathbf{Y}^{\mathrm{T}} V(\mathbf{Y})\mathbf{Y} \geq = \mathbf{tr}\mathbf{Y}^{\mathrm{T}} V(\mathbf{P})\mathbf{P}.$$

Миноризируя выпуклую s_3 линейной функцией $\mathbf{tr}\ \mathbf{V}^{\mathrm{T}} V(\mathbf{P})\mathbf{P}$, можно мажоризировать стресс:

$$\varepsilon_{\mathrm{MDS}} = s_1 + s_2 - 2s_3 = 1 + \mathbf{tr}\ \mathbf{Y}^{\mathrm{T}}\mathbf{V}_2\mathbf{Y} - 2\ \mathbf{tr}\ \mathbf{Y}^{\mathrm{T}} V(\mathbf{Y})\mathbf{Y} \leq$$

$$\leq 1 + \mathbf{tr}\ \mathbf{Y}^{\mathrm{T}}\mathbf{V}_2\mathbf{Y} - 2\ \mathbf{tr}\ \mathbf{Y}^{\mathrm{T}} V(\mathbf{P})\mathbf{P} = \varepsilon_{\mathrm{Maj}}.$$

Минимизация $\varepsilon_{\mathrm{Maj}}$ может быть сделана следующим образом:

$$\frac{\partial \varepsilon_{Maj}}{\partial \mathbf{Y}} = 2\mathbf{V}_2\mathbf{Y} - 2V(\mathbf{P})\mathbf{P} = 0\ .$$

Таким образом (преобразование [96])

$$\hat{\mathbf{Y}} = \mathbf{V}_2^{+} V(\mathbf{P})\mathbf{P}\ ,$$

где $\mathbf{V}_2^{+} = (\mathbf{V}_2 + N^{-1}\mathbf{I}\mathbf{I}^T)^{-1} - N^{-1}\mathbf{I}\mathbf{I}^T$.

Для случая единичных весов $\rho_{ij}=1$, $i \neq j$: $\mathbf{V}_2 = 2N(\mathbf{I} - N^{-1}\mathbf{I}\mathbf{I}^T)$, а решение $\hat{Y} = N^{-1}V(P)P$.

Алгоритм минимизации стресса выглядит следующим образом:

1. Выбор начальной конфигурации (инициализация) $\mathbf{Y}(0)$.
   $\mathbf{P} = \mathbf{Y}(0)$.
2. На итерации t:
   вычисление $\boldsymbol{Y}(t)$ и стресса $\varepsilon_{\mathrm{MDS}}(t)$.
3. Если $(\varepsilon_{\mathrm{MDS}}(t) - \varepsilon_{\mathrm{MDS}}(t-1)) < \in$ или $t = T$, то остановка,
   иначе переход к шагу 2.

Основным преимуществом SMACOF является невозрастание стресса и линейная сходимость [61].

A.4. Отображение Сэммона

Отображение Сэммона (Sammon's mapping), введенное J.W. Sammon в работе [257], основано на принципе сохранения попарных евклидовых расстояний между точками данных.

Метод минимизирует следующую ошибку отображения данных:

$$\varepsilon = \frac{1}{\sum_{i=1}^{N}\sum_{j=i+1}^{N} d(x_i, x_j)} \cdot \sum_{i=1}^{N}\sum_{j=i+1}^{N} \frac{\left(d(x_i, x_j) - d(y_i, y_j)\right)^2}{d(x_i, x_j)} \qquad \text{(A.2)}$$

где $d(x_i,x_j)$ и $d(y_i,y_j)$ - евклидово расстояние между точками i и j в исходном многомерном пространстве и выходном пространстве малой размерности, соответственно.

Для минимизации ошибки (A.2) используется информация о производных первого и второго порядка (учитываются диагональные элементы гессиана) [257]. Решение ищется путем применения следующего рекуррентного соотношения до стабилизации точек $y_i(t_0)$, $i = 1..N$ в выходном пространстве малой размерности:

$$y_{ik}(t+1) = y_{ik}(t) - \alpha \frac{\partial \varepsilon}{\partial y_{ik}} \Bigg/ \left| \frac{\partial^2 \varepsilon}{\partial y_{ik}^2} \right|, \qquad \text{(A.3)}$$

где α - постоянный коэффициент (размер шага), t - номер итерации, а частные производные задаются выражениями:

$$\begin{aligned}
&\frac{\partial \varepsilon}{\partial y_{ik}} = -\frac{2}{c} \sum_{j=1\,(i\neq j)}^{N} \frac{d(x_i, x_j) - d(y_i, y_j)}{d(x_i, x_j) d(y_i, y_j)} \left(y_{ik} - y_{jk}\right), \\
&\frac{\partial^2 \varepsilon}{\partial y_{ik}^2} = -\frac{2}{c} \sum_{j=1\,(i\neq j)}^{N} \frac{1}{d(x_i, x_j) d(y_i, y_j)} \left(\left(d(x_i, x_j) - d(y_i, y_j)\right) - \frac{d(x_i, x_j)\left(y_{ik} - y_{jk}\right)^2}{d^2(y_i, y_j)} \right), \\
&c = \sum_{i=1}^{N}\sum_{j=i+1}^{N} d(x_i, x_j).
\end{aligned} \qquad \text{(A.4)}$$

Допускается инициализация $y_i(t_0)$, $i = 1..N$ случайными значениями или с использованием метода главных компонент.

A.5. Самоорганизующиеся карты Кохонена

Нейронные сети Кохонена, называемые также самоорганизующимися картами Кохонена (англ., Self-organizing maps, SOM) представляют собой однослойные сети, в которой нейроны расположены в узлах кристаллической решетки некоторой малой

размерности, и каждый нейрон соединен со всеми M компонентами входного вектора $x \in R^M$ [355]. Для $M = 2$ схематически это можно изобразить следующим образом (рисунок А.1 и описание сети взяты из работы автора [349*]).

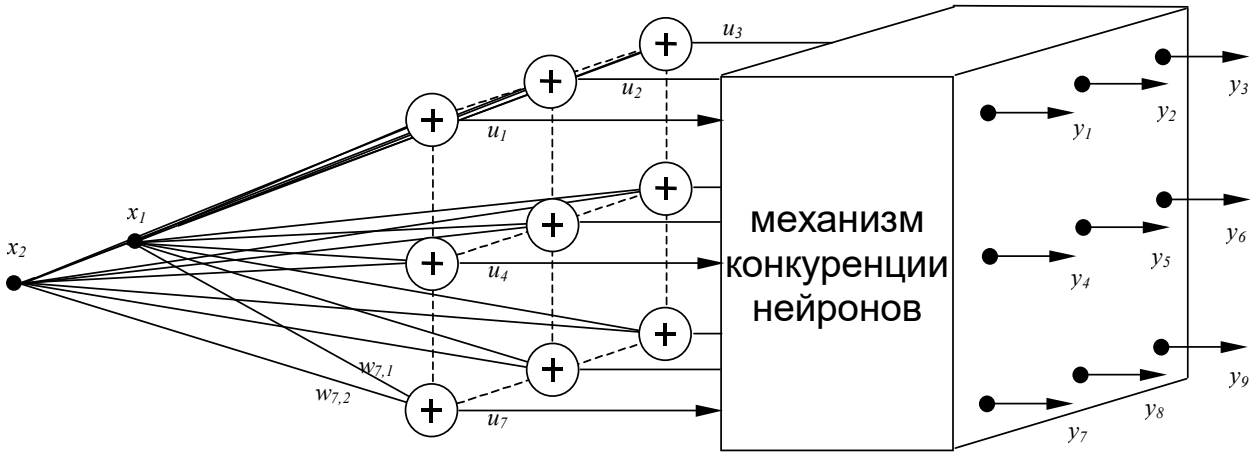


Рисунок А.1 - Структура сети Кохонена

Веса синаптических связей образуют для нейрона i, i=1..K вектор $w_i=(w_{i1},w_{i2},..,w_{iM})^T$. Каждый нейрон сети Кохонена выполняет взвешенное суммирование входных сигналов: $u_i = w_i^T x$, i=1..K. При активации сети вектором x, в конкурентной борьбе побеждает тот нейрон из общего числа K, веса которого в наименьшей степени отличаются от соответствующих компонент входного вектора. Для нейрона-победителя ν выполняется соотношение

$$d(x, w_\nu) = \min_{1 \le i \le K} d(x, w_i).$$

Здесь $d(x,w)$ - расстояние между векторами x и w. Выходной сигнал y_ν выигравшего нейрона устанавливается в единицу, выходы остальных нейронов y_i становятся равными нулю. Вокруг нейрона-победителя ν образуется топологическая окрестность $S_\nu(t)$ с определенной энергетикой, уменьшающейся с течением времени t. Нейрон-победитель и все нейроны из его окрестности подвергаются адаптации, в ходе которой их векторы весов изменяются в направлении вектора x по правилу Кохонена:

$$w_i(t+1) = w_i(k) + \eta_i(t)(x - w_i(t)), \text{ для всех } i \in S_\nu(k).$$

В этой формуле $\eta_i(t)$ - коэффициент обучения i-го нейрона из окрестности $S_\nu(t)$ в момент времени t. Значение $\eta_i(t)$ уменьшается с увеличением расстояния между i-ым нейроном и нейроном-победителем ν. Веса нейронов, находящихся за пределами окрестности $S_\nu(t)$ не изменяются. Размер окрестности и коэффициенты обучения являются функциями, значения которых со временем уменьшаются.

В литературе [355] отмечено, что существует доказательство [244] эквивалентности процедуры адаптации по приведенной выше формуле градиентному алгоритму обучения, основанному на минимизации целевой функции

$$\varepsilon(w) = \frac{1}{2} \cdot \sum_{i,j,k} S_i(x(k))\left[x_j(k) - w_{ij}(k)\right]^2.$$

Здесь $S_i(x(k))$ представляет собой функцию определения окрестности, изменяющуюся в процессе обучения.

Сеть Кохонена имеет несколько разновидностей, являющихся частными случаями рассмотренного выше алгоритма обучения. Один из них, называемый алгоритмом WTA (или сетью WTA), входит в число наилучших и наиболее быстрых алгоритмов. В соответствии с этим алгоритмом, после предъявления вектора $\boldsymbol{x}$, рассчитывается активность каждого нейрона и победителем становится тот нейрон, для которого скалярное произведение $(x^T w)$ оказывается наибольшим. При использовании нормализованных входных векторов, равносильным будет нахождение минимального евклидова расстояния. Далее победитель уточняет свои веса по правилу:

$$w_v(t+1) = w_v(t) + \eta(t)(x - w_v).$$

Веса других нейронов не корректируются. Таким образом, в алгоритме WTA в окрестности $S_v(k)$ в любой момент времени k содержится только нейрон-победитель.

Классический алгоритм Кохонена обучения нейросети относится к одному из самых старых алгоритмов обучения сетей с самоорганизацией на основе конкуренции, и в настоящее время существуют различные его версии. В классическом алгоритме Кохонена сеть инициализируется путем приписывания нейронам определенных позиций в пространстве и связывания их с соседями на постоянной основе. В момент выбора победителя уточняются не только его веса, но также и веса его соседей, находящихся в ближайшей его окрестности. Таким образом, нейрон победитель подвергается адаптации вместе со своими соседями.

Уточнение весов нейронов происходит по правилу:

$$w_i(t) = w_i(t\text{-}1) + \eta(t) \cdot G(i,\mathrm{v}) \cdot (x - w_i).$$

В приведенном выражении степень адаптации нейронов-соседей определяется функцией $G(i, v)$ b зависит от расстояния между i-м нейроном и v-м нейроном-победителем. Расстояние $\tilde{d}(i,v)$ между нейроном-победителем v и i-ым нейроном

здесь может как измеряться в пространстве весов нейронов $\tilde{d}(u,v) = d(w_u, w_v)$, так и определяться взаимным положением нейронов в кристаллической решетке нейронов сети. Например, при соседстве прямоугольного типа каждый нейрон, находящийся в окрестности победителя, адаптируется в равной степени. При соседстве гауссовского типа функция соседства определяется формулой

$$G(u,v) = e^{-\frac{1}{2}\left(\frac{\tilde{d}(u,v)}{\lambda}\right)^2}.$$

Здесь λ - параметр, называемый уровнем соседства.

Как правило, гауссовское соседство дает лучшие результаты обучения и обеспечивает лучшую организацию сети, чем соседство прямоугольного типа.

Другой алгоритм обучения нейронной сети Кохонена, алгоритм нейронного газа [167], был предложен М. Мартинесом, С. Берковичем и К. Шультеном и так назван авторами из-за подобия его динамики движению молекул газа. В этом алгоритме на каждой итерации все нейроны сортируются в зависимости от их расстояния до вектора x и размечаются в последовательности, соответствующей увеличению удаленности. Значение функции соседства для каждого нейрона в ряду отсортированных нейронов определяется по формуле

$$G(i,v) = e^{-\frac{m(i)}{\lambda}},$$

в которой $m(i)$ обозначает позицию нейрона в отсортированном ряду, а λ - параметр, аналогичный уровню соседства, уменьшающийся с течением времени. При $\lambda=0$ адаптации подвергается только нейрон-победитель, и алгоритм превращается в обычный алгоритм WTA.

Как видно из приведенного описания, нейронные сети Кохонена решают, прежде всего, задачу кластеризации или векторного квантования. Тем не менее, во многих источниках такие сети позиционируются как средство нелинейного снижения размерности. Можно сказать, что это верно лишь отчасти. Нейронные сети Кохонена, в которых нейроны расположены в узлах кристаллической решетки и используют при адаптации понятие пространственного соседства нейронов в решетке, действительно выполняют отображение из входного пространства в дискретное пространство, определяемое кристаллической решеткой, а также способны выполнять обратное отображение. Попытки же отойти от жесткой пространственной структуры, такие как

алгоритм WTA или нейронного газа приводят к более эффективным результатам векторного квантования, но приводят к потере способности создавать хоть какое-то выходное представление данных. Для целей отображения в этих случаях могут использоваться другие методы снижения размерности, например, нелинейное отображение.

A.6. CCA

Анализ криволинейных компонент (Curvilinear component analysis, CCA) был предложен P. Demartines и J. Hérault в работе [63].

Хотя метод формулируется в терминах нейросетей, он в многом схож с рассмотренным ранее методом нелинейного отображения. Каждый из N нейронов сети соединен, с одной стороны, с векторами входных данных, а, с другой стороны, формирует по одному из выходных векторов сети. На предварительном этапе осуществляется векторное квантование входных данных, целью которого является равномерное квантование исходного пространства, не зависящее, по возможности, от плотности распределения данных. На втором этапе осуществляется отображение данных в пространство сниженной размерности.

Собственно отображение выполняется, как и в методе нелинейного отображения, путем выполнения итеративной оптимизационной процедуры. В качестве функционала ошибки рассматривается следующее выражение:

$$\varepsilon_{ED} = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1\,(i\neq j)}^{N} F(d(y_{i,}y_j),\lambda_y)\left(d(x_i,x_j)-d(y_{i,}y_j)\right)^2$$

Функция $F()$, определяющая окрестность, в которой, по замыслу алгоритма, должна быть сохранена локальная топология, выбирается ограниченной монотонно убывающей функцией. В качестве нее в рассматриваемой работе использована функция

$$F(d,\lambda) = \begin{cases} 1, & \textit{если } (d \le \lambda), \\ 0, & \textit{иначе.} \end{cases}$$

Оптимизация функционала выполняется алгоритмом, похожим на алгоритм стохастического градиентного спуска. Уточнение положения векторов в выходном пространстве на каждом шаге алгоритма осуществляется путем коррекции всех векторов y_j, $j=1..N$, $j\neq i$ по одному случайно выбранному вектору y_i:

$$y_j(t+1) = y_j(t) + \alpha(t)\, F(d(y_{i,}y_j), \lambda_y)\left(d(x_i, x_j) - d(y_{i,}y_j)\right)\frac{y_j(t) - y_i(t)}{d(y_{i,}y_j)}.$$

Шаг α убывает с течением времени по закону $\alpha(t)= \alpha_0/(t+1)$.

Авторы отмечают более быструю сходимость процесса оптимизации и достижение более глубоких минимумов функционала ошибки по сравнению с алгоритмом стохастического градиентного спуска, впрочем, без теоретических доказательств.

A.7. ISOMAP

Метод Isomap был представлен J.B. Tenenbaum в статьях [290, 289]. Основная идея этого метода состоит в использовании геодезических расстояний между точками вместо евклидовых расстояний в классическом метрическом многомерном шкалировании (MDS).

Кратко этот метод можно описать так:

1. Для набора $X = \{x_i\}$, $i = 1..N$ точек данных в многомерном пространстве R^M найти соседние точки N_i каждой точки данных x_i, $i = 1..N$.

2. Построить граф окрестности $G = <V, E>$, вершины которого $V=\{v_i\}$, $i = 1..N$ соответствуют точкам данных x_i. Ребра $E=\{(v_i, v_j)\}$, $v_i \in V$, $v_j \in N_i$ соединяют только соседние точки данных, а веса w_{ij} ребер (v_i, v_j) определяются евклидовыми расстояниями между соответствующими точками данных:

$$w_{i,j} = \sqrt{\left(x_i - x_j\right)^T \left(x_i - x_j\right)} \qquad \text{(A.5)}$$

3. Вычислить кратчайшие расстояния $\mathbf{W} = \left(\widetilde{w}_{i,j}\right)_{i,j=1}^{N}$ между всеми парами точек данных, используя алгоритм Флойда-Уоршалла (или Дейкстры).

4. Выполнить многомерное шкалирование для вычисления координат $Y = \{y_i\}$, $i = 1..N$ точек данных в пространстве сниженной размерности.

A.8. Kernel PCA

В методе Kernel PCA [259] выбирается некоторая функция Φ, отображающая аргумент в пространство, возможно, очень высокой размерности

$$\Phi: \mathrm{R}^M \rightarrow \mathrm{R}^L,\ L>M,$$

и создается ядро N x N:

$$K=[k(x_i,x_j)]_{i,j=1..N}$$

$$k(x,y)=(\Phi(x),\Phi(y))=\Phi(x)^T\Phi(y)$$

Далее вычисляется проекция данных x на главные компоненты V_k в виде

$$V_k^T\Phi(x)=(\mathrm{Sum}_{i=1..N}a_{ki}\Phi(x_i))^T\Phi(x),$$

где $\Phi(x_i)\Phi(x)$ - скалярное произведение (элементы ядра), а a_{ki} вычисляются как собственные вектора ядра K:

$$N\lambda a=Ka,$$

нормированные к единице.

Так как $\Phi(x_i)$ необязательно центрированы даже при условии центрирования самих xi, при применении метода используется центрированное ядро:

$$K_c = K - (1/N)JK - (1/N)KJ + (1/N^2)JKJ,$$

где J – матрица единиц N x N.

A.9. MVU

Метод развертки максимальной дисперсии (maximum variance unfolding, MVU) – метод, предложенный K.Q. Weinberger, L.K. Saul и F.Sha в работе [309]. Первая версия метода, изложенная в работах [307, 308] была названа полуопределенным вложением (semidefinite embedding).

Основная идея метода заключается в максимизации дисперсии путем развертки набора данных при сохранении расстояний и углов между ближайшими соседями. Это достигается путем построения графа соседства с последующим добавлением связей таким образом, чтобы подграф, образуемый любой вершиной с ее k непосредственными соседями, был полносвязным (полносвязный k+1 клик). В таком случае сохранение задаваемых ребрами графа расстояний при отображении в пространство меньшей размерности можно считать достаточным и для сохранения углов. Описываемая задача может быть сформулирована следующим образом:

$$\Phi = \sum_{i=1}^{N}\sum_{j=1}^{N}\left\|y_i - y_j\right\|^2 \to \max$$

$$\textit{при ограничениях}: \quad \left\|y_i - y_j\right\| = d(x_i,x_j),\ i=1..N, j\in\eta_i$$

$$\sum_{i=1}^{N} y_i = 0$$

Здесь x_i – исходные позиции точек в R^M, y_i – искомые позиции точек в пространстве R^m сниженной размерности, η_i – множество индексов точек, являющихся k ближайшими соседями точки x_i.

В самом методе используется несколько модифицированная постановка задачи с использованием матрицы Грамма $\mathbf{\Gamma} = \left(\gamma_{ij}\right)_{i,j=1}^{N}$ скалярных произведений $\gamma_{ij} = y_i y_j$ для совокупности центрированных по началу координат векторов y_i, i= 1..N в выходном пространстве.

Метод состоит из следующих шагов:

1. Построение графа соседства $G = <V, E>$, вершины которого $V=\{v_i\}$, i = *1..N* соответствуют точкам данных x_i. Ребра $E=\{(v_i, v_j)\}$, $v_i \in V$, $v_j \in \eta_i$ соединяют только соседние точки данных таким образом, чтобы все подграфы, образуемые любыми вершинами $v_i \in V$ с их k непосредственными соседями $v_j \in \eta_i$, были полносвязными. Веса ребер определяются евклидовыми расстояниями между соответствующими точками данных.

2. Вычисление матрицы Грамма $\mathbf{\Gamma}$ путем решения следующей задачи полуопределенного программирования:

$$\Phi = tr(\mathbf{\Gamma}) \to \max$$

при ограничениях :

$$\mathbf{\Gamma} \succ= 0$$

$$\gamma_{ii} + \gamma_{jj} - 2\gamma_{ij} = d(x_i, x_j), \quad i = 1..N, j \in \eta_i$$

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\gamma_{ij} = 0$$

Первое ограничение здесь означает, что на матрицу $\mathbf{\Gamma}$ накладываются естественные ограничения, предъявляемые к матрицам скалярных произведений: симметричность и неотрицательность собственных значений. Указанная задача принадлежит к классу задач полуопределенного программирования [299] и решается в оригинальной работе с использованием пакета CSDP [26].

3. Путем диагонализации матрицы грамма $\mathbf{\Gamma}$ восстанавливаются искомые положения точек y_i, i=1..N в пространстве малой размерности.

$$y_{\alpha i} = \sqrt{\lambda_\alpha u_{\alpha i}},$$

где $u_{\alpha i}$ – i-ый элемент собственного вектора с индексом α и соответствующим собственным значением λ_α.

Несомненным достоинством подхода является то, что оптимизируемый на шаге 2 функционал является выпуклым и ограниченным сверху, а сама задача всегда имеет решение [309]. Недостатком является длительное время решения указанной задачи.

A.10. Локально линейное вложение

Метод локально-линейного вложения (англ., Locally linear embedding, LLE) был предложен S.T. Roweis и L.K. Saul в статье [252].

Этот метод основан на идее о том, что каждая конкретная точка данных и ее соседи лежат близко к локально линейному участку нелинейного многообразия и могут быть восстановлены как линейная комбинация своих соседей как в многомерном пространстве, так и в пространстве сниженной размерности.

Метод состоит из следующих трех шагов:

1. По заданному набору $X = \{x_i\}$, $i = 1..N$ точек данных в многомерном пространстве R^M, найти множество η_i индексов соседних точек для каждой из точек данных x_i, $i = 1..N$.

2. Найти веса реконструкции $\mathbf{W}=(w_{ij})$, минимизируя ошибку реконструкции

$$\varepsilon(X) = \sum_i \left(x_i - \sum_j \left(w_{ij} x_j \right) \right)^2$$

с ограничениями:

$$\begin{aligned} &\sum_j w_{ij} = 1, i = 1..N, \\ &w_{ij} = 0, i = 1..N,\ j \notin \eta_i. \end{aligned} .$$

3. Отобразить точки данных $\{x_i\}$, $i = 1..N$ в их представления $\{y_i\}$, $i = 1..N$ сниженной размерности, путем минимизации

$$\Phi(Y) = \sum_i \left(y_i - \sum_j \left(w_{ij} y_j \right) \right)^2 .$$

A.11. Лапласианские собственные карты

Метод лапласианских собственных карт (англ., Laplacian eigenmaps) был предложен M. Belkin и P. Niyogi в статье [18].

Этот метод основан на разложении по собственным значениям матрицы Лапласа графа.

Метод состоит из следующих трех шагов:

1. Для набора $X = \{x_i\}$, $i = 1..N$ точек данных в многомерном пространстве R^M найти множество η_i индексов соседних точек для каждой точки данных x_i, $i = 1..N$.
2. Построить граф окрестности $G = <V, E>$, вершины которого $V = \{v_i\}$, $i = 1..N$ соответствуют точкам данных x_i. Ребра $E = \{(v_i, v_j)\}$, $v_i \in V$, $v_j \in \eta_i$ соединяют только соседние точки данных, а веса ребер $\mathbf{W}=(w_{ij})$ равны единице $w_{ij}=1$ (первый вариант) или определены ядром Heat (второй вариант):

$$w_{ij} = \exp\left(- \left\| x_i - x_j \right\|^2 / t \right) \tag{A.6}$$

3. Построить диагональную матрицу $\mathbf{D}=(\delta_{ij})$:

$$\delta_{ii} = \sum_j w_{ij}$$

и матрицу Лапласа $\mathbf{L} = \mathbf{W} - \mathbf{D}$.

Затем решить задачу о собственных векторах:

$$\mathbf{LY} = \lambda \mathbf{DY} .$$

Решение $\mathbf{Y}_m$ в пространстве сниженной размерности R^m задается первыми m векторами $y_1..y_m$ в матрице $\mathbf{Y}$ после перестановки векторов y_i, $i = 1..M$ согласно их собственным значениям.

A.12. Выравнивание локальных касательных подпространств

Метод выравнивания локальных касательных подпространств (англ., Local Tangent Space Alignment, LTSA) был предложен Z. Zhang и H. Zha в работе [327]. Метод основан на идее представления локальной окрестности точек с использованием касательных подпространств с последующим выравниванием этих подпространств для получения единого представления исследуемого многообразия в

пространстве меньшей размерности. Метод исходит из предположения о том, что при правильном развертывании многообразия все касательные гиперплоскости должны быть выровнены.

Метод состоит из трех основных этапов.

1. На первом этапе для каждой входной точки $x_i \in R^M$, i=1..N определяются k ближайших соседей η_i и вычисляются m собственных векторов, соответствующих наибольшим собственным значениям матрицы

$$(\mathbf{X}_i - \bar{x}_i e^T)^T (\mathbf{X}_i - \bar{x}_i e^T),$$

а также множество

$$\mathbf{G}_i = \left[\frac{1}{\sqrt{k}} e, g_1, g_2, ... g_m \right]$$

Здесь $e = [1 \ \dots \ 1]^T$ – N-мерный вектор, все компоненты которого равны единице, $\mathbf{X}_i = [x_i^1, x_i^2, \dots x_i^k]$, $x_i^j \in \eta_i$ – матрица, составленная из ближайших соседей точки x_i, $\bar{x}_i = \mathbf{X}_i e / N$ - усредненное положение соседей точки x_i.

2. На втором этапе выполняется построение матрицы **B** для выравнивания касательных подпространств.
3. На третьем этапе выполняется выравнивание и формирование выходных векторов. Вычисляется (m+1) собственных вектора $u_2, u_3, \dots u_{m+1}$, соответствующих наименьшим собственным значениям матрицы **B** и формируется матрица $\mathbf{T}=[u_2, u_3, \dots u_{m+1}]^T$.

A.13. SNE

В методе стохастического вложения соседей (stochastic neighbor embedding, SNE) [109] для измерения рассогласования между точками используются условные вероятности:

$$p(j \mid i) = \frac{e^{-\frac{\|x_i - x_j\|^2}{2\sigma_i^2}}}{\sum_{k \neq i} e^{-\frac{\|x_i - x_k\|^2}{2\sigma_i^2}}}$$

Целевая функция строится с использованием дивергенции Кульбака-Лейблера

$$E = \sum_i KL(i \mid j) = \sum_i p_{j|i} \log_2 \frac{p_{j|i}}{q_{j|i}}$$

СКО распределений σ_i подбираются во входном многомерном пространстве индивидуально для каждой точки с использованием бинарного поиска по задаваемому параметру метода (т.н. перплексии) [109]. В выходном пространстве σ_i задаются константной величиной.

Оптимизация выполняется с использованием модификации градиентного спуска - метода моментов. Компоненты градиента имеют вид:

$$\frac{\partial E}{\partial y_k} = 2\sum_k (p_{k|i} - q_{k|i} + p_{i|k} - q_{i|k})(y_i - y_k)$$

Так как дивергенция Кульбака-Лейблера не является симметричной, то рассогласование близких точек по отношению к удаленным вносит иную составляющую ошибки, чем рассогласование удаленных точек по отношению к близким. Это обуславливает большую нацеленность на сохранение локальной структуры данных.

Позднее [50] было предложено симметричное стохастическое вложение соседей (англ. Symmetric Stochastic Neighbor Embedding, Symmetric SNE) с целевой функцией

$$E = \sum_i \sum_j p_{ji} \log_2 \frac{p_{ji}}{q_{ji}}$$

где $p_{ji} = p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$, а компоненты градиента задаются выражением:

$$\frac{\partial E}{\partial y_k} = 2\sum_k (p_{ik} - q_{ik})(y_i - y_k)$$

Позднее было предложено также стохастическое вложение соседей с использованием t-распределения Стьюдента с одной степенью свободы (вместо распределения Гаусса) в выходном пространстве низкой размерности [298]:

$$q_{ji} = \frac{\left(1 + \left\| y_i - y_j \right\|^2\right)^{-1}}{\sum_{k \neq i} \left(1 + \left\| y_i - y_k \right\|^2\right)^{-1}}$$

Соответствующие компоненты градиента имеют вид:

$$\frac{\partial E}{\partial y_k} = 4\sum_k (p_{ik} - q_{ik})\left(1 + \|y_i - y_k\|^2\right)^{-1}(y_i - y_k)$$

.

Последняя описанная модификация приобрела большую популярность при решении задачи визуализации данных.

A.14. Аппроксимация и проекция однородного многообразия

Метод аппроксимации и проекции однородного многообразия (англ., Uniform Manifold Approximation and Projection, UMAP) был введен L.McInnes, J.Healy и J. Melville и позже описан в статье [170].

Этот метод состоит в построении взвешенного графа с последующим размещением графа (силовой укладкой) в пространстве низкой размерности.

Основные этапы метода описаны ниже:

1. Для каждой точки x_i множества $X = \{x_i\}$, $i = 1..N$ точек данных в многомерном пространстве R^M найти множество η_i, состоящие из индексов k соседних точек.
2. Для каждой i-й точки данных найти ближайшего соседа и расстояние

$$\rho_i = \min\left(d(x_i, x_j) \mid x_j \in \eta_i, d(x_i, x_j) > 0\right),$$

а также значение σ_i такое что

$$\sum_{x_j \in \eta_i} \exp\left(\frac{-\max\left(0, d(x_i, x_j) - \rho_i\right)}{\sigma_i}\right) = \log_2 k\,.$$

3. Построить граф UMAP G как неориентированный взвешенный граф с матрицей смежности $\mathbf{B} = (b_{ij})$

$$\mathbf{B} = \mathbf{A} + \mathbf{A}^t - \mathbf{A} \circ \mathbf{A}^t,$$

где элементы $\mathbf{A}=(a_{ij})$ играют роль весов в соответствующем ориентированном графе:

$$a_{ij} = \exp\left(\frac{-\max\left(0, d(x_i, x_j) - \rho_i\right)}{\sigma_i}\right).$$

4. Координаты y_i, $i = 1..N$ точек данных в пространстве низкой размерности R^m определяются силовым размещением графа с использованием следующих сил притяжения F^a и отталкивания F^r между вершинами i и j (α, β, ε - постоянные параметры):

$$F_{i,j}^{a} = \frac{-2\alpha\beta\left\|y_i - y_j\right\|_2^{2(\beta-1)}}{1+\left\|y_i - y_j\right\|_2^{2}} b_{ij}\left(y_i - y_j\right),$$

$$F_{i,j}^{r} = \frac{\beta}{\left(\varepsilon+\left\|y_i - y_j\right\|_2^{2}\right)\left(1+\left\|y_i - y_j\right\|_2^{2}\right)}\left(1-b_{ij}\right)\left(y_i - y_j\right).$$

A.15. Автоассоциативные нейронные сети

Автоассоциативная нейронная сеть [13, 146] (англ., autoassociative neural network, AANN), часто называемая автоэнкодером (англ., autoencoder), представляет собой специальную искусственную нейронную сеть, состоящую из двух последовательных частей, называемых кодером и декодером.

Кодер обычно состоит из двух (иногда более двух) последовательных полностью связанных слоев. Первый уровень кодера принимает многомерный вектор $x \in R^M$ в качестве входного, а последний (выходной) слой создает вектора $y \in R^m$ меньшей размерности ($m<M$). Таким образом, кодер выполняет преобразование входных многомерных векторов $x \in R^M$ в их представление $y \in R^m$ в низкой размерности.

Декодер выполняет обратное отображение из R^m в R^M и обычно (хотя и необязательно) содержит такое же количество полностью связанных слоев и нейронов, что и кодер. Декодер принимает вектора $y \in R^m$ низкой размерности, сформированные кодером, и выдает аппроксимацию $\tilde{x} \in R^M$ исходного многомерного сигнала в качестве своего выхода.

Поскольку число нейронов, соответствующих среднему слою всей AANN, меньше, чем его входная и выходная размерность, этот уровень часто называют бутылочным горлышком (bottleneck), а всю архитектуру – bottleneck-архитектурой (см. рисунок А.2).

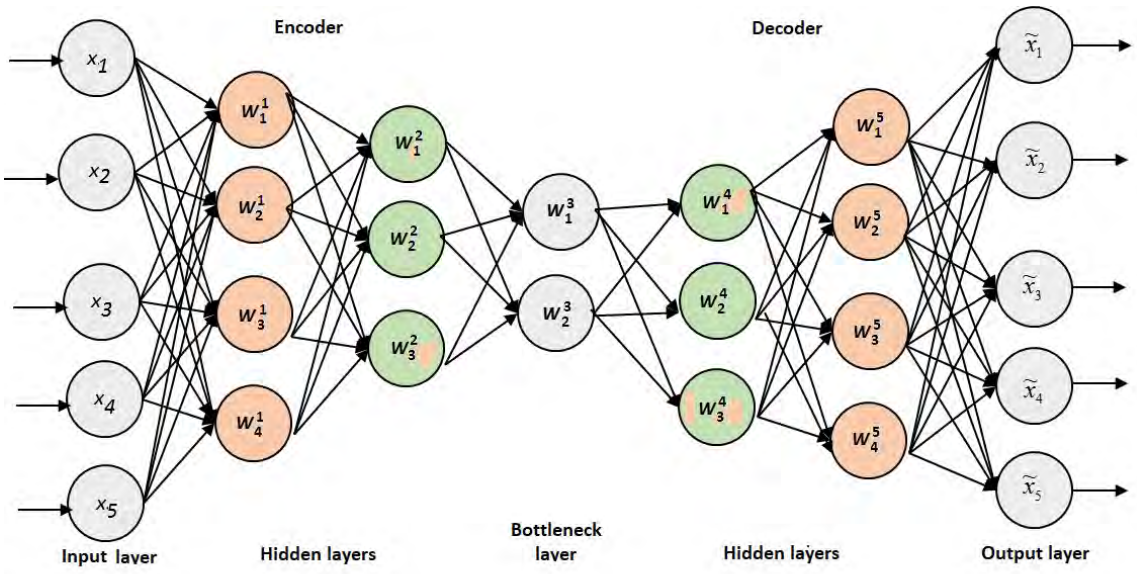


Рисунок А.2 - Архитектура сети AANN

Обучение AANN осуществляется в режиме самообучения, поскольку желаемым выходом сети фактически являются сами входные векторы. Функция ошибки (стоимости) - ошибка восстановления, выраженная следующим образом:

$$E = \frac{1}{N}\sum_{i=1}^{N}\left\|x_i - \tilde{x}_i\right\|^2$$

Для обучения сети обычно используется один из методов обратного распространения ошибки.

После обучения кодер и декодер могут использоваться независимо для выполнения снижения размерности входных данных (прямое отображение) и восстановления исходного сигнала по его редуцированному представлению (обратное отображение).

Линейный и нелинейный анализ главных компонент

Пусть $\mathbf{W}^i = \{w^i{}_1; w^i{}_2; \ldots\}$ - весовая матрица нейронов в i-м слое кодера, где w^i_j - вектор весов для j-го нейрона i-го слоя, а $b^i = \{b^1{}_1;\ b^1{}_2,\ \ldots\}$ - вектор смещения нейронов. Тогда для некоторого входного вектора $x \in R^M$ i-й слой кодера выдает

$$x^{\mathrm{i}} = f^{\mathrm{i}}(\mathbf{W}^{\mathrm{i}} x^{\mathrm{i-1}} + b^{\mathrm{i}});$$

где $f^{\,i}$ - функция активации нейронов в i-м слое кодера, а $x^0 = x$. Функция активации в скрытых слоях обычно представляет собой некоторую поэлементную нелинейную функцию, например, сигмоидальную или Rectified Linear Unit (ReLU).

Однако, если во всех слоях используется линейная функция активации (так называемый «линейный автоэнкодер»), отображение становится аналогичным PCA [236], поскольку используется линейное преобразование, а критерий минимизации аналогичен таковому в PCA. В [31, 12] было показано, что искомое решение для линейного автоэнкодера разделяет подпространство главных компонент, но отличается от PCA, поскольку найденное решение необязательно удовлетворяет требованию ортонормированности. Тем не менее, поскольку найденное решение разделяет то же подпространство, что и первые m главных компонент, решение остается оптимальным среди всех возможных линейных проекций входных данных в m-мерное пространство.

Как отмечалось в [146], использование нелинейных функций активации позволяет аппроксимировать любую нелинейную функцию $y=g(x)$ с произвольной точностью. В соответствии с [58], двух слоев в кодере достаточно для достижения такого приближения, если в первом слое используется любая непрерывная и монотонно возрастающая функция $f^1()$, а второй слой вычисляет взвешенные суммы с линейной активацией $f^2(x) = x$. Таким образом, число нейронов в первом слое влияет на качество отображения, а количество нейронов во втором слое определяет размерность m редуцированного пространства. Поскольку отображение, заданное кодером, больше не является линейным, соответствующий метод называют нелинейным анализом главных компонент [146].

Предварительное обучение AANN с использованием PCA

Тот факт, что оптимальное решение, получаемое с использованием линейного автоэнкодера, разделяет подпространство главных компонент, приводит к идее, что PCA можно использовать для инициализации линейного автоэнкодера [185*]. Кроме того, мы предполагаем, что AANN с нелинейными функциями активации также может быть инициализирован PCA с использованием предварительного обучения результатами PCA. Мы полагаем, что этот метод инициализации имеет смысл в силу следующих соображений.

Во-первых, влияние нелинейных эффектов не всегда является определяющим при формировании исходных данных. В таких случаях инициализация методов нелинейного снижения размерности с использованием PCA может дать хорошие результаты, поскольку глобальная структура данных уже хорошо отражена во время

инициализации. Например, для гиперспектральных изображений, несмотря на наличие нелинейных моделей, наиболее распространенной моделью является модель линейного смешивания спектральных сигнатур, которая дает вполне удовлетворительные результаты при анализе.

Во-вторых, раздельное предварительное обучение кодера и декодера может быть более эффективным, чем обучение всей AANN, поскольку в процессе обучения задействовано в два раза меньше нейронных слоев.

В-третьих, инициализация нейронной сети почти оптимальными (по крайней мере, среди возможных линейных преобразований) результатами дает больше шансов получить глобально оптимальное решение, чем обучение со случайной инициализацией.

Процесс обучения, предложенный в [185*], состоит из следующих шагов:

1. Выполнить PCA и найти проекцию $Y^{PCA}= \{y_i^{PCA}\}_{i=1..N}$ исходных данных $X= \{x_i\}_{i=1..N}$ на первые m главных компонент. В случае большого набора данных для оценки линейного преобразования можно использовать случайную подвыборку подходящего размера.

2. Предобучить кодировщик AANN, используя X в качестве входа кодировщика и Y^{PCA} в качестве желаемого выхода на протяжении предварительно определенного числа K_{pre} эпох обучения.

3. Предобучить декодер с использованием Y^{PCA} в качестве входного сигнала декодера и X в качестве желаемого выходного сигнала на протяжении предварительно определенного числа K_{pre} эпох. В качестве альтернативы, можно предобучить декодер, встроенный в AANN, зафиксировав веса кодера и используя его выход $\tilde{Y}^{PCA}$ в качестве входа декодера.

4. Обучить модель AANN с предварительно обученным кодером и декодером.

В работе [191*] автором было показано, что на этапе предварительного обучения могут быть использованы не только результаты анализа главных компонент, но и результаты метода нелинейного отображения (А.4).

А.16. Эластичные главные графы

Подход к снижению размерности, основанный на эластичных главных графах (англ., Principal Elastic Graphs) был предложен А.Н.Горбанем, А.Ю.Зиновьевым и др. [90].

В этом подходе рассматривается неориентированный граф $G=<Y, E>$, где Y – множество вершин, а E – множество ребер. Вводится понятие k-звезды (star), представляющей собой подграф из k+1 вершины $y_0, y_1,.. y_{k+1}$ и k ребер (y_0, y_i), i=1..k, в котором вершина y_0 соединена с остальными вершинами подграфа.

Граф с определенными на нем семействами k-звезд S_k называется эластичным, если для всех подмножеств его ребер $E^{(i)} \in E$ и семейств $S^{(k)} \in S_k$ определены соответствующие модули эластичности $\lambda_i > 0$ и $\mu_{kj} > 0$.

Пусть $E^{(i)}(0)$ и $E^{(i)}(1)$ – вершины ребра $E^{(i)}$, а $S_k^{(j)}(0)$, … $S_k^{(j)}(\mathrm{k})$ – вершины k-звезды $S_k^{(j)}$ Для любого отображения ϕ: $Y \to R^M$ энергия графа определяется следующим образом:

$$U^{\phi}(G) = \sum_{E^{(i)}} \lambda_i \left\| \phi\left(E^{(i)}(0)\right) - \phi\left(E^{(i)}(1)\right) \right\|^2 + \sum_{S_k^{(j)}} \mu_{kj} \left\| \sum_{i=1}^{k} \phi\left(S_k^{(j)}(0)\right) - k\phi\left(S_k^{(j)}(0)\right) \right\|^2$$

Было показано [95], что каждая система эластичных конечных элементов может быть представлена системой пружин с, возможно, отрицательными коэффициентами эластичности.

Энергия k-звезды s_k в R^m с центром y_0 и k конечными точками $y_1,\ldots,y_k$ может быть рассчитана следующим образом:

$$u_{s_k} = \mu_{s_k} \left(\sum_{i=1}^{k} y_i - k y_0 \right)^2 = k\mu_{s_k} \sum_{i=1}^{k} \left(y_i - k y_0 \right)^2 - \mu_{s_k} \sum_{i=j+1}^{k} \left(y_i - y_j \right)^2 .$$

Набор данных D разбивается на подмножества K^y, $y \in Y$, содержащие те точки, для которых $\phi(y)$ является ближайшим:

$$K^{y_j} = \left\{ x_i \middle| y_j = \arg\min_{y_k \in Y} \left\| y_k - x_i \right\| \right\}.$$

Энергия аппроксимации записывается в виде:

$$U_A^{\phi}(G, D) = \frac{1}{\sum_{x \in D} w(x)} \sum_{y \in Y} \sum_{x \in K^y} w(x) \left\| x - \phi(y) \right\|^2 ,$$

где $w(x)$ – неотрицательные веса, а $1/\Sigma w(x)$ – нормализующий множитель.

Для нахождения решения задачи $U^{\phi} \to \min$, где

$$U^{\phi} = U_A^{\phi}(G, D) + U^{\phi}(G),$$

используется алгоритм, схожий с алгоритмом k-внутригрупповых средних:

1. Выполняется минимизация положительной квадратичной функции U^{ϕ} путем решения системы линейных алгебраических уравнений для нахождения положения вершин $\{\phi(y_i)\}$

$$\sum_{k=1}^{p} a_{jk}\phi(y_k) = \frac{1}{\sum_{x\in D} w(x)} \sum_{x\in K^{y_j}} w(x)x$$

2. Для найденных ϕ обновляются $\{K^y\}$
3. Если ϕ не изменились, то выход, иначе – возврат к шагу 1.

$$a_{jk} = \frac{n_j \delta_{jk}}{\sum_{x\in D} w(x)} + e_{jk} + s_{jk}, \quad n_j = \sum_{x\in K^{y_j}} w(x), \quad j = 1,..p$$

Здесь δ_{jk} – дельта-функция Кронекера, матрицы e_{jk} и s_{jk} – зависят только от модулей эластичности и содержимого множеств $\{E^{(i)}\}$ и $\{S_k^{(i)}\}$ и могут не пересчитываться, если структура графа не меняется.

Приложение Б. Частные производные функционалов ошибки, построенных с использованием различных функций расстояния

Б.1. Норма Lp

Функционал ошибки:

$$\varepsilon = \mu \cdot \sum_{\substack{i,j=1 \\ i<j}}^{N} \rho_{ij} \left(\left\| x_i - x_j \right\|_p - \left\| y_i - y_j \right\|_p \right)^2 .$$

где $\left\| x_i - x_j \right\|_p = \left(\sum_{k=1}^{M} \left| x_i - x_j \right|^p \right)^{1/p}$.

Решение:

Пусть $y_{ik} \neq y_{jk}$, j=1..N, k=1..M, тогда

$$\frac{\partial \varepsilon}{\partial y_{ik}} = \mu \cdot \sum_{\substack{i,j=1 \\ i<j}}^{N} \rho_{ij} \cdot 2 \cdot \left(\left\| x_i - x_j \right\|_p - \left\| y_i - y_j \right\|_p \right) \cdot (-1) \cdot \frac{\partial \left(\left\| y_i - y_j \right\|_p \right)}{\partial y_{ik}} =$$

$$= -2\mu \cdot \sum_{\substack{j=1, \\ j \neq i}}^{N} \rho_{ij} \cdot \left(\left\| x_i - x_j \right\|_p - \left\| y_i - y_j \right\|_p \right) \cdot \frac{\partial \left(\left\| y_i - y_j \right\|_p \right)}{\partial y_{ik}}.$$

$$\frac{\partial \left(\left\| y_i - y_j \right\|_p \right)}{\partial y_{ik}} = \frac{\partial \left(\sum_k \left| y_{ik} - y_{jk} \right|^p \right)^{1/p}}{\partial y_{ik}} = \frac{1}{p} \cdot \left(\sum_k \left| y_{ik} - y_{jk} \right|^p \right)^{\frac{1-p}{p}} \cdot \frac{\partial \left(\sum_k \left| y_{ik} - y_{jk} \right|^p \right)}{\partial y_{ik}} =$$

$$= \frac{1}{p} \cdot \left(\sum_k \left| y_{ik} - y_{jk} \right|^p \right)^{\frac{1-p}{p}} \cdot \sum_k \frac{\partial \left| y_{ik} - y_{jk} \right|^p}{\partial y_{ik}}$$

$$= \frac{1}{p} \cdot \left(\sum_k \left| y_{ik} - y_{jk} \right|^p \right)^{\frac{1-p}{p}} \cdot p \cdot \left| y_{ik} - y_{jk} \right|^{(p-1)} \frac{\partial \left| y_{ik} - y_{jk} \right|}{\partial y_{ik}}$$

$$= \left(\sum_k \left| y_{ik} - y_{jk} \right|^p \right)^{\frac{1-p}{p}} \cdot \left| y_{ik} - y_{jk} \right|^{(p-1)} sign\left(y_{ik} - y_{jk} \right).$$

$$\frac{\partial \varepsilon}{\partial y_{ik}} = -2\mu \cdot \sum_{\substack{j=1, \\ j \neq i}}^{N} \rho_{ij} \cdot \left(\left\| x_i - x_j \right\|_p - \left\| y_i - y_j \right\|_p \right) \cdot \left(\sum_k \left| y_{ik} - y_{jk} \right|^p \right)^{\frac{1-p}{p}} \cdot \left| y_{ik} - y_{jk} \right|^{p-1} sign\left(y_{ik} - y_{jk} \right).$$

или

$$\frac{\partial \varepsilon}{\partial y_{ik}} = -2\mu \cdot \sum_{j=1\,(j\neq i)}^{N} \rho_{ij} \cdot \frac{\left\|x_i - x_j\right\|_p - \left\|y_i - y_j\right\|_p}{\left\|y_i - y_j\right\|_3^{p-1}} \cdot \left|y_{ik} - y_{jk}\right|^{p-1} \operatorname{sgn}\left(y_{ik} - y_{jk}\right).$$

При p=1:

$$\frac{\partial \varepsilon}{\partial y_{ik}} = -2\mu \cdot \sum_{j=1\,(j\neq i)}^{N} \rho_{ij} \cdot \left(\left\|x_i - x_j\right\|_p - \left\|y_i - y_j\right\|_p\right) \cdot \operatorname{sgn}\left(y_{ik} - y_{jk}\right).$$

Б.2. Спектральный угол (SAM)

Функция расстояния SAM:

$$SAM\left(x_i, x_j\right) = \arccos\left(\frac{\sum_{k=1}^{M} x_{ik} x_{jk}}{\sqrt{\sum_{k=1}^{M} x_{ik}^2}\sqrt{\sum_{k=1}^{M} x_{jk}^2}}\right).$$

Функционал ошибки:

$$\varepsilon_{SAM_C} = \mu \sum_{i,j=1(i<j)}^{N} \left(\rho_{i,j}\left(\theta(x_i, x_j) - \theta(y_i, y_j)\right)^2\right)$$

Решение:

$$\frac{\partial}{\partial y_{in}}\varepsilon = \frac{\partial}{\partial y_{in}}\mu \sum_{i,j=1\,(i<j)}^{N} \rho_{ij}\left(\theta(x_i, x_j) - \theta(y_{i,} y_j)\right)^2 = \mu \sum_{i,j=1\,(i<j)}^{N} \rho_{ij} \frac{\partial}{\partial y_{in}}\left(\theta(x_i, x_j) - \theta(y_{i,} y_j)\right)^2$$

$$= \mu \sum_{j=1\,(j\neq i)}^{N} \rho_{ij} \frac{\partial}{\partial y_{in}}\left(\theta(x_i, x_j) - \theta(y_{i,} y_j)\right)^2$$

$$= -2\mu \sum_{j=1\,(j\neq i)}^{N} \rho_{ij}\left(\theta(x_i, x_j) - \theta(y_{i,} y_j)\right)\frac{\partial}{\partial y_{in}}\theta(y_{i,} y_j)$$

Частная производная

$$\frac{\partial}{\partial y_{in}}\theta\left(y_i, y_j\right) = \frac{\partial}{\partial y_{in}}\arccos\left(\frac{\sum_{k=1}^{M} x_{ik} x_{jk}}{\sqrt{\sum_{k=1}^{M} x_{ik}^2}\sqrt{\sum_{k=1}^{M} x_{jk}^2}}\right)$$

$$= -\frac{1}{\sqrt{1-\frac{\left(\sum_{k=1}^{M} x_{ik}x_{jk}\right)^2}{\sum_{k=1}^{M} x_{ik}^2 \sum_{k=1}^{M} x_{jk}^2}}} \frac{\partial}{\partial y_{in}} \frac{\sum_{k=1}^{M} x_{ik}x_{jk}}{\sqrt{\sum_{k=1}^{M} x_{ik}^2}\sqrt{\sum_{k=1}^{M} x_{jk}^2}}$$

$$= -\sqrt{\frac{\sum_{k=1}^{M} x_{ik}^2 \sum_{k=1}^{M} x_{jk}^2}{\sum_{k=1}^{M} x_{ik}^2 \sum_{k=1}^{M} x_{jk}^2 - \left(\sum_{k=1}^{M} x_{ik}x_{jk}\right)^2}} \frac{1}{\sqrt{\sum_{k=1}^{M} x_{jk}^2}} \frac{\partial}{\partial y_{in}} \frac{\sum_{k=1}^{M} x_{ik}x_{jk}}{\sqrt{\sum_{k=1}^{M} x_{ik}^2}}$$

$$= -\sqrt{\frac{\sum_{k=1}^{M} y_{ik}^2}{\sum_{k=1}^{M} y_{ik}^2 \sum_{k=1}^{M} y_{jk}^2 - \left(\sum_{k=1}^{M} y_{ik}y_{jk}\right)^2}} \frac{y_{jk}\sqrt{\sum_{k=1}^{M} y_{ik}^2} - \frac{1}{2}\left(\sum_{k=1}^{M} y_{ik}^2\right)^{-1/2} 2y_{in}\sum_{k=1}^{M} y_{ik}y_{jk}}{\sum_{k=1}^{M} y_{ik}^2}$$

$$= -\sqrt{\frac{\sum_{k=1}^{M} y_{ik}^2}{\sum_{k=1}^{M} y_{ik}^2 \sum_{k=1}^{M} y_{jk}^2 - \left(\sum_{k=1}^{M} y_{ik}y_{jk}\right)^2}} \frac{y_{jk}\sum_{k=1}^{M} y_{ik}^2 - y_{in}\sum_{k=1}^{M} y_{ik}y_{jk}}{\sqrt{\sum_{k=1}^{M} y_{ik}^2}\sum_{k=1}^{M} y_{ik}^2}$$

$$= -\frac{y_{jn}\sum_{k=1}^{M} y_{ik}^2 - y_{in}\sum_{k=1}^{M} y_{ik}y_{jk}}{\sqrt{\sum_{k=1}^{M} y_{ik}^2 \sum_{k=1}^{M} y_{jk}^2 - \left(\sum_{k=1}^{M} y_{ik}y_{jk}\right)^2}\sum_{k=1}^{M} y_{ik}^2}$$

или

$$\frac{\partial}{\partial y_{in}}\theta\left(y_i, y_j\right) = -\frac{y_{jn} - y_{in}\left\langle y_i, y_j\right\rangle / \left\|y_i\right\|^2}{\sqrt{\left\|y_i\right\|^2\left\|y_j\right\|^2 - \left\langle y_i, y_j\right\rangle^2}}.$$

Окончательное решение:

$$\frac{\partial}{\partial y_{in}}\varepsilon = 2\mu \sum_{j=1\,(j\neq i)}^{N} \rho_{ij}\left(\theta(x_i, x_j) - \theta(y_i, y_j)\right)\frac{y_{jn} - y_{in}\left\langle y_i, y_j\right\rangle / \left\|y_i\right\|^2}{\sqrt{\left\|y_i\right\|^2\left\|y_j\right\|^2 - \left\langle y_i, y_j\right\rangle^2}}.$$

Б.3. Модифицированная мера рассогласования углов

Модифицированная функция расстояния:

$$\theta^*\left(\phi_i,\phi_j\right)=\sum_{k=1}^{m}\sin^2\frac{\phi_{ik}-\phi_{jk}}{2}.$$

Функционал ошибки:

$$\varepsilon_{\theta^*}=\mu\sum_{i,j=1\,(i<j)}^{N}\rho_{ij}\left(\theta^*\left(\psi_i,\psi_j\right)-\theta^*\left(\phi_i,\phi_j\right)\right)^2.$$

Решение:

$$\frac{\partial}{\partial\phi_{in}}\varepsilon_{\theta^*}=\frac{\partial}{\partial\phi_{in}}\mu\sum_{i,j=1\,(i<j)}^{N}\rho_{ij}\left(\theta^*(\psi_{i,}\psi_j)-\theta^*(\phi_i,\phi_j)\right)^2$$

$$=\mu\sum_{i,j=1\,(i<j)}^{N}\rho_{ij}\frac{\partial}{\partial\phi_{in}}\left(\theta^*(\psi_{i,}\psi_j)-\theta^*(\phi_i,\phi_j)\right)^2=\mu\sum_{j=1\,(j\neq i)}^{N}\rho_{ij}\frac{\partial}{\partial\phi_{in}}\left(\theta^*(\psi_{i,}\psi_j)-\theta^*(\phi_i,\phi_j)\right)^2$$

$$=-2\mu\sum_{j=1\,(j\neq i)}^{N}\rho_{ij}\left(\theta^*(\psi_{i,}\psi_j)-\theta^*(\phi_i,\phi_j)\right)\frac{\partial}{\partial\phi_{in}}\theta^*(\phi_i,\phi_j).$$

Частная производная

$$\frac{\partial}{\partial\phi_{in}}\theta^*(\phi_i,\phi_j)=\frac{\partial}{\partial\phi_{in}}\sum_{k=1}^{m}\sin^2\frac{\phi_{ik}-\phi_{jk}}{2}=\sum_{k=1}^{m}\frac{\partial}{\partial\phi_{in}}\sin^2\frac{\phi_{ik}-\phi_{jk}}{2}=\frac{\partial}{\partial\phi_{in}}\sin^2\frac{\phi_{in}-\phi_{jn}}{2}$$

$$=2\sin\frac{\phi_{in}-\phi_{jn}}{2}\frac{\partial}{\partial\phi_{in}}\sin\frac{\phi_{in}-\phi_{jn}}{2}=\sin\frac{\phi_{in}-\phi_{jn}}{2}\cos\frac{\phi_{in}-\phi_{jn}}{2}.$$

Окончательное решение:

$$\frac{\partial}{\partial\varphi_{in}}\varepsilon=-2\mu\sum_{j=1\,(j\neq i)}^{N}\rho_{ij}\left(\theta^*(\psi_{i,}\psi_j)-\theta^*(\phi_i,\phi_j)\right)\sin\frac{\phi_{in}-\phi_{jn}}{2}\cos\frac{\phi_{in}-\phi_{jn}}{2}$$

$$=-\mu\sum_{j=1\,(j\neq i)}^{N}\rho_{ij}\left(\theta^*(\psi_{i,}\psi_j)-\theta^*(\phi_i,\phi_j)\right)\sin\left(\phi_{in}-\phi_{jn}\right).$$

Б.4. Спектральная корреляция

Мера сходства:

$$r(x_i,x_j)=\frac{M\sum_{k=1}^{M}x_{ik}x_{jk}-\sum_{k=1}^{M}x_{ik}\sum_{k=1}^{M}x_{jk}}{\sqrt{M\sum_{k=1}^{M}x_{ik}^2-\left(\sum_{k=1}^{M}x_{ik}\right)^2}\sqrt{M\sum_{k=1}^{M}x_{jk}^2-\left(\sum_{k=1}^{M}x_{jk}\right)^2}}.$$

Функционал ошибки:

$$\varepsilon = \mu \sum_{i,j=1\,(i<j)}^{N} \left(r(x_i, x_j) - r(y_{i,} y_j)\right)^2 .$$

Решение:

$$\frac{\partial}{\partial y_{in}} \varepsilon = \frac{\partial}{\partial y_{in}} \mu \sum_{i,j=1\,(i<j)}^{N} \left(r(x_i, x_j) - r(y_{i,} y_j)\right)^2 = \mu \sum_{i,j=1\,(i<j)}^{N} \frac{\partial}{\partial y_{in}} \left(r(x_i, x_j) - r(y_{i,} y_j)\right)^2$$

$$= \mu \sum_{j=1\,(i\neq j)}^{N} \frac{\partial}{\partial y_{in}} \left(r(x_i, x_j) - r(y_{i,} y_j)\right)^2 = -2\mu \sum_{j=1\,(i\neq j)}^{N} \left(r(x_i, x_j) - r(y_{i,} y_j)\right) \frac{\partial}{\partial y_{in}} r(y_{i,} y_j) .$$

Частная производная меры:

$$\frac{\partial}{\partial y_{in}} r(y_{i,} y_j) = \frac{\partial}{\partial y_{in}} \frac{M \sum_{k=1}^{M} y_{ik} y_{jk} - \sum_{k=1}^{M} y_{ik} \sum_{k=1}^{M} y_{jk}}{\sqrt{M \sum_{k=1}^{M} y_{ik}^2 - \left(\sum_{k=1}^{M} y_{ik}\right)^2} \sqrt{M \sum_{k=1}^{M} y_{jk}^2 - \left(\sum_{k=1}^{M} y_{jk}\right)^2}} .$$

Частная производная числителя:

$$\frac{\partial}{\partial y_{in}} \left(M \sum_{k=1}^{M} y_{ik} x_{jk} - \sum_{k=1}^{M} y_{ik} \sum_{k=1}^{M} y_{jk} \right) = M \sum_{k=1}^{M} y_{jk} \frac{\partial}{\partial y_{in}} y_{ik} - \sum_{k=1}^{M} y_{jk} \sum_{k=1}^{M} \frac{\partial}{\partial y_{in}} y_{ik} = M y_{jn} - \sum_{k=1}^{M} y_{jk} .$$

Частная производная знаменателя:

$$\frac{\partial}{\partial y_{in}} \sqrt{M \sum_{k=1}^{M} y_{ik}^2 - \left(\sum_{k=1}^{M} y_{ik}\right)^2} = \frac{\partial}{\partial y_{in}} \sqrt{M \sum_{k=1}^{M} y_{ik}^2 - \left(\sum_{k=1}^{M} y_{ik}\right)^2} = \frac{1}{2} \frac{\frac{\partial}{\partial y_{in}} \left(M \sum_{k=1}^{M} y_{ik}^2 - \left(\sum_{k=1}^{M} y_{ik}\right)^2 \right)}{\sqrt{M \sum_{k=1}^{M} y_{ik}^2 - \left(\sum_{k=1}^{M} y_{ik}\right)^2}}$$

$$= \frac{1}{2} \frac{M \sum_{k=1}^{M} \frac{\partial}{\partial y_{in}} y_{ik}^2 - 2 \left(\sum_{k=1}^{M} y_{ik}\right) \frac{\partial}{\partial y_{in}} \sum_{k=1}^{M} y_{ik}}{\sqrt{M \sum_{k=1}^{M} y_{ik}^2 - \left(\sum_{k=1}^{M} y_{ik}\right)^2}} = \frac{1}{2} \frac{2 M y_{in} - 2 \sum_{k=1}^{M} y_{ik}}{\sqrt{M \sum_{k=1}^{M} y_{ik}^2 - \left(\sum_{k=1}^{M} y_{ik}\right)^2}} = \frac{M y_{in} - \sum_{k=1}^{M} y_{ik}}{\sqrt{M \sum_{k=1}^{M} y_{ik}^2 - \left(\sum_{k=1}^{M} y_{ik}\right)^2}} .$$

Подставляем в частную производную меры:

$$\frac{\partial}{\partial y_{in}} r(y_{i,} y_j) = \frac{\sqrt{M \sum_{k=1}^{M} y_{ik}^2 - \left(\sum_{k=1}^{M} y_{ik}\right)^2} \frac{\partial}{\partial y_{in}} \left(M \sum_{k=1}^{M} y_{ik} y_{jk} - \sum_{k=1}^{M} y_{ik} \sum_{k=1}^{M} y_{jk} \right)}{\sqrt{M \sum_{k=1}^{M} y_{jk}^2 - \left(\sum_{k=1}^{M} y_{jk}\right)^2} \left(M \sum_{k=1}^{M} y_{ik}^2 - \left(\sum_{k=1}^{M} y_{ik}\right)^2 \right)} -$$

$$-\frac{\left(M\sum_{k=1}^{M}y_{ik}y_{jk}-\sum_{k=1}^{M}y_{ik}\sum_{k=1}^{M}y_{jk}\right)\frac{\partial}{\partial y_{in}}\sqrt{M\sum_{k=1}^{M}y_{ik}^2-\left(\sum_{k=1}^{M}y_{ik}\right)^2}}{\sqrt{M\sum_{k=1}^{M}y_{jk}^2-\left(\sum_{k=1}^{M}y_{jk}\right)^2}\left(M\sum_{k=1}^{M}y_{ik}^2-\left(\sum_{k=1}^{M}y_{ik}\right)^2\right)}=$$

$$=\frac{\sqrt{M\sum_{k=1}^{M}y_{ik}^2-\left(\sum_{k=1}^{M}y_{ik}\right)^2}\left(M\ y_{jn}-\sum_{k=1}^{M}y_{jk}\right)-\left(M\sum_{k=1}^{M}y_{ik}y_{jk}-\sum_{k=1}^{M}y_{ik}\sum_{k=1}^{M}y_{jk}\right)\frac{My_{in}-\sum_{k=1}^{M}y_{ik}}{\sqrt{M\sum_{k=1}^{M}y_{ik}^2-\left(\sum_{k=1}^{M}y_{ik}\right)^2}}}{\sqrt{M\sum_{k=1}^{M}y_{jk}^2-\left(\sum_{k=1}^{M}y_{jk}\right)^2}\left(M\sum_{k=1}^{M}y_{ik}^2-\left(\sum_{k=1}^{M}y_{ik}\right)^2\right)}$$

$$=\frac{\sqrt{M\sum_{k=1}^{M}y_{ik}^2-\left(\sum_{k=1}^{M}y_{ik}\right)^2}\left(My_{jn}-\sum_{k=1}^{M}y_{jk}\right)-\left(M\sum_{k=1}^{M}y_{ik}y_{jk}-\sum_{k=1}^{M}y_{ik}\sum_{k=1}^{M}y_{jk}\right)\frac{My_{in}-\sum_{k=1}^{M}y_{ik}}{\sqrt{M\sum_{k=1}^{M}y_{ik}^2-\left(\sum_{k=1}^{M}y_{ik}\right)^2}}}{\sqrt{M\sum_{k=1}^{M}y_{jk}^2-\left(\sum_{k=1}^{M}y_{jk}\right)^2}\left(M\sum_{k=1}^{M}y_{ik}^2-\left(\sum_{k=1}^{M}y_{ik}\right)^2\right)}$$

$$=\frac{\left(My_{jn}-\sum_{k=1}^{M}y_{jk}\right)-\left(M\sum_{k=1}^{M}y_{ik}y_{jk}-\sum_{k=1}^{M}y_{ik}\sum_{k=1}^{M}y_{jk}\right)\frac{My_{in}-\sum_{k=1}^{M}y_{ik}}{M\sum_{k=1}^{M}y_{ik}^2-\left(\sum_{k=1}^{M}y_{ik}\right)^2}}{\sqrt{M\sum_{k=1}^{M}y_{jk}^2-\left(\sum_{k=1}^{M}y_{jk}\right)^2}\sqrt{M\sum_{k=1}^{M}y_{ik}^2-\left(\sum_{k=1}^{M}y_{ik}\right)^2}},$$

или

$$\frac{\partial}{\partial y_{in}}r(y_i,y_j)=\frac{1}{g_{ij}}\left(M\ y_{jn}-\sum_{k=1}^{M}y_{jk}-f_{ij}\left(My_{in}-\sum_{k=1}^{M}y_{ik}\right)\right),$$

$$g_{ij}=\sqrt{M\sum_{k=1}^{M}y_{jk}^2-\left(\sum_{k=1}^{M}y_{jk}\right)^2}\sqrt{M\sum_{k=1}^{M}y_{ik}^2-\left(\sum_{k=1}^{M}y_{ik}\right)^2},$$

$$f_{ij}=\frac{M\sum_{k=1}^{M}y_{ik}y_{jk}-\sum_{k=1}^{M}y_{ik}\sum_{k=1}^{M}y_{jk}}{M\sum_{k=1}^{M}y_{ik}^2-\left(\sum_{k=1}^{M}y_{ik}\right)^2}.$$

Б.5. Дивергенция спектральной информации (SID)

Функция расстояния SID:

$$SID(x_i,x_j)=D(x_i\|x_j)+D(x_j\|x_i),$$

$$D\left(x_i \middle\| x_j\right) = \sum_{k=1}^{M} p_k(x_i) \log\left(p_k(x_i) / p_k(x_j)\right),$$

$$p_k(x_i) = \frac{x_{ik}}{\sum_{l=1}^{M} x_{il}}.$$

Функционал ошибки:

$$\varepsilon = \mu \sum_{i,j=1\,(i<j)}^{N} \left(SID(x_i, x_j) - SID(y_{i,} y_j)\right)^2.$$

Решение:

$$\frac{\partial}{\partial y_{in}} \varepsilon = \frac{\partial}{\partial y_{in}} \mu \sum_{i,j=1\,(i<j)}^{N} \left(SID(x_i, x_j) - SID(y_{i,} y_j)\right)^2 = \mu \sum_{i,j=1\,(i<j)}^{N} \frac{\partial}{\partial y_{in}} \left(SID(x_i, x_j) - SID(y_{i,} y_j)\right)^2$$

$$= \mu \sum_{j=1\,(i\neq j)}^{N} \frac{\partial}{\partial y_{in}} \left(SID(x_i, x_j) - SID(y_{i,} y_j)\right)^2$$

$$= -2\mu \sum_{j=1\,(i\neq j)}^{N} \left(SID(x_i, x_j) - SID(y_{i,} y_j)\right) \frac{\partial}{\partial y_{in}} SID(y_{i,} y_j)$$

Вывод 1:

$$\frac{\partial}{\partial y_{in}} SID\left(y_i, y_j\right) = \frac{\partial}{\partial y_{in}} D\left(y_i \middle\| y_j\right) + \frac{\partial}{\partial y_{in}} D\left(y_j \middle\| y_i\right)$$

Вывод 1.1:

$$\frac{\partial}{\partial y_{in}} D\left(y_i \middle\| y_j\right) = \frac{\partial}{\partial y_{in}} \sum_{k=1}^{M} p_k(y_i) \log\left(p_k(y_i) / p_k(y_j)\right) = \sum_{k=1}^{M} \frac{\partial}{\partial y_{in}} p_k(y_i) \log\left(p_k(y_i) / p_k(y_j)\right)$$

$$= \sum_{k=1}^{M} \left(\log\left(p_k(y_i) / p_k(y_j)\right) \frac{\partial}{\partial y_{in}} p_k(y_i) + p_k(y_i) \frac{\partial}{\partial y_{in}} \log\left(p_k(y_i) / p_k(y_j)\right) \right)$$

Вывод 1.1.1

$$for\ k = n: \quad \frac{\partial}{\partial y_{in}} p_k(y_i) = \frac{\partial}{\partial y_{in}} p_n(y_i) = \frac{\partial}{\partial x_{in}} \frac{x_{in}}{\sum_{l=1}^{M} x_{il}} = \frac{1}{\sum_{l=1}^{M} x_{il}} - \frac{x_{in}}{\left(\sum_{l=1}^{M} x_{il}\right)^2} = \frac{1}{\sum_{l=1}^{M} x_{il}} - \frac{p_n(y_i)}{\sum_{l=1}^{M} x_{il}},$$

$$for\ k \neq n: \quad \frac{\partial}{\partial y_{in}} p_k(y_i) = \frac{\partial}{\partial x_{in}} \frac{x_{ik}}{\sum_{l=1}^{M} x_{il}} = -\frac{x_{ik}}{\left(\sum_{l=1}^{M} x_{il}\right)^2} = -\frac{p_k(y_i)}{\sum_{l=1}^{M} x_{il}}.$$

Для 1.1:

$$\sum_{k=1}^{M}\left(\log\left(p_k(y_i)/p_k(y_j)\right)\frac{\partial}{\partial y_{in}}p_k(y_i)\right)=\frac{\log\left(p_n(y_i)/p_n(y_j)\right)}{\sum_{l=1}^{M}x_{il}}-\sum_{k=1}^{M}\log\left(p_k(y_i)/p_k(y_j)\right)\frac{p_k(y_i)}{\sum_{l=1}^{M}x_{il}}$$

$$=\frac{1}{\sum_{l=1}^{M}x_{il}}\left(\log\left(p_n(y_i)/p_n(y_j)\right)-\sum_{k=1}^{M}p_k(y_i)\log\left(p_k(y_i)/p_k(y_j)\right)\right)$$

Вывод 1.1.2:

$$for\ k=n:\quad \frac{\partial}{\partial y_{in}}\log\left(p_k(y_i)/p_k(y_j)\right)=\frac{\partial}{\partial y_{in}}\log\left(\frac{p_n(y_i)}{p_n(y_j)}\right)=\frac{p_n(y_j)}{p_n(y_i)}\frac{\partial}{\partial y_{in}}\frac{p_n(y_i)}{p_n(y_j)}$$

$$=\frac{p_n(y_j)}{p_n(y_i)}\frac{1}{p_n(y_j)}\frac{\partial}{\partial y_{in}}p_n(y_i)=\frac{1}{p_n(y_i)}\left(\frac{1}{\sum_{l=1}^{M}y_{il}}-\frac{y_{in}}{\left(\sum_{l=1}^{M}y_{il}\right)^2}\right)=\frac{1}{p_n(y_i)}\left(\frac{1}{\sum_{l=1}^{M}y_{il}}-\frac{p_n(y_i)}{\sum_{l=1}^{M}y_{il}}\right)$$

$$=\frac{1}{p_n(y_i)\sum_{l=1}^{M}y_{il}}-\frac{1}{\sum_{l=1}^{M}y_{il}},$$

$$for\ k\neq n:\quad \frac{\partial}{\partial y_{in}}\log\left(p_k(y_i)/p_k(y_j)\right)=\frac{\partial}{\partial y_{in}}\log\left(\frac{p_k(y_i)}{p_k(y_j)}\right)=\frac{p_k(y_j)}{p_k(y_i)}\frac{\partial}{\partial y_{in}}\frac{p_k(y_i)}{p_k(y_j)}$$

$$=\frac{p_k(y_j)}{p_k(y_i)}\frac{1}{p_k(y_j)}\frac{\partial}{\partial y_{in}}p_k(y_i)=-\frac{1}{p_k(y_i)}\frac{y_{ik}}{\left(\sum_{l=1}^{M}y_{il}\right)^2}=-\frac{1}{p_k(y_i)}\frac{p_k(y_i)}{\sum_{l=1}^{M}y_{il}}=-\frac{1}{\sum_{l=1}^{M}y_{il}}.$$

Для 1.1:

$$\sum_{k=1}^{M}p_k(y_i)\frac{\partial}{\partial y_{in}}\log\left(p_k(y_i)/p_k(y_j)\right)=\frac{p_n(y_i)}{p_n(y_i)\sum_{l=1}^{M}y_{il}}-\sum_{k=1}^{M}\frac{p_k(y_i)}{\sum_{l=1}^{M}y_{il}}=\frac{1}{\sum_{l=1}^{M}y_{il}}-\sum_{k=1}^{M}\frac{p_k(y_i)}{\sum_{l=1}^{M}y_{il}}$$

$$=\frac{1}{\sum_{l=1}^{M}y_{il}}\left(1-\sum_{k=1}^{M}p_k(y_i)\right).$$

Вывод 1.1 (продолжение):

$$\frac{\partial}{\partial y_{in}}D\left(y_i\middle\|y_j\right)=\sum_{k=1}^{M}\left(\log\left(p_k(y_i)/p_k(y_j)\right)\frac{\partial}{\partial y_{in}}p_k(y_i)+p_k(y_i)\frac{\partial}{\partial y_{in}}\log\left(p_k(y_i)/p_k(y_j)\right)\right)$$

$$= \frac{1}{\sum_{l=1}^{M} x_{il}} \left(\log\left(p_n(y_i)/p_n(y_j)\right) - \sum_{k=1}^{M} p_k(y_i) \log\left(p_k(y_i)/p_k(y_j)\right) \right) + \frac{1}{\sum_{l=1}^{M} y_{il}} \left(1 - \sum_{k=1}^{M} p_k(y_i) \right)$$

Заметим, что $\sum_{k=1}^{M} p_k(y_i) = \sum_{k=1}^{M} \frac{y_{ik}}{\sum_{l=1}^{M} y_{il}} = \frac{1}{\sum_{l=1}^{M} y_{il}} \sum_{k=1}^{M} y_{ik} = 1$.

Тогда

$$\frac{\partial}{\partial y_{in}} D\left(y_i \| y_j\right) = \frac{1}{\sum_{l=1}^{M} x_{il}} \left(\log\left(p_n(y_i)/p_n(y_j)\right) - \sum_{k=1}^{M} p_k(y_i) \log\left(p_k(y_i)/p_k(y_j)\right) \right)$$

$$= \frac{1}{\sum_{l=1}^{M} x_{il}} \left(\log\left(p_n(y_i)/p_n(y_j)\right) - D\left(y_i \| y_j\right) \right).$$

Вывод 1.2:

$$\frac{\partial}{\partial y_{in}} D\left(y_j \| y_i\right) = \frac{\partial}{\partial y_{in}} \sum_{k=1}^{M} p_k(y_j) \log\left(p_k(y_j)/p_k(y_i)\right) = \sum_{k=1}^{M} p_k(y_j) \frac{\partial}{\partial y_{in}} \log\left(p_k(y_j)/p_k(y_i)\right)$$

$$= \sum_{k=1}^{M} p_k(y_j) \frac{\partial}{\partial y_{in}} \log\left(p_k(y_j)/p_k(y_i)\right).$$

Вывод 1.2.1:

$$\frac{\partial}{\partial y_{in}} \log\left(p_k(y_j)/p_k(y_i)\right) = -\frac{p_k(y_i)}{p_k(y_j)} \frac{p_k(y_j)}{\left(p_k(y_i)\right)^2} \frac{\partial}{\partial y_{in}} p_k(y_i) = -\frac{1}{p_k(y_i)} \frac{\partial}{\partial y_{in}} p_k(y_i) \quad \{\text{см. 1.1.1}\}$$

$$\textit{for } k = n: \quad \frac{\partial}{\partial y_{in}} \log\left(p_k(y_j)/p_k(y_i)\right) = -\frac{1}{p_k(y_i)} \left(\frac{1}{\sum_{l=1}^{M} y_{il}} - \frac{p_k(y_i)}{\sum_{l=1}^{M} y_{il}} \right) = \frac{1}{\sum_{l=1}^{M} y_{il}} \left(1 - \frac{1}{p_k(y_i)} \right),$$

$$\textit{for } k \neq n: \quad \frac{\partial}{\partial y_{in}} \log\left(p_k(y_j)/p_k(y_i)\right) = -\frac{1}{p_k(y_i)} \left(-\frac{p_k(y_i)}{\sum_{l=1}^{M} y_{il}} \right) = \frac{1}{\sum_{l=1}^{M} y_{il}}.$$

Вывод 1.2 (продолжение):

$$\frac{\partial}{\partial y_{in}} D\left(y_j \| y_i\right) = \sum_{k=1}^{M} p_k(y_j) \frac{\partial}{\partial y_{in}} \log\left(p_k(y_j)/p_k(y_i)\right) = \sum_{k=1}^{M} p_k(y_j) \frac{1}{\sum_{l=1}^{M} y_{il}} - \frac{p_n(y_j)}{p_n(y_i)} \frac{1}{\sum_{l=1}^{M} y_{il}}$$

$$= \frac{1}{\sum_{l=1}^{M} y_{il}} \left(\sum_{k=1}^{M} p_k(y_j) - \frac{p_n(y_j)}{p_n(y_i)} \right)$$

Заметим, что $\sum_{k=1}^{M} p_k(y_j) = 1$. Тогда $\frac{\partial}{\partial y_{in}} D(y_j \| y_i) = \frac{1}{\sum_{l=1}^{M} y_{il}} \left(1 - \frac{p_n(y_j)}{p_n(y_i)} \right)$.

Вывод 1 (продолжение):

$$\frac{\partial}{\partial y_{in}} SID(y_i, y_j) = \frac{\partial}{\partial y_{in}} D(y_i \| y_j) + \frac{\partial}{\partial y_{in}} D(y_j \| y_i)$$

$$= \frac{1}{\sum_{l=1}^{M} y_{il}} \left(\log\left(p_n(y_i) / p_n(y_j) \right) - D(y_i \| y_j) \right) + \frac{1}{\sum_{l=1}^{M} y_{il}} \left(1 - \frac{p_n(y_j)}{p_n(y_i)} \right)$$

$$= \frac{1}{\sum_{l=1}^{M} y_{il}} \left(1 - \frac{p_n(y_j)}{p_n(y_i)} + \log\left(\frac{p_n(y_i)}{p_n(y_j)} \right) - D(y_i \| y_j) \right).$$

Окончательное решение:

$$\frac{\partial}{\partial y_{in}} \varepsilon = -2\mu \sum_{\substack{i,j=1, \\ i<j}}^{N} \left(SID(x_i, x_j) - SID(y_{i,} y_j) \right) \frac{\partial}{\partial y_{in}} SID(y_{i,} y_j)$$

$$= -\frac{2\mu}{\sum_{l=1}^{M} y_{il}} \sum_{\substack{j=1, \\ j \neq i}}^{N} \left(SID(x_i, x_j) - SID(y_{i,} y_j) \right) \left(1 - \frac{p_n(y_j)}{p_n(y_i)} + \log \frac{p_n(y_i)}{p_n(y_j)} - D(y_i \| y_j) \right).$$

Б.6. Дивергенция Хеллингера (HD)

Функция расстояния HD:

$$HD(x_i, x_j) = \sqrt{1 - \sum_{k=1}^{M} \sqrt{p_k(x_i) p_k(x_j)}}$$

$$p_k(x_i) = \frac{x_{ik}}{\sum_{l=1}^{M} x_{il}}$$

Функционал ошибки:

$$\varepsilon = \mu \sum_{i,j=1\,(i<j)}^{N} \left(HD(x_i, x_j) - HD(y_{i,} y_j) \right)^2 .$$

Решение:

$$\frac{\partial}{\partial y_{in}}\varepsilon = \frac{\partial}{\partial y_{in}}\mu \sum_{i,j=1\,(i<j)}^{N}\left(HD(x_i,x_j) - HD(y_{i,}y_j)\right)^2 = \mu \sum_{i,j=1\,(i<j)}^{N}\frac{\partial}{\partial y_{in}}\left(HD(x_i,x_j) - HD(y_{i,}y_j)\right)^2$$

$$= \mu \sum_{j=1\,(j\neq i)}^{N}\frac{\partial}{\partial y_{in}}\left(HD(x_i,x_j) - HD(y_{i,}y_j)\right)^2$$

$$= -2\mu \sum_{j=1\,(j\neq i)}^{N}\left(HD(x_i,x_j) - HD(y_{i,}y_j)\right)\frac{\partial}{\partial y_{in}}HD(y_{i,}y_j).$$

Вывод 1:

$$\frac{\partial}{\partial y_{in}}HD\left(y_i,y_j\right) = \frac{\partial}{\partial y_{in}}\sqrt{1-\sum_{k=1}^{M}\sqrt{p_k(y_i)p_k(y_j)}}$$

$$= \frac{\partial}{\partial y_{in}}\left(1-\sum_{k=1}^{M}\left(p_k(y_i)p_k(y_j)\right)^{1/2}\right)^{1/2}$$

$$= \frac{1}{2}\left(1-\sum_{k=1}^{M}\left(p_k(y_i)p_k(y_j)\right)^{1/2}\right)^{-1/2}\frac{\partial}{\partial y_{in}}\left(1-\sum_{k=1}^{M}\left(p_k(y_i)p_k(y_j)\right)^{1/2}\right)$$

$$= -\frac{1}{2}\left(HD\left(y_i,y_j\right)\right)^{-1}\sum_{k=1}^{M}\frac{\partial}{\partial y_{in}}\left(p_k(y_i)p_k(y_j)\right)^{1/2}.$$

Вывод 1.1:

$$\frac{\partial}{\partial y_{in}}\left(p_k(y_i)p_k(y_j)\right)^{1/2} = \frac{1}{2}\left(p_k(y_i)p_k(y_j)\right)^{-1/2}\frac{\partial}{\partial y_{in}}\left(p_k(y_i)p_k(y_j)\right)$$

$$= \frac{1}{2}\left(p_k(y_i)p_k(y_j)\right)^{-1/2}p_k(y_j)\frac{\partial}{\partial y_{in}}\left(p_k(y_i)\right).$$

Вывод 1.1.1:

$$for\ k=n:\quad \frac{\partial}{\partial y_{in}}p_k\left(y_i\right) = \frac{\partial}{\partial y_{in}}p_n\left(y_i\right) = \frac{\partial}{\partial y_{in}}\frac{y_{in}}{\sum_{l=1}^{M}y_{il}} = \frac{1}{\sum_{l=1}^{M}y_{il}} - \frac{y_{in}}{\left(\sum_{l=1}^{M}y_{il}\right)^2},$$

$$= \frac{1}{\sum_{l=1}^{M}y_{il}} - \frac{p_n\left(y_i\right)}{\sum_{l=1}^{M}y_{il}} = \frac{1-p_n\left(y_i\right)}{\sum_{l=1}^{M}y_{il}}$$

$$for\ k\neq n:\quad \frac{\partial}{\partial y_{in}}p_k\left(y_i\right) = \frac{\partial}{\partial x_{in}}\frac{y_{ik}}{\sum_{l=1}^{M}y_{il}} = -\frac{y_{ik}}{\left(\sum_{l=1}^{M}y_{il}\right)^2} = -\frac{p_k\left(y_i\right)}{\sum_{l=1}^{M}y_{il}}.$$

Производная HD:

$$\frac{\partial}{\partial y_{in}} HD\left(y_i, y_j\right) = -\frac{1}{2}\left(HD\left(y_i, y_j\right)\right)^{-1} \sum_{k=1}^{M}\left(\frac{1}{2}\left(p_k(y_i)p_k(y_j)\right)^{-1/2} p_k(y_j)\frac{\partial}{\partial y_{in}}\left(p_k(y_i)\right)\right)$$

$$= -\frac{1}{2}\left(HD\left(y_i, y_j\right)\right)^{-1} \sum_{k=1}^{M}\left(\frac{1}{2}\left(\frac{p_k(y_j)}{p_k(y_i)}\right)^{1/2} \frac{\partial}{\partial y_{in}}\left(p_k(y_i)\right)\right)$$

$$= -\frac{1}{2}\left(HD\left(y_i, y_j\right)\right)^{-1} \frac{1}{2\sum_{l=1}^{M} y_{il}}\left(\left(\frac{p_n(y_j)}{p_n(y_i)}\right)^{1/2} - \sum_{k=1}^{M}\left(\left(\frac{p_k(y_j)}{p_k(y_i)}\right)^{1/2} p_k(y_i)\right)\right)$$

$$= -\frac{1}{2}\left(HD\left(y_i, y_j\right)\right)^{-1} \frac{1}{2\sum_{l=1}^{M} y_{il}}\left(\left(\frac{p_n(y_j)}{p_n(y_i)}\right)^{1/2} - \sum_{k=1}^{M}\left(p_k(y_i)p_k(y_j)\right)^{1/2}\right)$$

или

$$\frac{\partial}{\partial y_{in}} HD\left(y_i, y_j\right) = \frac{1}{4\ HD\left(y_i, y_j\right)\sum_{l=1}^{M} y_{il}}\left(\sum_{k=1}^{M}\sqrt{p_k(y_i)p_k(y_j)} - \sqrt{\frac{p_n(y_j)}{p_n(y_i)}}\right).$$

Окончательное решение:

$$\frac{\partial}{\partial y_{in}}\varepsilon = -\mu \sum_{j=1\,(j\neq i)}^{N} \frac{HD(x_i, x_j) - HD(y_{i,}y_j)}{2\ HD\left(y_i, y_j\right)\sum_{l=1}^{M} y_{il}}\left(\sum_{k=1}^{M}\sqrt{p_k(y_i)p_k(y_j)} - \sqrt{\frac{p_n(y_j)}{p_n(y_i)}}\right).$$

Б.7. Угол Бхаттачария (Bca)

Функция расстояния BCa:

$$BCa\left(x_i, x_j\right) = \arccos\left(\sum_{k=1}^{M}\sqrt{p_k(x_i)p_k(x_j)}\right)$$

$$p_k\left(x_i\right) = \frac{x_{ik}}{\sum_{l=1}^{M} x_{il}}$$

Функционал ошибки:

$$\varepsilon = \mu \sum_{i,j=1\,(i<j)}^{N}\left(BCa(x_i, x_j) - BCa(y_{i,}y_j)\right)^2$$

Решение:

$$\frac{\partial}{\partial y_{in}}\varepsilon = \frac{\partial}{\partial y_{in}}\mu \sum_{i,j=1\,(i<j)}^{N}\left(BCa(x_i,x_j) - BCa(y_{i,}y_j)\right)^2$$

$$= \mu \sum_{i,j=1\,(i<j)}^{N}\frac{\partial}{\partial y_{in}}\left(BCa(x_i,x_j) - BCa(y_{i,}y_j)\right)^2$$

$$= \mu \sum_{j=1\,(j\neq i)}^{N}\frac{\partial}{\partial y_{in}}\left(BCa(x_i,x_j) - BCa(y_{i,}y_j)\right)^2$$

$$= -2\mu \sum_{j=1\,(j\neq i)}^{N}\left(BCa(x_i,x_j) - BCa(y_{i,}y_j)\right)\frac{\partial}{\partial y_{in}}BCa(y_{i,}y_j).$$

Вывод 1:

$$\frac{\partial}{\partial y_{in}}BCa\left(y_i,y_j\right) = \frac{\partial}{\partial y_{in}}\arccos\left(\sum_{k=1}^{M}\sqrt{p_k(y_i)p_k(y_j)}\right)$$

$$= -\left(1-\left(\sum_{k=1}^{M}\left(p_k(y_i)p_k(y_j)\right)^{1/2}\right)^2\right)^{-1/2}\frac{\partial}{\partial y_{in}}\sum_{k=1}^{M}\left(p_k(y_i)p_k(y_j)\right)^{1/2}$$

$$= -\left(1-\left(\sum_{k=1}^{M}\left(p_k(y_i)p_k(y_j)\right)^{1/2}\right)^2\right)^{-1/2}\sum_{k=1}^{M}\frac{\partial}{\partial y_{in}}\left(p_k(y_i)p_k(y_j)\right)^{1/2}.$$

Вывод 1.1:

$$\frac{\partial}{\partial y_{in}}\left(p_k(y_i)p_k(y_j)\right)^{1/2} = \frac{1}{2}\left(p_k(y_i)p_k(y_j)\right)^{-1/2}p_k(y_j)\frac{\partial}{\partial y_{in}}\left(p_k(y_i)\right)\ \{\text{см. Б.6}\}.$$

Вывод 1.1.1:

$$for\ k=n:\ \ \frac{\partial}{\partial y_{in}}p_k(y_i) = \frac{1-p_n(y_i)}{\sum_{l=1}^{M}y_{il}}\ \ \{\text{см. Б.6}\}$$

$$for\ k\neq n:\ \ \frac{\partial}{\partial y_{in}}p_k(y_i) = -\frac{p_k(y_i)}{\sum_{l=1}^{M}y_{il}}\ \ \{\text{см. Б.6}\}$$

Производная BCa:

$$\frac{\partial}{\partial y_{in}}BCa\left(y_i,y_j\right) =$$

$$-\left(1-\left(\sum_{k=1}^{M}\left(p_k(y_i)p_k(y_j)\right)^{1/2}\right)^2\right)^{-1/2}\sum_{k=1}^{M}\left(\frac{1}{2}\left(p_k(y_i)p_k(y_j)\right)^{-1/2}p_k(y_j)\frac{\partial}{\partial y_{in}}\left(p_k(y_i)\right)\right)\ \{\text{см. Б.6}\}$$

$$=-\left(1-\left(\sum_{k=1}^{M}\left(p_k(y_i)p_k(y_j)\right)^{1/2}\right)^2\right)^{-1/2}\frac{1}{2\sum_{l=1}^{M}y_{il}}\left(\left(\frac{p_n(y_j)}{p_n(y_i)}\right)^{1/2}-\sum_{k=1}^{M}\left(p_k(y_i)p_k(y_j)\right)^{1/2}\right)$$

или

$$\frac{\partial}{\partial y_{in}}BCa\left(y_i,y_j\right)=\frac{1}{2\sqrt{1-\left(\sum_{k=1}^{M}\sqrt{p_k(y_i)p_k(y_j)}\right)^2}\sum_{l=1}^{M}y_{il}}\left(\sum_{k=1}^{M}\sqrt{p_k(y_i)p_k(y_j)}-\sqrt{\frac{p_n(y_j)}{p_n(y_i)}}\right).$$

Решение:

$$\frac{\partial}{\partial y_{in}}\varepsilon=-\mu\sum_{j=1\,(j\neq i)}^{N}\frac{BCa(x_i,x_j)-BCa(y_{i,}y_j)}{\sqrt{1-\left(\sum_{k=1}^{M}\sqrt{p_k(y_i)p_k(y_j)}\right)^2}\sum_{l=1}^{M}y_{il}}\left(\sum_{k=1}^{M}\sqrt{p_k(y_i)p_k(y_j)}-\sqrt{\frac{p_n(y_j)}{p_n(y_i)}}\right).$$

Приложение В. Программная реализация разработанных алгоритмов формирования информативных признаков на графических процессорах

В.1. Общие сведения о технологии CUDA

Представленный в настоящем приложении материал в значительной степени опирается на специфику и особенности программирования для графических процессоров, поддерживающих технологию CUDA. По этой причине в настоящем разделе приводятся некоторые известные сведения об указанной технологии, позволяющие лучше понять содержание приложения.

Модель вычислений в технологии CUDA [223] основана на использовании как центрального процессора (Central Processor Unit - CPU) и оперативной памяти, расположенной на хост-машине (host), так и графического процессора NVIDIA (Graphic Processor Unit - GPU) с его собственной памятью, расположенной на устройстве (device).

Хост управляет как распределением памяти на устройстве, так и запуском на нем вычислений. Сами алгоритмы вычислений на устройстве программно реализуются в виде специального вида функций, называемых функциями ядра (kernels). Функции ядра могут исполняться на устройстве одновременно тысячами вычислительных ядер CUDA (cores), за счет чего достигается высокая степень параллелизма.

Для управления большим количеством потоков используется архитектура, названная NVIDIA SIMT - Single-Instruction, Multiple-Thread (одна инструкция — множество потоков), схожая с архитектурой SIMD (одна инструкция — множество данных). В такой архитектуре одна исполняемая инструкция управляет сразу многими элементами данных. Ключевое отличие здесь в том, что в SIMT для каждого элемента обрабатываемых данных работает отдельный поток команд, каждый из которых может иметь свое поведение, включая выполнение различных веток кода.

Архитектура графического процессора NVIDIA построена на основе модели масштабируемого массива потоковых мультипроцессоров (Stream Multiprocessors - SM). Рассмотрим для примера широко распространенный графический процессор

NVIDIA GeForce GTX-1070 [88]. Этот процессор разделен на 3 кластера обработки графики (Graphics Processing Cluster - GPC), где каждый GPC, в свою очередь, имеет 5 потоковых мультипроцессоров (SM) со 128 CUDA-ядрами в каждом.

Каждый потоковый мультипроцессор здесь разделен на четыре части (см. рисунок В.1), каждая из которых содержит планировщик варпов (warp sheduler), ответственный за организацию выполнения потоков группами по 32 потока — варпами (warps). Каждый варп закрепляется за диспетчером (Dispatcher Unit).

При запуске функции ядра с хост-машины на GPU направляется запрос на создание сетки потоков (grid), в которой потоки группируются в блоки (blocks), как показано на рисунке В.2. Сетка потоков имеет конфигурируемый размер, определяемый при запуске ядра на выполнение. Настраиваться может как количество и размер блоков, так и их конфигурация: поддерживаются одно, двух и трехмерные структуры. Максимальное количество потоков в блоке ограничено числом 1024.

После получения запроса блок распределения (Work Distribution Unit), назначает блоки потоков потоковым мультипроцессорам (SM), выбирая те мультипроцессоры, у которых достаточно ресурсов для выполнения задачи.

Потоки в одном блоке выполняются одновременно на одном мультипроцессоре, и на одном мультипроцессоре могут выполняться одновременно несколько блоков. По мере завершения блоков на освободившихся мультипроцессорах запускаются новые блоки. Таким образом происходит равномерное распределение потоков между SM в графическом процессоре.

Когда мультипроцессору поручают выполнить один или несколько блоков потоков, он разделяет их на варпы, и каждый варп планируется для выполнения планировщиком варпов (warp sheduler). Блок разделяется на варпы всегда одним и тем же способом. Идентификаторы потоков в варпе увеличиваются последовательно, начиная с идентификатора 0 в первом варпе.

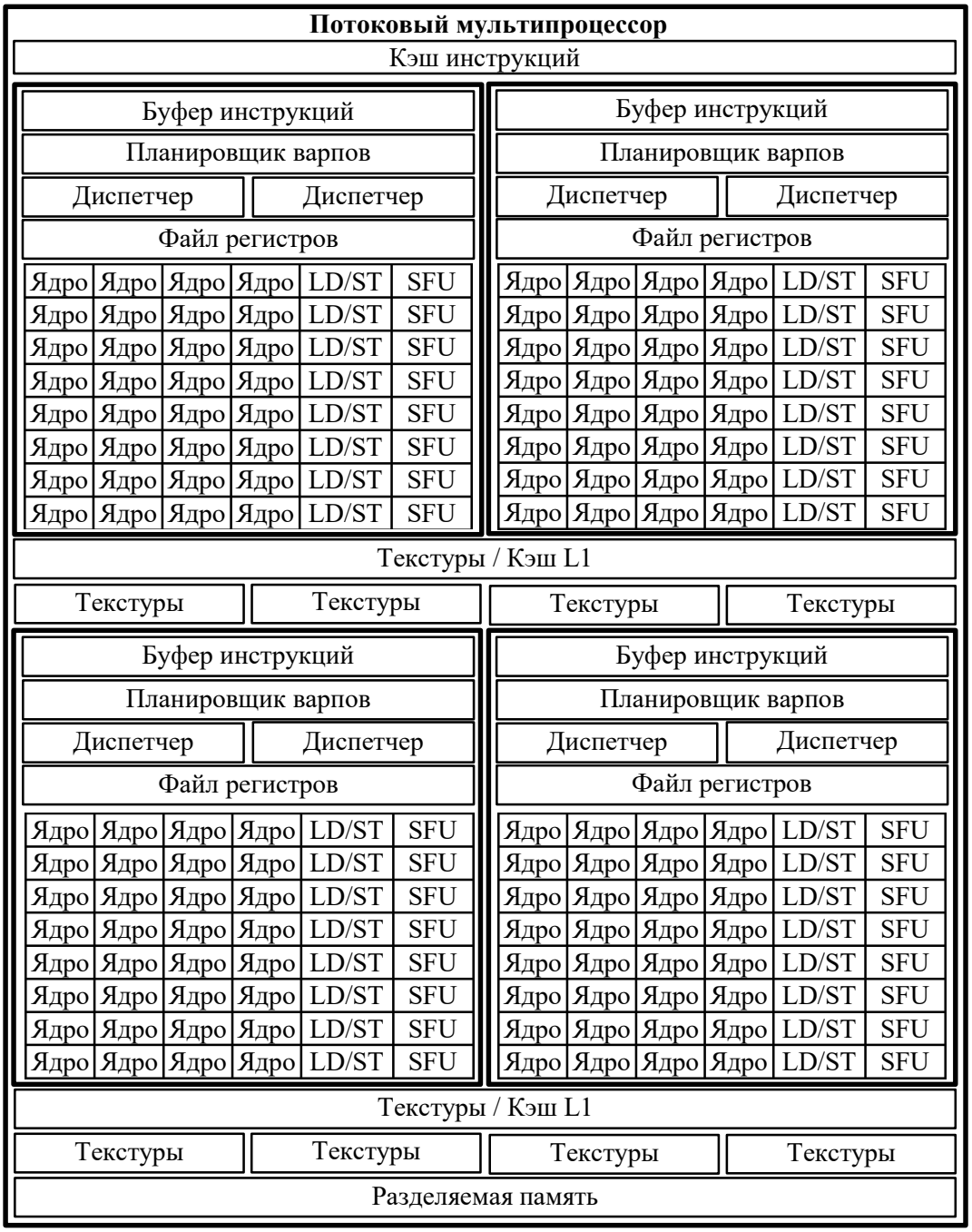


Рисунок В.1 - Архитектура потового мультипроцессора графического процессора NVIDIA GeForce GTX-1070 (перевод схемы из [271])

Ожидается, что все потоки блока будут находиться в одном потоковом многопроцессорном ядре и должны совместно использовать ограниченные ресурсы памяти этого ядра. В современных графических процессорах блок потоков может содержать до 1024 потоков. Однако ядро может выполняться несколькими блоками потоков одинаковой формы, так что общее количество потоков равно количеству потоков в блоке, умноженному на количество блоков.

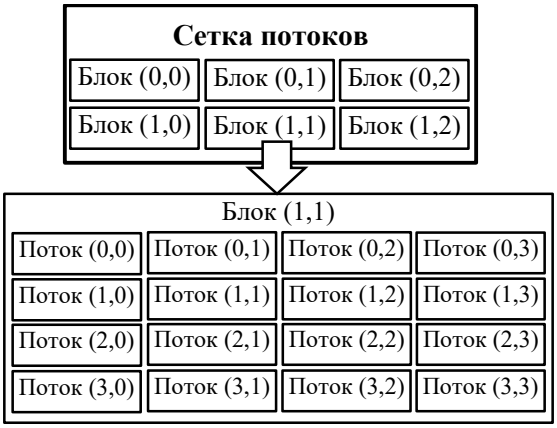


Рисунок В.2 - Сетка потоков

В новых версиях архитектуры (Compute Capability 9.0 , поддерживаемой сейчас графическим процессором NVIDIA H100 [224]) вводится дополнительный уровень иерархии — кластеры блоков потоков, которые, не используются в настоящей работе.

Выполнение отдельных потоков варпа начинается одновременно по одному и тому же адресу в программе. Однако у каждого потока есть собственный счетчик адреса команд и состояние регистров, они могут разветвляться и выполняться независимо (но в этом случае не одновременно).

Потоки варпа выполняют одну и ту же инструкцию в один момент времени. Если потоки варпа должны разойтись по причине условного ветвления, зависящего, например, от значений переменных, варп будет последовательно выполнять каждую из ветвей кода, отключая от исполнения потоки, находящиеся на альтернативном пути. Таким образом, наибольшая эффективность достигается, когда все 32 потока варпа согласованно выполняют одни и те же ветви кода. Следует отметить, что разные варпы выполняются независимо друг от друга и могут одновременно следовать по различным ветвям кода.

Сами инструкции при выполнении обрабатываются конвейерным образом, что позволяет использовать параллелизм на уровне инструкций в рамках одного потока, помимо общего параллелизма, реализуемого за счет аппаратной многопоточности. Инструкции выдаются по порядку, без предсказания ветвлений и спекулятивного расчета последующих инструкций (имеется в виду выполнение инструкций, которые, возможно, окажутся ненужными, в целях устранения задержек в случаях, когда инструкции все же окажутся необходимыми).

Контекст выполнения (программные счетчики, регистры и т. д.) для каждого варпа сохраняется в течение всего времени существования варпа. Переключение с одного контекста выполнения на другой выполняется без дополнительных затрат.

Каждый поток имеет собственную локальную память. Каждый блок потоков имеет общую память, видимую для всех потоков блока и имеющую то же время жизни, что и сам блок. Потоки в блоке могут выполнять операции чтения, записи и атомарные операции в общей памяти. Все потоки, в том числе из различных блоков, имеют доступ к одной и той же глобальной памяти. Таким образом, потоки могут взаимодействовать, разделяя данные через общую или глобальную память и синхронизируя выполнение.

Есть две дополнительные области памяти только для чтения, доступные всем потокам: память констант и память текстур, которые не используются в настоящем исследовании.

Компиляция программ для CUDA выполняется компилятором CUDA (nvcc.exe). При этом для кода, исполняемого на хосте, поддерживается полная версия языка C++. Для кода, исполняемого на устройстве, поддерживается лишь некоторое подмножество C++. При компиляции nvcc отделяет код, подлежащий исполнению на устройстве, от кода, исполняемого на хосте, после чего выполняет компиляцию кода устройства в ассемблерную форму (код PTX) и/или бинарную форму (объект cubin). Код, подлежащий исполнению на хосте, претерпевает изменения — в место вызова функций ядра вставляются необходимые вызовы функций библиотеки CUDA для загрузки и запуска кода, скомпилированного из кода PTX или объектов cubin.

Код PTX, загружаемый во время выполнения, компилируется драйвером устройства в двоичный код на лету (Just-in-time компиляция), что позволяет использовать обновленные версии компилятора и запускать приложения на новых устройствах, которые не существовали на момент создания приложения.

B.2. Реализация алгоритмов формирования информативных признаков на графических процессорах NVIDIA

Программная реализация алгоритмов формирования информативных признаков для исполнения на графических процессорах NVIDIA выполнена с использованием языка C++. Разработка проводилась в инструментальной среде Microsoft Visual Studio [301].

Основная цель реализации состоит в повышении вычислительной эффективности за счет распараллеливания обработки. Для достижения этой цели задача формирования информативных признаков должна быть разбита на укрупненные подзадачи, которые будут реализовываться функциями ядра так, чтобы каждая подзадача могла быть распараллелена на более мелкие части и решаться большим количеством потоков, исполняемых на cuda-ядрах.

B.2.1. Общая схема вычислений

Собственно реализация процедуры формирования признаков выполняется в несколько этапов:

1. Загрузка входных данных. На этом этапе в зависимости от возможного источника данных (обменный файл в бинарном или текстовом формате) хост-машина осуществляет загрузку входных данных в оперативную память.

2. Инициализация данных во входном и формируемом пространствах. На этом этапе на хост-машине выполняется преобразование входного многомерного пространства (например, формируется пространственный контекст или подготавливаются системы признаков для слияния, см. раздел 3), а также инициализируются данные в формируемом пространстве (например, выполняется линейное преобразование с использованием метода главных компонент).

3. Перенос данных на устройство. На этом этапе в глобальной памяти устройства GPU осуществляется выделение памяти для массивов входных, выходных данных и вектора ошибок, выполняется копирование инициализированных на предыдущем этапе данных на устройство.

4. Численная оптимизация функционала ошибки. Этап численной оптимизации выполняется совместно на хосте и устройстве и состоит из следующих шагов:

4.1. Предварительное вычисление масштабного множителя μ из выражения (2.9). Так как масштабный множитель зависит лишь от распределения данных во входном пространстве, имеет смысл предварительно рассчитать этот множитель, чтобы в дальнейших вычислениях использовать уже рассчитанное значение. Основная часть вычислений при расчете множителя производится на GPU.

4.2. Основной цикл оптимизации. На этом шаге с использованием алгоритма стохастического градиентного спуска (см. подраздел 4.2) итеративно выполняется обновление координат точек в формируемом пространстве признаков с контролем

ошибки. Основные вычисления как при расчете обновленных координат точек, так и при оценке ошибки представления данных в формируемом пространстве, производятся на устройстве.

4.3. Оценка окончательной ошибки представления данных в формируемом пространстве признаков. Этап также выполняется с использованием GPU.

5. Перенос данных с устройства на хост-машину. На этом этапе результат численной оптимизации копируется с устройства на хост-машину. Производится освобождение глобальной памяти на устройстве.

6. Сохранение данных. Полученные на предыдущем шаге результаты сохраняются из оперативной памяти в постоянную. Производится освобождение занимаемых программой ресурсов.

Как видно из представленной схемы, наибольший интерес здесь представляет четвертый этап обработки, на котором основной объем вычислений производится на графическом процессоре. Прежде чем перейти к этому этапу, рассмотрим некоторые важные для его реализации конструкции.

B.2.2. Реализация расчета расстояний

Для обеспечения применимости созданного решения к широкому множеству входных и выходных функций расстояния, изложенных в разделе 2.1, используется подход на основе шаблонов (templates) C++. При этом функции расстояния описываются в виде структур, которые далее используются в качестве параметров шаблонов для построения других функций и классов в программе.

Каждая входная и выходная функция расстояния описывается в виде структуры следующего вида:

```
struct Dissimilarity {
        static inline DEVICE FTYPE dist(FTYPE *p, FTYPE *q, int dx);
        static inline FTYPE dist_h(FTYPE* p, FTYPE* q, int dx);
        static inline DEVICE FTYPE dist(FTYPE* p, FTYPE* q,
                DPARAMS pars)
};
```

Здесь статическая встраиваемая функция dist осуществляет расчет расстояния между векторами p и q в пространстве размерности dx на устройстве CUDA. Аналогичная функция dist_h выполняет аналогичный расчет на хост-машине (центральном процессоре).

Перегруженный вариант функции `dist` с параметром типа `DPARAMS` применяется для функций рассогласования, представляющих собой параметризованные расстояния, например, метрики l^p; расстояния, используемые при слиянии признаков; расстояния, используемые при применении пространственного контекста и т. п. Реализация функции `dist` для базовых функций расстояния тривиальна и заменяется компилятором на вызов `dist(FTYPE*, FTYPE*, int)`.

Примечание - здесь и далее часто используются макроопределения, такие как DEVICE, FTYPE и др. Их основной целью является унификация кода при исполнении на устройстве и центральном процессоре, использование различных типов данных и т.д. В целях настоящего приложения будем считать, что DEVICE будет заменен на стандартный модификатор __device__ (функция, исполняемая и вызываемая только на устройстве), GLOBAL – на __global__ (функция, вызываемая на хост-машине и исполняемая на устройстве), FTYPE – на float.

Используя структуры, реализующие базовые функции расстояния, можно создавать структуры для расчета агрегирующих расстояний. Например, для функции расстояния, построенной по принципу обратного взвешивания (см. выражение (3.4) в подразделе 3.2), получаем следующий шаблон:

```
template <class BaseDissimilarity>
struct InverseWeightingDissimilarity {
	static inline DEVICE FTYPE dist(FTYPE* p, FTYPE* q, DPARAMS pars) {
		FTYPE d = 0;
		for (int z = 1; z <= pars.count; z++) {
			d += BaseDissimilarity::dist(p, q, pars.dx1) / z;
			p += pars.dx1;   q += xpars.dx1;
		}
		return d;
	}
	//...
};
```

Здесь `BaseDissimilarity` — базовый тип расстояния, `pars.count` — количество базовых расстояний, принимающих участие в расчете агрегирующей меры рассогласования, `pars.dx1` — размерность пространства, в котором определена базовая функция расстояния. Ясно, что при таком способе расчета компоненты агрегируемых векторов должны располагаться последовательно в смежных областях памяти.

B.2.3. Реализация численной оценки градиента

Фактически, пара типов (подобных Dissimilarity), задающая входную и выходную функции расстояния, определяет в рамках выбранного метода стохастической оптимизации применяемый алгоритм формирования признаков с точностью до некоторых параметров, таких как размерности пространств. Реализация набора алгоритмов формирования признаков в части численной оценки градиента основывается на семействе классов, порождаемых шаблоном следующего вида:

```
template <class InputDis, class OutputDis>
class Calc{
  public:
        inline static DEVICE void Increment(FTYPE *hHighData, int i, int j,
                DPARAMS pars, FTYPE *hLowData, FTYPE *y, int dy);
        static void InitHD(FTYPE *hData, XPARAMS XPARS);
        static void Init(FTYPE *hData, int dy);
        static void Uninit(FTYPE *hData, int dy);
        // ...
};
```

Здесь InputDis и OutputDis — пара типов, определяющих входную и выходную функции расстояния, Increment — функция, выполняющая оценку градиента, InitHD — функция, выполняющая инициализацию (преобразование) данных во входном пространстве, Init и Uninit — функции, выполняющие инициализацию координат в формируемом пространстве и очистку связанных ресурсов, соответственно.

Именно функции InitHD и Init реализуют основную часть второго этапа приведенной в п. B.2.1«Общей схемы вычислений».

Базовая реализация шаблона Calc выполнена для класса l^p метрик в качестве входной/выходной функций расстояния. Наиболее важной в шаблонном классе Calc является функция Increment. Пример ее реализации для метрики l^p в качестве функции расстояния в выходном (формируемом) пространстве приведен ниже:

```
inline static DEVICE void Increment( FTYPE *hHighData, int i, int j, DPARAMS pars,
                                     FTYPE *hLowData1, FTYPE *y, int dy )
{
        FTYPE dm = InputDis::dist( hHighData + i * xpars.dxk(),
                                   hHighData + j * xpars.dxk(),
                                   pars );
        FTYPE dl = OutputDis::dist( hLowData1 + i * dy,
                                    hLowData1 + j * dy,
                                    dy );
        for (int c = 0; c < dy; c++)
                y[c] += ( (dm - dl) / powf(dl, OutputDis::Param() - 1) ) *
```

```
                ( hLowData1[i*dy+c] - hLowData1[j*dy+c] ) *
                 powf( fabs(hLowData1[i*dy+c] - hLowData1[j*dy+c]),
                       OutputDis::Param()-2 );
}
```

На вход функции подаются матрицы входных hHighData и выходных hLowData1 данных, индекс i точки (пикселя ГСИ), для которой производится расчет градиента, индекс j точки (пикселя ГСИ), по которой производится оценка градиента, параметры pars входного пространства, а также вектор формируемого градиента y и его размерность dy.

Функция сначала производит расчет расстояний dm и dl во входном и выходном пространствах, соответственно, с использованием параметров шаблона InputDis и OutputDis. Затем покомпонентно (для каждой координаты c) рассчитывает градиент и аккумулирует его в вектор y. Здесь OutputDis::Param() возвращает параметр p метрики l^p.

Так как реализация функции Increment будет отличаться для различных функций расстояния в выходном пространстве, конкретное поведение функции Increment задается путем специализации самой функции шаблонного класса Calc или специализации класса в целом для конкретной функции расстояния в выходном пространстве. Кроме того, отличаться могут и способы инициализации данных (Init, InitHD) во входном / выходном пространствах.

В качестве примера приведем специализацию функции шаблонного класса InitHD для подготовки входного пространства при использовании функции расстояния с учетом пространственного контекста по методу из подраздела 3.2. Саму функцию расстояния введем как расстояние обратного взвешивания над евклидовыми расстояниями:

```
typedef InverseWeightingDissimilarity<Eucl> EucSp;
```

Далее в шаблонном классе Calc выполним специализацию метода InitHD, выполняющего подготовку данных во входном многомерном пространстве, следующим образом:

```
template <>
static void Calc<EucSp, Euc>::InitHD(FTYPE* hData, DPARAMS pars) {
        _InitHD_by_sort<Euc>(hData, pars);
}
```

Здесь реализация специализированного метода в свою очередь вызывает шаблонную функцию _InitHD_by_sort для формирования входного пространства путем

конкатенации пикселей входного изображения из локального пространственного контекста с выполнением сортировки по евклидову расстоянию в соответствии с методом, изложенным в подразделе 3.2.

B.2.4. Предварительное вычисление масштабного множителя

В случае, когда масштабный множитель μ в выражении (2.9) отличен от единицы и зависит от расстояний, его прямой расчет при большом количестве точек может занять существенные ресурсы. По этой причине в настоящей работе рассчитывается оценка множителя на основе случайного подмножества точек.

Основной объем вычислений выполняется на устройстве в функции ядра kernel_coeff. Эта функция представляет собой шаблон, порождающий семейство функций ядра для различных функций расстояния InputDis во входном пространстве. Приведем реализацию функции kernel_coeff для оценки масштабного множителя в соответствии с выражением (2.14):

```
template <class InputDis>
GLOBAL void kernel_coeff( FTYPE *dHighData,
                          FTYPE* E,
                          int* dApproxBatch,
                          int Start,
                          int Stop,
                          bool init,
                          int np,
                          DPARAMS xpars )
{
        int idx = blockDim.x * blockIdx.x + threadIdx.x;
        FTYPE dm, sum_dm;
        if (idx < np)  {
                sum_dm = 0.0;
                for (int k = Start; k <= Stop; k++){
                        int j = dApproxBatch[k];
                        dm = InputDis::dist( dHighData + idx*xpars.dxk(),
                                             dHighData + j*xpars.dxk(),
                                             xpars);
                        sum_dm += dm*dm;
                }
                if (init)  E[idx] = sum_dm;
                else  E[idx] += sum_dm;
        }
}
```

Функция имеет следующие входные параметры: dHighData – массив входных данных высокой размерности, xpars – параметры входной функции расстояния, np – количество точек данных, dApproxBatch – набор случайных индексов точек для

оценки масштабного множителя, Start и Stop – диапазон индексов в наборе dApproxBatch по которому будут вычисляться оценки, E – вектор результатов для оценки, init – признак необходимости очистки вектора результатов.

При выполнении на устройстве CUDA, с использованием индекса blockIdx.x и размера blockDim.x блока, а также индекса выполняемого потока threadIdx.x, каждый экземпляр рассчитывает индекс idx точки данных, за который этот экземпляр функции отвечает.

Далее, если этот индекс не выходит за границы набора данных, выполняется цикл, в котором происходит следующее:

- Для оценки масштабного множителя по случайному подмножеству dApproxBatch, индекс k в этом подмножестве пересчитывается в индекс j в общем наборе данных.

- Рассчитывается и сохраняется в переменную dm входное расстояние между idx-й и j-й точками.

- Квадраты входных расстояний накапливаются в переменной sum_dm, после чего рассчитанная сумма сохраняется (либо аккумулируется) в соответствующий индексу точки idx элемент массива E.

Таким образом, после одного запуска функции ядра на GPU элементы массива E будут заполнены частными суммами следующего вида:

$$E[idx] = \sum_{j=Start}^{Stop} \delta^2\left(x_{idx}, x_{r[j]}\right),$$

где функция расстояния δ соответствует статической функции InputDis::dist, а случайная выборка r – входному массиву dApproxBatch.

Фрагмент кода, осуществляющий запуск приведенной выше функции ядра kernel_coeff на устройстве CUDA так, чтобы в расчете приняли участие все индексы из dApproxBatch, показан ниже:

```
for (int ks = 0; ks < estimSz; ks += BATCH_SIZE) {
        int ke = fin_idx(ks, BATCH_SIZE, estimSz);
        kernel_coeff<InputDis>   <<<(np - 1) / block_size + 1, block_size>>>
                ( dHighData,
                  dErrors,
                  dApproxBatch,
```

```
                ks,
                ke,
                (ks == 0),
                np,
                PARS );
        CUDACHECK(cudaThreadSynchronize());
    }
```

Здесь estimSz — размер случайной выборки, по которой осуществляется оценка, BATCH_SIZE — часть случайной выборки, на которой будет выполняться функция ядра, ks...ke — диапазон индексов случайной выборки. Причиной, по которой случайная выборка объемом estimSz разбивается на фрагменты размером BATCH_SIZE, является то, что при достаточно большом объеме выборки и использовании слабого графического процессора функция ядра может выполняться слишком долго, что может быть принято операционной системой за зависание программы.

После запуска поблочного суммирования с использованием функции ядра kernel_coeff осуществляется копирование результатов на хост-машину с последующим агрегированием результатов и расчетом масштабного множителя.

B.2.5. Основной цикл оптимизации

Основной цикл оптимизации реализуется с использованием двух массивов данных для хранения координат точек в формируемом пространстве. На нечетных итерациях для расчета используются данные из первого массива, а сохранение осуществляется во второй массив. На четных итерациях, наоборот, рассчитанные на четной итерации данные, хранящиеся во втором массиве, используются для обновления координат в первом массиве.

Основная часть вычислений при оптимизации выполняется на устройстве в функции ядра kernel_base, отвечающей за обновление координат точек в формируемом пространстве. Базовая версия функции ядра выглядит следующим образом:

```
template <class InputDis, class OutputDis>
GLOBAL void kernel_base( const FTYPE* dLowData1,
                         FTYPE* dLowData2,
```

```
                                FTYPE m,
                                const FTYPE* dHighData,
                                const int* dApproxBatch,
                                int Start,
                                int Stop,
                                bool initbyzero,
                                int np,
                                DPARAMS pars,
                                int dy )
{
        int idx = blockDim.x * blockIdx.x + threadIdx.x;

        FTYPE y[MAX_DY];
        for (int c = 0; c < dy; c++)   y[c] = 0;

        if (idx < np) {
                for (int k = Start; k <= Stop; k++) {
                        int j = dApproxBatch[k];
                        Calc<InputDis, OutputDis>::Increment( dHighData,
                                                              idx,
                                                              j,
                                                              pars,
                                                              dLowData1,
                                                              y,
                                                              dy );
                }
                if (ENormType::ntNone == Calc<InputDis, OutputDis>::NormType()) {
                        if (initbyzero) {
                           for (int c = 0; c < dy; ++c)
                              dLowData2[idx*dy+c] = (dLowData1[idx*dy+c] + m*y[c]);
                        }
                        else{
                           for (int c = 0; c < dy; ++c)
                              dLowData2[idx*dy+c] = (dLowData2[idx*dy+c] + m*y[c]);
                        }
                }
                // …
        }
}
```

Функция имеет следующие входные параметры: dLowData1 и dLowData1 - массивы данных для координат точек в выходном пространстве, m – масштабный множитель, dHighData – массив входных данных высокой размерности, dApproxBatch – набор случайных индексов для оценки стохастического градиента, Start и Stop – диапазон индексов в наборе dApproxBatch, по которому будет вычисляться оценка, initbyzero - признак необходимости инициализации вектора градиента нулевыми значениями, np – количество точек данных, pars – параметры

функции расстояния во входном пространстве, dy – размерность выходного пространства.

В этой функции с использованием индекса blockIdx.x и размера блока blockDim.x, а также индекса выполняемого потока threadIdx.x рассчитывается индекс idx точки данных, за которую отвечает выполняемый экземпляр функции. Далее по диапазону Start…Stop подмножества случайных индексов dApproxBatch выполняется оценка стохастического градиента с использованием статической функции Increment шаблонного класса Calc<InputDis, OutputDis>, описанного в п. В.2.3. Промежуточный результат оценки градиента накапливается таким образом в векторе y. После оценки градиента выполняется шаг алгоритма стохастического градиентного спуска, результат которого сохраняется в соответствующую строку массива выходных координат dLowData2.

В зависимости от конкретного алгоритма снижения размерности выходные координаты могут быть тем или иным способом преобразованы. Например, при отображении на поверхность единичной сферы (см. п. 2.2.2.1) может потребоваться нормализация длины формируемого вектора. Эта часть реализации для краткости в приведенном выше коде опущена, а показана лишь реализация по-умолчанию, в которой нормализации не требуется (ENormType::ntNone).

Собственно реализация цикла оптимизации состоит из следующих шагов:

1. Запуск функции ядра kernel_base на GPU по всем диапазонам подмножества случайных индексов для расчета обновленных координат точек в выходном пространстве:

```
for ( int ks = estimSz*it; ks < estimSz*(it+1); ks += BATCH_SIZE ) {
        int ke = fin_idx( ks, BATCH_SIZE, estimSz*(it+1) );

        kernel_base<InputMetricFun, OutputMetricFun>
                <<<(np - 1) / BLOCK_SIZE + 1, BLOCK_SIZE>>>
                        ( dLowData1,
                                dLowData2,
                                m,
                                dHighData,
                                dApproxBatch,
                                ks,
                                ke,
                                (ks == BATCH_SIZE * it),
                                np,
                                pars,
                                dy );
                CUDACHECK(cudaThreadSynchronize());
```

}

2. Оценка текущей ошибки представления данных в формируемом пространстве меньшей размерности по случайному подмножеству dApproxBatch. Оценка выполняется также с использованием GPU и более подробно рассматривается в следующем подразделе.

3. Анализ поведения ошибки представления данных в формируемом пространстве меньшей размерности и принятие решения об останове алгоритма или модификации коэффициента градиентного спуска.

4. Взаимный обмен указателей на массивы данных dLowData1 и dLowData1 для хранения координат точек в формируемом пространстве и переход к шагу 1.

B.2.6. Оценка ошибки представления данных в формируемом пространстве меньшей размерности

Расчет ошибки представления данных в формируемом пространстве выполняется как один из шагов цикла оптимизации (шаг 3 алгоритма в разделе B.2.5), необходимый для принятия решения об остановке алгоритма или изменении коэффициента градиентного спуска, а также для оценки ошибки после завершения вычислений (шаг 4.3 алгоритма в разделе B.2.1). Разница в расчете состоит лишь в случайной выборке, по которой производится оценка. Так, в рамках цикла оптимизации для оценки используется та же выборка, что и при оценке стохастического градиента. После завершения оптимизации ошибка оценивается по выборке большего объема.

Как и в предыдущих случаях, основная работа выполняется на GPU в следующей шаблонной функции ядра:

```
template <class InputDis, class OutputDis>
GLOBAL void kernel_error( FTYPE *dLowData,
                          FTYPE *dHighData,
                          FTYPE* E,
                          int* dApproxBatch,
                          int Start,
                          int Stop,
                          bool initbyzero,
                          int np,
                          DPARAMS pars,
                          int dy )
{
    int idx = blockDim.x * blockIdx.x + threadIdx.x;
    FTYPE err = 0.0f;
```

```
        FTYPE dm, dl;

        if (idx < np)  {
                for (int k = Start; k <= Stop; k++) {
                        int j = dApproxBatch[k];
                        dm = InputDis::dist( dHighData + idx*xpars.dxk(),
                                             dHighData + j * xpars.dxk(),
                                             pars );
                        dl = OutputDis::dist( dLowData + idx*dy,
                                              dLowData + j*dy,
                                              dy );
                        err += ((dm - dl)*(dm - dl)) * ro_ij;
                }
                if (initbyzero)  E[idx] = err;
                else  E[idx] += err;
        }
}
```

Здесь параметры dLowData и dHighData - массивы координат точек в формируемом (выходном) и входном пространствах, соответственно, E – выходной вектор частных сумм для расчета ошибки представления данных в формируемом пространстве меньшей размерности, dApproxBatch – набор случайных индексов для оценки ошибки, Start и Stop – диапазон индексов в наборе dApproxBatch, по которому будет вычисляться оценка, initbyzero – признак необходимости инициализации выходного вектора частных сумм нулевыми значениями, np – количество точек данных, pars – параметры входной меры рассогласования, dy – размерность выходного пространства.

По своей структуре функция схожа с ранее описанными функциями ядра. После определения индекса idx точки, за обработку которой отвечает выполняемый экземпляр функции, функция итеративно проходит по диапазону Start…Stop случайных индексов из выборки dApproxBatch, рассчитывает значения расстояний dm и dl в пространствах высокой и низкой размерности и накапливает сумму:

$$err = \sum_{k=Start}^{Stop} \rho_{ij} \left(\delta\left(x_i, x_{r[k]}\right) - \psi\left(y_i, y_{r[k]}\right) \right)^2$$

для оценки функционала ошибки (2.9). Здесь ρ_{ij} соответствуют используемому в функции макроопределению ro_ij, функции расстояния δ и ψ - статическим функциям InputDis::dist и OutputDis::dist, а случайная выборка r – входному массиву dApproxBatch.

Накопленная сумма сохраняется (аккумулируется) в выходной вектор частных сумм для расчета ошибки E представления данных в формируемом пространстве меньшей размерности.

Фрагмент кода, выполняющий расчет ошибки представления данных в пространстве меньшей размерности, включающий запуск рассмотренной функции ядра kernel_error выглядит следующим образом:

```
for (int ks = 0; ks < ErrBatchSz; ks +=BATCH_SIZE)  {
        int ke = fin_idx(ks, BATCH_SIZE, ErrBatchSz);
        kernel_error< InputDis, OutputDis >
                        <<<(ErrBatchSz - 1) / BLOCK_SIZE + 1, BLOCK_SIZE>>>
                                                ( dLowData2,
                                                  dHighData,
                                                  dErrors,
                                                  dApproxBatch,
                                                  ks,
                                                  ke,
                                                  (ks == 0),
                                                  np,
                                                  pars,
                                                  dy );
        CUDACHECK(cudaThreadSynchronize());
}
CUDACHECK( cudaMemcpy( hErrors,
                                dErrors,
                                ErrBatchSz * sizeof(FTYPE),
                                cudaMemcpyDeviceToHost ) );
double er = 0;
for (int i = 0; i < np; i++)  er += hErrors[i];
er = 0.5 * er / m;
```

Здесь ErrBatchSz — размер случайной выборки индексов, по которым будет выполняться оценка ошибки представления данных в пространстве меньшей размерности, dApproxBatch - сама случайная подвыборка, m – вычисленный заранее по той же выборке масштабный множитель (см. п. В.2.4).

В представленном фрагменте программы сначала выполняется расчет вектора частных сумм dErrors для ошибки представления данных в пространстве меньшей размерности. Затем рассчитанные суммы переносятся с устройства на хост-машину с использованием функции cudaMemcpy. Далее производится аккумулирование частных сумм и расчет окончательного значения ошибки er с учетом масштабного коэффициента m.

B.3. Повышение эффективности решения при взаимодействии с памятью

К сожалению, вычислительная эффективность приведенного выше решения может быть ограничена по причине ряда особенностей взаимодействия с памятью. Рассмотрим их.

B.3.1. Особенности взаимодействия с памятью при вычислении на GPU

В приведенных выше функциях ядра основными параметрами были указатели на массивы координат во входном и формируемом (выходном) пространствах. Ввиду большого размера этих массивов, они размещались в глобальной памяти устройства.

Глобальная память находится в динамической памяти с произвольным доступом (DRAM) устройства. Из приведенного в предыдущих пунктах кода понятно, что доступ ней можно получить как с хост-машины, так и с устройства. Глобальная память может быть объявлена как переменная в глобальной области со спецификатором __device__, однако при реализации программы применялось динамическое выделение глобальной памяти с помощью cudaMalloc(). Выделенная память существовала на протяжении всего необходимого для работы наших функций ядра времени и освобождалась с использованием cudaFree().

В процессе работы программы глобальная память может кэшироваться на чипе, а может и не кэшироваться - в зависимости от исполняемого программного кода и возможностей устройства.

Устройство пытается объединять выпускаемые потоками варпа запросы на чтение и запись в глобальную память так, чтобы минимизировать количество транзакций, сокращая требования к пропускной способности. В тех же случаях, когда параллельные потоки одновременно обращаются к адресам памяти, которые находятся далеко друг от друга в физической памяти, у устройства нет возможности объединять эти запросы, что приводит к резкому снижению эффективной полосы пропускания [115].

В нашем случае вычисления производятся с применением двумерных массивов значительного размера и при классической схеме адресации элементов таких массива избежать падения производительности невозможно.

Одним из способов повышения производительности в таких случаях является использование разделяемой памяти [294]. В таком способе фрагмент многомерного массива, находящегося в глобальной памяти, переносится в разделяемую память, и вычислительные потоки работают с данными, находящимися в разделяемой, а не глобальной памяти. В отличие от глобальной памяти, разделяемая память встроена в сам мультипроцессор (on-chip) и работает намного быстрее, чем глобальная память. Доступ к различным областям разделяемой памяти не приводит к снижению производительности, поэтому вычислительная эффективность решения может быть повышена.

Физически разделяемая память реализуется несколькими модулями памяти одинакового размера, к которым можно обращаться одновременно. Таким образом, если операции с памятью затрагивают несколько модулей памяти, такие операции могут быть выполнены быстрее за счет одновременного выполнения обращений к различным модулям.

Если же к одному модулю разделяемой памяти параллельно обращаются несколько различных потоков, такие обращения сериализуются. В то же время, если все потоки варпа обращаются по одному и тому же адресу разделяемой памяти, производится широковещательная рассылка запрашиваемых данных.

При написании программы в том случае, когда необходимый объем разделяемой памяти известен во время компиляции программы, можно явно объявить массив нужного размера с использованием спецификатора __shared__. Если объем разделяемой памяти становится известен лишь во время запуска функции ядра, массив объявляется без указания размера с использованием спецификаторов extern и __shared__, а размер передается в качестве дополнительного параметра конфигурации при запуске функции ядра.

Как уже говорилось ранее, разделяемая память выделяется для каждого блока потоков, и все потоки в блоке имеют доступ к одной и той же разделяемой памяти. Время жизни разделяемой памяти соответствует времени жизни блока потоков. Каждый блок потоков имеет свою область разделяемой памяти. По умолчанию размер разделяемой памяти для блока составляет 48 КБ, что существенно меньше доступного объема глобальной памяти.

B.3.2. Причины низкой эффективности базовой реализации методов формирования признаков на GPU

Для того, чтобы повысить эффективность представленного в предыдущем подразделе решения, рассмотрим более подробно, как производится взаимодействие с памятью при выполнении функции ядра, входящей в основной цикл оптимизации. Схематично такое взаимодействие, относящееся к наиболее затратной в вычислительном плане операции, а именно, расчету расстояний во входном многомерном пространстве, показано на рисунке B.3.

Как видно из рисунка, когда все потоки варпа idx=0..31 начинают исполнять цикл по k для перебора индексов случайной выборки dApproxBatch, и k=0, все потоки варпа запрашивают компонент c=0 точек с индексами 0, 1, ...31, а также соответствующий компонент c=0 точки с индексом dApproxBatch[0] (отмечены на рисунке сплошным зеленым цветом). Далее все потоки варпа запрашивают компонент c=1 точек с индексами 0, 1, ...31, а также соответствующий компонент c=1 точки с индексом dApproxBatch[0] (отмечены на рисунке диагональной зеленой текстурой) и далее до получения компонента c=dx-1.

При k = 1 процесс повторяется: все потоки варпа запрашивают компонент c=0 точек с индексами 0, 1, ...31, а также соответствующий компонент c=0 точки с индексом dApproxBatch[1] (компоненты вектора отмечены полупрозрачной розовой текстурой) и т.д.

Компоненты точек с индексами 0 и 31 находятся друг от друга на расстоянии 31*dx*sizeof(FTYPE)=27776 байт друг от друга в адресном пространстве (при dx=224), они редко оказываются выровнены по границе отличной от 4 байт. Компоненты точки с индексом dApproxBatch[k] могут быть от запрашиваемых компонент точек 0...31 на произвольно большом расстоянии, ограниченном лишь размерностью dx и количеством точек np.

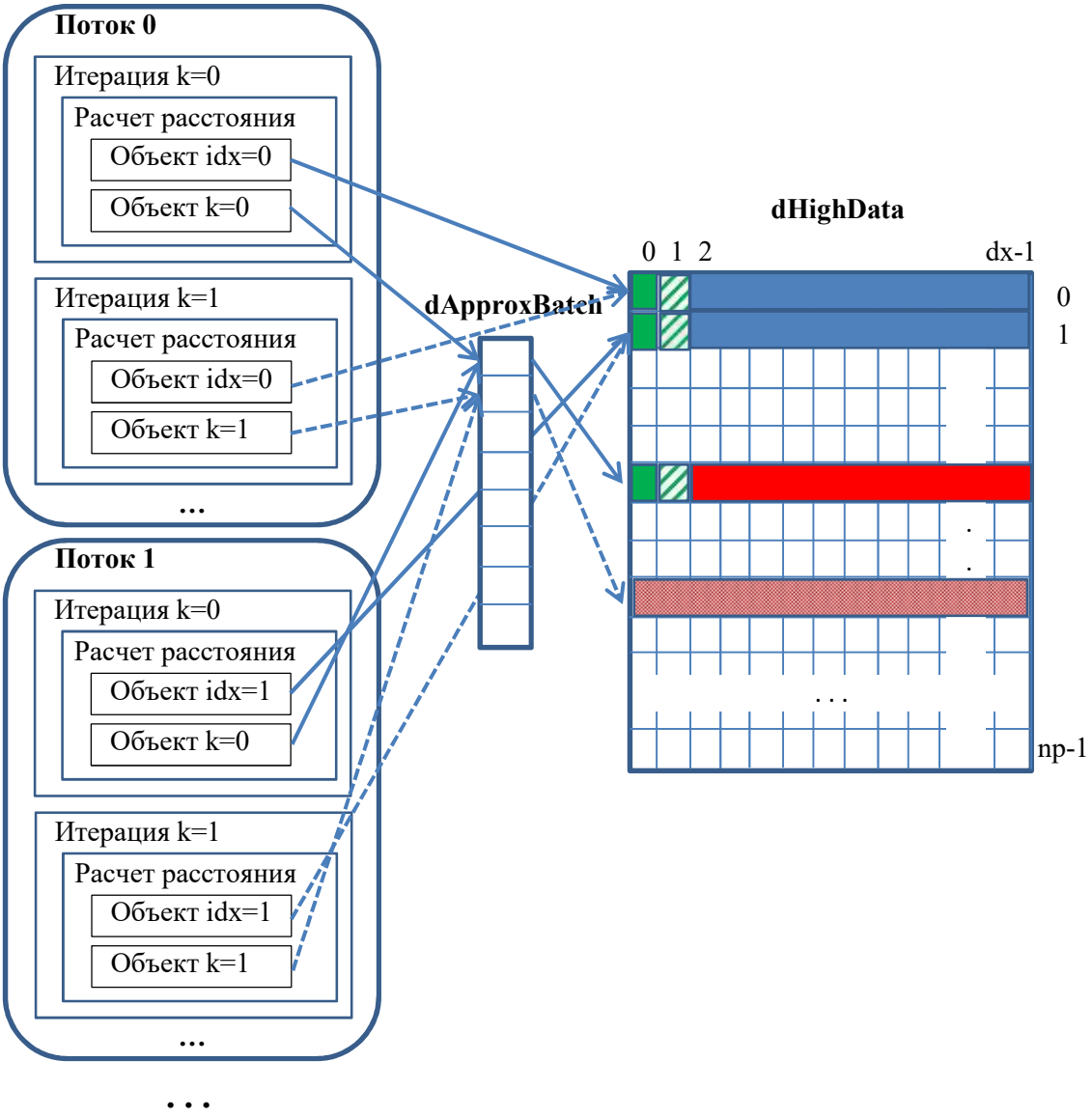


Рисунок В.3 - Взаимодействие с памятью при расчете расстояний во входном многомерном пространстве

Таким образом, устройство не может объединить существенную часть запросов к памяти, что приводит к существенному падению производительности, которое можно преодолеть за счет описанных ниже модификаций.

В.3.3. Повышение эффективности с использованием комбинированной блочно-столбцовой схемы организации памяти

Для повышения эффективности взаимодействия с памятью, будем действовать в нескольких направлениях. Во-первых, изменим схему хранения данных о координатах точек в пространстве высокой размерности так, чтобы соответствующие

компоненты векторов, обрабатываемых потоками одного варпа, оказывались рядом. Для этого будем организовывать массив так, как это показано на рисунке В.4.

На рисунке массив разделен на непересекающиеся блоки, размером 32*dx элементов, содержащие информацию о входных координатах для последовательных наборов по 32 точки. Каждый такой блок, таким образом, будет обрабатываться потоками одного варпа.

Данные внутри блока, предназначенного для выполнения в рамках одного варпа, расположены не по классической построчной схеме, а постолбцово, так чтобы соответствующие координаты точек, которые обрабатываются одновременно потоками одного варпа, располагались компактно в смежных областях памяти. Каждая строка показанного на рисунке массива соответствует какой-либо координате всех 32 точек блока.

Теперь, когда все потоки варпа idx=0..31 начинают исполнять очередную итерацию по переменной k, они запрашивают компонент c=0 точек с индексами 0, 1, ...31 (строка показана на рисунке синим цветом), после чего все потоки варпа запрашивают компонент c=1 точек с индексами 0, 1, ...31 (строка показана на рисунке зеленым цветом) и далее до получения компонента c=dx-1. Указанные запросы при предложенной организации памяти, удовлетворяются быстрее, чем при классической организации массива, за счет объединения запросов на чтение глобальной памяти. Кроме того, загружаемые данные могут кэшироваться, при этом помимо данных текущей обрабатываемой компоненты c (размером 32*sizeof(FTYPE)=128 байт) в кэше оказываются и данные следующей обрабатываемой компоненты c+1, что приводит к сокращению обращений на чтение внешней памяти и повышению производительности.

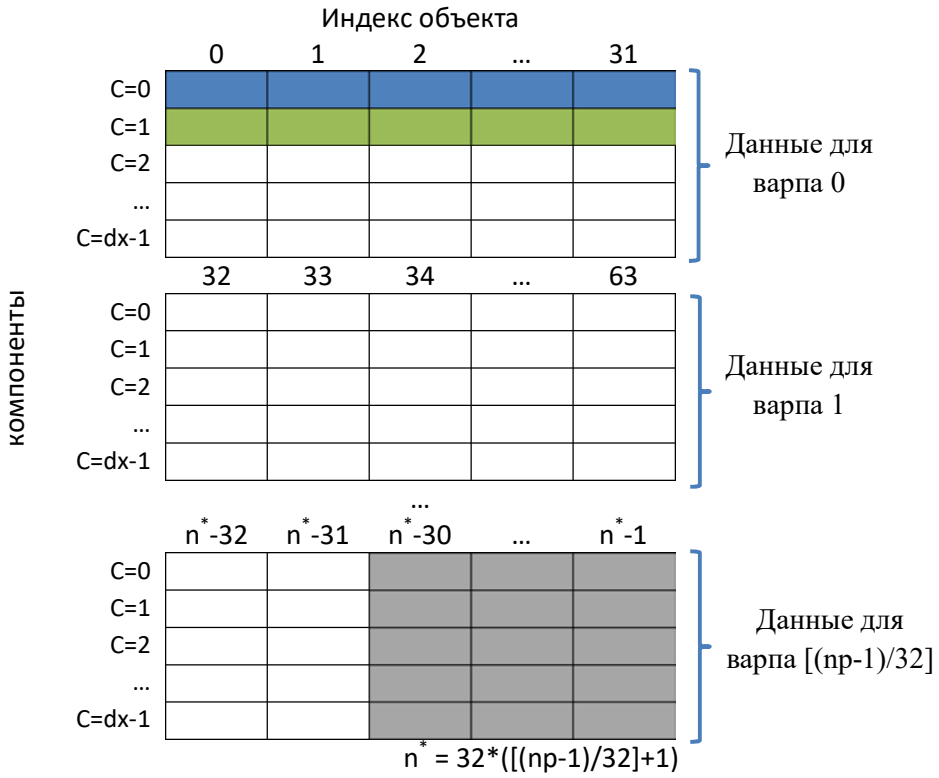


Рисунок В.4 - Организация массива координат точек в исходном пространстве

К недостаткам предложенной схемы организации можно отнести:

- необходимость перестановки элементов массива на этапе подготовки данных,
- дополнительный расход памяти в случае, когда число точек не кратно размеру варпа.

Указанные недостатки не являются существенными, так как указанная перестановка производится однократно на хост-машине (если данные не используются другими алгоритмами, они могут и сразу загружаться в массив с указанной структурой). Дополнительный расход памяти можно оценить следующим образом: (32 — np%32)%32*dx*sizeof(FTYPE), где % - операция взятия остатка от деления. Таким образом, в худшем случае будет израсходовано 31*dx*sizeof(FTYPE) = 27776 байт (расчет сделан для размерности пространства dx=224), что незначительно на фоне объема памяти современных графических процессоров (гигабайты и десятки гигабайт).

B.3.4. Копирование случайной выборки с построчной организацией данных и асимметричной реализацией функции расстояния

При предложенной выше схеме организации оптимизируется доступ потоков варпа к векторам, представляющим собой первый аргумент функции расчета расстояния, однако доступ к векторам, соответствующим второму аргументу, остается неоптимальным. Действительно, хотя на каждой итерации k все потоки варпа фактически обращаются к одной точке dApproxBatch[k], кэш работает неэффективно, т.к. при запросе некоторой компоненты c в кэш попадают большей частью не последующие компоненты c+1, c+2 … точки dApproxBatch[k], а соответствующие компоненты соседних точек dApproxBatch[k]+1, dApproxBatch[k]+2, …

Для борьбы с указанным недостатком предлагается комбинация предложенной выше блочно-столбцовой схемы организации памяти с классической построчной схемой (см. рисунок B.5). Программная реализация функции расстояния при этом будет носить ассиметричный характер – ее первый аргумент будет расположен в памяти по столбцам, а второй – по строкам.

Для исключения полного дублирования данных классическая построчная схема организации будет применяться лишь для небольшой доли точек, входящих в состав случайных подвыборок для расчета градиента. Теперь при первом обращении к некоторой компоненте c вектора из случайной подвыборки, в кэш попадут другие соседние с ней компоненты запрашиваемого вектора, а не компоненты соседних векторов.

Предложенная модификация требует использования дополнительной памяти. Так, при размере случайной выборки в BATCH_SIZE точек потребуется BATCH_SIZE*dx*sizeof (FTYPE) байт. Более существенным недостатком является необходимость копировать BATCH_SIZE точек на устройство на каждой итерации, что отрицательно влияет на производительность. По этой причине можно ожидать, что эффективным использование такой схемы организации будет лишь при достаточно большом количестве точек и размерности данных.

На представленном рисунке B.5 синими стрелками показано обращение к данным, организованным по блочно-столбцовой схеме, красными – по классической

построчной схеме. Сплошными линиями показаны обращения к векторам на итерации k=0, пунктирными – на итерации k=1.

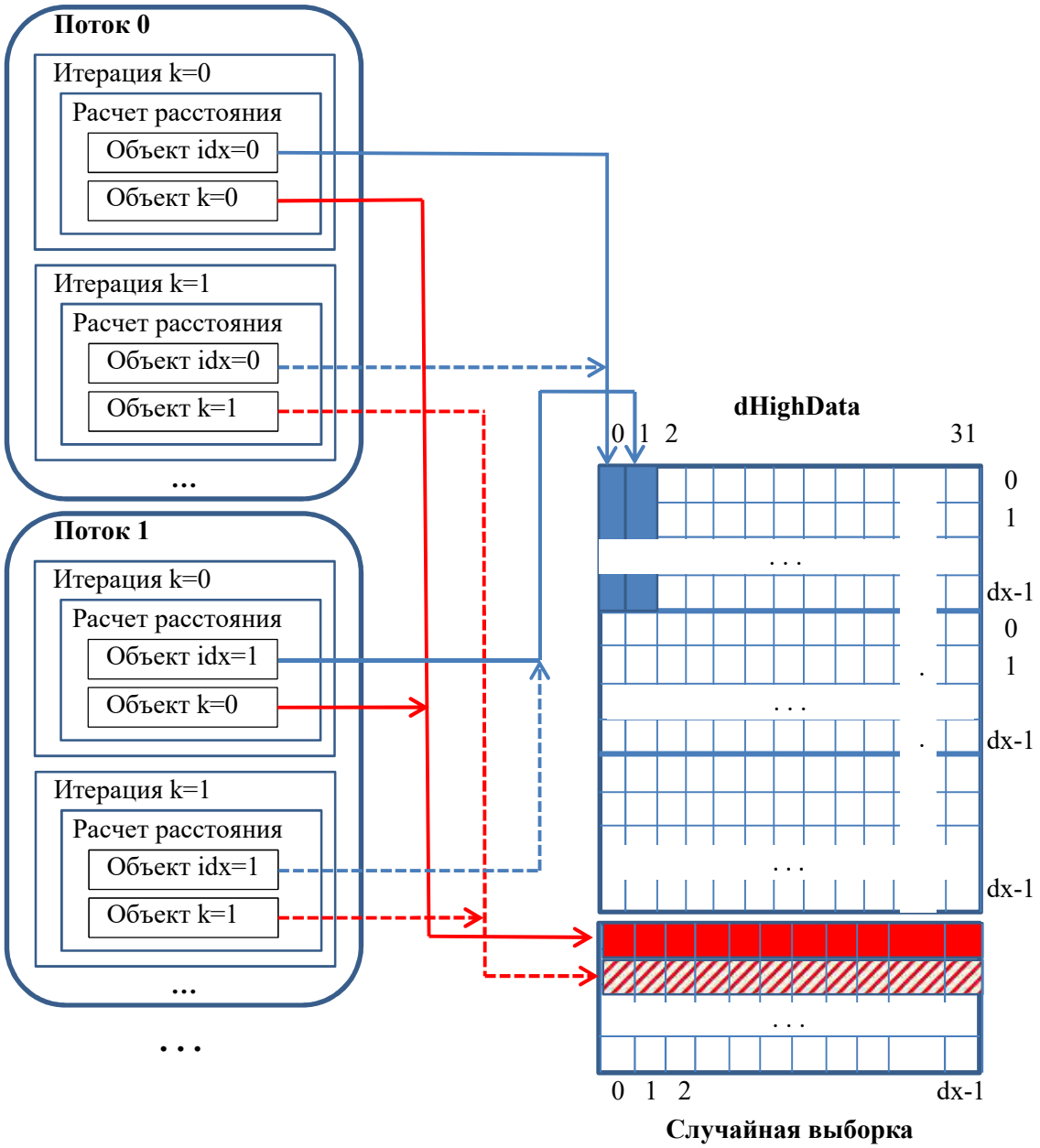


Рисунок В.5 - Комбинация блочно-столбцовой схемы организации памяти с классической построчной организацией

B.3.5. Использование разделяемой памяти

Вторым примененным способом повышения производительности стало использование разделяемой памяти для сохранения вектора градиента.

Хотя автоматическая переменная, объявленная в коде устройства без спецификаторов __device__, __shared__ или __constant__, при выполнении кода на устройстве обычно располагается в регистре, в ряде случаев компилятор размещает переменную в локальной памяти потока. В частности, такое наблюдается для массива FTYPE y[MAX_DY]; .

В этом случае фактически локальная переменная располагается в глобальной памяти устройства и доступ к ней становится неэффективным. По этой причине имеет смысл разместить переменную в разделяемой памяти. Для этого функция ядра была изменена так, как показано ниже:

```
template <class InputDis, class OutputDis>
GLOBAL void kernel2_base( const FTYPE* dLowData1,
                          FTYPE *dLowData2,
                          FTYPE m,
                          const FTYPE *dHighData,
                          const int* dApproxBatch,
                          int Start, int Stop,
                          bool initbyzero,
                          int np,
                          DPARAMS pars,
                          int dy )
{
    int idx = blockDim.x * blockIdx.x + threadIdx.x;
    __shared__ FTYPE shared_y[BLOCK_SIZE * MAX_DY];
    FTYPE* y = shared_y + threadIdx.x * MAX_DY;

    // …
```

Здесь мы создаем массив shared_y в разделяемой памяти для хранения векторов градиента всех потоков блока. При размере блока BLOCK_SIZE и максимальной предполагаемой размерности выходного пространства MAX_DY вектор градиента потока с индексом threadIdx.x внутри блока будет располагаться по смещению threadIdx.x * MAX_DY , выраженному в элементах типа FTYPE от начала массива shared_y.

При таком размещении векторов градиента исключаются конфликты доступа к памяти, и отсутствует необходимость синхронизации, так как каждый поток обращается к своей части массива в разделяемой памяти.

Стоит отметить, что модификации алгоритма, связанные с попытками повышения эффективности доступа к выходным массивам, вносят меньший вклад в общую производительность ввиду существенно меньшей размерности формируемого выходного пространства.

B.3.6. Развертка циклов

Еще одним важным способом ускорения является развертка циклов. Управлять процессом развертки в коде можно с использованием директивы #pragma unroll.

Применение развертки в коде, исполняемом на устройстве, позволяет освободить регистры вследствие использования в коде предрассчитанных на этапе компиляции значений, а также повысить эффективность использования памяти путем оптимизации доступа по предварительно рассчитанным адресам и смещениям.

B.4. Характеристики графических процессоров

Характеристики использованных в исследовании графических процессоров приведены в таблице B.1. Данные взяты из [89].

Таблица В.1 – Характеристики использованных в исследовании графических процессоров

Модель	**GTX 1070**	**RTX 2070**	**RTX 3090**
Архитектура	Pascal (2016−2021)	Turing (2018−2021)	Ampere (2020−2022)
Графический процессор	Pascal GP104	Turing TU106	Ampere GA102
Дата выхода	6 мая 2016	27 августа 2018	24 сентября 2020
Количество потоковых мультипроцессоров	15	36	82
Количество ядер CUDA	1920	2304	10496
Частота ядра	1506 МГц	1410 МГц	1400 МГц
Технологический процесс	16 нм	12 нм	8 нм
Тип памяти	GDDR5	GDDR6	GDDR6X
объём памяти	8 Гб	8 Гб	24 Гб
Ширина шины памяти	256 бит	256 бит	384 бит
Пропускная способность памяти	256 Гб/с	448.0 Гб/с	936.2 Гб/с
Теоретическая производительность (float)	6,463 GFLOPS	7,465 GFLOPS	35.58 TFLOPS

Приложение Г.
Акты об использовании результатов диссертационной работы

УТВЕРЖДАЮ
Первый проректор – проректор
по науке Самарского университета
_____ /А.И. Розенцвайг
«23» апреля 2026 г.

АКТ

об использовании результатов диссертационного исследования
Е.В. Мясникова

Настоящим подтверждаем, что результаты диссертационного исследования по теме «Формирование информативных признаков гиперспектральных изображений на основе функций расстояния», полученные в Научно-исследовательском институте акустики машин (НИИ-201) Самарского университета, использованы при выполнении научно-исследовательского проекта «Фундаментальные проблемы разработки аэрокосмических транспортных систем и управления в аэрокосмической технике для обеспечения связанности территории Российской Федерации» № 75-15-2024-558 (2024 – 2026 гг.) в рамках государственной программы Российской Федерации «Научно технологическое развитие Российской Федерации».

Разработанные при выполнении диссертационного исследования методы и алгоритмы использованы при выполнении п.4.9 «Создание комплекса методов и алгоритмов совмещения, повышения разрешения и реконструкции разнородных и/или разновременных данных ДЗЗ», п. 4.18 «Создание комплекса методов и алгоритмов распознавания объектов на данных ДЗЗ с использованием разнородных и/или разновременных данных ДЗЗ» и п. 4.27 «Создание комплекса методов и алгоритмов тематической интерпретации и понимания данных ДЗЗ с использованием разнородных и/или разновременных данных ДЗЗ» Технического задания на выполнение проекта.

Научный руководитель проекта
академик РАН, д.т.н., проф. Шахматов Е.В.

Директор НИИ-201, д.т.н., проф. Крючков А.Н.

УТВЕРЖДАЮ
Первый проректор – проректор
по науке Самарского университета
_____ /А.И. Розенцвайг
«__» _____ 2026 г.

АКТ

об использовании результатов
диссертационного исследования Е.В. Мясникова

Результаты диссертационного исследования по теме «Формирование информативных признаков гиперспектральных изображений на основе функций расстояния», выполненного в научно-исследовательской лаборатории геоинформатики и информационной безопасности (НИЛ-55) Самарского университета, были использованы при выполнении НИОКР по договору №77/24 от 24.06.2024 на тему «Технологии автоматической геопривязки, совмещения и пространственно-временного оперативного анализа данных дистанционного зондирования Земли», шифр 055х-336 (заказчик - ФГБУ ВО «Московский государственный университет геодезии и картографии» МИИГАиК, г. Москва).

Применение предложенных в диссертационном исследовании алгоритмических средств формирования информативных признаков гиперспектральных изображений, основанных на принципах сохранения попарных расстояний, учета пространственного контекста и слияния разнородной признаковой информации, а также алгоритмические средства ускорения обработки позволили повысить эффективность решения задач обработки изображений ДЗЗ по сравнению с известными методами.

Ответственный исполнитель НИР
с.н.с., к.т.н., Чернов А.В.

Исполнительный директор НИЛ-55 Бирюков П.В.

УТВЕРЖДАЮ
Проректор по учебной работе
_____ А.В. Гаврилов
«__» _____ 2026 г.

АКТ

об использовании результатов
диссертационного исследования в учебном процессе

Настоящим подтверждаем, что результаты диссертационного исследования Е.В. Мясникова по теме «Формирование информативных признаков гиперспектральных изображений на основе функций расстояния», выполненного на кафедре геоинформатики и информационной безопасности (ГИиИБ) Самарского университета, использованы в учебном процессе на кафедре ГИиИБ на основании решения кафедры протокол № 6 от 16.03.2026.

Указанные результаты включены в курсы «Машинное обучение и распознавание образов», «Нейронные сети и глубокое обучение» специалитета 10.05.03 Информационная безопасность автоматизированных систем.

Заведующий кафедрой
геоинформатики и информационной
безопасности
_____ В.А. Федосеев
«03» апреля 2026 г.

Начальник учебно-методического
управления
_____ Н.В. Соловова
«8» апреля 2026 г.

Начальник отдела сопровождения
научных исследований
_____ Л.В. Родионов
«8» апреля 2026 г.

САМИС

АО «Самара-Информспутник»
проспект Карла Маркса, д.192, офис 717,
Самара, 443080
тел. +7 (846) 255-63-71
e-mail: mail@geosamara.ru
https://geosamara.ru
ОКПО 21154996 ОГРН 1036300444223
ИНН 6315559354 КПП 631601001

УТВЕРЖДАЮ
Заместитель генерального директора
АО «Самара-Информспутник»
_____ Н.И. Глумов
«15» апреля 2026 г.

Акционерное общество «Российская аэрокосмическая корпорация г. Самара» "Самара-Информспутник" ОГРН 1036300444223

АКТ

об использовании результатов диссертационной работы Е.В. Мясникова
«Формирование информативных признаков гиперспектральных
изображений на основе функций расстояния»

Комиссия в составе начальника отдела эксплуатации вычислительных систем и защиты информации, к.т.н. В. Н. Копенкова и ведущего инженера-математика, к.т.н. М. А. Чичевой, рассмотрев диссертацию Е. В. Мясникова, представляемую на соискание ученой степени доктора технических наук по специальности 2.3.8 – Информатика и информационные процессы, подтверждает, что разработанные в диссертационной работе методы, алгоритмы и программные модули были использованы при выполнении следующих работ: договор № 235 от 01.04.2009 с ФГУП "ГНПРКЦ "ЦСКБ-ПРОГРЕСС" (г. Самара), тема «Разработка предложений по технологии моделирования, обработки и распознавания многоспектральных изображений», шифр "Плеск-МС"; договор № 608 от 12.07.2023 с ФГБУ ВО «Московский государственный университет геодезии и картографии» (МИИГАиК, г. Москва), НИОКР «Технологии составления цифровых профилей объектов и цифровой модели пространства на основе оперативного получения и совмещения данных из различных источников («робот-картограф»)».

Применение разработанных методов, алгоритмов и программных средств позволило повысить эффективность решения задач обработки изображений по сравнению с известными методами.

Ведущий инженер-математик, М.А. Чичева
к. т. н.

Начальник отдела эксплуатации вычислительных систем
и защиты информации, В.Н. Копенков
к. т. н.

УТВЕРЖДАЮ

Руководитель Института систем
обработки изображений РАН – филиала
ФНИЦ «Кристаллография и фотоника» РАН,
д.ф.-м.н.
Н.Л.Казанский
«21» декабря 2023 г.

АКТ

об использовании результатов диссертационного исследования Е.В. Мясникова в Институте систем обработки изображений Российской академии наук – филиале Федерального научно-исследовательского центра «Кристаллография и фотоника» Российской академии наук

Разработанные Е.В. Мясниковым в процессе выполнения диссертационного исследования методы формирования компактных информативных признаков на основе различных функций спектрального рассогласования и учета локального пространственного контекста многоспектральных изображений дистанционного зондирования Земли (ДЗЗ) были использованы при выполнении научно-исследовательской работы по государственному заданию по теме «Разработка методов интеллектуального анализа и криптозащиты изображений в задачах обработки данных дистанционного зондирования Земли», номер темы 0026-2021-0014.

Эффективность разработанных Е.В. Мясниковым методов верифицирована вычислительными экспериментами на изображениях ДЗЗ.

Руководитель темы,
г.н.с. ИСОИ РАН, д.т.н. В.В. Сергеев

Ученый секретарь ИСОИ РАН,
д.ф.-м.н. В.В. Котляр

Экз. № __

УТВЕРЖДАЮ
Генеральный директор
ФГУП «ГосНИИПП»
канд. техн. наук

С.А. Егоров
«19» июня 2026 г.

АКТ

об использовании результатов диссертационной работы
Мясникова Евгения Валерьевича
«Формирование информативных признаков гиперспектральных изображений на основе функций расстояния» в Федеральном государственном унитарном предприятии «Государственный научно-исследовательский институт прикладных проблем»

Материалы диссертационной работы Мясникова Е.В. были использованы в ФГУП «ГосНИИПП» в рамках составной части научно-исследовательской работы, выполненной АО «Самара-Информспутник» по договору № 4/2021 от 22 сентября 2021 года.

Автореферат диссертационной работы Мясникова Е.В. на соискание ученой степени доктора технических наук обсужден на заседании секции «Д» научно-технического совета (протокол № 18 от 19 июня 2026 года).

Заместитель генерального директора
по научной работе, канд. воен. наук К.В. Миночкин