Российская Академия Наук

Автоматика и телемеханика

Журнал основан в 1936 году
Выходит 12 раз в год

10
октябрь

Москва 2022

Учредители журнала:
Отделение энергетики, машиностроения, механики и процессов управления РАН,
Институт проблем управления им. В.А. Трапезникова РАН (ИПУ РАН),
Институт проблем передачи информации им. А.А. Харкевича РАН (ИППИ РАН)

Адрес редакции: 117997, Москва, Профсоюзная ул., 65
Тел./факс: (495) 334-87-70
Электронная почта: redacsia@ipu.ru

Зав. редакцией *Е.А. Мартехина*

Москва
ООО «Тематическая редакция»

Тематический выпуск

ВСТУПИТЕЛЬНОЕ СЛОВО ПРОГРАММНОГО КОМИТЕТА КОНФЕРЕНЦИИ «МАТЕМАТИЧЕСКИЕ МЕТОДЫ РАСПОЗНАВАНИЯ ОБРАЗОВ»

DOI: 10.31857/S0005231022100014, **EDN:** AJSYNX

В тематическом выпуске представлены избранные статьи 20-й Всероссийской конференции с международным участием «Математические методы распознавания образов» (ММРО), прошедшей с 7 по 10 декабря 2021 г. в Москве.

Конференция впервые прошла в 1983 г. и с тех пор проводится один раз в два года, что позволяет считать ее старейшим российским форумом в области интеллектуального анализа данных, машинного обучения, искусственного интеллекта, включающим как теоретические, так и прикладные сферы указанных областей.

Конференция ММРО неразрывно связана с именами ее создателей — двух выдающихся ученых академика РАН Юрия Ивановича Журавлева (1935–2022 гг.) и его ученика академика РАН Константина Владимировича Рудакова (1954–2021 гг.). Их вклад в развитие математических методов классификации, распознавания образов, прогнозирования и методов машинного обучения — основных тематик конференции ММРО — трудно переоценить.

Ю.И. Журавлев организовал первую конференцию ММРО в Звенигороде и был бессменным руководителем организационного комитета всех следующих конференций, которые проходили в Дилижане (1985), Звенигороде (1983, 1991, 1993, 2001, 2005), Казани (2013), Львове (1987), Москве (2019, 2021), Петрозаводске (2011), Пущино (1995, 2003), Риге (1989), Санкт-Петербурге (2007), Светлогорске (2015), Суздале (2009), Таганроге (2017), Тверской области (1997, 1999).

К.В. Рудаков в 1983 г., тогда еще будучи молодым кандидатом физико-математических наук, участвовал в организации первой конференции ММРО. Подготовка всех последующих конференций также проводилась при активном участии Константина Владимировича. В 2001 г. был создан программный комитет конференции, а К.В. Рудаков стал его председателем. Программным комитетом ММРО в 2021 г. руководили профессор РАН Константин Вячеславович Воронцов и доктор физико-математических наук Вадим Викторович Стрижов, также отдавшие десятки лет организации конференции.

На протяжении почти четырех десятилетий работы конференции организаторы включали в тематику актуальную научную повестку, сохраняя при этом неизменным требования к глубокой математической проработке представляемых результатов. Отдавалось предпочтение исследованиям, направленным на развитие фундаментального теоретического аппарата их получения и оценки. В первые годы конференция была в большей степени ориен-

Рис. 1. Ю.И. Журавлев и К.В. Рудаков на 12-й конференции ММРО в 2005 г. (Звенигород, Московская область).

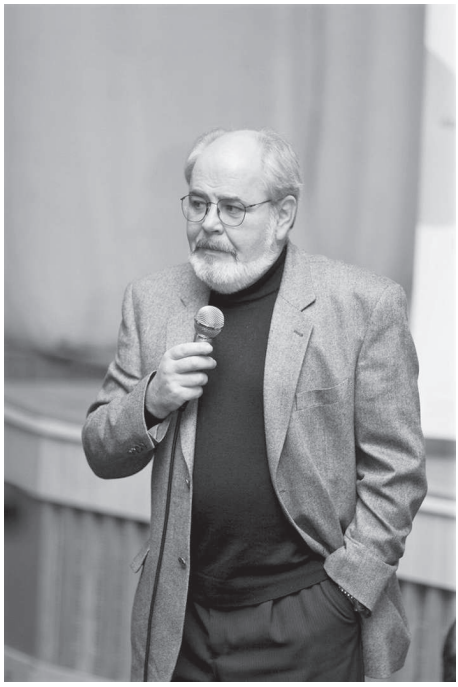


Рис. 2. К.В. Рудаков ведет заседание на 12-й конференции ММРО в 2005 г. (Звенигород, Московская область).

Рис. 3. Ю.И. Журавлев комментирует выступления на 12-й конференции ММРО в 2005 г. (Звенигород, Московская область).

Рис. 4. Выступление Юрия Ивановича Журавлева с пленарным докладом на 19-й конференции ММРО. По информации авторов — последнее выступление Ю.И. Журавлева с научным докладом.

тирована на проблематику задач распознавания образов, классификации и регрессии, имеющую отправную точку в анализе изображений и сигналов и опирающуюся на алгебраические и статистические методы. Нельзя не упомянуть научную дискуссию по обсуждению статистической теории надежности алгоритмов Вапника–Червоненкиса с участием одного из ее авторов Владимира Наумовича Вапника, развернувшуюся на первой конференции ММРО в 1983 г.

На конференции уделялось внимание комбинаторному подходу к оценке надежности алгоритмов, методам прогнозирования временных рядов, методам оптимизации, задачам и методам анализа медицинских данных, биоинформатике, экспертным системам. Затем тематика расширилась на область интеллектуального анализа данных (data mining) и машинного обучения (machine learning). В последние годы произошли значительные изменения, связанные в первую очередь с бурным развитием методов глубокого обучения (deep learning), методов обработки текстов на естественных языках (natural language processing), методов анализа больших разнородных данных (big data), области компьютерного зрения (computer vision).

При этом конференция не оставляла без внимания и решение прикладных задач. В последние годы растет интерес к ММРО представителей самых разных направлений российской и зарубежной IT-индустрии.

На конференции ММРО уделялось значительное внимание участию молодых ученых. ММРО дала им возможность узнать о результатах новых исследований, получить представление о тенденциях развития области и, конечно, впервые представить свои результаты научному сообществу. Для многих активно работающих ученых конференция ММРО стала отправной точкой в их научной карьере.

В 2021 г. конференция ММРО проводилась в смешанном формате в Вычислительном центре им. А.А. Дородницына Федерального исследовательского центра РАН. В рамках открытия было проведено мемориальное заседание, посвященное Константину Владимировичу Рудакову, на котором выступали ученики и близкие коллеги К.В. Рудакова. Всего в конференции приняли участие 215 человек, было сделано 105 докладов по следующим основным научным направлениям:

- Интеллектуальный анализ данных
- Нейронные сети и глубокое обучение
- Методы оптимизации для интеллектуального анализа данных
- Вычислительная сложность и приближенные методы
- Обработка и анализ изображений и сигналов, компьютерное зрение
- Информационный поиск и анализ текстов
- Анализ данных веба и социальных сетей
- Индустриальные приложения науки о данных
- Анализ биомедицинских данных, биоинформатика
- Методы математического моделирования в интеллектуальном анализе данных
- Интеллектуальный анализ геопространственных данных

• Интеллектуальная оптимизация и эффективный менеджмент.

По результатам обсуждения сделанных докладов были отобраны работы для публикации в специальном выпуске журнала «Автоматика и телемеханика» № 10 от 2022 г. В настоящем 10-м номере журнала представлены работы, посвященные распознаванию изображений, клонированию и конверсии голоса, реконструкции поверхности для движения марсохода и ряду фундаментальных математических задач.

Значительная часть работ посвящена тематике распознавания изображений. В работе Е.Ю. Минаева, Л.А. Жердевой и В.А. Фурсова «Визуальная одометрия по изображениям опорной поверхности с малыми межкадровыми поворотами» показывается решение задачи визуальной одометрии по последовательности видеокадров, которые формируются с использованием камеры, направленной перпендикулярно вниз на опорную поверхность. В работе А.С. Маркова, Е.Ю. Котлярова, Н.П. Аносовой, В.А. Попова, Я.М. Карандашева и Д.Е. Апушкинской «Использование нейронных сетей для выявления аномалий на рентгеновских снимках, полученных на сканерах персонального досмотра» изучается выявление аномалий на рентгеновских снимках, показаны предварительные результаты использования нейронной сети для выделения аномалий. В работе Д.В. Свитова и С.А. Алямкина «Дистилляция моделей для распознавания лиц, обученных с применением функции Софтмакс с отступами» предлагается метод дистилляции, который использует центры классов сети-учителя для инициализации сети-ученика, а затем сеть-ученик обучается производить биометрические вектора, углы от которых до центров классов равны углам в сети-учителе. В работе А.И. Базаровой, А.В. Грабового и В.В. Стрижова «Анализ свойств вероятностных моделей в задачах обучения с экспертом» решается задача аппроксимации набора фигур на контурном изображении, вычисления проводятся на примере задачи аппроксимации радужной оболочки глаза на контурном изображении. В статье А.А. Захарова «Метод сопоставления изображений с использованием тепловых ядер на графах» представлен метод сопоставления изображений на основе тепловых ядер, который позволяет выделять на начальном этапе с помощью тепловых ядер на графах наиболее устойчивые особенности изображений для последующего сопоставления. В работе М. Горпинич, О.Ю. Бахтеева и В.В. Стрижова «Градиентные методы оптимизации метапараметров в задаче дистилляции знаний» предлагается обобщение задачи дистилляции на случай оптимизации метапараметров градиентными методами, предложенный подход проиллюстрирован с помощью вычислительного эксперимента на выборках CIFAR-10 и Fashion-MNIST, а также на синтетических данных.

Две работы данного номера посвящены исследованию не менее интересных практических задач. В работе Д.С. Обухова «Клонирование и конверсия произвольного голоса с использованием генеративных потоков» предложен подход на основе потоковых генеративных моделей, который позволяет решать задачи клонирования голоса за счет использования полученных из внешней системы вещественных векторов фиксированной размерности, содержащих информацию о спикере, благодаря которому система синтезирует более естественную речь голосом, похожим на заданный целевой голос, как в задаче клонирования голоса, так и в задаче конверсии голоса. В статье

А.В. Бобкова и И. Дай «Методы 3D-реконструкции поверхности в задаче автономной навигации робота-марсохода» рассмотрены алгоритмы и методы трехмерной реконструкции поверхности, которые могут быть использованы для обеспечения автономной навигации робота-марсохода, приведены классификация методов, сравнение их свойств и оценка технической реализуемости. Также несколько статей в журнале посвящены фундаментальным математическим исследованиям. В работе Е.А. Карацубы «Быстрый алгоритм вычисления пси-функции» рассматривается быстрый алгоритм вычисления пси-функций, подробно исследуется механизм построения быстрого вычисления E-функций, приведены и доказаны соответствующие теоремы. В исследовании А.Ю. Горнова, А.С. Аникина, Т.С. Зароднюк и П.С. Сороковикова «Модификация алгоритма доверительного бруса, основанного на аппроксимации главной диагонали матрицы Гессе, для решения задач оптимального управления» предложен подход решения задачи оптимального управления, основанный на использовании редукции к конечномерной задаче оптимизации с последующим использованием аппроксимации главной диагонали гессиана на примере задач оптимизации сепарабельных, квазисепарабельных функций и функций Розенброка–Скокова. В работе Н.А. Драгунова и Е.В. Дюковой «Об одном подходе к расшифровке монотонной логической функции» рассматривается задача расшифровки двузначной монотонной функции f, определенной на k-значном n-мерном кубе, предложен и исследован подход, основанный на применении асимптотически оптимального алгоритма дуализации над произведением k-значных цепей, выявлены условия применимости. В статье А.Н. Тырсина «Энтропийное моделирование сетевых структур» рассмотрены вопросы использования дифференциальной энтропии для сетевых структур, представленных в виде связных графов с корреляционными связями, предложены новые характеристики, которые расширяют возможности энтропийного моделирования для исследования сетевых структур. В статье З.М. Шибзухова «Об одной робастной схеме градиентного бустинга на основе агрегирующих функций, нечувствительных к выбросам» предложена новая робастная схема построения алгоритмов градиентного бустинга, основанная на применении дифференцируемых оценок среднего значения, нечувствительных или малочувствительных к выбросам, для задания робастного функционала эмпирического риска, которая позволяет находить искомую зависимость по данным, содержащим относительно большую долю выбросов.

Каждая рукопись прошла слепое рецензирование как минимум двумя рецензентами, была одобрена к публикации программным комитетом конференции и редколлегией журнала.

Воронцов К.В., Громов А.Н., Забежайло М.И., Инякин А.С.,
Лазарев А.А., Лемтюжникова Д.В., Соколов И.А.,
Стрижов В.В., Чехович Юл.В., Чехович Ю.В.

© 2022 г. Е.Ю. МИНАЕВ, канд. техн. наук (eminaev@gmail.com),
Л.А. ЖЕРДЕВА (lara.zherdeva.taskina@gmail.com)
(Самарский университет),
В.А. ФУРСОВ, д-р техн. наук (fursov@ssau.ru)
(Институт систем обработки изображений — филиал ФНИЦ
"Кристаллография и фотоника" РАН, Самара)

ВИЗУАЛЬНАЯ ОДОМЕТРИЯ ПО ИЗОБРАЖЕНИЯМ ОПОРНОЙ ПОВЕРХНОСТИ С МАЛЫМИ МЕЖКАДРОВЫМИ ПОВОРОТАМИ

В работе рассматривается задача визуальной одометрии по последовательности видеокадров, которые формируются с использованием камеры, направленной перпендикулярно вниз на опорную поверхность. Задача решается в предположении, что частота съемки велика, поэтому параметры межкадрового поворота и сдвига невелики. Технология реализуется в виде последовательности следующих этапов: определение сдвига и поворота с точностью до целого числа пикселей с использованием корреляционного метода, уточнение параметров сдвига и поворота методом оптического потока и коррекция ошибок оценивания, связанных с неравномерностью движения и флуктуациями расстояния камеры до опорной поверхности путем оценки отклонений локальных калибровочных характеристик от их средних значений. Приводятся результаты экспериментальных исследований технологии на тестовых траекториях, полученных путем моделирования движения аппарата по опорной поверхности.

Ключевые слова: визуальная одометрия, опорная поверхность, последовательность кадров, моделирование.

DOI: 10.31857/S0005231022100026, **EDN:** AJTRMC

1. Введение

Системы визуальной одометрии [1–4], в которых последовательность видеокадров формируется с использованием камеры, направленной перпендикулярно вниз (на опорную поверхность) в последние годы завоевывает все большую популярность [5–10]. Возможно, это связано с тем, что не всегда есть надежные ориентиры в окружающих сценах, в то время как опорная поверхность наблюдается непрерывно. Например, в работе [11] приведен пример удачной реализации системы, основанной на корреляционном методе.

В [12] авторы настоящей работы предложили технологию визуальной одометрии по последовательности изображений опорной поверхности, регистрируемых БПЛА с малой высоты. Технология включает три этапа: определение сдвига и поворота с использованием корреляции фрагментов (1), уточнение параметров сдвига и поворота методом оптического потока (2) и коррекция ошибок оценивания траекторий, связанных с неравномерностью движения и

флуктуациями высоты (3). Предполагалось, что камера установлена в гироподвесе и сохраняет ориентацию относительно глобальной системы координат в течение всего полета.

Эксперименты подтвердили эффективность предложенной технологии. Однако попытка реализации этой технологии в случае жестко закрепленной на корпусе аппарата камеры натолкнулась на принципиальные трудности. Дело в том, что в этом случае необходимо оценивать не только межкадровый сдвиг, но и параметры поворота. Для их определения обычно строят последовательные оценки матриц сдвига и поворота по соответствующим точкам соседних кадров. Для этого кадры должны иметь значительные перекрытия, что достигается увеличением частоты съемки.

Известны различные подходы к решению задачи определения матрицы сдвига и поворота. В частности, непрямые методы визуальной одометрии [13–15] используют анализ соответствующих ключевых точек. По координатам этих точек определяется фундаментальная матрица и затем формируется матрица сдвига и поворота. Известны также алгоритмы построения матрицы сдвига и поворота непосредственно по ключевым точкам. Этот подход в рамках технологии, основанной на использовании изображений опорной поверхности, иногда оказывается неработоспособным, так как эти изображения часто представляют собой некоторую текстуру без выделяющихся оригинальных ключевых точек.

В рамках прямых методов одометрии [16–18] параметры сдвига и поворота определяются путем анализа плотного поля соответствий между кадрами изображений. В рамках этого подхода наиболее популярным является метод экстремальной корреляции. Исследования [12] также подтвердили эффективность этого метода в задаче оценивания траектории аппарата, у которого ориентация камеры относительно глобальной системы координат не изменяется в течение всего времени движения.

В случае жесткого закрепления камеры на аппарате реализация этого метода существенно усложняется. В данном случае для каждой пары кадров нормированный коэффициент корреляции необходимо рассчитывать как для всех возможных сдвигов, так и для различных относительных углов поворота. При этом, если для повышения точности уменьшать шаг дискретизации, существенно возрастает вычислительная сложность, что ставит под сомнение возможность реализации такой системы оценки текущей траектории в реальном времени.

В настоящей работе предлагается технология визуальной одометрии, принципиальное отличие которой от системы, описанной в работе [12], состоит в реализации метода корреляции на этапе определения межкадровых сдвигов и поворотов. Идея состоит в том, что для определения сдвига и поворота используются два фрагмента изображения, находящиеся на значительном удалении от центра изображения. Размеры фрагментов и их расстояния от центра выбираются так, чтобы при максимально возможном относительном повороте разность координат соответствующих точек не превышала половины межпиксельного расстояния. Вид сверху на аппарат при его движении по

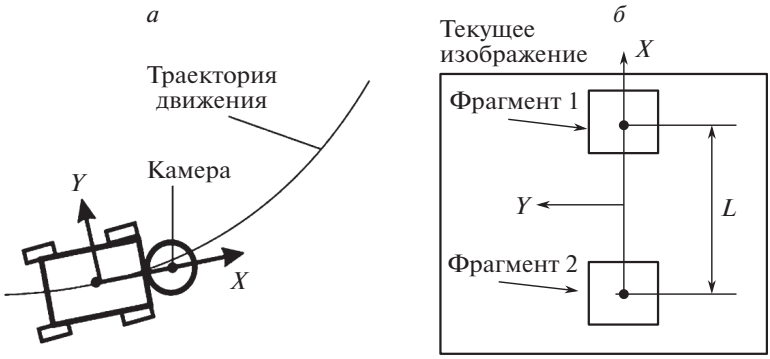


Рис. 1. Схема движения аппарата ($а$) и схема расположения фрагментов на кадре ($б$).

траектории и схема расположения фрагментов на одном кадре изображения показаны на рис. 1,$а$ и 1,$б$ соответственно.

В рамках описанной схемы съемки реализация основных этапов технологии, применявшихся в работе [12], существенно изменена. Теперь первый этап — экстремальной корреляции и второй этап — уточнение сдвигов методом оптического потока выполняются для каждого из указанных фрагментов отдельно. Появляются также новые этапы — оценка межкадрового и текущего накопленного углов поворота аппарата, а также формирование оценок траектории с использованием текущих оценок накопленного угла поворота. Кроме того, на завершающем этапе технологии коррекция оценок траектории осуществляется только по одной координате (OX) в направлении текущего вектора скорости аппарата (см. рис. 1,$а$). Подчеркнем, что указанные новые решения связаны с тем, что последовательность изображений формируется камерой, жестко связанной с корпусом аппарата.

Цель настоящего исследования состоит в разработке методов и алгоритмов, реализующих перечисленные этапы технологии, а также в экспериментальной проверке достижимой точности сквозной технологии в целом, включающей все указанные этапы.

2. Описание технологии

2.1. Основные предположения и обозначения

Рассматривается задача автономной навигации аппарата, движущегося по земной (далее опорной) поверхности, с использованием последовательности изображений этой поверхности. Изображения формируются камерой, направленной перпендикулярно вниз. Задача состоит в том, чтобы в каждой точке траектории определить относительный сдвиг и поворот соседних кадров. Поскольку расстояние от камеры до поверхности невелико, при определении межкадровых сдвигов и поворотов можно пренебречь проективными искажениями.

Для каждого кадра последовательности будем задавать локальную систему координат. Для определенности примем, что прямоугольная система ко-

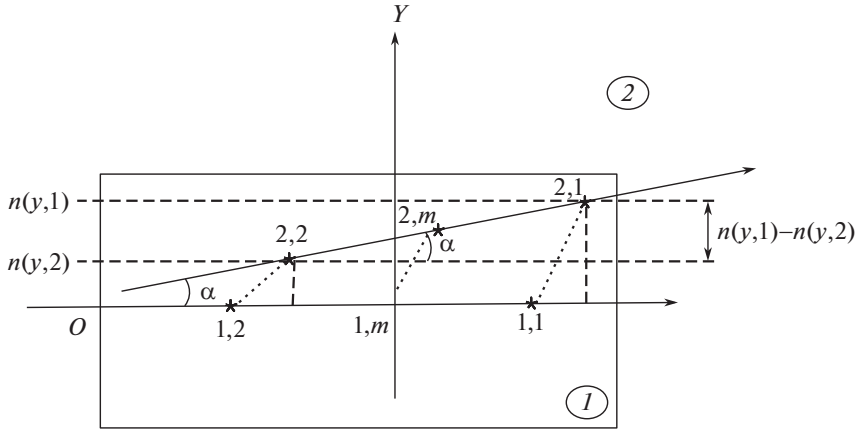


Рис. 2. Локальная система координат.

ординат связана с первым (по времени регистрации) изображением. Точка O (начало системы координат) находится в центре кадра, а положительное направление оси Ox совпадает с направлением движения аппарата (см. рис. 1). На рис. 2 показана схема взаимного расположения двух кадров изображений, обозначенных 1 и 2 (в кружках) в локальной системе координат, связанной с первым изображением. Первая цифра в обозначениях точек означает номер кадра, а вторая — номер фрагмента в соответствии с нумерацией, показанной на рис. 1. Например, 1.1 означает фрагмент 1 на первом кадре, 2.1 — фрагмент 1 на втором кадре и т.д. Центральные точки кадров изображений 1 и 2 обозначены «1.m» и «2.m» соответственно.

2.2. Алгоритм экстремальной корреляции

Используется двухпозиционный алгоритм корреляции по двум фрагментам с центрами в точках 1.1 и 1.2 на первом изображении и в точках 2.1 и 2.2 на втором изображении (см. рис. 2). Сдвиги определяются для пары фрагментов 1.1–2.1 и для пары 1.2–2.2.

Для того, чтобы обеспечить возможность применения стандартной процедуры корреляции на фрагментах, формулируется следующее требование. Координаты соответствующих точек фрагментов, используемых для вычисления нормированного коэффициента корреляции, должны различаться менее чем на половину межпиксельного расстояния при предельно возможных относительных межкадровых поворотах. Ясно, что это требование будет выполняться для всех точек фрагмента, если оно выполняется для точки наиболее удаленной от центра фрагмента. Если фрагмент квадратный с размерами $N \times N$, достаточно сформулировать это условие для одной угловой точки внешнего контура фрагмента.

На рис. 3 приведена схема взаимного положения угловых точек одноименных фрагментов на соседних изображениях. Эта схема иллюстрирует связь размеров фрагмента с максимально допустимым относительным углом поворота изображений. Здесь α — угол межкадрового поворота, l — дуга, соеди-

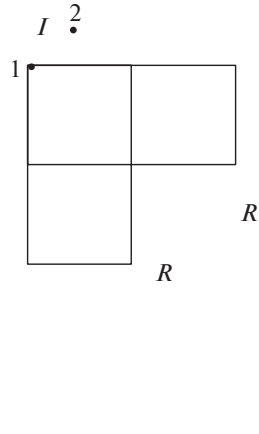


Рис. 3. К обоснованию размеров фрагментов.

няющая наиболее удаленные (угловые) точки (1 и 2) фрагментов соседних изображений; R — расстояния от центральной точки совмещенных фрагментов до точек 1 и 2.

Длина дуги l не может быть меньше расстояния между точками 1 и 2. Тогда полагая, что $\operatorname{tg}\alpha \cong \alpha$, для квадратной $N \times N$-области можно записать условие, при котором это расстояние будет меньше половины межпиксельного расстояния:

$$N \leq \frac{\Delta_{pix}}{2\sqrt{2} \cdot \alpha}, \tag{1}$$

где α — угол относительного поворота (в радианах), Δ_{pix}— масштабный коэффициент, зависящий от выбранной единицы измерения сдвига ($\Delta_{pix} = 1$, если сдвиг измеряется целым числом пикселей).

В таблице приведены предельные значения относительных поворотов для некоторых широко используемых размеров областей, используемых при вычислении нормированного коэффициента корреляции. Заметим, что указанные значения предельных углов обычно не создают проблем. Даже при сравнительно малой скорости регистрации 30 кадров в секунду и ограничении 3° угловая скорость аппарата на поворотах может достигать скорости 90 град/с, т.е. четверть полного оборота за одну секунду. При этом область корреляции 11×11 содержит 121 точку, что вполне достаточно для получения надежного результата.

Таким образом, первый этап технологии сводится к следующей последовательности шагов. На соседних кадрах изображений задаются два фрагмента (на рис. 2 центры этих фрагментов на первом кадре обозначены 1.1 и 1.2), разнесенные по возможности на достаточно большое расстояние L, при котором фрагменты «видны» на обоих изображениях. Размеры фрагментов задаются в соответствии с условием (1), при выполнении которого сдвиг одноименных пикселей, соответствующих фрагментов на двух видах (здесь 1 и 2) не превы-

Предельные углы относительных поворотов

Размеры фрагмента	Предельный угол (рад.)	Предельный угол (град.)
5×5	0,1750	10,0250
7×7	0,1173	6,7213
9×9	0,0882	5,0511
11×11	0,0589	3,3723

шает половины межпиксельного расстояния при максимальном возможном межкадровом повороте.

Далее задавая центры областей поиска в точках 2.1 и 2.2 второго кадра, с использованием процедуры экстремальной корреляции, определяем целопиксельные координаты сдвига точки 2.1 по отношению к точке 1.1

$$m_1, \quad n_1 \tag{2}$$

и точки 2.2 по отношению к точке 1.2

$$m_2, \quad n_2. \tag{3}$$

Полученные методом корреляции сдвиги выражаются целыми числами пикселей. В задачах визуальной одометрии даже малые ошибки в пределах долей пикселя вследствие накопления на длинных участках траектории могут приводить к значительным ошибкам в конечной точке. Поэтому важным этапом технологии является субпиксельное уточнение полученных оценок.

2.3. Уточнение оценок сдвига методом оптического потока

Для уточнения оценок в пределах одного пикселя так же, как в [12], применяется метод оптического потока. Важным отличием в данном случае является то, что уравнения оптического потока должны применяться отдельно для каждого фрагмента. Таким образом, формируются две независимые системы

$$\mathbf{I}^1_{x,y}\Delta_1 = \mathbf{I}^1_t, \tag{4}$$

$$\mathbf{I}^2_{x,y}\Delta_2 = \mathbf{I}^2_t, \tag{5}$$

где $\mathbf{I}^1_{x,y} = \left[\mathbf{I}^1_x, \mathbf{I}^1_y\right]$, $\mathbf{I}^2_{x,y} = \left[\mathbf{I}^2_x, \mathbf{I}^2_y\right]$ — $N \times 2$-матрицы для фрагментов 1 и 2 соответственно, а фигурирующие здесь $N \times 1$-матрицы $\mathbf{I}^1_x$, $\mathbf{I}^1_y$, $\mathbf{I}^2_x$, $\mathbf{I}^2_y$, $\mathbf{I}^1_t$, $\mathbf{I}^2_t$ — определяются как

$$\begin{aligned} &\mathbf{I}^1_x = \left[\frac{\partial I^1_1}{\partial x}, \frac{\partial I^1_2}{\partial x}, \ldots, \frac{\partial I^1_N}{\partial x}\right]^{\mathrm{T}}, \quad \mathbf{I}^1_y = \left[\frac{\partial I^1_1}{\partial y}, \frac{\partial I^1_2}{\partial y}, \ldots, \frac{\partial I^1_N}{\partial y}\right]^{\mathrm{T}}, \\ &\mathbf{I}^1_t = \left[-\partial I^1_1, -\partial I^1_2, \ldots, -\partial I^1_N\right]^{\mathrm{T}}, \end{aligned} \tag{6}$$

$$
(7)\qquad
\begin{aligned}
&\mathbf{I}_x^2 = \left(\frac{\partial I_1^2}{\partial x}, \frac{\partial I_2^2}{\partial x}, \ldots, \frac{\partial I_N^2}{\partial x}\right)^{\mathrm{T}}, \quad \mathbf{I}_y^2 = \left(\frac{\partial I_1^2}{\partial y}, \frac{\partial I_2^2}{\partial y}, \ldots, \frac{\partial I_N^2}{\partial y}\right)^{\mathrm{T}},\\
&\mathbf{I}_t^2 = \Big[-\partial I_1^2, -\partial I_2^2, \ldots, -\partial I_N^2 \Big]^{\mathrm{T}}.
\end{aligned}
$$

Размеры фрагментов, из отсчетов которых формируются уравнения (4), (5), задаются так, что эти системы являются переопределенными, поэтому вычисляются параметры сдвига, оптимальные в среднеквадратическом смысле:

$$
(8)\qquad \Delta_1 = \Big[\Delta_{x,1}, \Delta_{y,1}\Big]^{\mathrm{T}} = \Big[I_{x,y}^{1\ \mathrm{T}} \mathbf{I}_{x,y}^1\Big]^{-1} \mathbf{I}_{x,y}^{1\ \mathrm{T}} \mathbf{I}_t^1,
$$

$$
(9)\qquad \Delta_2 = \Big[\Delta_{x,2}, \Delta_{y,2}\Big]^{\mathrm{T}} = \Big[\mathbf{I}_{x,y}^{2\ \mathrm{T}} \mathbf{I}_{x,y}^2\Big]^{-1} \mathbf{I}_{x,y}^{2\ \mathrm{T}} \mathbf{I}_t^2.
$$

Полученные субпиксельные смещения добавляются к найденным ранее корреляционным методом целопиксельным относительным смещениям (2) и (3):

$$
(10)\qquad m_{x,1} = m_1 + \Delta_{x,1}, \quad n_{y,1} = n_1 + \Delta_{y,1},
$$

$$
(11)\qquad m_{x,2} = m_2 + \Delta_{x,2}, \quad n_{y,2} = n_2 + \Delta_{y,2}.
$$

Следует напомнить, что уравнения оптического потока формируются по данным, которые являются линейными членами разложения функции яркости в малой окрестности. Поэтому точность оценок существенным образом зависит от интервала аппроксимации. С учетом этого для повышения точности оценок предварительно совмещаются соответствующие пары фрагментов с использованием результатов определения целопиксельных сдвигов, полученных методом корреляции. При этом оценки (8), (9) не могут превышать одного межпиксельного расстояния. Указанное свойство может использоваться в качестве одного из признаков плохой обусловленности или даже вырожденности задачи. Здесь не затрагиваются эти проблемы, являющиеся предметом специальных исследований.

2.4. Формирование текущих оценок координат траектории

В предположении, что камера жестко закреплена на корпусе аппарата, далее вычисляем координаты центральных точек $m_{x,0}$ и $n_{y,0}$ обоих изображений:

$$
(12)\qquad m_{x,0} = \left(m_{x,1} + m_{x,2}\right)/2,
$$

$$
(13)\qquad n_{y,0} = \left(n_{y,1} + n_{y,2}\right)/2
$$

и определяем общий межкадровый сдвиг S центральных точек кадров 1 и 2 как евклидову норму сдвигов в направлении осей Ox, Oy локальной системы координат:

$$
(14)\qquad S = W^* \sqrt{m_{x,0}^2 + n_{y,0}^2},
$$

где W — коэффициент («цена» пикселя), определяемый на этапе калибровки в метрических единицах.

Угол $\Delta\alpha$ межкадрового поворота в локальной системе координат, в соответствии с принятыми обозначениями, определяется как

$$\Delta\alpha = \arcsin\left(n_{y,1} - n_{y,2}\right)/L, \tag{15}$$

а текущее направление движения аппарата для любой точки траектории в глобальной системе координат (далее угол α) является суммой углов $\Delta\alpha_i$, $i = 1, 2, \ldots$ локальных межкадровых поворотов всех предшествующих данной точке пар кадров.

Для вычисленного в каждой точке траектории угла α, характеризующего текущее направление движения аппарата, определяются приращения координат в глобальной системе координат:

$$\Delta x = S^* \cos\alpha, \tag{16}$$

$$\Delta y = S^* \sin\alpha. \tag{17}$$

Текущие координаты траектории в любой точке, как и текущий угол, являются суммой приращений, вычисленных в соответствии с (15), (16) на всех предшествующих точках. Знаки приращений определяются текущими значениями накопленного угла α. Ясно, что координаты x_0, y_0 и угол α_0 в начальной точке глобальной системы координат, с которых «стартует» алгоритм формирования текущих оценок параметров движения, должны быть заданы точно.

2.5. Алгоритмы коррекции текущих оценок и общая схема технологии

Точность оценивания траектории в значительной степени зависит от качества калибровки. Если задана эталонная траектория, калибровка может быть выполнена точно. Однако это не гарантирует высокого качества оценивания траектории, так как параметры движения и установки камеры могут отличаться от эталонных. Если эталонной траектории нет, «цена пикселя» определяется путем расчета по заданным параметров движения и конструкции аппарата. Ясно, что в этом случае, тем более, возможны отличия от заданных значений. Средние калибровочные характеристики можно корректировать по результатам предыдущих проездов. Однако кратковременные отклонения вследствие неравномерности движения носят случайный характер и могут корректироваться только в процессе движения.

Здесь используется алгоритм эпизодической коррекции траектории по наблюдениям текущих отклонений калибровочных характеристик от средних значений на коротких отрезках траектории. В отличие от [12] в данном случае коррекции подвергается только составляющая перемещения в направлении движения аппарата (в направлении оси Ox локальной системы координат). Алгоритм корректировки оценок траектории состоит из следующих шагов.

С использованием найденных по формуле (14) межкадровых сдвигов центральных точек S_k, $k = 1, K$ (здесь K — число точек на отрезке коррекции)

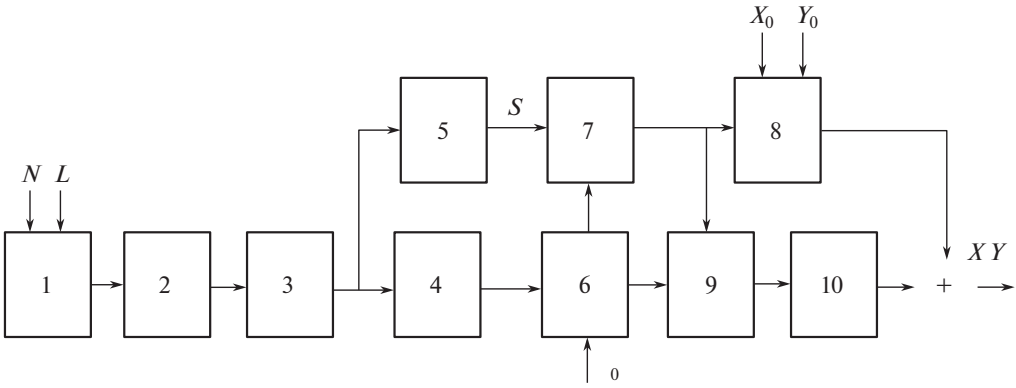


Рис. 4. Связь основных блоков системы визуальной одометрии, осуществляющих вычисление: межкадровых сдвигов методом корреляции (1), субпиксельных оценок методом оптического потока (2), координат центральных точек кадров изображений (3), локального угла межкадрового поворота $\Delta\alpha$ (4), локального межкадрового сдвига центров кадров (5), угла α текущего направления в глобальной системе координат (6), локальных приращений координат (7), текущих оценок координат траектории (8), суммарного приращения координат на малом отрезке (9), корректирующих добавок координат (10).

определяем общий сдвиг S_Σ на заданном локальном отрезке траектории:

$$S_\Sigma = \sum_{k=1}^{K} \Big[S_k. \tag{18}$$

Далее определяем отношение отклонения общего сдвига от эталонного значения (вычисленного на этапе калибровки) к общему сдвигу на локальном участке:

$$\Delta S_{relate} = (S_\Sigma - S_{etal}) / S_\Sigma. \tag{19}$$

Решающее правило корректировки состоит в проверке условий:

$$\Delta S_{relate} < thr_{\Delta S_{\text{relat}}} \quad \text{и} \quad |\Delta S_\Sigma > | \, thr_{\Delta S_\Sigma}, \tag{20}$$

где $thr_{\Delta S_{\text{relat}}}$, $thr_{\Delta S_\Sigma}$ — заданные пороговые значения. Если оба неравенства выполняются, вычисляется корректирующая добавка

$$\Delta x_{kor} = par \Delta S_{relate} sign(\Delta S_\Sigma), \tag{21}$$

которая на следующем такте оценки траектории суммируется с текущей оценкой.

Общая структурная схема описанной системы визуальной одометрии представлена на рис. 4. Здесь в блоках 1–8 реализуется технология оценки текущих координат траектории, включающая следующие этапы: определение целопиксельных сдвигов методом экстремальной корреляции, субпиксельное уточнение оценок методом оптического потока по соотношениям (4)–(11) и формирование текущих оценок координат траектории по соотношениям (12)–(17). Блоки 9, 10 реализуют алгоритм коррекции текущих оценок в соответствии с соотношениями (18)–(21).

3. Экспериментальные исследования

3.1. Моделирование тестовых траекторий

К сожалению, в настоящее время в открытом доступе пока отсутствуют тестовые последовательности изображений опорной поверхности, соответствующие изучаемой схеме съемки. Поэтому первая часть экспериментов состояла в моделировании эталонных траекторий для различных типов опорных поверхностей. Моделирование проводилось на плоских горизонтальных поверхностях. Это не является серьезным недостатком, так как траекторию движения по заданному рельефу всегда можно восстановить по локальным проекциям отрезков траектории на плоскость, касательную к каждой точке рельефа.

Технология моделирования последовательностей изображений опорных поверхностей строилась следующим образом. На основе игрового движка Unreal Engine 4 [19] проектировалась виртуальная сцена с имитацией различного типа подстилающей поверхности, в нулевых координатах которой располагалась модель робота с установленной камерой регистрации. В момент старта симуляции робот проходил заданные траектории (прямая, окружность, квадрат, восьмерка, не пересекающаяся и пересекающаяся поверхность) с заданной скоростью перемещения (2, 8 и 15 км/ч), повторяя итерацию для каждого типа подстилающей поверхности. Камера, закрепленная на роботе и направленная вниз, записывала изображения с частотой 60 кадров в секунду. Последовательности изображений опорных поверхностей сформированы для следующих типов поверхности: древесное, каменистое (гравий), почвенное, бетонное и металлическое покрытия, а также песок, грязь и уличная плитка. Последовательности изображений различной длины для указанных типов опорных поверхностей размещены на сайте [20].

3.2. Эксперимент по оценке точности технологии

Для проверки качества разработанной технологии визуальной одометрии использовалась тестовая траектория, включающая 2111 изображений опорной поверхности типа гравий на отрезке времени 35,18 с и соответствующий этой последовательности файл точных координат траектории.

На рис. 5 приведены тестовая (отмечена символом *) и оцененная (символ о) траектории, полученные при следующих значениях параметров алгоритмов: размеры фрагментов — 9×9; расстояние от центра кадра до фрагмента — $L/2 = 100$ (пикселей), соответственно расстояние между фрагментами $L = 200$ (пикселей); среднее значение калибровочного коэффициента $W = 0{,}0782$, длина участка для вычисления локальных калибровочных коэффициентов — 25 точек.

Тестовая траектория строилась по значениям, взятым из файла точных координат. За время между соседними точками на траекториях обрабатывалось 25 кадров последовательности изображений. Координаты начальных точек тестовой и оцененной траектории совпадали (точка — 0).

На рис. 6 приведены графики ошибок оценивания угла (в рад.) и текущих координат траектории (в см), полученные при указанных выше условиях

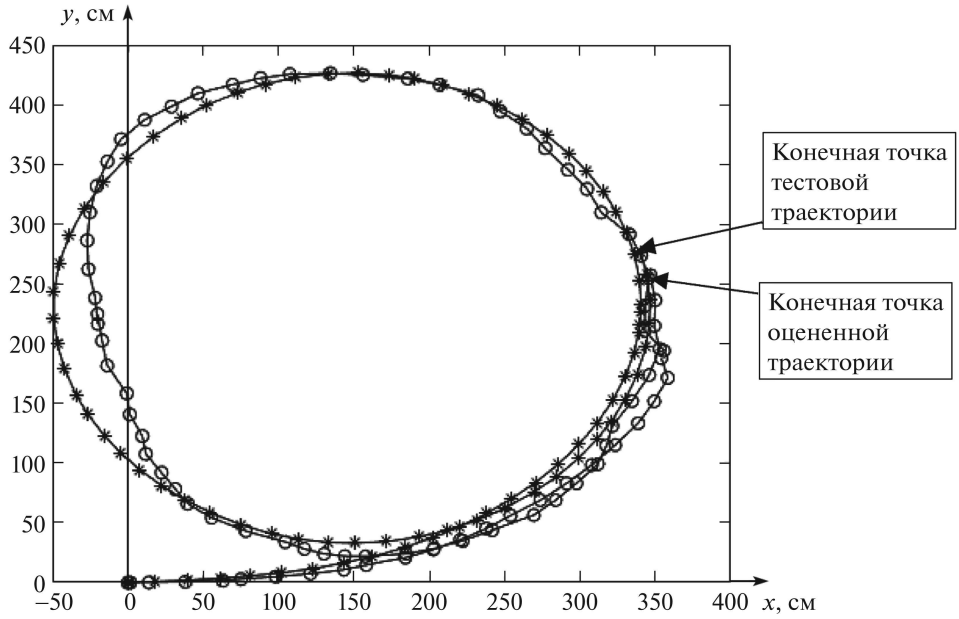


Рис. 5. Тестовая (*) и оцененная (о) траектории (общая точка старта — 0).

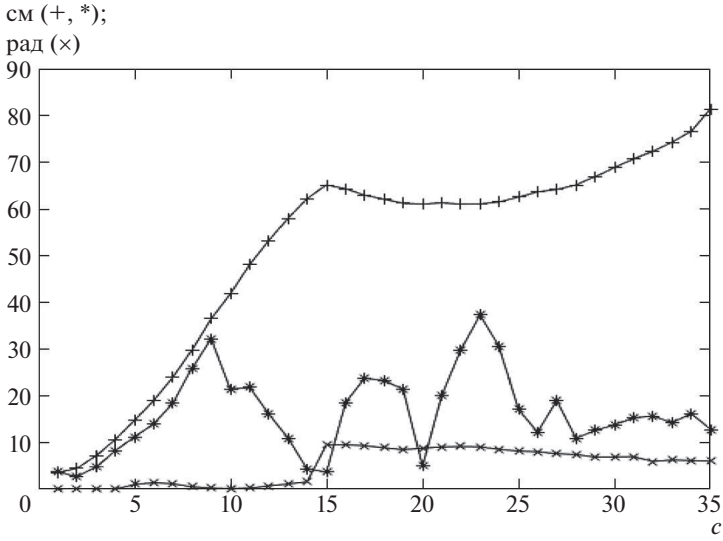


Рис. 6. Графики изменения во времени текущего угла α (×) (рад.) и евклидовой ошибки оценивания траектории (см): без коррекции (+), с коррекцией оценок (*).

эксперимента и значениях параметров алгоритмов. При этом были получены следующие относительные значения СКО:

— по координате X — 0,0647,

— по координате Y — 0,0491,

— по евклидовой метрике — 0,0569.

На графиках видно, что при отсутствии коррекции оценок (+) ошибка одометрии возрастает при быстром изменении (возрастании) текущего угла (×). Ошибка существенно снижается в результате коррекции оценок траектории (*). Относительные ошибки в конечной точке траектории составили менее 0,02 и 0,04 для координат X и Y соответственно.

4. Заключение

Предложена новая технология прямой визуальной одометрии по последовательности изображений опорной поверхности, регистрируемых камерой, жестко закрепленной на корпусе аппарата. Технология включает следующие основные этапы: определение межкадровых (целопиксельных) сдвигов методом экстремальной корреляции (1); субпиксельное уточнение оценок сдвига методом оптического потока (2); определение локальных межкадровых и накопленного углов поворота аппарата (3); формирование текущих оценок координат траектории с использованием локальных межкадровых сдвигов и текущего угла поворота (4); коррекция текущих оценок координат путем вычисления локальных тарировочных характеристик в процессе движения.

Важной особенностью разработанной технологии является определение межкадровых сдвигов, как целопиксельных, так и субпиксельных отдельно на двух разнесенных фрагментах каждого кадра. Для реализации этого подхода наложены ограничения на размеры этих фрагментов, при которых сдвиг соответствующих точек фрагментов не превышает половины межпиксельного расстояния. Выполнение указанного требования позволило независимо определить локальные межкадровые повороты, используя уже найденные координаты центров фрагментов. Это обеспечило повышение точности оценок траектории и существенное сокращение объема вычислений, так как отпала необходимость на каждом такте вычислять матрицу поворота.

В рамках предложенной технологии реализован алгоритм коррекции текущих оценок траектории путем вычисления отклонений локальных тарировочных характеристик от их заданных средних значений. В данном случае для жестко закрепленной камеры особенность состоит в том, что коррекция осуществляется только в направлении оси OX локальной системы координат. Приведенные результаты показывают возможность оценивания траектории в реальном времени с высокой точностью.

СПИСОК ЛИТЕРАТУРЫ

1. *Delmerico J., Scaramuzza D.* A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots // IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 2018. P. 2502–2509. https://doi.org/10.1109/ICRA.2018.8460664
2. *He M., Zhu C., Huang Q., Ren B., Liu J.* A review of monocular visual odometry // Visual Comput. 2020. V. 36. P. 1053–1065.

   https://doi.org/10.1007/s00371-019-01714-6
3. *Gao L., Su J., Cui J., Zeng X., Peng X., Kneip L.* Efficient Globally-Optimal Correspondence-Less Visual Odometry for Planar Ground Vehicles // IEEE In-

ternational Conference on Robotics and Automation (ICRA), Paris, France, 2020. P. 2696–2702 (2020). https://doi.org/10.1109/ICRA40945.2020.9196595

4. *Forster C., Zhang Z., Gassner M., Werlberger M., Scaramuzza D.* SVO: Semidirect visual odometry for monocular and multicamera systems // IEEE Transac. on Robot. 2017. V. 33. No. 2. P. 249–265.

   https://doi.org/10.1109/TRO.2016.2623335

5. *Muller P., Savakis A.* Flowdometry: An optical flow and deep learning based approach to visual odometry // IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, USA, 2017. P. 624–631.

   https://doi.org/10.1109/WACV.2017.75

6. *Gonzalez R., Rituerto A., Guerrero J.J.* Improving robot mobility by combining downward-looking and frontal cameras // Robotics. 2016. V. 5. No. 4. P. 25.

   https://doi.org/10.3390/robotics5040025

7. *Birem M., Kleihorst R., El-Ghouti N.* Visual odometry based on the Fourier transform using a monocular ground-facing camera // J. Real-Time Image Proc. 2018. V. 14. P. 637–646. https://doi.org/10.1007/s11554-017-0706-3

8. *Gilles M., Ibrahimpasic S.* Unsupervised deep learning based ego motion estimation with a downward facing camera // Vis Comput, 2021.

   https://doi.org/10.1007/s00371-021-02345-6

9. *Fu B., Shankar K.S., Michael N.* RaD-VIO: Rangefinder-aided Downward Visual-Inertial Odometry // International Conference on Robotics and Automation (ICRA), Montreal, Canada, 2019, P. 1841–1847.

   https://doi.org/10.1109/ICRA.2019.8793741

10. *Goecke R., Asthana A., Pettersson N., Petersson L.* Visual Vehicle Egomotion Estimation using the Fourier-Mellin Transform // IEEE Intelligent Vehicles Symposium, Istanbul, Turkey, 2007. P. 450–455. https://doi.org/10.1109/IVS.2007.4290156

11. *Nourani-Vatani N., Borges P.V.K.* Correlation-based visual odometry for ground vehicles // J. Field Robot. 2011. V. 28. No. 5. P. 742–768.

    https://doi.org/10.1002/rob.20407

12. *Фурсов В.А., Минаев Е.Ю., Котов А.П.* Оценивание параметров движения аппарата по наблюдениям опорной поверхности // АиТ. 2021. № 10. С. 124–139. *Fursov V.A., Minaev E.Y., Kotov A.P.* Vehicle Motion Estimation Using Visual Observations of the Elevation Surface // Autom. Remote Control. 2021. V. 82. No. 10. P. 1730–1741.

13. *Kitt B., Geiger A., Lategahn H.* Visual odometry based on stereo image sequences with ransacbased outlier rejection scheme // IEEE intelligent vehicles symposium, La Jolla, USA, 2010. P. 486–492. https://doi.org/10.1109/IVS.2010.5548123

14. *Pire T., Fischer T., Civera J., Cristoforis P.D., Berlles J.J.* Stereo parallel tracking and mapping for robot localization // IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 2015. P. 1373–1378. https://doi.org/10.1109/IROS.2015.7353546

15. *Mur-Artal R., Tardos J.D.* Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras // IEEE Transac. Robot. 2017. V. 33. No. 5. P. 1255–1262. https://doi.org/10.1109/TRO.2017.2705103

16. *Newcombe R.A., Lovegrove S.J., Davison A.J.* Dtam: Dense tracking and mapping in real-time // International Conference on Computer Vision, Barcelona, Spain, 2011. P. 2320–2327. https://doi.org/10.1109/ICCV.2011.6126513

17. *Engel J., Schops T., Cremers D.* Lsd-slam: Large-scale direct monocular slam // European conference on computer vision, Zurich, Switzerland, 2014. P. 834–849. https://doi.org/10.1007/978-3-319-10605-2_54
18. *Kerl C., Sturm J., Cremers D.* Dense visual slam for rgb-d cameras // IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 2013. P. 2100–2106. https://doi.org/10.1109/IROS.2013.6696650
19. https://www.unrealengine.com // Unreal engine 4 – Game Engine. 2021.
20. https://github.com/by-LZ-for/Synthetic-Dataset-Robot // Synthetic Dataset for Visual Odometry. 2021.

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 23.01.2022

После доработки 14.04.2022

Принята к публикации 29.06.2022

© 2022 г. А.С. МАРКОВ (to.asmarkov@gmail.com),
Е.Ю. КОТЛЯРОВ (tyztot@gmail.com),
Н.П. АНОСОВА (anosova-np@rudn.ru),
В.А. ПОПОВ, канд. физ.-мат. наук (popov-va@rudn.ru)
(Российский университет дружбы народов, Москва),
Я.М. КАРАНДАШЕВ, канд. физ.-мат. наук (karandashev@niisi.ras.ru)
(Российский университет дружбы народов, Москва;
Научно-исследовательский институт
системных исследований РАН, Москва),
Д.Е. АПУШКИНСКАЯ, д-р физ.-мат. наук (apushkinskaya@gmail.com)
(Российский университет дружбы народов, Москва)

ИСПОЛЬЗОВАНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ВЫЯВЛЕНИЯ АНОМАЛИЙ НА РЕНТГЕНОВСКИХ СНИМКАХ, ПОЛУЧЕННЫХ НА СКАНЕРАХ ПЕРСОНАЛЬНОГО ДОСМОТРА

В данной работе рассматривается решение задачи выявления аномалий на рентгеновских снимках, полученных сканерами персонального досмотра (СПД). В работе описана последовательность и описание методов предобработки изображений, с помощью которых оригинальные изображения, полученные на СПД, преобразуются к изображениям с визуально различимыми аномалиями. Приведены примеры обработанных снимков. Показаны первые (предварительные) результаты использования нейронной сети для выделения аномалий.

Ключевые слова: сканеры персонального досмотра, рентгеновские снимки, выделение аномалий, выравнивание гистограммы изображения, нейронные сети, U-2-Net.

DOI: 10.31857/S0005231022100038, **EDN:** AJXBSP

1. Введение

На объектах, требующих повышенного контроля безопасности, часто используются сканеры персонального досмотра. Они позволяют быстро сделать снимок человека в рентгеновском диапазоне излучения, на котором оператор сканера персонального досмотра (СПД) может увидеть все объекты на теле человека и визуально подтвердить или опровергнуть наличие запрещенных среди них.

Процесс обладает рядом существенных недостатков, в том числе связанных с человеческим фактором: для качественного анализа снимка требуются существенное время и повышенное внимание, что приводит к быстрой утомляемости оператора СПД и может негативно сказаться на качестве анализа снимков. В данный процесс можно внести существенную долю автоматиза-

ции, сделав его более дешевым для организации и более комфортным для человека.

Для решения поставленной задачи использовались глубокие нейронные сети. В этой работе представлены способы предобработки данных и применение сети U-2-Net, а также анализ результатов.

2. Постановка задачи

Требуется разработать решение, ставящее в соответствие каждому рентгеновскому снимку булеву маску, единичные значения которой соответствовали бы пикселям, на которых присутствовали инородные объекты, такие как телефоны, оружие, металлические предметы и прочее. Эти объекты в дальнейшем будем называть аномалиями.

Набор данных, предоставленный компанией, специализирующейся на разработке сканеров персонального досмотра, состоит из оригинальных снимков людей, полученных с аппарата персонального досмотра человека «Express inspection» в рентгеновском спектре волн. Всего предоставлено четыре набора данных с различных сканеров персонального досмотра. Каждый снимок представляет из себя одноканальное 16-битное изображение в формате tiff размера 1600×500 пикселей. На снимках присутствуют различные аномалии, такие как: элементы одежды, аксессуары, оружие, протезы и прочее. Всего набор данных содержит 1654 снимка.

Выделим две проблемы с данными:

1. На оригинальных снимках аномалии визуально не различимы, что затрудняет их визуальный анализ.

2. Различные аппараты выдают снимки с различным распределением значений интенсивности пикселей. По этой причине исходные данные не подходят для автоматической обработки нейронной сетью.

Таким образом, во-первых, необходимо было разработать алгоритм для предобработки снимков, который решает обе поставленные задачи. После этого требовалось осуществить разметку аномалий на снимках. Наконец, для автоматизации этого процесса нужно было обучить нейронную сеть выделению аномалий.

3. Методы

3.1. Существующие подходы

Сегментация объектов на рентгеновских снимках является весьма распространенной задачей. Сначала эта задача решалась с помощью классических методов обработки изображений [1, 2]. Но со временем эта задача стала решаться преимущественно с помощью сверточных нейронных сетей.

В [3] произведена модификация архитектуры SegNet [4], упрощающая оригинальную сеть и позволяющая проводить обучение на небольшом наборе данных. Дальнейшее развитие сети SegNet — архитектура XNet [5], специализирующаяся на сегментации рентгеновских изображений, а именно на разделении мягкий тканей и костей.

Задача сегментации получила особое распространение в сфере медицины. Огромное количество работ было посвящено сегментации клеточных струк-

тур [6–8]. Среди множества подобных работ выделяется сеть U-Net [6], подходящая для более широкого класса задач. Эта сеть входит в число базовых архитектур для сегментации. В ней используются только сверточные слои, что позволяет подавать на вход изображения произвольного размера и получать маску с классами на выходе.

Следующем этапом развития U-Net является нейронная сеть U-2-Net [9]. Она показывает лучшие результаты на различных наборах данных по сравнению с U-Net. При этом она так же осталась применима для весьма широкого спектра задач, в том числе для анализа рентгеновских снимков. В данной работе использовалась именно эта архитектура.

3.2. Предобработка изображений

Для исправления описанных выше недостатков данных требовалось создать алгоритм предобработки снимков, состоящий из последовательности преобразований, решающий обе поставленные задачи.

В рамках этого направления произведена поэтапная обработка снимков, в результате которой снимки приведены к единому формату. Использованы алгоритмы увеличения контрастности снимков и удаления аномальных значений интенсивности пикселей. В частности, для повышения визуальной видимости применены методы трансформации гистограммы распределений пикселей к более светлым тонам, что позволяет оставить на снимке только самые темные объекты.

Для начала посмотрим, как выглядит гистограмма распределений пикселей в оригинальном изображении. Значение 0 — это черный цвет пикселя, 255 — белый (в дальнейшем для улучшения визуализации все гистограммы нормализуются от 0 до 255).

Из рис. 1 видно, что большинство пикселей оригинального изображения имеют яркость в окрестности значений 25 (черный цвет — человек) и 240 (белый цвет — фон). Цель состоит в уменьшении количества темных пикселей

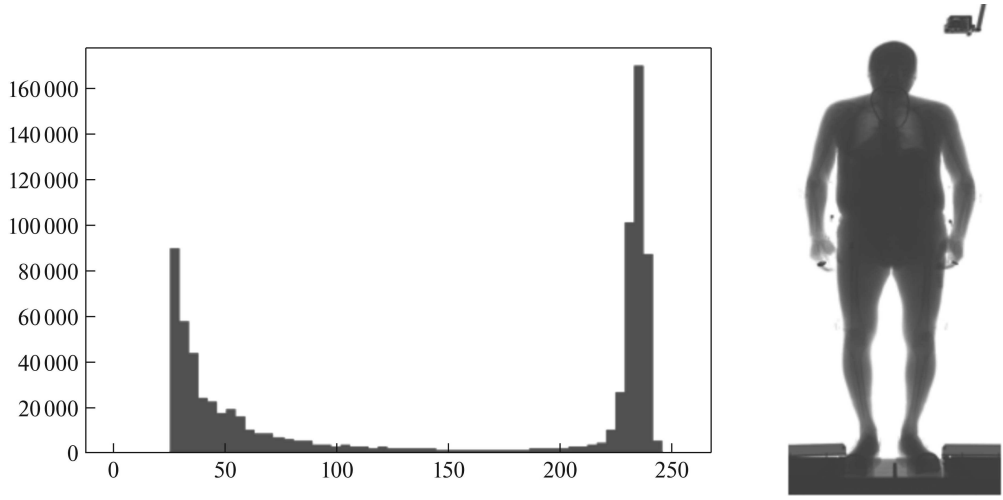


Рис. 1. Гистограмма оригинального изображения.

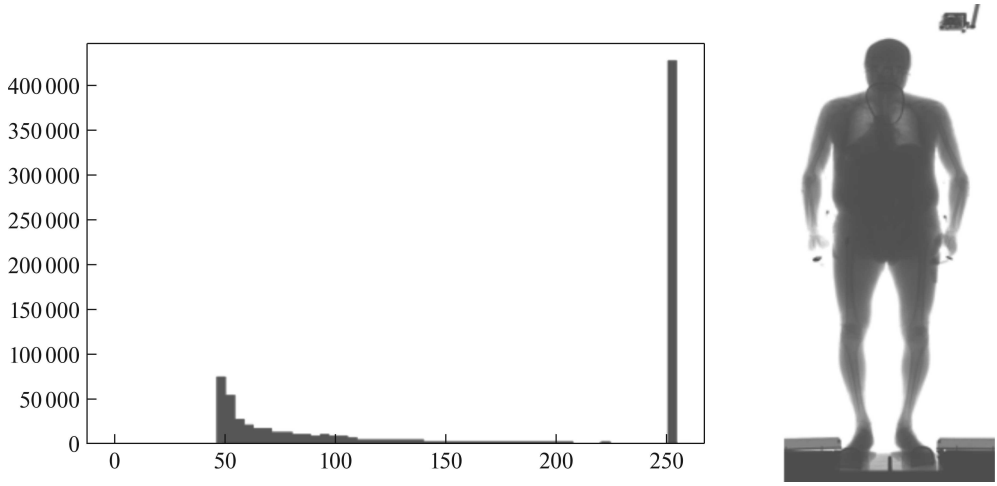


Рис. 2. Гистограмма после ThreshTrunc.

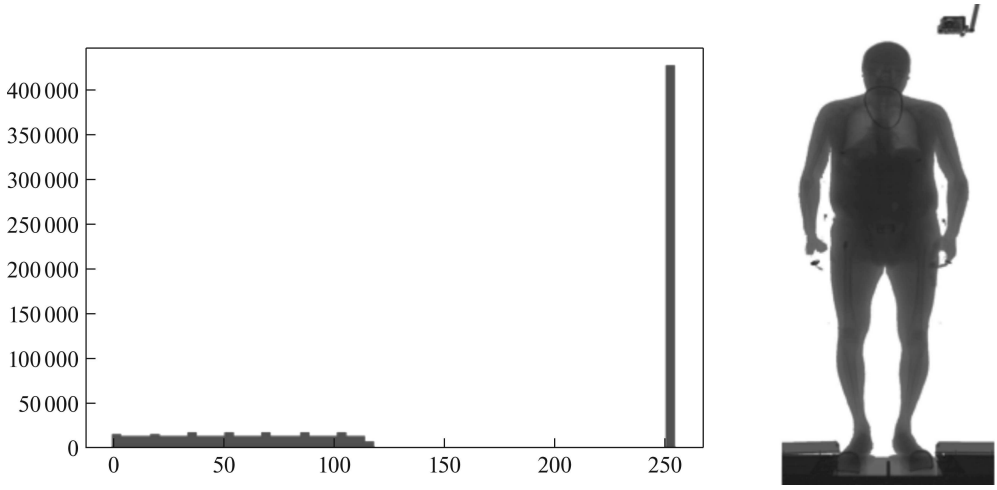


Рис. 3. Гистограмма после EqualizeHist.

так, чтобы менее темные участки (мягкие ткани) перешли в более светлый цвет, а самые темные участки (кости и металлические объекты) оставались темными. К оригинальным снимкам применяется следующая последовательность процедур обработки:

1) ThreshTrunc (обрезание по порогу)
2) EqualizeHist (выравнивание гистограммы)
3) ThreshTrunc
4) EqualizeAdapthist (адаптивное выравние гистограммы)
5) ThreshTrunc.

Ниже каждая из этих операций описана подробнее.

3.2.1. ThreshTrunc. ThreshTrunc [10] — операция изменения гистограммы распределения таким образом, что значения интенсивности всех пиксе-

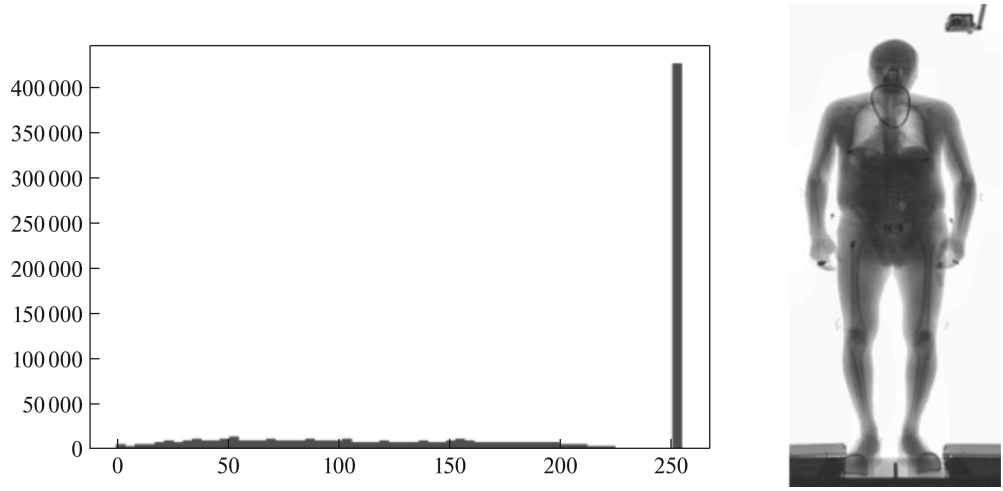


Рис. 4. Гистограмма после EqualizeAdapthist.

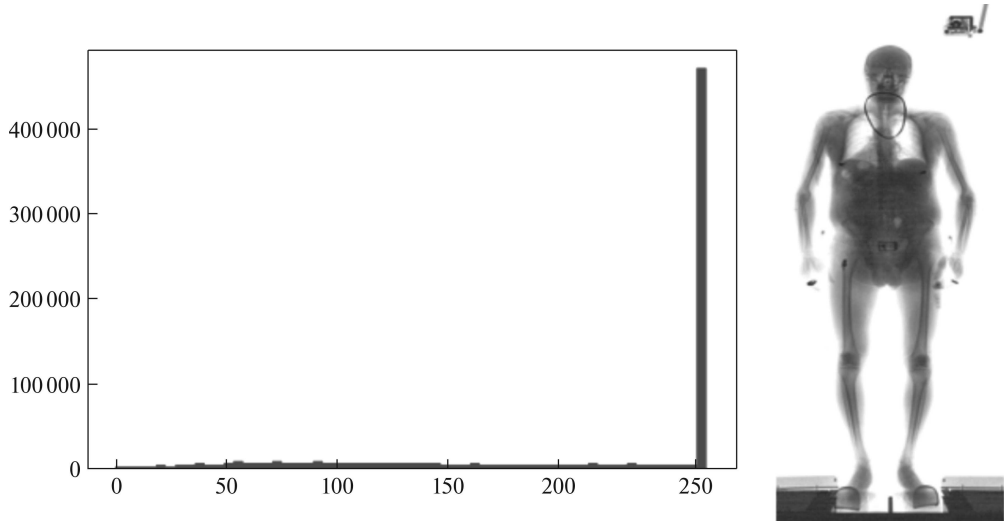


Рис. 5. Итоговая гистограмма.

лей, большие среднего значения интенсивности на снимке, заменяются на это среднее значение. Это влечет за собой смещение гистограммы распределения интенсивности пикселей в более светлую часть.

$$\text{ThreshTrunc}\,(x,y) = \begin{cases} \text{src}\,(x,y), & \text{если src}\,(x,y) < \text{treshold}, \\ \text{treshold}, & \text{если src}\,(x,y) \geqslant \text{treshold}, \end{cases}$$

где threshold — порог, равный среднему значению интенсивности всех пикселей снимка, $\text{src}\,(x,y)$ — значение интенсивности пикселя с координатами (x,y).

3.2.2. EqualizeHist. EqualizeHist — процедура выравнивания гистограммы [11], позволяющая увеличить общий контраст изображения. Данное пре-

образование особенно эффективно, когда изображение представлено узким диапазоном значений интенсивности. Благодаря этому преобразованию можно лучше распределить интенсивности на гистограмме, равномерно используя весь диапазон кодирования интенсивности цвета.

$H(i)$ — гистограмма для значения интенсивности каждого пикселя. Находим кумулятивное распределение

$$H'(i) = \sum_{0 \leqslant j < i} H(j).$$

Затем заменяем значения интенсивности пикселя в изображении на получившееся значение из распределения:

$$\text{EqualizeHist}\,(x, y) = H'(\text{src}\,(x, y)).$$

3.2.3. EqualizeAdapthist. EqualizeAdapthist — адаптивное выравнивание гистограммы. В отличие от EqualizeHist, данный метод вычисляет несколько гистограмм, каждая из которых соответствует отдельному участку изображения, и использует их для перераспределения значений интенсивности (по алгоритму EqualizeHist). Поэтому он подходит для улучшения локального контраста и улучшения четкости краев в каждой области изображения.

Для применения функций ThreshTrunc, EqualizeHist и EqualizeAdapthist использовался язык программирования Python3 и библиотека OpenCV [12].

Проведя все преобразования, мы получаем распределение пикселей, представленное на рис. 5. Как следует из этой гистограммы, большинство пикселей являются светлыми (т.е. фоном и участками мягких тканей), а остальные участки снимка распределены по градациям интенсивности более равномерно.

После применения всех преобразований металлические объекты на снимке становятся визуально различимы. Это позволяет выделять инородные телу объекты, не прибегая к специализированному программному обеспечению для индивидуальной предобработки каждого изображения. На рис. 6 показан процесс улучшения снимка после каждого этапа обработки.

До обработки средние значения интенсивности пикселей на разных сканерах сильно различаются (см. таблицу). После обработки средние значения становятся практически равными, т.е. изображения приводятся к единому формату, и их можно подавать на вход нейронной сети для выделения аномалий.

Таким образом, применение данного алгоритма предобработки изображений позволяет нивелировать различия в конструкции и настройках сканеров персонального досмотра.

Средние значения интенсивности пикселей

	Набор A	Набор B	Набор C	Набор D
До обработки	176,71	180,41	173,61	158,81
После обработки	201,36	201,16	203,36	201,62

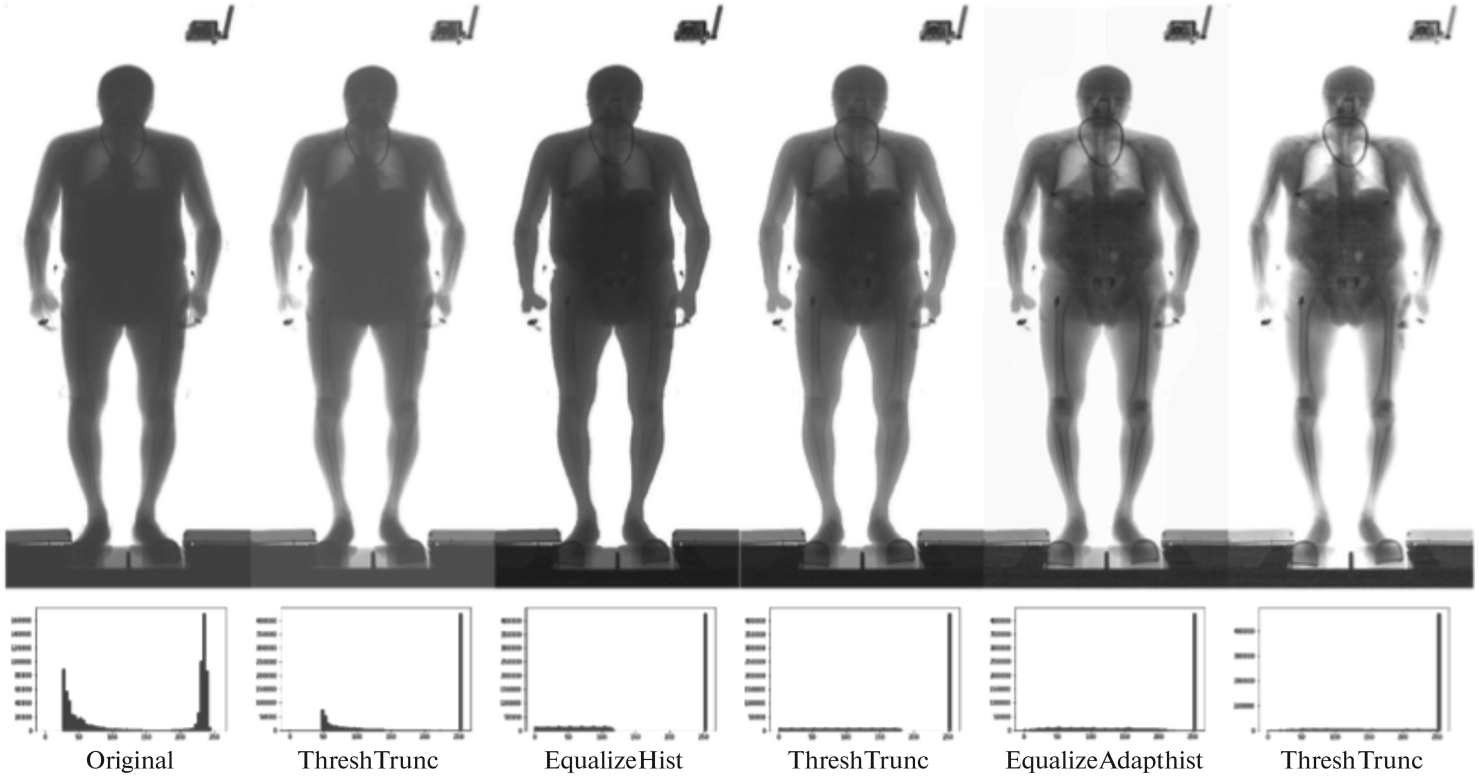


Рис. 6. Результат обработки после каждого преобразования.

3.3. Разметка

После обработки набора данных аномалии стали визуально различимыми. Благодаря этому с помощью программы labelme [13] на снимках вручную были выделены инородные объекты. Ручная разметка производилась в том числе с помощью сервиса Yandex Toloka [14]. Всего было выделено около 18 000 аномалий на 1654 изображениях. В результате были получены обработанные изображения и маски инородных объектов для них (пример приведен на рис. 7).

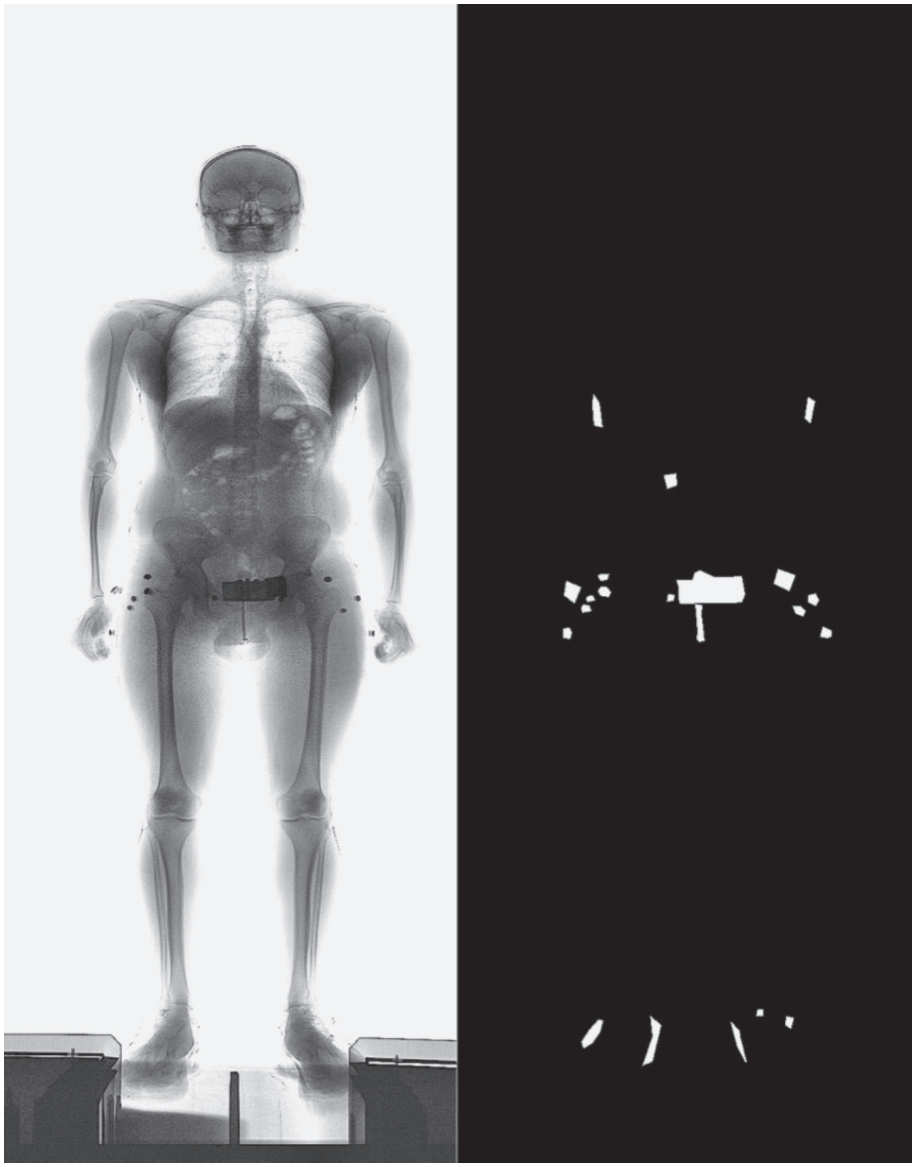


Рис. 7. Пример размеченной вручную маски.

3.4. Нейронная сеть

Для решения задачи выявления аномалий использовалась сеть U-2-Net [9], которая была представлена в 2020 г. Для реализации нейронной сети использовались язык программирования Python и фреймворк PyTorch. Архитектура использовалась без дополнительных модификаций. В качестве оптимизатора был выбран алгоритм Adam [15] с параметром скорости обучения (learning rate), равным 0,001. Обучение проводилось на графическом процессоре Nvidia gtx 1080ti с 11гб памяти. Размер изображений на входе сети составляет 512×512 пикселей типа float32. В связи с ограничением памяти и большим количеством слоев нейронной сети размер батча был выбран равным десяти.

На вход модели подавались одноканальные снимки с указанной выше предобработкой. В процессе аугментации для увеличения вариативности данных использовалась операция случайной обрезки изображения (cropping), что позволило сгенерировать большое количество изображений, содержащих различные части тела. Модель отображает снимок в матрицу, характеризующую вероятности нахождения аномалий в соответствующих участках снимка. Далее матрица вероятностей приводится к бинарному виду. Таким образом, на выходе имеем булеву маску. В качестве функции потерь была использована функция

$$L(A, B) = 1 - \mathrm{IoU}\,(A, B),$$

где A — оригинальная маска, B — маска, полученная из нейронной сети.

В качестве метрики была выбрана функция IoU (intersection over union) [16], которая характеризует, насколько предсказанная маска покрывает настоящую маску:

$$\mathrm{IoU}\,(A, B) = \frac{A \cap B}{A \cup B}.$$

Обучение проводилось на 1454 снимках. Тестовый набор данных содержал 200 изображений.

4. Заключение

В результате проведенной на первом этапе работы разработан универсальный алгоритм, который позволяет обрабатывать снимки и приводить их к виду, подходящему для обучения нейронной сети. Была обучена нейронная

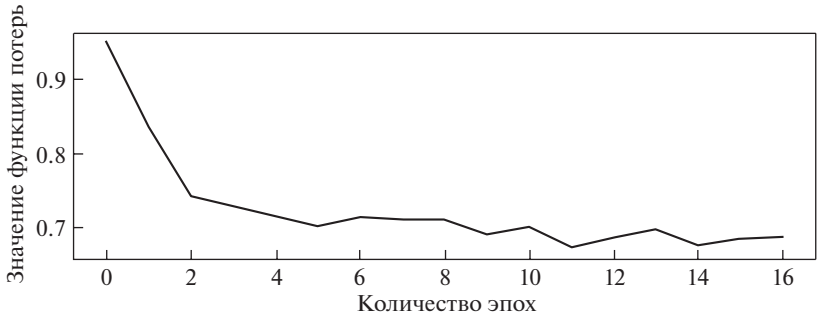


Рис. 8. Значение функции потерь на тестовой выборке в процессе обучения.

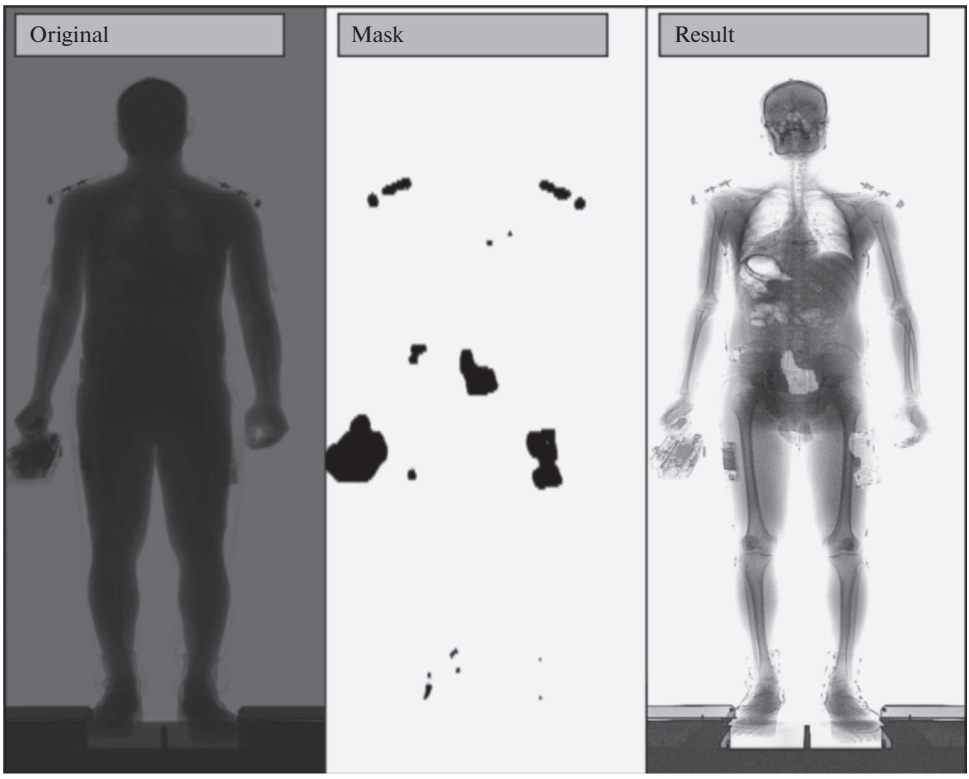


Рис. 9. Слева оригинальный снимок, в центре маска, полученная с помощью нейронной сети, справа маска, наложенная на обработанный снимок.

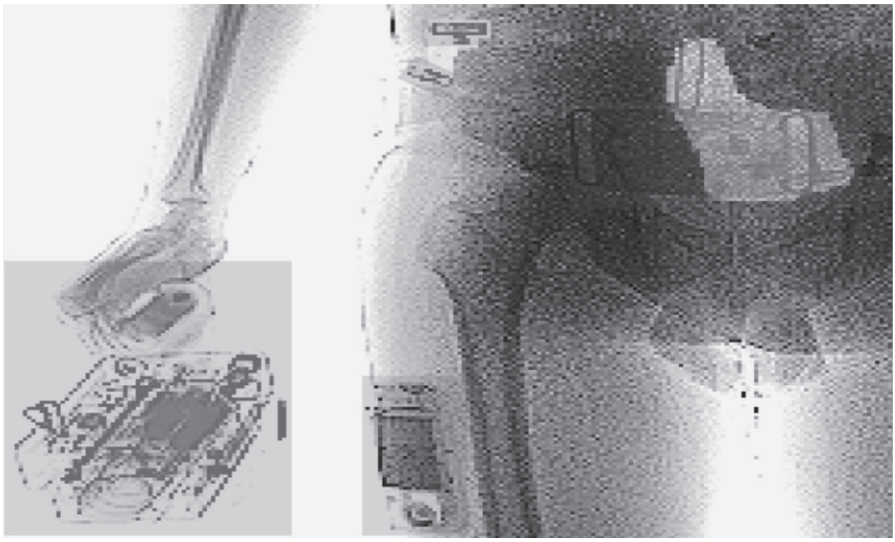


Рис. 10. Детальное рассмотрение полученной маски.

сеть (см. рис. 8), выполняющая поставленную задачу. В результате проверки модели на тестовом наборе данных с помощью метрики IoU сеть показала точность в 0,838.

Результаты работы этой модели приведены ниже (см. рис. 9 и 10). Можно видеть, что нейронная сеть научилась выделять большие инородные объекты, но границы этих объектов пока выделяются плохо. Получение маски на выходе нейронной сети одного снимка на CPU в среднем занимает 4 с.

Обычно сети типа U-2-net используются для сегментации цветных фотографий и выделения объектов на них [9]. К сожалению, не удалось найти открытых источников, где бы подобные сети применялись для рентгеновских снимков. По этой причине нет возможности сравнить качество полученной модели.

5. Выводы

Предложена универсальная схема предобработки изображений, которая позволяет приводить данные с различных аппаратов персонального досмотра (или с различными настройками излучения) к нормализованному формату, в котором средние значения интенсивности пикселей совпадают, и при этом все объекты становятся различимы человеком.

С использованием предложенного подхода обработки из четырех различных наборов данных сформирована обучающая выборка. На основе полученной выборки обучена нейронная сеть U-2-net. Качество сегментации обученной модели позволяет распознавать аномалии большого и среднего размера. На объектах сравнительно меньшего размера качество сегментации существенно снижается. Несмотря на это, модель может быть использована на промышленных объектах, в качестве средства автоматизации работы СПД для поиска объектов среднего размера, таких как оружие, телефоны, слитки металлов и прочее, значительно увеличивая скорость работы оператора.

СПИСОК ЛИТЕРАТУРЫ

1. *Sharma N., Aggarwal L.M.* Automated medical image segmentation techniques // J. Med. Phys. 2010. V. 35. No. 1. P. 3–14.
2. *Mansoor A., Bagci U., Foster B., et. al.* Segmentation and image analysis of abnormal lungs at CT: current approaches, challenges, and future trends // Radiographics. 2015. V. 35. No. 4. P. 1056–1076.
3. *Badrinarayanan V., Handa A., Cipolla R.* Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labeling. arXiv preprint. arXiv:1505.07293, 2015.
4. *Badrinarayanan V., Kendall A., Cipolla R.* Segnet: a deep convolutional encoder-decoder architecture for image segmentation // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017. V. 39. No. 12. P. 2481–2495.
5. *Aylett-Bullock J., Cuesta-Lázaro C., Quera-Bofarull A.* XNet: a convolutional neural network (CNN) implementation for medical X-Ray image segmentation suitable for small datasets // Proc. SPIE. Medical Imaging 2019: Biomedical Applications in Molecular, Structural, and Functional Imaging. 2019. V. 10953.

6. *Ronneberger O., Fischer P., Brox T.* U-Net: convolutional networks for biomedical image segmentation / Navab N., Hornegger J., Wells W., Frangi A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Lecture Notes in Computer Science. 2015. V. 9351. P. 234–241.
7. *Ciresan D., Giusti A., Gambardella L.M., Schmidhuber J.* Deep neural networks segment neuronal membranes in electron microscopy images / Advances in Neural Information Processing Systems 25, Pereira F., Burges C.J.C., Bottou L., and Weinberger K.Q., eds., 2843-2851, Curran Associates, Inc., 2012.
8. *Arganda-Carreras I., Turaga S.C., Berger D.R., et. al.* Crowdsourcing the creation of image segmentation algorithms for connectomics // Front. Neuroanat. 2015. V. 9. No. 142.
9. *Xuebin Q., Zhang Z., Huang C., et. al.* U2-Net: Going deeper with nested U-structure for salient object detection // Pattern Recognition. 2020. V. 106. P. 107404.
10. `https://docs.opencv.org/3.4/db/d8e/tutorial_threshold.html`
11. `https://docs.opencv.org/3.4/d6/dc7/group_imgproc_hist.html`
12. OpenCV. https://opencv.org/
13. LabelMe. http://labelme.csail.mit.edu/Release3.0/
14. Яндекс Толока. https://toloka.ai/
15. *Kingma D.P., Ba J.L.* Adam: a method for stochastic optimization. arXiv preprint. arXiv:1412.6980, 2017.
16. `https://id.wikipedia.org/wiki/Indeks_Jaccard`

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 01.02.2022

После доработки 31.05.2022

Принята к публикации 29.06.2022

© 2022 г. Д.В. СВИТОВ (d.svitov@expasoft.tech)
(ООО Экспасофт, Новосибирск;
Институт автоматики и электрометрии СО РАН, Новосибирск),
С.А. АЛЯМКИН, канд. техн. наук (s.alyamkin@expasoft.com)
(ООО Экспасофт, Новосибирск)

ДИСТИЛЛЯЦИЯ МОДЕЛЕЙ ДЛЯ РАСПОЗНАВАНИЯ ЛИЦ, ОБУЧЕННЫХ С ПРИМЕНЕНИЕМ ФУНКЦИИ СОФТМАКС С ОТСТУПАМИ

Использование сверточных нейронных сетей в сочетании с функцией Софтмакс (анг. softmax) с отступами позволяет достичь наибольшей точности в задаче распознавания лиц. Развитие встраиваемых систем, таких как умные домофоны, породило интерес к легковесным нейронным сетям. Так были предложены облегченные нейросетевые модели, обученные с применением функции Софтмакс с отступами, для задачи идентификации по лицу. В данной работе предлагается метод дистилляции, который позволяет получить большую точность, чем другие методы для задачи распознавания лиц на наборах данных LFW, AgeDB-30 и Megaface. Основная идея предлагаемого подхода заключается в использовании центров классов сети-учителя для инициализации сети-ученика. Затем сеть-ученик обучается производить биометрические вектора, углы от которых до центров классов равны углам в сети-учителе.

Ключевые слова: сверточные нейронные сети, дистилляция, биоидентификация.

DOI: 10.31857/S000523102210004X, **EDN:** AJXJDG

1. Введение

В недавнее время большой интерес получила разработка систем распознавания лиц, требующих малых вычислительных ресурсов. Это вызвано широким распространением встраиваемых систем, таких как домофоны с функцией доступа по лицу и камеры наружного наблюдения. Такие системы распознавания лиц основываются на нейросетевых моделях для мобильных вычислителей. Во время работы нейросетевая модель получает на вход изображение лица и восстанавливает по нему вектор фиксированной длины. В таком векторе закодирована информация о лице на изображении, и он называется биометрическим. Чем дальше вектора для различных людей друг от друга в векторном пространстве и ближе для различных изображений одного человека, тем выше качество работы нейронной сети. Близость векторов определяется согласно выбранной метрике, в рассматриваемых в данной работе моделях это косинусное расстояние.

К таким моделям относится архитектура MobileFaceNet [1], разработанная специально для распознавания лиц на устройствах с малой вычислительной

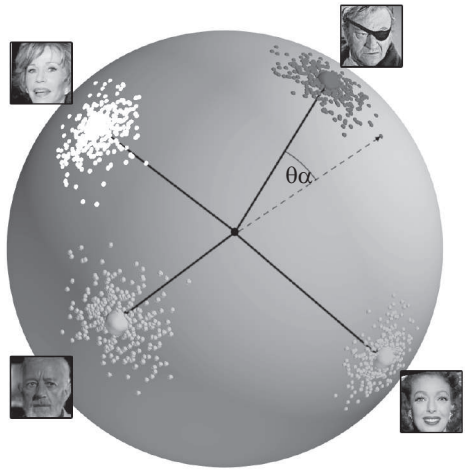


Рис. 1. Гиперсфера нормированных биометрических векторов. Различными цветами обозначены разные классы из обучающего набора данных: большими кругами обозначены центры классов, меньшими — изображения, относящиеся к классу. Для вычисления близости изображения к классу необходимо вычислить угол θ между векторами.

мощностью. В свою очередь использование функции Софтмакс с отступами [3–5] для обучения таких моделей позволяет достичь наибольшую точность в задачах идентификации и верификации по лицу. Нейросетевые модели для биоидентификации обучаются на классификацию с функцией Софтмакс большого количества классов, затем вектор, получаемый на предпоследнем слое, используется для кодирования и сравнения новых изображений. Добавление отступов в функцию Софтмакс в процессе обучения сети добавляет дополнительный отступ для векторов, относящихся к одному классу, что вынуждает сеть минимизировать угол от вектора до центра класса. Таким образом, сеть обучается минимизировать косинусное расстояние между изображениями одного человека (рис. 1).

Быстрые и компактные мобильные нейросетевые архитектуры достигают меньшей точности, чем серверные решения. В задачах биометрического доступа такое снижение точности может играть критическую роль. Для увеличения точности мобильных нейросетевых архитектур используется дистилляция [9]. Дистилляция нейронной сети — это метод передачи знаний от сети-учителя с большим числом обучаемых параметров в сеть-ученика с малым числом обучаемых параметров. В данной работе предлагается новый подход дистилляции, позволяющий сократить разницу в точности между сетью-учителем и сетью-учеником.

Идея предложенного подхода заключается в копировании последнего слоя сети, содержащего обученные центры классов обучающего набора данных, из сети-учителя в сеть-ученика и заморозке этого слоя во время всей процедуры дистилляции. Дистилляция заключается в обучении сети-ученика воспроизводить углы между центрами классов и биометрическими векторами для лиц, равные углам между соответствующими векторами и центрами классов

в сети-учителе. Такой подход позволяет сети-ученику лучше воспроизводить результат работы сети-учителя.

Основной вклад данной работы заключается в следующем:

1) предлагается новый метод для дистилляции нейронных сетей, обученных с применением функции Софтмакс с отступами;

2) предложенный метод позволяет сократить разницу в точности между сетью-учителем и сетью-учеником. Использование предложенного метода позволило для мобильной нейросетевой архитектуры получить наибольшую точность в сравнении с другими методами дистилляции для наборов данных LFW [6], AgeDB-30 [7] и MegaFace [8];

3) в данной работе производится прямое сравнение различных методов дистилляции. Программный код с реализацией рассматриваемых методов и проводимых экспериментов доступен в открытом доступе на сайте github.com [27].

2. Обзор литературы

2.1. Функция Софтмакс с отступами

Существует несколько вариаций функции Софтмакс (анг. softmax) с отступами, используемых при обучении нейронных сетей для систем распознавания лиц. Они включают подходы Cosface [5], Sphereface [4] и Arcface [3]. Все три подхода могут быть описаны общей формулой, задающей функцию ошибки классификации изображений лиц:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\left[\log\frac{e^{s(\cos(\theta_{y_i}m_1+m_2)-m_3)}}{e^{s(\cos(\theta_{y_i}m_1+m_2)-m_3)}+\sum\limits_{j=1,j\neq y_i}^{n}\left[e^{s\ \cos\theta_j}\right.}\right. . \tag{1}$$

Данные подходы могут быть получены из представленной формулы (1) подстановкой значений параметров m_1, m_2, m_3. Подход Sphereface получается при $m_1 = 4$, $m_2 = m_3 = 0$; подход Cosface получается при $m_1 = 1$; $m_2 = 0$; $m_3 = 0{,}35$; подход Arcface получается при $m_1 = 1$; $m_2 = 0{,}5$; $m_3 = 0$. В представленной формуле N — это количество изображений, используемых на каждом шаге стохастического градиентного спуска; θ_j — угол между векторами, соответствующими обучающему примеру с индексом i и центру класса с индексом j; y_i — соответствует индексу класса, к которому относится пример с индексом i согласно разметке; s — константа масштабирования, во всех случаях имеющая значение 64. Детальный анализ представленной формулы можно найти в статье Arcface [3]. Из рассмотренных подходов Arcface демонстрирует наибольшую точность на наборах данных LFW, AgeDB-30 и MegaFace.

2.2. Дистилляция

Дистилляция знаний от сети-учителя к сети-ученику была впервые предложена в [9]. Данный подход заключается в обучении сети-ученика с малым числом обучаемых параметров за счет передачи знаний от сети-учителя с большим числом обучаемых параметров. Под передачей знаний понимается

передача некоторой информации от обученной более точной модели для улучшения сходимости обучаемой малой модели. Ключевая идея подхода, предложенного в [9], заключается в передаче знаний через приближение сглаженного вероятностного распределения на выходе сети-учителя. Это достигалось за счет добавления делителя в формулу функции Софтмакс.

Некоторые исследования продолжают развивать идею использования сглаженного распределения вероятности в качестве разметки для обучения сети-ученика. Так, например, в [10] был предложен подход к дистилляции ансамбля нейронных сетей в одну сеть-ученика, используя данный метод. В [11] предложен подход к обучению сети-ученика с помощью сети-учителя, подверженной зашумленности выхода. В [12] сеть-ученик и сеть-учитель обучались с одинаковой параметризацией. В [25] предлагается механизм дистилляции через задание априорного распределения сети-ученика на основе апостериорного распределения сети-учителя с измененной структурой модели для совпадения пространства параметров. Также делаются шаги в сторону теоретического обоснования методов дистилляции через вероятностную интерпретацию [25, 26].

Другой подход к дистилляции — это дистилляция через скрытые слои нейронной сети. В [13] предлагается обучать сеть-ученика копировать распределение весов на скрытых слоях сети-учителя. В [14] промежуточные слои сетей ученика и учителя используются для регуляризации обучения. В [15] для регуляризации во время дистилляции используется ограничение на сохранение взаимоотношения между локальными объектами. Для этого L_2 расстояние между биометрическими векторами для сети-ученика минимизируется на основе информации от сети-учителя. В [16] предлагается относительная дистилляция знаний, которая штрафует за структурные различия во взаимоотношении между обучающими примерами.

Для дистилляции легковесных нейронных сетей для задачи распознавания лиц, обученных с применением функции Софтмакс с отступами, применяются следующие подходы: триплетная дистилляция [17], дистилляция по углу [18] и дистилляция на основе отступа [19].

В *триплетной дистилляции* сеть-ученик обучается с триплетной функцией ошибки, отступ в которой вычисляется основываясь на расстоянии между якорным и негативным примерами и якорным и позитивным примерами, предсказанными сетью-учителем.

В *дистилляции по углу* минимизируется угол между биометрическими векторами сетей ученика и учителя для каждого примера в обучающей выборке.

В *дистилляции на основе отступа* предлагается производить дистилляцию через сглаженное распределение вероятности. Для этого в формулу (1) добавляется деление на параметр T по аналогии с [9].

3. Предлагаемый подход

3.1. Описание сети-учителя и сети-ученика

В качестве сети-учителя рассматривалась нейронная сеть с архитектурой ResNet100 [20]. Выбор данной архитектуры был обусловлен тем,

Таблица 1. Сравнение параметров рассматриваемых нейросетевых архитектур. Время работы сетей замерялось для входного изображения размерностью $112 \times 112 \times 3$ на процессоре Intel Xeon(R) CPU E3-1270 v3 @ 3,50GHz×8

	ResNet100	MobileFaceNet
Число операций с плавающей точкой / 10^9	24,2	0,44
Размер / Мегабайт	261,2	5,3
Число обучаемых параметров / 10^6	52,56	1,19
Время работы / Миллисекунд	$401 \pm 25,7$	$42,2 \pm 5,48$

что она содержит большое число обучаемых параметров и, следовательно, позволяет достигать высокой точности в задаче распознавания лиц. В качестве сети-ученика была выбрана недавно предложенная архитектура MobileFaceNet(ReLU) [1], содержащая менее 1 млн параметров и разработанная специально для решения задачи распознавания лиц на мобильных процессорах. Данная архитектура состоит из блоков предложенных в MobileNetV2 [2], но позволяет обрабатывать изображения вдвое быстрее за счет уменьшения пространственной размерности изображения на ранних слоях и использования меньшего числа фильтров на промежуточных слоях блоков. В описываемых экспериментах была сделана следующая модификация данной архитектуры: размерность выходного биометрического вектора была увеличена с 256 до 512 элементов для совпадения с размерностью архитектуры ResNet100. В табл. 1 приводится сравнение рассматриваемых архитектур.

3.2. Описание метода

Обозначим биометрический вектор сети-ученика для изображения с индексом i как $x_{S_i} \in R^D$, где D — размерность вектора. И обозначим через $x_{T_i} \in R^D$ биометрический вектор соответствующего изображения для сети-учителя. Матрицы весов последнего слоя для сетей ученика и учителя будем обозначать соответственно $W_S \in R^{D \times n}$ и $W_T \in R^{D \times n}$, где n — количество классов в обучающем наборе данных. Столбец матрицы с индексом j, соответствующий центру класса y_i, к которому относится изображение с индексом i из обучающего набора данных, обозначается как $W_{S_j} \in R^D$ для сети-ученика и $W_{T_j} \in R^D$ для сети-учителя.

Подходы, основанные на добавлении отступа m в функцию Софтмакс, производят нормировку столбцов матрицы весов и биометрических векторов на 1: $||W_j|| = 1$ и $||x_i|| = 1$, где W_j — это j-й столбец матрицы W, а x_i — это биометрический вектор, соответствующий i-му обучающему примеру. Такая нормировка позволяет рассматривать результаты произведений векторов на последнем слое сети как косинусные расстояния $\cos(\theta_j)$ между биометрическими векторами и соответствующими центрами классов: ${W_j}^{\mathrm{T}} x_i = ||W_j|| \times$ $\times ||x_i|| \cos(\theta_j) = \cos(\theta_j)$ (рис. 1), где θ_j — угол между векторами, соответствующими обучающему примеру с индексом i и центру класса с индексом j. Далее в качестве частного случая обучения с использованием функции Софтмакс с отступами рассматривается метод Arcface [3], задаваемый формулой,

подробно описанной в разделе “Обзор литературы”:

$$L_{\text{ArcFace}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))}+\sum\limits_{j=1,j\neq y_i}^{n} e^{s\,\cos\theta_j}}. \tag{2}$$

Данный метод был выбран для рассмотрения, так как позволяет получать наибольшую точность среди методов обучения с применением функции Софтмакс с отступами. В подходе Arcface значение отступа m фиксируется равным 0,5, как предложено в [3]. В данной работе предлагается производить дистилляцию знаний от сети-учителя через вычисление значения отступа m для каждого изображения i. Предлагаемый метод дистилляции содержит две основные идеи:

— Центры классов, найденные сетью-учителем, используются в сети-ученике: $W_S = W_T$. Так как центры классов обучаемые параметры, более глубокая сеть способна сформировать более хорошее их положение на гиперсфере. Такое положение, при котором относящиеся к этим центрам кластеры векторов будут разделимы с большей точностью.

— Для дистилляции знаний используются вычисляемые значения m_i для каждого изображения. Они явным образом контролируют расстояние между биометрическими векторами x_{S_i} и соответствующими центрами классов W_{S_j}. Большее значение m_i способствует приближению вектора x_{S_i} к центру класса.

Отступ m_i вычисляется, основываясь на угле между биометрическим вектором сети-учителя $x_{T_i} \in R^D$ и соответствующим вектором $W_{T_j} \in R^D$ центра класса, задаваемого как y_i. Отступ m для изображения с индексом i вычисляется аналогично отступу в триплетной дистилляции:

$$m_i = f(a_i) = \frac{m_{\max} - m_{\min}}{a_{\max}} a_i + m_{\min}, \tag{3}$$

$$a_i = \frac{W_{T_j}^{\mathrm{T}} x_{T_i}}{||W_{T_j}|| \cdot ||x_{T_i}||}, \tag{4}$$

где параметры $m_{\max} = 0{,}5$ и $m_{\min} = 0{,}2$ задают максимальное и минимальное значение отступа, по аналогии с формулой триплетной дистилляции, предложенной в [17]. А значение параметра $a_{\max}$ вычисляется как максимальное значение угла a в мини-пакете изображений на каждом шаге обучения.

Формула (4) используется для вычисления углов a_i между центрами классов W_{T_j} и биометрическими векторами x_{T_i} сети-учителя (рис. 2,*а*). Углы a_i, получаемые от сети-учителя, используются для вычисления отступа $m_i = f(a_i)$ по формуле (3). Чем меньше значение угла между биометрическим вектором и центром класса, к которому он относится, тем больший отступ m_i используется для этого вектора при обучении сети-ученика (рис. 2,*б*). Большее значение отступа вынуждает сеть минимизировать угол от биометрического вектора до центра класса, чтобы компенсировать эффект от отступа. Тем самым векторы, которые были близки к центру класса в сети-учителе, будут близки к центру класса в сети-ученике.

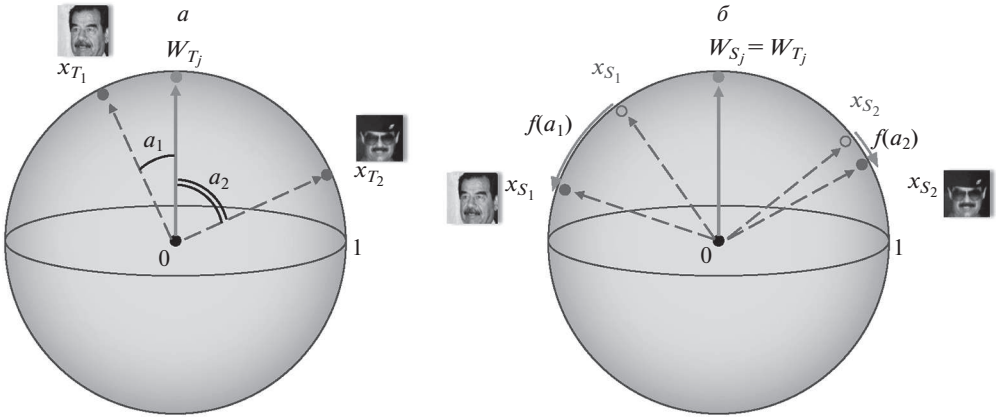


Рис. 2. *а* — Гиперсфера векторов сети-учителя. *б* — Вычисление сдвига для векторов на гиперсфере сети-ученика на основе углов до центра класса в сети-учителе.

Интуиция, лежащая в основе предлагаемого подхода, заключается в том, чтобы изменять биометрический вектор сети-ученика для уменьшения расстояния до центра класса, если соответствующий вектора сети-учителя находится близко к центру класса. Это позволяет осуществлять передачу знаний от сети-учителя к сети-ученику более эффективно потому, что сеть-ученик концентрируется на более уверенно классифицированных сетью-учителем изображениях.

Предлагаемый подход к дистилляции позволяет передавать информацию об относительном положении векторов на гиперсфере, не вводя явных ограничений на биометрические векторы сети-ученика.

4. Эксперименты

4.1. Детали реализации

Предобработка данных. Для обнаружения и выравнивания лиц на изображениях применялся метод MTCNN [21]. В качестве обучающего набора данных использовался набор данных для классификации по лицу MS1MV2 [3]. Данный набор данных — это очищенный в полуавтоматическом режиме набор данных MS-Celeb-1M [22], предложенный в работе, посвященной Arcface. Рассматриваемый набор данных содержит 5,8 млн изображений для 85 тыс. классов.

После процедуры выравнивания лиц, выполненной с применением ключевых точек найденных MTCNN, получаются изображения размером 112×112 пикселей. Значения яркости пикселей полученных изображений затем нормируются до диапазона $[-1, 1]$.

Процедура обучения. В качестве сети-учителя была обучена нейронная сеть с архитектурой ResNet100 и функцией Arcface. Сравнение всех методов дистилляции проводилось в одинаковых условиях, где передача знаний осуществлялась от ResNet100 к MobileFaceNet(ReLU). Дистилляция производи-

лась со следующими значениями гиперпараметров: размер мини-пакета изображений равнялся 512, шаг обучения равнялся 0,1 и увеличивался в 10 раз после 10 000, 160 000 и 200 000 итераций. Оптимизация обучаемых параметров выполнялась алгоритмом SGD [24] со значением моментов, равным 0,9. Коэффициент регуляризации обучаемых параметров равнялся $5e-4$. Значения максимального и минимального возможных значений отступа фиксировались равными $m_{\max} = 0{,}5$ и $m_{\min} = 0{,}2$. Значение параметра s метода Arcface не изменялось и равнялось 64. Описанные далее эксперименты основываются на официальной реализации Arcface авторами на фреймворке MXNet.

Для сравнения предложенного метода дистилляции с существующими подходами были реализованы триплетная дистилляция [17], дистилляция по углу [18] и дистилляция на основе отступа [19] на фреймворке MXNet. Нейронные сети дистиллировались с помощью этих методов с указанными в опорных статьях параметрами. Реализация этих методов на MXNet доступна в репозитории данной работы на GitHub [27].

Процедура тестирования. Обученные с помощью дистилляции нейронные сети для распознавания лиц тестировались на задачах верификации и идентификации.

Верификация. Для замеров точности обученных нейронных сетей на задаче верификации использовались наборы данных LFW и AgeDB-30. Каждый набор данных содержит порядка 3000 позитивных и 3000 негативных пар изображений. В процедуре тестирования обученная нейронная сеть использовалась для получения биометрических векторов для пары изображений лиц. Верификация производилась основываясь на косинусном расстоянии между векторами. Точность измерялась как процент верно верифицированных пар изображений.

Идентификация. Наиболее представительным и сложным протоколом тестирования для задачи идентификации лиц является MegaFace. В набор данных MegaFace входит миллион изображений лиц для 690 000 человек для формирования негативных примеров. И 100 000 изображений для 530 людей из набора данных FaceScrub [23], идентификацию которых по базе лиц необходимо произвести. Значением метрики качества является топ-1 точность для задачи идентификации по базе лиц с миллионом негативных примеров.

4.2. Результаты тестирования

Как показано в табл. 2, обученная с функцией Arcface, сеть-учитель достигает точности 99,76% для набора данных LFW и 98,21% для AgeDB-30. сеть-ученик, обученная с функцией Arcface, достигает 99,51% для набора данных LFW и 96,13% для AgeDB-30. Предложенный подход позволяет сократить разницу в точности сильнее, чем остальные рассмотренные подходы: до 99,61% для LFW и 96,55% для AgeDB-30.

В табл. 3 представлены результаты замеров точности для задачи идентификации по протоколу MegaFace с миллионом негативных примеров в базе лиц. Для данного протокола тестирования сеть-учитель достигает точности 98,35%, а сеть-ученик 90,62%. Предложенный подход позволяет увеличить точность идентификации до 91,70%. Также для задачи идентификации бы-

Таблица 2. Точность верификации на наборах данных LFW и AgeDB-30. В экспериментах использовалась версия MobileFaceNet с функцией активации ReLU

Архитектура	Метод обучения	LFW %	AgeDB-30 %
ResNet100 (сеть-учителя)	ArcFace [3]	99,76	98,21
MobileFaceNet (сеть-ученик)	ArcFace [3]	99,51	96,13
MobileFaceNet	Триплетная дистилляция по L2 [17]	99,56	96,23
MobileFaceNet	Триплетная дистилляция по cos [17]	99,55	95,60
MobileFaceNet	Дистилляция с отступом для T=4 [19]	99,41	96,01
MobileFaceNet	Дистилляция по углу [18]	99,55	96,01
MobileFaceNet	Предлагаемый метод	**99,61**	**96,55**

Таблица 3. Точность идентификации с использованием протокола MegaFace с 1 млн негативных примеров. В экспериментах использовалась версия MobileFaceNet с функцией активации ReLU

Архитектура	Метод-обучения	MegaFace %
ResNet100 (сеть-учитель)	ArcFace [3]	98,35
MobileFaceNet (сеть-ученик)	ArcFace [3]	90,62
MobileFaceNet	Триплетная дистилляция по L2 [17]	89,10
MobileFaceNet	Триплетная дистилляция по cos [17]	86,52
MobileFaceNet	Дистилляция с отступом для T=4 [19]	90,77
MobileFaceNet	Дистилляция по углу [18]	90,73
MobileFaceNet	Предлагаемый метод	**91,70**

ло замечено уменьшение точности для метода триплетной дистилляция, но данный метод показал хорошую точность для задачи верификации.

5. Оценка метода

Для более детального анализа предлагаемого подхода была проведена его оценка методом удаления различных элементов. При удалении элементов подхода оценивалось их влияние на точность верификации на наборе данных LFW. В табл. 4 оценивается влияние следующих аспектов метода:

— Копирование центров — копирование центров классов, представленных обучаемыми параметрами последнего слоя нейронной сети, от сети-учителя к сети-ученику: $W_S = W_T$.

— Использование m_i вместо m-использование вычисляемых m_i для дистилляции через Arcface вместо фиксированного $m = 0{,}5$.

— Фиксирование центров — пометка центров классов сети-ученика W_S как необучаемые параметры. Чтобы в процессе дистилляции они оставались равными центрам классов сети-учителя W_T.

Наибольший прирост в точности в 0,9% вызван копированием центров классов сети-учителя в сеть-ученика и дальнейшая пометка их как необуча-

Таблица 4. Оценка точности метода на наборе данных LFW удалением различных его элементов

Копирование центров	Использование m_i вместо m	Фиксирование центров	LFW %
√	√	√	**99,61**
√	√		98,31
√			99,55
	√		99,43
√		√	99,60

емые параметры. Самостоятельное обучение сетью-учеником центров классов W_S во всех сценариях ведет к уменьшению точности на наборе данных LFW. Соответственно ключевую роль в предложенном методе дистилляции играет копирование центров классов. Использование адаптивного отступа m_i позволяет дополнительно увеличить точность получаемой модели.

6. Заключение

В данной работе был предложен подход к дистилляции нейронных сетей, обученных для задачи распознавания лиц с функцией Софтмакс с отступами. Была продемонстрирована эффективность использования центров классов сети-учителя в сети-ученике. Было проведено сравнение предложенного метода с другими методами дистилляции для нейронных сетей, использующих Софтмакс с отступами. Описанный в данной работе подход позволяет получить лучшую точность на наборах данных LFW и AgeDB-30 для задачи верификации и для MegaFace для задачи идентификации.

Предложенный метод дистилляции может применяться для увеличения точности нейросетевых моделей с малым числом параметров для встраиваемых устройств. Таких, например, как камеры наружного наблюдения или домофоны с функцией доступа по лицу.

СПИСОК ЛИТЕРАТУРЫ

1. *Chen S., Liu Y., Gao X., Han Z.* Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices // Chinese Conference on Biometric Recognition. Springer, Cham, 2018. С. 428–438.
2. *Sandler M., Howard A., Zhu M., Zhmoginov A., Chen L.C.* Mobilenetv2: Inverted residuals and linear bottlenecks // Proceedings of the IEEE conference on computer vision and pattern recognition. 2018. С. 4510–4520.
3. *Deng J., Guo J., Xue N., Zafeiriou S.* Arcface: Additive angular margin loss for deep face recognition // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019. С. 4690–4699.
4. *Liu W., Wen Y., Yu Z., Li M., Raj B., Song L.* Sphereface: Deep hypersphere embedding for face recognition // Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. С. 212–220.

5. *Wang H., Wang Y., Zhou Z., Ji X., Gong D., Zhou J., Li Z., Liu W.* Cosface: Large margin cosine loss for deep face recognition // Proceedings of the IEEE conference on computer vision and pattern recognition. 2018. С. 5265–5274.
6. *Huang G.B., Mattar M., Berg T., Learned-Miller E.* Labeled faces in the wild: A database forstudying face recognition in unconstrained environments // Workshop on faces in'Real-Life'Images: detection, alignment, and recognition. 2008.
7. *Moschoglou S., Papaioannou A., Sagonas C., Deng J., Kotsia I., Zafeiriou S.* Agedb: the first manually collected, in-the-wild age database // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2017. С. 51–59.
8. *Kemelmacher-Shlizerman I., Seitz S.M., Miller D., Brossard E.* The megaface benchmark: 1 million faces for recognition at scale // Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. С. 4873–4882.
9. *Hinton G., Vinyals O., Dean J.* Distilling the knowledge in a neural network // arXiv preprint arXiv:1503.02531. 2015.
10. *Fukuda T., Suzuki M., Kurata G., Thomas S., Cui J., Ramabhadran B.* Efficient Knowledge Distillation from an Ensemble of Teachers // Interspeech. 2017. С. 3697–3701.
11. *Sau B.B., Balasubramanian V.N.* Deep model compression: Distilling knowledge from noisy teachers // arXiv preprint arXiv:1610.09650. 2016.
12. *Furlanello T., Lipton Z., Tschannen M., Itti L., Anandkumar A.* Born again neural networks // International Conference on Machine Learning. PMLR, 2018. С. 1607–1616.
13. *Huang Z., Wang N.* Like what you like: Knowledge distill via neuron selectivity transfer // arXiv preprint arXiv:1707.01219. 2017.
14. *Romero A., Ballas N., Kahou S.E., Chassang A., Gatta C., Bengio Y.* Fitnets: Hints for thin deep nets // arXiv preprint arXiv:1412.6550. 2014.
15. *Chen H., Wang Y., Xu C., Xu C., Tao D.* Learning student networks via feature embedding // IEEE Transactions on Neural Networks and Learning Systems. 2020. Т. 32. No. 1. С. 25–35.
16. *Park W., Kim D., Lu Y., Cho M.* Relational knowledge distillation // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019. С. 3967–3976.
17. *Feng Y., Wang H., Hu H.R., Yu L., Wang W., Wang S.* Triplet distillation for deep face recognition // 2020 IEEE International Conference on Image Processing (ICIP). IEEE. 2020. С. 808–812.
18. *Duong C.N., Luu K., Quach K.G., Le N.* Shrinkteanet: Million-scale lightweight face recognition via shrinking teacher-student networks // arXiv preprint arXiv:1905.10620. 2019.
19. *Nekhaev D., Milyaev S., Laptev I.* Margin based knowledge distillation for mobile face recognition // Twelfth International Conference on Machine Vision (ICMV 2019). – International Society for Optics and Photonics, 2020. Т. 11433. С. 114330O.
20. *He K., Zhang X., Ren S., Sun J.* Deep residual learning for image recognition // Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. С. 770–778.
21. *Zhang K., Zhang Z., Li Z., Qiao Y.* Joint face detection and alignment using multitask cascaded convolutional networks // IEEE Signal Processing Letters. 2016. Т. 23. № 10. С. 1499–1503.
22. *Guo Y., Zhang L., Hu Y., He X., Gao J.* Ms-celeb-1m: A dataset and benchmark for large-scale face recognition // European conference on computer vision. Springer, Cham, 2016. С. 87–102.

23. *Ng H.W., Winkler S.* A data-driven approach to cleaning large face datasets // 2014 IEEE international conference on image processing (ICIP). IEEE, 2014. С. 343–347.

24. *Robbins H., Monro S.* A stochastic approximation method // The annals of mathematical statistics. 1951. С. 400–407.

25. *Грабовой А.В., Стрижов В.В.* Байесовская дистилляция моделей глубокого обучения // АиТ. 2021. № 11. С. 16–29.

    Grabovoy A.V., Strijov V.V. Bayesian Distillation of Deep Learning Models // Autom. Remote Control. 2021. Т. 82. No. 11. С. 1846–1856.

26. *Грабовой А.В., Стрижов В.В.* Вероятностная интерпретация задачи дистилляции // АиТ. 2022. № 1. С. 150–168.

    Grabovoy A.V., Strijov V.V. Probabilistic Interpretation of the Distillation Problem // Autom. Remote Control. 2022. Т. 83. No. 1. С. 123–137.

27. MarginDistillation: distillation for margin-basedsoftmax: https://github.com/davidsvitov/margindistillation (дата обращения: 08.01.2022).

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 10.01.2022

После доработки 08.05.2022

Принята к публикации 29.06.2022

© 2022 г. А.И. БАЗАРОВА (bazarova.ai@phystech.edu),
А.В. ГРАБОВОЙ (grabovoy.av@phystech.edu)
(Московский физико-технический институт),
В.В. СТРИЖОВ, д-р физ.-мат. наук (strijov@phystech.edu)
(Вычислительный центр им. А.А. Дородницына ФИЦ ИУ РАН, Москва)

АНАЛИЗ СВОЙСТВ ВЕРОЯТНОСТНЫХ МОДЕЛЕЙ В ЗАДАЧАХ ОБУЧЕНИЯ С ЭКСПЕРТОМ[1]

Работа посвящена построению интерпретируемых моделей машинного обучения. Решается задача аппроксимации набора фигур на контурном изображении. Вводятся предположения, что фигуры являются кривыми второго порядка. При аппроксимации фигур используются информация о типе, расположении и форме кривых, а также о множестве их возможных преобразований. Такая информация называется *экспертной*, а метод машинного обучения, основанный на экспертной информации, называется *обучение с экспертом*. Предполагается, что набор фигур аппроксимируется набором *локальных моделей*. Каждая локальная модель, основанная на экспертной информации, аппроксимирует одну фигуру на контурном изображении. Для построения моделей предлагается отображать кривые второго порядка в пространство признаков, в котором каждая локальная модель является линейной. Таким образом, кривые второго порядка аппроксимируются набором линейных моделей. В вычислительном эксперименте рассматривается задача аппроксимации радужной оболочки глаза на контурном изображении.

Ключевые слова: смесь экспертов, экспертное обучение, линейные модели, интерпретируемые модели.

DOI: 10.31857/S0005231022100051, **EDN:** AKCUFQ

1. Введение

Современные решения задачи классификации изображений на основе сетей глубокого обучения ResNet, VGG, Intercept [1] представляют собой плохо интерпретируемые модели [2]. В [3, 4] показано, что сети глубокого обучения являются не устойчивыми даже к небольшому шуму в данных [5].

В данной работе предлагается метод *обучения с экспертом*. Метод предполагает использование экспертной информации для повышения качества аппроксимации, а также для получения интерпретируемых моделей машинного обучения. Предметные знания экспертов о данных называются *экспертная информация*. Предполагается, что использование экспертной информации позволяет аппроксимировать выборку простыми интерпретируемыми мо-

[1] Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований в рамках научных проектов 20-37-90050, 19-07-01155 и проекта Национальной технологической инициативы 13/1251/2018.

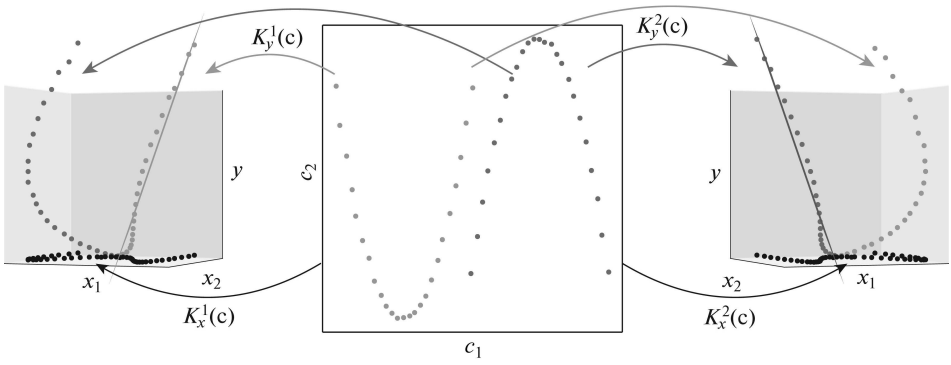


Рис. 1. Визуализация экспертной информации в случае с двумя экспертами. Слева направо: экспертная информация первого эксперта; исходные данные; экспертная информация второго эксперта.

делями, такими как линейные модели. Метод машинного обучения, учитывающий экспертные знания при построении моделей, называется *обучение с экспертом*. В данной работе решается задача аппроксимации кривых второго порядка на контурном изображении. В работе анализируются кривые второго порядка, так как они описываются линейными моделями. Параметры кривых второго порядка необходимо восстановить в задаче распознавания радужной оболочки глаза [6–8], в задаче описания трека частицы в адронном коллайдере [9]. Экспертная информация о кривой второго порядка отображает точки на плоскости в новое признаковое описание объекта. Каждая кривая аппроксимируется одной линейной моделью. Модель, аппроксимирующая кривую, называется *локальной моделью*. Для аппроксимации всего контурного изображения требуется аппроксимация нескольких кривых второго порядка с использованием локальных моделей. Вводятся следующие ограничения на изображения: а) изображение состоит только из кривых второго порядка; б) изображение аппроксимируется небольшим числом кривых второго порядка; в) известно число и тип кривых на изображении.

На рис. 1 показан пример кривых второго порядка, а также экспертная информация о кривых. Рассматривается пример двух кривых, которые задаются своим цветом. На центральном изображении показаны точки, лежащие на кривых, а на рисунках справа и слева представлены экспертные признаковые описания рассмотренных кривых. В каждом из экспертных признаковых описаний получаем, что одна из кривых аппроксимируется линейной моделью, а вторая является шумом относительно построенной линейной модели. Отображение $K_x^1(\mathbf{c}), K_x^2(\mathbf{c}), K_y^1(\mathbf{c}), K_y^2(\mathbf{c})$ описывается на основе экспертной информации. На рис. 1 слева показана экспертная информация о первом эксперте. С использованием этой информации первая кривая аппроксимируется линейной моделью, а вторая кривая представляет собой шум. На рис. 1 справа показана экспертная информация второго эксперта. С использованием этой информации вторая кривая аппроксимируется линейной моделью, а первая кривая представляет собой шум.

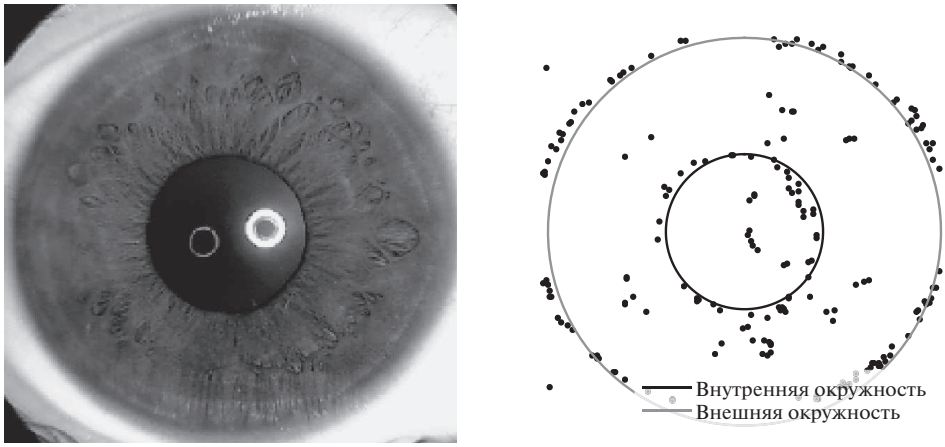


Рис. 2. Пример изображения радужной оболочки глаза и его контурное представление. Слева: изображение радужной оболочки глаза. Справа: контурное изображение радужной оболочки и аппроксимирующие заданное изображение окружности.

При аппроксимации нескольких кривых на одном контурном изображении строится мультимодель. Примером мультимоделей является случайный лес [10], бустинг деревьев [11], смесь экспертов [12]. В данной работе в качестве мультимодели рассматривается смесь экспертов. Смесь экспертов — это мультимодель, которая линейно взвешивает локальные модели, аппроксимирующие часть выборки. Значения весовых коэффициентов зависят от объекта, для которого делается прогноз. Для решения задачи используется вариационный EM-алгоритм [13–16]. Смесь экспертов имеет множество применений в ряде приложений. В [17] решается задача классификации текстов. В [18–24] смесь экспертов используется для прогнозирования временных рядов, для распознавания речи, распознавания повседневной деятельности человека и прогнозирования стоимости ценных бумаг. В [14] рассматривалась смесь экспертов для решения задачи распознавания рукописных цифр на изображениях.

В качестве примера рассматривается задача аппроксимации изображения радужной оболочки глаза. На рис. 2 показан пример изображения, которое необходимо аппроксимировать. В данной работе рассматривается обработанное изображение в виде контурного изображения.

Для задачи аппроксимации радужной оболочки используется экспертная информация: радужная оболочка глаза аппроксимируется двумя концентрическими окружностями. Экспертная информация используется для построения описания признаков точек плоскости, а также для построения регуляризатора в функции оптимизации. Часть функции ошибок для оптимизации, использующая экспертную информацию, называется регуляризатором. Таким образом, информация о том, что изображение является окружностями, задается видом признакового описания, а информация о том, что окружности концентрические, задается с помощью специального регуляризатора.

Вычислительный эксперимент анализирует качество аппроксимации контурного изображения в зависимости от экспертной информации и уровня шума в синтетически сформированных данных. Проведен анализ качества аппроксимации радужной оболочки в зависимости от объема экспертной информации, которая использовалась для построения модели. Каждое аппроксимированное изображение представляет собой отдельный набор точек, которые необходимо аппроксимировать.

2. Постановка задачи восстановления параметров кривых второго порядка на изображении

Задано бинарное изображение

$$\mathbf{M} \in \{0,1\}^{m_1 \times m_2},$$

где 1 соответствует черной точке изображения, а 0 соответствует белой точке фона. С использованием изображения $\mathbf{M}$ строится выборка $\mathbf{C}$, элементами которой являются координаты (x_i, y_i) черных точек:

$$\mathbf{C} \in \mathbb{R}^{N \times 2}.$$

Каждый эксперт предполагает, что изображение состоит из кривой второго порядка Ω. Пусть для множества точек $\mathbf{C} \in \mathbb{R}^{N \times 2}$, образующих кривую Ω, задана экспертная информация о фигуре $E(\Omega)$. Множество $E(\Omega)$ состоит из ожидаемого экспертом образа Ω и множества его допустимых преобразований. На основе экспертного описания введем отображения в задачу аппроксимации:

$$K_x\big(E(\Omega)\big) : \mathbb{R}^2 \to \mathbb{R}^n, \quad K_y\big(E(\Omega)\big) : \mathbb{R}^2 \to \mathbb{R}, \tag{1}$$

где K_x — отображение объектов в признаковое описание объектов, n — число признаков, а K_y — отображение в целевую переменную для аппроксимации. Применяя отображения K_x, K_y для выборки $\mathbf{C}$ поэлементно, получаем:

$$K_x\big(E(\Omega), \mathbf{c}\big) = \mathbf{x}, \quad K_y\big(E(\Omega), \mathbf{c}\big) = y, \tag{2}$$

где $\mathbf{c} = (x_i, y_i)$ — точка из множества точек $\mathbf{C}$.

Применяя отображения (2) к исходному множеству точек $\mathbf{C}$, получаем выборку

$$\mathfrak{D} = \{(\mathbf{x}, y) \mid \forall \mathbf{c} \in \mathbf{C}\ \mathbf{x} = K_x(\mathbf{c}),\ y = K_y(\mathbf{c})\}. \tag{3}$$

Получаем, что исходная задача аппроксимации кривой Ω сводится к аппроксимации выборки $\mathfrak{D}$. В работе предполагается, что выборка $\mathfrak{D}$ аппроксимируется линейной моделью:

$$g(\mathbf{x}, \mathbf{w}) = \mathbf{x}^{\mathsf{T}}\mathbf{w},$$

где $\mathbf{w}$ — вектор параметров для аппроксимации.

Для нахождения оптимального вектора параметров $\hat{\mathbf{w}}$ решается оптимизационная задача

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}\in\mathbb{R}^n} \sum_{(\mathbf{x},y)\in\mathfrak{D}} \|g(\mathbf{x},\mathbf{w}) - y\|_2^2.$$

Задача аппроксимации исходной кривой Ω сводится к решению задачи линейной регрессии, т.е. к нахождению компонент вектора $\hat{\mathbf{w}}$.

В случае, когда на изображении K кривых второго порядка $\Omega_1, \ldots, \Omega_K$, где для каждой задана экспертная информация $E_k = E(\Omega_k)$, $k \in \{1, \ldots, K\}$ получаем задачу построения мультимодели, которая называется смесью K экспертов.

Определение 1. Назовем мультимодель f смесью K экспертов

$$f = \sum_{k=1}^{K} \pi_k(\mathbf{x},\mathbf{V}) g_k(\mathbf{w}_k), \quad \pi_k(\mathbf{x},\mathbf{V}) : \mathbb{R}^{n\times|\mathbf{V}|} \to [0,\,1], \quad \sum_{k=1}^{K} \pi_k(\mathbf{x},\mathbf{V}) = 1,$$

где g_k — локальная модель, называемая экспертом. Вектор $\mathbf{x}$ — признаковое описание объекта, π_k — шлюзовая функция, вектор $\mathbf{w}_k$ — параметры локальной модели, $\mathbf{V}$ являются параметрами шлюзовой функции.

Для каждой кривой второго порядка Ω_k заданы отображения (1). Введем обозначения: $K_x^k(\mathbf{c}) = K_x(\Omega_k, \mathbf{c})$ и $K_y^k(\mathbf{c}) = K_y(\Omega_k, \mathbf{c})$. Используя локальные линейные модели g_k, строим мультимодель f, описывающую кривые $\Omega_1, \ldots, \Omega_K$ на изображении $\mathbf{M}$:

$$f = \sum_{\mathbf{c}\in\mathbf{C}} \sum_{k=1}^{K} \pi_k(\mathbf{c},\mathbf{V}) g_k(K_x^k(\mathbf{c}), \mathbf{w}_k), \tag{4}$$

где π_k — шлюзовая функция. В работе рассматривается случай, когда $\mathbf{x} = K_x^1(\mathbf{c}) = \ldots = K_x^K(\mathbf{c})$. В этом случае выражение (4) переписывается в виде

$$f = \sum_{\mathbf{c}\in\mathbf{C}} \sum_{k=1}^{K} \pi_k(\mathbf{x},\mathbf{V}) g_k(\mathbf{x}, \mathbf{w}_k),$$

где шлюзовая функция π_k имеет вид

$$\pi_k(\mathbf{x},\mathbf{V}) : \mathbb{R}^{n\times|\mathbf{V}|} \to [0,\,1], \quad \sum_{k=1}^{K} \pi_k(\mathbf{x},\mathbf{V}) = 1,$$

где $\mathbf{V}$ — параметры функции шлюза, а g_k — локальная модель.

В работе рассматривается следующий вид шлюзовой функции:

$$\boldsymbol{\pi}(\mathbf{x}, \mathbf{V}) = \operatorname{softmax}\big(\mathbf{V}_1^{\mathsf{T}}\boldsymbol{\sigma}(\mathbf{V}_2^{\mathsf{T}}\mathbf{x})\big),$$

где $\mathbf{V} = \{\mathbf{V}_1, \mathbf{V}_2\}$ — параметры шлюзовой функции, $\mathbf{V}_1 \in \mathbb{R}^{p\times k}$, $\mathbf{V}_2 \in \mathbb{R}^{n\times p}$.

Для нахождения оптимальных параметров мультимодели решается оптимизационная задача

$$\mathcal{L} = \sum_{(\mathbf{x},y)\in\mathfrak{D}} \sum_{k=1}^{K} \Big[\pi_k(\mathbf{x}, \mathbf{V})(y - \mathbf{w}_k^{\mathsf{T}}\mathbf{x})^2 + R\big(\mathbf{V}, \mathbf{W}, E(\Omega)\big) \to \min_{\mathbf{V},\mathbf{W}}, \tag{5}$$

где $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_k]$ — параметры локальных моделей, $R\big(\mathbf{V}, \mathbf{W}, E(\Omega)\big)$ — параметры регуляризации, основанные на экспертной информации.

3. Построение признакового описания фигур

Единое пространство для кривых второго порядка. Произвольная кривая второго порядка, главная ось которой не параллельна оси ординат, задается уравнением

$$x^2 = B'xy + C'y^2 + D'x + E'y + F',$$

где на коэффициенты B', C' накладываются ограничения, зависящие от типа кривой. Выражение (2) для кривой второго порядка принимает вид

$$K_x\big(\mathbf{c}_i\big) = \big[x_i y_i, y_i^2, x_i, y_i, 1\big], \quad K_y\big(\mathbf{c}_i\big) = x_i^2,$$

откуда получаем задачу линейной регрессии для восстановления параметров B', C', D', E', F' по заданой выборке.

Окружность. В качестве частного случая кривой второго порядка рассмотривается окружность. Пусть (x_0, y_0) — центр окружности, которую требуется восстановить на бинарном изображении $\mathbf{M}$, а r — ее радиус. Элементы выборки $(x_i, y_i) \in \mathbf{C}$ представляют собой геометрическое место точек, которое аппроксимируется уравнением окружности:

$$(2x_0)x_i + (2y_0)y_i + \big(r^2 - x_0^2 - y_0^2\big) = x_i^2 + y_i^2.$$

Тогда выражение (2) принимает следующий вид:

$$K_x(\mathbf{c}_i) = [x_i, y_i, 1] = \mathbf{x}, \quad K_y(\mathbf{c}_i) = x_i^2 + y_i^2 = y.$$

Получаем задачу линейной регрессии (3). Компоненты вектора $\mathbf{w} = [w_0, w_1, w_2]^{\mathsf{T}}$ восстанавливают параметры окружности:

$$x_0 = \frac{w_0}{2}, \quad y_0 = \frac{w_1}{2}, \quad r = \sqrt{w_3 + x_0^2 + y_0^2}.$$

4. Композиция фигур

Для построения композиции фигур используется уравнение (5), которое принимает вид

$$\mathcal{L} = \sum_{\mathbf{c}\in\mathbf{C}} \sum_{k=1}^{K} \pi_k(\mathbf{c}, \mathbf{V}) \left(K_y^k(\mathbf{c}) - \mathbf{w}_k^{\mathsf{T}} K_x^k(\mathbf{c}) \right)^2 + R\big(\mathbf{V}, \mathbf{W}, E(\Omega)\big) \to \min_{\mathbf{V},\mathbf{W}},$$

где K_x^k, K_y^k — экспертное представление k-го эксперта. Предполагая, что все кривые на изображении описываются одним признаковым описанием $\mathbf{x} = K_x^1(\mathbf{c}) = \ldots = K_x^K(\mathbf{c})$, $x = K_y^1(\mathbf{c}) = \cdots = K_y^K(\mathbf{c})$, получаем оптимизационную задачу:

$$\mathcal{L} = \sum_{(\mathbf{x},y)\in\mathfrak{D}} \sum_{k=1}^{K} \pi_k(\mathbf{x}, \mathbf{V}) \left(y - \mathbf{w}_k^{\mathsf{T}} \mathbf{x} \right)^2 + R\big(\mathbf{V}, \mathbf{W}, E(\Omega)\big) \to \min_{\mathbf{V},\mathbf{W}}, \tag{6}$$

где регуляризатор R учитывает дополнительные ограничения параметров локальных моделей. Для решения задачи оптимизации (6) используется EM-алгоритм, описанный в [16].

5. Вычислительный эксперимент

Проведен вычислительный эксперимент по анализу качества аппроксимации кривых второго порядка на изображении. Эксперимент разделен на несколько частей. Первая часть описывает эксперимент с несколькими окружностями на изображении. Во второй части анализируется сходимость метода в зависимости от уровня шума в данных и от заданной экспертной информации. В третьей части проводится эксперимент по аппроксимации радужной оболочки глаза.

5.1. Эксперимент по восстановлению параметров окружности

В этой части эксперимента анализируется аппроксимация нескольких кривых второго порядка предложенной мультимоделью. Аппроксимируется сгенерированная синтетическая выборка. Выборка состоит из трех произвольных непересекающихся окружностей. К окружностям добавлен шум. Шум добавлялся к каждой точке окружности в отдельности, а также в выборку добавлялись случайные точки, не относящиеся к окружности.

На рис. 3 показан результат построения ансамбля локальных аппроксимирующих моделей. Каждая локальная модель аппроксимирует одну окружность. Видно, что при добавлении шума качество аппроксимации падает. На рис. 4 показан график зависимости радиуса окружностей r и их центров (x_0, y_0) от номера итерации.

5.2. Анализ качества аппроксимации для выборки с разным уровнем шума

В этой части эксперимента анализируется качество аппроксимации S в зависимости от уровня шума β в данных и от параметра априорных рас-

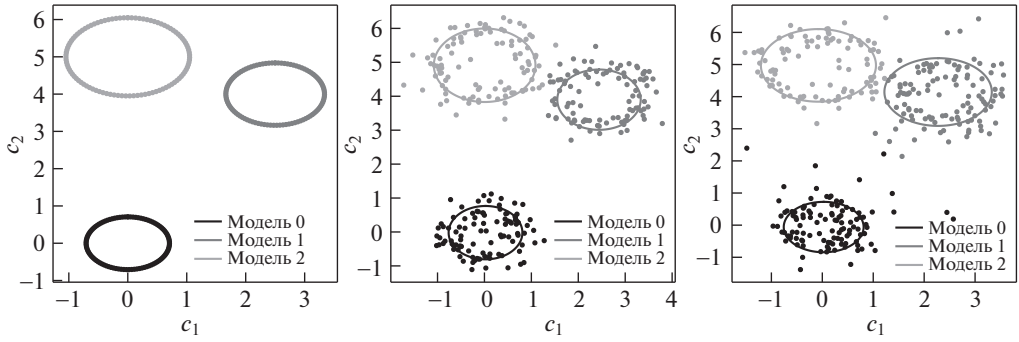


Рис. 3. Мультимодель в зависимости от уровня шума в выборке. Слева направо: окружности без шума; шум в радиусе круга; шум в радиусе круга, шум по всему изображению.

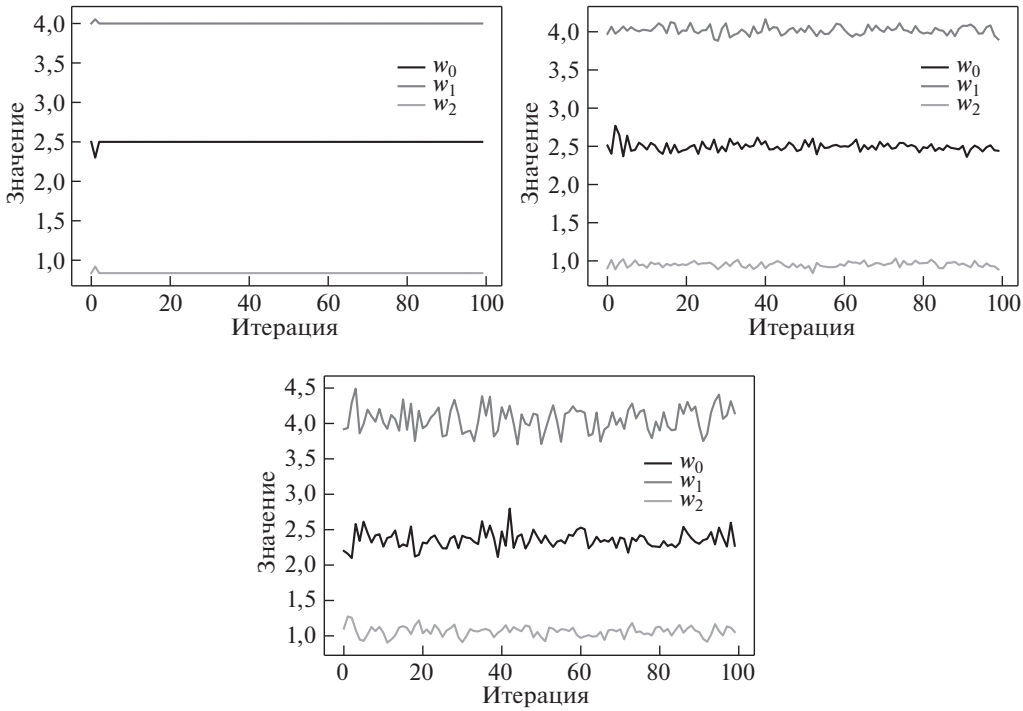


Рис. 4. Зависимость параметров r, x_0 и y_0 от номера итерации в зависимости от уровня шума в выборке. Слева направо: окружности без шума; шум в радиусе круга; шум в радиусе круга, шум по всему изображению.

пределений γ. Выборка сгенерирована в таком виде: сначала случайным образом выбираются два вектора параметров $\mathbf{w}_1^{\text{true}}$ и $\mathbf{w}_2^{\text{true}}$ — коэффициенты двух парабол. Векторы $\mathbf{w}_1^{\text{true}}$ и $\mathbf{w}_2^{\text{true}}$ используются для генерации точек x_i и y_i с добавлением нормального шума $\varepsilon \sim \mathcal{N}\big(0, \beta\big)$. При обучении мультимодели учитывается априорное распределение параметров $\mathbf{w}_1 \sim \mathcal{N}\big(\mathbf{w}_1^{\text{true}}, \gamma\mathbf{I}\big)$, $\mathbf{w}_2 \sim \mathcal{N}\big(\mathbf{w}_2^{\text{true}}, \gamma\mathbf{I}\big)$.

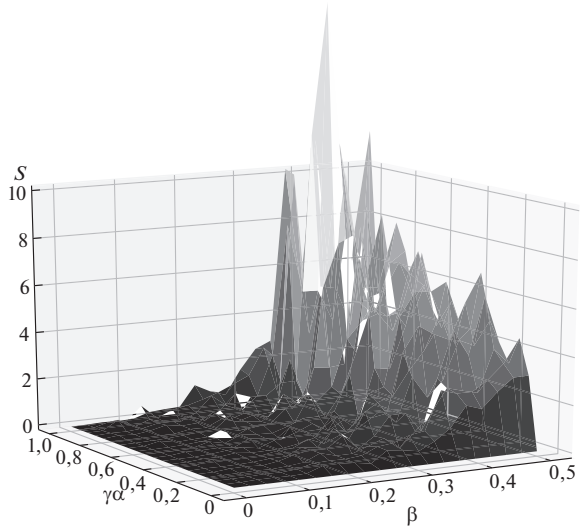


Рис. 5. Зависимость моделей от уровня шума β в данных, а также от дисперсии априорного распределения γ.

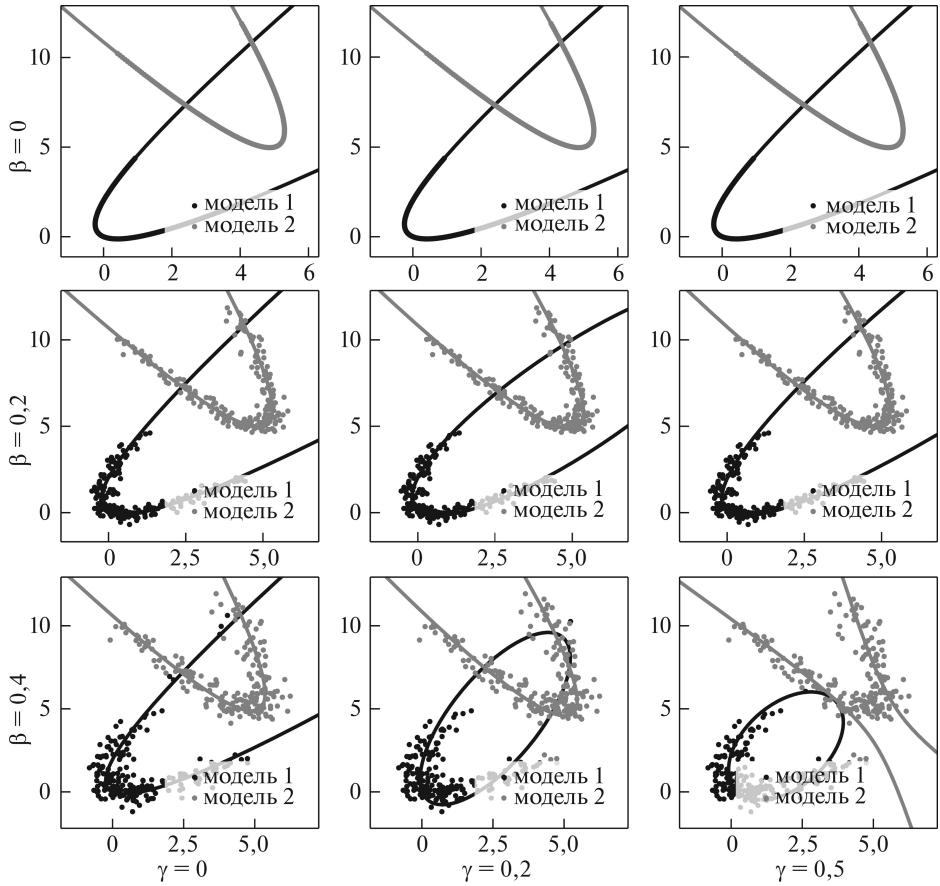


Рис. 6. Результат аппроксимации данных с разным уровнем шума β и по дисперсии априорного распределения γ.

Рассматривается критерий качества:

$$S = ||\mathbf{w}_1^{\text{pred}} - \mathbf{w}_1^{\text{true}}||_2^2 + ||\mathbf{w}_2^{\text{pred}} - \mathbf{w}_2^{\text{true}}||_2^2,$$

где $\mathbf{w}_1^{\text{pred}}$ — аппроксимация вектора параметров первой локальной модели, а $\mathbf{w}_2^{\text{pred}}$ — аппроксимация вектора параметров второй локальной модели.

На рис. 5 показана зависимость критерия качества S от уровня шума β и параметра априорного распределения γ. Из графика видно, что при малом уровне шума β качество аппроксимации не зависит от параметра γ, а при увеличении шума β качество аппроксимации S уменьшается.

На рис. 6 показан пример работы алгоритма с разными параметрами β и γ. Видно, что в отсутствие шума β обе локальные модели аппроксимируют выборку корректно. С ростом уровня шума качество аппроксимации падает: при $\beta = 0{,}2$ с ростом γ первая локальная модель из параболы преобразовывается в эллипс; для $\beta = 0{,}4$ с ростом γ первая локальная модель из параболы преобразовывается в эллипс, а вторая модель — из параболы в гиперболу.

5.3. Аппроксимация радужки глаза

Анализ качества аппроксимации проводится для задачи аппроксимации радужной оболочки глаза на изображении. Радужная оболочка глаза состоит из двух концентрических окружностей, поэтому рассматривается мультимодель, состоящая из двух экспертов: каждый эксперт аппроксимирует одну из окружностей. В вычислительном эксперименте сравнивается качество аппроксимации окружностей в случае задания разных регуляризаторов R_0, R_1, R_2. Регуляризатор $R_0\big(\mathbf{V}, \mathbf{W}, E(\Omega)\big) = 0$, что соответствует отсутствию регуляризатора. Регуляризатор

$$R_1\big(\mathbf{V}, \mathbf{W}, E(\Omega)\big) = -\sum_{k=1}^{K} \Big[\mathbf{w}_k^{\mathsf{T}} \mathbf{w}_k$$

способствует околонулевым параметрам локальных моделей. Регуляризатор

$$R_2\big(\mathbf{V}, \mathbf{W}, E(\Omega)\big) = -\sum_{k=1}^{K} \Big[\mathbf{w}_k^{\mathsf{T}} \mathbf{w}_k + \sum_{k=1}^{K} \sum_{k'=1}^{K} \sum_{j=1}^{2} \Big[\Big(w_k^j - w_k'^j \Big)^2$$

способствует совпадению центров окружностей и близким к нулю параметрам локальных моделей.

Рисунок 7 показывает результат аппроксимации радужной оболочки глаза после 10 итераций. Видно, что при отсутствии регуляризатора одна из окружностей находится некорректно. В случае задания регуляризатора R_1 модель аппроксимирует обе окружности, но окружности неконцентричны. В случае задания регуляризатора R_2 получаем концентрические окружности на изображении.

На рис. 8–10 показан процесс сходимости мультимоделей в случае указания разных регуляризаторов R_0, R_1, R_2. Видно, что модели с регуляризатором типа R_1 и R_2 аппроксимируют обе окружности, а мультимодель с регуляризатором R_0 аппроксимирует только большую окружность.

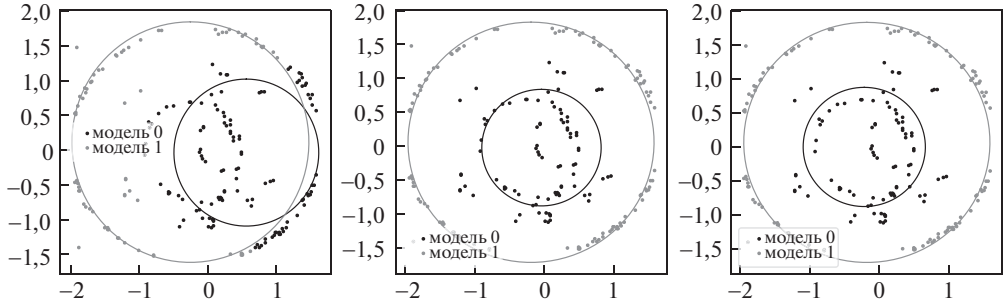


Рис. 7. Визуализация аппроксимации радужной оболочки. Слева направо: если указан регуляризатор R_0; если указан регуляризатор R_1; если указан регуляризатор R_2.

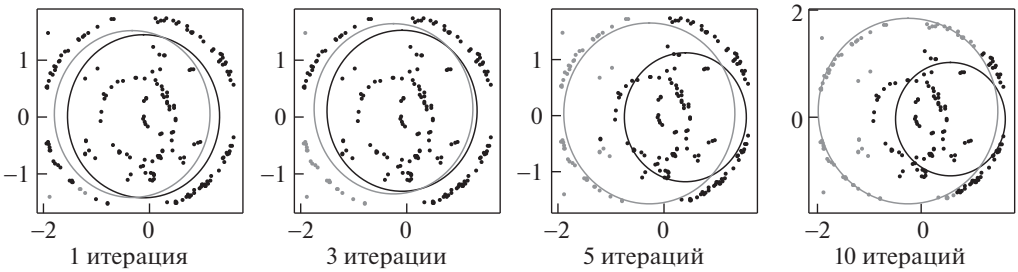


Рис. 8. Визуализация мультимодели в случае регуляризатора R_0.

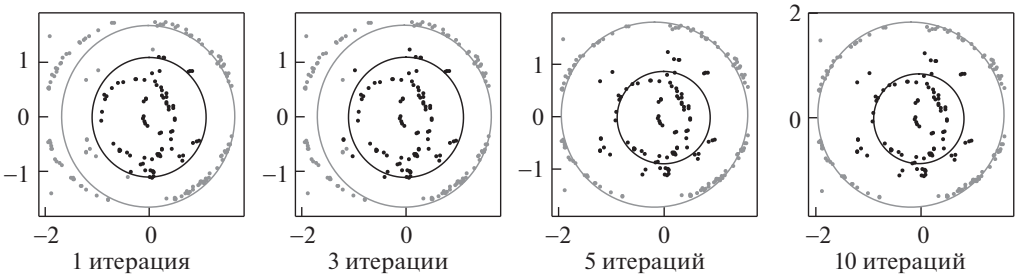


Рис. 9. Визуализация мультимодели в случае регуляризатора R_1.

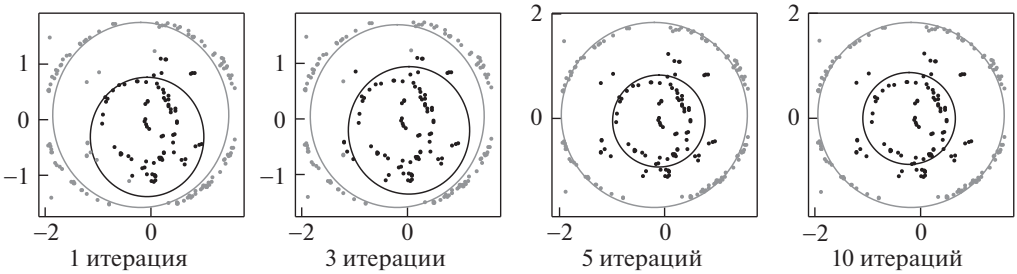


Рис. 10. Визуализация мультимодели в случае регуляризатора R_2.

6. Заключение

В статье предлагается метод построения интерпретируемых моделей машинного обучения на основе экспертной информации. В качестве задачи рассматривается задача аппроксимации кривых второго порядка: параболы, гиперболы, эллипса. Аппроксимация кривых второго порядка применяется в задаче об аппроксимации радужной оболочки.

Проведен эксперимент, в ходе которого анализируется качество аппроксимации кривых второго порядка в зависимости от начального уровня шума в данных, а также в зависимости от регуляризатора функции ошибки. В ходе эксперимента показано, что с увеличением уровня шума в исходных данных точность аппроксимации снижается: при большом шуме форма аппроксимируемой фигуры меняется с параболы на гиперболу. Проведен вычислительный эксперимент по аппроксимации радужной оболочки глаза двумя концентрическими окружностями. Эксперимент показывает, что регуляризация на основе экспертной информации улучшает качество аппроксимации.

СПИСОК ЛИТЕРАТУРЫ

1. *He K., Ren S., Sun J., Zhang X.* Deep Residual Learning for Image Recognition // Proc. IEEE Conf. on Computer Vision and Pattern Recognition. Las Vegas. 2016. P. 770–778.
2. *Ribeiro M., Singh S., Guestrin C.* Why Should I Trust You?: Explaining the Predictions of Any Classifier // Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016. P. 1135–1144.
3. *Han X., Yao M., Debayan D., Hui L., Ji-Liang T., Anil J.* Adversarial Attacks and Defenses in Images, Graphs and Text: A Review // Int. J. Automat. Comput. 2020. V. 17. P. 151–178.
4. *Akhtar N., Mian A.* Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey // IEEE Access. 2018. V. 6. P. 14410–14430.
5. *Grabovoy A., Strijov V.* Probabilistic Interpretation of the Distillation Problem // Autom. Remote Control. 2022. V. 83. P. 123–137.
6. *Matveev I.* Detection of iris in image by interrelated maxima of brightness gradient projections // Appl. Comput. Math. 2010. V. 9. P. 252–257.
7. *Matveev I., Simonenko I.* Detecting precise iris boundaries by circular shortest path method // Pattern Recognition and Image Analysis. 2014. V. 24. P. 304–309.
8. *Bowyer K., Hollingsworth K., Flynn P.* A Survey of Iris Biometrics Research: 2008-2010 // Handbook of iris recognition. 2010. P. 15–54.
9. *Salamani D., Gadatsch S., Golling T., Stewart G., Ghosh A., Rousseau D., Hasib A., Schaarschmidt J.* Deep Generative Models for Fast Shower Simulation in ATLAS // IEEE 14th International Conference on e-Science. 2018. P. 348–348.
10. *Chen Xi., Ishwaran H.* Random Forests for Genomic Data Analysis // Genomics. 2012. V. 6. P. 323–329.
11. *Chen T., Guestrin C.* XGBoost: A Scalable Tree Boosting System // Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016. P. 785–794.
12. *Yuksel S., Wilson J., Gader P.* Twenty Years of Mixture of Experts // IEEE Transactions on Neural Networks and Learning Systems. 2012. V. 8. P. 1177–1193.

13. *Dempster A., Laird N., Rubin D.* Maximum Likelihood from Incomplete Data via the EM Algorithm // Journal of the Royal Statistical Society. Series B (Methodological). 1977. V. 39. P. 1–38.
14. *Ebrahimpour R., Moradian R., Esmkhani A., Jafarlou F.* Recognition of Persian handwritten digits using characterization loci and mixture of experts // J. Digital Content Technol. Appl. 2009. P. 42–46.
15. *Peng F., Jacobs R., Tanner M.* Bayesian inference in mixtures-of-experts and hierarchical mixtures-of-experts models with an application to speech recognition // J. Amer. Stat. Assoc. 1996. V. 91. P. 953–960.
16. *Grabovoy A., Strijov V.* Prior Distribution Selection for a Mixture of Experts // Comput. Math. and Math. Phys. 2021. P. 1140–1152.
17. *Estabrooks A., Japkowicz N.* A mixture-of-experts framework for text classification // Proc. Workshop Comput. Natural Lang. Learn., Assoc. Comput. Linguist. 2001. P. 1–8.
18. *Cheung Y., Leung W., Xu L.* Application of mixture of experts model to financial time series forecasting // Proc. Int. Conf. Neural Netw. Signal Process. 1995. P. 1–4.
19. *Weigend A., Shi S.* Predicting daily probability distributions of S&P500 returns // J. Forecast. 2000. V. 19. P. 375–392.
20. *Cao L.* Support vector machines experts for time series forecasting // Neurocomputing. 2003. V. 51. P. 321–339.
21. *Mossavat S., Amft O., Vries B., Petkov P., Kleijn W.* A Bayesian hierarchical mixture of experts approach to estimate speech quality // Proc. 2nd Int. Workshop Qual. Multimedia Exper. 2010. P. 200–205.
22. *Sminchisescu C., Kanaujia A., Metaxas D.* Discriminative density propagation for visual tracking // IEEE Trans. Pattern Anal. Mach. Intell. 2007. V. 29. P. 2030–2044.
23. *Tuerk A.* The state based mixture of experts HMM with applications to the recognition of spontaneous speech // Ph.D. thesis, University of Cambridge. 2001.
24. *Yumlu M., Gurgen F., Okay N.* Financial time series prediction using mixture of experts // Proc. 18th Int. Symp. Comput. Inf. Sci. 2003. P. 553–560.

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 31.01.2022

После доработки 25.06.2022

Принята к публикации 29.06.2022

© 2022 г. А.А. ЗАХАРОВ, канд. техн. наук (aa-zaharov@ya.ru)
(Муромский институт (филиал) ФГБОУ ВО
«Владимирский государственный университет
им. Александра Григорьевича и Николая Григорьевича
Столетовых», Муром)

МЕТОД СОПОСТАВЛЕНИЯ ИЗОБРАЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ ТЕПЛОВЫХ ЯДЕР НА ГРАФАХ[1]

В работе представлен метод сопоставления изображений на основе тепловых ядер. Метод позволяет выделять на начальном этапе с помощью тепловых ядер на графах наиболее устойчивые особенности изображений для последующего сопоставления. Для этого могут использоваться популярные дескрипторы. При использовании метода значительно сокращается количество ложных соответствий по сравнению с традиционными подходами.

Ключевые слова: компьютерное зрение, сопоставление изображений, тепловые ядра на графах.

DOI: 10.31857/S0005231022100063, **EDN:** AKDWHB

1. Введение

Методы сопоставления изображений часто используются в системах технического зрения. На точность сопоставления изображений влияют ракурс съемки, особенности датчиков, освещенность сцены, взаимное перекрытие объектов, наличие однородных объектов и поверхностей. При сопоставлении изображений важны следующие показатели: скорость сопоставления; автоматизация процесса; робастность к помехам, перекрытиям, эффектам освещения; инвариантность к изменению ракурса и масштабированию сцены и т.д.

Дескрипторы описывают особенности изображений в некоторой области с использованием конкретных признаков. Было разработано большое количество дескрипторов изображений. Все дескрипторы изображений можно разделить на группы: локальные двоичные дескрипторы [1–5], спектральные дескрипторы [6–11], дескрипторы базового пространства [12, 13], дескрипторы формы полигона [14, 15]. Локальные двоичные дескрипторы основаны на бинарном сравнении пикселей. Спектральные дескрипторы используют широкий диапазон характеристик: интенсивность света, цвет, градиенты локальной области, статистические характеристики, моменты локальной области, нормали поверхности и т.д. Дескрипторы базового пространства кодируют вектор признаков в набор базовых функций. Дескрипторы формы полигона используют такие характеристики, как площадь, периметр, центр тяжести и т.д. Был разработан метод RANSAC, который позволяет уменьшать количество ошибок [16]. Также в последнее время стали применяться методы глубокого обучения [17, 18] для сопоставления изображений. Слабыми сторонами

[1] Работа выполнена при финансовой поддержке Министерства науки и высшего образования РФ (Госзадание ВлГУ № ГБ-1187/20).

методов с использованием глубокого обучения являются высокие вычислительные затраты, необходимость обучения, высокие требования к качеству и объему обучающего набора.

Разработаны методы для сопоставления изображений на основе сравнения структур с применением графов [19–25]. Методы основаны на использовании спектральной теории графов [26]. Ограничением подобных методов является то, что часто не принимаются во внимание признаки локальных окрестностей изображений.

К одним из недостатков методов и алгоритмов нахождения соответствий можно отнести то, что сопоставления находятся между всеми наборами обнаруженных особенностей. При этом не учитывается то, что многие особенности могут пропадать при изменении ракурса, наличии взаимных перекрытий, бликов. Это обстоятельство увеличивает вероятность нахождения ложных соответствий.

В работе для сопоставления изображений предлагается использовать дескриптор SURF и тепловые ядра (heat kernel) на графах. Тепловые ядра описываются в рамках спектральной теории графов, которая часто используется при разработке методов компьютерного зрения: сегментации изображений, обнаружении объектов, распознавании сцен и т.д. [27–30].

В статье представлен новый метод сопоставления изображений, который позволяет выделять и сопоставлять наиболее устойчивые особенности, присутствующие на снимках. Это позволяет значительно снизить количество ложных соответствий при сопоставлении снимков.

2. Метод сопоставления изображений

При сопоставлении снимков угол поворота сцены относительно наблюдателя может существенно меняться. При этом в сопоставлении могут участвовать множества особенностей, которые значительно отличаются. Это приводит к увеличению количества неправильных соответствий. Для устранения проблемы предлагается выделять на изображениях только самые устойчивые особенности, которые с большей вероятностью будут присутствовать на снимках.

На основе детектора SURF на сопоставляемых изображениях находятся точечные особенности. Эти особенности используются для построения графа Делоне. Для выделения устойчивых особенностей изображений используются тепловые ядра на графах [31].

Тепловое ядро представляет собой изменение температуры в области вокруг вершины с начальной тепловой энергией в момент времени $t = 0$. Для графа $G(V, E)$ тепловое ядро показывает, как информация проходит через ребра с течением времени.

Нормализованная матрица Лапласа записывается в следующем виде [26]:

$$L_n = \begin{cases} 1, & \text{если } x = 0, \\ -\dfrac{w_{uv}}{\sqrt{d_u, d_v}}, & \text{если } u \neq v \text{ и } (u, v) \in E, \\ 0, & \text{иначе.} \end{cases}$$

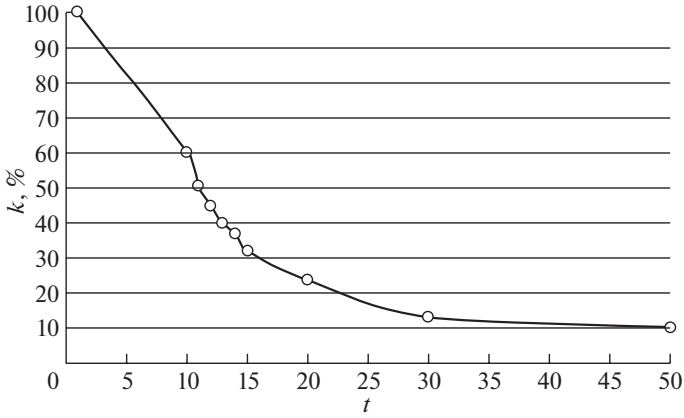


Рис. 1. Изменение количества устойчивых вершин от времени.

L_n можно разложить на $\Phi\Lambda\Phi^T$, где $\Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_{|V|})$ — диагональная матрица, содержащая упорядоченные собственные значения, и $\Phi = (\Phi_1|\Phi_2|\ldots|\Phi_{|V|})$ — квадратная матрица с упорядоченными собственными векторами в столбцах.

Уравнение теплопроводности, связанное с L_n, имеет вид [31, 32]:

$$\frac{dH_t}{dt} = -L_n H_t,$$

где t — время, а тепловое ядро H_t — решение уравнения теплопроводности.

Выражение теплового ядра имеет вид

$$H_t = e^{-tL_n}.$$

Тепловое ядро можно также представить следующим образом:

$$H_t = \Phi e^{-t\Lambda}\Phi^T.$$

Тепловое ядро для вершин u и v графа G описывается выражением

$$H_t(u,v) = \sum_{i=1}^{|V|} e^{-\lambda_i t}\Phi_i(u)\Phi_i(v).$$

На основе предыдущего выражения определяются устойчивые вершины графа. Вершина считается устойчивой, если $H_t(u,v) > \sigma$, где σ — минимальное значение из p наибольших элементов матрицы H_t. Между особенностями снимков, которые определены устойчивыми вершинами, будут находиться соответствия.

При увеличении времени t количество устойчивых вершин k уменьшается (рис. 1). При проведении исследований выбиралось значение $t = 10$.

Таким образом, алгоритм сопоставления изображений состоит из следующих шагов:

Шаг 1. Выделяются особенности на сопоставляемых изображениях на основе детектора SURF.

Шаг 2. По выделенным особенностям строится граф Делоне.

Шаг 3. На основе тепловых ядер определяются устойчивые вершины.

Шаг 4. На основе дескриптора SURF находятся соответствия между устойчивыми особенностями сопоставляемых изображений.

3. Исследование разработанного метода

Эксперименты проводились с наборами снимков CMU, COIL-100, MIT-CBCL. Разработанный метод сравнивался с методом SURF, который часто используется при сопоставлении изображений и включен в различные библиотеки компьютерного зрения. При проведении исследований вычислялось количество «выбросов» (неправильных соответствий) при изменении ракурса одного из изображений. При использовании метода SURF сопоставляется большее количество особенностей изображения, чем при применении представленного в работе метода. Это объясняется тем, что в разработанном методе в сопоставлении участвуют только самые устойчивые особенности изображения. Подобные особенности практически всегда выделяются при изменении ракурса изображения. На рис. 2 и 3 представлены результаты сопоставления изображений с использованием метода SURF и разработанного метода. Неправильные соответствия представлены красными линиями.

При проведении исследований выявлено, что при увеличении угла поворота объекта α количество верных соответствий q сокращается. Это происходит как при использовании разработанного метода, так и при использовании метода SURF. Однако при сопоставлении изображений с помощью SURF количество «выбросов» значительно больше. При использовании ме-

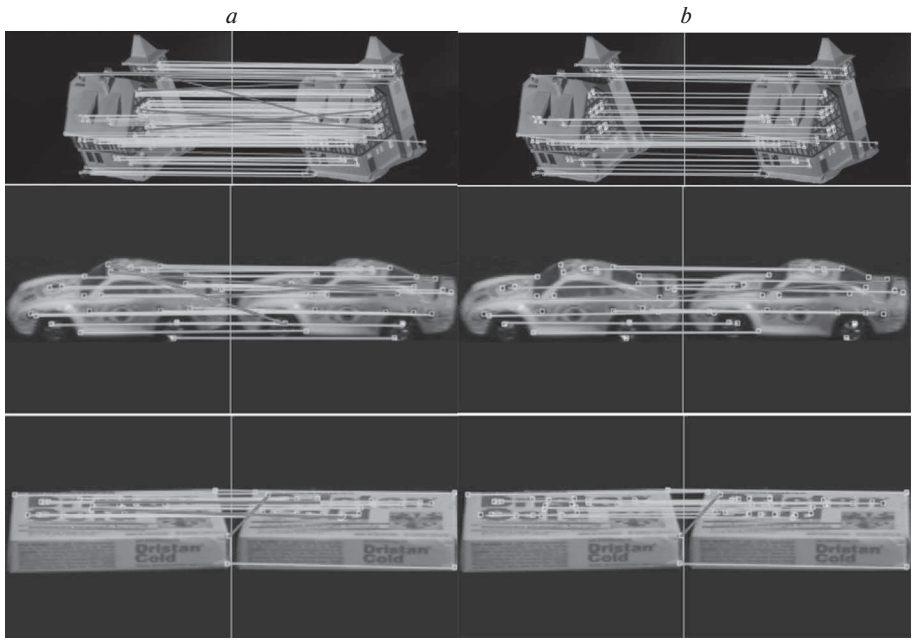


Рис. 2. Сопоставление изображений (CMU, COIL-100): *a* — метод SURF, *b* — разработанный метод.

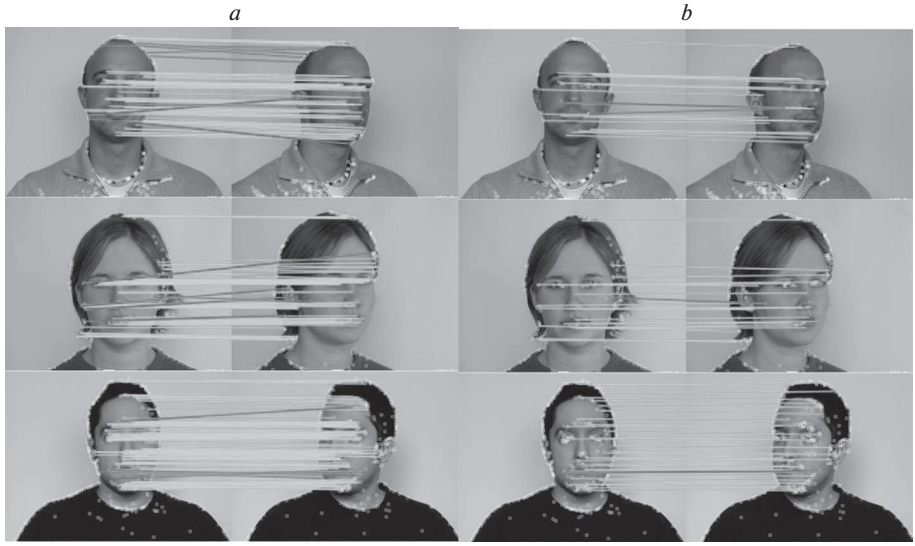


Рис. 3. Сопоставление изображений (MIT-CBCL): *a* — метод SURF, *b* — разработанный метод.

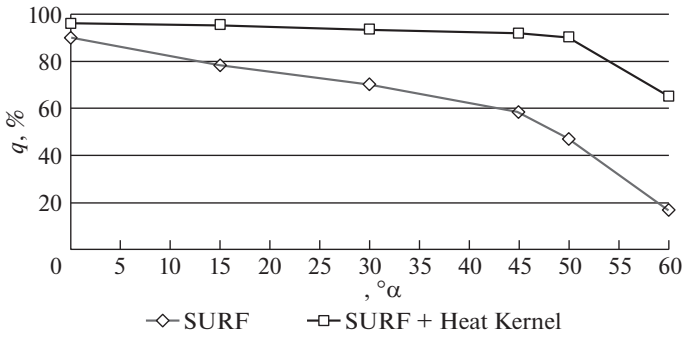


Рис. 4. Зависимость количества верных сопоставлений от ракурса изображений.

тода SURF количество неправильных соответствий может превосходить 40% (рис. 4). При использовании предложенного в работе метода с изменением ракурса на угол 45° количество «выбросов» не превышает 10%.

4. Заключение

Таким образом, в работе предложен метод сопоставления изображений с использованием тепловых ядер на графах. Метод позволяет значительно снижать количество ложных соответствий при изменении ракурса изображения. Также метод не требует этапа предварительного обучения. Метод можно использовать при решении различных задач компьютерного зрения: поиска и отслеживания объектов, реконструкции трехмерных сцен, создания мозаик и т.д.

СПИСОК ЛИТЕРАТУРЫ

1. *Krig S.* Computer vision metrics. Survey, taxonomy, and analysis. Berkeley: Apress, 2014.
2. *Ojala T., Pietikainen M., Hardwood D.* A comparative study of texture measures with classification based on feature distributions // Patt. Recognit. 1996. V. 29 (1). P. 51–59. https://doi.org/10.1016/0031-3203(95)00067-4.
3. *Calonder M., Lepetit V., Strecha C., Fua P.* BRIEF-Binary Robust Independent Elementary Features // ECCV. 2010. P. IV. P. 778–792. https://doi.org/10.1007/978-3-642-15561-1_56
4. *Rublee E., Rabaud V., Konolige K., Bradski G.* ORB: an efficient alternative to SIFT or SURF // ICCV. 2011. P. 2564–2571. https://doi.org/10.1109/ICCV.2011.6126544.
5. *Leutenegger S., Chli M., Siegwart R.* BRISK: Binary Robust Invariant Scalable Keypoints // International Conference on Computer Vision (ICCV'11). 2011. P. 2548–2555. https://doi.org/10.1109/ICCV.2011.6126542.
6. *Lowe D.G.* Distinctive Image Features from Scale-Invariant Keypoints // Int. J. Comput. Vision. 2004. V. 60(2). P. 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94.
7. *Bay H., Ess A., Tuytelaars T., Van Gool L.* SURF: Speeded Up Robust Features // Comput. Vision Image Understand. 2008. V. 110(3). P. 346–359. https://doi.org/10.1007/11744023_32.
8. *Tola E., Lepetit V., Fua P.* DAISY: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo // IEEE Transact. Patt. Anal. Machine Intelligence. 2010. V. 32(5). P. 815–830. https://doi.org/10.1109/TPAMI.2009.77.
9. *Dalal N., Triggs B.* Histograms of Oriented Gradients for Human Detection // Comput. Vision Patt. Recognit. 2005. V. 1. P. 886–893. https://doi.org/10.1109/CVPR.2005.177.
10. *Scharstein D., Szeliski R.* A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms // Int. J. Comput. Vision. 2002. No. 47. P. 7–42. https://doi.org/10.1023/A:1014573219977.
11. *Jun B., Kim D.* Robust Face Detection Using Local Gradient Patterns and Evidence Accumulation // Patt. Recognit. 2012. V. 45(9). P. 3304–3316. https://doi.org/10.1016/j.patcog.2012.02.031.
12. *Bracewell R.* The Fourier Transform and Its Applications. McGraw-Hill Science/Engineering/Math; 3 edition, 1999.
13. *Ren X., Ramanan D.* Histograms of Sparse Codes for Object Detection // Conference on Computer Vision and Pattern Recognition. 2013. https://doi.org/10.1109/CVPR.2013.417.
14. *Matas J., Chum O., Urba M., Pajdla T.* Robust Wide Baseline Stereo from Maximally Stable Extremal Regions // Proceedings of British Machine Vision Conference. 2002. https://doi.org/10.1016/j.imavis.2004.02.006.
15. *Yang M., Kpalma K., Ronsin J.* A Survey of Shape Feature Extraction Techniques // Patt. Recognit. 2008. P. 43–90. https://doi.org/10.5772/6237.
16. *Szeliski R.* Computer Vision: Algorithms and Applications. Springer, 2010.
17. *Ufer N., Ommer B.* Deep Semantic Feature Matching // CVPR 2017, 2017. P. 6914–6923. https://doi.org/10.1109/CVPR.2017.628.
18. *Gao Q., Wang F., Xue N., Yu J.G., Xia G.S.* Deep Graph Matching under Quadratic Constraint // CVPR 2021, 2021. P. 5069–5078. https://doi.org/10.1109/CVPR46437.2021.00503.

19. *Scott G., Longuet-Higgins H.* An algorithm for associating the features of two images // Proc. of Royal Soc. 1991. V. 244. P. 21–26. https://doi.org/10.1098/rspb.1991.0045.
20. *Zakharov A.A., Zhiznyakov A.L., Titov V.S.* A method for feature matching in images using descriptor structures // Comput. Opt. 2019. V. 43(5). P. 811–818. https://doi.org/10.18287/2412-6179- 2019-43-5-811-818.
21. *Shapiro L.S., Brady J.M.* Feature-based correspondence – an eigenvector approach // Image Vision Comput. 1992. V. 10(5). P. 283–288. https://doi.org/10.1016/0262-8856(92)90043-3.
22. *Carcassoni M., Hancock E.* Spectral correspondence for point pattern matching // Patt. Recognit. 2003. V. 36(1). P. 193–204. https://doi.org/10.1016/S0031-3203(02)00054-7.
23. *Leordeanu M., Hebert M.* A spectral technique for correspondence problems using pairwise constraints // Tenth IEEE International Conference on Computer Vision. 2005. V. 1. https://doi.org/10.1109/ICCV.2005.20.
24. *Cour T., Srinivasan P., Shi J.* Balanced Graph Matching // Proceedings Conference Neural Information Processing Systems. 2006. https://doi.org/10.7551/mitpress/7503.003.0044.
25. *Delponte E., Isgro F., Odone F., Verri A.* SVD-matching using SIFT features // Graphical Models. 2006. V. 68 (5-6). P. 415–431. https://doi.org/10.1016/j.gmod.2006.07.002.
26. *Chung F.R.K.* Spectral graph theory. AMS, 1997.
27. *Zakharov A.A., Titov D.V., Zhiznyakov A.L., Titov V.S.* Visual attention method based on vertex ranking of graphs by heterogeneous image attributes // Comput. Opt. 2020. V. 44(3). P. 427–435. https://doi.org/10.18287/2412-6179-CO-658.
28. *Zakharov A.A., Barinov A.E., Zhiznyakov A.L., Titov V.S.* Object detection in images with a structural descriptor based on graphs // Comput. Opt. 2018. V. 42(2). P. 283–290. https://doi.org/10.18287/2412-6179-2018-42-2-283-290.
29. *Zakharov A., Barinov A., Zhiznyakov A.* Faces selection in images using the spectral graph theory and constraints // 2017 International Conference on Industrial Engineering, Applications and Manufacturing. 2017. https://doi.org/10.1109/ICIEAM.2017.8076407.
30. *Zakharov A., Tuzhilkin A., Zhiznyakov A.* Automatic building detection from satellite images using spectral graph theory // International Conference on Mechanical Engineering, Automation and Control Systems (MEACS 2015). 2015. https://doi.org/10.1109/MEACS.2015.7414937.
31. *Bai X., Wilson R.C., Hancock E.R.* Characterising graphs using the heat kernel // British machine vision conference. 2005. P. 315–324. https://doi.org/10.5244/C.19.92.
32. *Ghawalby H., Hancock E.R.* Heat kernel embeddings, differential geometry and graph structure // Axioms. 2015. V. 4. P. 275–293. https://doi.org/10.3390/axioms4030275.

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 01.02.2022

После доработки 20.06.2022

Принята к публикации 29.06.2022

© 2022 г. М. ГОРПИНИЧ (gorpinich.m@phystech.edu)
(Московский физико-технический институт
(государственный университет)),
О.Ю. БАХТЕЕВ, канд. физ.-мат. наук (bakhteev@phystech.edu),
В.В. СТРИЖОВ, д-р физ.-мат. наук (strijov@gmail.com)
(Вычислительный центр имени А.А. Дородницына
Федерального исследовательского центра
«Информатика и управление» РАН, Москва)

ГРАДИЕНТНЫЕ МЕТОДЫ ОПТИМИЗАЦИИ МЕТАПАРАМЕТРОВ В ЗАДАЧЕ ДИСТИЛЛЯЦИИ ЗНАНИЙ[1]

В работе исследуется задача дистилляции моделей глубокого обучения. Дистилляция знаний — это задача оптимизации метапараметров, в которой происходит перенос информации модели более сложной структуры, называемой моделью-учителем, в модель более простой структуры, называемой моделью-учеником. В работе предлагается обобщение задачи дистилляции на случай оптимизации метапараметров градиентными методами. Метапараметрами являются параметры оптимизационной задачи дистилляции. В качестве функции потерь для такой задачи выступает сумма слагаемого классификации и кросс-энтропии между ответами модели-ученика и модели-учителя. Назначение оптимальных метапараметров в функции потерь дистилляции является вычислительно сложной задачей. Исследуются свойства оптимизационной задачи с целью предсказания траектории обновления метапараметров. Проводится анализ траектории градиентной оптимизации метапараметров и предсказывается их значение с помощью линейных функций. Предложенный подход проиллюстрирован с помощью вычислительного эксперимента на выборках CIFAR-10 и Fashion-MNIST, а также на синтетических данных.

Ключевые слова: машинное обучение, дистилляция знаний, оптимизация метапараметров, градиентная оптимизация, назначение метапараметров.

DOI: 10.31857/S0005231022100075, **EDN:** AKGKQX

1. Введение

В работе рассматривается задача дистилляции моделей глубокого обучения. Оптимизация модели глубокого обучения является вычислительно сложной задачей [12]. В работе исследуется частный случай задачи оптимизации, называемый дистилляцией знаний. Он позволяет использовать одновременно обучающую выборку и информацию, содержащуюся в предобученных моделях. Дистилляцией знаний [5] назовем задачу оптимизации параметров модели, в которой учитывается не только информация, содержащаяся в исходной

[1] Работа выполнена при поддержке Научной академической стипендии имени К.В. Рудакова.

Таблица 1. Сложность различных методов оптимизации метапараметров и гиперпараметров. Здесь $|\mathbf{w}|$ является числом параметров модели, $|\boldsymbol{\lambda}|$ — числом метапараметров, r — это количество запусков стохастических методов оптимизации, s — сложность порождения из вероятностных моделей

Метод	Тип метода оптимизации	Сложность
Случайный поиск [2]	Стохастический	$O(r \cdot \|\mathbf{w}\|)$
Основанный на вероятностных моделях [3]	Стохастический	$O\left(r \cdot (\|\mathbf{w}\| + s)\right)$
Жадный градиентный [8]	Градиентный	$O(\|\mathbf{w}\| \cdot \|\boldsymbol{\lambda}\|)$
Жадный градиентный с разностной аппроксимацией [7]	Градиентный	$O(\|\mathbf{w}\| + \|\boldsymbol{\lambda}\|)$

выборке, но также и информация, содержащаяся в модели-учителе. Модель-учитель имеет высокую сложность. В ней содержится информация о выборке, а также о распределениях параметров модели, перенос которых будет осуществлен. Модель более простой структуры, называемая моделью-учеником, оптимизируется путем переноса знаний модели-учителя.

Исследуется процедура оптимизации метапараметров в задаче дистилляции знаний. Метапараметрами являются параметры оптимизационной задачи. Корректное назначение метапараметров может существенно повлиять на качество итоговой модели [11]. В отличие от [9, 11], в данной работе учитывается различие между гиперпараметрами, вероятностными параметрами априорного распределения [4] и метапараметрами. Несмотря на количество методов оптимизации метапараметров и гиперпараметров, использующихся в глубоком обучении, таких как случайный поиск [2] или модели, основанные на использовании вероятностных моделей [3], во многих подходах предлагается последовательно порождать случайное значение метапараметров и оценивать качество модели, обученной при данных значениях гиперпараметров. Данный подход может не подойти в случае обучения моделей, требующих значительных временных затрат для обучения. В табл. 1 содержатся сложности различных подходов к оптимизации метапараметров. Видно, что в случае, если оптимизация параметров занимает значительное время, подходы, требующие несколько запусков оптимизации, являются неэффективными.

Предлагается рассматривать задачу оптимизации метапараметров как двухуровневую задачу оптимизации. На первом уровне оптимизируются параметры модели, на втором — метапараметры [1, 8, 9]. Жадный градиентный метод для решения двухуровневой задачи описан в [8]. В [1] проанализированы различные градиентные методы и случайный поиск. В данной работе анализируется подход к оптимизации и предсказанию метапараметров, полученных после применения градиентных методов. Из табл. 1 можно увидеть, что для больших задач предпочтительны градиентные методы оптимизации метапараметров. Тем не менее, даже с применением жадного алгоритма оптимизации метапараметров с разностной аппроксимацией, оптимизация метапараметров становится значительно требовательнее к вычислительным ресурсам, что было продемонстрировано в [7]. Для уменьшения затрат на оптимизацию в настоящей работе проводится анализ траектории оптимизации

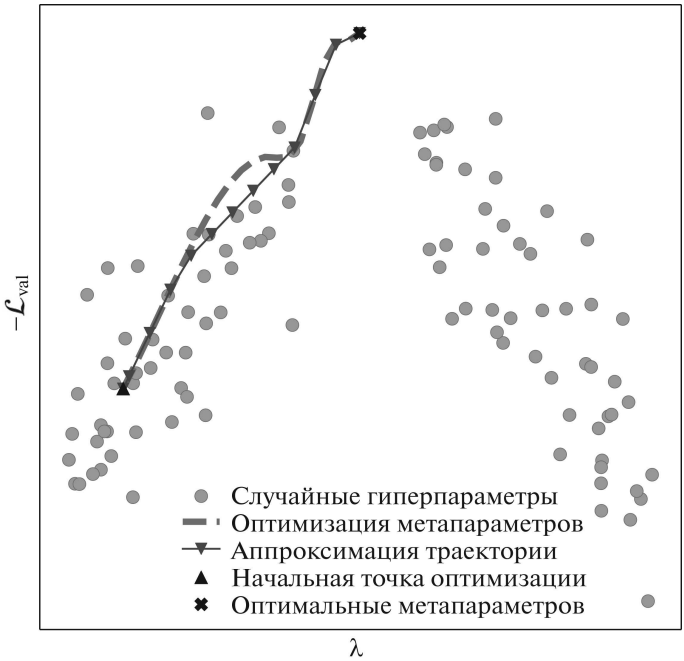


Рис. 1. Схема работы предложенного метода: вместо непосредственной оптимизации значений метапараметра λ предлагается аппроксимировать траекторию оптимизации с помощью линейных моделей для достижения минимума функции потерь на валидационной части выборки $\mathcal{L}_{\text{val}}$. Случайные метапараметры не являются точками минимума функции $\mathcal{L}_{\text{val}}$ и доставляют субоптимальное качество модели.

метапараметров и предсказывается ее значение с помощью линейных моделей. Этот метод проиллюстрирован на рис. 1. Данный метод оценивается и сравнивается с другими методами оптимизации метапараметров на выборках изображений CIFAR-10 [6], Fashion-MNIST [14] и синтетической выборке.

2. Постановка задачи

Решается задача классификации вида

$$\mathfrak{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{m},\ \mathbf{x}_i \in \mathbb{R}^n,\ \mathbf{y}_i \in \mathbb{Y} = \{\mathbf{e}_k | k = \overline{1, K}\},$$

где $\mathbf{e}_k$ — k-й столбец единичной матрицы, $\mathbf{y}_i$ — вектор с единицей на месте класса $\mathbf{x}_i$.

Разделим выборку на два подмножества $\mathfrak{D}$: $\mathfrak{D} = \mathfrak{D}_{\text{train}} \sqcup \mathfrak{D}_{\text{val}}$. Подмножество $\mathfrak{D}_{\text{train}}$ будем использовать для оптимизации параметров модели, а подмножество $\mathfrak{D}_{\text{val}}$ — для оптимизации метапараметров.

Рассмотрим модель-учителя $\mathbf{f}(\mathbf{x})$, которая была обучена на выборке $\mathfrak{D}_{\text{train}}$. Оптимизируем модель-ученика $\mathbf{g}(\mathbf{x}, \mathbf{w})$, $\mathbf{w} \in \mathbb{R}^s$ путем переноса знаний модели-учителя. Определим данную задачу формально.

Определение 1. Пусть функция $D : \mathbb{R}^s \to \mathbb{R}_+$ *задает расстояние между моделями* $\mathbf{g}$ *и* $\mathbf{f}$. *Назовем* D*-дистилляцией модели-ученика такую*

задачу оптимизации параметров модели-ученика, которая минимизирует функцию D.

Определим функцию потерь $\mathcal{L}_{\text{train}}$, которая учитывает перенос знаний от модели $\mathbf{f}$ к модели $\mathbf{g}$:

$$\mathcal{L}_{\text{train}}(\mathbf{w}, \boldsymbol{\lambda}) = -\lambda_1 \sum_{(\mathbf{x},\mathbf{y})\in\mathfrak{D}_{\text{train}}} \underbrace{\sum_{k=1}^{K} y_k \log \frac{e^{\mathbf{g}(\mathbf{x},\mathbf{w})_k}}{\sum\limits_{j=1}^{K} e^{\mathbf{g}(\mathbf{x},\mathbf{w})_j}}}_{\text{слагаемое классификации}} -$$

$$- (1-\lambda_1) \sum_{(\mathbf{x},y)\in\mathfrak{D}_{\text{train}}} \underbrace{\sum_{k=1}^{K} \frac{e^{\mathbf{f}(\mathbf{x})_k/T}}{\sum\limits_{j=1}^{K} e^{\mathbf{f}(\mathbf{x})_j/T}} \log \frac{e^{\mathbf{g}(\mathbf{x},\mathbf{w})_k/T}}{\sum\limits_{j=1}^{K} e^{\mathbf{g}(\mathbf{x},\mathbf{w})_j/T}}}_{\text{слагаемое дистилляции}},$$

где y_k — это k-я компонента вектора ответов, T — параметр температуры в задаче дистилляции. Температура T имеет следующие свойства:

1) если $T \to 0$, то получаем единичный вектор $\left\{ \left\lfloor \frac{e^{\mathbf{g}(\mathbf{x},\mathbf{w})_k/T}}{\sum\limits_{j=1}^{K} e^{\mathbf{g}(\mathbf{x},\mathbf{w})_j/T}} \right\rfloor \right\}_{k=1}^{K}$;

2) если $T \to \infty$, то получаем вектор с равными вероятностями.

Покажем, что оптимизация $\mathcal{L}_{\text{train}}$ является D-дистилляцией при $\lambda_1 = 0$.

Предложение 1. Если $\lambda_1 = 0$, то оптимизация функции потерь (1), *является D-дистилляцией с $D = D_{KL}\left(\sigma\left(\mathbf{f}(\mathbf{x})/T\right), \sigma\left(\mathbf{g}(\mathbf{x},\mathbf{w})/T\right)\right)$, где σ — это функция softmax $= \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}$, D_{KL} — дивергенция Кульбака–Лейблера.*

Доказательство. При $\lambda_1 = 0$ имеем:

$$(1)\quad \mathcal{L}_{\text{train}}(\mathbf{w}, \boldsymbol{\lambda}) = \sum_{(\mathbf{x},y)\in\mathfrak{D}_{\text{train}}} \sum_{k=1}^{K} \frac{e^{\mathbf{f}(\mathbf{x})_k/T}}{\sum\limits_{j=1}^{K} e^{\mathbf{f}(\mathbf{x})_j/T}} \log \frac{e^{\mathbf{g}(\mathbf{x},\mathbf{w})_k/T}}{\sum\limits_{j=1}^{K} e^{\mathbf{g}(\mathbf{x},\mathbf{w})_j/T}} =$$

$$= D_{KL}\left(\sigma(\mathbf{f}(\mathbf{x})/T), \sigma(\mathbf{g}(\mathbf{x},\mathbf{w})/T)\right) - C.$$

Получаем, что $\mathcal{L}_{\text{train}}(\mathbf{w}, \boldsymbol{\lambda})$ равняется $D_{KL}\left(\sigma(\mathbf{f}(\mathbf{x})/T), \sigma(\mathbf{g}(\mathbf{x},\mathbf{w})/T)\right)$ с точностью до константы C, не влияющей на оптимизацию. Константа является энтропией от $\sigma(\mathbf{f}(\mathbf{x})/T)$. Функция $D_{KL}\left(\sigma\left(\mathbf{f}/T\right), \sigma\left(\mathbf{g}/T\right)\right)$ определяет расстояние между логитами модели $\mathbf{f}$ и модели $\mathbf{g}$. Получаем, что определение D-дистилляции выполняется.

Определим множество метапараметров $\boldsymbol{\lambda}$ как вектор, компонентами которого являются коэффициент λ_1 перед слагаемыми в $\mathcal{L}_{\text{train}}$ и температура T:

$$\boldsymbol{\lambda} = [\lambda_1, T].$$

Определим двухуровневую задачу

$$\hat{\boldsymbol{\lambda}} = \arg\min_{\boldsymbol{\lambda}\in\mathbb{R}^2} \mathcal{L}_{\text{val}}(\hat{\mathbf{w}}, \boldsymbol{\lambda}), \tag{2}$$

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}\in\mathbb{R}^s} \mathcal{L}_{\text{train}}(\mathbf{w}, \boldsymbol{\lambda}), \tag{3}$$

где $\mathcal{L}_{\text{val}}$ — это функция потерь на валидации:

$$\mathcal{L}_{\text{val}}(\mathbf{w}, \boldsymbol{\lambda}) = -\sum_{(\mathbf{x},y)\in\mathfrak{D}_{\text{val}}} \sum_{k=1}^{K} \left[y^k \log \frac{e^{\mathbf{g}(\mathbf{x},\mathbf{w})_k / T_{\text{val}}}}{\sum\limits_{j=1}^{K} \left[e^{\mathbf{g}(\mathbf{x},\mathbf{w})_j / T_{\text{val}}} \right.} \right.,$$

метапараметр T_{val} определяет температуру в валидационной функции потерь. Его значение выбрано вручную и не является предметом оптимизации.

3. Градиентная оптимизация метапараметров

Одним из методов оптимизации метапараметров является использование градиентных методов. Ниже приведены схема их применения и подход к оптимизации траектории метапараметров.

Определение 2. Определим оператор оптимизации как алгоритм U, который выбирает вектор параметров модели $\mathbf{w}'$, используя значения параметров на предыдущем шаге $\mathbf{w}$.

Оптимизируем параметры $\mathbf{w}$, используя η шагов оптимизации:

$$\hat{\mathbf{w}} = U \circ U \circ \cdots \circ U(\mathbf{w}_0, \boldsymbol{\lambda}) = U^{\eta}(\mathbf{w}_0, \boldsymbol{\lambda}),$$

где $\mathbf{w}_0$ — начальное значение вектора параметров $\mathbf{w}$, $\boldsymbol{\lambda}$ — множество метапараметров.

Переформулируем оптимизационную задачу, используя определение оператора U:

$$\hat{\boldsymbol{\lambda}} = \arg\min_{\boldsymbol{\lambda}\in\mathbb{R}^2} \mathcal{L}_{\text{val}}\big(U^{\eta}(\mathbf{w}_0, \boldsymbol{\lambda})\big).$$

Решим оптимизационную задачу (2) и (3) с помощью оператора градиентного спуска:

$$U(\mathbf{w}, \boldsymbol{\lambda}) = \mathbf{w} - \gamma \nabla \mathcal{L}_{\text{train}}(\mathbf{w}, \boldsymbol{\lambda}),$$

где γ — длина шага градиентного спуска. Для оптимизации метапараметров используется жадный градиентный метод, который зависит только от значения параметров $\mathbf{w}$ на предыдущем шаге. На каждой итерации получим следующее значение метапараметров:

$$\boldsymbol{\lambda}' = \boldsymbol{\lambda} - \gamma_{\boldsymbol{\lambda}} \nabla_{\boldsymbol{\lambda}} \mathcal{L}_{\text{val}}(U(\mathbf{w}, \boldsymbol{\lambda}), \boldsymbol{\lambda}) = \boldsymbol{\lambda} - \gamma_{\boldsymbol{\lambda}} \nabla_{\boldsymbol{\lambda}} \mathcal{L}_{\text{val}}(\mathbf{w} - \gamma \nabla \mathcal{L}_{\text{train}}(\mathbf{w}, \boldsymbol{\lambda}), \boldsymbol{\lambda}). \tag{4}$$

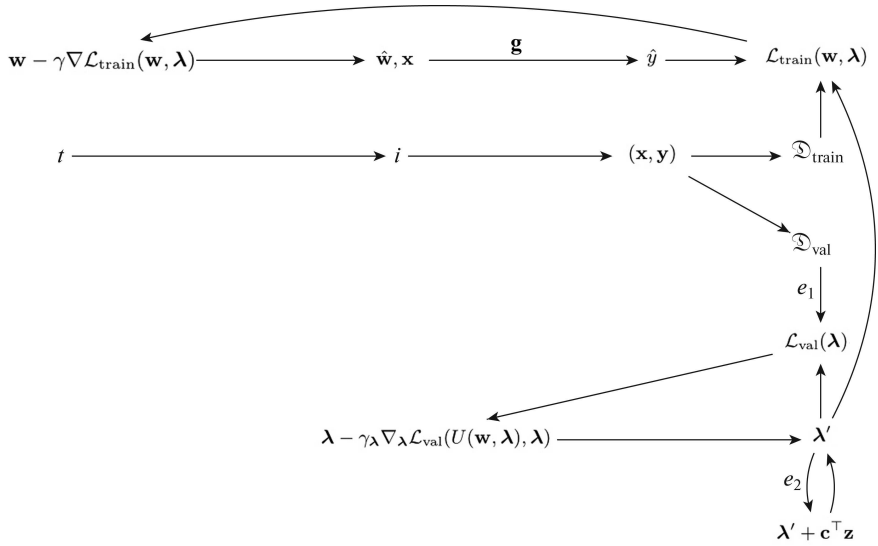


Рис. 2. Схема оптимизации метапараметров.

В данной работе используется численная разностная аппроксимация для данной процедуры оптимизации [7]:

$$\frac{d\mathcal{L}_{\text{val}}(\mathbf{w}', \boldsymbol{\lambda})}{d\boldsymbol{\lambda}} = \nabla_{\boldsymbol{\lambda}}\mathcal{L}_{\text{val}}(\mathbf{w}', \boldsymbol{\lambda}) - \gamma\nabla^2_{\boldsymbol{\lambda},\mathbf{w}'}\mathcal{L}_{\text{val}}(\mathbf{w}', \boldsymbol{\lambda})\nabla_{\mathbf{w}'}\mathcal{L}_{\text{val}}(\mathbf{w}', \boldsymbol{\lambda}),$$

$$\nabla^2_{\boldsymbol{\lambda},\mathbf{w}'}\mathcal{L}_{\text{val}}(\mathbf{w}', \boldsymbol{\lambda})\nabla_{\mathbf{w}'}\mathcal{L}_{\text{val}}(\mathbf{w}', \boldsymbol{\lambda}) \approx \frac{\nabla_{\boldsymbol{\lambda}}\mathcal{L}_{\text{val}}(\mathbf{w}^+, \boldsymbol{\lambda}) - \nabla_{\boldsymbol{\lambda}}\mathcal{L}_{\text{val}}(\mathbf{w}^-, \boldsymbol{\lambda})}{2\varepsilon},$$

$$\boldsymbol{\lambda}' \approx \boldsymbol{\lambda} - \gamma_{\boldsymbol{\lambda}}\nabla_{\boldsymbol{\lambda}}\mathcal{L}_{\text{val}}(\mathbf{w}', \boldsymbol{\lambda}) + \gamma\frac{\nabla_{\boldsymbol{\lambda}}\mathcal{L}_{\text{val}}(\mathbf{w}^+, \boldsymbol{\lambda}) - \nabla_{\boldsymbol{\lambda}}\mathcal{L}_{\text{val}}(\mathbf{w}^-, \boldsymbol{\lambda})}{2\varepsilon},$$

где $\mathbf{w}' = \mathbf{w} - \gamma\nabla\mathcal{L}_{\text{train}}(\mathbf{w}, \boldsymbol{\lambda})$, $\mathbf{w}^\pm = \mathbf{w}' \pm \varepsilon\nabla_{\mathbf{w}'}\mathcal{L}_{\text{val}}(\mathbf{w}', \boldsymbol{\lambda})$, ε — некоторая заданная константа.

Для дальнейшего уменьшения стоимости оптимизации предлагается аппроксимировать траекторию оптимизации метапараметров. Траектория предсказывается с помощью линейных моделей, которые используются периодически после заданного числа итераций e_1. После этого линейная модель используется для предсказания метапараметров на протяжении e_2 итераций:

$$\boldsymbol{\lambda}' = \boldsymbol{\lambda} + \mathbf{c}^\top \left(\begin{matrix} z \\ 1 \end{matrix}\right), \tag{5}$$

где $\mathbf{c}$ — это вектор параметров линейной модели, оптимизированный с помощью метода наименьших квадратов, z — число итераций оптимизации.

Диаграмма на рис. 2 описывает полученный метод оптимизации. Параметры модели оптимизируются на первом уровне двухуровневой оптимизационной задачи с помощью подмножества $\mathfrak{D}_{\text{train}}$ и функции потерь $\mathcal{L}_{\text{train}}$. Метапараметры оптимизируются на втором уровне с помощью подмножества $\mathfrak{D}_{\text{val}}$ и функции потерь $\mathcal{L}_{\text{val}}$. На протяжении e_1 итераций метапараметры оптимизируются с помощью метода стохастического градиентного спуска. На протяжении e_2 итераций предсказываются с помощью линейных моделей.

Алгоритм 1. Оптимизация метапараметров

Require: число e_1 итераций с использованием градиентной оптимизации
Require: число e_2 итераций с предсказанием $\boldsymbol{\lambda}$ линейными моделями
1: **while** нет сходимости **do**
2: Оптимизация $\boldsymbol{\lambda}$ и $\mathbf{w}$ на протяжении e_1 итераций, решая двухуровневую задачу
3: **traj** =траектория $(\nabla\boldsymbol{\lambda})$ изменяется во время оптимизации;
4: Положим $\mathbf{z} = [1, \ldots, e_1]^\mathsf{T}$
5: Оптимизация $\mathbf{c}$ с помощью МНК:

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}\in\mathbb{R}^2} ||\mathbf{traj} - \mathbf{z}\cdot c_1 + c_2||_2^2$$

6: Оптимизация $\mathbf{w}$ и предсказание $\boldsymbol{\lambda}$ на протяжении e_2 итераций с помощью линейной модели с параметрами $\mathbf{c}$.
7: **end while**

Алгоритм для предложенного метода

Следующая теорема доказывает корректность предложенной аппроксимации для простого случая: когда параметры $\mathbf{w}$ модели $\mathbf{g}$ достигли оптимума задачи (3), гессиан $\mathbf{H} = \nabla^2_{\mathbf{w}}\mathcal{L}_{\text{train}}$ является единичной матрицей, и оптимизация метапараметров ведется в области, в которой градиент метапараметров можно аппроксимировать константой. Отметим, что в общем случае данные условия при оптимизации моделей глубокого обучения не выполняются. В [8, 13] было показано, что использование методов нормализации промежуточных представлений выборки под действием нелинейных функций, входящих в модель глубокого обучения, приближает гессиан функции потерь к единичному. Анализ качества градиентной оптимизации метапараметров для случая, когда параметры модели не достигли оптимума, приведен в [11].

Теорема 1. Если функция $\mathcal{L}_{\text{train}}(\mathbf{w}, \boldsymbol{\lambda})$ *является гладкой и выпуклой, и ее гессиан* $\mathbf{H} = \nabla^2_{\mathbf{w}}\mathcal{L}_{\text{train}}$ *является единичной матрицей,* $\mathbf{H} = \mathbf{I}$, *а также если параметры* $\mathbf{w}$ *равны* $\mathbf{w}^*$, *где* $\mathbf{w}^*$ *— точка локального минимума для текущего значения* $\boldsymbol{\lambda}$, *тогда жадный алгоритм* (4) *находит оптимальное решение двухуровневой задачи. Если существует область* $\mathcal{D} \in \mathbb{R}^2$ *в пространстве метапараметров, такая что градиент метапараметров может быть аппроксимирован константой, то оптимизация является линейной по метапараметрам.*

Доказательство. В работе [11] была выведена формула для $\nabla_{\boldsymbol{\lambda}}\mathcal{L}_{\text{val}} = \nabla_{\boldsymbol{\lambda}}\mathcal{L}_{\text{val}}(U(\mathbf{w}, \boldsymbol{\lambda}))$ в случае, если $\mathcal{L}_{\text{train}}(\mathbf{w}, \boldsymbol{\lambda})$ является гладкой и выпуклой, и найдена $\mathbf{w}^*$ — точка локального минимума для текущего значения $\boldsymbol{\lambda}$:

$$\nabla_{\boldsymbol{\lambda}}\mathcal{L}_{\text{val}}(\boldsymbol{\lambda}) = \nabla_{\boldsymbol{\lambda}}\mathcal{L}_{\text{val}} - (\nabla^2_{\mathbf{w},\boldsymbol{\lambda}}\mathcal{L}_{\text{train}})^\top(\nabla^2_{\mathbf{w}}\mathcal{L}_{\text{train}})^{-1}\nabla_{\mathbf{w}}\mathcal{L}_{\text{val}}.$$

Эта формула упрощается исключением первого слагаемого, так как функция $\mathcal{L}_{\text{val}}$ явно не зависит от метапараметров:

$$\nabla_{\boldsymbol{\lambda}}\mathcal{L}_{\text{val}}(\boldsymbol{\lambda}) = -(\nabla^2_{\mathbf{w},\boldsymbol{\lambda}}\mathcal{L}_{\text{train}})^\top(\nabla^2_{\mathbf{w}}\mathcal{L}_{\text{train}})^{-1}\nabla_{\mathbf{w}}\mathcal{L}_{\text{val}}.$$

Если $\nabla^2_{\mathbf{w}}\mathcal{L}_{\text{train}}$ равен единичной матрице, то жадный алгоритм дает оптимум двухуровневой задачи в том случае, если его шаг выражается следующей

формулой [8]:

$$\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t + \eta_1 (\nabla^2_{\mathbf{w},\boldsymbol{\lambda}} \mathcal{L}_{\text{train}})^\top \nabla_{\mathbf{w}} \mathcal{L}_{\text{val}}.$$

Также заменим $\nabla^2_{\mathbf{w}} \mathcal{L}_{\text{train}}$ на единичную матрицу.

Вернемся к упрощенной формуле градиента:

$$\nabla_{\boldsymbol{\lambda}} \mathcal{L}_{\text{val}}(\boldsymbol{\lambda}) = -(\nabla^2_{\mathbf{w},\boldsymbol{\lambda}} \mathcal{L}_{\text{train}})^\top \nabla_{\mathbf{w}} \mathcal{L}_{\text{val}}.$$

Предположим, что существует область $\mathcal{D}$, в которой $\nabla_{\boldsymbol{\lambda}} \mathcal{L}_{\text{val}}(\boldsymbol{\lambda})$ равен константному вектору

$$\nabla_{\boldsymbol{\lambda}} \mathcal{L}_{\text{val}}(\boldsymbol{\lambda}) \approx \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}.$$

Тогда в $\mathcal{D}$ шаг оптимизации можно представить в виде

$$\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t - \gamma_{\boldsymbol{\lambda}} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix},$$

и имеет вид, аналогичный (5).

4. Вычислительный эксперимент

Целью эксперимента являются оценка качества предложенного метода дистилляции и анализ полученных моделей и их метапараметров. Метод оценивался на синтетической выборке, а также выборках CIFAR-10 и Fashion-MNIST. На выборке CIFAR-10 было проведено два вида экспериментов: на всей выборке, $|\mathfrak{D}_{\text{train}}| = 50\,000$, и на уменьшенной обучающей выборке, $|\mathfrak{D}_{\text{train}}| = 12\,800$.

Были проанализированы следующие методы оптимизации метапараметров:

1) оптимизация без дистилляции;

2) оптимизация со случайной инициализацией метапараметров. Метапараметры порождаются из равномерного распределения

$$\lambda_1 \sim \mathcal{U}(0;1), \quad T \sim \mathcal{U}(0{,}1, 10).$$

3) оптимизация с "наивным" назначением метапараметров:

$$\lambda_1 = 0{,}5, \quad T = 1;$$

4) градиентная оптимизация;

5) предложенный метод с $e_1 = e_2 = 10$.

6) оптимизация с помощью вероятностной модели. Для данного типа оптимизации использовалась библиотека hyperopt [3], в которой реализована оптимизация с помощью метода парзеновского окна. Для этого метода проводилось 5 запусков перед итоговым предсказанием метапараметров.

Для методов 1–3 использовалась вся обучающая выборка $\mathfrak{D}$. Для методов 4–6 выборка разбивалась на обучение, валидацию, контроль $\mathfrak{D} = \mathfrak{D}_{\text{train}} \sqcup \mathfrak{D}_{\text{val}} \sqcup \mathfrak{D}_{\text{test}}$.

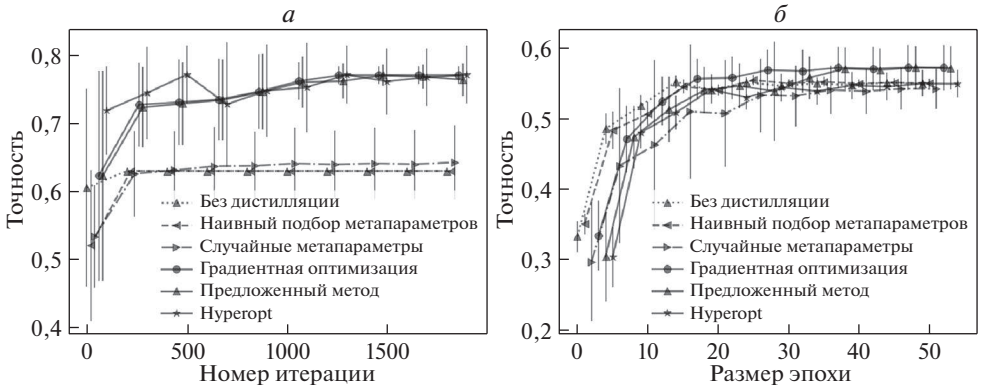


Рис. 3. Точность модели на выборках: *а* – синтетической, *б* – уменьшенной CIFAR-10. Здесь и далее точки незначительно смещены относительно оси абсцисс для лучшей читаемости графиков.

В качестве внешнего критерия качества была использована метрика accuracy:

$$\text{accuracy} = \frac{1}{m}\sum_{i=1}^{m}\Big[\mathbf{g}(\mathbf{x}_i, \mathbf{w}) = y_i\Big].$$

Для всех экспериментов порождение начальных значений метапараметров происходило следующим образом:

$$\lambda_1 \sim \mathcal{U}(0,1), \quad \log_{10} T \sim \mathcal{U}(-1,1).$$

Для каждого эксперимента проводилось 10 запусков, затем результаты усреднялись. Код эксперимента доступен в [15].

Итоговые результаты представлены в табл. 2. Зависимость точности от номера итерации на синтетической выборке и уменьшенной версии CIFAR-10 изображена на рис. 3.

Таблица 2. Результаты эксперимента. Числа в скобках являются максимальным полученным значением точности в конкретном эксперименте

Метод	Синтетическая выборка	Fashion-MNIST	Уменьшенный CIFAR-10	CIFAR-10
Без дистилляции	0,63 (0,63)	0,87 (0,88)	0,55 (0,56)	0,65 (0,66)
Наивные метапараметры	0,63 (0,63)	0,87 (0,88)	0,55 (0,56)	0,66 (0,67)
Случайные метапараметры	0,64 (0,72)	0,79 (0,88)	0,54 (0.57)	0.64 (0.67)
Градиентная оптимизация	**0.77** (0,78)	**0,88** (0,89)	**0,57** (0,61)	**0,70** (0,72)
Hyperopt	**0,77** (0,78)	0,87 (0,88)	0,55 (0,58)	0,65 (0,69)
Предложенный метод	0,76 (0,78)	**0,88** (0,89)	**0,57**	**0,70** (0,72)

4.1. Эксперимент на синтетической выборке

Для оценки полученного метода был проведен эксперимент на синтетической выборке:

$$\mathfrak{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}, \quad x_{ij} \in \mathcal{N}(0,1),\ j = 1,2, \quad x_{i3} = [\operatorname{sign}(x_{i1}) + \operatorname{sign}(x_{i2}) > 0],$$

$$y_i = \operatorname{sign}(x_{i1} \cdot x_{i2} + \delta),$$

где $\delta \in \mathcal{N}(0,\ 0{,}5)$ — это шум. Размер выборки модели-ученика значительно меньше размера выборки модели-учителя и $\mathfrak{D}_{\text{train}}$. Для корректной демонстрации предложенного метода в этом эксперименте выборка была поделена на 3 части: обучающая выборка для модели-учителя, состоящая из 200 объектов; обучающая выборка для модели-ученика, состоящая из 15 объектов; и валидационная выборка, которая также является тестовой, $\mathfrak{D}_{\text{val}} = \mathfrak{D}_{\text{test}}$. Она также состоит из 200 объектов. Визуализация выборки изображена на рис. 4. Модель-учитель была обучена на протяжении 20 000 итераций методом стохастического градиентного спуска с длиной шага, равной 10^{-2}. Для ее обучения было использовано модифицированное признаковое пространство:

$$x_{i3} = [\operatorname{sign}(x_{i1}) + \operatorname{sign}(x_{i2}) + 0{,}1 > 0].$$

Данная модификация не позволяет модели-учителю безошибочно предсказывать обучающую выборку. В данном случае, для обучения модели-ученика предпочтительно использование только слагаемого дистилляции, $\lambda_1 = 0$. Обучение модели-ученика происходило на протяжении 2000 итераций методом стохастического градиентного спуска с длиной шага, равной 1,0 и $T_{\text{val}} = 0{,}1$.

Была проведена серия экспериментов для определения наилучших значений e_1 и e_2. На рис. 5,*а* приведен график точности для различных e_1 с e_2 равным 10. На рис. 5,*б* изображена точность для различных значений e_2. Можно заметить, что с возрастанием e_1 и e_2 качество аппроксимации траектории обновления метапараметров уменьшается.

На рис. 3,*а* изображена точность модели для различных методов. Наилучшие результаты были получены для оптимизированных значений метапараметров и предложенного метода. Можно заметить, что предложенный метод хорошо аппроксимирует оптимизацию метапараметров в данном эксперименте.

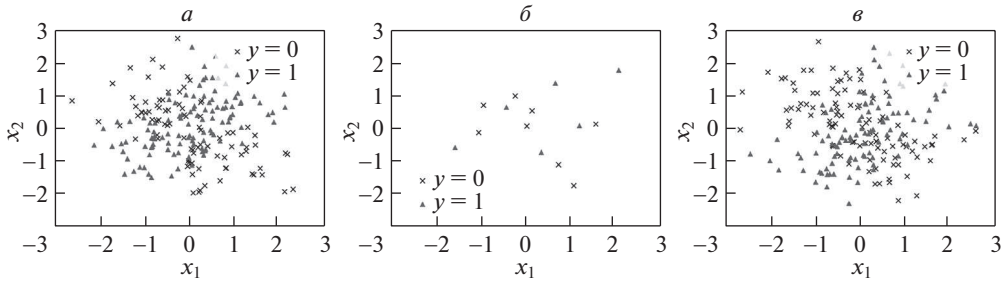


Рис. 4. Визуализация выборки для *а* – модели-учителя; *б* – модели-ученика; *в* – тестовой выборки.

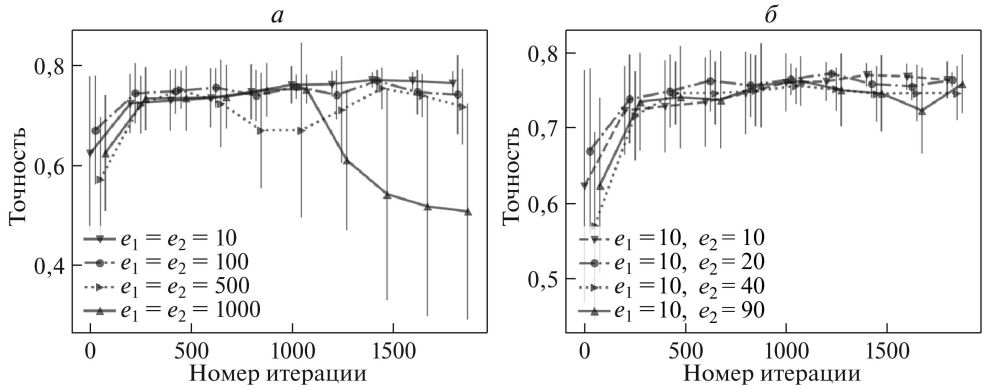


Рис. 5. Точность модели со значениями e_1 и e_2: *а* – $e_1 = e_2$; *б* – подбор e_2 при $e_1 = 10$.

4.2. Эксперименты на выборках CIFAR-10 и Fashion-MNIST

Обе выборки были разделены в пропорции 9:1 для обучения и валидации. Для оптимизации параметров модели был использован метод стохастического градиентного спуска с начальной длиной шага, равной 1,0. Длина шага умножалась на 0,5 каждые 10 эпох. Значение T_{val} задано равным 1,0.

Для эксперимента на выборке CIFAR-10 была использована предобученная модель ResNet из [10] в качестве модели-учителя. В качестве модели-ученика была использована модель CNN с тремя сверточными слоями и двумя полносвязными слоями.

Для экспериментов на уменьшенной выборке длина шага для оптимизации метапараметров была равна 0,25 и модель обучалась 50 эпох. Для эксперимента на полной выборке была использована дина шага, равная 0,1. Модель обучалась 100 эпох.

Для эксперимента на выборке Fashion-MNIST использовались архитектуры модели-ученика и модели-учителя, аналогичные архитектурам в эксперименте на выборке CIFAR-10. Для оптимизации метапараметров была использована длина шага, равная 0,1, и модель обучалась 50 эпох.

Из результатов в табл. 2 видно, что предложенный метод и градиентные методы дают высокое значение точности. Однако недостаток градиентных методов заключается в «застревании» в точках локального минимума, из-за чего дисперсия результатов получается гораздо выше, чем у остальных методов. Этот эффект можно заметить на рис. 3 и в табл. 2.

5. Заключение

Была исследована задача оптимизации параметров модели глубокого обучения. Было предложено обобщение методов дистилляции, заключающееся в градиентной оптимизации метапараметров. На первом уровне оптимизируются параметры модели, на втором — метапараметры, задающие вид оптимизационной задачи. Был предложен метод, уменьшающий вычислительную

сложность оптимизации метапараметров для градиентной оптимизации. Были исследованы свойства оптимизационной задачи и методы предсказания траектории оптимизации метапараметров модели. Под метапараметрами модели понимаются параметры оптимизационной задачи дистилляции. Предложенное обобщение позволило производить дистилляцию модели с лучшими эксплуатационными характеристиками и за меньшее число итераций оптимизации. Данный подход был проиллюстрирован с помощью вычислительного эксперимента на выборках CIFAR-10 и Fashion-MNIST, и на синтетической выборке. Вычислительный эксперимент показал эффективность градиентной оптимизации для задачи выбора метарапараметров функции потерь дистилляции. Проанализирована возможность аппроксимировать траекторию оптимизации метапараметров локально-линейной моделью. Планируются дальнейшее исследование оптимизационной задачи и анализ качества аппроксимации траектории оптимизации метапараметров более сложными прогностическими моделями.

СПИСОК ЛИТЕРАТУРЫ

1. *Bakhteev O.Y., Strijov V.V.* Comprehensive analysis of gradient-based hyperparameter optimization algorithms // Ann. Oper. Res. 2020. Vol. 289. No. 1. P. 51–65.
2. *Bergstra J., Bengio Y.* Random search for hyper-parameter optimization // MACHINE LEARNING RES. 2012. Vol. 13. No. 2.
3. *Bergstra J., Yamins D., Cox D.* Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures // International conference on machine learning. 2013. P. 115–123.
4. *Bishop C.M.* Pattern recognition and machine learning (information science and statistics). 2006.
5. *Hinton G.E., Vinyals O., Dean J.* Distilling the knowledge in a neural network // CoRR. 2015. Vol. abs/1503.02531. URL: http://arxiv.org/abs/1503.02531.
6. *Krizhevsky A., et al.* Learning multiple layers of features from tiny images, 2009.
7. *Liu H., Simonyan K., Yang Y.* Darts: Differentiable architecture search // arXiv preprint arXiv:1806.09055, 2018.
8. *Luketina J., Berglund M., Greff K., Raiko T.* Scalable gradient-based tuning of continuous regularization hyperparameters // CoRR. 2015. Vol. abs/1511.06727. URL: http://arxiv.org/abs/1511.06727.
9. *Maclaurin D., Duvenaud D., Adams R.P.* Gradient-based hyperparameter optimization through reversible learning // CoRR. 2015. Vol. abs/1502.03492. URL: http://arxiv.org/abs/1502.03492.
10. *Passalis N., Tzelepi M., Tefas A.* Heterogeneous knowledge distillation using information flow modeling // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2020.
11. *Pedregosa F.* Hyperparameter optimization with approximate gradient // CoRR, 2016. Vol. abs/1602.02355. URL: http://arxiv.org/abs/1602.02355.
12. *Rasley J., Rajbhandari S., Ruwase O., He Y.* Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters // Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020. P. 3505–3506.

13. *Vatanen T., Raiko T., Valpola H., LeCun Y.* Pushing stochastic gradient towards second-order methods – backpropagation learning with transformations in nonlinearities // International Conference on Neural Information Processing. Springer. Berlin, Heidelberg. 2013. P. 442–449.
14. *Xiao H., Rasul K., Vollgraf R.* Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms // CoRR. 2017. Vol. abs/1708.07747. URL: http://arxiv.org/278abs/1708.07747.
15. Код вычислительного эксперимента. URL: https://github.com/Intelligent-Systems-Phystech/MetaOptDistillation. Дата обращения: 14.06.2022.

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 17.02.2022

После доработки 23.06.2022

Принята к публикации 29.06.2022

© 2022 г. Д.С. ОБУХОВ (bstodin@gmail.com)
(Новосибирский государственный технический университет)

КЛОНИРОВАНИЕ И КОНВЕРСИЯ ПРОИЗВОЛЬНОГО ГОЛОСА С ИСПОЛЬЗОВАНИЕМ ГЕНЕРАТИВНЫХ ПОТОКОВ

С целью повышения качества формируемого речевого сигнала в данной работе предложен способ учета переменной во времени информации о спикере. Благодаря этой технике система синтезирует более естественную речь голосом, похожим на заданный целевой голос, как в задаче клонирования голоса, так и в задаче конверсии голоса.

Ключевые слова: клонирование голоса, конверсия голоса, синтез речи, потоковые генеративные модели, эмбеддинги спикера, частота основного тона.

DOI: 10.31857/S0005231022100087, **EDN:** AKGYHA

1. Введение

В настоящее время сфера применения синтеза речи стремительно расширяется и уже нашла свое применение в области медицины [1, 2], в голосовых колонках, умных ассистентах и других окружающих человека умных устройствах [3, 4], а также в различных задачах бизнеса [5, 6]. Одним из актуальных направлений развития синтеза речи сегодня является синтез голосом произвольного человека [7]. Умение генерировать речь с заданным голосом является необходимым требованием для ряда задач, например, диалоговых систем.

Современные подходы на основе глубокого обучения позволили эффективно и качественно формировать естественную речь голосом одного заданного диктора, представленного в наборе данных обучения. Предложенные недавно техники позволяют учитывать несколько дикторов при обучении, однако множество голосов, которыми формируется речь, по-прежнему остается ограниченным. Построение систем клонирования и конверсии произвольного голоса становится следующим вызовом в области формирования речевых сигналов.

Задача клонирования голоса подразумевает использование заданного образца речи человека для синтеза таким же голосом речевого сигнала с произвольным содержанием, заданным текстом [8]. Важной отличительной чертой клонирования голоса от обычного синтеза речи является то, что обученная модель может синтезировать речь голосами даже тех спикеров, которые не были представлены в наборе данных обучения.

Задача конверсии голоса заключается в преобразовании аудиосигнала с голосом исходного спикера в аудиосигнал с тем же лингвистическим содержанием, т.е. произнесенным текстом, но с произношением голосом целевого

спикера [9]. В зависимости от того, с какими голосами система может работать, конверсия голоса подразделяется на: один к одному, несколько к одному, несколько к нескольким, много к нескольким, много ко многим. Наибольший интерес представляет конверсия много ко многим, поскольку при таком типе конверсии происходит преобразование аудиосигнала с произвольными исходным и целевым голосами.

В совокупности задачи клонирования и конверсии голоса обеспечивают полный набор возможностей по преобразованию голоса речи — как для случая, когда исходная речь имеет текстовое представление, так и для случая, когда исходная речь задана в виде аудиосигнала.

За счет техники, предложенной в [10], которая заключается в использовании открытых представлений, так называемых эмбеддингов спикера, содержащих информацию о скрытых характеристиках спикера, многоголосый синтез речи можно обобщить на клонирование голоса. Современные системы синтеза речи имеют нейросетевую архитектуру [11], как правило, на основе трансформеров [12–14] и генеративных потоков [15–17]. Модели на основе генеративных потоков ко всему прочему позволяют выполнять задачу конверсии голоса за счет применения обратимых преобразований. Для построения системы, способной выполнять и синтез речи, и конверсию голоса, в настоящей работе предлагается использовать нейросетевую архитектуру на основе генеративных потоков с использованием открытых эмбеддингов спикера.

В [16, 17] авторы также используют генеративные потоки и обращают внимание на возможность выполнения конверсии голоса. Модели с использованием генеративных потоков недавно показали впечатляющие результаты в области синтеза речи, позволяя формировать разнообразные произнесения заданного текста. Однако в этих работах авторы делают основной акцент на качественный синтез речи голосом одного заданного диктора. Кроме того, работы [16, 17] без дополнительных модификаций не предусматривают возможность клонирования голоса. В отличие от [16, 17] решение, предложенное в настоящей работе, позволяет выполнять и клонирование голоса, и конверсию голоса.

Одним из недостатков моделей на основе генеративных потоков [16, 17] является монотонность синтезированной речи. Ранее в [12] было показано, что учет основного тона позволяет добиться более совершенного произношения заданным голосом. Частота основного тона является характеристикой спикера, которая, будучи переменной во времени и зависящей от лингвистического содержания речи, дополняет эмбеддинги спикера. В архитектурах моделей синтеза речи, предложенных в [12, 14], используется информация о частоте основного тона. Однако [12, 14] это трансформерные архитектуры, которые лишены возможности конверсии голоса, поэтому предложенный в этих работах подход хоть и может быть реализован в решениях на основе генеративных потоков, но не позволяет учитывать питч сигнала при выполнении конверсии голоса.

Предложенный в данной работе подход на основе потоковых генеративных моделей позволяет выполнять задачу клонирования голоса за счет использования полученных из внешней системы вещественных векторов фиксиро-

ванной размерности, содержащих информацию о спикере, т.н. эмбеддингов спикера. За счет своих архитектурных возможностей генеративные потоки позволяют одновременно с этим решать задачу конверсии голоса, таким образом обеспечивая полный набор возможностей по преобразованию голоса речи — как для случая, когда исходная речь имеет текстовое представление, так и для случая, когда исходная речь задана в виде аудиосигнала. С целью улучшения конверсии голоса в настоящей работе предложен новый способ учета частоты основного тона.

Таким образом, вклад автора в данной работе следующий:

— объединены предложенные техники использования внешних эмбеддингов спикера и декодировщика на основе генеративных потоков для создания модели, способной одновременно выполнять задачи синтеза речи несколькими голосами, клонирования голоса и конверсии голоса;
— предложен новый способ учета информации о частоте основного тона для задач синтеза речи, клонирования и конверсии голоса для решений на основе генеративных потоков.

Структура работы следующая: во втором разделе приведена архитектура предложенной модели и описан процесс ее обучения, также описана математическая модель, которая лежит в основе генеративных потоков. В третьем разделе описан процесс выполнения клонирования голоса. В четвертом разделе описан процесс выполнения задачи конверсии голоса. В пятом разделе описаны возможные техники и предложен новый подход для учета информации о частоте основного тона. В шестом разделе приведены результаты экспериментов. Заключение дано в седьмом разделе.

2. Архитектура модели

Предложенная система для выполнения синтеза речи, а также клонирования и конверсии голоса схематично изображена на рис. 1. На этом рисунке синим цветом показан сценарий синтеза речи и клонирования голоса, оранже-

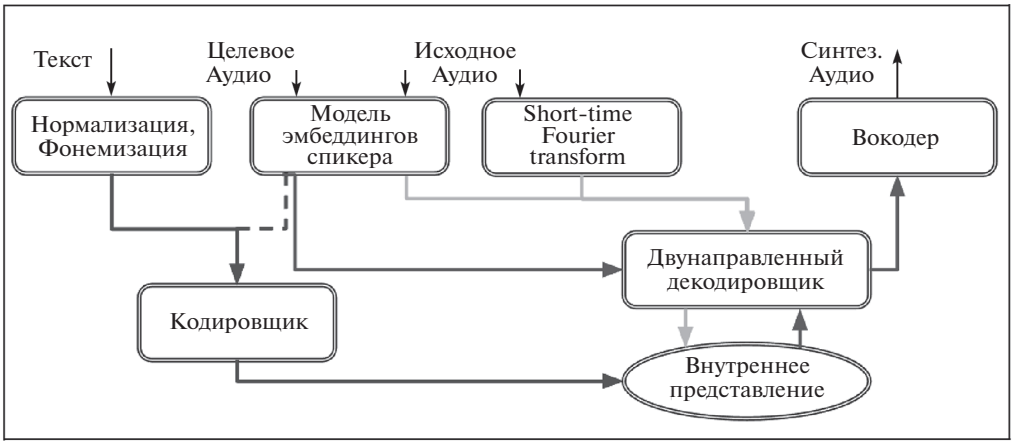


Рис. 1. Архитектура предложенной системы для выполнения синтеза речи, клонирования и конверсии голоса.

вым цветом показан сценарий конверсии голоса, красным цветом обозначены преобразования, относящиеся ко всем сценариям.

Технически, в предложенном подходе синтез речи и клонирование голоса выполняются одинаково, разница лишь в том, что при выполнении клонирования голоса для получения эмбеддинга спикера используется целевое аудио с голосом спикера, который не встречался в данных обучения модели. Таким образом, при выполнении сценария синтеза речи и клонирования голоса, текст нормализуется, т.е. приводится к упрощенному формату за счет раскрытия чисел, сокращений и пр., и фонемизируется, т.е. преобразуется в последовательность звуков, соответствующих произнесению этого текста. Затем последовательность фонем подается на вход в кодировщик, после чего полученное внутреннее представление декодируется с использованием эмбеддинга, полученного из целевого аудиосигнала. На последнем шаге полученная спектрограмма преобразуется в аудиосигнал. Подробнее сценарий клонирования голоса описан в разделе 3.

Сценарий конверсии голоса выполняется другим образом. Спектрограмма исходного аудиосигнала декодируется в обратном направлении с использованием эмбеддинга спикера, полученного из этого же сигнала, а затем полученное внутреннее представление декодируется в прямом направлении с использованием эмбеддинга, полученного из целевого сигнала. Полученная спектрограмма преобразуется в аудиосигнал. Подробнее сценарий конверсии голоса описан в разделе 4.

Два важных свойства обеспечивают выполнение описанного сценария. Во-первых, внутреннее представление не зависит от характеристик спикера, а зависит только от лингвистического содержания. Во-вторых, декодировщик является двунаправленным. Это означает, что в прямом направлении декодировщик преобразует внутреннее представление в спектрограмму, а в обратном направлении, наоборот, преобразует спектрограмму во внутреннее представление, причем эти преобразования происходят без потерь. Далее по ходу этого раздела будет показано, за счет чего данные свойства достигаются.

На приведенной схеме верхние четыре блока не являются обучаемыми. Это либо детерминированные преобразования, как в случае с нормализацией текста и преобразованием Фурье, либо преобразования, выполненные предобученными моделями. Для корректной работы предложенной системы требуется обучить центральную часть — акустическую модель, которая включает кодировщик и декодировщик, а также несколько дополнительных модулей.

Предложенная акустическая модель состоит из нескольких основных модулей: текстовый кодировщик, потоковый декодировщик и модуль предсказания продолжительностей произнесения фонем. На рис. 2 приведена схема взаимодействия этих компонент во время обучения системы. В описанной схеме обучения рассматривается случай, когда дополнительная информация о частоте основного тона не используется. Подходы учета этой информации описаны в разделе 5.

Текстовый кодировщик отображает последовательность токенов фонем $x = x_{1:Ttext}$ в скрытое векторное представление $h = h_{1:Ttext}$. После текстового кодировщика два линейных слоя используются для получения стати-

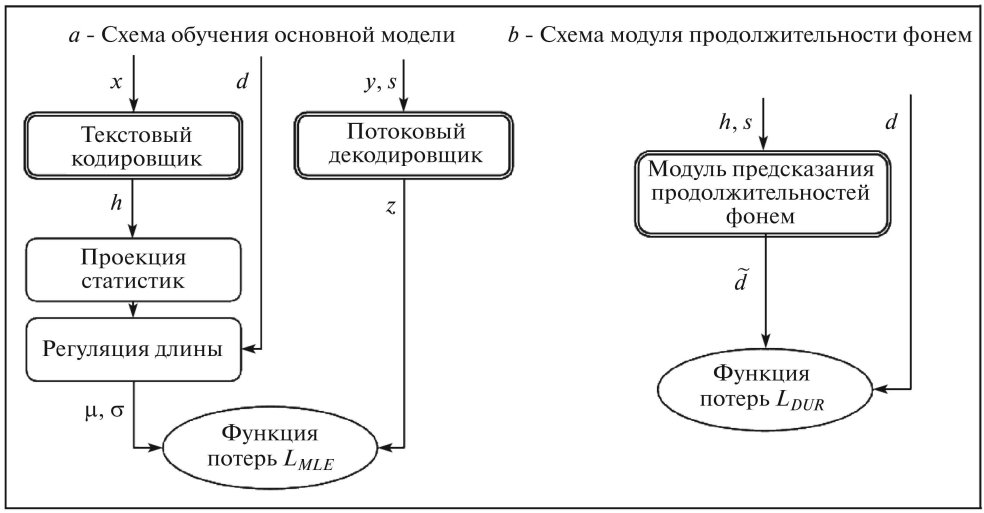


Рис. 2. Схема обучения предложенной модели.

стик $\mu = \mu_{1:Ttext}$ и $\sigma = \sigma_{1:Ttext}$ априорного распределения потокового декодера. В настоящей работе архитектура текстового кодировщика состоит из прямо направленных трансформер блоков. Заметим, что такая архитектура идентична предложенной в [16] за исключением того, что скрытая размерность и число фильтров в слоях были увеличены.

По аналогии с [16] в данной работе моделируется условное распределение спектрограмм $P_y(y \mid t, s)$ путем преобразования условного априорного распределения $P_z(z \mid t, s)$ через потоковый декодер $f_{dec} : z \to y$, где y, t и s обозначают входную спектрограмму, текстовую последовательность и информацию о спикере соответственно.

Потоковый декодировщик представляет из себя последовательность потоковых слоев, которые применяют обратимые преобразования. Такие обратимые преобразования гарантируют важное свойство потокового декодировщика — его двунаправленность. На рис. 2 направление работы декодировщика обозначено от спектрограммы y к внутреннему представлению z. В разделах 3 и 4 при выполнении клонирования и конверсии голоса будет показано, как декодировщик работает в обратном направлении.

Генеративные потоки позволяют оценить правдоподобие данных и обучаются так, чтобы максимизировать это правдоподобие. Используя замену переменных, можно вычислить логарифм правдоподобия данных следующим образом:

$$\log P_y(y \mid c) = \log P_z(z \mid c) + \log \left| \det \frac{\partial f_{dec}^{-1}(x)}{\partial x} \right|. \tag{1}$$

Априорное распределение P_z в (1) является изотропным многомерным распределением Гаусса, и процесс обучения выстраивается так, чтобы его статистики соответствовали статистическим данным априорного распределения, μ и σ, полученным из текстового кодировщика f_{enc}.

Таким образом, априорное распределение можно выразить следующим образом:

$$\log P_z(z \mid c; \theta, A) = \sum_{j=1}^{Tmel} \Big[\log N\left(z_j; \mu_{A(j)}, \sigma_{A(j)}\right) \Big[\tag{2}$$

где T_{mel} обозначает продолжительность спектрограммы.

На этапе обучения параметры модели подбираются так, чтобы максимизировать логарифм правдоподобия:

$$\max_{\theta, A} L(\theta, A) = \max_{\theta, A} \log P_y(y \mid c; \theta, A). \tag{3}$$

Для обучения предложенной модели по формулам (2) и (3) статистики априорного распределения потокового декодера требуется выровнять по фреймам спектрограммы, т.е. сопоставить индексы этих двух последовательностей. На схеме рис. 2 этот блок обозначен как регуляция длины. Индексы статистик соотносятся со спектрограммой за счет выравнивания A, полученного из внешней системы. $A(j) = i$, если j-я фонема произносится на i-м фрейме спектрограммы:

$$d_i = \sum_{j=1}^{Tmel} \Big[1_{A(j)=i}, \quad i = 1, \ldots, T_{text}. \tag{4}$$

Значения d можно интерпретировать как продолжительности фонем, поскольку до регуляции длины векторы статистик имеют такую же длину, как и последовательность токенов фонем, которая подается на вход в текстовый кодировщик.

По аналогии с системами FastPitch [12], FastSpeech [13], FastSpeech 2 [14] для того, чтобы предсказывать продолжительности фонем при выполнении клонирования голоса, обучается дополнительный модуль — предсказатель продолжительностей фонем, рис. 2,*б*. Для каждого токена входной последовательности данный модуль предсказывает число $\log d$ — логарифм количества фреймов, на протяжении которых будет длиться соответствующая фонема. Для получения продолжительности фреймов d округляется до ближайшего целого. Обучение модуля предсказания продолжительностей фонем достигается за счет минимизации среднеквадратичной ошибки между продолжительностями, полученными из выравниваний внешней системы и предсказанными:

$$L_{dur} = MSE(d, \overline{d}). \tag{5}$$

Модуль предсказания продолжительности фонем аналогичен предложенному в [13].

Заметим, что хоть на рис. 2 есть другие входы, помимо токенов текстовой последовательности x и спектрограммы y, а именно продолжительности

фонем d и эмбеддинги спикера s, однако для их получения требуется только аудио и текст.

Для построения продолжительностей фонем по формуле (4) используются выравнивания, полученные также из внешней системы. В рамках данной работы обучена собственная модель для построения выравниваний на основе смеси гауссовских моделей, на базе Kaldi Speech Recognition Toolkit [18].

Построение эмбеддингов спикера осуществляется за счет сторонней модели ECAPA-TDNN [19], так как имеет минимальную ошибку на задаче верификации спикера. Поскольку для построения эмбеддингов спикера используется предобученная модель, постольку по тексту настоящей работы они называются внешними. Для построения такого эмбеддинга необходим только аудиосигнал.

3. Клонирование голоса

Задача клонирования голоса заключается в том, чтобы синтезировать речевой сигнал образцом голоса, который не присутствовал в тренировочных данных модели синтеза речи. Образец с речью целевого голоса обычно прилагается в виде аудиофайла.

Предложенный подход позволяет использовать эмбеддинг спикера, полученный из аудиофайла с образцом целевого голоса, для синтеза речи заданным голосом. За счет того, что модель ECAPA-TDNN не ограничена никаким фиксированным набором спикеров, возможно получить эмбеддинг для голоса любого произвольного спикера.

Более подробно процедура клонирования голоса изображена на рис. 3.

Сначала для заданного речевого сигнала с голосом целевого спикера строится вектор с характеристиками этого спикера, эмбеддинг спикер s. Текст,

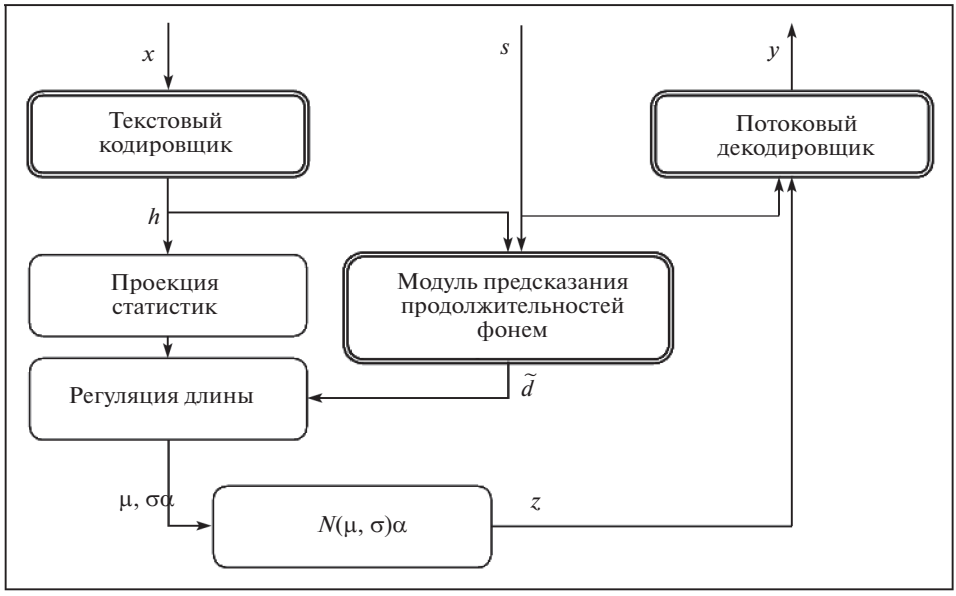


Рис. 3. Схема выполнения клонирования голоса.

который требуется озвучить, представляется в виде последовательности токенов x и направляется в текстовый кодировщик, как и во время обучения. Выход текстового кодировщика h используется в двух местах. Во-первых, вместе с эмбеддингом спикера s для предсказания продолжительностей $\overline{d}$. Во-вторых, для построения статистик μ и σ, длина которых регулируется за счет продолжительностей $\overline{d}$. Направление действия потокового декодировщика во время работы клонирования голоса меняется на противоположное относительно обучения. Случайная величина из гауссовского распределения $N(\mu, \sigma)$ проходит через потоковый декодировщик, а эмбеддинг спикера учитывается в нем как дополнительное глобальное условие. Выходом декодировщика является спектрограмма y. Для того чтобы получить аудиосигнал из спектрограммы, в настоящей работе во всех экспериментах был использован вокодер HiFi-GAN [20].

4. Конверсия голоса

Задача конверсии голоса заключается в преобразовании аудиосигнала с голосом исходного спикера в аудиосигнал с тем же лингвистическим содержанием, но произношением голосом целевого спикера. При этом конверсия голоса позволяет копировать естественную интонацию и тембр голоса исходного диктора. Образцы исходного аудиосигнала и сигнала с речью целевого голоса обычно прилагаются в виде аудиофайла.

Схема выполнения конверсии голоса изображена на рис. 4.

Модель ECAPA-TDNN позволяет получить эмбеддинги спикеров с исходным и целевым голосами s_{source}, s_{target}. За счет двунаправленности потокового декодера не составляет труда получить представление z для первоначального аудиосигнала x, в котором содержится речь исходного спикера s_{source}:

$$z = f_{dec}^{-1}(y \,|\, s_{source})\,. \tag{6}$$

Это представление не зависит от спикера, поскольку при обучении требовалось, чтобы апостериорное распределение являлось изотропным многомерным распределением Гаусса со статистиками, полученными из текстового

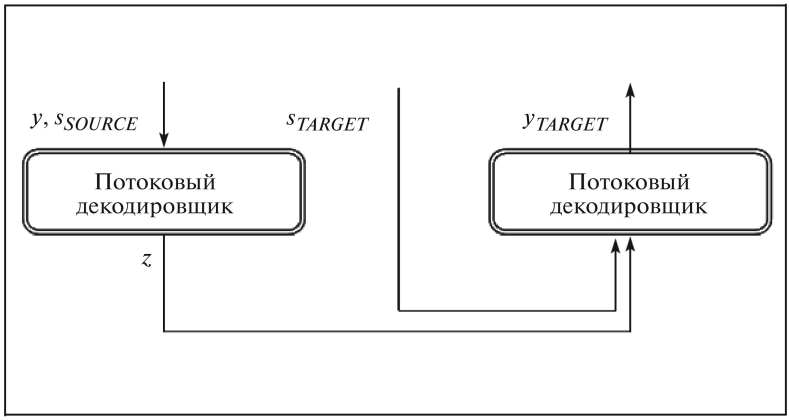


Рис. 4. Схема выполнения конверсии голоса.

энкодера. В свою очередь, эти статистики не зависят от спикера, а зависят лишь от лингвистического содержания. Таким образом, применение прямого прохода по декодировщику с условием, заданным в виде эмбеддинга целевого спикера s_{target}, позволяет получить аудиосигнал, с голосом целевого спикера и исходным лингвистическим содержанием:

$$y_{target} = f_{dec}(z \mid s_{target}). \tag{7}$$

Поскольку в процессе выполнения этих преобразований не происходит изменений продолжительностей фонем, постольку целевой голос сохранит темп речи, близкий к темпу речи исходного диктора.

Как уже упоминалось выше, модель ECAPA-TDNN позволяет получить эмбеддинги даже для спикеров, не представленных в данных обучения предложенной модели. За счет этого предложенный подход позволяет выполнять конверсию любого произвольного голоса в произвольный целевой голос, даже если образцы речи с этими голосами не встречались в данных обучения.

5. Учет частоты основного тона

В описанном выше подходе вся информация, специфичная для спикера, сосредоточена в одном эмбеддинге спикера. Однако это представление фиксированного размера и не зависящее от времени. Такое поведение не является достаточным для описания всей вариативности человеческой речи, поскольку в естественной речи присутствует и переменная во времени, специфичная каждому голосу информация.

Частота основного тона является характеристикой переменной во времени и специфичной для заданного спикера. Поэтому учет частоты основного тона должен дополнить информацию из эмбеддинга спикера и сделать синтезируемую речь более естественной.

В настоящей работе рассматриваются три следующие стратегии учета частоты основного тона:

— не учитывать;
— добавлять к выходам энкодера, по аналогии с подходами FastPitch [12] и FastSpeech2 [14];
— добавлять как локальное условие в декодер (предложенный подход).

Первая стратегия является базовым вариантом для сравнения.

Вторая стратегия используется в работах FastPitch [12] и FastSpeech2 [14]. Идея заключается в том, чтобы добавлять нормализованные значения частоты основного тона, либо полученные из них векторные представления, к выходам энкодера h. На этапе инференса при таком подходе требуется предсказывать значения частоты основного тона, для этого обучается дополнительный модуль предсказания частоты основного тона.

Недостатком такого подхода в предложенной архитектуре на основе генеративных потоков является то, что статистики μ и σ, полученные из представления h, становятся зависимыми от частоты основного тона, а значит и от переменных во времени характеристик спикера. Это не только нарушает базовую идею, заложенную в подход, так как представление z перестает быть

независимым от спикера, но и ограничивает возможности эффективного выполнения конверсии голоса.

Поэтому в настоящей работе предложена третья стратегия учета частоты основного тона спикера. Вместо того, чтобы учитывать ее на выходе кодировщика, здесь предполагается учитывать частоту основного тона в декодере как дополнительное локальное условие [21]. На этапе инференса по-прежнему потребуется предсказывать значения частоты основного тона, и для этого, как и во втором подходе, необходимо обучить дополнительный модуль предсказания частоты основного тона. Однако принципиальное отличие в том, что статистики μ и σ априорного распределения потокового декодера, как и внутреннее представление z, больше не зависят от характеристик спикера.

6. Эксперименты и результаты

Во всех экспериментах для обучения использованы открытые англоязычные данные. В табл. 1 приведена информация по каждому используемому набору данных.

Все эксперименты были проведены на машине со следующей конфигурацией: CPU: AMD Ryzen Threadripper 2950X 16-Core Processor; GPU: 3x NVidia GeForce RTX 2080 Ti.

На предварительном этапе до начала обучения тексты из набора данных были нормализованы и фонемизированы. Нормализация включала раскрытие сокращений, чисел, аббревиатур и специальных знаков. Фонемизация текста заключается в преобразовании заданного текста в последовательность фонем с учетом фонетических, морфологических и грамматических особенностей языка. В данной работе нормализация и фонемизация выполнялись с использованием инструмента Kyubyong/g2p [27]. Инструмент [27] также позволяет расставлять ударения в словах.

Для обучения рассмотренных моделей для каждого из спикеров было использовано не более двух часов данных. В обучающую выборку были включены только спикеры, для которых имелось не менее 30 минут записанной речи. Обучение каждой модели длилось три дня.

Для оценки предложенного подхода на задаче клонирования речи был проведен MOS (mean opinion score, усредненная оценка опрашиваемых) тест на естественность речи и похожесть голоса.

В рамках MOS теста на естественность речи асессору предлагалось прослушать аудиозапись и оценить их по шкале от 1 до 5, где 1 — это речь

Таблица 1. Используемые для обучения датасеты

Набор данных	Количество записей	Количество часов обучения	Среднее количество часов на спикера
Blizzard 2013 [22]	147 249	198,2	4,4
HiFi-TTS dataset [23]	323 978	291,7	29,2
LibriTTS [24]	375 086	585,8	0,26
LJSpeech [25]	13 100	23,9	23,9
M AI Labs [26]	69 853	143,6	35,9

Таблица 2. Сравнение предложенных систем на задаче клонирования голоса

	Естественность речи	Похожесть голоса
Оригинальная речь	$3{,}969 \pm 0{,}034$	$4{,}037 \pm 0{,}043$
Без учета частоты основного тона	$3{,}711 \pm 0{,}045$	$3{,}101 \pm 0{,}053$
Учет частоты основного тона сразу после кодировщика	$3{,}745 \pm 0{,}043$	$3{,}237 \pm 0{,}058$
Учет частоты основного тона как локальное условие в декодировщике	$3{,}795 \pm 0{,}04$	$3{,}306 \pm 0{,}062$

Таблица 3. Анализ MOS теста по оценке качества клонирования голоса предложенного решения по категориям

	Естественность речи	Похожесть голоса
Детские голоса	$3{,}94 \pm 0{,}081$	$3{,}24 \pm 0{,}144$
Женские голоса	$3{,}883 \pm 0{,}082$	$3{,}433 \pm 0{,}121$
Мужские голоса	$3{,}64 \pm 0{,}099$	$3{,}167 \pm 0{,}152$

совершенно неестественная, 5 — речь не отличима от человеческой. Каждую из записей оценивали по 20 раз. Всего в оценке принимало участие 75 записей для каждой из моделей, по 3 записи для каждого из 5 мужских, 5 женских и 5 детских голосов.

В рамках MOS теста на похожесть голоса асессору требовалось оценить, насколько голос в двух предложенных записях похож. Одна из предложенных записей являлась целевой записью с речью человека. Оценивание также происходило по шкале от 1 до 5, и каждая из тех же 75 записей сравнивалась с записями из оригинальной речи по 20 раз.

В табл. 2 приведено сравнение предложенных систем на задаче клонирования голоса.

В первой строке табл. 2 приведена оценка для оригинальной человеческой речи. В последующих строках приведены оценки для синтезированных записей в зависимости от способа учета в обученной модели частоты основного тона. Во второй строке оценивались записи, полученные из модели, которая не учитывает частоту основного тона никаким образом. В третьей строке приведена оценка для модели, в которой частота основного тона учитывается в представлениях, полученных из текстового кодировщика. В четвертой строке приведена оценка для модели, в которой учет частоты основного тона происходит в потоковом декодировщике.

Анализ результатов этого MOS теста показал, что система работает хуже для голосов, которые в меньшем объеме были представлены в данных обучения акустической модели, табл. 3. Так, для мужских и детских голосов результаты похожести голоса ниже, чем для женских голосов, которые присутствовали в данных обучения в большей степени. Интересно, что для детских голосов естественность речи при этом высока, но это достигается за счет того, что эти голоса больше звучат как женские, чем детские.

Таблица 4. Результаты MOS теста по оценке качества многоголосого синтеза речи

	Естественность речи	Похожесть голоса
Оригинальная речь	$4{,}163 \pm 0{,}067$	$3{,}872 \pm 0{,}05$
Предложенное решение	$3{,}859 \pm 0{,}076$	$3{,}751 \pm 0{,}053$
Модель FastPitch [12]	$3{,}556 \pm 0{,}084$	$3{,}785 \pm 0{,}058$
Модель FastSpeech 2 [14]	$3{,}965 \pm 0{,}065$	$3{,}701 \pm 0{,}067$
Модель Glow-TTS [16]	$3{,}639 \pm 0{,}056$	$3{,}639 \pm 0{,}056$

Кроме того, было проведено сравнение с другими решениями на задаче синтеза речи. В табл. 4 приведены результаты MOS теста на задаче синтеза речи. Помимо предложенного решения, в сравнении рассматривались упомянутые модели из [12, 14, 16]. Поскольку результаты при учете частоты основного тона в декодировщике оказались лучше, постольку далее в сравнении с другими системами рассматривался именно этот подход.

Несмотря на то что на задаче синтеза речи предложенное решение не является лучшим по всем критериям, работы [12, 14, 16] не позволяют выполнять клонирование голоса и работы [12, 14], не позволяют выполнять конверсию голоса.

7. Заключение

В данной работе была предложена архитектура, позволяющая выполнять задачи синтеза речи, клонирования и конверсии голоса. За счет использования внешних эмбеддингов спикера, предложенная архитектура, единожды обучившись, позволяет выполнять данные задачи даже с голосами спикеров, которые не встречались при обучении. Также предложена техника учета частоты основного тона, за счет которой удалось повысить естественность синтезированной речи и синтезировать речь, более похожую на заданный голос. Несмотря на это, результаты показывают, что речь человека звучит более естественно, а степень клонирования голоса остается недостаточно высокой.

Подход, предложенный в данной работе, является вычислительно эффективным, поскольку использует не авторегрессионный метод генерации последовательности, за счет чего асимптотика генерации аудиосигнала относительно входной последовательности является линейной. За счет того, что предложенная система требует однократного обучения, она является простой в использовании и внедрении в другие продукты.

В будущем можно улучшить качество клонирования и конверсии голоса за счет использования дополнительной информации из аудио, например, энергии сигнала, а также за счет увеличения объема данных обучения, в том числе и за счет данных из разных языков.

СПИСОК ЛИТЕРАТУРЫ

1. *Cooper F.S., Gaitenby J.H., Nye P.W.* Evolution of reading machines for the blind: Haskins Laboratories' research as a case history // J. of Rehabilit. Res. Development. 1984. No. 21.1. P. 51–87.

2. *Miyabe M., Yoshino T.* Development of multilingual medical reception support system with text-to-speech function to combine utterance data with voice synthesis / ICIC '10: Proceedings of the 3rd international conference on Intercultural collaboration. 2010. P. 195–198.
3. *Kargathara A., Vaidya K., Kumbharana C.K.* Analyzing Desktop and Mobile Application for Text to Speech Conversation / Rising Threats in Expert Applications and Solutions. 2020. P. 331–337.
4. *Sokol K., Flach P.* Glass-Box: Explaining AI Decisions With Counterfactual Statements Through Conversation With a Voice-enabled Virtual Assistant / Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence. 2018. P. 5868–5870.
5. *Hoy M.B.* Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants // Medical Reference Services Quarterly. 2018. No. 37. P. 81–88.
6. *Nasirian F., Ahmadian M., Lee O.* AI-Based Voice Assistant Systems: Evaluating from the Interaction and Trust Perspectives / Twenty-third Americas Conference on Information Systems. 2017.
7. *Obukhov D.S.* Многоголосый синтез естественной речи с использованием генеративных потоков // Современные информационные технологии и ИТ-образование. 2021. No. 17.4.
8. *Xie Q., Tian X., Liu G., et. al.* The Multi-Speaker Multi-Style Voice Cloning Challenge 2021 // International Conference on Acoustics, Speech, and Signal Processing. 2021.
9. *Sisman B., Yamagishi J., King S., Li H.* An overview of voice conversion and its challenges: From statistical modeling to deep learning // IEEE/ACM Transactions on Audio, Speech, and Language Processing. 2020.
10. *Jia Y., Zhang Y., Weiss R.J., et. al.* Transfer learning from speaker verification to multispeaker text-to-speech synthesis / Conference on Neural Information Processing Systems. 2018.
11. *Tan X., Qin T., Soong F., Liu T.-Y.* A survey on neural speech synthesis / arXiv preprint arXiv:2106.15561. 2021. URL: https://arxiv.org/pdf/2106.15561.pdf (дата обращения: 22.01.2022).
12. *Lancucki A.* Fastpitch: Parallel text-to-speech with pitch prediction / arXiv preprint arXiv:2006.06873. 2020. URL: https://arxiv.org/pdf/2006.06873.pdf (дата обращения: 22.01.2022).
13. *Ren Y., Ruan Y., Tan X., Qin T., Zhao S., Zhao Z., Liu T.-Y.* Fastspeech: Fast, robust and controllable text to speech / In Advances in Neural Information Processing Systems. 2019. P. 3165–3174.
14. *Ren Y., Hu C., Tan X., Qin T., Zhao S., Zhao Z., Liu T.-Y.* Fastspeech 2: Fast and high-quality end-to-end text to speech / arXiv:2006.0455. 2020. URL: https://arxiv.org/pdf/2006.04558.pdf (дата обращения: 22.01.2022).
15. *Valle R., Shih K., Prenger R., Catanzaro B.* Flowtron: an autoregressive flow-based generative network for text-to-speech synthesis / arXiv preprint arXiv:2005.05957. 2020. URL: https://arxiv.org/pdf/2005.05957.pdf (дата обращения: 22.01.2022).
16. *Kim J., Kim S., Kong J., Yoon S.* Glow-TTS: A Generative Flow for Text-to-Speech via Monotonic Alignment Search / In Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems. 2020.
17. *Kim J., Kong J., Son J.* Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech / arXiv preprint arXiv:2106.06103. 2021. URL: https://arxiv.org/pdf/2112.02418.pdf (дата обращения: 22.01.2022).

18. *Povey D., Ghoshal A., Boulianne G., et. al.* The Kaldi Speech Recognition Toolkit / In IEEE 2011 Workshop on Automatic Speech Recognition and Understanding. 2011.
19. *Desplanques B., Thienpondt J., Demuynck K.* Ecapatdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification, arXiv preprint arXiv:2005.07143. 2020. URL: https://arxiv.org/pdf/2005.07143.pdf (дата обращения: 22.01.2022).
20. *Kong J., Kim J., Bae J.* Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis / Advances in Neural Information Processing Systems. 2020.
21. *Oord A., Dieleman S., Zen H., et. al.* Wavenet: A generative model for raw audio / arXiv:1609.03499. 2016. URL: https://arxiv.org/pdf/1609.03499.pdf (дата обращения: 22.01.2022).
22. *King S., Karaiskos V.* The blizzard challenge 2013 / Proc. Blizzard Challenge workshop 2013. 2013.
23. *Bakhturina E., Lavrukhin V., Ginsburg B., Zhang Y.* Hi-fi multi-speaker english tts dataset / arXiv preprint arXiv:2104.01497. 2021.
URL: https://arxiv.org/pdf/2104.01497.pdf (дата обращения: 22.01.2022).
24. *Zen H., Dang V., Clark R. et. al.* LibriTTS: A corpus derived from LibriSpeech for text-to-speech / arXiv preprint arXiv:1904.02882. 2019.
URL: https://arxiv.org/abs/1904.02882 (дата обращения: 22.01.2022).
25. *Ito K., Johnson L.*, The LJ speech dataset / Электронный ресурс: The LJ Speech Dataset. 2017. URL: https://keithito.com/LJ-Speech-Dataset/ (дата обращения: 22.01.2022).
26. *Solak I.* The M-AILABS Speech Dataset / Электронный ресурс: The M-AILABS Speech Dataset. 2019. URL: https://www.caito.de/2019/01/the-m-ailabs- speech-dataset (дата обращения: 22.01.2022).
27. *Kyubyong P., Jongseok K.* g2pE: A Simple Python Module for English Grapheme To Phoneme Conversion / Электронный ресурс: GitHub repository. 2018. URL: https://github.com/Kyubyong/g2p (дата обращения: 22.04.2022).

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 22.01.2022

После доработки 25.04.2022

Принята к публикации 29.06.2022

© 2022 г. А.В. БОБКОВ, канд. техн. наук
(Alexander.Bobkov@bmstu.ru),
Х. АУНГ (happyland27057@gmail.com)
(Московский государственный технический университет
им. Н.Э. Баумана, Москва)

ИДЕНТИФИКАЦИЯ ЧЕЛОВЕКА ПО ВИДЕОИЗОБРАЖЕНИЮ В РЕАЛЬНОМ ВРЕМЕНИ НА ОСНОВЕ СЕТЕЙ YOLOv2 И VGG 16

Данная работа посвящена задаче распознавания лиц по видео. На сегодняшний день методы распознавания лиц сделали большой шаг вперед, однако распознавание видео с его низким качеством, сложными условиями освещенности и требованиями работы в реальном времени попрежнему остается сложной и до конца нерешенной задачей.

В работе используется аппарат сверточных сетей для различных этапов обработки: для захвата и обнаружения лица, для построения вектора признаков, и наконец, для распознавания. Все алгоритмы реализованы и исследованы в среде Matlab для упрощения их дальнейшего экспорта во встраиваемые приложения.

Ключевые слова: сверточная нейронная сеть VGG16, распознавание лиц, алгоритм обнаружения объектов YOLOv2, глубокое обучение, база данных лиц.

DOI: 10.31857/S0005231022100099, **EDN:** AKSNXL

1. Введение

Задача распознавания лиц — одна из наиболее интересных тем в области компьютерного зрения. Это способность распознавать или идентифицировать личность человека, анализируя различные черты лиц. Система распознавания лиц обеспечивает огромные преимущества по сравнению с другими решениями биометрической безопасности, такими как распознавание радужной оболочки глаза и отпечатков пальцев. Система фиксирует биометрические измерения человека с определенного расстояния, не взаимодействуя с ним. В приложениях для сдерживания преступности эта система может помочь многим организациям идентифицировать человека, у которого есть какое-либо уголовное прошлое или другие юридические проблемы.

В последние годы задача распознавания лиц оставалась крайне популярной и актуальной темой исследований, и в этом направлении было разработано значительное количество методов, которые обеспечивают очень высокую точность распознавания и которые можно использовать на практике [1, 2]. Из классических методов машинного обучения наиболее широкое распространение получили фильтр Виолы–Джонса и его модификация, использующая гистограмму направленности градиента. Основным направлением развития исследований сегодня остается поиск эффективных подходов с использованием сверточных сетей и глубокого обучения.

Современный подход к распознаванию лиц связан с построением сетей со сложной топологией и большим числом сверточных слоев, что обеспечивает возможность высокоточной классификации при наличии смены ракурса, мимики, маскирующих признаков и т.д. Это такие сети, как VGG-Face [11], Google FaceNet [12], Facebook DeepFace [13], FaceID и другие. Однако попытки применять их распознаванию лиц на видеоизображении дают низкие результаты из-за низкой скорости работы и низкого качества изображения. Попытки же использовать более быстрые и менее точные сети, такие как OpenFace на основе MobieNet [14], сразу ведут к резкому снижению точности распознавания.

При распознавании лиц на видео, помимо высокой точности, также требуется обеспечить высокую скорость обработки больших массивов информации, причем, как правило, качество изображения достаточно низкое из-за воздействия шума и ракурса съемки. Тем не менее видео несет в себе гораздо больше информации о лице, нежели одиночная фотография. Все это заставляет искать новые методы для распознавания лиц по видео. Особенности задачи распознавания по видео не позволяют решить задачу каким-либо одним инструментом, например, обучив нейронную сеть, как во многих других задачах: такое решение будет либо слишком громоздко и неспособно работать в режиме реального времени, либо, наоборот, будет быстрым, но неспособным решать задачу с требуемой точностью. Решением здесь будет являться использование совокупности сетей, каждая со своими свойствами, для наиболее качественного решения задач каждого из этапов распознавания.

В данной работе рассматривается подход, связанный с обнаружением и отслеживанием области интереса при помощи быстродействующей поисковой сети. Наличие области интереса позволяет распознавать не все изображение во все моменты времени, а лишь отдельные фрагменты наиболее удачных кадров. Потенциально такой подход позволяет повышать качество распознавания, собирая более качественное изображение по последовательности кадров, с коррекцией ракурса съемки и восстановлением трехмерной формы, но в данной работе данная задача не ставилась.

Методы детектирования лиц традиционно делятся на методы жестких шаблонов и методы деформируемых моделей.

Методы деформируемых моделей строят модель лица на основе деформируемых частей [15–17] для моделирования потенциальной деформации между элементами лица. Методы также могут сочетать обнаружение всего лица и локализацию его элементов [18].

Методы с использованием жестких шаблонов, в свою очередь, делятся на следующие группы:

— каскадные фильтры и вариации бустинга; к основным представителям этого семейства алгоритмов относятся алгоритм распознавания лиц Виолы–Джонса и его варианты [1, 19];

— методы на основе обобщенного преобразования Хафа и его вариации [20, 21];

— алгоритмы, основанные на сверточных нейронных сетях и сетях глубокого обучения [22, 23–25].

Сети глубокого обучения в последнее время показывают очень высокую точность, поэтому в настоящее время именно они рассматриваются как наиболее перспективный подход для поиска лиц.

В данной статье используется аппарат сверточных сетей с отдельным детектором общих признаков, детектором лиц и классификатором лиц.

В роли детектора признаков выступает относительно небольшая и производительная сеть ResNet-18 с редуцированными верхними слоями, предварительно обученная на большом количестве классов объектов.

В качестве быстродействующего детектора лиц выбрана модифицированная сеть YOLOv2 [5] (You Only Look Once — посмотри на изображение только один раз). Сеть YOLO обладает очень высокой производительностью и широко применяется, например, в задачах распознавания дорожных сцен, однако ее применение для идентификации лиц затруднено низкой точностью классификации.

Для решения задачи окончательного высокоточного распознавания лица в найденном положении использовалась предварительно обученная сверточная сеть VGG16 [10] без последних слоев многослойного классификатора. Эта сеть обеспечивает построение вектора признаков, устойчивого к изменениям освещенности и к небольшим изменениям ракурса, что является важным для рассматриваемой задачи.

Наконец, для поиска наилучшего совпадения с параметрами лиц из базы использовалась косинус-метрика, величину которой удобно рассматривать как вероятность совпадения лиц.

Для упрощения экспорта полученных алгоритмов на целевую вычислительную платформу решено было использовать среду MATLAB. Данная среда, с одной стороны, содержит библиотеки с открытым исходным кодом, доступным для изучения и воспроизведения, а с другой — позволяет непосредственно переносить написанный код на другие языки программирования.

Использование среды MATLAB потребовало создания интерактивной среды для исследования алгоритмов и методов распознавания. Для этого использовался дизайнер приложений MATLAB, который представляет собой интерактивную среду разработки для проектирования макета приложения и программирования его поведения.

Исследование, проведенное в статье, направлено на изучение работоспособности, тестирование производительности и сравнение точности результатов обнаружения и распознавания лица комбинации методов YOLOv2 и VGG16 с другими известными методами. Эксперименты показали как высокую производительность подхода, позволяющего работать в режиме реального времени, так и высокую точность, позволяющую использовать подход в реальных практических задачах.

2. Обнаружение лиц с использованием метода YOLOv2 на базе ResNet-18

Сеть ResNet — это одна из самых мощных глубоких нейронных сетей, которая достигла прорывных результатов в классификации ILSVRC 2015 [7].

Таблица 1. Архитектура ResNet-18 для извлечения признаков

Имя слоя	Размер фильтра	Размер выхода
Входной слой изображения	224×224×3	224×224×3
Conv_1	7×7×64 Maxpool 3×3	112×112
Conv_2	[3×3×64]×2	56×56
Conv_3	[3×3×128]×2	28×28
Conv_4	[3×3×256]×2	14×14
Входной слой признаков	14×14	14×14

ResNet добилась отличных результатов обобщения по другим задачам распознавания и заняла первое место по обнаружению ImageNet, локализации ImageNet, обнаружению COCO и сегментации COCO в конкурсах ILSVRC и COCO 2015. Существует много вариантов архитектуры ResNet: это одна и та же структура, но с разным количеством слоев. Различные версии ResNet — это ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-110 и т.д.

В данной работе использовалась предварительно обученная сеть ResNet-18 только для извлечения признаков изображений. Размер входного изображения составляет 224×224 с RGB. После того, как изображение прошло через сеть, на выходе получается отклик размером 14×14, т.е. сеть осуществляет 16-кратную понижающую дискретизацию. Это достаточно для извлечения признаков лица (табл. 1).

Выход алгоритма обнаружения объектов YOLO v2 — отклик размера $S \times S$, где S — количество ячеек сетки. Каждая ячейка содержит пять параметров (x, y, w, h) и $Pr(obj)$, где x, y — координаты центра ограничивающей рамки, w, h — ее ширина и высота, $Pr(obj)$ — вероятность нахождения объекта внутри рамки. Показатель достоверности отражает вероятность включения в модель целевого объекта и точность блока обнаружения предсказания. Показатель $C(obj)$ достоверности определяется так:

$$C(obj) = Pr(obj)^* IoU(Pred, Gtruth).$$

Если искомый объект отсутствует в ячейке, то $Pr(obj)$ будет равен нулю, а доверительный балл должен быть равен нулю: $C(obj) = 0$.

IoU — это величина перекрытия найденной ограничивающей рамки и рамки из обучающей выборки, т.е. отношение их пересечения и объединения:

$IoU(Pred, Gtruth)$ = (перекрывающаяся область предсказанной рамки и рамки обучающей выборки)/(вся область предсказанной рамки и рамки обучающей выборки).

После получения достоверности каждой рамки, рамки с низкой достоверностью удаляется путем сравнения с пороговым значением, а затем выполняется удаление оставшихся рамок, отклик которых не является локальным максимумом.

Метод YOLO v2 использует суммарную квадратическую ошибку в качестве функции потерь. Метод пытается оптимизировать следующие многосо-

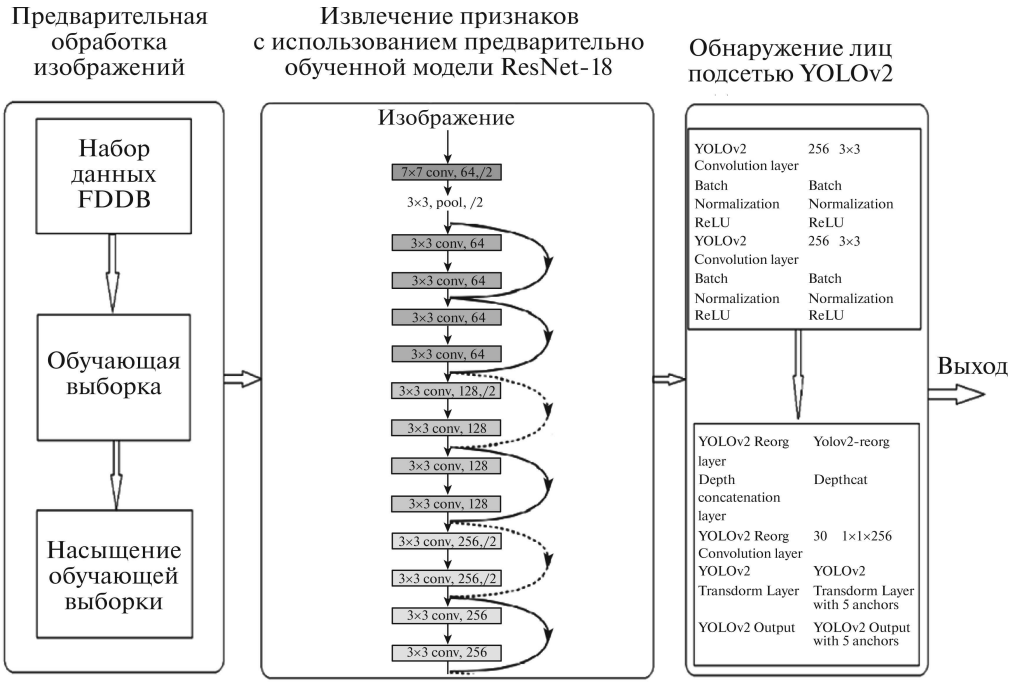


Рис. 1. Схема процесса обнаружения лиц предлагаемой модели.

ставные потери: потери локализации (ошибка определения положения объекта), потери доверия (ошибка определения вероятности обнаружения) и потери классификации (ошибка определения класса объекта).

$$\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\left[\mathbb{1}_{ij}^{obj}\left[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2\right]\right]+$$

$$+\lambda_{coord}\sum_{i=0}^{S^2}\left[\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}\left[\left(\sqrt{w_i}-\sqrt{\hat{w}_i}\right)^2+\left(\sqrt{h_i}-\sqrt{\hat{h}_i}\right)^2\right]\right]+$$

$$+\sum_{i=0}^{S^2}\left[\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}\left(C_i-\hat{C}_i\right)^2\right]+\lambda_{noobj}\sum_{i=0}^{S^2}\left[\sum_{j=0}^{B}\mathbb{1}_{ij}^{noobj}\left(C_i-\hat{C}_i\right)^2\right]+$$

$$+\sum_{i=0}^{S^2}\left[\mathbb{1}_{i}^{obj}\sum_{C\in classes}\left[\left(P_i(c)-\hat{P}_i(c)\right)^2\right],\right.$$

где $\mathbb{1}_{ij}^{obj}=1$ если j-й граничный прямоугольник в i-й ячейке сетки отвечает за обнаружение объекта, то в противном случае 0;

$\lambda_{coord}=$ увеличение веса для потери в координатах граничного поля, по умолчанию 5,

$\lambda_{noobj}=0{,}5$;

$(x, y, w, h) =$ потери локализации между обучающей выборкой и предсказанной рамкой;

C — потери достоверности обнаружения и $P(c)$ — вероятности принадлежности к классу. На рис. 1 представлена схема процесса обнаружения лиц предлагаемой модели.

3. Распознавание лиц с использованием предварительно обученной модели VGG16

Для распознавания лиц использовалась предварительно обученная модель сети VGG16. Модель VGG16 имеет большое количество гиперпараметров. Размер входного изображения первого слоя составляет 224×224 с кодированием RGB. Изображение пропускается через последовательность сверточных слоев, в которых использовался сверточный фильтр размером 3×3 с шагом 1, и всегда используется один и тот же слой субдискретизации maxpool 2×2 с шагом 2. Расположение слоев в этой архитектуре выглядит следующим образом: сверточные слои, слои ReLU и слои субдискретизации. В конце модели есть два полносвязных слоя, за которыми следует слой классификатора softmax для вывода данных. Эта сеть VGG16 является довольно большой сетью и имеет около 138 млн обучаемых параметров (рис. 2). При предъявлении сети изображения лица на выходе сети появляется его описание в виде вектора признаков. При этом одинаковые лица будут иметь схожие признаки, а разные соответственно несхожие, даже при наличии мешающих факторов: изменения ракурса, освещенности и т.д. Это позволяет распознавать лица, заранее неизвестные сети, путем сравнения вектора признака, сгенерированного сетью, с ранее заданным образцом.

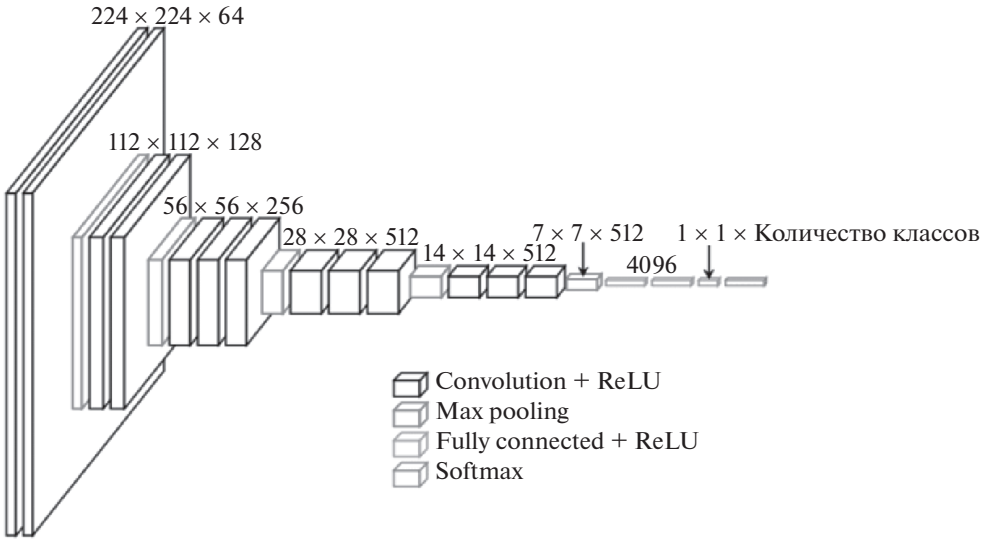


Рис. 2. Предварительно обученная сетевая модель VGG16.

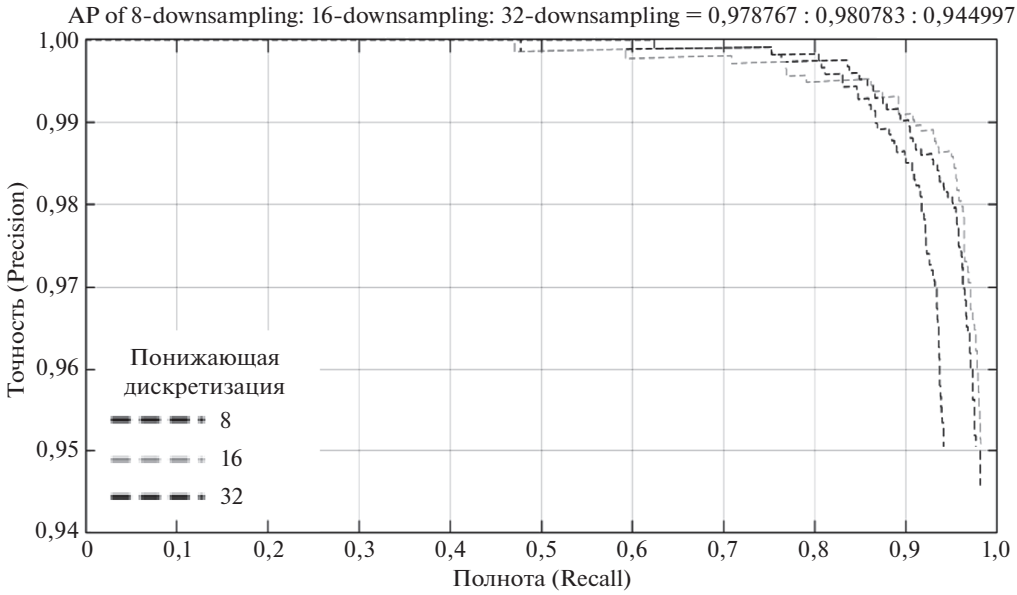


Рис. 3. Сравнение результатов по различным критериям понижающей дискретизацией.

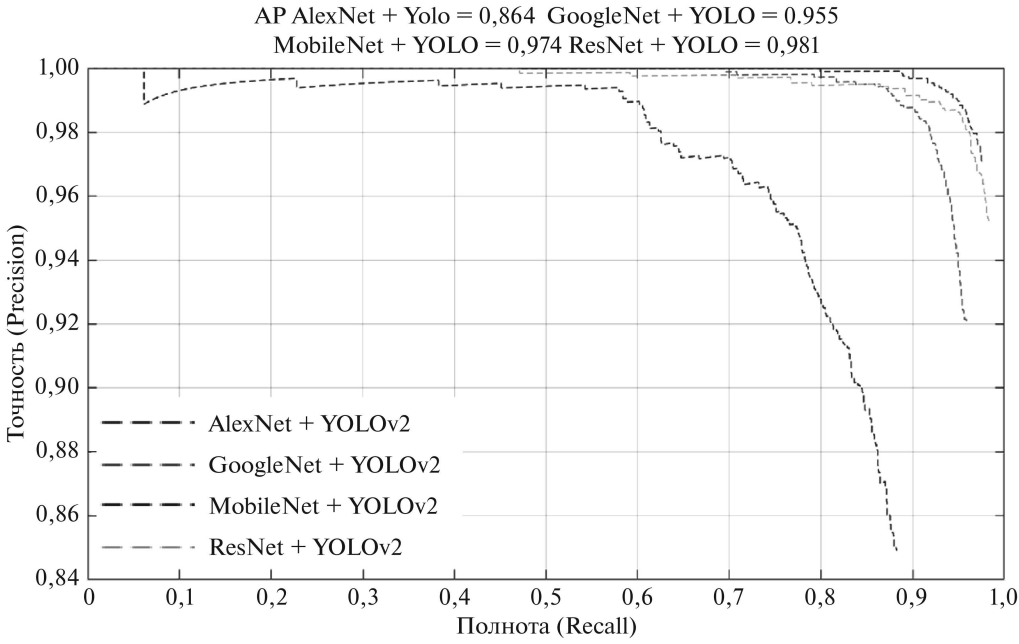


Рис. 4. Сравнение результатов на основе других предварительно обученных сетевых моделей.

Таблица 2. Сравнение результатов по различным критериям понижающей дискретизации

Понижающая дискретизация	Средняя точность
8-кратная	0,978%
16-кратная	0,980%
32-кратная	0,945%

Таблица 3. Сравнение результатов на основе других предварительно обученных сетевых моделей

Архитектура	Средняя точность набора тестовых изображений
Alexnet+YOLOv2	0,86%
Googlenet+YOLOv2	0,95%
Mobilenet+YOLOv2	0,97%
Resnet18+YOLOv2 (предлагаемая модель)	0,98%

Таблица 4. Точность распознавания лиц с использованием сети VGG 16

Сеть	База данных FEI	База данных Face94
VGG16	98%	98,5%

4. Результаты экспериментов

В табл. 2 и на рис. 3 представлены результаты экспериментов по обнаружению лиц сетями с различной кратностью понижающей дискретизации. Из таблицы видно, что 16-кратная понижающая дискретизация более эффективна для извлечения признаков лица: на тестовых данных была получена средняя точность 98%. В табл. 3 и на рис. 4 представлено сравнение результатов на основе других предварительно обученных сетевых моделей. На рис. 5 представлены примеры работы предлагаемой модели системы обнаружения лиц.

В табл. 4 представлены результаты исследования точности распознавания лица с использованием VGG16. Здесь использовались две различные базы данных (FEI и Face94) и получили точность распознавания более 98%. На рис. 6 представлены примеры работы полученного алгоритма поиска и распознавания лиц, реализованного в виде приложения MATLAB.

Для реализации системы, способной работать в режиме реального времени, был использован тулбокс App designer среды MATLAB. App designer позволяет инженерам и исследователям легко создавать профессиональные приложения, не требуя специализированных навыков программирования. App designer объединяет две основные задачи создания приложений: создание визуальных компонентов графического пользовательского интерфейса (GUI) и программирование поведения приложения. App designer также предоставляет

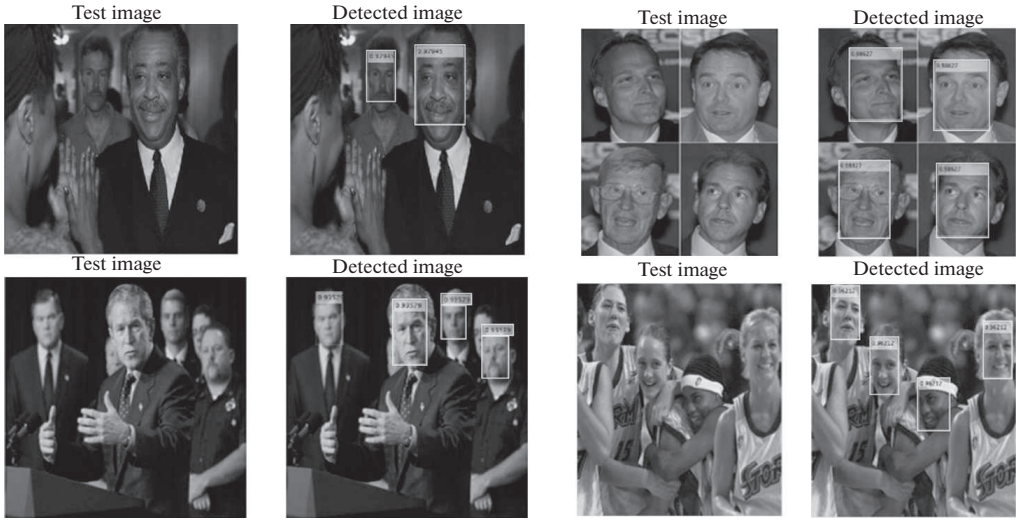


Рис. 5. Пример работы предлагаемой модели обнаружения лиц.

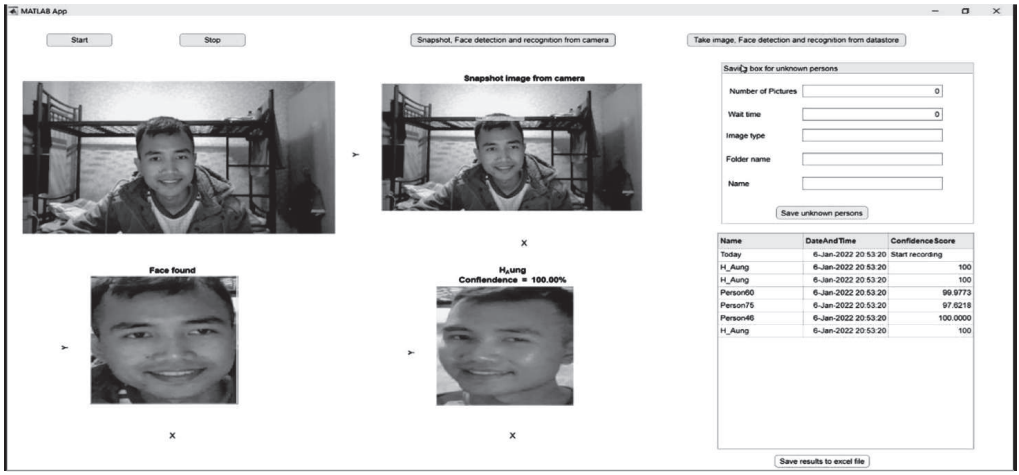


Рис. 6. Общий вид интерфейса системы распознавания лиц по видеоизображению.

различные библиотеки компонентов, такие как кнопки, флажки, оси пользовательского интерфейса, раскрывающиеся списки и т.д. Используя его, была создана система распознавания лиц, общий вид интерфейса которой показан на рис. 6.

5. Заключение

Эксперименты показали, что система обнаружения лиц на базе YOLOv2 обладает не только высокой точностью, но и позволяет находить лицо на видео в режиме реального времени, с высокой скоростью обнаружения. Для этапа распознавания лиц использовалась предварительно обученная сетевая

модель VGG16, перенесенная в MATLAB. Для 101 разной персоны и общего количества изображений 5050 получена точность более 98%. Это довольно хороший результат, достаточный для работы многих приложений захвата и распознавания изображения с видеокамеры в реальном времени.

СПИСОК ЛИТЕРАТУРЫ

1. *Viola P., Jones M.* Rapid object detection using a boosted cascade of simple features // Institute of Electrical and Electronics Engineers (IEEE), 15 April 2003, ISSN: 1063-6919, 9 pages, https://doi.org/10.1109/CVPR.2001.990517.
2. *Guennouni S., Ahaitouf A., Mansouri A.* Face Detection: Comparing Haar-like combined with Cascade Classifiers and Edge Orientation Matching // Institute of Electrical and Electronics Engineers (IEEE), 29 May 2017, ISBN:978-1-5090-6681-0, 4 pages, https://doi.org/10.1109/WITS.2017.7934604.
3. *Хачумов М.В., Нгуен Т.З.* Распознавание лиц по фотографиям на основе инвариантных моментов // Современные проблемы науки и образования. 2015. № 2-2, url: http://science-education.ru/ru/article/view?id=23235 (дата обращения: 23.05.2021).
4. *Рудинская Е.А., Парингер Р.А.* Разработка алгоритма детектирования лиц с использованием комбинаций каскадов Хаара // Сб. тр. ИТНТ-2019. Новая техника. 2019. С. 6–12.
5. *Redmon J., Farhadi A.* YOLO9000: Better, Faster, Stronger // Institute of Electrical and Electronics Engineers (IEEE), 09 November 2017, ISSN: 1063-6919, 9 pages, https://doi.org/10.1109/CVPR.2017.690.
6. *Simonyan K., Zisserman A.* Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556v6 [cs.CV] // 10 Apr 2015, 14 pages.
7. *Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun.* Deep Residual Learning for Image Recognition, arXiv:1512.03385v1 [cs.CV] // 10 Dec 2015.
8. *Коломиец В.* Анализ существующих подходов к распознаванию лиц [Электронный ресурс]. URL: http://habrahabr.ru/company/synesis/blog/ 238129/ (дата обращения: 15.12.2021).
9. *Redmon J., Santosh D., Girshick R., Farhadi A.* You only look once: Unified, real-time object detection // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779–788. Las Vegas, NV: CVPR, 2016.
10. *Russakovsky O., Deng J., Su H., et al.* ImageNet Large Scale Visual Recognition Challenge // International Journal of Computer Vision (IJCV). 2015, Vol. 115, Issue 3, p. 211–252.
11. *Qawaqneh Z., Mallouh A.A., Barkana B.D.* Deep convolutional neural network for age estimation based on VGG-face model // arXiv preprint arXiv:1709.01664. – 2017.
12. *Schroff F., Kalenichenko D., Philbin J.* FaceNet: A unified embedding for face recognition and clustering // Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2015.
13. *Taigman Y., Yang M., Ranzato M., Wolf L.* DeepFace: Closing the gap to human-level performance in face verification // Proc IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 2014, p. 1701–1708.
14. *Amos B., Ludwiczuk B., Satyanarayanan M.* Openface: A general-purpose face recognition library with mobile applications // CMU School of Computer Science. 2016, Vol. 6. No. 2, p. 20.

15. *Felzenszwalb P.F., Girshick R.B., McAllester D., Ramanan D.* Object detection with discriminatively trained part-based models // IEEE Trans. Pattern Anal. Mach. Intell. 2010, Vol. 32. Issue 9, p. 1627–1645.
16. *Felzenszwalb P.F., Huttenlocher D.P.* Pictorial structures for object recognition // Int. J. Comput. Vision. 2005, Vol. 61. Issue 1, p. 55–79.
17. *Fischler M.A., Elschlager R.A.* The representation and matching of pictorial structures // IEEETrans. Comput. 1973, Vol. 22, Issue 1, p. 67–92.
18. *Zhu X., Ramanan D.* Face, detection pose estimation, and landmark localization in the wild // 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, p. 2879–2886.
19. *Viola P., Jones M.J.* Robust real-time face detection // Int. J. Comput. Vis. 2004, Vol. 57. Issue 2, p. 137–154.
20. *Li H., Lin Z., Brandt J., Shen X., Hua G.* Efficient boosted exemplar-based face detection // 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013.
21. *Shen X., Lin Z., Brandt J., Wu Y.* Detecting and aligning faces by image retrieval // 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2013, p. 3460–3467.
22. *Krizhevsky A., Sutskever I., Hinton G.E.* Imagenet classification with deep convolutional neural networks // Advances in Neural Information Processing Systems, 2012, p. 1097–1105.
23. *LeCun Y., Bottou L., Bengio Y., Haffner P.* Gradient-based learning applied to document recognition // Proc. IEEE. 1998, Vol. 86. Issue 11, p. 2278–2324.
24. *Girshick R., Donahue J., Darrell T., Malik J.* Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. https://doi.org/10.48550/arXiv.1311.2524.
25. *Zhang C., Zhang Z.* Improving multiview face detection with multi-task deep convolutional neural networks // 2014 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2014, p. 1036–1041.

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 17.02.2022

После доработки 22.04.2022

Принята к публикации 29.06.2022

© 2022 г. Е.А. КАРАЦУБА, д-р. физ.-мат. наук
(ekaratsuba@gmail.com)
(ФИЦ "Информатика и управление" РАН, Москва)

БЫСТРЫЙ АЛГОРИТМ ВЫЧИСЛЕНИЯ ПСИ-ФУНКЦИИ[1]

Памяти Константина Владимировича Рудакова

Построен быстрый алгоритм вычисления логарифмической производной гамма-функции Эйлера, основанный на БВЕ методе. Сложность алгоритма близка к оптимальной. Структура алгоритма допускает его распараллеливание.

Ключевые слова: быстрые алгоритмы, пси-функция, гамма-функция Эйлера, сложность вычисления, метод БВЕ.

DOI: 10.31857/S0005231022100105, **EDN:** ALDLAI

1. Введение. Быстрые алгоритмы. Метод БВЕ

Быстрые алгоритмы предназначены для оптимизации высокоточных вычислений. Основным критерием оценки быстрых алгоритмов является сложность вычисления. Первые задачи по оценке (битовой) сложности вычисления функции/арифметической операции с точностью до n знаков были сформулированы А.Н. Колмогоровым в 1950-х гг. (см. [1, 2]), первый быстрый алгоритм (алгоритм умножения n-значных чисел) был найден А.А. Карацубой в 1960 г. (см. [1–3]), что привело в дальнейшем к созданию серии алгоритмов быстрого вычисления различных функций (алгоритм умножения эквивалентен алгоритму вычисления функции $y = x^2$) со сложностью $O\left(n^{1+\varepsilon}\right)$ для любого $\varepsilon > 0$ и $n > n_1(\varepsilon)$ (вместо $O\left(n^{2+\varepsilon}\right)$ — лучшей сложности вычисления функций до эры быстрых алгоритмов).

Далее считаем, что числа записаны в двоичной системе счисления, знаки которой 0 и 1 называются битами.

Определение 1. Запись знаков 0, 1, *плюс, минус, скобка; сложение, вычитание и умножение двух битов назовем одной элементарной или битовой операцией.*

Определение 2. Вычислить (вещественную) функцию $y = f(x)$ *в точке* $x = x_0$ *с точностью до* n *знаков, значит найти такое число* A, *что* $|f(x_0) - A| \quad 2^{-n}$.

Определение 3. Количество битовых операций, достаточное для вычисления функции $f(x)$ *в точке* x_0 *с точностью до* n *знаков посредством данного алгоритма, называется сложностью вычисления* $f(x)$ *в точке* x_0 *и обозначается* $s_f(n) = s_{f,x_0}(n)$.

[1] Работа выполнена при финансовой поддержке Российского научного фонда (грант № 22-21-00727).

Будем далее предполагать, что для сложности умножения двух n-значных чисел справедлива оценка

$$M(n) = O\left(n^{1+\varepsilon}\right), \tag{1}$$

т.е. сложность используемого алгоритма умножения не хуже, чем $M(n) = O(n \log^C n)$, где C — константа (см. алгоритмы из [4, 5]).

Определение 4. Будем называть **быстрыми** *такие алгоритмы вычисления функции f, что для этих алгоритмов*

$$s_f(n) = O\left(n^{1+\varepsilon}\right) \text{ для любого } \varepsilon > 0 \text{ и } n > n_1(\varepsilon).$$

В теории быстрых алгоритмов основным растущим параметром является точность вычисления n, $n \to \infty$. Поскольку только запись некоторого числа с точностью до n знаков требует не менее, чем $n+1$ операций, из определения 4 следует, что быстрым вычислительным алгоритмам соответствует правильный порядок оценки сверху сложности вычисления $s_f(n)$ по n, $n \to \infty$,

$$n < s_f(n) < n^{1+\varepsilon}.$$

В 1975 г. были предложены первые алгоритмы быстрого вычисления элементарных алгебраических функций [6]. Например, самый простой алгоритм деления числа a на число b заключается в вычислении методом Ньютона обратной величины $\frac{1}{b}$ с точночтью до n знаков с последующим умножением на a по алгоритму быстрого умножения.

В 1976 г. (см., например, [7]) были предложены первые быстрые алгоритмы вычисления константы π, а затем и простейших трансцендентных функций, основанные на АГС-методе Гаусса (см., например, [8]). В [9], кроме алгоритмов вычисления простейших трансцендентных функций, основанных на АГС-методе Гаусса, содержатся также алгоритмы вычисления некоторых высших трансцендентных функций (гамма-функции Эйлера, например). Однако сложность этих алгоритмов несколько хуже определенной выше сложности быстрых алгоритмов и оценивается как

$$O\left(n^{3/2+\varepsilon}\right).$$

В 1991 г. автором был построен новый быстрый метод — метод Быстрого Вычисления Е-функций (БВЕ) (см. [10], см. также [11, 12], о Е-функциях — см. [13]).

В настоящее время известны два быстрых метода вычисления простейших трансцендентных функций — метод АГС и метод БВЕ. При этом с помощью БВЕ удается построить быстрые алгоритмы и для вычисления некоторых высших трансцендентных функций.

Иногда метод БВЕ по ошибке принимают за вариант метода Байнэри Сплиттинг (Binary Splitting). В 2010 г. Билл Роско (Andrew William Roscoe) глава факультета информатики Оксфордского университета (Department of Computer Science, University of Oxford), ученик Тони Хоара (Charles Antony

Richard Hoare) ученика Андрея Николаевича Колмогорова протестировал оба метода — БВЕ и Байнэри Сплиттинг и, отметив, что ошибки от шага к шагу в этих методах "копятся" по-разному, сделал объективный вывод о том, что это разные методы. С другой стороны, ни в одной из публикаций до 1991 г. не содержалось ни одного быстрого алгоритма вычисления какой-либо трансцендентной функции или классической константы, основанного на Байнэри Сплиттинг. Похоже, метод Байнэри Сплиттинг был неудачной попыткой воспользоваться методом А.А. Карацубы (изложен в [1–3]) для получения алгоритмов эффективного вычисления трансцендентных функций. При этом только ошибки в сложности вычисления могли неправильно сориентировать тех, кто решил, что Байнэри Сплиттинг является быстрым методом вычисления трансцендентных функций.

Метод БВЕ — это метод суммирования рядов специального вида. С помощью БВЕ можно вычислить любую элементарную трансцендентную функцию для любого аргумента, классические константы e, π, постоянную Эйлера γ, постоянные Апери и Каталана, такие высшие трансцендентные функции как гамма-функцию Эйлера, гипергеометрические функции, сферические функции, цилиндрические функции и т.д. для алгебраических значений аргумента и параметров, такие специальные интегралы, как интеграл вероятности, интегралы Френеля, интегральную экспоненциальную функцию, интегральные синус и косинус и т.д. с оценкой сложности

$$s_f(n) = O\left(M(n)\log^2 n\right) = O\left(n^{1+\varepsilon}\right),$$

для любого $\varepsilon > 0$ и $n > n_1(\varepsilon)$. Структура метода БВЕ дает возможность распараллеливания основанных на БВЕ алгоритмов.

В 2008 г. профессор Эрик Бах (Eric Bach, Univ. of Wisconsin, Madison) в письме заметил, что никто не знает, как быстро вычислить пси-функцию (про ψ-функцию см., например, [14]). Алгоритм вычисления пси-функции, основанный на БВЕ-методе, казался мне достаточно очевидным, но поскольку за эти годы никто его не описал, настоящая статья призвана восполнить это упущение.

2. Гамма-функция и пси-функция

Одна из самых широко распространенных в анализе и математической физике высших трансцендентных функций гамма-функция Эйлера определяется соотношением (см., например, [14])

$$\Gamma(z) = \int_0^{+\infty} t^{z-1}e^{-t}dt, \qquad \Gamma(z+1) = z\Gamma(z), \quad Re\, z > 0, \tag{2}$$

$\Gamma(N+1) = N!$, где N — натуральное число. Логарифмическая производная гамма-функции называется пси-функцией

$$\psi(z) = \frac{d}{dz}\log\Gamma(z), \quad \psi(z+1) = \frac{1}{z} + \psi(z). \tag{3}$$

Представление пси-функции в виде ряда

$$\psi(z) = -\gamma + \sum_{k=0}^{\infty} \left(\frac{1}{k+1} - \frac{1}{z+k} \right), \quad z \neq 0, -1, -2, \ldots, \tag{4}$$

где γ есть константа Эйлера

$$\gamma = \lim_{n\to\infty} \left(1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} - \log n \right),$$

не дает возможности быстро вычислить эту функцию ввиду медленной сходимости ряда из правой части (4) (о быстром алгоритме вычисления константы Эйлера см. [10, 15]).

Далее для простоты будем рассматривать лишь случаи вещественного аргумента (для комплексного аргумента все рассуждения нужно проводить отдельно для каждой из частей — вещественной и мнимой).

Так же, как и асимптотическое выражение для гамма-функции, получаемое потенциированием ряда Стирлинга (см., например, [14]), асимптотическое выражение для пси-функции (см. доказательство в [15])

$$\psi(x) = \log x - \frac{1}{2x} - \frac{1}{12x^2} + \frac{1}{120x^4} - \frac{\theta}{126x^6}, \quad 0 \quad \theta = \theta(x) \quad 1,$$

не может служить основой для построения быстрого процесса.

Быстрый алгоритм вычисления ψ-функции можно построить, воспользовавшись уже существующим БВЕ-алгоритмом вычисления гамма-функции и построив БВЕ-алгоритм вычисления производной (обычной, не логарифмической) гамма-функции. Из формулы (3) легко видеть, что

$$\psi(x) = \frac{\Gamma'(x)}{\Gamma(x)}. \tag{5}$$

Таким образом, вычислив значения $\Gamma'(x)$ и $\Gamma(x)$ в точке $x = x_0$ с точностью $2^{-(n+1)}$ за $O\left(n^{1+\varepsilon}\right)$ операций, а затем разделив одно число на другое с точностью 2^{-n} методом Ньютона за

$$O\left(M(n)\log n\right) \tag{6}$$

операций, получим с учетом (1) значение пси-функции, вычисленное в точке $x = x_0$ с точностью 2^{-n} за $O\left(n^{1+\varepsilon}\right)$ операций.

В настоящее время как для гамма-функции, так и для пси-функции быстрые алгоритмы на основе метода БВЕ можно построить только для алгебраического аргумента. Как правило, задача изучения, а также уменьшения константы в знаке O в сложности вычисления (что приводит к практическому улучшению эффективности вычисления в частных случаях) редко рассматривается в теории быстрых алгоритмов (см. [16]), однако по построению алгоритма можно заметить, что алгоритм для алгебраического нерационального аргумента предполагает гораздо большую константу “в O”, чем алгоритм для рационального аргумента. Поэтому, хотя рациональные числа являются подмножеством алгебраических, отдельно строится алгоритм для рационального аргумента.

3. БВЕ алгоритмы вычисления гамма-функции

На основе применения метода БВЕ можно доказать следующие утверждения относительно сложности вычисления гамма-функции Эйлера (рассмотрим для простоты случай вещественного аргумента, для комплексного аргумента соответствующие теоремы доказываются отдельно для вещественной и мнимой частей аргумента).

Теорема 1. Пусть $y = \Gamma(x_0)$; $x_0 = p/q$; p, q — *целые*, $(p, q) = 1$. *Тогда*

$$s_{\Gamma(p/q)}(n) = O\left(M(n)\log^2 n\right). \tag{7}$$

Теорема 2. Пусть $y = \Gamma(x_0)$; $x_0 = \alpha$; α — *алгебраическое число, которое является корнем многочлена с целыми (заранее известными) коэффициентами. Тогда*

$$s_{\Gamma(\alpha)}(n) = O\left(M(n)\log^2 n\right). \tag{8}$$

Для удобства читателя напомним основные моменты доказательства теоремы 1 из [10]. Вычисляем значение $\Gamma(x_0)$, предполагая сначала, что $0 < x_0 < 1$, т.е. что $0 < p < q$. Воспользовавшись представлением гамма-функции в виде интеграла из (2) и тем, что в разложении этого интеграла на сумму двух интегралов

$$\int_0^{+\infty} e^{-t}t^{x_0-1}dt = \int_0^{a} e^{-t}t^{x_0-1}dt + \int_a^{+\infty} e^{-t}t^{x_0-1}dt,$$

для последнего интеграла при $a = n$, $0 < x_0 < 1$, справедлива оценка

$$\int_a^{+\infty} e^{-t}t^{x_0-1}dt \quad \frac{1}{a^{1-x_0}} \int_a^{+\infty} e^{-t}dt = \theta_1 2^{-n}, \quad 0 < \theta_1 < 1,$$

получаем отсюда, что

$$\Gamma(x_0) = \int_0^{n} e^{-t}t^{x_0-1}dt + \theta_1 2^{-n} = n^{x_0}S + \theta_2 2^{-n}, \quad |\theta_1| \quad 1, \quad |\theta_2| \quad 1, \tag{9}$$

где

$$S = \sum_{j=0}^{r} (-1)^j \frac{n^j}{j!(x_0 + j)}, \quad r + 1 \geqslant 4n, \quad n \geqslant 8. \tag{10}$$

Будем вычислять $\Gamma(x_0) = \Gamma(p/q)$ по формулам (9)–(10). Заметим, что для вычисления $n^{x_0} = n^{p/q}$, с точностью до n знаков достаточно $O(M(n)\log n) = O(n\log^2 n \log\log n)$ операций (методом Ньютона (см. [6]) при выполнении умножения методом Шенхаге–Штрассена (см. [4])). Чтобы вычислить S,

возьмем в (10) $r+1=2^k$, $2^{k-1}<4n \quad 2^k$, $k\geqslant 1$. Вычисляем сумму S с помощью БВЕ-процесса за k шагов, имея на j-м шаге ($j \quad k$)

$$S=S_1(j)+S_2(j)+\cdots+S_{r_j-1}(j)+S_{r_j}(j), \quad r_j=\frac{r+1}{2^j},$$

где $S_{r_j-\nu}(j)$; $\nu=0,1,\ldots,r_j-1$; определяются равенствами

$$S_{r_j-\nu}(j)=S_{r_{j-1}-2\nu}(j-1)+S_{r_{j-1}-2\nu-1}(j-1)=\frac{n^{r-(2^j\nu+2^j-1)}}{(r-2^j\nu)!}\frac{p_{r_j-\nu}(j)}{q_{r_j-\nu}(j)}.$$

При этом на 1-м шаге ($j=1$) вычисляются целые числа

$$p_{r_1-\nu}(1)=-qn(qr-2q\nu-q+p)+(qr-2q\nu)(qr-2q\nu+p);$$
$$q_{r_1-\nu}(1)=(qr-2q\nu+p)(qr-2q\nu-q+p);$$
$$\nu=0,1,\ldots,\frac{r+1}{2}-1; \quad r+1=2^k, \quad k\geqslant 1;$$

а на j-м шаге ($1<j \quad k$) вычисляются целые числа

$$p_{r_j-\nu}(j)=n^{2^{j-1}}p_{r_{j-1}-2\nu}(j-1)q_{r_{j-1}-2\nu-1}(j-1)+$$
$$+\frac{(r-2^j\nu)!}{(r-2^{j-1}(2\nu+1))!}p_{r_{j-1}-2\nu-1}(j-1)q_{r_{j-1}-2\nu}(j-1);$$
$$q_{r_j-\nu}(j)=q_{r_{j-1}-2\nu}(j-1)q_{r_{j-1}-2\nu-1}(j-1);$$
$$\nu=0,1,\ldots,r_j-1; \quad r_j=(r+1)/2^j; \quad r+1=2^k, \quad k\geqslant 1.$$

На k-м (последнем) шаге вычисляются целые числа $p_{r_k}(k)=p_1(k)$, $q_{r_k}(k)=q_1(k)$, $r!$ и производится одно деление с точностью до 2^{-2n} знаков по формуле

$$S=S_1(k)=\frac{p_{r_k}(k)}{q_{r_k}(k)}\frac{1}{r!},$$

что дает сумму S с точностью 2^{-n-1}, а следовательно и значение $\Gamma(x_0)=\Gamma(p/q)$ с точностью 2^{-n+1}, $n\geqslant 1$. Сложность такого вычисления есть (см. [10])

$$s_{\Gamma(p/q)}(n)=O\left(M(n)\log^2 n\right). \tag{11}$$

Представим доказательство теоремы 2 и алгоритм быстрого вычисления гамма-функции Эйлера при алгебраическом аргументе α в виде, удобном для его применении для быстрого вычисления пси-функции.

Доказательство. Как и для случая рационального аргумента, в соответствии с замечанием 1 сведем вычисление $\Gamma(x_0)$ при алгебраическом x_0 к вычислению $\Gamma(\alpha)$, $0<\alpha<1$, где α есть алгебраическое число степени N, $N\geqslant 2$. Для простоты рассмотрим случай, когда α является вещественным

числом. Предполагается, что известен многочлен $g(x)$, наименьшей степени с целыми коэффициентами, корнем которого является число α, т.е.

$$g(x) = g_N x^N + g_{N-1}x^{N-1} + \cdots + g_1 x + g_0; \quad g(\alpha) = 0; \tag{12}$$

$g_N, g_{N-1}, \ldots, g_0$ — целые числа, $N \geqslant 2$. Как и в случае рационального аргумента, воспользуемся формулами (9), (10). Отметим, что вычисление n^α с точностью 2^{-2n} по формуле $n^\alpha = e^{\alpha \log n}$ требует $O(M(n)\log^2 n)$ операций. Сумма S из (10) принимает вид

$$S = \sum_{j=0}^{r} (-1)^j \frac{n^j}{j!(j+\alpha)}, \tag{13}$$

и

$$S = S_1(0) + S_2(0) + \cdots + S_{r+1}(0), \tag{14}$$

где

$$S_{r+1-\nu}(0) = (-1)^{r-\nu} \frac{n^{r-\nu}}{(r-\nu)!(r-\nu+\alpha)}; \quad \nu = 0, 1, \ldots, r. \tag{15}$$

Вычисление S выполняется за k шагов, $r+1 = 2^k$; $2^{k-1} < 4n \quad 2^k$; $k \geqslant 1$, БВЕ процесса для "алгебраического случая", особенностью которого является использование основного свойства алгебраических чисел для ограничения роста сложности вычисления.

До шага i, где i определяется условиями $1 \quad i \quad k$; $2^{i-1} < N \quad 2^i$, производим с суммой (14), (15) действия, аналогичные описанным выше для случая рационального аргумента. Однако, группируя слагаемые суммы (14) таким же образом, теперь вычисляем на каждом шаге не числитель и знаменатель дробей, находящихся в скобках, а лишь целые коэффициенты при степенях α, многочленов, находящихся в числителе и знаменателе дробей "из скобок".

На 1-м шаге в скобках находятся дроби

$$\beta_{r_1-\nu}(1) = \frac{\alpha(r-2\nu-n) + (r-2\nu)^2 - n(r-2\nu-1)}{\alpha^2 + \alpha(2r-4\nu-1) + (r-2\nu)(r-2\nu-1)},$$
$$\nu = 0, 1, 2, \ldots; \frac{r+1}{2} - 1;$$

вычисляются числа

$$\delta_{r_1-\nu}(0,1) = (r-2\nu)^2 - n(r-2\nu-1); \quad \delta_{r_1-\nu}(1,1) = r-2\nu-n;$$
$$\rho_{r_1-\nu}(0,1) = (r-2\nu)(r-2\nu-1); \quad \rho_{r_1-\nu}(1,1) = 2r-4\nu-1.$$

На i-м шаге $(1 < i \quad k)$ в скобках находятся дроби

$$\beta_{r_i-\nu}(i) = \frac{\delta_{r_i-\nu}(i)}{\rho_{r_i-\nu}(i)}, \tag{16}$$

где

$$\delta_{r_i-\nu}(i)=\sum_{m=0}^{2^i-1}\delta_{r_i-\nu}(m,i)\alpha^m;\quad \rho_{r_i-\nu}(i)=\sum_{l=0}^{2^i}\rho_{r_i-\nu}(l,i)\alpha^l; \tag{17}$$

вычисляются числа

$$\delta_{r_i-\nu}(m,i)=\sum_{\substack{m_1+m_2=m\\ 0\leqslant m_1\leqslant 2^{i-1}-1\\ 0\leqslant m_2\leqslant 2^{i-1}}}\Big(\rho^{2^{i-1}}\delta_{r_{i-1}-2\nu}(m_1,i-1)\rho_{r_{i-1}-2\nu-1}(m_2,i-1)+$$

$$+\frac{(r-2^i\nu)!}{(r-2^i\nu-2^{i-1})}\delta_{r_{i-1}-2\nu-1}(m_1,i-1)\rho_{r_{i-1}-2\nu}(m_2,i-1)\Big);$$

$$\rho_{r_i-\nu}(l,i)=\sum_{\substack{l_1+l_2=l\\ 0\leqslant l_1\leqslant 2^{i-1}\\ 0\leqslant l_2\leqslant 2^{i-1}}}\rho_{r_{i-1}-2\nu}(l_1,i-1)\rho_{r_{i-1}-2\nu-1}(l_2,i-1);$$

$$m=0,1,2,\ldots,2^i-1,\quad l=0,1,2,\ldots,2^i,\quad \nu=0,1,2,\ldots,\frac{r+1}{2^i}-1.$$

Заметим, что умножение на степень α и вычисление δ и ρ, так же как и деление δ на ρ, не производится до последнего шага. Перед $i+1$-м шагом ($1\quad i\quad k$, $2^{i-1}<N\quad 2^i$) многочлены (17) редуцируются по модулю многочлена $g(x)$ из (12) при $x=\alpha$. Таким образом, если

$$\delta_{r_i-\nu}(i)=P(x)=p_{N-1}x^{N-1}+p_{N-2}x^{N-2}+\cdots+p_1x+p_0,$$
$$\rho_{r_i-\nu}(i)=Q(x)=q_Nx^N+q_{N-1}x^{N-1}+\cdots+q_1x+q_0,$$

где $p_{N-1},\ldots,p_0;q_N,\ldots,q_0$ — целые, $N=2^i$, то мы делим $P(x)$ и $Q(x)$ на $g(x)$ с остатками:

$$P(x)=g(x)P_0(x)+P_1(x),\quad Q(x)=g(x)Q_0(x)+Q_1(x),$$

где $P_1(x)$ и $Q_1(x)$ есть многочлены с рациональными коэффициентами, и при этом степени многочленов $P_1(x)$ и $Q_1(x)$ не превышают $N-1$. Отсюда и из (12) получаем

$$P(\alpha)=P_1(\alpha),\quad Q(\alpha)=Q_1(\alpha),$$

т.е. в (16) имеем

$$\beta_{r_i-\nu}(i)=\frac{P_1(\alpha)}{Q_1(\alpha)}. \tag{18}$$

Домножая, если нужно, числитель и знаменатель дроби (18) на целый общий множитель, получаем в числителе и знаменателе дробей “из скобок” многочлены с целыми коэффициентами степени не большей, чем $N-1$.

Начиная с шага i, на каждом шаге $i, i+1, \ldots, k$, производится редукция многочленов от α, расположенных в числителе и знаменателе $\beta_\mu(j)$, по модулю $g(x)$. Поскольку коэффициенты $g(x)$, так же как и степень $g(x)$ являются абсолютными постоянными, то значения этих коэффициентов, участвующие в представленных выше редукциях, могут возрасти лишь не более чем в g^N раз, где

$$g = \max_{0 \ i \ N} |g_i|,$$

что не влияет на оценку сложности проводимых вычислений. На последнем k-м шаге такого процесса получаем

$$S = S_1(k) = \frac{\beta_{r_k}(k)}{r!} = \frac{\delta_{r_k}(k)}{r!\rho_{r_k}(k)},$$

где

$$\delta_{r_k}(k) = \delta_{r_k}(0,k) + \delta_{r_k}(1,k)\alpha + \cdots + \delta_{r_k}(j,k)\alpha^j,$$
$$\rho_{r_k}(k) = \rho_{r_k}(0,k) + \rho_{r_k}(1,k)\alpha + \cdots + \rho_{r_k}(\hat{j},k)\alpha^{\hat{j}};$$

$j, \hat{j}$ $N-1$, и вычисляем $\delta_{r_k}(k)$, $\rho_{r_k}(k)$ и S, и следовательно, $\Gamma(\alpha)$ с точностью 2^{-n+1}.

Следовательно, сложность вычисления

$$s_{\Gamma(\alpha)}(n) = O\left(M(n)\log^2 n\right).$$

Что и требовалось доказать. Из (1) и оценок (7), (8) следует, что

$$s_{\Gamma(x_0)}(n) = O\left(n^{1+\varepsilon}\right),$$

для любого $\varepsilon > 0$ и $n > n_1(\varepsilon)$, и любого алгебраического аргумента x_0.

Замечание 1. Мы вычислили $\Gamma(x_0)$, предполагая, что $0 < x_0 < 1$. Если $x_0 \geqslant 1$, то пользуясь соотношением $\Gamma(x+m) = x(x+1)(x+2)\ldots$ $\ldots(x+m-1)\Gamma(x)$, сводим вычисление в нужной точке к вычислению в точке из интервала $(0,1)$ и вычислению фиксированного (константа) числа перемножений, что, хотя и увеличивает константу в O из оценок (7), (8), сами оценки не меняет.

4. БВЕ-алгоритмы вычисления производной гамма-функции

При вещественном x, $x > 0$, дифференцируя (2) по параметру x, получаем

$$\Gamma'(x) = \int_0^{+\infty} t^{x-1}e^{-t}\log t dt. \tag{19}$$

Далее считаем, что $0 < x < 1$. Представим интеграл (19) в виде суммы двух интегралов

$$\Gamma'(x) = \int_0^a t^{x-1}e^{-t}\log t\,dt + \int_a^{+\infty} t^{x-1}e^{-t}\log t\,dt, \quad a \geqslant 1. \tag{20}$$

Оценим последний интеграл в (20). Так как при $t \geqslant 1$, $t > \log t$, то

$$\int_a^{+\infty} t^{x-1}e^{-t}\log t\,dt < \int_a^{+\infty} t^x e^{-t}dt \quad \int_a^{+\infty} te^{-t}dt \quad (a+1)e^{-a}. \tag{21}$$

Рассмотрим теперь первый интеграл правой части формулы (20):

$$I = \int_0^a t^{x-1}e^{-t}\log t\,dt. \tag{22}$$

Разложим функцию e^{-t} в знакопеременный ряд Тейлора по степеням t. Для $0 \quad t \quad a$, $m \geqslant a$,

$$e^{-t} = 1 - \frac{t}{1!} + \frac{t^2}{2!} - \frac{t^3}{3!} + \cdots + \frac{t^{2m}}{(2m)!} + \theta\frac{t^{2m}}{(2m)!}, \quad |\theta| \quad 1. \tag{23}$$

Подставляя разложение (23) в (22), находим

$$I = I_1 + R(a), \tag{24}$$

где

$$I_1 = \sum_{j=0}^{2m} \frac{(-1)^j}{j!} \int_0^a t^{j+x-1}\log t\,dt, \tag{25}$$

$$R(a) = \theta \int_0^a \frac{t^{2m+x-1}}{(2m)!}\log t\,dt, \quad |\theta| \quad 1. \tag{26}$$

Оценим остаточный член $R(a)$. Имеем из (26)

$$|R(a)| \quad \int_0^a \frac{t^{2m+x-1}|\log t|}{(2m)!}dt = \tag{27}$$

$$= -\int_0^1 \frac{t^{2m+x-1}\log t}{(2m)!}dt + \int_1^a \frac{t^{2m+x-1}\log t}{(2m)!}dt \quad \frac{a^{2m+x}\log a}{(2m)!(2m+x)}.$$

Рассмотрим интеграл I_1 из (25). Интегрируя $\int_0^a t^{j+x-1}\log t dt$ по частям, получаем

$$I_1 = \log a \sum_{j=0}^{2m} \frac{(-1)^j}{j!}\frac{a^{j+x}}{j+x} - \sum_{j=0}^{2m}\frac{(-1)^j}{j!}\frac{a^{j+x}}{(j+x)^2} = a^x \log a G_1 - a^x G_2, \tag{28}$$

где

$$G_1 = \sum_{j=0}^{2m}\frac{(-1)^j}{j!}\frac{a^j}{j+x}, \tag{29}$$

$$G_2 = \sum_{j=0}^{2m}\frac{(-1)^j}{j!}\frac{a^j}{(j+x)^2}. \tag{30}$$

Из (20), (21), (24), (27)–(30) находим, что

$$\Gamma'(x) = a^x \log a G_1 - a^x G_2 + O\left(\frac{a^{2m+x}\log a}{(2m)!(2m+x)}\right) + O\left((a+1)e^{-a}\right), \tag{31}$$

откуда при $m = 2n$, $a = n$, $n \geqslant 8$, имеем следующее приближение для производной гамма-функции в точке $x = x_0$:

$$\Gamma'(x_0) = n^{x_0}\log n \sum_{j=0}^{4n}(-1)^j \frac{n^j}{j!(j+x_0)} - \tag{32}$$

$$- n^{x_0}\sum_{j=0}^{4n}(-1)^j\frac{n^j}{j!(j+x_0)^2} + \theta_0 2^{-n}, \quad |\theta_0| \quad 1.$$

Легко видеть из (13), (32), что сумма (29) совпадает с суммой (13) и поэтому вычисляется быстро с помощью ББЕ процесса для любого алгебраического аргумента в соответствии с теоремами 1 и 2. Заметим, что как и ранее, быстрое вычисление функций $\log n$ и $n^{x_0} = e^{x_0 \log n}$ требует не более $O(M(n)\log^2 n)$ операций, что не ухудшает общую оценку сложности вычисления первого слагаемого в (32), которая совпадает с оценкой (8).

Вычислим сумму (30), сначала предполагая, что $x_0 = p/q$, $(p,q) = 1$. Запишем эту сумму в виде ($r = 2m \geqslant 4n$)

$$S = \sum_{j=0}^{r}(-1)^j\frac{n^j}{j!(j+p/q)^2}. \tag{33}$$

Возьмем

$$r+1 = 2^k, 2^{k-1} < 4n \quad 2^k, k \geqslant 1; \tag{34}$$

членов ряда, из которого выделена сумма S. Пусть

$$S_{r+1-\nu}(0) = (-1)^{r-\nu}\frac{n^{r-\nu}}{(r-\nu)!(r-\nu+p/q)^2}; \quad \nu = 0,1,2,\ldots,r.$$

Тогда (33) можно записать в виде

$$S = S_1(0) + S_2(0) + \cdots + S_{r+1}(0).$$

Вычислим сумму S с помощью БВЕ-процесса за k шагов, объединяя на каждом шаге слагаемые S последовательно попарно и вынося за скобки "очевидный" общий множитель. При этом на каждом шаге будут вычисляться целый числитель и целый знаменатель дроби, стоящей в скобках (деление не производится до последнего шага).

На 1-м шаге имеем

$$S = S_1(1) + S_2(1) + \cdots + S_{r_1}(1); \quad r_1 = \frac{r+1}{2},$$

$$S_{r_1-\nu}(1) = S_{r+1-2\nu}(0) + S_{r-2\nu}(0) = (-1)^{r-1}\frac{n^{r-2\nu-1}}{(r-2\nu)!}\frac{p_{r_1-\nu}(1)}{q_{r_1-\nu}(1)} = \\ = \frac{n^{r-2\nu-1}}{(r-2\nu)!}\frac{p_{r_1-\nu}(1)}{q_{r_1-\nu}(1)}.$$

На 1-м шаге вычисляем целые числа

$$p_{r_1-\nu}(1) = q^2\left(-n(qr-2q\nu-q+p)^2 + (r-2\nu)(qr-2q\nu+p)^2\right), \tag{35}$$

$$q_{r_1-\nu}(1) = (qr-2q\nu+p)^2(qr-2q\nu-q+p)^2, \tag{36}$$

$\nu = 0, 1, \ldots, \dfrac{r+1}{2} - 1;\ r+1 = 2^k,\ k \geqslant 1.$

На j-м шаге ($j \quad k$) имеем

$$S = S_1(j) + S_2(j) + \cdots + S_{r_j-1}(j) + S_{r_j}(j); \quad r_j = \frac{r+1}{2^j};$$

где $S_{r_j-\nu}(j);\ \nu = 0, 1, \ldots, r_j - 1$; определяются равенствами

$$S_{r_j-\nu}(j) = S_{r_{j-1}-2\nu}(j-1) + S_{r_{j-1}-2\nu-1}(j-1) = \frac{n^{r-(2^j\nu+2^j-1)}}{(r-2^j\nu)!}\frac{p_{r_j-\nu}(j)}{q_{r_j-\nu}(j)}.$$

На j-м шаге ($j \quad k$) вычисляем целые числа

$$p_{r_j-\nu}(j) = n^{2^{j-1}}p_{r_{j-1}-2\nu}(j-1)q_{r_{j-1}-2\nu-1}(j-1) + \\ + \frac{(r-2^j\nu)!}{(r-2^{j-1}(2\nu+1))!}p_{r_{j-1}-2\nu-1}(j-1)q_{r_{j-1}-2\nu}(j-1), \tag{37}$$

$$q_{r_j-\nu}(j) = q_{r_{j-1}-2\nu}(j-1)q_{r_{j-1}-2\nu-1}(j-1), \tag{38}$$

$\nu = 0, 1, \ldots, r_j - 1;\ r_j = (r+1)/2^j$. И так далее.

На k-м (последнем) шаге вычисляем целые числа $p_{r_k}(k) = p_1(k)$, $q_{r_k}(k) = q_1(k)$, $r!$ и производим одно деление с точностью до 2^{-2n} знаков по формуле

$$S = S_1(k) = \frac{p_{r_k}(k)}{q_{r_k}(k)} \frac{1}{r!},$$

что дает сумму S с точностью 2^{-n-1}. Следовательно, значение $\Gamma'(x_0) = \Gamma'(p/q)$ вычислено с точностью 2^{-n+1}, $n \geqslant 1$.

При алгебраическом $x_0 = \alpha$ первая сумма из (32) совпадает с суммой (13) и вычисляется так же. Вычисление суммы

$$S = \sum_{j=0}^{r} (-1)^j \frac{n^j}{j!(j+\alpha)^2} \tag{39}$$

проводится аналогично.

5. Сложность вычисления пси-функции

Оценим сложность вычисления $\Gamma'(x_0)$, если $x_0 = p/q$. Подсчитаем сложность вычисления суммы (33). Сначала, пользуясь формулами (37), (38), подсчитаем количество операций, достаточное для вычисления на $j+1$-м шаге, $j+1$ k, чисел $p_{r_{j+1}-\nu}(j+1)$, $q_{r_{j+1}-\nu}(j+1)$; $\nu = 0, 1, 2, \ldots, r_{j+1} - 1$, $r_{j+1} = (r+1)/2^{j+1}$, предполагая при этом, что числа $p_{\mu_1}(j)$, $q_{\mu_2}(j)$; $\mu_1 = 1, 2, \ldots, k-1$, $\mu_2 = 1, 2, \ldots, k-1$; уже вычислены. Для этого, прежде всего, оценим сверху разрядность чисел, с которыми проводятся вычисления на $j+1$-м шаге. Пусть

$$p(j) = \max_{\mu_1} p_{\mu_1}(j), \quad q(j) = \max_{\mu_2} q_{\mu_2}(j).$$

Поскольку

$$\frac{(r - 2^{j+1}\nu)!}{(r - 2^{j+1}\nu - 2^j)!} \quad r^{2^j}$$

из (37), (38) находим

$$p(j+1) \quad p(j)q(j)(n^{2^j} + r^{2^j}) \quad p(j)q(j)(nr)^{2^j}; \quad q(j+1) \quad (q(j))^2.$$

Отсюда и из (36), (38) получаем

$$\frac{p(j+1)}{p(1)} \quad q(j)\, q(j-1) \ldots q(1)\, (nr)^{2^j + 2^{j-1} + \cdots + 2}$$

$$(q(1))^{2^{j-1} + 2^{j-2} + \cdots + 1}\, (nr)^{2^{j+1} - 2}.$$

Следовательно,

$$p(j+1) \quad p(1)(q(1))^{2^j - 1}(nr)^{2^{j+1}-2}, \quad q(j+1) \quad (q(1))^{2^j}. \tag{40}$$

Учитывая, что из (35), (36)

$$p(1) \quad q^4r^3p^2, \quad q(1) \quad q^4r^4p^4,$$

из (40) имеем

$$p(j+1) \quad (nqp)^{2^{j+2}} r^{3(2^{j+1}-1)} \quad (qprn)^{2^{j+3}}, \quad q(j+1) \quad (qpr)^{2^{j+2}}. \tag{41}$$

Рассмотрим формулу (37). Сложность вычисления значений произведений

$$\frac{(r-2^{j+1}\nu)!}{(r-2^{j+1}\nu-2^j)!}; \quad \nu = 0,1,2,\dots,r_{j+1}-1, \quad r_{j+1} = \frac{r+1}{2^{j+1}},$$

составляет

$$O\left(\sum_{\tau=1}^{j} M(2^\tau \log r)\right) \tag{42}$$

операций. Из (41), (42) получаем, что для вычисления $p_{r_{j+1}-\nu}(j+1)$ достаточно $O(B(j+1))$ операций, где

$$B(j+1) = \sum_{\tau=1}^{j} M(2^\tau \log r) + M\left(2^{j+3} \log qprn\right)$$

Чтобы вычислить все $\alpha_{r_{j+1}-\nu}(j+1)$, которых ровно $r_{j+1} = 2^{-(j+1)}(r+1)$, достаточно $O(r_{j+1}B(j+1))$ операций. Предполагая для сложности умножения $M(\mu)$ оценку Шенхаге–Штрассена (см. [4]), получаем, что для вычисления $\alpha_1(k)$ на последнем шаге достаточно

$$O\left(\sum_{j=1}^{k-1} r_{j+1}B(j+1)\right) =$$

$$= O\left(\sum_{j=1}^{k-1} (r+1)2^{-(j+1)} \sum_{\tau=1}^{j} 2^\tau \log r(\tau + \log\log r)\log(\tau+\log\log r) + \right.$$

$$\left. + \sum_{j=1}^{k-1} (r+1)2^{-(j+1)} 2^{j+3} \log qprn(j+3+\log\log qprn)\log(j+3+\log\log qprn)\right) =$$

$$= O\left(\sum_{j=1}^{k-1} r\log qprn(j+\log\log qprn)\log(j+\log\log qprn)\right) =$$

$$= O(r\log^2 r \log qprn \log\log qprn) \tag{43}$$

операций. Учитывая, что q и p — это фиксированные константы, а параметр r удовлетворяет условиям (34), получаем из (43) оценку сложности вычисления суммы S

$$s_S(n) = O(n\log^3 n \log\log n). \tag{44}$$

Сложность вычисления суммы (30) при алгебраическом аргументе $x = \alpha$ до момента редукции по модулю многочлена (12), скажем до шага i, оценивается так же, как и в изложенном выше случае рационального аргумента. Затем, как и ранее, на каждом шаге: $i,\ i+1,\ i+2, \ldots$; проводится редукция многочленов от α, стоящих в числителе и знаменателе дробей "в скобках" по модулю $g(x)$ при $x = \alpha$. Поскольку коэффициенты в $g(x)$ являются абсолютными постоянными, степень $g(x)$ также абсолютная постоянная, при таких редукциях значения участвующих в вычислениях коэффициентов могут увеличиваться лишь в постоянное число раз, что не меняет оценку сложности вычисления.

Тем самым, учитывая (8), (10), (32), (33), (44), для сложности вычисления производной гамма функции при любом алгебраическом аргументе справедлива оценка

$$s_{\Gamma'(\alpha)}(n) = O(n \log^3 n \log\log n) = O\left(n^{1+\varepsilon}\right). \tag{45}$$

Замечание 2. Мы вычисляли $\Gamma'(x_0)$, предполагая, что $0 < x_0 < 1$. Если $x_0 \geqslant 1$, то пользуясь соотношением

$$\Gamma'(x+m) = ((x+1)(x+2)\ldots(x+m-1) + x(x+2)\ldots(x+m-1) + \ldots + \\ + x(x+1)\ldots(x+m-2))\,\Gamma(x) + x(x+1)(x+2)\ldots(x+m-1)\Gamma'(x),$$

сводим вычисление производной гамма функции в нужной точке к вычислению в точке из интервала $(0,1)$ и вычислению фиксированного (константа) числа произведений, вычислению гамма-функции и произведению гамма функции на сумму произведений фиксированных чисел, что, хотя и увеличивает константу в O из оценки (45), саму оценку не меняет.

Таким образом, доказаны следующие утверждения

Теорема 3. Пусть $y = \Gamma'(x_0)$; $x_0 = p/q$; p, q – целые, $(p,q) = 1$. Тогда

$$s_{\Gamma'(p/q)}(n) = O\left(M(n)\log^2 n\right).$$

Теорема 4. Пусть $y = \Gamma'(x_0)$; $x_0 = \alpha$; α — алгебраическое число, которое является корнем многочлена с целыми (заранее известными) коэффициентами. Тогда

$$s_{\Gamma'(\alpha)}(n) = O\left(M(n)\log^2 n\right).$$

На основе формулы (5) и теорем 1, 2, 3, 4, а также оценки (8) получаем доказательство следующего утверждения

Теорема 5. Пусть $y = \psi(x_0)$; $x_0 = \alpha$; α — алгебраическое число. Тогда

$$s_{\psi(\alpha)}(n) = O\left(M(n)\log^2 n\right). \tag{46}$$

6. Заключение

Отметим, что аналогичные быстрые алгоритмы можно построить для логарифмических производных гамма функции любого порядка. Например, для производной пси-функции находим

$$\frac{d}{dx}\frac{\Gamma'(x)}{\Gamma(x)} = \frac{\Gamma''(x)}{\Gamma(x)} - \left(\frac{\Gamma'(x)}{\Gamma(x)}\right)^2. \tag{47}$$

Легко видеть, что для второй производной гамма функции

$$\Gamma''(x) = \int_0^{+\infty} t^{x-1}e^{-t}\log^2 t dt,$$

можно построить ББЕ-процесс, аналогичный представленному выше. Другие функции и другое слагаемое из (47) у нас уже вычислены со сложностью (46). Таким образом, последовательно вычислив все производные до порядка K и переходя к порядку $K+1$, можно вычислять логарифмические производные любого порядка со сложностью (46). Поскольку K здесь константа, общая сложность вычисления не изменится, обеспечивая тем самым справедливость утверждения

Теорема 6. Пусть $F(z_0) = \frac{d^K}{dz^K}\log\Gamma(z_0)$; $z_0 = \alpha$; α — *алгебраическое число. Тогда*

$$s_{F(\alpha)}(n) = O\left(M(n)\log^2 n\right).$$

СПИСОК ЛИТЕРАТУРЫ

1. *Dynkin E.B., Kolmogorov A.N., Kostrikin A.I., et. al.* Six Lectures Delivered at the International Congress of Mathematicians in Stockholm, 1962 (American Mathematical Society Translations–series 2). 31. Publ. by AMS, 1963.
2. *Карацуба А.А.* Сложность вычислений // Тр. МИАН. 1995. № 211. С. 186–202.
3. *Карацуба А., Офман Ю.* Умножение многозначных чисел на автоматах // Докл. Академии наук СССР. 1962. Т. 145. № 2. С. 293–294.
4. *Schönhage A., Strassen V.* Schnelle Multiplikation grosser Zahlen // Computing. 1971. No. 7. P. 281–292.
5. *Fürer M.* Faster Integer Multiplication // SIAM J. Comput. 2009. V. 39. No. 3. P. 979–1005.
6. *Бендерский Ю.В.* Быстрые вычисления // Докл. Академии наук СССР. 1975. Т. 223. № 5. С. 1041–1043.
7. *Salamin E.* Computation of π using arithmetic-geometric mean // Math. Comp. 1976. V. 30. No. 135. P. 565–570.
8. *Carlson B.C.* Algorithms involving arithmetic and geometric means // Amer. Math. Monthly. 1971. No. 78. P. 496–505.
9. *Borwein J.M., Borwein P.B.* Pi and the AGM–A Study in Analytic Number Theory and Computational Complexity. New York: Wiley, 1987.

10. *Карацуба Е.А.* Быстрые вычисления трансцендентных функций // Пробл. передачи информ. 1991. Т. 27. № 4. С. 76–99.
11. *Карацуба Е.А.* О вычислении функции Бесселя путем суммирования рядов // Сиб. журн. вычисл. мат. 2019. Т. 27. № 4. С. 76–99.
12. *Lozier D.W., Olver F.W.J.*, Numerical Evaluation of Special Functions // Mathematics of Computation 1943–1993: A Half -Century of Computational Mathematics, Gautschi W., eds., Proc. Sympos. Applied Mathematics. AMS. 1994. No. 48. P. 79–125.
13. *Siegel C.L.* Transcendental numbers. Princeton: Princeton University Press, 1949.
14. *Temme N.M.* Special Functions. An Introduction to the Classical Functions of Mathematical Physics. New York: J. Wiley and Sons, 1996.
15. *Karatsuba E.A.* On the computation of the Euler constant γ // Numerical Algorithms. 2000. No. 24. P. 83–97.
16. *Bach E.* The complexity of number-theoretic constants // Info. Proc. Letters. 1997. No. 62. P. 145–152.

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 01.02.2022

После доработки 28.04.2022

Принята к публикации 29.06.2022

**© 2022 г. А.Ю. ГОРНОВ, д-р техн. наук (gornov@icc.ru),
А.С. АНИКИН, канд.физ.-мат. наук (anikin@icc.ru),
Т.С. ЗАРОДНЮК, канд. техн. наук (tz@icc.ru),
П.С. СОРОКОВИКОВ (sorokovikov.p.s@gmail.com)
(Институт динамики систем и теории управления
им. В.М. Матросова СО РАН, Иркутск)**

МОДИФИКАЦИЯ АЛГОРИТМА ДОВЕРИТЕЛЬНОГО БРУСА, ОСНОВАННОГО НА АППРОКСИМАЦИИ ГЛАВНОЙ ДИАГОНАЛИ МАТРИЦЫ ГЕССЕ, ДЛЯ РЕШЕНИЯ ЗАДАЧ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ[1]

В работе предложен подход к исследованию стандартной задачи оптимального управления, основанный на использовании редукции к конечномерной задаче оптимизации с последующим использованием аппроксимации главной диагонали гессиана. Приведены результаты вычислительных экспериментов по решению вспомогательных задач оптимизации сепарабельных, квазисепарабельных функций и функций Розенброка–Скокова.

Ключевые слова: алгоритм доверительного бруса, квазисепарабельная функция, матрица Гессе, задача оптимального управления.

DOI: 10.31857/S0005231022100117, **EDN:** ALGZQW

1. Введение

Задачи оптимального управления (ЗОУ) возникли из стремления учесть ограничения на управляющие воздействия и фазовые координаты объектов, динамика которых описывается системами обыкновенных дифференциальных уравнений. Эта область исследований, подобно другим направлениям теории экстремальных задач, возникла в связи с запросом со стороны приложений: появление постановок задач автоматического регулирования с ограничениями на управления (А.А. Фельдбаум [15], Д.В. Бушау [3]), которые уже не укладывались в теорию вариационного исчисления. Хотя первые подобные постановки точечно возникали и ранее (например, задача Р. Годдарда (1919 г.) о подъеме ракеты на заданную высоту с минимальными затратами топлива), активное развитие этого направления традиционно связывают с появлением фундаментальных принципов теории оптимального управления: принципа максимума [14] и метода динамического программирования [7]. Сейчас теория оптимального управления активно развивается как в связи с наличием трудных и интересных математических постановок, так и с обилием приложений, в том числе и в таких областях, как космонавигация,

[1] Работа выполнена за счет субсидии Минобрнауки России в рамках проекта «Теория и методы исследования эволюционных уравнений и управляемых систем с их приложениями» (№ 121041300060-4).

робототехника, динамика полета, экономика, биология, медицина, ядерная энергетика, нанофизика и других.

Несмотря на активные исследования этой области стандартная ЗОУ с параллелепипедными ограничениями на управление по-прежнему довольно часто возникает на практике и может характеризоваться наличием особенностей, требующих использования специализированных подходов к их рассмотрению.

2. Задачи оптимального управления: стандартная постановка

Разбиение переменных на фазовые и ограниченное управление, при наличии связывающих их дифференциальных уравнений — стандартная модель для процесса, развивающегося по законам природы, но испытывающего управляющие воздействия, стремящиеся сделать его в некотором смысле оптимальным. Наличие параллелепипедных ограничений на управления возникает естественным образом в силу ограниченности имеющихся ресурсов и присутствует в стандартной постановке задачи оптимального управления. Управляемая динамическая система с начальными условиями описывается дифференциальными уравнениями в нормальной форме Коши:

$$\dot{x} = f(x(t), u(t), t), \quad x(t_0) = x^0, \quad t \in [t_0, t_1]\,, \tag{1}$$

где t — независимая переменная (время), $x(t)$ — вектор фазовых координат размерности n, $\dot{x} = dx/dt$ — производная фазовой траектории, $u(t)$ — r-вектор управляющих воздействий, x^0 — вектор начального состояния системы. Допустимые управления — это вектор-функции, определенные на временном отрезке T и удовлетворяющие ограничениям:

$$u(t) \in U = \{u(t) \in R^n : \underline{u} \quad u(t) \quad \overline{u}\}\,. \tag{2}$$

Задача состоит в поиске допустимого управления $u^*(t)$, доставляющего минимум терминальному функционалу, зависящему от траектории системы (1) в конечный момент времени t_1

$$I(u) = \varphi(x(t_1)) \to \min\,. \tag{3}$$

Допустимый процесс (u, x), $u = u(t) \in U$, на котором функционал минимален, называется оптимальным — (u^*, x^*). Вектор-функция $f(x(t), u(t), t)$ и скалярная функция $\varphi(x)$ предполагаются непрерывно дифференцируемыми по всем аргументам, кроме t. Постановки с интегральными и смешанными функционалами (задачи Больца и Лагранжа) путем введения дополнительных фазовых координат могут редуцироваться к представленной задаче Майера. Для подобных детерминированных задач, в которых уравнения движения, критерий качества и ограничения известны точно, оптимальное значение критерия качества (3), реализуемое в классе программных управлений и управлений по принципу обратной связи, является одним и тем же. ЗОУ в данной формулировке считается классической [12, 16] и часто встречается в различных приложениях.

3. Традиционные подходы к исследованию стандартной ЗОУ

В теории оптимального управления принято различать два типа численных подходов: прямые (direct) и непрямые (indirect). Прямые методы заключаются в дискретизации состояния и управления и тем самым сводят исходную задачу к задаче нелинейной оптимизации с ограничениями. Непрямые методы состоят из численного решения краевой задачи, вытекающей из применения принципа максимума Понтрягина, и приводят к методам стрельбы (или методам пристрелки — shooting methods). В связи со значительными трудностями построения аналитического решения прикладных задач оптимального управления ключевое значение приобрели различные приближенные и численные методы их исследования. В зависимости от алгоритмической основы метода он может быть отнесен к той или иной группе.

3.1. Методы, основанные на использовании принципа максимума Понтрягина

В непрямых методах вместо предварительной дискретизации, как в прямых методах, сначала применяется принцип максимума Понтрягина (ПМП) как условие первого порядка к задаче оптимального управления. Согласно этому принципу оптимальную траекторию следует искать среди соответствующих экстремалей, для которых он выполняется. Подобные подходы могут опираться на сведение исходной задачи к краевой задаче нахождения экстремального решения сопряженной системы. Решить такую нелинейную систему из n уравнений с n неизвестными на практике можно, например, с помощью методов ньютоновского типа. Таким образом, ПМП является универсальным необходимым условием оптимальности первого порядка в стандартной ЗОУ. Традиционно он наиболее эффективен в системах управления с максимальным быстродействием и минимальным расходом энергии, где применяются управления релейного типа, принимающие крайние, а не промежуточные значения на допустимом интервале управления.

Построение вычислительных схем, основанных на применении ПМП, может опираться на следующие этапы: а) формируется и решается система уравнений из условия равенства нулю градиента функции Понтрягина; б) в критических точках исследуется на знакоопределенность матрица вторых производных, в случае ее положительной определенности получается точка строгого локального минимума, отрицательно определенная матрица характеризует локальный максимум; в) анализируются критические точки, в которых матрица вторых производных не является знакоопределенной; г) найденные точки локальных экстремумов исследуются на глобальный экстремум, если это возможно.

Для линейно-выпуклых задач оптимального управления выполнение условия максимума функции Понтрягина является и достаточным условием оптимальности, и может быть использовано для нахождения глобального минимума функционала. В общем случае, когда правые части системы и подынтегральная функция критерия качества дифференцируемы по управлениям, может использоваться линеаризованный или дифференциальный принцип

максимума [8, 9], часто применяемый в вычислительных схемах для проверки оптимальности полученного в результате итерационного процесса решения. Методы последовательных приближений для поиска управлений, удовлетворяющих линеаризованному ПМ, опираются на использование информации о градиенте функции Понтрягина и применяются, фактически, для решения конечномерных задач максимизации гамильтониана в заданных точках временного отрезка. Важно помнить, что принцип максимума как необходимое условие оптимальности, вообще говоря, порождает не оптимальные траектории, а экстремали, для которых требуется отдельно обосновывать оптимальность. Для этой цели могут быть использованы достаточные условия оптимальности.

3.2. Методы, опирающиеся на дискретизацию ЗОУ

Для использования многочисленных существующих прямых методов необходимо выбирать конечномерные представления управления и состояния, а затем дискретно выражать дифференциальные уравнения, описывающие динамическую систему, критерий минимизации и присутствующие в задаче ограничения. После того, как все статические и динамические ограничения редуцированы к задаче с конечным числом переменных, необходимо решить полученную задачу оптимизации, используя какой-либо адаптированный метод.

В качестве примера можно привести самый простой способ дискретизации, основанный на равномерном разбиении временного отрезка на подынтервалы. Управления дискретизированы таким образом, чтобы являться кусочно-постоянными на каждом подынтервале и удовлетворять заданным ограничениям. Самым простым методом дискретизации обыкновенных дифференциальных уравнений для численной реализации (в том числе для организации параллельных вычислений) является стандартный явный метод Эйлера. Множество допустимых управлений можно также дискретизировать, например, кусочно-постоянными функциями или сплайнами, а обыкновенные дифференциальные уравнения аппроксимировать дискретными соотношениями с использованием методов типа Рунге–Кутты различных порядков. В результате из непрерывной задачи оптимального управления получаем задачу конечномерной минимизации, в которой переменные подчиняются ограничениям, вытекающим из дифференциальной системы и ограничений на управления.

В плане соответствия редуцированной задачи исходной можно утверждать следующее: стандартная задача оптимального управления характеризуется управляющими воздействиями кусочно-непрерывного типа, соответствующая фазовая траектория является кусочно-гладкой и при выполнении условия роста с учетом ограничений на управления можно с уверенностью считать, что соответствующая конечномерная задача, полученная путем дискретизации, будет поддаваться численному решению с использованием методов конечномерной оптимизации при небольших размерностях. Это связано с наличием дифференциальной связи фазовых координат и управляющих воздействий, которая позволяет получать редуцированные конечномерные задачи с адекватными свойствами. Непрерывная дифференцируемость правых частей системы дифференциальных уравнений по фазовым переменным и

выполнение соответствующего условия Липшица позволяют обеспечить существование и единственность решения в задаче Коши для любого допустимого управления.

4. Модификация алгоритма доверительного бруса

Предлагаемый алгоритм основан на использовании информации о главной диагонали матрицы Гессе. Потому как использование полного гессиана при реализации алгоритмов может оказаться не слишком рентабельной операцией. Помимо большого объема памяти, требуемой для размещения матрицы Гессе, вычисление информации второго порядка может оказаться достаточно трудоемкой задачей — при недоступных аналитических формулах для вторых производных. В такой ситуации обычно используются варианты разностных схем — либо второго порядка, опирающихся на алгоритм вычисления функции, либо первого порядка, основанных на вычислении вектора градиентов.

В отличие от классических методов ньютоновского типа в реализованном алгоритме используются элементы только главной диагонали матрицы Гессе и для нахождения направления движения на каждой итерации формулируется и решается задача квадратичного программирования на брусе. Решение этой вспомогательной задачи, в данном случае сепарабельной, тривиальное и получается в замкнутом виде, что позволяет достичь как хорошей скорости, так и достаточно высокой точности. Заметим, что от классических методов доверительного интервала (см., например, [6]) реализованный подход отличается способом ограничения вариации на каждой итерации: вместо эллипсоидального ограничения в данном случае применяется брусовое. Это влечет за собой другую постановку вспомогательной задачи и, очевидно, другие свойства общего алгоритма.

Реализованные варианты алгоритма используют аппроксимацию диагонали с помощью двух градиентов ($Var1$, [11]) и классическую аппроксимацию по разностной схеме второго порядка ($Var2$, см., например, [13]). Помимо оптимизации памяти такой подход существенно ускоряет решение внутренней задачи линейной алгебры, которая в таком случае становится тривиальной. Для обеспечения способности алгоритма генерировать улучшающие итерации с любой начальной точки, в конструкции используется техника криволинейного шага поиска [11]. Алгоритм позволяет быстро находить решение в классе квазисепарабельных функций, особенно на функциях с диагональным преобладанием в матрице Гессе [1].

Для вспомогательной процедуры одномерного поиска в зависимости от трудоемкости обработки матрицы Гессе может использоваться как грубый алгоритм одномерного поиска, начинающий итерации с единичного шага, и дробящий шаг до достижения релаксирующего приближения, так и специализированный высокоточный алгоритм локального одномерного поиска (в разработанном подходе — комбинация методов золотого сечения и обратной параболической интерполяции (см., например, [2, 10]). В обоих реализованных вариантах алгоритма вспомогательные поисковые методы опираются на использование квадратичной вариационной конструкции, приводящей к аппроксимации «ньютоновской» точки при единичном шаге и приближающейся к направлению антиградиента при малых шагах (см. алгоритм 1).

Алгоритм 1. Алгоритм доверительного бруса

Require: Задаются алгоритмические параметры, выбираются $\mathbf{x}^0$, $\delta^K = \delta_{\text{start}}$.

for $K = 0, \ldots, T_{\text{out}}$ **do**
  if $\left\|\nabla f(\mathbf{x}^K)\right\| <= \varepsilon_{ng}$ **then**
    $\mathbf{x}^* = \mathbf{x}^K$
    brake
  end if
  if $Var1$ **then**
    Вычисляется аппроксимация Z^K диагонали гессиана
    с помощью алгоритма, основанного на градиентах
    с шагом сдвига α_D
  end if
  if $Var2$ **then**
    Вычисляется аппроксимация Z^K диагонали гессиана
    с использованием алгоритма, основанного на разностной схеме
    с шагом $Stdif$
  end if
  $B^K = \left[\mathbf{x}^K - \delta^K, \mathbf{x}^K + \delta^K\right[$
  $p^k = \arg\min pZ^K p - \nabla f(\mathbf{x}^K)p$, $p \in B^K$
  $\mathbf{x}(\alpha) = \mathbf{x}^K + \alpha^2 p^K - \alpha(1-\alpha)\nabla f(\mathbf{x}^K)$.
  $\alpha^K = \arg\min_{\alpha\in[0,1]} f(\mathbf{x}(\alpha))$.
  $\mathbf{x}^{K+1} = \mathbf{x}^K + \alpha^{2K} p^K - \alpha^K(1-\alpha^K)\nabla f(\mathbf{x}^K)$.
  if $\alpha^K < 0{,}5$ **then**
    $\delta^K = 0{,}9\delta^K$
    if $\delta^K < \delta_{\min}$ **then**
      $\delta^K = \delta_{\text{start}}$
    end if
  end if
  if $\alpha^K > 0{,}5$ **then**
    $\delta^K = 1{,}1\delta^K$
    if $\delta^K > \delta_{\max}$ **then**
      $\delta^K = \delta_{\text{start}}$
    end if
  end if
end for

Выход: $\mathbf{x}^*$ если было достигнуто условие досрочного завершения, иначе $x_{T_{\text{out}}+1}$.

Влиять на вычислительные свойства программной реализации предложенного алгоритма можно путем изменения значений алгоритмических параметров. К общим алгоритмическим параметрам для двух реализованных вариантов относятся: ε_{ng} — точность критерия остановки по норме градиента — $\left[10^{-12}, 100\right]$, в качестве стандартного значения выбирается 10^{-5}; α_0 — начальный шаг одномерного поиска, принимающий на старте наибольшее значение из интервала $\left[10^{-10}, 1\right]$; tol_n — точность одномер-

ного поиска выбирается из интервала $\left[10^{-12},\ 0{,}1\right]$ и задается равной 10^{-4}; ε_{fi} — точность критерия остановки по релаксации $\in \left[10^{-15}, 100\right]$ и выбирается по умолчанию равной 10^{-6}; δ_{start} — начальный размер доверительных брусов — 0,1, — из интервала $\left[10^{-6}, 10\right]$; $\delta_{\min}$ — минимальный размер доверительных брусов — 10^{-4}, $\left[10^{-12}, 1\right]$; $\delta_{\max}$ — максимальный размер доверительных брусов выбирается из интервала $\left[10^{-6}, 10^{3}\right]$ равным 10.

К специализированным алгоритмическим параметрам для первого варианта относится шаг сдвига от точки первого градиента до второго $\alpha_d = 0{,}1 \in \in \left[10^{-6}, 1\right]$. Для второго варианта алгоритма – шаг численного дифференцирования $Stdif$, принимающий значение из интервала $\left[10^{-12}, 1\right]$, стандартное значение равно 10^{-10}. Настройка значений алгоритмических параметров может повысить эффективность программной реализации алгоритма для выбранного класса задач.

5. Вычислительные эксперименты

Расчеты проводились на персональном компьютере, тактовая частота процессора 2.8GHz, Intel Core i7. Использован компилятор BCC 5.5 под управлением виртуальной машины MaC OS, Windows XP. Для исследования свойств реализованного алгоритма сформирована небольшая коллекция тестовых задач, включающая: тестовые примеры сепарабельных и квазисепарабельных функций, а также функции Розенброка–Скокова [5], ориентированные на сравнение и исследование свойств программных реализаций предложенного алгоритма.

5.1. Результаты решения вспомогательной задачи конечномерной оптимизации

5.1.1. Сепарабельная функция. Наиболее простая по структуре функция — это первая тестовая функция, на которой можно проверить работоспособность алгоритмов оптимизации и сравнить их свойства (рис. 1). Размерность данной задачи легко изменяется (вычислительные эксперименты проводились в том числе для Large-size problem (табл. 1).

$$f(x) = \sum_{i=1}^{n} \left[x_i^2 \to \min, \quad x_i^0 = 0{,}5 + i \cdot 10^{-6}, \quad i = \overline{1, n}.\right.$$

5.1.2. Квазисепарабельная функция. Сложность квазисепарабельных функций при изменении значения множителя, входящего в состав второго слагаемого, может возрастать. Функции данного типа также являются популярными для проведения вычислительных экспериментов по исследованию алгоритмов оптимизации (см., например, [4]).

$$f(x) = \sum_{i=1}^{n} \left[x_i^2 + 0{,}001 \sum_{i=1}^{n-1} \right.\left[(x_i - x_{i+1})^2 \to \min, \quad x_i^0 = 1{,}0 + i \cdot 10^{-7}, \quad i = \overline{1, n}.\right.$$

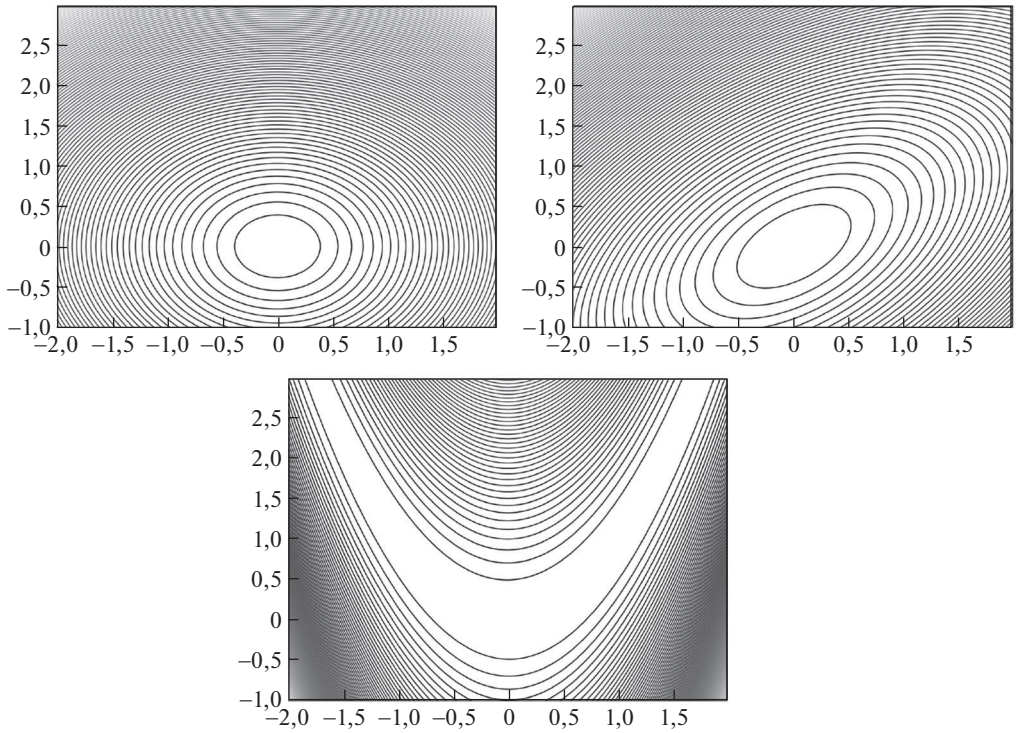


Рис. 1. Линии уровня тестовых сепарабельной, квазисепарабельной функций и функции Розенброка–Скокова.

5.1.3. Функция Розенброка–Скокова.

$$f(x) = (1 - x_1)^2 + 100 \sum_{i=2}^{n} (x_i - x_{i-1}^2)^2 \to \min, \quad x_i^0 = -2{,}0 + i \cdot 10^{-7}, \quad i = \overline{1, n}.$$

Сравнение работоспособности предложенного алгоритма с авторской библиотекой алгоритмов оптимизации, включающей программные реализации методов Ньютона, Барзилаи–Борвейна, Поляка, Бройдена–Флетчера–

Таблица 1. Результаты вычислительных экспериментов для семейства сепарабельных функций. Здесь $Var1$ — результаты расчетов для первого варианта алгоритма, $Var2$ — для второго варианта алгоритма, f_{rec} — наилучшее достигнутое значение функции, ng — достигнутое значение нормы градиента, $iter$ — число итераций алгоритма

n	5	50	$5 \cdot 10^2$	$5 \cdot 10^3$	$5 \cdot 10^4$	$5 \cdot 10^5$
$Var1$ f_{rec}	0,00	0,00	0,00	0,00	0,00	0,00
$Var1$ $iter$	8	8	8	8	8	8
$Var1$ ng	0,0	0,0	0,0	0,0	0,0	0,0
$Var2$ f_{rec}	$1{,}34 \cdot 10^{-15}$	$1{,}33 \cdot 10^{-11}$	$9{,}94 \cdot 10^{-31}$	$2{,}64 \cdot 10^{-15}$	—	—
$Var2$ $iter$	8	8	9	9	—	—
$Var2$ ng	$7{,}3 \cdot 10^{-8}$	$7{,}0 \cdot 10^{-6}$	$3{,}0 \cdot 10^{-15}$	$2{,}0 \cdot 10^{-7}$	—	—

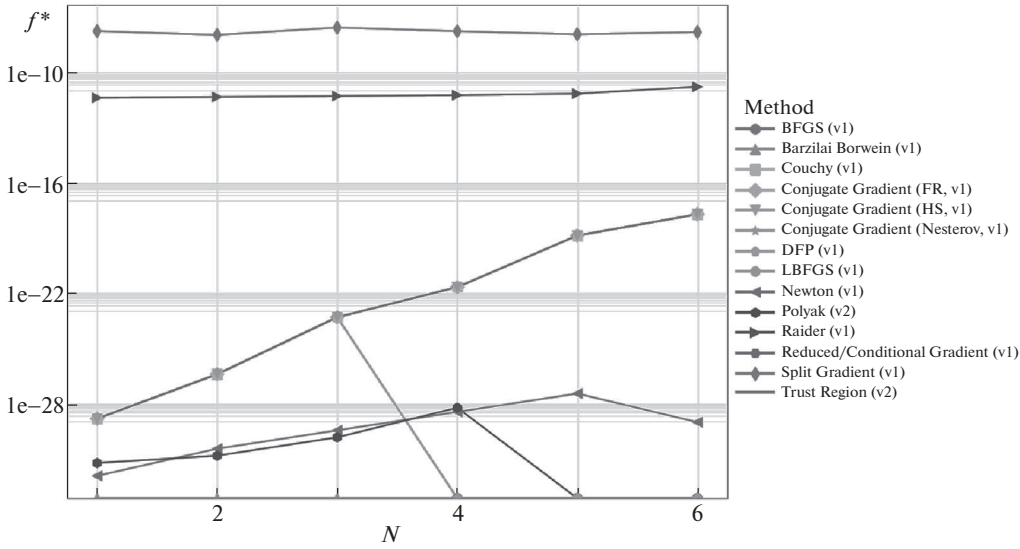


Рис. 2. Результаты вычислительных экспериментов для библиотеки методов оптимизации на сепарабельной функции возрастающих размерностей.

Гольдфарба–Шанно и других, отображено на рис. 2. В данном случае тестирование проводилось с использованием сепарабельной функции размерности от 5 до $5 \cdot 10^5$ переменных (табл. 1). Видно, что с использованием предложенной модификации метода доверительных брусов удается решить задачу для всех рассматриваемых размерностей сепарабельной функции.

5.2. Результаты решения задачи оптимального управления

Приведем пример модельной тестовой задачи оптимального управления с невыпуклым множеством достижимости, характеризующимся малой областью притяжения глобального экстремума.

$$\dot{x}_1(t) = u(t) - \sin\sqrt{|x_1(t)|}, \quad \dot{x}_2(t) = u(t) + \cos\sqrt{|x_1(t)|}, \tag{4}$$

$$x_1(t_0) = 1, \quad x_2(t_0) = 1, \quad u(t) \in [-0{,}45, 0{,}45]\,, \quad t \in T = [0, 27]\,, \tag{5}$$

$$I(u) = (x_1(t_1) + 3)^2 + (x_2(t_1) + 0{,}5)^2 \downarrow . \tag{6}$$

Данная тестовая задача позволяет смоделировть вычислительные трудности, характерные для рассматриваемых задач оптимального управления с нелинейными системами дифференциальных уравнений и невыпуклыми функционалами. На рис. 3 представлено множество достижимости в приведенной тестовой задаче с выделенной экстремальной точкой, в которой достигается наименьшее значение целевого функционала на соответствующих оптимальных траекториях и управлении (рис. 3).

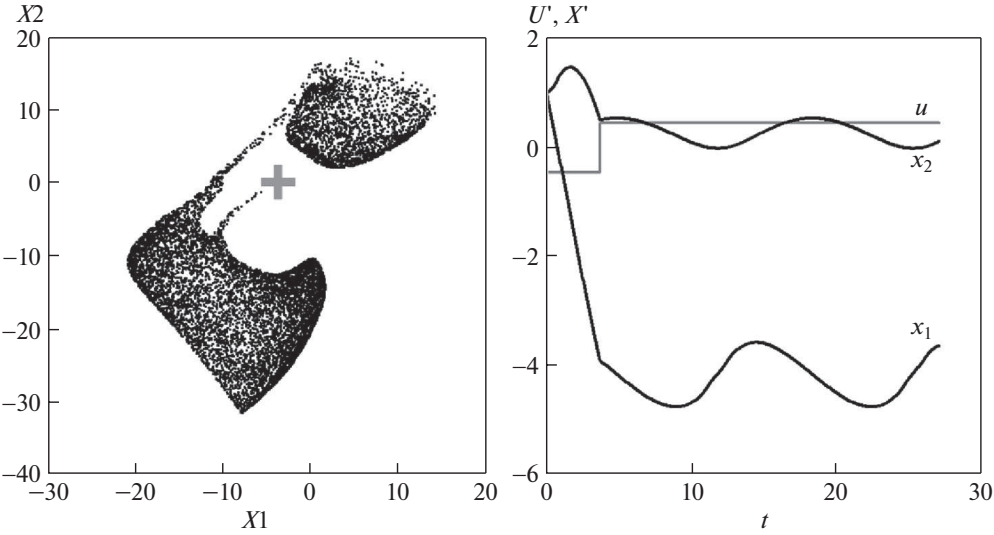


Рис. 3. Множество достижимости и экстремальная точка (слева), оптимальные траектории и управление (справа) в представленной модельной задаче.

6. Заключение

Проведенные эксперименты подтвердили теоретический вывод о неприемлемости использования разностных схем для решения задач размерности более 10^3. Время, необходимое для получения информации о матрице Гессе (точнее, только о ее диагонали), становится совершенно неприемлемым. Это продемонстрировано в табл. 2 и 3 соотношением производительности вариантов алгоритма, второй из которых основан на разностной схеме для аппроксимации диагонали гессиана.

К достоинствам предложенного алгоритма следует отнести его способность решать задачи существенно больших размерностей, чем любые методы, требующие полной квадратичной памяти, в том числе, методы ньютоновского типа, квазиньютоновские методы, метод Пауэлла и другие. Наблюдение графиков сходимости (значений функций с ростом числа итераций) показывает, что, к сожалению, эффект ускорения за счет информации второго порядка в данной конструкции алгоритмов оказывается существенным только на начальных стадиях расчетов, позволяя очень быстро получить неплохие результаты. Далее проявляется «эффект малых вариаций», что объясняется, очевидно, переходом алгоритма в режим «почти градиентного» метода.

Данный алгоритм может использоваться для решения стандартной задачи оптимального управления, которая на практике характеризуется наличием нелинейных систем дифференциальных уравнений и невыпуклых функционалов, что, как правило, приводит к неединственности решения и необходимости разрабатывать специализированные вычислительные технологии их исследования. С другой стороны, рост вычислительных мощностей современных компьютеров позволяет повышать эффективность алгоритмов поиска решения в задачах оптимизации, в том числе за счет использования технологий

Таблица 2. Результаты вычислительных экспериментов для семейства квазисепарабельных функций

n	5	50	$5 \cdot 10^2$	$5 \cdot 10^3$	$5 \cdot 10^4$	$5 \cdot 10^5$
$Var1\ f_{rec}$	$4{,}29 \cdot 10^{-21}$	$2{,}99 \cdot 10^{-19}$	$1{,}55 \cdot 10^{-7}$	$1{,}05 \cdot 10^{-11}$	$1{,}29 \cdot 10^{-11}$	$1{,}36 \cdot 10^{-11}$
$Var1\ iter$	10	10	10	121	1413	17 231
$Var1\ ng$	$1{,}3 \cdot 10^{-10}$	$1{,}1 \cdot 10^{-9}$	$8{,}6 \cdot 10^{-9}$	$9{,}0 \cdot 10^{-6}$	$9{,}9 \cdot 10^{-6}$	$1{,}0 \cdot 10^{-5}$
$Var2\ f_{rec}$	$1{,}07 \cdot 10^{-12}$	$9{,}09 \cdot 10^{-12}$	$1{,}11 \cdot 10^{-11}$	$7{,}69 \cdot 10^{-12}$	—	—
$Var2\ iter$	11	13	24	149	—	—
$Var2\ ng$	$2{,}1 \cdot 10^{-6}$	$6{,}0 \cdot 10^{-6}$	$6{,}7 \cdot 10^{-6}$	$9{,}0 \cdot 10^{-6}$	—	—

Таблица 3. Результаты вычислительных экспериментов для семейства функций Розенброка–Скокова

n	5	50	$5 \cdot 10^2$	$5 \cdot 10^3$	$5 \cdot 10^4$	$5 \cdot 10^5$
$Var1\ f_{rec}$	$1{,}03 \cdot 10^{-9}$	$2{,}6 \cdot 10^{-13}$	$2{,}51 \cdot 10^{-13}$	$2{,}5 \cdot 10^{-13}$	$1{,}17 \cdot 10^{-12}$	—
$Var1\ iter$	$> 6{,}3 \cdot 10^6$	13 587	43 997	139 219	$> 0{,}2 \cdot 10^6$	—
$Var1\ ng$	$1{,}0 \cdot 10^{-5}$	$1{,}0 \cdot 10^{-5}$	$1{,}0 \cdot 10^{-5}$	$1{,}0 \cdot 10^{-5}$	$2{,}2 \cdot 10^{-5}$	—
$Var2\ f_{rec}$	$1{,}06 \cdot 10^{-9}$	$2{,}6 \cdot 10^{-13}$	$2{,}51 \cdot 10^{-13}$	—	—	—
$Var2\ iter$	$> 6{,}6 \cdot 10^6$	13 596	43 999	—	—	—
$Var2\ ng$	$1{,}0 \cdot 10^{-5}$	$1{,}0 \cdot 10^{-5}$	$1{,}0 \cdot 10^{-5}$	—	—	———

параллельного запуска однородных процессов. Проведенные вычислительные эксперименты продемонстрировали возможность использования предложенного алгоритма для решения вспомогательных задач больших размерностей, возникающих при редукции исходной задачи на мелкой сетке к последовательности конечномерных в рамках стандартной идеологии использования прямых методов для решения задач оптимального управления.

СПИСОК ЛИТЕРАТУРЫ

1. *Andrianov A.N., Anikin A.S., Bychkov I.V., Gornov A.Y.* Numerical solution of huge-scale quasiseparable optimization problems // Lobachevskii Journal of Mathematics. 2017. Vol. 38. No. 5. P. 870–873.
2. *Brent R.* Algorithms for Minimization without Derivatives. Prentice Hall (reprinted by Dover, 2013).
3. *Bushaw D.W.* Differential equations with a discontinuous forcing term. Stevens Inst. Technol. Experimental Towing Tank Rept. 469, Hoboken, N.J., 1953.
4. *Gornov A.Yu., Andrianov A.N., Anikin A.S.* Algorithms for the solution of huge quasiseparable optimization problems // Proc. of VI International Workshop: “Critical Infrastructures in the Digital World”. Irkutsk. 2016. P. 76–77.
5. *Skokov V.A.* Methods and algorithms for unconditional minimization of functions of many variables (review) // Scientific report. 1974.
6. *Ya-Xiang Yuan.* Recent advances in trust region algorithms // Math. Program., Ser. B. 2015. Vol. 151. P. 249–281.
7. *Беллман Р.* Динамическое программирование. М.: Изд-во иностр. лит., 1960.

8. *Васильев О.В.* Лекции по методам оптимизации. Иркутск: Изд-во Иркут. ун-та, 1994.
9. *Васильев О.В., Аргучинцев А.В.* Методы оптимизации в задачах и упражнениях. М.: Физматлит, 1999.
10. *Горнов А.Ю.* Вычислительные технологии решения задач оптимального управления. Новосибирск: Наука, 2009.
11. *Деннис Дж. мл., Шнабель Р.Б.* Численные методы безусловной оптимизации и решения нелинейных уравнений. М.: Мир, 1988.
12. *Дыхта В.А.* Оптимизация динамических систем с разрывными траекториями и импульсными управлениями // Соросовский образоват. журн. 1999. № 8. С. 110–115.
13. *Иванов В.В.* Методы вычислений на ЭВМ. Киев: Наукова Думка, 1986.
14. *Понтрягин Л.С., Болтянский В.Г., Гамкрелидзе Р.В., Мищенко Е.Ф.* Математическая теория оптимальных процессов. М.: Наука, 1961.
15. *Фельдбаум А.А.* Оптимальные процессы в системах автоматического регулирования // АиТ. 1953. Т. 14. № 6. С. 712–728.
16. *Черноусько Ф.Л., Колмановский В.Б.* Вычислительные и приближенные методы оптимального управления // Итоги науки и техники. Сер. Мат. анализ. Т. 20. 1977. С. 101–166.

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 01.02.2022

После доработки 19.04.2022

Принята к публикации 29.06.2022

© 2022 г. Н.А. ДРАГУНОВ (nikitadragunovjob@gmail.com),
Е.В. ДЮКОВА, д-р физ.-мат. наук (edjukova@mail.ru)
(Федеральный исследовательский центр
«Информатика и управление»
Российской академии наук, Москва)

ОБ ОДНОМ ПОДХОДЕ К РАСШИФРОВКЕ МОНОТОННОЙ ЛОГИЧЕСКОЙ ФУНКЦИИ

Рассматривается задача расшифровки двузначной монотонной функции f, определенной на k-значном n-мерном кубе. Традиционным подходом к решению данной задачи является построение оптимального по Шеннону алгоритма. Оптимальный по Шеннону алгоритм расшифровки имеет минимальную сложность в «худшем случае» (эффективен для наиболее трудного варианта задачи). Авторами предложен и исследован подход к задаче расшифровки, основанный на применении асимптотически оптимального алгоритма дуализации над произведением k-значных цепей. Асимптотически оптимальная расшифровка функции f нацелена на «типичный случай» (на типичный вариант задачи). Экспериментально выявлены условия применимости традиционного и нового подходов.

Ключевые слова: верхний ноль монотонной логической функции, нижняя единица монотонной логической функции, оптимальный по Шеннону алгоритм расшифровки, асимптотически оптимальный алгоритм расшифровки, максимальный частый элемент, минимальный нечастый элемент, дуализация над произведением k-значных цепей.

DOI: 10.31857/S0005231022100129, **EDN:** ALIJYE

1. Введение

Расшифровка двузначной монотонной функции, определенной на k-значном n-мерном кубе, — одна из важнейших задач дискретной математики. Задача формулируется следующим образом.

Положим

$$E_k^n = \{(\alpha_1, \ldots, \alpha_n) \,|\, \alpha_i \in \{0, 1, \ldots, k-1\} \text{ при } i = 1, 2, \ldots, n,\ k \geq 2\}.$$

Множество E_k^n называется k-значным n-мерным кубом. На E_k^n устанавливается частичный порядок, согласно которому элемент $\beta = (\beta_1, \ldots, \beta_n)$ из E_k^n следует за элементом $\alpha = (\alpha_1, \ldots, \alpha_n)$ из E_k^n (α предшествует β), если $\beta_i \geq \alpha_i$ при $i = 1, 2, \ldots, n$. Для обозначения того, что $\beta \in E_k^n$ следует за $\alpha \in E_k^n$, используется запись $\alpha \preccurlyeq \beta$ или $\beta \succcurlyeq \alpha$.

Функция f, определенная на E_k^n и принимающая два значения 0 и 1, называется монотонной, если для любых двух элементов α и β из E_k^n таких, что $\alpha \preccurlyeq \beta$, выполнено $f(\beta) \geq f(\alpha)$. Функция f задается при помощи некоторого оператора B, который для любого $x \in E_k^n$ выдает значение $f(x)$. Если

$f(x) = 0$, то элемент x называется нулем функции f, если же $f(x) = 1$, то элемент x называется единицей функции f. Требуется путем «минимального» числа обращений к оператору B найти множество всех нулей функции f и множество всех ее единиц.

Традиционный подход к решению задачи расшифровки основан на построении оптимального по Шеннону алгоритма (предложен В.К. Коробковым в 1965 г. в [1]). Согласно данному подходу, сложность алгоритма расшифровки следует оценивать числом обращений к оператору B в худшем случае. Это означает следующее. Пусть V — множество всех двузначных монотонных функций, определенных на E_k^n. Пусть A — некоторый алгоритм, выполняющий расшифровку функций из V. Под сложностью алгоритма A понимается максимум числа обращений к оператору B, где максимум берется по всем функциям из V. Алгоритм A называется оптимальным по Шеннону на множестве V, если его сложность минимальна среди всех алгоритмов, выполняющих расшифровку функций из V.

Оптимальный по Шеннону алгоритм расшифровки монотонной булевой функции построен Ж. Анселем в 1968 г. [2]. В 1976 г. В.Б. Алексеевым предложен алгоритм расшифровки функции из V, который является оптимальным в случае $k = 2$ и близок по сложности к оптимальному в случае $k > 2$ [3].

Введем понятия верхнего нуля и нижней единицы функции f, $f \in V$. Эти понятия являются центральными для рассматриваемой задачи расшифровки. Ноль функции f называется верхним, если он не предшествует никакому другому нулю этой функции. Единица функции f называется нижней, если она не следует ни за какой другой единицей этой функции. Очевидно, что для расшифровки f достаточно найти все ее верхние нули и все ее нижние единицы.

Пусть D — произвольная совокупность элементов из E_k^n, $x \in E_k^n$, $s \in [0, 1]$. Элемент x называется s-частым, если доля элементов в D, следующих за x, не меньше s, иначе x — s-нечастый элемент. Элемент x называется максимальным s-частым, если x — s-частый элемент и любой другой следующий за ним элемент является s-нечастым. Элемент x называется минимальным s-нечастым, если x — s-нечастый элемент и любой другой предшествующий ему элемент является s-частым.

Пусть $X_{\max}$ и $Y_{\min}$ — множества, состоящие соответственно из всех максимальных s-частых и минимальных s-нечастых элементов множества E_k^n. На множестве E_k^n определим монотонную функцию $f_{D,s}$, которая принимает значения 0 и 1 соответственно на s-частых элементах и s-нечастых элементах этого множества. Фактически $f_{D,s}$ задается при помощи оператора B_D, который для произвольного x из E_k^n выдает значение $f_{D,s}(x)$ путем вычисления частоты встречаемости x в D. Таким образом, для расшифровки функции $f_{D,s}$ могут применяться методы поиска множеств $X_{\max}$ и $Y_{\min}$, и наоборот, для поиска $X_{\max}$ и $Y_{\min}$ применимы методы расшифровки $f_{D,s}$.

Следует отметить, что основным приложением методов поиска частых и нечастых элементов в данных, в том числе частично упорядоченных, являются вопросы построения ассоциативных правил, впервые возникшие в связи с задачей анализа потребительской корзины [4]. В машинном обучении логи-

ческий анализ признаковых описаний прецедентов фактически основан на поиске в этих описаниях часто и нечасто встречающихся фрагментов [5].

В [6] анонсирована идея последовательно-совместного перечисления $X_{\max}$ и $Y_{\min}$, основанная на решении задачи дуализации над произведением k-значных цепей, и приведены результаты тестирования последовательно-совместного поиска $X_{\max}$ и $Y_{\min}$ на случайных модельных данных, показавшие его эффективность в случае, когда мощности $X_{\max}$ и $Y_{\min}$ примерно равны.

В настоящей работе проведено экспериментальное сравнение двух методов расшифровки функции $f_{D,s}$. Первый метод — это упомянутый выше алгоритм расшифровки В.Б. Алексеева. Второй метод основан на предложенной в [6] идее последовательно-совместного перечисления $X_{\max}$ и $Y_{\min}$ с применением асимптотически оптимального алгоритма дуализации над произведением k-значных цепей RUNC-M+ [7]. Задача дуализации относится к числу труднорешаемых дискретных задач и асимптотически оптимальные алгоритмы дуализации лидируют по скорости счета. На случайных модельных данных для каждого тестируемого метода расшифровки оценивалось время работы и число обращений к оператору B_D. Показано, что асимптотически оптимальная расшифровка функции $f_{D,s}$, нацеленная на типичный вариант задачи, в определенных случаях имеет лучшие результаты по сравнению с оптимальной по Шеннону расшифровкой, ориентирующейся на самый сложный вариант задачи.

2. Традиционный подход к расшифровке монотонной логической функции. Алгоритм Алексеева

В настоящем разделе рассматривается алгоритм A_{opt} расшифровки функции f из V, описанный в [3]. Как уже было отмечено во введении, этот алгоритм является оптимальным по Шеннону в случае $k = 2$ и близок по сложности к оптимальному в случае $k > 2$.

Алгоритм A_{opt} работает в два этапа. На первом этапе куб E_k^n разбивается на непересекающиеся подмножества, каждое из которых является цепью. На втором этапе выполняется расшифровка f на каждой построенной цепи с помощью хорошо известного алгоритма двоичного поиска.

Пусть $i \in \{2, \ldots, n\}$, $r \in \{0, 1, \ldots, k-1\}$. Положим

$$S_r^i = \left\{(\alpha_1, \ldots, \alpha_{i-1}, r) \mid (\alpha_1, \ldots, \alpha_{i-1}) \in E_k^{i-1}\right\}.$$

Процесс разбиения куба E_k^n на непересекающиеся цепи происходит путем последовательного построения разбиений на непересекающиеся цепи кубов меньшей размерности.

На первом шаге рассматривается куб E_k^1, представляющий собой цепь согласно установленному частичному порядку.

Пусть на шаге i — $1, 2 \le i \le n$, куб E_k^{i-1} разбит на непересекающиеся цепи. Тем самым, очевидным образом на непересекающиеся цепи разбито каждое из множеств S_r^i, $r \in \{0, 1, \ldots, k-1\}$. Далее построенные разбиения множеств $S_0^i, \ldots, S_{k-1}^i$ изменяются. Сначала в S_0^i добавляются все максимальные

элементы цепей из построенных разбиений для множеств $S_1^i, \ldots, S_{k-1}^i$, при этом все добавленные к S_0^i элементы удаляются из множеств $S_1^i, \ldots, S_{k-1}^i$. Затем аналогичная процедура проводится для измененной последовательности $S_1^i, \ldots, S_{k-1}^i$ и т.д. В результате, учитывая, что $E_k^i = \bigcup_{r=0}^{k-1} S_r^i$, получается требуемое разбиение для куба E_k^i.

Построенные на первом этапе работы алгоритма цепи просматриваются в порядке не убывания их мощности. Пусть C_i — очередная цепь. Если для некоторого элемента $x \in C_i$ известно, что $x \quad y$, где y — ноль, принадлежащий ранее просмотренной цепи C_j $(j < i)$, то x — ноль цепи C_i. Аналогично, если для некоторого элемента $x \in C_i$ известно, что $y \quad x$, где y — единица, принадлежащая ранее просмотренной цепи C_j $(j < i)$, то x — единица цепи C_i. Таким образом, цепь C_i делится на три отрезка: сначала следуют найденные нули, затем следуют элементы, на которых значение функции f неизвестно, после чего следуют найденные единицы. Для расшифровки цепи C_i на втором отрезке запускается алгоритм двоичного поиска. При этом для определения значения функции f происходит обращение к оператору B.

Пусть $t_A(f)$ — общее число обращений к оператору B алгоритма A, выполняющего расшифровку функции f из V. Сложностью алгоритма A по Шеннону (сложностью в худшем случае) называется величина $\max[t_A(f)]$, где максимум берется по всем функциям из V. Пусть A_0 — любой алгоритм расшифровки функций из V с наименьшей сложностью. Тогда $t_{A_{\text{opt}}}(f)/t_{A_0}(f) \leq 0{,}5\,(\log_2(k)+1)$.

3. Новый подход к расшифровке монотонной логической функции. Асимптотически оптимальная расшифровка

В данном разделе описывается подход к задаче расшифровки монотонной логической функции, базирующийся на применении алгоритмов дуализации над произведением k-значных цепей.

3.1. Дуализация над произведением k-значных цепей

Задача дуализации над произведением k-значных цепей относится к числу труднорешаемых перечислительных задач дискретной математики и ставится следующим образом.

Представим множество E_k^n в виде декартова произведения n цепей мощности k, положив $E_k^n = E_1 \times E_2 \times \ldots \times E_n$, где каждое $E_i = \{0, 1, \ldots, k-1\}$. Считается, что элемент $\beta = (\beta_1, \ldots, \beta_n)$ из E_k^n следует за элементом $\alpha = (\alpha_1, \ldots, \alpha_n)$ из E_k^n (α предшествует β), если $\beta_i \geq \alpha_i$ при $i = 1, 2, \ldots, n$.

Элемент $x \in E$, $E \subset E_k^n$, называется минимальным элементом множества E, если не существует другого элемента множества E, предшествующего x. Элемент $x \in E$, $E \subset E_k^n$ называется максимальным элементом множества E, если не существует другого элемента множества E, следующего за x.

Пусть $E \subset E_k^n$. Введем обозначения: E^+ — множество всех элементов из E_k^n, каждый из которых следует хотя бы за одним элементом из E; E^- —

множество всех элементов из E_k^n, каждый из которых предшествует хотя бы одному элементу из E; $Q(E)$ — множество, состоящее из минимальных элементов множества $E_k^n \backslash E^-$ (здесь и далее обозначение $A \backslash B$ используется для разности множеств A и B); $G(E)$ — множество, состоящее из всех максимальных элементов множества $E_k^n \backslash E^+$.

Каждая из задач построения множества $Q(E)$ или $G(E)$ называется задачей дуализации над произведением k-значных цепей.

Если $k = 2$, то к построению $Q(E)$ сводится задача поиска нижних единиц монотонной булевой функции, заданной множеством нулей E, называемая задачей монотонной дуализации. В матричной формулировке монотонная дуализация — это задача построения неприводимых покрытий булевой матрицы, строками которой являются элементы из E. Эквивалентной задачей является перечисление минимальных вершинных покрытий гиперграфа.

Если $k \geq 2$ и множество E состоит из попарно несравнимых элементов, то к построению $Q(E)$ сводится поиск нижних единиц двузначной монотонной функции k-значной логики, заданной множеством верхних нулей E. Аналогично к построению $G(E)$ сводится поиск верхних нулей двузначной монотонной функции k-значной логики, заданной множеством нижних единиц E.

В теории алгоритмической сложности дискретных задач эффективность алгоритмов для перечислительных задач принято оценивать временем выполнения одного шага, т.е. временем нахождения очередного нового решения. Наиболее эффективными считаются алгоритмы с полиномиальными временны́ми оценками. Такие алгоритмы имеют временны́е оценки вида $O(N)$, где N — полином от размера входа задачи, и называются алгоритмами с полиномиальными задержками. Причем временны́е оценки даются для самой сложной индивидуальной задачи. Полиномиальные алгоритмы удалось построить для немногих частных случаев монотонной дуализации [8]. Наилучший теоретический результат получен в [9]. Предложенный в [9] инкрементальный квазиполиномиальный алгоритм монотонной дуализации имеет временную оценку вида $N^{o(logN)}$, где N — полином не только от размера входа задачи, но и от числа решений, найденных на предыдущих шагах, т.е. N — полином от размера входа и выхода задачи.

В [10] предложен подход к построению асимптотически оптимальных алгоритмов монотонной дуализации. Эти алгоритмы имеют теоретическое обоснование эффективности и показывают хорошие результаты по скорости счета. Асимптотически оптимальный алгоритм отличается от алгоритма с полиномиальной задержкой тем, что имеет лишние полиномиальные шаги. Это шаги, на которых не строятся новые решения. Основное требование: для почти всех индивидуальных задач число лишних шагов алгоритма должно быть мало по сравнению с числом всех решений задачи. При этом проверка того, является ли шаг лишним, должна происходить за полиномиальное от размера входа время. Данный подход к задаче монотонной дуализации значительно позже был продемонстрирован в работе [11] на примере задачи построения минимальных вершинных покрытий гиперграфа.

3.2. Последовательно-совместное перечисление верхних нулей и нижних единиц монотонной логической функции

Алгоритм последовательно-совместного перечисления верхних нулей X_{up} и нижних единиц Y_{low} функции f, $f \in V$, заданной при помощи оператора B, основан на решении задачи дуализации над произведением k-значных цепей. Рассматриваемый алгоритм является синтезом последовательного и совместного подходов к поиску X_{up} и Y_{low}, которые подробно описаны в разделе 3.3. Метод базируется на приведенных ниже утверждениях 1–3.

Утверждение 1. Если $X \subset X_{\text{up}}$, то $Q(X)$ содержит хотя бы один ноль функции f.

Доказательство. Пусть $X \subset X_{\text{up}}$, $x \in X_{\text{up}} \backslash X$. Из того, что x не сравним ни с одним другим элементом множества X_{up}, следует, что $x \in E_k^n \backslash X^-$. Таким образом, в $Q(X)$ существует элемент q такой, что $q \quad x$ и $f(q) = 0$.

Утверждение 2. Если $X \subset X_{\text{up}}$, а элемент $y \in Q(X)$ является единицей функции f, то y — нижняя единица функции f.

Доказательство. Пусть $y \notin Q(X_{\text{up}}) = Y_{\text{low}}$. Тогда, так как y — единица функции f, то в $E_k^n \backslash X_{\text{up}}^-$ найдется нижняя единица z такая, что $z \neq y$, $z \quad y$. Из $(E_k^n \backslash X_{\text{up}}^-) \subseteq (E_k^n \backslash X^-)$, следует, что $z \in E_k^n \backslash X^-$, что противоречит условию $y \in Q(X)$.

Утверждение 3. Пусть $X \subseteq X_{\text{up}}$, $Y \subseteq Y_{\text{low}}$. Тогда $Q(X) = Y$ в том и только в том случае, если $X = X_{\text{up}}$, $Y = Y_{\text{low}}$.

Доказательство. Пусть $X \subset X_{\text{up}}$. Из утверждения 1 следует, что $Q(X)$ содержит хотя бы один ноль f. Однако в множестве Y нет нулей функции f, следовательно $Q(X) \neq Y_{\text{low}}$. Если же $X = X_{\text{up}}$, то $Q(X) = Y_{\text{low}}$. Таким образом, $Q(X) = Y$ тогда и только тогда, когда $X = X_{\text{up}}$, $Y = Y_{\text{low}}$.

Последовательно-совместный алгоритм работает следующим образом. Положим $X_0 = \varnothing$. Строится последовательность $X_1 \subset X_2 \subset \ldots \subset X_{\text{up}}$.

Шаг 1. Рассматривается множество $X_1 = \{x\}$, где x — произвольный верхний ноль f.

Шаг i+1 ($i \geq 1$). Решается задача дуализации множества $X_i \backslash X_{i-1}$. Пусть множество Z есть результат дуализации $X_i \backslash X_{i-1}$. Согласно утверждению 1, множество Z содержит хотя бы один ноль функции f. Для каждого нуля из Z находится один содержащий его верхний ноль. Все найденные верхние нули, которые не содержатся в множестве X_i, добавляются к X_i, формируя в результате множество X_{i+1}. Если же все найденные верхние нули уже содержатся в X_i, то происходит дуализация множества X_i, в результате чего формируется множество $Q(X_i)$. Если в $Q(X_i)$ нет нулей, то, согласно утверждению 3, следует, что $Q(X_i) = Y_{\text{low}}$, $X_i = X_{\text{up}}$, и алгоритм завершает свою работу. Иначе для каждого нуля из $Q(X_i)$ находится один содержащий его верхний ноль, который пополняет множество X_i, формируя в результате множество X_{i+1}.

3.3. Последовательный и совместный поиск X_{up} и Y_{low}

Достаточно очевидным является поиск X_{up} и Y_{low} функции f, $f \in V$, заданной при помощи оператора B, путем последовательного построения этих множеств. Сначала строится множество X_{up}, например, алгоритмом Apriori [4, 12], модифицированным на случай произведения k-значных цепей. Затем используется свойство двойственности $Q(X_{\text{up}}) = Y_{\text{low}}$. Аналогично можно сначала строить Y_{low} модифицированным алгоритмом Apriori, а затем искать X_{up} путем дуализации множества Y_{low}.

В [13] предложена идея совместного перечисления множеств $X_{\max}$ и $Y_{\min}$, которая в применении к задаче построения X_{up} и Y_{low} заключается в следующем.

Выбирается произвольный элемент q из E_k^n. Если q — ноль функции f, то ищется верхний ноль, следующий за q. Если же q — единица функции f, то ищется нижняя единица, предшествующая q. Пусть на очередной итерации построены множества $X \subseteq X_{\text{up}}$ и $Y \subseteq Y_{\text{low}}$. Если $X \neq \varnothing$ и $Y = \varnothing$, то выбирается любой $x \in X$ и ищется элемент q, который не предшествует x. Если $X = \varnothing$ и $Y \neq \varnothing$, то выбирается любой $y \in Y$ и ищется элемент q, который не следует за y. Если же $X \neq \varnothing$ и $Y \neq \varnothing$, то ищется элемент q, который не предшествует x и не следует за y. Затем аналогичным образом в зависимости от значения элемента q находится верхний ноль или нижняя единица функции f.

Алгоритм, основанный на описанной выше идее совместном перечислении множеств X_{up} и Y_{low}, строит вложенные последовательности: $X_1 \subset X_2 \subset \ldots \subset X_{\text{up}}$ и $Y_1 \subset Y_2 \subset \ldots \subset Y_{\text{low}}$.

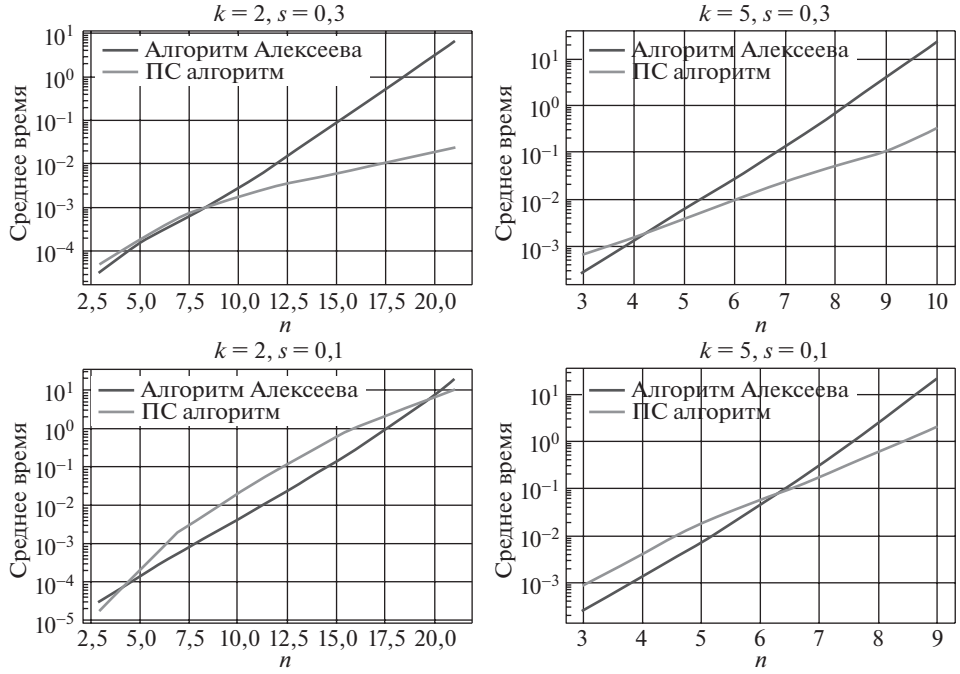
Шаг 1. $X_1 = \{x\}$, $Y_1 = \{y\}$, где x и y находятся модифицированным алгоритмом Apriori.

Шаг $i+1$ ($i \geq 1$). Строится либо $Q(X_i)$, либо $G(Y_i)$. Пусть построено множество $Q(X_i)$. Если $Q(X_i)$ не содержит нулей функции f, то, согласно утверждению 3, оно совпадает с множеством Y_{low} (в этом случае $X_i = X_{\text{up}}$ и алгоритм заканчивает работу). Согласно утверждению 2, каждая единица из множества $Q(X_i)$ является нижней единицей. Эти единицы пополняют множество Y_i, формируя в результате множество Y_{i+1}. Для каждого нуля из $Q(X_i)$ находится один содержащий его верхний ноль элемент путем последовательного увеличения текущего нуля согласно заданному порядку, который затем пополняет множество X_i, формируя в результате множество X_{i+1}.

В [6] приведены результаты тестирования последовательного, совместного и последовательно-совместного поиска $X_{\max}$ и $Y_{\min}$ на случайных модельных данных. Согласно этим результатам, последовательно-совместное перечисление наиболее эффективно, когда мощности множеств $X_{\max}$ и $Y_{\min}$ примерно равны, иначе более эффективным является последовательное перечисление. Наихудшие показатели у совместного перечисления множеств $X_{\max}$ и $Y_{\min}$.

4. Эксперименты

Проведено экспериментальное сравнение двух алгоритмов расшифровки функции $f_{D,s}$, описанных в разделах 3.2 и 3.3. Алгоритмы реализованы



Зависимость времени работы алгоритмов от n в секундах (при различных k и s).

на языке C++. При реализации последовательно-совместного перечисления верхних нулей и нижних единиц функции $f_{D,s}$ использовался асимптотически оптимальный алгоритм над произведением k-значных цепей RUNC-M+ [7].

Эксперименты проведены для случайных множеств D, $D \subset E_k^n$, с числом элементов $m = 100$. Данные выбирались из равномерного распределения. Результаты усреднялись по 20 независимым запускам.

Из представленных на рисунке графиков следует, что при $s = 0{,}3$ и $n > 5$ независимо от значения k последовательно-совместный алгоритм (ПС алгоритм) работает быстрее алгоритма Алексеева. Алгоритм Алексеева более эффективен при $s = 0{,}1$, $k = 2$, но при этом время его работы растет быстрее с ростом n. В случае $k = 5$, $n > 7$ последовательно-совместный алгоритм на порядки быстрее алгоритма Алексеева.

Таблица 1. Среднее число обращений к оператору B_D, $m = 100$

n; k; s	Алгоритм Алексеева	ПС алгоритм
5; 3; 0,1	**111**	529
5; 3; 0,3	**56**	343
5; 20; 0,3	25 484	**3039**
10; 3; 0,1	**2709**	10 172
10; 3; 0,3	**323**	3111
10; 5; 0,1	45 528	**31 791**
15; 3; 0,3	**1142**	13 940

Как видно из табл. 1, в случае небольших значений k, независимо от значения n, наилучший результат по числу обращений к оператору B_D показывает алгоритм Алексеева. Заметим, что при небольших значениях k сложность этого алгоритма почти оптимальна. При значениях $k \geq 5$ лучший результат по числу обращений к оператору B_D показывает последовательно-совместная расшифровка функции $f_{D,s}$.

5. Заключение

Исследованы актуальные вопросы уменьшения временных затрат, возникающие при логическом анализе частично упорядоченных данных. Разработан и исследован новый подход к задаче расшифровки двузначной монотонной функции, определенной на k-значном n-мерном кубе, основанный на последовательно-совместном перечислении верхних нулей и нижних единиц этой функции. Экспериментальное исследование предлагаемого подхода проведено с использованием авторской идеи последовательно-совместного поиска максимальных частых и минимальных нечастых элементов произведения k-значных цепей, базирующейся на решении задачи дуализации над произведением k-значных цепей. Показана целесообразность применения асимптотически оптимальных алгоритмов дуализации над произведением k-значных цепей для рассматриваемой задачи расшифровки монотонной логической функции.

СПИСОК ЛИТЕРАТУРЫ

1. *Коробков В.К.* О монотонных функциях алгебры логики // Сб. Проблемы кибернетики. Вып. 13. М.: Наука, 1965. С. 5–28.
2. *Анселъ Ж.* О числе монотонных булевых функций n переменных // Кибернетич. сб. Нов. сер. Вып. 5. М.: Мир, 1968. С. 53–57.
3. *Алексеев В.Б.* О расшифровке некоторых классов монотонных многозначных функций // Журн. вычисл. матем. и матем. физики. 1976. Т. 16. № 1. С. 189–198.

   Alekseev V.B. Deciphering of some classes of monotonic many-valued functions // Zh. vychisl. Mat. mat. Fiz. 1976. V. 16. No. 1. P. 189–198.
4. *Agrawal R., Imielinski T., Swami A.* Mining association rules between sets of items in large databases // Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data. 1993. P. 207–216.
5. *Журавлев Ю.И., Рязанов В.В., Сенько О.В.* Распознавание. Математические методы. Программная система. Практические применения. М.: ФАЗИС, 2006. Т. 176.
6. *Драгунов Н.А., Дюкова Е.В.* Поиск минимальных нечастых и максимальных частых наборов в частично упорядоченных данных // Математические методы распознавания образов: Тезисы докладов 9-й Всероссийской конференции с международным участием, 2019. С. 10–12.
7. *Дюкова Е.В., Масляков Г.О., Прокофьев П.А.* Дуализация над произведением цепей: асимптотические оценки числа решений // ДАН. 2018. Т. 483. № 2. С. 130–133.
8. *Johnson D.S., Yannakakis M., Papadimitriou C.H.* On general all maximal independent sets // Inform. Proc. Lett. 1988. V. 27. Iss. 3. P. 119–123.

9. *Fredman M.L., Khachiyan L.* On the complexity of dualization of monotone disjunctive normal forms // J. Algorithms. 1996. No. 21. P. 618–628.
10. *Дюкова Е.В.* Об асимптотически оптимальном алгоритме построения тупиковых тестов // ДАН СССР. 1977. Т. 233. № 4. С. 527–530.

    Djukova E.V. On an asymptotically optimal algorithm for constructing irredundant tests // DAN SSSR. 1977. V. 233. No. 4. P. 423–426.
11. *Murakami K., Uno T.* Efficient algorithms for dualizing large-scale hypergraphs // Discr. Appl. Math. 2014. V. 170. P. 83–94.
12. *Aggarwal C.* Frequent Pattern Mining. Springer International Publishing. Switzerland, 2014.
13. *Elbassioni K.* On Finding Minimal Infrequent Elements in Multidimensional Data Defined Over Partially Ordered Sets. 2014. arXiv:1411.2275.

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 01.02.2022

После доработки 27.03.2022

Принята к публикации 29.06.2022

© 2022 г. А.Н. ТЫРСИН, д-р техн. наук (at2001@yandex.ru)
(Уральский федеральный университет, Екатеринбург;
Научно-инженерный центр
«Надежность и ресурс больших систем и машин»
УрО РАН, Екатеринбург)

ЭНТРОПИЙНОЕ МОДЕЛИРОВАНИЕ СЕТЕВЫХ СТРУКТУР[1]

В настоящее время довольно часто используется энтропия для описания сложных систем в различных областях. Рассмотрены вопросы использования дифференциальной энтропии для сетевых структур, представленных в виде связных графов с корреляционными связями. Известно, что энтропию непрерывного случайного вектора можно разложить на две составляющие: энтропию случайности и энтропию самоорганизации. Для сетевых структур наряду с оценкой самой энтропии предложены другие полезные характеристики — энтропийная мера взаимосвязи между несколькими подсистемами и энтропия системы в отдельной вершине, которые расширяют возможности энтропийного моделирования для исследования сетевых структур: позволят оценить взаимосвязанность разных участков между собой и определить, как изменяется энтропия внутри таких систем. Рассмотрены примеры на модельных данных.

Ключевые слова: сетевая структура, случайный вектор, дифференциальная энтропия, граф, система, подсистема, взаимозависимость.

DOI: 10.31857/S0005231022100130, **EDN:** ALLDVU

1. Введение

Структура — это совокупность устойчивых связей между элементами системы, обеспечивающих воспроизводимость при изменяющихся условиях [1]. В холистическом понимании структура вместе с элементами образует систему.

Сетевая структура рассматривается как децентрализованный комплекс взаимосвязанных элементов, способный расширяться путем включения новых звеньев, что придает сети гибкость и динамичность [2]. В сетевых структурах потенциально могут существовать связи между всеми элементами, причем эти связи могут быть разнонаправленными, т.е. могут быть и двойное подчинение, и межуровневое взаимодействие [3]. Также в системе могут быть подсистемы, что тоже должно отражаться в сетевой структуре как взаимосвязи на уровне подсистем. Поведение реальных систем часто обладает стохастичностью, а связи между их элементами можно адекватно описывать как корреляционные. Модели таких систем обычно акцентируют внимание на явном количественном описании вероятностных характеристик тех или иных ситуаций в жизненном цикле системы [4–7]. Однако такие модели не

[1] Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проект № 20-51-00001).

позволяют учесть системные характеристики сетевых структур, что может ограничить возможности выработки эффективных управленческих решений.

Энтропия является универсальным параметром, позволяющим объединять различные проявления физического мира в единое целое, т.е. может служить системной характеристикой. В настоящее время достаточно распространено использование энтропии для описания поведения открытых стохастических систем в различных областях [8–11]. Общим в этих работах является использование введенной К. Шенноном информационной энтропии [12]

$$H(\mathbf{D}) = -\sum_{i=1}^{L} p_i \ln p_i, \tag{1}$$

где $p_1, \ldots, , p_L$ — вероятности того, что система принимает конечное число соответствующих состояний D_i, т.е. $p_i = P(S \in D_i)$.

Согласно (1) модель системы представляется как функция от множества ее состояний $\mathbf{D}$. Однако использование информационной энтропии в качестве модели такой системы имеет существенные недостатки:

1. Требуется оценить вероятности p_i. Это требует больших выборок, для некоторых состояний статистику получить практически невозможно.

2. Некоторые состояния систем заранее могут быть вообще не известны.

3. Затруднено моделирование взаимосвязей между элементами многомерных систем.

4. Не учитывается изменение дисперсии.

5. Основные системные закономерности не учитываются.

6. Адекватные энтропийные модели разработаны только для частных задач.

Поэтому актуальна задача поиска более информативных энтропийных характеристик, описывающих поведение сетевых структур.

2. Постановка задачи энтропийного моделирования сетевых структур

Более адекватным подходом к описанию стохастических систем является выделение не отдельных состояний, а ее элементов, которые всегда можно задать для конкретной системы. Представим систему в виде непрерывного случайного вектора $\mathbf{Y} = (Y_1, \ldots, Y_m)$ с взаимосвязанными компонентами. Плотность вероятности $p_{\mathbf{Y}}(x_1, \ldots, x_m)$ полностью описывает распределение вероятностей многомерной случайной величины $\mathbf{Y}$. Поэтому вместо информационной энтропии будем использовать дифференциальную энтропию [12], являющуюся функционалом от плотности вероятности случайного вектора $\mathbf{Y}$,

$$H(\mathbf{Y}) = -\int_{-\infty}^{+\infty} \ldots \int_{-\infty}^{+\infty} p_{\mathbf{Y}}(x_1, \ldots, x_m) \ln p_{\mathbf{Y}}(x_1, \ldots, x_m) dx_1 \ldots dx_m. \tag{2}$$

Формула (2) была предложена К. Шенноном как формальный аналог понятия информационной энтропии (энтропия непрерывного распределения)

для m-мерного непрерывного случайного вектора $\mathbf{Y}$. Эта величина впоследствии А.Н. Колмогоровым совместно с И.М. Гельфандом и А.М. Ягломом была исследована и названа дифференциальной энтропией [13].

В [14] доказано, что если все компоненты Y_i имеют дисперсии $\sigma^2_{Y_i}$, то дифференциальная энтропия (далее, энтропия) $H(\mathbf{Y})$ случайного вектора $\mathbf{Y}$ равна

$$H(\mathbf{Y}) = \sum_{i=1}^{m} \ln \sigma_{Y_i} + \sum_{i=1}^{m} \kappa_i + \frac{1}{2} \sum_{k=2}^{m} \ln \left(1 - R^2_{Y_k|Y_1 \ldots Y_{k-1}}\right), \tag{3}$$

где $\kappa_i = H(Y_i/\sigma_{Y_i}) = H(\widehat{Y}_i) = -\int_{-\infty}^{+\infty} p_{\widehat{Y}_i}(x) \ln p_{\widehat{Y}_i}(x) dx$ — дифференциальные энтропии по плотностям с единичными дисперсиями; $R^2_{Y_k|Y_1 \ldots Y_{k-1}} = \dfrac{\sigma^2_{Y_k|Y_1 \ldots Y_{k-1}}}{\sigma^2_{Y_k}}$ — индекс детерминации в общем случае нелинейной регрессионной зависимости случайной величины Y_k, от случайных величин $Y_1, \ldots, Y_{k-1}$, (доля дисперсии Y_k, объясняемая изменением переменных $Y_1, \ldots, Y_{k-1}$). При неизвестном виде зависимости для определения $R^2_{Y_k|Y_1 \ldots Y_{k-1}}$ можно воспользоваться методами непараметрического регрессионного анализа [15].

Первые два слагаемых $H_V(\mathbf{Y}) = \sum\limits_{i=1}^{m} \ln \sigma_{Y_i} + \sum\limits_{i=1}^{m} \kappa_i = \sum\limits_{i=1}^{m} H(Y_i)$ равны сумме энтропий элементов $H(Y_i)$, что соответствует случаю, когда элементы Y_i системы $\mathbf{Y}$ взаимно независимы. Величина $H_V(\mathbf{Y})$ равна предельной энтропии, соответствующей взаимной независимости элементов системы, и характеризует рассмотрение целостного объекта как состоящего из частей (аддитивность системы). Поэтому $H_V(\mathbf{Y})$ назовем «энтропией хаотичности».

Возьмем третье слагаемое в (3) со знаком «$-$»:

$$-G_R(\mathbf{Y}) = \frac{1}{2} \sum_{k=2}^{m} \ln \left(1 - R^2_{Y_k|Y_1 \ldots Y_{k-1}}\right), \text{ т.е. } G_R(\mathbf{Y}) = -\frac{1}{2} \sum_{k=2}^{m} \ln \left(1 - R^2_{Y_k|Y_1 \ldots Y_{k-1}}\right).$$

Назовем $G_R(\mathbf{Y})$ энтропийной «мерой самоорганизации». Она отражает степень взаимосвязи между элементами системы $\mathbf{Y}$, характеризуя свойства системы как целого (целостность системы): при полной взаимной независимости между элементами системы $G_R(\mathbf{Y}) = 0$; при строгой функциональной зависимости между хотя бы двумя элементами системы $G_R(\mathbf{Y}) \to +\infty$. Отметим, что в случае двухкомпонентного вектора $\mathbf{Y} = (Y_1, Y_2)$ величина $G_R(\mathbf{Y})$ определяется средней взаимной информацией между Y_1 и Y_2, которая всегда неотрицательна [16].

Если $\mathbf{Y}^\circ$ — гауссов случайный вектор, тогда

$$H_V(\mathbf{Y}^\circ) = \sum_{i=1}^{m} \ln \sigma_{Y_i} + m \ln \sqrt{2\pi e}, \quad G_R(\mathbf{Y}^\circ) = -\frac{1}{2} \ln |\mathbf{R}_{\mathbf{Y}}|, \tag{4}$$

где $\mathbf{R}_{\mathbf{Y}^\circ} = \left\{\rho_{Y_i^\circ Y_j^\circ}\right\}_{m \times m}$ — корреляционная матрица случайного вектора $\mathbf{Y}^\circ$.

В рамках энтропийного моделирования сетевыми структурами будем называть системы, каждый из элементов которой связан хотя бы с одним из других элементов системы. Они могут быть представлены в виде связных графов, в которых связь между элементами (вершинами) задается в виде тесноты корреляционной связи.

Физические компоненты инфраструктуры моделируются как связный граф, где узлы представляют собой такие компоненты, как районы, развязки автодорог, железнодорожные депо, генераторы, телефонные коммутаторы, резервуары воды и т.д. Дуги графа характеризуют взаимосвязи между элементами. В рамках энтропийного моделирования эти взаимосвязи будем описывать теснотой корреляционной связи между элементами системы (узлами графа). Для сетевых структур недостаточно ограничиться только моделью (2)–(3), так как наряду с оценкой самой энтропии (2) и ее составляющими $H_V(\mathbf{Y})$ и $G_R(\mathbf{Y})$ необходимо оценивать энтропийные характеристики как взаимодействие между собой подсистем, так и для каждого элемента сетевой структуры.

Цель статьи — предложить дополнительный инструментарий, который бы позволил в рамках энтропийного моделирования учесть системные свойства сетевых структур для задач принятия решений.

3. Энтропийная мера взаимосвязи между несколькими подсистемами сетевой структуры

Пусть заданы n подсистем $\mathbf{Y}^{(j)}$ системы $\mathbf{Y} = (Y_1, \ldots, Y_m)$, таких что $\mathbf{Y}^{(j)} = (Y_{j,1}, \ldots Y_{j,m_j}) \subset \mathbf{Y}$, $j = 1, \ldots, n$, $n \in \{2, 3, \ldots, m\}$, любая компонента Y_i входит в состав только одной подсистемы (случайного вектора) $\mathbf{Y}^{(j)}$. Также считаем, что все компоненты $\mathbf{Y}_i$ имеют дисперсии. Определим «энтропийную меру взаимосвязи» между подсистемами (случайными векторами) $\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(n)}$ как разность между суммой энтропий подсистем и энтропией системы (объединения подсистем) $\mathbf{Y} = \bigcup_{j=1}^{n} \mathbf{Y}^{(j)} = (Y_1, \ldots, Y_m)$

$$G\left(\bigcap_{j=1}^{n} \mathbf{Y}^{(j)}\right) = \sum_{j=1}^{n} H\left(\mathbf{Y}^{(j)}\right) - H\left(\bigcup_{j=1}^{n} \mathbf{Y}^{(j)}\right). \tag{5}$$

Теорема 1. Энтропийная мера взаимосвязи между несколькими подсистемами $\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(n)}$ сетевой структуры $\mathbf{Y} = \bigcup_{j=1}^{n} \mathbf{Y}^{(j)}$ равна

$$G\left(\bigcap_{j=1}^{n} \mathbf{Y}^{(j)}\right) = -\frac{1}{2} \ln \frac{1 - d_e\left(\bigcup_{j=1}^{n} \mathbf{Y}^{(j)}\right)}{\prod_{j=1}^{n} \left(1 - d_e\left(\mathbf{Y}^{(j)}\right)\right)}, \tag{6}$$

где $d_e(\mathbf{Y}) = 1 - \prod_{k=2}^{m} \left(1 - R^2_{Y_k|Y_1 \ldots Y_{k-1}}\right)$, $d_e(\mathbf{Y}^{(j)}) = 1 - \prod_{k=2}^{m_j} \left(1 - R^2_{Y_{j,k}|Y_{j,1} \ldots Y_{j,k-1}}\right)$ *— коэффициенты тесноты корреляционной связи между компонентами случайных векторов $\mathbf{Y}$, $\mathbf{Y}^{(j)}$ соответственно.*

Доказательство теоремы 1 приведено в Приложении.

Замечание 1. Энтропийная мера взаимосвязи (6), как видно из (5), не содержит компоненты, характеризующей хаотичность подсистем. Если все элементы подсистем между собой взаимно независимы, то $G\left(\bigcap_{j=1}^{n}\mathbf{Y}^{(j)}\right)=0$. В случае, когда хотя бы два элемента разных подсистем между собой строго функционально зависимы, то $G\left(\bigcap_{j=1}^{n}\mathbf{Y}^{(j)}\right)\to+\infty$.

Замечание 2. Рассмотрим частные случаи:

1. Для $n=2$ $G\left(\mathbf{Y}^{(1)}\cap\mathbf{Y}^{(2)}\right)=-\frac{1}{2}\ln\left(1-\frac{1-d_e\left(\mathbf{Y}^{(1)}\cup\mathbf{Y}^{(2)}\right)}{\left(1-d_e(\mathbf{Y}^{(1)})\right)\left(1-d_e(\mathbf{Y}^{(2)})\right)}\right)$.

2. Пусть U и V — непрерывные случайные величины, у которых существуют дисперсии. Пусть $\mathbf{X}$ — случайный вектор, у всех компонент которого существуют дисперсии. Тогда энтропийные меры взаимосвязи между $\mathbf{X}$ и U и между U и V равны

$$G(\mathbf{X}\cap U)=1-\frac{1-d_e(\mathbf{X}\cup U)}{1-d_e(\mathbf{X})}, \tag{7}$$

$$G(U\cap V)=-\frac{1}{2}\ln\left(1-R^2_{V|U}\right)=-\frac{1}{2}\ln\left(1-R^2_{U|V}\right), \tag{8}$$

где $R_{U|V}$, $R_{V|U}$ — теоретические корреляционные отношения между U и V.

3. Если $\mathbf{Y}^{\circ}$ — гауссов случайный вектор, то

$$G\left(\bigcap_{j=1}^{n}\mathbf{Y}^{(j)\circ}\right)=-\frac{1}{2}\ln\frac{\left|\mathbf{R}_{\bigcup_{j=1}^{n}\mathbf{Y}^{(j)\circ}}\right|}{\prod_{j=1}^{n}\left|\mathbf{R}_{\mathbf{Y}^{(j)\circ}}\right|}. \tag{9}$$

4. Энтропия в отдельном узле сетевой структуры

Пусть $\mathbf{Y}=(Y_1,\ldots,Y_m)$ — сетевая структура. Определим «энтропию сетевой структуры в узле Y_l» как разницу между энтропией всей $\mathbf{Y}$ системы и энтропией системы без элемента Y_l

$$H\left(Y_l(\mathbf{Y})\right)=H(\mathbf{Y})-H\left(\mathbf{Y}\backslash Y_l\right). \tag{10}$$

Поскольку нумерация элементов в системе не влияет на ее энтропию, то без потери общности считаем, что $l=m$. Тогда выражение (10) примет вид

$$\begin{aligned}H(\mathbf{Y})-H\left(\mathbf{Y}\backslash Y_m\right)&=\sum_{i=1}^{m}\ln\sigma_{Y_i}+\sum_{i=1}^{m}\kappa_i+\frac{1}{2}\sum_{k=2}^{m}\ln\left(1-R^2_{Y_k|Y_1\ldots Y_{k-1}}\right)-\\&-\sum_{i=1}^{m-1}\ln\sigma_{Y_i}-\sum_{i=1}^{m-1}\kappa_i-\frac{1}{2}\sum_{k=2}^{m-1}\ln\left(1-R^2_{Y_k|Y_1\ldots Y_{k-1}}\right)=\\&=\underbrace{\ln\sigma_{Y_m}+\kappa_m}_{H(Y_m)}+\frac{1}{2}\ln\left(1-R^2_{Y_m|Y_1\ldots Y_{m-1}}\right).\end{aligned}$$

Поэтому для произвольного номера l элемента в сетевой структуре имеем

$$H\left(Y_l(\mathbf{Y})\right) = H\left(Y_l\right) + \frac{1}{2}\ln\left(1 - R^2_{Y_l|Y_1\ldots Y_{l-1}Y_{l+1}\ldots Y_m}\right). \tag{11}$$

С учетом (11) можно указать на теоретико-информационную интерпретацию формулы (10) как энтропию в узле сети в виде условной энтропии $H(\mathbf{Y}) - I\left(Y_l; \mathbf{Y}\backslash Y_l\right)$, где $I\left(Y_l; \mathbf{Y}\backslash Y_l\right)$ — неотрицательная средняя взаимная информация между вершиной Y_l и сетью $\mathbf{Y}\backslash Y_l$, выраженная через коэффициенты корреляции.

Таким образом, справедлива следующая

Лемма 1. Энтропия сетевой структуры $\mathbf{Y} = (Y_1, \ldots, Y_m)$ *в узле* Y_l *определяется по формуле* (11).

Замечание 3. Энтропию сетевой структуры в узле $H\left(Y_l(\mathbf{Y})\right)$ можно разложить на «энтропию хаотичности» $H\left(Y_l(\mathbf{Y})\right)_R$ и «энтропийную меру самоорганизации» $H\left(Y_l(\mathbf{Y})\right)_V$ в узле Y_l как

$$H\left(Y_l(\mathbf{Y})\right) = H_V\left(Y_l(\mathbf{Y})\right) - G_R\left(Y_l(\mathbf{Y})\right), \tag{12}$$

где $H_V\left(Y_l(\mathbf{Y})\right) = H(Y_l) = \ln\sigma_{Y_l} + \kappa_l$ — дифференциальная энтропия элемента (случайной величины) Y_l, $G_R\left(Y_l(\mathbf{Y})\right) = -\frac{1}{2}\ln\left(1 - R^2_{Y_l|Y_1\ldots Y_{l-1}Y_{l+1}\ldots Y_m}\right)$ — энтропийная мера взаимосвязи между случайной величиной Y_l и случайным вектором $\mathbf{Y}\backslash Y_l$ (при отсутствии корреляционной связи между Y_l и случайным вектором $\mathbf{Y}\backslash Y_l$ $G_R\left(Y_l(\mathbf{Y})\right) = 0$, при строгой функциональной зависимости между Y_l и хотя бы двумя элементами случайного вектора $\mathbf{Y}\backslash Y_l$ $G_R\left(Y_l(\mathbf{Y})\right) \to +\infty$).

Замечание 4. Если $\mathbf{Y}^\circ$ — гауссов случайный вектор, то

$$H\left(Y_l^\circ(\mathbf{Y}^\circ)\right) = \ln\sigma_{Y_l^\circ} + \ln\sqrt{2\pi e} + \frac{1}{2}\ln\frac{|\mathbf{R}_{\mathbf{Y}^\circ}|}{\left|\mathbf{R}_{\mathbf{Y}^\circ\backslash Y_l^\circ}\right|}. \tag{13}$$

5. Примеры реализации. Обсуждение результатов

Выражения (5)–(13) позволят исследовать сетевые структуры: оценивать взаимосвязанность меду собой различных участков, а также оценивать, как меняется энтропия внутри таких систем. Рассмотрим примеры. Для упрощения интерпретации результатов рассмотрим гауссовы сетевые структуры.

Пример 1. Пусть дана гауссова система $\mathbf{X}$, граф которой приведен на рис. 1. Она состоит из двух подсистем $\mathbf{X}^{(1)} = (X_1, X_2, X_3, X_4, X_5)$ и $\mathbf{X}^{(2)} = (X_6, X_7, X_8, X_9, X_{10})$.

В табл. 1, 2 приведены значения среднеквадратических отклонений и корреляционной матрицы случайного вектора $\mathbf{X}$.

Выполним вспомогательные расчеты: $|\mathbf{R}_{\mathbf{X}^{(1)}}| = 0{,}30920$, $|\mathbf{R}_{\mathbf{X}^{(2)}}| = 0{,}017723$, $|\mathbf{R}_{\mathbf{X}^{(1)}\cup\mathbf{X}^{(2)}}| = |\mathbf{R}_{\mathbf{X}}| = 2{,}6472 \cdot 10^{-6}$. Отсюда энтропийная мера взаимосвязи $H\left(\mathbf{X}^{(1)} \cap \mathbf{X}^{(2)}\right) = 3{,}82$.

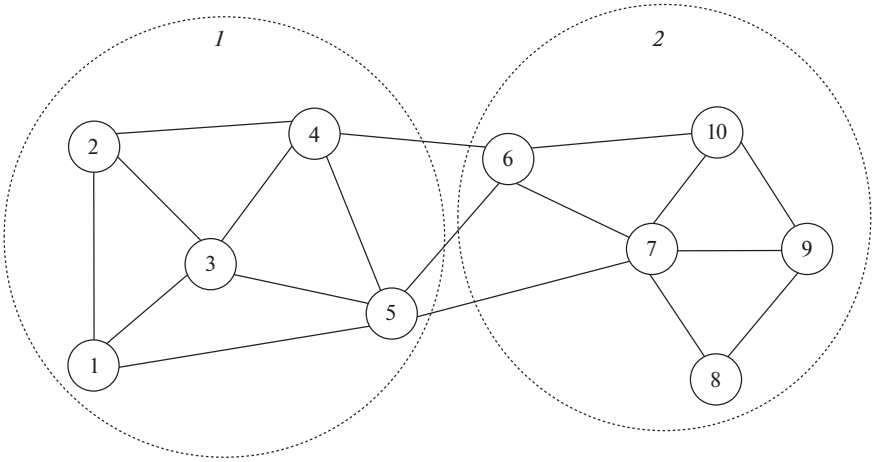


Рис. 1. Граф сетевой структуры **X**, состоящей из двух подсистем.

Выполнив расчеты по формулам (12), (13), получим значения энтропий в узлах сетевой структуры **X**, которые приведены в табл. 3. Видим, что наибольшие значения энтропийной меры самоорганизации находятся в вершинах X_5, X_6, X_7, которые расположены на границе между подсистемами. Самые высокие значения энтропии хаотичности наблюдаются в вершинах, имеющих наибольшие дисперсии.

Таблица 1. Среднеквадратические отклонения элементов случайного вектора **X**

X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
1,53	1,08	2,00	1,23	1,21	1,84	1,34	1,78	1,00	1,54

Таблица 2. Корреляционная матрица случайного вектора **X**

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
X_1	1,00	0,41	–0,13	0,21	0,04	0,05	–0,01	–0,04	–0,05	0,07
X_2	0,41	1,00	0,08	0,35	0,00	–0,04	–0,01	0,03	0,00	–0,01
X_3	–0,13	0,08	1,00	0,60	0,41	0,23	–0,12	0,04	0,00	–0,01
X_4	0,21	0,35	0,60	1,00	0,46	0,58	0,02	0,05	0,01	0,14
X_5	0,04	0,00	0,41	0,46	1,00	0,40	–0,34	0,01	0,02	–0,02
X_6	0,05	–0,04	0,23	0,58	0,40	1,00	0,59	0,18	0,02	0,63
X_7	–0,01	–0,01	–0,12	0,02	–0,34	0,59	1,00	0,59	0,33	0,45
X_8	–0,04	0,03	0,04	0,05	0,01	0,18	0,59	1,00	0,74	–0,28
X_9	–0,05	0,00	0,00	0,01	0,02	0,02	0,33	0,74	1,00	–0,58
X_{10}	0,07	–0,01	–0,01	0,14	–0,02	0,63	0,45	–0,28	–0,58	1,00

Таблица 3. Значения энтропий в вершинах сетевой структуры **X**

X_l	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
$H(X_l(\mathbf{X}))$	1,17	–0,35	0,21	–1,34	–1,72	–1,74	–2,18	–1,02	–0,11	0,08
$H_V(X_l(\mathbf{X}))$	1,84	1,50	**2,11**	1,63	1,61	**2,03**	1,71	**2,00**	1,42	1,85
$G_R(X_l(\mathbf{X}))$	0,67	1,85	1,90	2,97	**3,33**	**3,76**	**3,89**	3,02	1,53	1,77

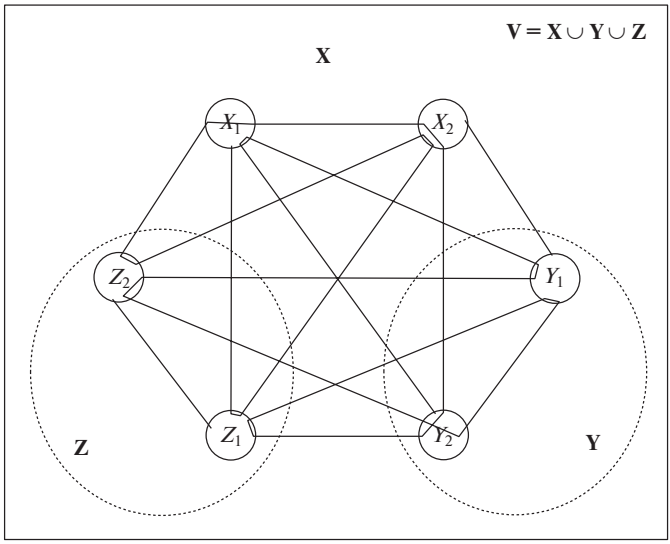


Рис. 2. Граф гауссовой системы $\mathbf{V}$, состоящей из трех подсистем.

Пример 2. Имеем гауссову систему $\mathbf{V}$, состоящую из подсистем $\mathbf{X} = (X_1, X_2)$, $\mathbf{Y} = (Y_1, Y_2)$, $\mathbf{Z} = (Z_1, Z_2)$, т.е.

$$\mathbf{V} = \mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z} = (X_1, X_2, Y_1, Y_2, Z_1, Z_2).$$

Граф системы приведен на рис. 2.

Зададим дисперсии всех шести элементов равными 1. Корреляционная матрица системы $\mathbf{V}$ приведена в табл. 4.

Найдем по формулам (12), (13) энтропию во всех шести узлах системы. Результаты расчетов приведены в табл. 5. Самая высокая энтропия наблюдается в узлах Y_1 и Z_1, а самая низкая — в вершинах X_2 и Z_2.

Изменим значения парных корреляций.

Случай 1. Коэффициенты парной линейной корреляции приведены в табл. 4.

Таблица 4. Исходная корреляционная матрица

Элемент	X_1	X_2	Y_1	Y_2	Z_1	Z_2
X_1	1	0,5	0,5	0,7	0,3	0,5
X_2	0,5	1	0,5	0,5	0,2	–0,3
Y_1	0,5	0,5	1	0,7	0,6	0,3
Y_2	0,7	0,5	0,7	1	0,5	0,5
Z_1	0,3	0,2	0,6	0,5	1	0,5
Z_2	0,5	–0,3	0,3	0,5	0,5	1

Таблица 5. Значения энтропии в вершинах системы $\mathbf{V}$

Вершина	X_1	X_2	Y_1	Y_2	Z_1	Z_2
Энтропия в узле	0,66	0,47	0,95	0,72	0,97	0,40

Таблица 6. Значения энтропийной меры взаимосвязи между подсистемами **X**, **Y**, **Z**

Случаи	1	2	3
Энтропия взаимосвязи	1,52	1,76	1,68

Случай 2. Значение коэффициента парной линейной корреляции между элементами Y_1 и Z_2 изменим с 0,3 на 0,45.

Случай 3. Значение коэффициента парной линейной корреляции между элементами Z_1 и Z_2 изменим с 0,5 на 0,1.

Результаты расчета энтропийной меры взаимосвязи между подсистемами **X**, **Y**, **Z** для трех случаев приведены в табл. 6.

Результаты в табл. 6 говорят о следующем:

1. Увеличение тесноты корреляции между элементами разных подсистем приводит к росту тесноты взаимосвязи в целом между подсистемами.

2. Уменьшение тесноты корреляции между элементами в отдельной подсистеме приводит к росту тесноты взаимосвязи в целом между подсистемами.

6. Заключение

В рамках векторной энтропийной модели введены новые энтропийные характеристики — энтропийная мера взаимосвязи между несколькими подсистемами и энтропия системы в отдельном узле. Эти характеристики расширяют возможности исследования сетевых структур: позволят оценивать взаимосвязанность между собой различных участков, а также определять, как меняется энтропия внутри таких систем.

Предложенный подход рассчитан на сетевые структуры, представимые в виде корреляционных графов. Например, он может применяться для исследования городских систем, транспортных систем, систем связи, систем промышленной безопасности и т.д. Представляет интерес связать векторное энтропийное моделирование с методами риск-анализа. Из-за ограниченности объема статьи был рассмотрен частный случай гауссовой сетевой структуры. Для него определение введенных энтропийных характеристик сводится к определению средних квадратических отклонений компонент системы и определителей корреляционных матриц подсистем и всей системы (сетевой структуры). В дальнейшем планируется рассмотреть применение метода для практических задач.

ПРИЛОЖЕНИЕ

Доказательство теоремы 1.

Зададим n непрерывных случайных векторов произвольных размерностей, у которых нет совпадающих компонент. Обозначим их как $\mathbf{Y}^{(1)} = \left(Y_1^{(1)}, \ldots, Y_{m_1}^{(1)}\right), \mathbf{Y}^{(2)} = \left(Y_1^{(2)}, \ldots, Y_{m_2}^{(2)}\right), \ldots, \mathbf{Y}^{(n)} = \left(Y_1^{(n)}, \ldots, Y_{m_n}^{(n)}\right)$. Компоненты в любом из случайных векторов $\mathbf{Y}^{(l)}$ могут быть корреляционно взаимно зависимыми. Для этого множества случайных векторов введем их

объединение в виде вектора

$$\text{(П.1)}\quad \mathbf{Y} = \bigcup_{j=1}^{n} \mathbf{Y}^{(j)} =$$

$$= (\underbrace{Y_1, \ldots, Y_{m_1}}_{Y^{(1)}}, \underbrace{\mathbf{Y}_{m_1+1}, \ldots, Y_{m_1+m_2}}_{\mathbf{Y}^{(2)}}, \ldots, \underbrace{Y_{m_1+\ldots+m_{n-1}+1}, \ldots, Y_{m_1+\ldots+m_n}}_{\mathbf{Y}^{(n)}}).$$

Поскольку

$$H\left(\bigcup_{j=1}^{n} \mathbf{Y}^{(j)}\right) = H(\mathbf{Y}),$$

то

$$\sum_{j=1}^{n} H_V\left(\mathbf{Y}^{(j)}\right) = H_V\left(\bigcup_{j=1}^{n} \mathbf{Y}^{(j)}\right) = H_V(\mathbf{Y}).$$

Поэтому формула (5) с учетом (3) примет вид

$$\text{(П.2)}\qquad G\left(\bigcap_{j=1}^{n} \mathbf{Y}^{(j)}\right) = G_R(\mathbf{Y}) - \sum_{j=1}^{n} G_R\left(\mathbf{Y}^{(j)}\right).$$

Пусть $M_j = \sum\limits_{k=1}^{j} m_k,\ j = 1, \ldots, n,\ M_0 = 0,\ M_n = m_1 + \ldots + m_n = m$. В [17] показано, что

$$\text{(П.3)}\qquad G_R\left(\mathbf{Y}^{(j)}\right) = -\frac{1}{2}\sum_{k=2}^{m_j} \ln\left(1 - R^2_{Y_k^{(j)}|Y_1^{(j)}\ldots Y_{k-1}^{(j)}}\right),$$

$$\text{(П.4)}\qquad G_R(\mathbf{Y}) = -\frac{1}{2}\sum_{k=2}^{M_n} \ln\left(1 - R^2_{Y_k|Y_1\ldots Y_{k-1}}\right)$$

Очевидно, что если $m_j = 1$, то $G_R\left(\mathbf{Y}^{(j)}\right) = 0$. С учетом (П.1)–(П.4) имеем

$$e^{2G\left(\bigcap_{j=1}^{n} \mathbf{Y}^{(j)}\right)} = \frac{e^{-2\sum_{j=1}^{n} G_R(\mathbf{Y}^{(j)})}}{e^{-2G_R(\mathbf{Y})}} = \frac{\prod\limits_{j=1}^{n} \prod\limits_{k=M_{j-1}+2}^{M_j} \left(1 - R^2_{Y_k^{(j)}|Y_1^{(j)}\ldots Y_{k-1}^{(j)}}\right)}{\prod\limits_{k=2}^{M_n}\left(1 - R^2_{Y_k|Y_1\ldots Y_{k-1}}\right)} =$$

$$= \prod_{j=1}^{n-1}\left(\left(1 - R^2_{Y_{M_j+1}|Y_1\ldots Y_{M_j}}\right) \prod_{k=M_j+2}^{M_{j+1}} \frac{1 - R^2_{Y_k|Y_1\ldots Y_{k-1}}}{1 - R^2_{Y_k|Y_{M_j+1}Y_{M_j+2}\ldots Y_{k-1}}}\right)$$

Далее умножим слева и справа последнее выражение на

$$\prod_{j=1}^{n-1}\prod_{k=M_j+2}^{M_{j+1}}\left(1-R^2_{Y_k|Y_{M_j+1}Y_{M_j+2}\ldots Y_{k-1}}\right)$$

и учтем, что $\forall j \quad 1-d_e\left(\mathbf{Y}^{(j)}\right)=\prod_{k=M_{j-1}+2}^{M_j}\left(1-R^2_{Y_k|Y_{M_{j-1}+1}Y_{M_{j-1}+2}\ldots Y_{k-1}}\right)$:

$$e^{2G\left(\bigcap_{j=1}^{n}\mathbf{Y}^{(j)}\right)}\prod_{j=2}^{n}\left(1-d_e(Y^{(j)})\right)=$$

$$=\prod_{j=1}^{n-1}\left(1-R^2_{Y_{M_j+1}|Y_1\ldots Y_{M_j}}\right)\prod_{k=M_j+2}^{M_{j+1}}\left(1-R^2_{Y_k|Y_{M_j+1}Y_{M_j+2}\ldots Y_{k-1}}\right)$$

Умножив и разделив последнее выражение справа на

$$\prod_{k=2}^{M_1}\left(1-R^2_{Y_k|Y_1\ldots Y_{k-1}}\right)$$

и учтя, что $1-d_e\left(\mathbf{Y}^{(1)}\right)=\prod_{k=2}^{M_1}\left(1-R^2_{Y_k|Y_1\ldots Y_{k-1}}\right)$, получим

$$e^{2G\left(\bigcap_{j=1}^{n}\mathbf{Y}^{(j)}\right)}\prod_{j=1}^{n}\left(1-d_e\left(\mathbf{Y}^{(j)}\right)\right)=\prod_{k=1}^{m}\left(1-R^2_{Y_k|Y_1\ldots Y_{k-1}}\right)=$$

$$=1-d_e\left(\bigcup_{j=1}^{n}\mathbf{Y}^{(j)}\right).$$

Следовательно,

$$G\left(\bigcap_{j=1}^{n}\mathbf{Y}^{(j)}\right)=-\frac{1}{2}\ln\frac{1-d_e\left(\bigcup_{j=1}^{n}\mathbf{Y}^{(j)}\right)}{\prod_{j=1}^{n}\left(1-d_e(\mathbf{Y}^{(j)})\right)}$$

что и требовалось доказать.

СПИСОК ЛИТЕРАТУРЫ

1. Современный философский словарь: 2-е изд. / Под общей ред. проф. В.Е. Кемерова. Лондон, Франкфурт-на-Майне, Париж, Люксембург, Москва, Минск: ПАНПРИНТ, 1998.

2. *Лысак И.В., Косенчук Л.Ф.* Современное общество как общество сетевых структур // Информационное общество. 2015. № 2–3. С. 45–51.
3. *Новиков Д.А.* Сетевые структуры и организационные системы. М.: ИПУ РАН, 2003.
4. *Goldstein H.* Multilevel Statistical Models: 4th ed. Wiley, 2011.
5. *Pardoe I.* Applied Regression Modeling: 2nd ed. Wiley, 2012.
6. *Lanchier N.* Stochastic Modeling. Springer, 2017.
7. *Булатов В.В.* Введение в математические методы моделирования сложных систем. М.: ОнтоПринт, 2018.
8. *Frank S.A., Smith D.E.* Measurement Invariance, Entropy, and Probability // Entropy. 2010. V. 12. No. 3. P. 289–303.
9. *Wilson A.G.* Entropy in Urban and Regional Modelling: Retrospect and Prospect // Geographical Analysis. 2010. V. 42. No. 4. P. 364–394.
10. *Czyz T., Hauke J.* Entropy in Regional Analysis // Quaestiones Geographicae. 2015. V. 34. No. 4. P. 69–78.
11. *Попков Ю.С., Дубнов Ю.А., Попков А.Ю.* Энтропийная редукция размерности в задачах рандомизированного машинного обучения // АиТ. 2018. № 11. С. 106–122.

    Popkov Y.S., Dubnov Y.A., Popkov A.Y. Entropy Dimension Reduction Method for Randomized Machine Learning Problems // Autom. Remote Control. 2018. V. 79. No. 11. P. 2038–2051.
12. *Shannon C.E.* A Mathematical Theory of Communication // The CityplaceBell System Technical Journal. 1948. V. 27. No. 3. P. 379–423; No. 4. P. 623–656.
13. *Гельфанд И.М., Колмогоров А.Н., Яглом А.М.* Количество информации и энтропия для непрерывных распределений / Труды III Всесоюзного математического съезда. М.: АН СССР, 1958. Т. 3. С. 300–320.
14. *Тырсин А.Н.* Энтропийное моделирование многомерных стохастических систем. Воронеж: Научная книга, 2016.
15. *Хардле В.* Прикладная непараметрическая регрессия: Пер. с англ. М.: Мир, 1993.
16. *Галлагер Р.* Теория информации и надежная связь: Пер. с англ. М.: Советское радио, 1974.
17. *Тырсин А.Н.* Скалярная мера взаимозависимости между случайными векторами // Зав. лаборатория. Диагностика материалов. 2018. Т. 84. № 7. С. 76–82. https://doi.org/10.26896/1028-6861-2018-84-7-76-82.

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 23.01.2022

После доработки 11.06.2022

Принята к публикации 29.06.2022

© 2022 г. З.М. ШИБЗУХОВ, д-р физ.-мат. наук (intellimath@mail.ru)
(Институт математики и информатики Московского педагогического государственного университета; Московский физико-технический институт)

ОБ ОДНОЙ РОБАСТНОЙ СХЕМЕ ГРАДИЕНТНОГО БУСТИНГА НА ОСНОВЕ АГРЕГИРУЮЩИХ ФУНКЦИЙ, НЕЧУВСТВИТЕЛЬНЫХ К ВЫБРОСАМ[1]

Предложена одна новая робастная схема построения алгоритмов градиентного бустинга. Она основана на применении дифференцируемых оценок среднего значения, нечувствительных или малочувствительных к выбросам, при построении робастного функционала эмпирического риска. Это позволило применить метод итеративного перевозвешивания для для поиска очередной базовой функции и ее веса. Такая процедура градиентного бустинга позволяет находить искомую зависимость по данным, которые содержат относительно большую долю выбросов.

Ключевые слова: градиентный бустинг, робастная оценка, регрессия, классификация.

DOI: 10.31857/S0005231022100142, **EDN:** ALQEEL

1. Введение

Методы бустинга [1] являются разновидностью методов машинного обучения для построения *ансамблей базовых алгоритмов*. Модель базовых алгоритмов позволяет строить *слабые алгоритмы*, которые имеют относительно небольшую сложность и заведомо не являются переобученными. Модель базовых алгоритмов также может позволять строить *сложные алгоритмы* с высокими показателями качества, но склонные к переобучению. В таких случаях в методах бустинга они, как правило, используются с ограничениями на сложность, которые позволяют исключить переобучение базовых алгоритмов, но в то же время делают их более слабыми. Целевой алгоритм, как правило, строится в виде линейной комбинации базовых алгоритмов. Такой подход к построению алгоритмов машинного обучения позволяет строить сильные алгоритмы машинного обучения из более слабых алгоритмов.

Метод *градиентного бустинга* направлен на решение задачи построения линейной композиции некоторого заранее неизвестного количества базовых алгоритмов, которые минимизируют оценку эмпирического риска на обучающем множестве примеров. В классической схеме построения алгоритмов машинного обучения для решения задач регрессии и классификации эмпириче-

[1] Работа выполнена при поддержке научного проекта № АААА-А20-120122190034-9 Московского педагогического государственного университета.

ский риск оценивается как среднее арифметическое от потерь:

$$\mathcal{Q}(w) = \frac{1}{N}\sum_{k=1}^{N}\ell(f(\tilde{x}_k; w), \tilde{y}_k), \tag{1}$$

где $f(x; w)$ — параметризованная зависимость, $\{\tilde{x}_1, \ldots, \tilde{x}_N\} \subset \mathbb{R}^n$ — обучающие входы, $\{\tilde{y}_1, \ldots, \tilde{y}_N\}$ — ожидаемые значения на выходе, $\ell(y, \tilde{y})$ – неотрицательная дифференцируемая функция потерь. Например:

1) в задаче регрессии $\ell(y, \tilde{y}) = \varrho(y - \tilde{y})$, где $\varrho(r)$ — квазивыпуклая функция с минимумом в нуле, например $\varrho(r) = r^2$;

2) в задаче классификации для двух классов $\ell(y, \tilde{y}) = \varrho(1 - \tilde{y}y)$, где $\varrho(r)$ — монотонно убывающая функция, строго положительная при $r < 0$ и стремящаяся к нулю при $r \to +\infty$, например, $\varrho(r) = \max(0, 1 - \tilde{y}y)$ (функция Хинжа).

Требуется найти

$$w^\star = \arg\min_w \mathcal{Q}(w).$$

Для повышения робастности ранее предлагалось использовать более робастные функции потерь [2]. Например, в задаче регрессии:

1) $\varrho(r) = \sqrt{\varepsilon^2 + r^2} - \varepsilon$ ($\varrho'(r)$ ограничена);

2) $\varrho(r) = \ln(a^2 + r^2) - 2\ln a$ ($\varrho'(r) \to 0$ при $r \to \pm\infty$);

3) $\varrho(r) = |r| / \sqrt{\varepsilon^2 + r^2}$ ($\varrho(r)$ — ограничена),

а в задаче классификации:

1) $\varrho(r) = \ln(1 + \max(0, 1 - r))$ ($\varrho'(r) \to 0$ при $r \to \pm\infty$);

2) $\varrho(r) = \eta(\max(0, 1 - r))$, где $\eta(s)$ монотонно возрастающая ограниченная функция при $s > 0$.

Для поиска $w^\star$, которая минимизирует $\mathcal{Q}(w)$ с более "робастными" функциями потерь, применяется *метод итеративного перевзвешивания* [3, 4]. Например,

1) в случае робастной регрессии решение задачи

$$w^\star = \arg\min_w \frac{1}{N}\sum_{k=1}^{N}\varrho\left(f(\tilde{x}_k; w) - \tilde{y}_k\right)$$

сводится к решению цепочки задач:

$$w^{t+1} = \arg\min \sum_{k=1}^{N} v_k^t \left(f(\tilde{x}_k; w) - \tilde{y}_k\right)^2, \tag{2}$$

где

$$v_k^t = \varphi\left(f(\tilde{x}_k; w^t) - \tilde{y}_k\right), \quad \varphi(r) = \varrho'(r)/r;$$

2) в случае задачи классификации решение задачи

$$w^\star = \arg\min_w \frac{1}{N}\sum_{k=1}^{N} \varrho\left(\max\left(0, 1 - \tilde{y}_k f(\tilde{x}_k; w)\right)\right)$$

сводится к решению цепочки задач:

$$w^{t+1} = \arg\min \sum_{k=1}^{N} v_k^t \max\left(0, 1 - \tilde{y}_k f(\tilde{x}_k; w)\right), \tag{3}$$

где

$$v_k^t = \varphi\left(1 - \tilde{y}_k f(\tilde{x}_k; w^t)\right), \quad \varphi(r) = \varrho'(r)/r \ \text{ при } \ r < 0 \ \text{ и } \ \varphi(r) = 0 \ \text{ при } \ r \geqslant 0.$$

Здесь на каждом шаге процедуры итерационного перевзвешивания минимизируется взвешенная сумма квадратов ошибки (в задаче регрессии) или взвешенная сумма отступов с обратным знаком (в задаче классификации). Подобные схемы хорошо известны. Однако если обучающие данные содержат выбросы, из-за которых распределение значений потерь неизбежно будет содержать выбросы, то такой подход сталкивается с трудностями из-за неустойчивости среднего арифметического. Поэтому для преодоления этой проблемы было предложено использовать оценки среднего значения, которые нечувствительны или малочувствительны к выбросам [4, 5]. В этом случае робастная оценка средних потерь имеет вид

$$\mathcal{Q}(w) = \mathsf{M}\left\{\ell(f(\tilde{x}_1; w), \tilde{y}_1), \ldots, \ell(f(\tilde{x}_N; w), \tilde{y}_N)\right\},$$

где $\mathsf{M}\{z_1, \ldots, z_N\}$ — усредняющая агрегирующая функция. В [6, 7] было предложено использовать дифференцируемые оценки среднего, которые являются сглаженными вариантами известных робастных оценок среднего — медианы, α-квантиля и винзоризированного среднего арифметического. Это позволяет тоже применить метод итеративного перевзвешивания, но с другой схемой пересчета весов в (2) и (3). В настоящей работе эта робастная схема распространяется на метод градиентного бустинга. Далее сначала опишем классическую схему градиентного бустинга, а затем — робастную.

2. Классическая схема градиентного бустинга

Классическую схему метода *градиентного бустинга* [8] можно описать следующим образом. Рассмотрим класс функций $L(\mathcal{H})$, состоящий из линейных комбинаций *базовых функций* из некоторого класса функций $\mathcal{H}$

$$H(x) = \sum_{j=1}^{p} \alpha_j h_j(x),$$

где $\alpha_j \in \mathbb{R}$, $h_j \in \mathcal{H}$, $x \in \mathbb{R}^n$.

В классе $L(\mathcal{H})$ ищется оптимальная функция H^*, которая доставляет минимум

$$H^* = \arg \min_{H \in L(\mathcal{H})} \mathcal{Q}(H)$$

функционалу $\mathcal{Q}(H)$:

$$\mathcal{Q}_V(H) = \sum_{k=1}^{N} v_k \ell(H(\tilde{x}_k), \tilde{y}_k), \tag{4}$$

где $V = \{v_k \colon k = 1, \ldots, N\}$, $v_k \geqslant 0$ — веса примеров, такие что $v_1 + \cdots + v_N = 1$. Например, $v_k = 1/N$.

Для произвольных $\alpha \in \mathbb{R}$ и $h \in \mathcal{H}$ рассматривается функционал

$$\mathcal{Q}_V\big(h, \alpha\big) = \mathcal{Q}_V\big(H + \alpha h\big) = \sum_{k=1}^{N} v_k \ell(\tilde{H}_k + \alpha h(\tilde{x}_k), \tilde{y}_k), \tag{5}$$

где $\tilde{H}_k = H(\tilde{x}_k)$.

Функция h и параметр α в (5) выбираются в результате решения задачи минимизации:

$$h^\star, \alpha^\star = \arg \min_{h, \alpha} \mathcal{Q}_V(h, \alpha). \tag{6}$$

Для поиска минимума $\mathcal{Q}_V(h, \alpha)$ можно применить процедуру поиска h и α из известных алгоритмов градиентного бустинга, которые основаны на минимизации взвешенной суммы потерь. Для нахождения экстремума $\mathcal{Q}_V$ будем применять итеративный метод *поочередной минимизации* (*alternating minimization*) [9]

$$\begin{aligned} h^{p+1} &= \arg \min_{h} \mathcal{Q}_V(h, \alpha^p) \\ \alpha^{p+1} &= \arg \min_{\alpha} \mathcal{Q}_V(h^{p+1}, \alpha). \end{aligned} \tag{7}$$

На каждом шаге итерации сначала решается первая задача для поиска h^{p+1}, а затем вторая задача для поиска α^{p+1}. Итерационный процесс завершается, если $\big|\mathcal{Q}(h^{p+1}, \alpha^{p+1}) - \mathcal{Q}(h^p, \alpha^p)\big| < \varepsilon$ для заданного $\varepsilon > 0$, или если $t = t_{\max}$, где $t_{\max}$ — максимальное число шагов итерации. Для упрощения вычислений иногда в алгоритмах градиентного бустинга выполняется *только один* шаг метода (7). Практика также показала, что достаточно использовать небольшое число таких шагов. В некоторых случаях α^{p+1} можно вычислить явно (опираясь на необходимое условие экстремума $\mathcal{Q}_V$ по α), например

1) для задачи регрессии с $\ell(y, \tilde{y}) = \frac{1}{2}(y - \tilde{y})^2$ следующим образом:

$$\alpha^{p+1} = \frac{\sum\limits_{k=1}^{N} v_k (H_k - \tilde{y}_k) h^{p+1}(\tilde{x}_k)}{\sum\limits_{k=1}^{N} v_k \left(h^{p+1}(\tilde{x}_k)\right)^2};$$

2) для задачи классификации с $\ell(y, \tilde{y}) = \max(0, 1-\tilde{y}y)$ следующим образом:

$$\alpha^{p+1} = \frac{\sum\limits_{k \in I_p} \left\lfloor \tilde{v}_k (1 - \tilde{y}_k H_k) \tilde{y}_k h^{p+1}(\tilde{x}_k) \right\rfloor}{\sum\limits_{k \in I_p} \left\lfloor \tilde{v}_k \left(\tilde{y}_k h^{p+1}(\tilde{x}_k)\right)^2 \right\rfloor},$$

где

$$\tilde{v_k} = \frac{v_k}{1 - \tilde{y}_k H_k - \alpha^p \tilde{y}_k h^{p+1}(\tilde{x}_k)},$$

а

$$I_p = \left\lfloor \left\{k \colon 1 - \tilde{y}_k H_k - \alpha^p \tilde{y}_k h^{p+1}(\tilde{x}_k) > 0\right\} \right\rfloor$$

В целом алгоритм градиентного бустинга можно выразить при помощи следующего псевдокода:

def gb_fit(M,V):
| $H_0 = 0$
| **for** j **in** $[1, \ldots, M]$:
| | $h_j, \alpha_j = \arg\min\limits_{h,\alpha} \mathcal{Q}_V\big(H_{j-1} + \alpha h\big)$.
| | $H_j = H_{j-1} + \alpha_j h_j(x)$
| **return** H_M

3. Робастная схема градиентного бустинга

Эмпирическое распределение значений

$$\left\{ \left\lfloor z_k = z_k(h, \alpha) = \ell(\tilde{H}_k + \alpha h(\tilde{x}_k), \tilde{y}_k) \colon k = 1, \ldots, N \right\} \right\rfloor$$

может содержать выбросы из-за искажений в данных или неадекватности части данных по отношению к выбранной модели зависимости, особенно на начальной стадии градиентного бустинга. Так как среднее арифметическое чувствительно к выбросам, то в результате минимизации (5), как правило, получаются искаженные h и α.

Проблему выбросов можно было бы решить путем подбора набора весов $v_1, \ldots, v_N$, так чтобы для индексов k, соответствующих выбросам, значения v_k были достаточно малы, чтобы невелировать их влияние. Однако задача поиска таких значений весов по сложности сопоставима с задачей идентификации выбросов. Ниже сформулируем подход, который может позволить преодолеть влияние выбросов, а также найти соответствующие значения весов $v_1, \ldots, v_N$.

Для этого сформулируем более робастную постановку задачи:

$$h^\star, \alpha^\star = \arg\min_{h,\alpha} \mathcal{Q}_{\mathsf{M}}(h, \alpha), \tag{8}$$

где

$$\mathcal{Q}_{\mathsf{M}}(h,\alpha) = \mathsf{M}\{z_1(h,\alpha),\ldots,z_N(h,\alpha)\},$$

где $\mathsf{M}\{z_1,\ldots,z_N\}$ — дифференцируемая усредняющая агрегирующая функция, более устойчивая к выбросам в данных [10].

Необходимое условие экстремума дает систему уравнений

$$\sum_{k=1}^{N} v_k(h,\alpha)\nabla_{h,\alpha}\ell(\tilde{H}_k + \alpha h(\tilde{x}_k),\tilde{y}_k) = 0,$$

где

$$\nu_k(h,\alpha) = \frac{\partial \mathsf{M}\{z_1(h,\alpha),\ldots,z_N(h,\alpha)\}}{\partial z_k}. \tag{9}$$

Дифференцируемые усредняющие агрегирующие функции $\mathsf{M}\{z_1,\ldots,z_N\}$, по построению, такие что $\partial\mathsf{M}/\partial z_k \geqslant 0$ для всех $k = 1,\ldots,N$ и

$$\partial\mathsf{M}/\partial z_1 + \cdots + \partial\mathsf{M}/\partial z_N = 1.$$

Для поиска оптимальных значений $h^\star$ и $\alpha^\star$ (решения задачи (8)) будем применять процедуру итеративного перевзвешивания, следуя [11]:

$$h^t,\alpha^t = \arg\min_{h,\alpha}\sum_{k=1}^{N}\nu_k\left(h^{t-1},\alpha^{t-1}\right)\ell(\tilde{H}_k + \alpha h(\tilde{x}_k),\tilde{y}_k). \tag{10}$$

Данная схема итеративного перевзвешивания возникает в результате применения общего метода Якоби для решения системы нелинейных уравнений

$$\begin{cases} v_k = \dfrac{\partial \mathsf{M}\{z_1(h,\alpha),\ldots,z_N(h,\alpha)\}}{\partial z_k} \\ \displaystyle\sum_{k=1}^{N} v_k\nabla_{h,\alpha}\ell(\tilde{H}_k + \alpha h(\tilde{x}_k),\tilde{y}_k) = 0, \end{cases}$$

которая возникает из необходимого условия экстремума для (8).

В этой итеративной схеме на шаге t осуществляется минимизация взвешенной суммы потерь

$$\mathcal{Q}^t_{V_t}(h,\alpha) = \sum_{k=1}^{N} v_k^t\ell(\tilde{H}_k + \alpha h(\tilde{x}_k),\tilde{y}_k),$$

где

$$V_t = \left\{v_k^t = \nu_k(h^{t-1},\alpha^{t-1})\colon k = 1,\ldots,N\right\}.$$

def gb_fit_step_M(H, $t_{\max}$):
| инициализация h^0, α^0
| $\tilde{H}_k = H(\tilde{x}_k)$, $k = 1, \ldots, N$
| **for** $t = 1, \ldots, t_{\max}$:
| | $h^t, \alpha^t = \arg\min\limits_{h,\alpha} \mathcal{Q}^t_{V_t}(h, \alpha)$.
| | **if** выполнено условие останова:
| | | **break**
| **return** h^t, α^t

def gb_fit_M(M):
| $H_0 = 0$
| **for** $j = 1, \ldots, M$:
| | $h_j, \alpha_j =$ gb_fit_step_M(H_{j-1}, $t_{\max}$)
| | $H_j = H_{j-1} + \alpha_j h_j(x)$
| **return** H_M

Для поиска решения задачи минимизации $\mathcal{Q}^t_{V_t}(h, \alpha)$ будем применять процедуру *альтернативной минимизации* (*alternating minimization*) [9]

$$h^t_p = \arg\min_h \sum_{k=1}^{N} \Big[v^t_k \ell(\tilde{H}_k + \alpha^{t-1}_{p-1} h(\tilde{x}_k), \tilde{y}_k)$$

$$\alpha^t_p = \arg\min_\alpha \sum_{k=1}^{N} \Big[v^t_k \ell(\tilde{H}_k + \alpha h^t_p(\tilde{x}_k), \tilde{y}_k),$$

где $h^{t-1}_0 = h^{t-1}$, $\alpha^{t-1}_0 = \alpha^{t-1}$.

Для решения приведенных задач минимизации использовался метод градиентного спуска с применением схемы ADAM [12].

Рассмотрим отдельно некоторые варианты реализации метода робастного градиентного бустинга для задачи регрессии и задачи классификации, которые можно получить в рамках предложенной выше схемы.

3.1. Задача регрессии

В задаче регрессии функция потерь, как правило, имеет вид: $\ell(y, \tilde{y}) = \varrho(y - \tilde{y})$, где ϱ — неотрицательная дифференцируемая квазивыпуклая унимодальная функция, $0 \in \arg\min \varrho(r)$.

Итерационная схема (10) принимает вид:

$$h^t, \alpha^t = \arg\min_{h,\alpha} \sum_{k=1}^{N} \nu_k(h^{t-1}, \alpha^{t-1}) \varrho\big(\Big[\tilde{H}_k - \tilde{y}_k + \alpha h(\tilde{x}_k)\big)\big),$$

где $\tilde{H}_k - \tilde{y}_k + \alpha h(\tilde{x}_k)$ — величина ошибки для k-го прецедента.

Типичный пример $\varrho(r) = r^2$. В рамках классической схемы построения робастной регрессии [13] можно построить следующую процедуру итеративного перевзвешивания:

$$h^t = \arg\min_{h\in\mathcal{H}} \sum_{k=1}^{N} \Big[v_k(h^{t-1}, \alpha^{t-1}) \Big(\tilde{H}_k - \tilde{y}_k + \alpha^{t-1} h(\tilde{x}_k) \Big)^2$$

$$\alpha^t = \arg\min_{\alpha} \sum_{k=1}^{N} \Big[v_k(h^{t-1}, \alpha^{t-1}) \Big(\tilde{H}_k - \tilde{y}_k + \alpha h^t(\tilde{x}_k) \Big)^2,$$

где $v_k(h,\alpha) = \nu_k(h,\alpha)\varphi\big(\tilde{H}_k - \tilde{y}_k + \alpha h(\tilde{x}_k)\big)$, $\varphi(r) = \varrho'(r)/r$, $\nu_k(h,\alpha)$ вычисляется по формуле (9).

Величину α^t в данной схеме можно вычислить явно

$$\alpha^t = \frac{\sum\limits_{k=1}^{N} \Big[v_k(h^{t-1}, \alpha^{t-1})(\tilde{y}_k - \tilde{H}_k) h^t(\tilde{x}_k)}{\sum\limits_{k=1}^{N} \Big[v_k(h^{t-1}, \alpha^{t-1}) \left(h^t(\tilde{x}_k)\right)^t}.$$

3.2. Задача классификации

В задаче классификации для двух классов функция потерь может иметь вид $\ell(y,\tilde{y}) = \varrho(1 - y\tilde{y})$, где $\varrho(r)$ — неотрицательная монотонно возрастающая функция, $\lim\limits_{r\to+\infty} \varrho(r) = +\infty$, $\varrho(r) > 0$ при $r < 0$.

Итерационная схема (10) принимает вид:

$$h^t, \alpha^t = \arg\min_{h,\alpha} \sum_{k=1}^{N} \Big[\nu_k(h^{t-1}, \alpha^{t-1}) \varrho\big(1 - \tilde{y}_k \tilde{H}_k - \alpha \tilde{y}_k h(\tilde{x}_k)\big),$$

где $\tilde{y}_k \tilde{H}_k + \alpha \tilde{y}_k h(\tilde{x}_k)$ — величина отступа для k-го прецедента.

Приведем примеры:

1) $\varrho(r) = \max(0, r)$;
2) $\varrho(r) = \dfrac{1}{\lambda} \ln(1 + e^{\lambda r})$;
3) $\varrho(r) = \dfrac{1}{2}\big(-r + \sqrt{\varepsilon^2 + r^2}\big)$.

В рамках классической схемы построения робастной регрессии [13] построим следующую процедуру итеративного перевзвешивания:

$$h^t = \arg\min_{h\in\mathcal{H}} \sum_{k=1}^{N} \Big[v_k(h^{t-1}, \alpha^{t-1}) \Big(1 - \tilde{y}\tilde{H}_k - \alpha^{t-1} \tilde{y}_k h(\tilde{x}_k) \Big)^2$$

$$\alpha^t = \arg\min_{\alpha} \sum_{k=1}^{N} \Big[v_k(h^{t-1}, \alpha^{t-1}) \Big(1 - \tilde{y}_k \tilde{H}_k - \alpha \tilde{y}_k h^t(\tilde{x}_k) \Big)^2,$$

где

$$v_k(h,\alpha) = \nu_k(h,\alpha)\varphi\big(1 - \tilde{y}_k\tilde{H}_k - \alpha^{t-1}\tilde{y}_k h^{t-1}(\tilde{x}_k)\big),$$

$$\varphi(r) = \varrho'(r)/r \quad \text{при} \quad r < 0$$

и

$$\varphi(r) = 0 \quad \text{при} \quad r \geqslant 0.$$

Величину α^t можно вычислить явно:

$$\alpha^t = \frac{\sum\limits_{k=1}^{N} v_k(h^{t-1},\alpha^{t-1})(1 - \tilde{y}_k\tilde{H}_k)\tilde{y}_k h^{t-1}(\tilde{x}_k)}{\sum\limits_{k=1}^{N} v_k(h^{t-1},\alpha^{t-1})\left(\tilde{y}_k h^{t-1}(\tilde{x}_k)\right)^2}.$$

4. Иллюстративные примеры

В следующих примерах будет использоваться робастная оценка среднего

$$\mathsf{WM}_\alpha\{z_1,\ldots,z_N\} = \frac{1}{N}\sum_{k=1}^{N}\min(z_k,\bar{z}_\alpha),$$

где

$$\bar{z}_\alpha = \mathsf{M}_\alpha\{z_1,\ldots,z_N\} = \arg\min_u \sum_{k=1}^{N}\rho_\alpha(z_k - u),$$

$$\rho_\alpha(r) = \begin{cases}\alpha\rho(r), & \text{если } r \geqslant 0\\ (1-\alpha)\rho(r), & \text{если } r < 0,\end{cases} \qquad \rho(r) = \sqrt{\varepsilon^2 + r^2} - \varepsilon.$$

Здесь M_α — «гладкий вариант» α-квантиля, $\varepsilon = 0{,}001$. Робастная оценка WM_α — среднее арифметическое предварительно отцензурированных неотрицательных значений при помощи порогового значения $\bar{z}_\alpha$. В качестве функции потерь в задачах регрессии будет выступать $\ell(y,\tilde{y}) = \frac{1}{2}(y-\tilde{y})^2$ — квадрат ошибки.

Функции $h(x,w)$ выбираются из класса сигмоидальных нейронов

$$h(x,w) = \sigma(w_0 + w_1x_1 + \cdots + w_nx_n),$$

где $\sigma(s) = \operatorname{th}\lambda s$ (по умолчанию $\lambda = 1$, если не оговорено иное). Таким образом, класс функций $L(\mathcal{H})$ описывает функции преобразования нейронной сети со скрытым слоем из сигмоидальных нейронов. Количество нейронов в скрытом слое относительно небольшое во избежание переобучения.

Все вычисления выполнены с помощью языка программирования python и библиотеки mlgrad (https://bitbucket.org/intellimath/mlgrad).

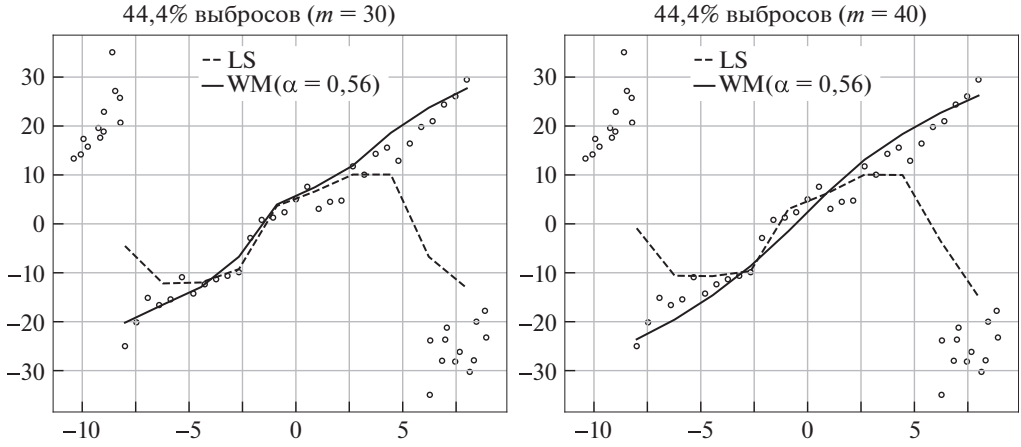


Рис. 1. Графики восстановленных функций для примера с линей ной регрессией.

1. Наглядный пример с линейной регрессией. В этом примере выбран набор точек на плоскости, расположенных вдоль некотор ой прямой линии. К ним добавлены новые точки — выбросы, которые расположены кучно по разные стороны от прямой линии, так чтобы при восстановлении линейной функции при помощи метода наименьших квадратов найденная прямая линия сильно поворачивалась, притягиваясь к выбросам. Выбр осы составляют 44% выборки. Параметр $\lambda = 0{,}5$ в $\sigma(s)$. На рис. 1 приведены графики восстановленных функций.

2. Набор данных breast_cancer. [2] Ко входным векторам предварительно была применена процедура стандартного масштабиров ания при помощи преобразования $\{x_k\} \to \left\{\frac{x_k - \bar{x}}{\sigma}\right\}$, где $\bar{x}$ — среднее арифметическое, а σ — стандартное отклонение, для приведения значений признаков ко взаимно сопоставимым масштабам значений. Для этого набора строили сь два варианта функции $H(x)$, которые содержат небольшое число слагаемых ($m = 20$ и $m = 30$). В робастном варианте $\alpha = 0{,}95$. На рис. 3 построены кривые распределения абсолютных значений ошибок в логарифмических к оординатах. Нетрудно увидеть, что применение более робастной функции оценки среднего значения может позволить уменьшить абсолютную величину ошибок для подавляющего большинства примеров.

3. Сгенерированная однослойная нейронная сеть с одним скрытым слоем. Это искусственно сгенерированный набор данных на основе функции

$$H(x) = \sum_{k=1}^{m} \alpha_j \sigma(w_{j,0} + w_{j,1} x_1 + w_{\mathrm{j},2} x_2), \quad m = 40,$$

в которой значения весов $w_{j,0}$, $w_{j,1}$, $w_{j,2}$ и коэффициентов α_j для простоты выбраны случайно из равномерного распределения на $[-1, 1]$ (то, что значения выбраны из равномерного распределения принципиального значения не

[2] https://archive.ics.uci.edu/ml/datasets/Breast+Cancer

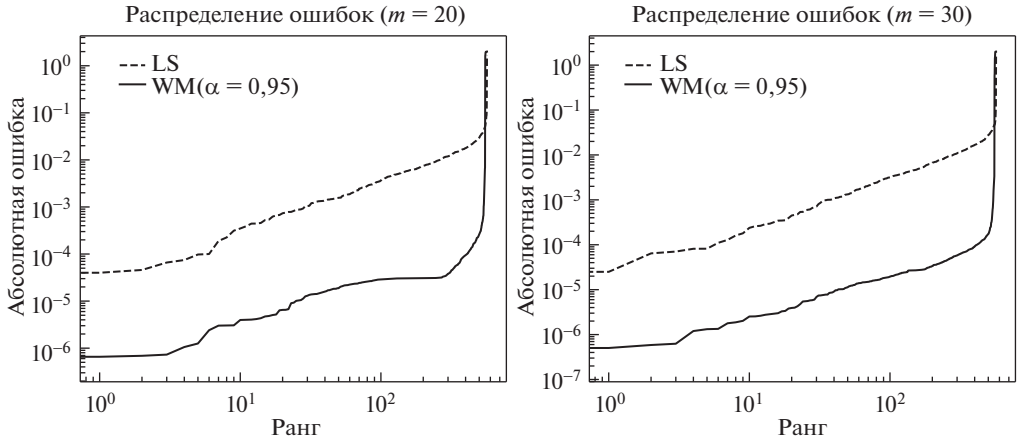


Рис. 2. Графики распределения абсолютных значений ошибок в примере 2.

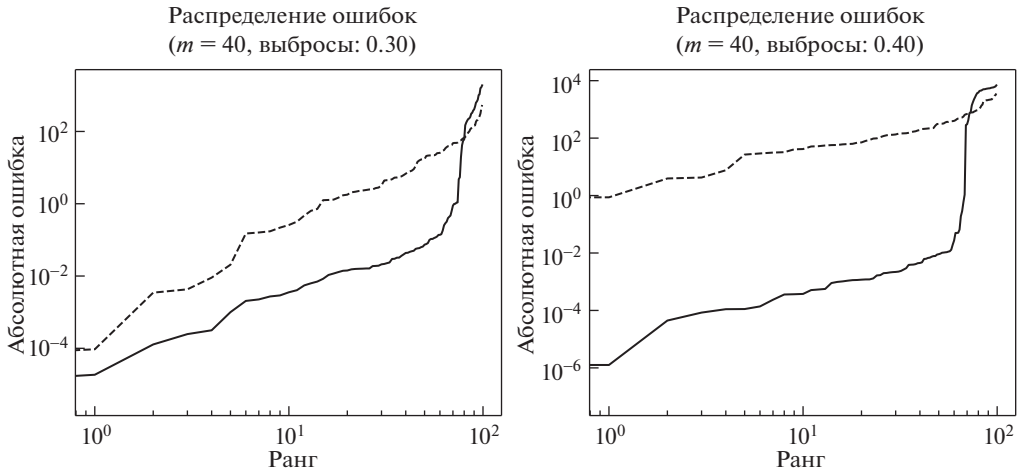


Рис. 3. Графики распределения абсолютных значений ошибок в примере 3.

имеет). Аналогично случайно выбирается набор входов $\{x_k\colon k = 1, \ldots, 100\} \subset$ $\subset [-3, 3]^2$. Для всех k вычисляются значения $y_k = H(x_k)$. Из этого набора данных создаются два набора с долями выбросов $M = 30\%$ и $M = 40\%$. Значение $\tilde{y}_k$ в точке выбросов увеличивается в 10 раз. На рис. 3 построены кривые распределения абсолютных значений ошибок в логарифмических координатах. На рисунках сплошная кривая соответствует робастному варианту градиентного бустинга. Нетрудно увидеть, что применение более робастной функции среднего значений может позволить ощутимо уменьшить абсолютную величину ошибок практически для всех выбросов. При применении стандартной процедуры градиентного бустинга в точках, которые не являются выбросами, наблюдаются очень большие значения ошибок. В результате применения робастной процедуры градиентного бустинга ошибки для нормальных точек могут стать достаточно малы.

5. Заключение

Предложенный в данной статье подход сравним с известным подходом к повышению робастности алгоритмов регрессии и классификации, основанным на применении более робастных функций потерь. Существенное отличие предложенной выше робастной схемы состоит в способе пересчета весов примеров в процедуре итеративного перевзвешивания. В случае применения в (5) с $v_k = 1/N$ более робастных функций потерь веса примеров вычисляются по формуле вида

$$v_k = \varphi(z_k),$$

где $\varphi(z)$ — неотрицательная, как правило, убывающая функция от z или $|z|$. Эффект снижения влияния выбросов достигается за счет малости весов примеров, которые являются выбросами (как правило, с большими значениями z или $|z|$). В нашем подходе веса пересчитываются по формуле вида:

$$v_k = \psi(z_k - \bar{z}),$$

где $\varphi(z)$ — тоже неотрицательная убывающая функция от z, $\bar{z}$ — величина робастной оценки среднего значения $z_1, \ldots, z_N$, которая нечувствительна или малочувствительна к выбросам. Отличие состоит в том, здесь вес примера является функцией отклонения z_k от среднего значения. Так, в задаче регрессии, когда значения z_k соответствуют ошибкам, в ситуации, со значением $\bar{z}$ существенно отличающимся от нуля, значения весов примеров в предложенном робастном подходе оказываются существенно меньше. Это получается потому, что когда все ошибки существенно отделены от нуля, они оказываются в области значений z, где значение функции φ (в (2) и (3)) убывает медленнее, чем около нуля. В предложенном робастном подходе случае разность $z_k - \bar{z}$ оказывается ближе к нулю и поэтому происходит более быстрое падение значений весов примеров по мере удаления z_k от $\bar{z}$. В результате в рамках предложенного здесь метода примеры, соответствующие выбросам, получают такие малые значения весов (по сравнению с весами примеров, которые не являются выбросами), достаточные для того, чтобы преодолеть их влияние.

СПИСОК ЛИТЕРАТУРЫ

1. *Freund Y., Schapire R.E.* A decision-theoretic generalization of on-line learning and an application to boosting // J. of Comput. and Syst. Sci. 1997. V. 55. No. 1. P. 119–139.
2. *Kanamori T., Takenouchi T., Eguchi S., Murata N.* Robust loss functions for boosting // Neural Computation. 2007. V. 19. No. 8. P. 2183–2244.
3. *Holland P.W., Welsch R.E.* Robust regression using iteratively reweighted least squares // Communications in Statistics — Theory and Methods. 1977. V. 6. No. 9. P. 813–827.
4. *Rousseeuw P.J., Leroy A.M.* Robust Regression and Outlier Detection. New York: John Wiley and Sons. 1987.

5. *Rousseeuw P.J., Hubert M.* High-breakdown estimators of multivariate location and scatter / Becker C., Fried R., Kuhnt S., editors. Robustness and Complex Data Structures. Springer, 2013. P. 49–66.
6. *Шибзухов З.М.* О принципе минимизации эмпирического риска на основе усредняющих агрегирующих функций // Докл. РАН. 2017. Т. 476. № 5. С. 495–499.
7. *Shibzukhov Z.M.* Machine learning based on the principle of minimizing robust mean estimates / Advances in Intelligent Systems and Computing. V. 1310. P. 472–477. Springer International Publishing. 2020.
8. *Friedman J.H.* Greedy function approximation: A gradient boosting machine // Annals Statist. 2001. V. 29. No. 5.
9. *Csiszar I., Tusnady G.* Information geometry and alternating minimization procedures // Statistics and Decisions, Supplement Issue. 1984. No. 1. P. 205–237.
10. *Calvo T., Beliakov G.* Aggregation functions based on penalties // Fuzzy Sets and Systems. 2010. V. 161. No. 10. P. 1420–1436.
11. *Shibzukhov Z.M., Semenov T.A.* Machine learning based on minimizing robust mean estimates. In: Pattern Recognition. ICPR International Workshops and Challenges. P. 112–119. Springer International Publishing. 2021.
12. *Kingma D.P., Ba J.* Adam: A method for stochastic optimization. arXiv:1412.6980. https://doi.org/10.48550/arXiv.1412.6980
13. *Huber P.J.* Robust Statistics. John Wiley and Sons. 1981.

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 31.01.2022

После доработки 23.05.2022

Принята к публикации 29.06.2022

СОДЕРЖАНИЕ

Тематический выпуск

CONTENTS

Topical issue