МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»
(САМАРСКИЙ УНИВЕРСИТЕТ)

*О.Н. САПРЫКИН*

# ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ

Рекомендовано редакционно-издательским советом федерального государственного автономного образовательного учреждения высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева» в качестве учебного пособия для обучающихся по основной образовательной программе высшего образования по направлению подготовки 23.03.01 Технологии транспортных процессов

САМАРА
Издательство Самарского университета
2020

УДК 656.05(075)+004.056(075)
ББК 65.37я7+32.973-018.2я7
С197

Рецензенты: д-р техн. наук, проф. А. Н. Коптев,
канд. экон. наук А. В. Зиновьев

***Сапрыкин, Олег Николаевич***
**С197** **Интеллектуальный анализ данных** : учебное пособие / *О.Н. Сапрыкин*. – Самара: Издательство Самарского университета, 2020. – 80 с.: ил.

**ISBN 978-5-7883-1563-8**

Изложены основные методики интеллектуального анализа данных при разработке проактивных систем защиты информации, методы автоматического вывода правил из накопленных данных. Рассмотрены вопросы решения задач классификации и кластеризации такими методами машинного обучения как логистическая регрессия, k-средних и EM-алгоритм.

Содержание учебного пособия соответствует тематике лекций для бакалавров по дисциплине «Информационные системы и технологии в комплексной безопасности», читаемых автором в Самарском университете.

Предназначено для студентов направления подготовки 23.03.01 Технологии транспортных процессов.

Подготовлено на кафедре организации и управления перевозками на транспорте.

УДК 656.05(075)+004.056(075)
ББК 65.37я7+32.973-018.2я7

ISBN 978-5-7883-1563-8 

## СОДЕРЖАНИЕ

# ПРЕДИСЛОВИЕ

В современном мире подавляющее большинство сфер деятельности человека имеют глубокое проникновение информационных технологий, позволяющие наиболее эффективно решать как повседневные, так и производственные задачи. Однако широкое применение информационных технологий имеет и обратную сторону – безоговорочное доверие к ним может использоваться злоумышленниками для достижения собственных целей. Для того, чтобы избежать негативных последствий подобного вмешательства необходимо организовывать системы проактивной защиты, ограждающей пользователей от действий злоумышленников. Оснащение информационных систем комплексными средствами защиты создает безопасное информационное пространство как для индивидуумов и социальных групп в сети Интернет, так и на предприятии в корпоративной сети.

Упомянутые обстоятельства обусловливают необходимость подготовки высококвалифицированных специалистов в области разработки средств интеллектуального анализа данных для систем комплексной безопасности. Целью данного пособия является методическая поддержка подготовки данных специалистов.

Содержание книги соответствует тематике лекций для бакалавров направления подготовки «Технологии транспортных процессов» по дисциплине «Информационные системы и технологии в комплексной безопасности», читаемых автором в Самарском университете. Вопросы, затронутые в учебном пособии, могут быть интересны также магистрантам и аспирантам.

Глубокую благодарность автор выражает доктору технических наук, профессору, профессору кафедры эксплуатации авиационной техники Самарского университета А.Н. Коптеву и генеральному директору ООО «Средневолжская логистическая компания», кандидату экономических наук А.В. Зиновьеву за полезные замечания, сделанные при рецензировании рукописи.

# ПЕРЕЧЕНЬ ОСНОВНЫХ СОКРАЩЕНИЙ

| | | |
|---|---|---|
| ОШ | – | Отношение шансов |
| OR | – | Odds ratio |
| ML | – | Machine Learning |
| CBR | – | Case-based-reasoning |
| TP | – | True-positive |
| TN | – | True negative |
| FP | – | False positive |
| FN | – | False negative |
| EM | – | Expectation maximization |
| DBSCAN | – | Density-based spatial clustering of applications with noise |

# ВВЕДЕНИЕ

Темпы развития технологий обработки, хранения и передачи информации растут с каждым днем. Активное их применение диктует повышенные требования к вопросам информационной безопасности, поскольку искажение, уничтожение или хищение информации могут вызвать серьезные проблемы, как у отдельных граждан, так и у социальных групп, компаний и даже государства.

Информационная безопасность является комплексным понятием, подразумевающим все аспекты, связанные с определением, достижением и поддержанием конфиденциальности, целостности, доступности, аутентичности и достоверности информации и средств её обработки [1]. Принято выделять следующие проблемы информационной безопасности:

- проблемы гуманитарного характера, возникающие в связи с бесконтрольным использованием и распространением персональных данных граждан, вторжениями в частную жизнь, клеветой и кражами личности;
- проблемы экономического и юридического характера, возникающие в результате утечки, искажения и потери коммерческой и финансовой информации, краж брендов и интеллектуальной собственности, раскрытия информации о материальном положении граждан, промышленного шпионажа и распространения материалов, наносящих ущерб репутации компаний;
- проблемы политического характера, возникающие из-за информационных войн и электронной разведки в интересах политических групп, компрометации государственной тайны, атак на информационные системы важных оборонных, транспортных и промышленных объектов;

В последнее время решение проблем информационной безопасности усложнилось из-за появления таких сложных задач, как [2]:

- разработка и реализация надёжных систем электронно-цифровой подписи, электронных выборов, закупок и платежей;

- создание и внедрение передовых средств аутентификации (биометрических и других);
- разработка и внедрение новых методов обеспечения надежности и отказоустойчивости (инновационные технологии кластеризации, виртуализации и др.);
- защита беспроводных соединений, мобильных устройств и «умной» электроники;
- обеспечение безопасности веб-сервисов и «облачных» технологий;
- защита от вирусных и хакерских атак, направленных на конкретные предприятия.

Одним из передовых методов решения указанных задач является интеллектуальный анализ данных, концепция которого заключается в гипотезе, что данные могут быть приближенными, неполными, двойственными, непоследовательными, неявными, но при этом обладать огромным скрытым потенциалом. Для извлечения полезных знаний из данных применяются методы правдоподобного вывода, позволяющие делать общие выводы на основе большого числа наблюдений.

По мере возрастания объёма данных упрощается рассмотрение их как стохастического процесса с некоторой тенденцией. Выявление закономерностей в этих данных может иметь большой потенциал благодаря комплексному применению методов анализа данных [3]. Методы интеллектуального анализа данных основываются на таких концептуальных направлениях как математическая статистика, машинное обучение и искусственный интеллект [4]. При этом точность выявленных моделей будет возрастать с ростом объема и разнотипности данных [5].

В сфере компьютерной безопасности методы интеллектуального анализа данных связаны с созданием перспективных систем защиты информации. Именно методика интеллектуального анализа данных обеспечивает такие возможности как эволюционная адаптация, наследование и представление опыта экспертов информационной безопасности в виде доступной для интерпретации системы нечетких правил [6].

Интеллектуальный анализ данных применим к таким проблемам, как выявление вторжений, проверка безопасности. К примеру, способ выявления отклонений используется в целях обнаружения

подозрительного поведения пользователей, исследование ссылок применяется для отслеживания вредоносного кода, а классификация для обнаружения атак на ресурсы [7].

При помощи прогнозирования возможно определение потенциальных атак в будущем, основываясь на информации о злоумышленниках при помощи социальных сетей, электронной почты или разговорах по телефону. Более того, для многих угроз достаточно анализа только статичных данных, а для других угроз, таких как сетевое вторжение, необходим анализ потоковой информации. Если необходим анализ данных в реальном времени, то модели строятся и обновляются в режиме online. Например, чтобы обнаружить мошенничество с кредитной картой, необходимо использовать обработку модели в реальном времени.

Таким образом можно сделать вывод, что интеллектуальный анализ данных комплексной безопасности является необходимым и современным средством защиты информации и сохранения конфиденциальности данных.

# 1 МАШИННОЕ ОБУЧЕНИЕ

## 1.1 Введение

Машинное обучение появляется в тот момент, когда переменные, которыми описывается объект, можно разделить на две части: наблюдаемые и скрытые, или латентные, переменные. *Наблюдаемые переменные* – это те переменные, которые можно измерить для произвольного объекта. *Скрытые переменные* можно померить для ограниченного числа объектов просто потому, что, как правило, их измерение сопряжено с финансовыми затратами, человеческими, временными, либо в принципе невозможно. Например, эти переменные характеризуют свойства объекта в будущем. При этом предполагается, что между наблюдаемыми и скрытыми переменными есть некоторая взаимосвязь. И, собственно, на поиск этой взаимосвязи направлены современные алгоритмы машинного обучения.

Предположим, мы банкиры и у нас есть клиенты, которые жаждут получить от нас кредит. Вопрос в том, кому кредит можно выдавать, а кому лучше не выдавать, потому что он его не вернет. Это классическая задача машинного обучения. В качестве наблюдаемых переменных у нас есть характеристики клиентов, например то, что они заполняют в своих анкетах: пол, возраст, образование, уровень доходов, состав семьи. Все эти переменные мы можем легко измерить, заставив клиента заполнить анкету. Как скрытая компонента в простейшем случае выступает бинарная величина: вернет клиент кредит или не вернет. В более сложных случаях предполагается, что мы оцениваем риск невозврата кредита как некоторую вероятность, с которой кредит может быть не возвращен.

## 1.2 Определение, типы и способы машинного обучения. Задачи, решаемые с помощью машинного обучения

### *1.2.1 Определения*

**Машинное обучение** (англ. Machine Learning, ML) – обширный подраздел искусственного интеллекта, математическая дисциплина, использующая разделы математической статистики, численных методов оптимизации, теории вероятностей, дискретного анализа и извлекающая знания из данных, а также занимающаяся поиском закономерностей этих данных. Поэтому, чтобы понять, что такое машинное обучение, необходимо разобраться в том, что же представляют собой данные [8]:

**Данные** – это воспринимаемые человеком факты, события, сообщения, измеряемые характеристики, регистрируемые сигналы, то есть некая совокупность объектов.

Специфика данных в том, что они, с одной стороны, существуют независимо от наблюдателя, а с другой – становятся собственно «данными» лишь тогда, когда существует целенаправленно собирающий их субъект. В итоге данные должны быть тем основанием, на котором возводятся все заключения, выводы и решения [9].

**Объект** – явление, предмет, на который направлена какая-либо деятельность. Например, банковский клиент, транспорт, изображение и т.п.

### *1.2.2 Типы машинного обучения*

1. *Дедуктивное обучение* предполагает формализацию знаний экспертов и их перенос в компьютер в виде базы знаний.

Дедуктивное обучение принято относить к области *экспертных систем*, поэтому термины машинное обучение и *обучение по прецедентам* можно считать синонимами [10].

Многие методы индуктивного обучения разрабатывались как альтернатива классическим *статистическим подходам*. Многие методы тесно связаны с извлечением информации (Information Extraction), интеллектуальным анализом данных (Data mining).

2. Обучение по прецедентам, или *индуктивное обучение*, основано на выявлении закономерностей в эмпирических данных.

*Обучение на примерах* (англ. Learning from Examples) – вид обучения, при котором интеллектуальной системе предъявляется набор положительных и отрицательных примеров, связанных с какой-либо заранее неизвестной закономерностью. В интеллектуальных системах вырабатываются решающие правила, с помощью которых происходит разделение множества примеров на положительные и отрицательные. Качество разделения, как правило, проверяется экзаменационной выборкой примеров [11].

Комбинированное обучение.

### *1.2.3 Задачи машинного обучения*

1. ***Обучение с учителем* (*supervised learning*):**

- *задача классификации* (classification) отличается тем, что множество допустимых ответов конечно. Их называют метками классов (class label). Класс – это множество всех объектов с данным значением метки;
- *задача регрессии* (regression) отличается тем, что допустимым ответом является действительное число или числовой вектор;
- *задача ранжирования* (learning to rank) отличается тем, что ответы надо получить сразу на множестве объектов, после чего отсортировать их по значениям ответов. Может сводиться к задачам классификации или регрессии. Часто применяется в информационном поиске и анализе текстов;
- *задача прогнозирования* (forecasting) отличается тем, что объектами являются отрезки временных рядов, обрывающиеся в тот момент, когда требуется сделать прогноз на будущее. Для решения задач прогнозирования часто удаётся приспособить методы регрессии или классификации, причём во втором случае речь идёт скорее о задачах принятия решений.

2. ***Обучение без учителя* (*unsupervised learning*):**

- *задача кластеризации* (clustering) заключается в том, чтобы сгруппировать объекты в кластеры, используя данные о попарном сходстве объектов. Функционалы качества могут

определяться по-разному, например, как отношение средних межкластерных и внутрикластерных расстояний;

- *задача поиска ассоциативных правил* (association rules learning). Исходные данные представляются в виде признаковых описаний. Требуется найти такие наборы признаков и такие значения этих признаков, которые особенно часто (неслучайно часто) встречаются в признаковых описаниях объектов;
- *задача фильтрации выбросов* (outliers detection) – обнаружение в обучающей выборке небольшого числа нетипичных объектов. В некоторых приложениях их поиск является самоцелью (например, обнаружение мошенничества). В других приложениях эти объекты являются следствием ошибок в данных или неточности модели, то есть шумом, мешающим настраивать модель, и должны быть удалены из выборки;
- *задача построения доверительной области* (quantile estimation) – области минимального объёма с достаточно гладкой границей, содержащей заданную долю выборки;
- *задача сокращения размерности* (dimensionality reduction) заключается в том, чтобы по исходным признакам с помощью некоторых функций преобразования перейти к наименьшему числу новых признаков, не потеряв при этом никакой существенной информации об объектах выборки. В классе линейных преобразований наиболее известным примером является метод главных компонент;
- *задача заполнения пропущенных значений* (missing values) – замена недостающих значений в матрице объекты–признаки их прогнозными значениями.

3. ***Частичное обучение* (*semi-supervised learning*).** Пример прикладной задачи – автоматическая рубрикация большого количества текстов при условии, что некоторые из них уже отнесены к каким-то рубрикам.

4. ***Трансдуктивное обучение* (*transductive learning*).** Дана конечная обучающая выборка прецедентов. Требуется по этим частным данным сделать предсказания относительно других частных данных – тестовой выборки. В отличие от стандартной постановки,

здесь не требуется выявлять общую закономерность, поскольку известно, что новых тестовых прецедентов не будет. С другой стороны, появляется возможность улучшить качество предсказаний за счёт анализа всей тестовой выборки целиком, например, путём её кластеризации. Во многих приложениях трансдуктивное обучение практически не отличается от частичного обучения.

5. ***Обучение с подкреплением* (*reinforcement learning*).** Примеры прикладных задач: формирование инвестиционных стратегий, автоматическое управление технологическими процессами, самообучение роботов и т.д.

6. ***Метаобучение* (*meta-learning или learning-to-learn*)** отличается тем, что прецедентами являются ранее решённые задачи обучения. Требуется определить, какие из используемых в них эвристик работают более эффективно. Конечная цель – обеспечить постоянное автоматическое совершенствование алгоритма обучения с течением времени.

*Многозадачное обучение* (multi-task learning). Набор взаимосвязанных или схожих задач обучения решается одновременно, с помощью различных алгоритмов обучения, имеющих схожее внутреннее представление. Информация о сходстве задач между собой позволяет более эффективно совершенствовать алгоритм обучения и повышать качество решения основной задачи.

*Индуктивный перенос* (inductive transfer). Опыт решения отдельных частных задач обучения по прецедентам переносится на решение последующих частных задач обучения. Для формализации и сохранения этого опыта применяются реляционные или иерархические структуры представления знаний.

Иногда к метаобучению ошибочно относят *построение алгоритмических композиций*, в частности, бустинг; однако в композициях несколько алгоритмов решают одну и ту же задачу, тогда как метаобучение предполагает, что решается много разных задач.

***Специфические прикладные задачи.***

Некоторые задачи, возникающие в прикладных областях, имеют черты сразу нескольких стандартных типов задач обучения, поэтому их трудно однозначно отнести к какому-то одному типу.

1. *Формирование инвестиционного портфеля* (portfolio selection) – это динамическое обучение с подкреплением, в котором

очень важен отбор информативных признаков. Роль признаков играют финансовые инструменты. Состав оптимального набора признаков (портфеля) может изменяться со временем. Функционалом качества является долгосрочная прибыль от инвестирования в данную стратегию управления портфелем [17].

2. *Коллаборативная фильтрация* (collaborative filtering) – это прогнозирование предпочтений пользователей на основе их прежних предпочтений и предпочтений схожих пользователей. Применяются элементы классификации, кластеризации и восполнения пропущенных данных [18].

# 2 КЛАССИФИКАЦИЯ

Классификация – системное распределение изучаемых предметов, явлений, процессов по родам, видам, типам, по каким-либо существенным признакам для удобства их исследования; группировка исходных понятий и расположение их в определенном порядке, отражающем степень этого сходства [19].

Классификация – упорядоченное по некоторому принципу множество объектов, которые имеют сходные классификационные признаки (одно или несколько свойств), выбранные для определения сходства или различия между этими объектами.

Классификация требует соблюдения следующих правил:

- в каждом акте деления необходимо применять только одно основание;
- деление должно быть соразмерным, т.е. общий объем видовых понятий должен равняться объему делимого родового понятия;
- члены деления должны взаимно исключать друг друга, их объемы не должны перекрещиваться;
- деление должно быть последовательным.

Различают:

- вспомогательную (искусственную) классификацию, которая производится по внешнему признаку и служит для придания множеству предметов (процессов, явлений) нужного порядка;
- естественную классификацию, которая производится по существенным признакам, характеризующим внутреннюю общность предметов и явлений. Она является результатом и важным средством научного исследования, т.к. предполагает и закрепляет результаты изучения закономерностей классифицируемых объектов.

В зависимости от выбранных признаков, их сочетания и процедуры деления понятий классификация может быть:

- простой – деление родового понятия только по признаку и только один раз до раскрытия всех видов. Примером такой классификации является дихотомия, при которой членами деления бывают только два понятия, каждое из которых является противоречащим другому (т.е. соблюдается принцип: «А и не А»);
- сложной – применяется для деления одного понятия по разным основаниям и синтеза таких простых делений в единое целое. Примером такой классификации является периодическая система химических элементов.

Под классификацией будем понимать отнесение объектов (наблюдений, событий) к одному из заранее известных классов.

Классификация – это закономерность, позволяющая делать вывод относительно определения характеристик конкретной группы. Таким образом, для проведения классификации должны присутствовать признаки, характеризующие группу, к которой принадлежит то или иное событие или объект (обычно при этом на основании анализа уже классифицированных событий формулируются некие правила).

Классификация относится к стратегии обучения с учителем (supervised learning), которое также именуют контролируемым или управляемым обучением.

*Задачей классификации часто называют предсказание категориальной зависимой переменной (т.е. зависимой переменной, являющейся категорией) на основе выборки непрерывных и/или категориальных переменных.*

Например, можно предсказать, кто из клиентов фирмы является потенциальным покупателем определенного товара, а кто – нет, кто воспользуется услугой фирмы, а кто – нет, и т.д. Этот тип задач относится к задачам **бинарной классификации**, в них зависимая переменная может принимать только два значения (например, да или нет, 0 или 1) [20].

Другой вариант классификации возникает, если зависимая переменная может принимать значения из некоторого множества предопределенных классов. Например, когда необходимо предсказать, какую марку автомобиля захочет купить клиент. В этих случаях рассматривается множество классов для зависимой переменной.

Классификация может быть **одномерной** (по одному признаку) и **многомерной** (по двум и более признакам).

**Многомерная классификация** была разработана биологами при решении проблем дискриминации для классифицирования организмов [21]. Одной из первых работ, посвященных этому направлению, считают работу Р. Фишера (1930 г.), в которой организмы разделялись на подвиды в зависимости от результатов измерений их физических параметров. Биология была и остается наиболее востребованной и удобной средой для разработки многомерных методов классификации.

Рассмотрим задачу классификации на простом примере. Допустим, имеется база данных о клиентах туристического агентства с информацией о возрасте и доходе за месяц. Есть рекламный материал двух видов: более дорогой и комфортный отдых и более дешевый, молодежный отдых. Соответственно, определены два класса клиентов: класс 1 и класс 2. База данных приведена в табл. 1.

Таблица 1. **База данных туристического агентства**

| Код клиента | Возраст | Доход | Класс |
|---|---|---|---|
| 1 | 18 | 25 | 1 |
| 2 | 22 | 100 | 1 |
| 3 | 30 | 70 | 1 |
| 4 | 32 | 120 | 1 |
| 5 | 24 | 15 | 2 |
| 6 | 25 | 22 | 1 |
| 7 | 32 | 50 | 2 |
| 8 | 19 | 45 | 2 |
| 9 | 22 | 75 | 1 |
| 10 | 40 | 90 | 2 |

**Задача**. Определить, к какому классу принадлежит новый клиент и какой из двух видов рекламных материалов ему стоит отсылать.

Для наглядности представим нашу базу данных в двухмерном измерении (возраст и доход), в виде множества объектов, принадлежащих классам 1 (оранжевая метка) и 2 (серая метка). На рис. 1 приведены объекты из двух классов.

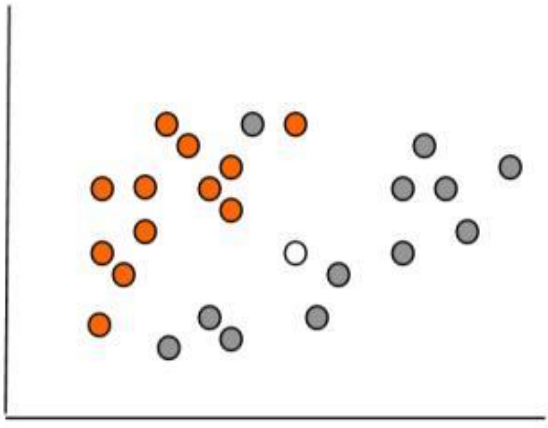


Рис. 1. Множество объектов базы данных в двухмерном измерении

Решение нашей задачи будет состоять в том, чтобы определить, к какому классу относится новый клиент, на рисунке обозначенный белой меткой.

## 2.1 Процесс классификации

Цель процесса классификации состоит в том, чтобы построить модель, которая использует прогнозирующие атрибуты в качестве входных параметров и получает значение зависимого атрибута. Процесс классификации заключается в разбиении множества объектов на классы по определенному критерию.

**Классификатором** называется некая сущность, определяющая, какому из предопределенных классов принадлежит объект по вектору признаков.

Для проведения классификации с помощью математических методов необходимо иметь формальное описание объекта, которым можно оперировать, используя математический аппарат классификации. Таким описанием в нашем случае выступает база данных. Каждый объект (запись базы данных) несет информацию о некотором свойстве объекта.

Набор исходных данных (или выборку данных) разбивают на два множества: обучающее и тестовое.

***Обучающее множество*** (training set) – множество, которое включает данные, использующиеся для обучения (конструирования) модели. Такое множество содержит входные и выходные (целевые) значения примеров. Выходные значения предназначены для обучения модели.

***Тестовое*** (test set) множество также содержит входные и выходные значения примеров. Здесь выходные значения используются для проверки работоспособности модели.

Процесс классификации состоит из двух этапов: конструирования модели и ее использования.

1. Конструирование модели (описание множества предопределенных классов):

a) каждый пример набора данных относится к одному предопределенному классу;

b) на этом этапе используется обучающее множество, на нем происходит конструирование модели;

c) полученная модель представлена классификационными правилами, деревом решений или математической формулой.

2. Использование модели (классификация новых или неизвестных значений):

a) оценка правильности (точности) модели:

- известные значения из тестового примера сравниваются с результатами использования полученной модели;
- уровень точности – процент правильно классифицированных примеров в тестовом множестве;
- тестовое множество, т.е. множество, на котором тестируется построенная модель, не должно зависеть от обучающего множества.

b) если точность модели допустима, возможно использование модели для классификации новых примеров, класс которых неизвестен.

Процесс классификации, а именно, конструирование модели и ее использование, представлен на рисунках 2 и 3

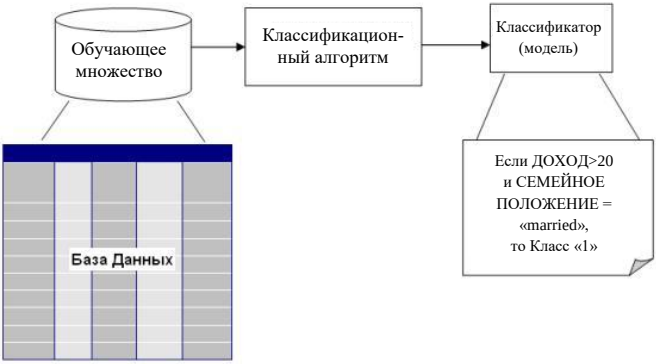



Рис. 2. Процесс классификации. Конструирование модели

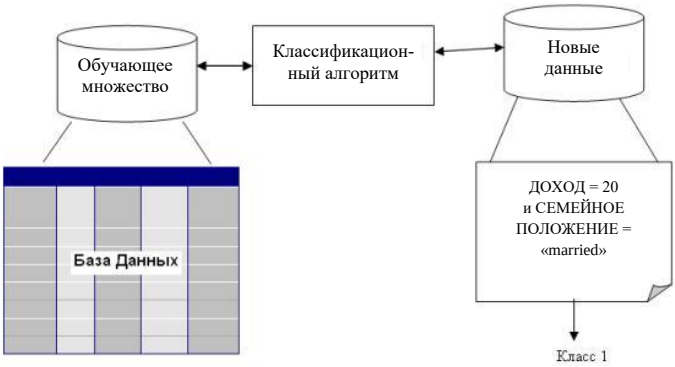



Рис. 3. Процесс классификации. Использование модели

## 2.2 Методы решения задач классификации

Для классификации используются различные методы:

- классификация с помощью деревьев решений;
- байесовская (наивная) классификация;
- классификация при помощи искусственных нейронных сетей;
- классификация методом опорных векторов;

- статистические методы, в частности, линейная регрессия;
- классификация при помощи метода ближайшего соседа;
- классификация CBR-методом;
- классификация при помощи генетических алгоритмов.

Схематическое решение задачи классификации некоторыми методами (при помощи линейной регрессии, деревьев решений и нейронных сетей) приведены на рис. 4, 5, 6.

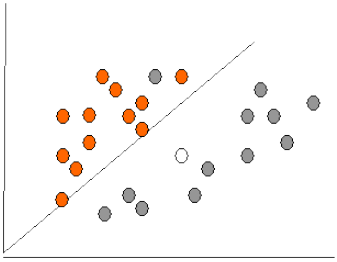


Рис. 4. Решение задачи классификации методом линейной регрессии

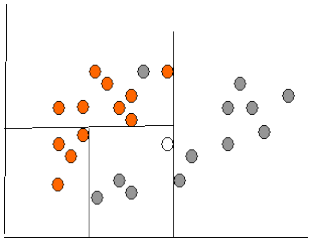


Рис. 5. Решение задачи классификации методом деревьев решений

Рис. 6. Решение задачи классификации методом нейронных сетей

## 2.3 Точность классификации: оценка уровня ошибок

Оценка точности классификации может проводиться при помощи кросс-проверки. Кросс-проверка (Cross-validation) – это процедура оценки точности классификации на данных из тестового множества, которое также называют кросс-проверочным множеством. Точность классификации тестового множества сравнивается с точностью классификации обучающего множества. Если классификация тестового множества дает приблизительно такие же результаты по точности, как и классификация обучающего множества, считается, что данная модель прошла кросс-проверку.

Разделение на обучающее и тестовое множества осуществляется путем деления выборки в определенной пропорции, например обучающее множество – две трети данных и тестовое – одна треть данных. Этот способ следует использовать для выборок с большим количеством примеров. Если же выборка имеет малые объемы, рекомендуется применять специальные методы, при использовании которых обучающая и тестовая выборки могут частично пересекаться.

Оценивание методов следует проводить, исходя из следующих характеристик: скорость, робастность, интерпретируемость, надежность.

**Скорость** характеризует время, которое требуется на создание модели и ее использование.

**Робастность**, т.е. устойчивость к каким-либо нарушениям исходных предпосылок, означает возможность работы с зашумленными данными и пропущенными значениями в данных [22].

**Интерпретируемость** обеспечивает возможность понимания модели аналитиком.

**Надежность** методов классификации предусматривает возможность работы этих методов при наличии в наборе данных шумов и выбросов.

# 3 ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ

## 3.1 Понятие логистической регрессии

Линейная регрессионная модель не всегда способна качественно предсказывать значения зависимой переменной. Выбирая для построения модели линейное уравнение, мы естественным образом не накладываем никаких ограничений на значения зависимой переменной. А такие ограничения могут быть существенными [23].

Например, при проектировании оптимальной длины шахты лифта в новом здании необходимо учесть, что эта длина не может превышать высоту здания вообще.

Линейная регрессионная модель может дать результаты несовместимые с реальностью. С целью решения данных проблем полезно изменить вид уравнения регрессии и подстроить его для решения конкретной задачи.

Вообще, логистическая регрессионная модель предназначена для решения задач предсказания значения непрерывной зависимой переменной при условии, что эта зависимая переменная может принимать значения на интервале от 0 до 1.

В статистике логистическая регрессия – модель, используемая для предсказания вероятности возникновения события «подгоном» данных к логистической кривой. При этом используют несколько предсказывающих переменных, которые могут быть или числовыми, или категориальными. Например, вероятность того, что у человека случится сердечный приступ в определенный период времени, может быть предсказана в зависимости от возраста человека, пола и индекса массы тела. Логистическая регрессия широко используется в медицинских и общественных науках, так же в маркетинговых исследованиях, таких как предсказание склонности клиента купить определенный продукт или прекратить подписку [24].

Другие названия для логистической регрессии, используемые в различных прикладных областях, включают логистическую модель и классификатор максимальной энтропии.

Логистическая регрессия относится к классу моделей, известных как обобщенные линейные модели.

Объяснение логистической регрессии начинается с объяснения логистической функции:

$$f(z) = \frac{1}{1 + e^{-z}}.$$

Типичный график логистической функции показан на рис. 7.

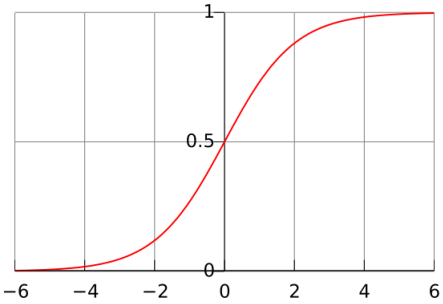



Рис. 7. Логистическая функция, с $z$ на горизонтальной оси и $f(z)$ на вертикальной оси

По оси абсцисс – управляющий параметр $z$ («Вход»), по оси ординат $f(z)$ – «отклик». Логистическая функция полезна, потому что она может принимать любые входные значения от минус бесконечности до плюс бесконечности, тогда как отклик (функция) ограничена диапазоном [0; 1]. Переменная $z$ отражает подверженность некоторому набору факторов риска, в то время как $f(z)$ представляет вероятность конкретного исхода, при заданном наборе рисков. Переменная $z$ является мерой полного вклада всех факторов риска, используемых в модели, и известна как *logit*:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_k x_k \ ,$$

где $\beta_0$ называют «точкой пересечения», а $\beta_1$, $\beta_2$, $\beta_3$ и т. д. называют «коэффициентами регрессии» для управляющих параметров (факторов риска) $x_1$, $x_2$, $x_3$ соответственно.

Точка пересечения – фоновая величина риска, т.е. величина $z$, при нулевых значениях всех специфических факторов риска. Каждый из коэффициентов регрессии описывает размер вклада соответствующего фактора риска. Положительный коэффициент регрессии означает, что данный фактор увеличивает общий риск (т.е. повышает вероятность анализируемого исхода), в то время как отрицательный коэффициент означает, что этот фактор уменьшает риск; большой коэффициент регрессии означает, что данный фактор существенно влияет на совокупный риск, в то время как почти нулевой коэффициент регрессии означает, что этот фактор имеет небольшое влияние на вероятность результата.

Логистическая регрессия – полезный способ описать влияние одного или нескольких факторов риска (например, возраст, пол и т.д.) на результат, такой как смерть (логистическая функция может принимать только два возможных значения: мертвый или не мертвый). Применимость логистической регрессии может быть продемонстрирована на фиктивном примере смертности от болезней сердца. Эта упрощенная модель использует только три фактора риска (возраст, пол и уровень холестерина в крови), чтобы предсказать риск смерти от заболеваний сердца на десятилетний период. Вот пример подгоночной модели:

$\beta = -0{,}5 -$ пересечение с осью ординат;

$\beta_1 = 2{,}0$;

$\beta_2 = -1{,}0$;

$\beta_3 = 1{,}2$;

$x_1$– барьер превышения пятидесятилетнего возраста;

$x_2 -$ пол (принимает значение 0 (мужчина)или 1(женщина);

$x_3 -$ уровень холестерина в $\frac{\text{моль}}{\text{л}}$, уменьшенный на 5,0;

Согласно этой модели, вероятность смерти в результате болезни сердца определяется формулой, зависящей от возраста, пола и уровня холестерина:

$$\text{Риск смерти} = \frac{1}{1 + e^{-z}},$$

где $z = -5{,}0 + 2{,}0x_1 - 1{,}0x_2 + 1{,}2x_3$

В этой модели увеличение возраста приводит к увеличению риска смерти от болезни сердца ($z$ повышается на 2,0 в течение каждых 10 лет в возрасте старше 50), женщины менее подвержены сердечным заболеваниям, чем мужчины ($z$ понижается на 1,0, если пациентка – женщина), и превышение содержания холестерина над пороговым уровнем приводит к увеличению риска смерти ($z$ повышается на 1,2 для каждого 1 ммоль/л холестерина свыше 5 ммоль/л).

Применим эту модель для оценки риска смерти некоего Петренко Ивана Карловича: ему 50 лет, и его уровень холестерина – 7,0 mmol/L.

$$\frac{1}{1+e^{-z}},$$

где $z = -5{,}0 + 2{,}0 \times (5{,}0 - 5{,}0) - 1{,}0 \times 0 + 1{,}2 \times (7{,}0 - 5{,}0)$.

Согласно модели, риск смерти господина Петренко от болезни сердца за следующие 10 лет составляет 0,07 (или 7 %).

### 3.2 Преобразования логистической функции

Логистическую функцию:

$$f(z) = \frac{1}{1+e^{-z}}$$

легко линеаризовать с помощью логистического преобразования.

Введем новый вектор $\beta = (\beta_0,\ \beta_1, \beta_2 \ldots\ \beta_l)$. Тогда

$$f(z) = \frac{1}{1+e^{-\beta^T z}}.$$

Пусть $f(z) = P$. Получаем:

$$P = \frac{1}{1+e^{-\beta^T z}} \Rightarrow\ 1 + e^{-\beta^T z} = \frac{1}{P}\ \Rightarrow e^{-\beta^T z} = \frac{1-P}{P}\ \Rightarrow e^{\beta^T z} = \frac{P}{1-P} \Rightarrow$$
$$\Rightarrow \beta^T = \ln\frac{P}{1-P}.$$

Получаем $\ln\frac{P}{1-P} = \beta^T$. Функция $logit(p) = \ln\frac{P}{1-P}$ называется логит функцией (впервые термин *logit* был употреблен в 1944 Джозефом Берксоном (Joseph Berkson)).

Фактически, при проведении логистического преобразования обеих частей логистического регрессионного уравнения, приведенного выше, мы получили стандартную линейную модель множественной регрессии:

$$logit(p) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n .$$

Подобное уравнение нам уже знакомо. Решив его, мы получим значения регрессионных коэффициентов (табл. 2), по которым затем можно восстановить вероятность $p$.

Таблица 2. **Значения регрессионных коэффициентов**

| P | 1-P | $\frac{P}{1-P}$ | $\ln\frac{P}{1-P}$ |
|---|---|---|---|
| 0 | 1 | 0 | - ∞ |
| 0,2 | 0,8 | 0,25 | -1,39 |
| 0,4 | 0,6 | 0,67 | -0,40 |
| 0,5 | 0,5 | 1 | 0,00 |
| 0,6 | 0,4 | 1,33 | 0,29 |
| 0,8 | 0,8 | 4 | 1,39 |
| 1 | 0 | ∞ | ∞ |

Построим график зависимости $\ln\frac{P}{1-P}$ от $P$.

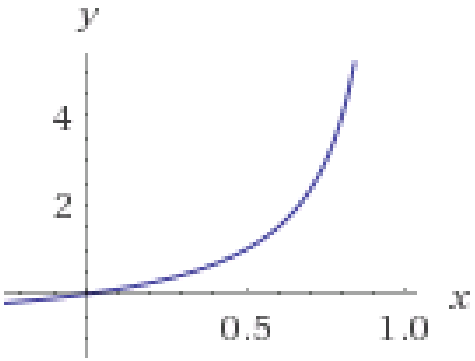



Рис. 8. График зависимости $\frac{P}{1-P}$ от P

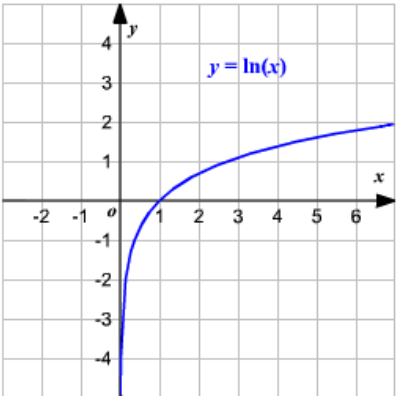



Рис. 9. График зависимости $\ln\frac{P}{1-P}$ от $\frac{P}{1-P}$

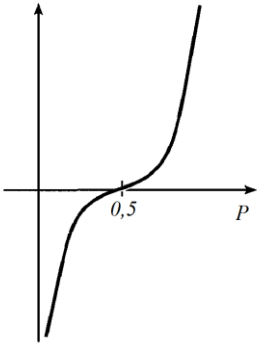



Рис. 10. График зависимости $\ln\frac{\mathrm{P}}{1-\mathrm{P}}$ от P

### 3.3 Интерпретация коэффициентов логистической регрессии

Отношение шансов – статистический показатель (на русском его название принято сокращать как ОШ, а на английском – OR от «odds ratio»), один из основных способов описать в численном выражении то, насколько отсутствие или наличие определённого исхода связано с присутствием или отсутствием определённого фактора в конкретной статистической группе [25].

Рассмотрим случай $logit(P) = \beta_0 + \beta_1 x$ . Пусть $x$ – категориальная переменная (характер дорожного покрытия), принимающая значения либо 0 (сухое покрытие), либо 1 (грязное или мокрое дорожное покрытие). Соответственно, $y$ будет принимать значения либо 0 (несерьезная авария), либо 1 (серьезная авария). Оформим результаты обработки данных в виде табл. 3.

Таблица 3. **Данные по авариям**

| y \ x | 0 | 1 | $\sum$ |
|---|---|---|---|
| 0 | 10 | 20 | 30 |
| 1 | 30 | 40 | 70 |
| $\sum$ | 40 | 60 | 100 |

При $x = 0$

$$logit(P) = \beta_0,$$

$$\beta_0 = \ln \frac{P(y = 1|x = 1)}{1 - P(y = 1|x = 1)}.$$

Вероятность того, что авария произошла на грязной или мокрой дороге и она серьезная

$$P(y = 1|x = 1) = \frac{40}{60} = 0{,}66,$$

тогда

$$\beta_0 = \ln \frac{P(y = 1|x = 1)}{1 - P(y = 1|x = 1)} = \ln \frac{0{,}66}{1 - 0{,}66} = \ln 2 = 0{,}69,$$

$\beta_1$ – отношение шансов возникновения аварий на грязной и мокрой дороге и на чистой дороге, т.е. $\beta_1 = \ln(OR)$

$$P(x = 0|y = 0) = \frac{10}{40} = 0{,}25;$$

$$P(x = 0|y = 1) = \frac{30}{40} = 0{,}75;$$

$$\frac{P(x = 0|y = 1)}{P(x = 0|y = 0)} = \frac{0{,}75}{0{,}25} = 3;$$

$\beta_1 = \ln\frac{2}{3} = -0{,}41$;

$logit(P) = 0{,}69 - 0{,}41x$.

Как видно из примера, если известна регрессионная модель, можно рассчитать вероятности наступления всех событий и определить OR возникновения событий в разных ситуациях.

В случае, если $x$ – непрерывная величина, то коэффициент $\beta$ представляет собой процентные значения роста переменной логистического преобразования $logit(P)$ при изменении переменной $x$ на единицу.

### 3.4 Оценка качества классификационных моделей

Оценить качество классификатора легко на основании таблицы контингентности, которая составляется для каждого класса отдельно [26].

Таблица 4. **Контингентность**

| Категория *i* | | Экспертная оценка | |
|---|---|---|---|
| | | Положительная | Отрицательная |
| Оценка системы | Положительная | TP | FP |
| | Отрицательная | FN | TN |

В таблице содержится информация, сколько раз система приняла верное и сколько раз неверное решение по данным заданного класса (например, отнесение аварии к тому или иному классу серьезности). А именно:

- TP – истинно-положительное решение;
- TN – истинно-отрицательное решение;
- FP – ложно-положительное решение;
- FN – ложно-отрицательное решение.

Существует 3 способа численной оценки качества алгоритма:

**1. Accuracy**

На основании таблицы контингентности может быть вычислен процент ошибок Accuracy:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}.$$

У этой метрики есть одна особенность, которую необходимо учитывать. Она присваивает всем данным одинаковый вес, что может быть не корректно в случае, если распределение данных в обучающей выборке сильно смещено в сторону какого-то одного или нескольких классов. В этом случае у классификатора есть больше информации по этим классам и соответственно в рамках этих классов он будет принимать более адекватные решения. На практике это приводит к тому, что вы имеете accuracy, скажем, 80%, но при этом в рамках какого-то конкретного класса классификатор работает плохо, не определяя правильно даже треть данных.

Один выход из этой ситуации заключается в том, чтобы обучать классификатор на специально подготовленной, сбалансированной выборке данных. Минус этого решения в том, что вы отбираете у классификатора информацию об относительной частоте данных. Эта информация при прочих равных может оказаться очень кстати для принятия правильного решения.

Другой выход заключается в использовании метрик Precision и Recall.

**2. Точность и полнота**

Точность (precision) и полнота (recall) являются метриками, которые используются при оценке большей части алгоритмов извлечения информации. Иногда они используются сами по себе, иногда в качестве базиса для производных метрик, таких как F-мера или R-Precision. Суть точности и полноты очень проста.

Точность системы в пределах класса – это доля данных (аварий) действительно принадлежащих данному классу относительно всех данных, которые система отнесла к этому классу. Полнота системы – это доля найденных классификатором данных, принадлежащих классу относительно всех данных этого класса в тестовой выборке.

Точность и полнота определяются следующим образом:

$$\text{Precision} = \frac{TP}{TP + FP},$$

$$\text{Recall} = \frac{TP}{TP + FN}.$$

Рассмотрим пример. Допустим, у вас есть тестовая выборка, в которой содержится 10 сообщений, 4 из них– спам. Обработав все сообщения, классификатор пометил 2 сообщения как спам, причем одно действительно является спамом, а второе было помечено в тестовой выборке как нормальное. Мы имеем одно истинно-положительное решение, три ложно-отрицательных и одно ложно-положительное. Тогда для класса “спам” точность классификатора составляет $\frac{1}{2}$ (50% положительных решений правильные), а полнота $\frac{1}{4}$ (классификатор нашел 25% всех спам-сообщений).

**3. F-мера**

Понятно, что чем выше точность и полнота, тем лучше. Но в реальной жизни максимальная точность и полнота недостижимы одновременно и приходится искать некий баланс. Поэтому, хотелось бы иметь некую метрику, которая объединяла бы в себе информацию о точности и полноте нашего алгоритма. Именно такой метрикой является F-мера.

F-мера представляет собой *гармоническое среднее* между точностью и полнотой. Она стремится к нулю, если точность или полнота стремится к нулю:

$$F = 2 \times \frac{Precision \times Recall}{Precision + Recall}.$$

Данная формула придает одинаковый вес точности и полноте, поэтому F-мера будет падать одинаково при уменьшении и точности, и полноты. Возможно рассчитать F-меру придав различный вес точности и полноте, если вы осознанно отдаете приоритет одной из этих метрик при разработке алгоритма:

$$F = (\beta^2 + 1)\frac{Precision \times Recall}{\beta^2 Precision + Recall},$$

где $\beta$ принимает значения в диапазоне $0 < \beta < 1$ если вы хотите отдать приоритет точности, а при $\beta > 1$ приоритет отдается полноте.

При $\beta = 1$ формула сводится к предыдущей и вы получаете сбалансированную F-меру (также ее называют $F_1$).

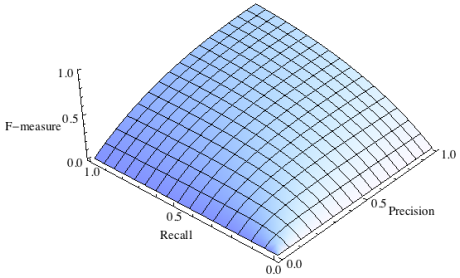



Рис. 11. Сбалансированная F-мера

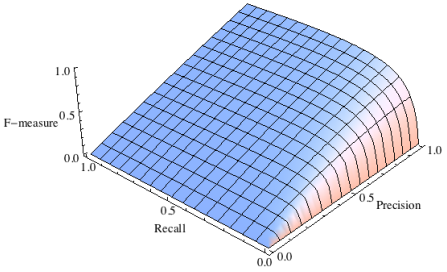



Рис. 12. F-мера с приоритетом точности ($\beta^2 = \frac{1}{4}$)

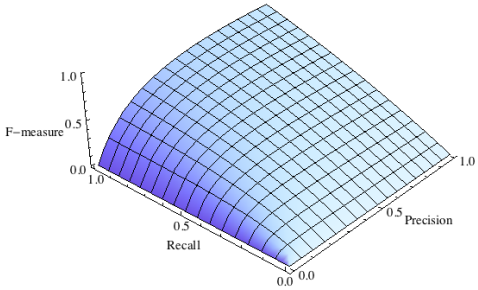



Рис. 13. F-мера с приоритетом полноты ($\beta^2 = 2$)

F-мера является хорошим кандидатом на формальную метрику оценки качества классификатора. Она сводит к одному числу две другие основополагающие метрики: точность и полноту.

# 4 ОСНОВЫ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

## 4.1 Биологический прототип

Развитие искусственных нейронных сетей вдохновляется биологией. То есть, рассматривая сетевые конфигурации и алгоритмы, исследователи применяют термины, заимствованные из принципов организации мозговой деятельности, однако на этом аналогия завершается. Знания о работе мозга столь ограничены, что мало бы нашлось точно доказанных закономерностей для тех, кто пожелал бы руководствоваться ими. Именно поэтому разработчикам сетей приходится выходить за пределы современных биологических знаний в поисках структур, способных выполнять полезные функции. Во многих случаях это приводит к необходимости отказа от биологического правдоподобия, мозг становится всего лишь метафорой, и создаются сети, невозможные в живой материи или требующие неправдоподобно больших допущений об анатомии и функционировании мозга.

Несмотря на то, что связь с биологией слаба и зачастую несущественна, искусственные нейронные сети продолжают сравнивать с мозгом. Их функционирование часто имеет внешнее сходство с человеческим познанием, поэтому трудно избежать этой аналогии. К сожалению, такие сравнения неплодотворны и создают неоправданные ожидания, неизбежно ведущие к разочарованию.

Нервная система человека построена из элементов, которые называют нейронами, имеет ошеломляющую сложность. Около 1011 нейронов участвуют в примерно 1015 передающих связях, имеющих длину метр и более. Каждый нейрон обладает многими свойствами, общими с другими органами тела, но ему присущи абсолютно уникальные способности: принимать, обрабатывать и передавать электрохимические сигналы по нервным путям, которые образуют коммуникационную систему мозга.

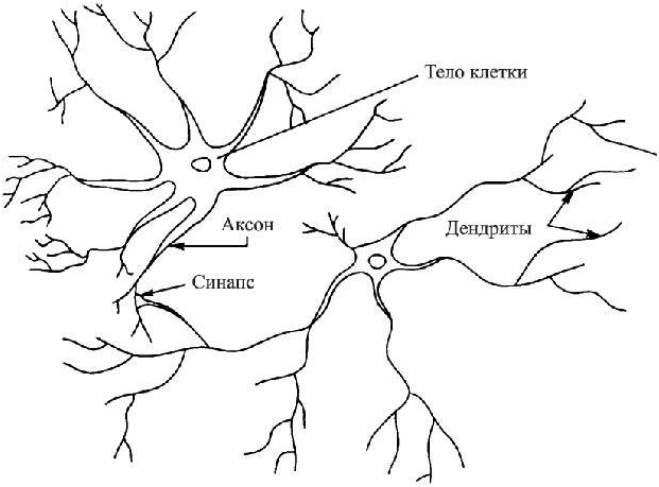



Рис. 14. Структура пары типичных биологических нейронов

Дендриты идут от тела нервной клетки к другим нейронам, где они принимают сигналы в точках соединения, называемых синапсами. Принятые синапсом входные сигналы передаются к телу нейрона. Здесь они суммируются, причем одни входы стремятся возбудить нейрон, другие – препятствуют его возбуждению.

Когда суммарное возбуждение в теле нейрона превышает некоторый порог, нейрон возбуждается, посылая по аксону сигнал другим нейронам. У этой основной функциональной схемы много усложнений и исключений, тем не менее, большинство искусственных нейронных сетей моделируют лишь эти простые свойства.

## 4.2 Искусственный нейрон

Искусственный нейрон имитирует в первом приближении свойства биологического нейрона. На вход искусственного нейрона поступает некоторое множество сигналов, каждый из которых является выходом другого нейрона. Каждый вход умножается на соответствующий вес, аналогичный синаптической силе, и все произведения суммируются, определяя уровень активации нейрона.

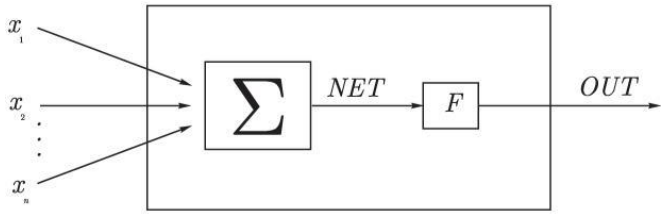



Рис. 15. «Сжимающая» функция

Множество входных сигналов, обозначенных *x*1, *x*2, ..., *xn*, поступает на искусственный нейрон. Эти входные сигналы, в совокупности обозначаемые вектором, соответствуют сигналам, приходящим в синапсы биологического нейрона. Каждый сигнал умножается на соответствующий вес и поступает на суммирующий блок. Каждый вес соответствует «силе» одной биологической синаптической связи. Множество весов в совокупности обозначается вектором. Суммирующий блок, соответствующий телу биологического элемента, складывает взвешенные входы алгебраически, создавая выход, который мы будем называть NET. В векторных обозначениях это может быть компактно записано следующим образом: NET=XW.

Сигнал NET далее, как правило, преобразуется активационной функцией F и дает выходной нейронный сигнал OUT. Активационная функция может быть обычной линейной функцией OUT=F(NET), где F – константа, пороговой функцией:

$$\text{OUT} = \begin{cases} 1, & \text{если } NET > T \\ 0, & \text{если } NET \leq T \end{cases},$$

где T – некоторая постоянная пороговая величина, или же функция, более точно моделирующая нелинейную передаточную характеристику биологического нейрона и предоставляющая нейронной сети большие возможности.

На рис. 15 блок, обозначенный F, принимает сигнал NET и выдает сигнал OUT. Если блок F сужает диапазон изменения величины NET так, что при любых значениях NET значения OUT при-

надлежат некоторому конечному интервалу, то F называется «сжимающей» функцией. В качестве «сжимающей» функции часто используется логистическая или «сигмоидальная» (S-образная) функция, показанная на рис. 16. Эта функция математически выражается как

$$F(x) = 1/(1+e^{-x}) .$$

Таким образом, $OUT = \frac{1}{1+ e^{-NET}}$ .

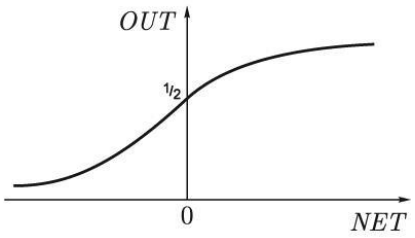



Рис. 16. «Сигмоидальная» функция

По аналогии с электронными системами активационную функцию можно считать нелинейной усилительной характеристикой искусственного нейрона. Коэффициент усиления вычисляется как отношение приращения величины OUT к вызвавшему его небольшому приращению величины NET.

Он выражается наклоном кривой при определенном уровне возбуждения и изменяется от малых значений при больших отрицательных возбуждениях (кривая почти горизонтальна) до максимального значения при нулевом возбуждении и снова уменьшается, когда возбуждение становится большим положительным. С. Гроссберг (1973) обнаружил, что подобная нелинейная характеристика решает поставленную им дилемму шумового насыщения. Каким образом одна и та же сеть может обрабатывать и слабые, и сильные сигналы? Слабые сигналы нуждаются в большом сетевом усилении, для того, чтобы дать пригодный к использованию выходной сигнал. Однако усилительные каскады с большими коэффициентами усиления могут привести к насыщению выхода шумами усилителей (слу-

чайными флуктуациями), которые присутствуют в любой физически реализованной сети. Сильные входные сигналы, в свою очередь, также будут приводить к насыщению усилительных каскадов и исключать возможность полезного использования выхода. Центральная область логистической функции, которая имеет большой коэффициент усиления, решает проблему обработки слабых сигналов, в то время как области с падающим усилением на положительном и отрицательном концах подходят для больших возбуждений. Таким образом, нейрон функционирует с большим усилением в широком диапазоне уровня входного сигнала:

$$\text{OUT} = \frac{1}{1 + e^{-NET}} = F(NET).$$

Другой, широко используемой активационной функцией является гиперболический тангенс. По форме она сходна с логистической функцией и часто используется биологами в качестве математической модели активации нервной клетки. В качестве активационной функции искусственной нейронной сети она записывается следующим образом: OUT=th(x).

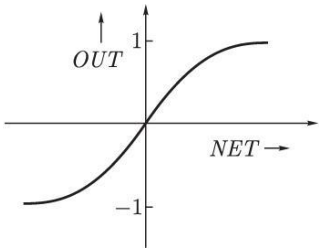



Рис. 17. Гиперболический тангенс

Подобно логистической функции гиперболический тангенс – S-образная функция, но он симметричен относительно начала координат и в точке NET=0 значение выходного сигнала OUT равно нулю (рис. 17). В отличие от логистической функции, гиперболический тангенс принимает значения различных знаков, и это его свойство применяется для целого ряда сетей.

Рассмотренная простая модель искусственного нейрона игнорирует многие свойства своего биологического двойника. Например, она не принимает во внимание задержки во времени, которые воздействуют на динамику системы. Входные сигналы сразу же порождают выходной сигнал. И, что более важно, она не учитывает воздействий функции частотной модуляции или синхронизирующей функции биологического нейрона, которые ряд исследователей считают решающими в нервной деятельности естественного мозга.

Несмотря на данные ограничения, сети, построенные из таких нейронов, обнаруживают свойства, которые сильно напоминают биологическую систему. Только время и исследования смогут ответить на вопрос, являются ли подобные совпадения случайными или же они – следствие того, что в модели верно схвачены важнейшие черты биологического нейрона.

## 4.3 Однослойные искусственные нейронные сети

Хотя один нейрон и способен выполнять простейшие процедуры распознавания, но для серьезных нейронных вычислений необходимо соединять нейроны в сети. Простейшая сеть состоит из группы нейронов, образующих слой, как показано в правой части рис. 18. Отметим, что вершины-круги слева служат лишь для распределения входных сигналов. Они не выполняют каких-либо вычислений и именно поэтому не считаются слоем. Для большей наглядности обозначим их кругами, чтобы отличать их от вычисляющих нейронов, обозначенных квадратами. Каждый элемент из множества $X$ входов отдельным весом соединен с каждым искусственным нейроном. А каждый нейрон выдает взвешенную сумму входов в сеть. В искусственных и биологических сетях многие соединения могут отсутствовать, но здесь они показаны все для демонстрации общей картины. Могут существовать также соединения между выходами и входами элементов в слое.

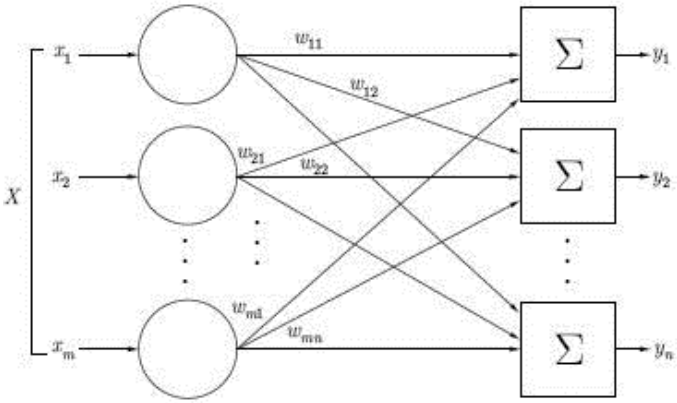



Рис. 18. Простейшая сеть

Удобно считать веса элементами матрицы *W*. Матрица имеет *m* строк и *n* столбцов, где *m* – число входов, а *n* – число нейронов. Например, *w*2,3 – это вес, связывающий второй вход с третьим нейроном. Таким образом, вычисление выходного вектора *N*, компонентами которого являются выходы OUT нейронов, сводится к матричному умножению *N*=*XW*, где *N* и *X* – векторы-строки.

### 4.4 Многослойные искусственные нейронные сети

Более крупные и сложные нейронные сети обладают, как правило, и большими вычислительными возможностями. Хотя созданы сети всех конфигураций, какие только можно себе представить, послойная организация нейронов копирует слоистые структуры определенных отделов мозга. Оказалось, что такие многослойные сети обладают большими возможностями, чем однослойные, и в последние годы были разработаны алгоритмы для их обучения. Многослойные сети могут строиться из каскадов слоев. Выход одного слоя является входом для последующего слоя. Подобная сеть показана на рис. 19 и снова изображена со всеми соединениями. Многослойные сети не могут привести к увеличению вычислительной мощности по сравнению с однослойной сетью, если активационная

функция между слоями линейна. Вычисление выхода слоя заключается в умножении входного вектора на первую весовую матрицу с последующим умножением (если отсутствует нелинейная активационная функция) результирующего вектора на вторую весовую матрицу

$$\text{OUT} = (XW_1)W2.$$

Так как умножение матриц ассоциативно, то

$$(XW_1)W2 = X(W_1\,W_2).$$

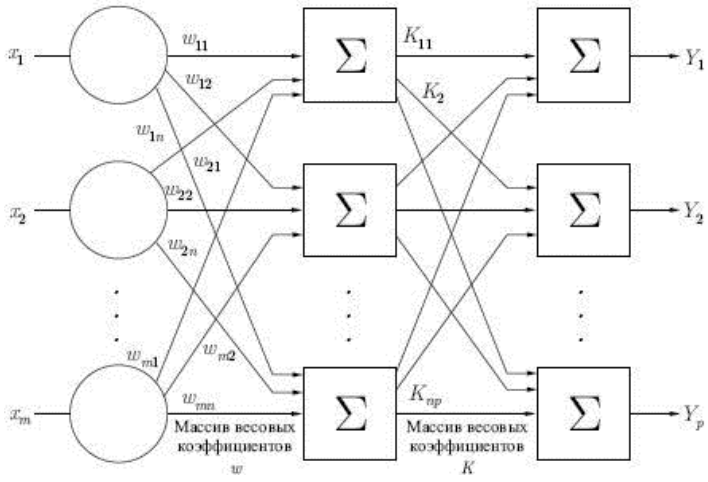



Рис. 19. Многослойная сеть

У рассмотренных сетей не было обратных связей, т.е. соединений, идущих от выходов некоторого слоя к входам этого же слоя или предшествующих слоев. Этот специальный класс сетей, называемый сетями без обратных связей или сетями прямого распространения, представляет большой интерес и широко используется. Сети более общего вида, имеющие соединения от выходов к входам, называются сетями с обратными связями. У сетей без обратных связей нет памяти, их выход полностью определяется текущими входами и значениями весов. В некоторых конфигурациях сетей с

обратными связями предыдущие значения выходов возвращаются на входы; выход, следовательно, определяется как текущим входом, так и предыдущими выходами. Поэтому сети с обратными связями могут обладать свойствами, сходными с кратковременной человеческой памятью, где сетевые выходы тоже частично зависят от предыдущих входов.

## 4.5 Обучение искусственных нейронных сетей

Среди всех интересных свойств искусственных нейронных сетей ни одно не захватывает так воображения, как их способность к обучению. Их обучение до такой степени напоминает процесс интеллектуального развития человеческой личности, что может показаться, будто нами достигнуто глубокое понимание этого процесса. Но, проявляя осторожность, следует сдерживать эйфорию. Возможности обучения искусственных нейронных сетей ограничены, и нужно решить много сложных задач, чтобы определить, находимся ли мы на правильном пути.

Сеть обучается, чтобы для некоторого множества входов давать желаемое (или, по крайней мере, сообразное с ним) множество выходов. Каждое такое входное (или выходное) множество рассматривается как вектор. Обучение осуществляется путем последовательного предъявления входных векторов с одновременной подстройкой весов в соответствии с определенной процедурой. В процессе обучения веса сети постепенно становятся такими, чтобы каждый входной вектор вырабатывал выходной вектор.

## 4.6 Обучение с учителем

Различают алгоритмы обучения с учителем и без учителя. Обучение с учителем предполагает, что для каждого входного вектора существует целевой вектор, представляющий собой требуемый выход. Вместе они называются обучающей парой. Обычно сеть обуча-

ется на некотором числе таких обучающих пар. Предъявляется выходной вектор, вычисляется выход сети и сравнивается с соответствующим целевым вектором, разность (ошибка) с помощью обратной связи подается в сеть, и веса изменяются в соответствии с алгоритмом, стремящимся минимизировать ошибку. Векторы обучающего множества предъявляются последовательно, ошибки вычисляются и веса подстраиваются для каждого вектора до тех пор, пока ошибка по всему обучающему массиву не достигнет приемлемо низкого уровня.

## 4.7 Обучение без учителя

Несмотря на многочисленные прикладные достижения, обучение с учителем критиковалось за свою биологическую неправдоподобность. Трудно вообразить обучающий механизм в мозге, который бы сравнивал желаемые и действительные значения выходов, выполняя коррекцию с помощью обратной связи. Обучение без учителя является намного более правдоподобной моделью обучения для биологической системы. Развитая Кохоненом и многими другими, она не нуждается в целевом векторе для выходов и, следовательно, не требует сравнения с предопределенными идеальными ответами. Обучающее множество состоит лишь из входных векторов. Обучающий алгоритм подстраивает веса сети так, чтобы получались согласованные выходные векторы, т.е. чтобы предъявление достаточно близких входных векторов давало одинаковые выходы. Процесс обучения, следовательно, выделяет статистические свойства обучающего множества и группирует сходные векторы в классы. Предъявление на вход вектора из данного класса даст определенный выходной вектор, но до обучения невозможно предсказать, какой выход будет производиться данным классом входных векторов. Следовательно, выходы подобной сети должны трансформироваться в некоторую понятную форму, обусловленную процессом обучения. Это не является серьезной проблемой. Обычно не сложно выявить связь между входом и выходом, установленную сетью.

## 4.8 Персептронная представляемость

Доказательство теоремы обучения персептрона показало, что персептрон способен научиться всему, что он способен представлять. Важно при этом уметь различать представляемость и обучаемость. Понятие представляемости относится к способности персептрона (или другой сети) моделировать определенную функцию. Обучаемость же требует наличия систематической процедуры настройки весов сети для реализации этой функции.

Для иллюстрации проблемы представляемости допустим, что у нас есть множество карт, помеченных цифрами от 0 до 9. Допустим также, что мы обладаем гипотетической машиной, способной отличать карты с нечетным номером от карт с четным номером и зажигающей индикатор на своей панели при предъявлении карты с нечетным номером. Представима ли такая машина персептроном? То есть возможно ли сконструировать персептрон и настроить его веса (неважно, каким образом) так, чтобы он обладал такой же разделяющей способностью? Если это достижимо, то говорят, что персептрон способен представлять желаемую машину. Мы увидим, что возможности представления однослойными персептронами весьма ограниченны. Имеется много простых машин, которые не могут быть представлены персептроном, независимо от того, как настраиваются его веса.

Один из самых пессимистических результатов М.Л. Минского гласит, что однослойный персептрон не может воспроизвести такую простую функцию, как исключающее или. Это функция от двух аргументов, каждый из которых может быть нулем или единицей. Она принимает значение единицы, когда один из аргументов равен единице (но не оба).

## 4.9 Линейная разделимость

Имеется обширный класс функций, не реализуемых однослойной сетью. Об этих функциях говорят, что они являются линейно

неразделимыми: они-то и накладывают определенные ограничения на возможности однослойных сетей.

Линейная разделимость ограничивает однослойные сети задачами классификации, в которых множества точек (соответствующих входным значениям) могут быть разделены геометрически. Для нашего случая с двумя входами разделитель является прямой линией. В случае трех входов разделение осуществляется плоскостью, рассекающей трехмерное пространство. Для четырех или более входов визуализация невозможна, и необходимо мысленно представить -мерное пространство, рассекаемое «гиперплоскостью» – геометрическим объектом, который делит пространство четырех или большего числа измерений.

Так как линейная разделимость ограничивает возможности персептронного представления, то важно знать, является ли данная функция разделимой. К сожалению, не существует простого способа определить это, если число переменных велико.

### 4.10 Преодоление ограничения линейной разделимости

К концу 1960-х годов проблема линейной разделимости была хорошо понята. К тому же, было известно, что это серьезное ограничение представляемости однослойными сетями можно преодолеть, добавив дополнительные слои. Например, двухслойные сети можно получить каскадным соединением двух однослойных сетей. Они способны выполнять более общие классификации, отделяя те точки, которые содержатся в выпуклых ограниченных или неограниченных областях. Область называется выпуклой, если для любых двух ее точек соединяющий их отрезок целиком лежит в области. Область называется ограниченной, если ее можно заключить в некоторый круг. Неограниченную область невозможно заключить внутрь круга (например, область между двумя параллельными линиями). Примеры выпуклых ограниченных и неограниченных областей представлены на рис. 20.

Чтобы уточнить требование выпуклости, рассмотрим простую двуслойную сеть с двумя входами, которые подведены к двум нейронам первого слоя, соединенными с единственным нейроном в

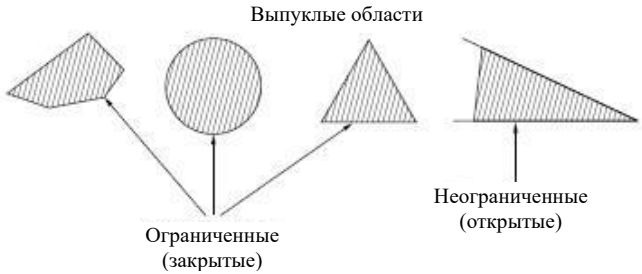
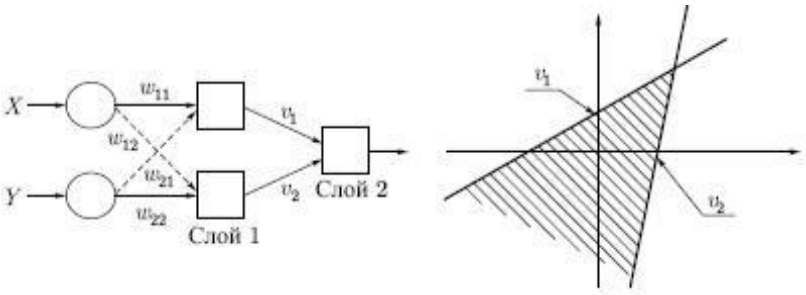



Рис. 20. Выпуклые ограниченные и неограниченные области

слое 2 (см. рис. 21,а). Пусть порог выходного нейрона равен 0,75, а оба его веса равны 0,5. В этом случае для того, чтобы порог был превышен и на выходе появилась единица, требуется, чтобы оба нейрона первого уровня на выходе имели единицу. Таким образом, выходной нейрон реализует логическую функцию И. На рис. 21,а каждый нейрон слоя 1 разбивает плоскость XOY на две полуплоскости, один обеспечивает единичный выход для входов ниже верхней линии, другой – для входов выше нижней линии. На рис. 21,б показан результат такого двойного разбиения, где выходной сигнал нейрона второго слоя равен единице только внутри V-образной области. Аналогично, во втором слое может быть использовано три нейрона с дальнейшим разбиением плоскости и созданием области треугольной формы. Включением достаточного числа нейронов во входной слой может быть образован выпуклый многоугольник любой желаемой формы. Все такие многогранники выпуклы, так как они образованы с помощью операции и над областями, задаваемыми линиями: следовательно, только выпуклые области и возникают. Точки, не составляющие выпуклой области, не могут быть отделены от других точек плоскости двухслойной сетью.

Нейрон второго слоя не ограничен функцией. Он может реализовывать многие другие функции при подходящем выборе весов и порога. Например, можно сделать так, чтобы единичный выход любого из нейронов первого слоя приводил к появлению единицы на выходе нейрона второго слоя, реализовав тем самым логическое ИЛИ. Например, имеется 16 двоичных функций от двух переменных. Если выбирать подходящим образом веса и порог, то

можно воспроизвести 14 из них (все, кроме «исключающее или» и «исключающее нет»).




а б

Рис. 21. Выходной нейрон

Входы не обязательно должны быть двоичными. Вектор непрерывных входов может представлять собой произвольную точку на плоскости XOY. В этом случае мы имеем дело со способностью сети разбивать плоскость на непрерывные области, а не с разделением дискретных множеств точек.

Для всех этих функций, однако, линейная разделимость показывает, что выход нейрона второго слоя равен единице только в части плоскости XOY, ограниченной многоугольной областью. Поэтому для разделения плоскостей P и Q необходимо, чтобы все P лежали внутри выпуклой многоугольной области, не содержащей точек Q (или наоборот).

Трехслойная сеть, впрочем, есть более общий случай. Ее классифицирующие возможности ограничены лишь числом искусственных нейронов и весов. Ограничения на выпуклость отсутствуют. Теперь нейрон третьего слоя принимает в качестве входа набор выпуклых многоугольников, и их логическая комбинация может быть невыпуклой. На рис. 22,б иллюстрируется ситуация, когда два треугольника A и B, скомбинированные с помощью функций «A и не B», задают невыпуклую область. При добавлении нейронов и весов число сторон многоугольников может неограниченно возрастать. Это позволяет аппроксимировать область любой формы с любой точностью. Вдобавок, не все выход-

ные области второго слоя должны пересекаться. Возможно, следовательно, объединять различные области, выпуклые и невыпуклые, выдавая на выходе единицу всякий раз, когда входной вектор принадлежит одной из них.

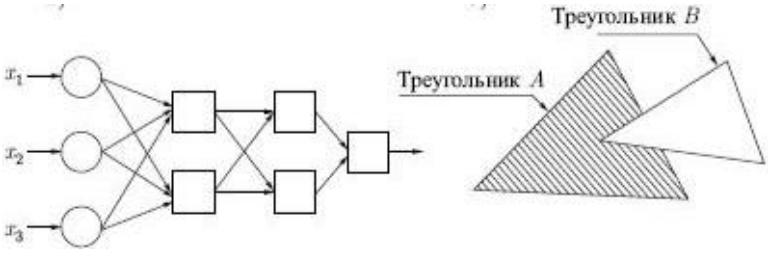



Рис. 22. Невыпуклая область

# 5 КЛАСТЕРИЗАЦИЯ

Кластерный анализ (англ. Cluster analysis) – многомерная статистическая процедура, выполняющая сбор данных, содержащих информацию о выборке объектов, и затем упорядочивающая объекты в сравнительно однородные группы. Задача кластеризации относится к статистической обработке, а также к широкому классу задач обучения без учителя [30].

Большинство исследователей склоняются к тому, что впервые термин «кластерный анализ» (англ. Cluster – гроздь, сгусток, пучок) был предложен математиком Р. Трионом. Впоследствии возник ряд терминов, которые в настоящее время принято считать синонимами термина «кластерный анализ»: автоматическая классификация, ботриология.

Спектр применений кластерного анализа очень широк: его используют в археологии, медицине, психологии, химии, биологии, государственном управлении, филологии, антропологии, маркетинге, социологии и других дисциплинах. Однако универсальность применения привела к появлению большого количества несовместимых терминов, методов и подходов, затрудняющих однозначное использование и непротиворечивую интерпретацию кластерного анализа.

## 5.1 Задачи и условия

Кластерный анализ выполняет следующие основные задачи:

- разработка типологии или классификации;
- исследование полезных концептуальных схем группирования объектов;
- порождение гипотез на основе исследования данных;
- проверка гипотез или исследования для определения, действительно ли типы (группы), выделенные тем или иным способом, присутствуют в имеющихся данных.

Независимо от предмета изучения применение кластерного анализа предполагает следующие этапы:

- отбор выборки для кластеризации. Подразумевается, что имеет смысл кластеризовать только количественные данные;
- определение множества переменных, по которым будут оцениваться объекты в выборке, то есть признакового пространства;
- вычисление значений той или иной меры сходства (или различия) между объектами;
- применение метода кластерного анализа для создания групп сходных объектов;
- проверка достоверности результатов кластерного решения.

Можно встретить описание двух фундаментальных требований предъявляемых к данным – однородность и полнота. Однородность требует, чтобы все кластеризуемые сущности были одной природы, описывались сходным набором характеристик. Если кластерному анализу предшествует факторный анализ, то выборка не нуждается в «ремонте» – изложенные требования выполняются автоматически самой процедурой факторного моделирования (есть ещё одно достоинство – z-стандартизация без негативных последствий для выборки; если её проводить непосредственно для кластерного анализа, она может повлечь за собой уменьшение чёткости разделения групп). В противном случае выборку нужно корректировать.

Применение кластерного анализа в общем виде сводится к следующим этапам [31]:

- отбор выборки объектов для кластеризации;
- определение множества переменных, по которым будут оцениваться объекты в выборке. При необходимости – нормализация значений переменных;
- вычисление значений меры сходства между объектами;
- применение метода кластерного анализа для создания групп сходных объектов (кластеров);
- представление результатов анализа;
- после получения и анализа результатов возможна корректировка выбранной метрики и метода кластеризации до получения оптимального результата.

## 5.2 Меры расстояний

Итак, как же определять «похожесть» объектов? Для начала нужно составить вектор характеристик для каждого объекта – как правило, это набор числовых значений, например, рост-вес человека. Однако существуют также алгоритмы, работающие с качественными (т.е. категорийными) характеристиками.

После того, как мы определили вектор характеристик, можно провести нормализацию, чтобы все компоненты давали одинаковый вклад при расчете «расстояния». В процессе нормализации все значения приводятся к некоторому диапазону, например, [-1,-1] или [0,1].

Наконец, для каждой пары объектов измеряется «расстояние» между ними – степень похожести. Существует множество метрик, вот лишь основные из них:

1) евклидово расстояние – наиболее распространенная функция расстояния. Представляет собой геометрическое расстояние в многомерном пространстве:

$$\rho(x, x') = \sqrt{\sum_{i}^{n} (x_i - x_i)^2};$$

2) квадрат евклидова расстояния применяется для придания большего веса более отдаленным друг от друга объектам. Это расстояние вычисляется следующим образом:

$$\rho(x, x') = \sum_{i}^{n} (x_i - x_i)^2;$$

3) расстояние городских кварталов (манхэттенское расстояние). Это расстояние является средним разностей по координатам. В большинстве случаев эта мера расстояния приводит к таким же результатам, как и для обычного расстояния Евклида. Однако для этой меры влияние отдельных больших разностей (выбросов) уменьшается (т.к. они не возводятся в квадрат).

Формула для расчета манхэттенского расстояния:

$$\rho(x, x') = \sum_{i}^{n} |x_i - x_i|;$$

5) расстояние Чебышева может оказаться полезным, когда нужно определить два объекта как «различные», если они различаются по какой-либо одной координате. Расстояние Чебышева вычисляется по формуле:

$$\rho(x, x') = max(|x_i - x_i|);$$

6) степенное расстояние применяется в случае, когда необходимо увеличить или уменьшить вес, относящийся к размерности, для которой соответствующие объекты сильно отличаются. Степенное расстояние вычисляется по следующей формуле:

$$\rho(x, x') = \sqrt[r]{\sum_{i}^{n}(x_i - x_i)^p},$$

где $r$ и $p$ – параметры, определяемые пользователем. Параметр $p$ ответственен за постепенное взвешивание разностей по отдельным координатам, параметр $r$ ответственен за прогрессивное взвешивание больших расстояний между объектами. Если оба параметра – $r$ и $p$ – равны двум, то это расстояние совпадает с расстоянием Евклида.

Выбор метрики полностью лежит на исследователе, поскольку результаты кластеризации могут существенно отличаться при использовании разных мер.

## 5.3 Классификация алгоритмов

Выделяют две основные классификации алгоритмов кластеризации:

- иерархические;
- плоские.

Иерархические алгоритмы (также называемые алгоритмами таксономии) строят не одно разбиение выборки на непересекающиеся кластеры, а систему вложенных разбиений. Таким образом на

выходе мы получаем дерево кластеров, корнем которого является вся выборка, а листьями – наиболее мелкие кластеры.

Плоские алгоритмы строят одно разбиение объектов на кластеры:

- четкие;
- нечеткие.

Четкие (или непересекающиеся) алгоритмы каждому объекту выборки ставят в соответствие номер кластера, т.е. каждый объект принадлежит только одному кластеру. Нечеткие (или пересекающиеся) алгоритмы каждому объекту ставят в соответствие набор вещественных значений, показывающих степень отношения объекта к кластерам. Т.е. каждый объект относится к каждому кластеру с некоторой вероятностью.

## 5.4 Типология задач кластеризации

### *5.4.1 Типы входных данных*

Различают следующие типы входных данных:

- признаковое описание объектов. Каждый объект описывается набором своих характеристик, называемых признаками. Признаки могут быть числовыми или нечисловыми;
- матрица расстояний между объектами. Каждый объект описывается расстояниями до всех остальных объектов метрического пространства;
- матрица сходства между объектами. Учитывается степень сходства объекта с другими объектами выборки в метрическом пространстве. Сходство здесь дополняет расстояние (различие) между объектами до 1.

В современной науке применяется несколько алгоритмов обработки входных данных. Анализ путём сравнения объектов, исходя из признаков, (наиболее распространённый в биологических науках) называется Q-типом анализа, а в случае сравнения признаков, на основе объектов – R-типом анализа. Существуют попытки использования гибридных типов анализа (например, RQ-анализ), но данная методология ещё должным образом не разработана.

### *5.4.2 Цели кластеризации*

- понимание данных путём выявления кластерной структуры. Разбиение выборки на группы схожих объектов позволяет упростить дальнейшую обработку данных и принятия решений, применяя к каждому кластеру свой метод анализа (стратегия «разделяй и властвуй»);
- сжатие данных. Если исходная выборка избыточно большая, то можно сократить её, оставив по одному наиболее типичному представителю от каждого кластера;
- обнаружение новизны (англ. Novelty detection). Выделяются нетипичные объекты, которые не удаётся присоединить ни к одному из кластеров.

В первом случае число кластеров стараются сделать поменьше. Во втором случае важнее обеспечить высокую степень сходства объектов внутри каждого кластера, а кластеров может быть сколько угодно. В третьем случае наибольший интерес представляют отдельные объекты, не вписывающиеся ни в один из кластеров.

Во всех этих случаях может применяться иерархическая кластеризация, когда крупные кластеры дробятся на более мелкие, те в свою очередь дробятся ещё мельче, и т.д. Такие задачи называются задачами таксономии. Результатом таксономии является древообразная иерархическая структура. При этом каждый объект характеризуется перечислением всех кластеров, которым он принадлежит, обычно от крупного к мелкому.

## 5.5 Объединение кластеров

Как вычислять «расстояния» между ними. Существует несколько метрик:

1. Одиночная связь (расстояния ближайшего соседа). В этом методе расстояние между двумя кластерами определяется расстоянием между двумя наиболее близкими объектами (ближайшими соседями) в различных кластерах. Результирующие кластеры имеют тенденцию объединяться в цепочки.

2. Полная связь (расстояние наиболее удаленных соседей). В этом методе расстояния между кластерами определяются наибольшим расстоянием между любыми двумя объектами в различных кластерах (т.е. наиболее удаленными соседями). Этот метод обычно работает очень хорошо, когда объекты происходят из отдельных групп. Если же кластеры имеют удлиненную форму или их естественный тип является «цепочечным», то этот метод непригоден.

3. Невзвешенное попарное среднее. В этом методе расстояние между двумя различными кластерами вычисляется как среднее расстояние между всеми парами объектов в них. Метод эффективен, когда объекты формируют различные группы, однако он работает одинаково хорошо и в случаях протяженных («цепочечного» типа) кластеров.

4. Взвешенное попарное среднее. Метод идентичен методу невзвешенного попарного среднего, за исключением того, что при вычислениях размер соответствующих кластеров (т.е. число объектов, содержащихся в них) используется в качестве весового коэффициента. Поэтому данный метод должен быть использован, когда предполагаются неравные размеры кластеров.

5. Невзвешенный центроидный метод. В этом методе расстояние между двумя кластерами определяется как расстояние между их центрами тяжести.

6. Взвешенный центроидный метод (медиана). Этот метод идентичен предыдущему, за исключением того, что при вычислениях используются веса для учета разницы между размерами кластеров. Поэтому, если имеются или подозреваются значительные отличия в размерах кластеров, этот метод оказывается предпочтительнее предыдущего.

## 5.6 Методы кластеризации

Общепринятой классификации методов кластеризации не существует, но можно выделить ряд групп подходов (некоторые методы можно отнести сразу к нескольким группам и потому предлагается рассматривать данную типизацию как некоторое приближение к реальной классификации методов кластеризации) [32]:

1. Вероятностный подход. Предполагается, что каждый рассматриваемый объект относится к одному из k классов. Некоторые авторы (например, А. И. Орлов) считают, что данная группа вовсе не относится к кластеризации и противопоставляют её под названием «дискриминация», то есть выбор отнесения объектов к одной из известных групп (обучающих выборок):

- K-средних (K-means);
- K-medians;
- EM-алгоритм;
- алгоритмы семейства FOREL;
- дискриминантный анализ.

2. Подходы на основе систем искусственного интеллекта: весьма условная группа, так как методов очень много и методически они весьма различны:

- метод нечеткой кластеризации C-средних (C-means);
- нейронная сеть Кохонена;
- генетический алгоритм.

3. Логический подход. Построение дендрограммы осуществляется с помощью дерева решений.

- теоретико-графовый подход.

4. Графовые алгоритмы кластеризации.

5. Иерархический подход. Предполагается наличие вложенных групп (кластеров различного порядка). Алгоритмы в свою очередь подразделяются на агломеративные (объединительные) и дивизивные (разделяющие). По количеству признаков иногда выделяют монотетические и политетические методы классификации:

- иерархическая дивизивная кластеризация или таксономия. Задачи кластеризации рассматриваются в количественной таксономии.

6. Другие методы. Не вошедшие в предыдущие группы:

- статистические алгоритмы кластеризации;
- ансамбль кластеризаторов;
- алгоритмы семейства KRAB;
- алгоритм, основанный на методе просеивания;
- DBSCAN и др.

Подходы 4 и 5 иногда объединяют под названием структурного или геометрического подхода, обладающего большей формализованностью понятия близости. Несмотря на значительные различия между перечисленными методами все они опираются на исходную «гипотезу компактности»: в пространстве объектов все близкие объекты должны относиться к одному кластеру, а все различные объекты соответственно должны находиться в различных кластерах.

**Метод k-средних**

Метод k-средних – это метод кластерного анализа, цель которого является разделение $m$ наблюдений (из пространства $R^n$) на $k$ кластеров, при этом каждое наблюдение относится к тому кластеру, к центру (центроиду) которого оно ближе всего.

В качестве меры близости используется Евклидово расстояние:

$$\rho(x, y) = ||x - y|| = \sqrt{\sum_{p=1}^{n}(x_p - y_p)^2},$$

где $x, y \in R^n$.

Итак, рассмотрим ряд наблюдений $\left(x^{(1)}, x^{(2)}, \dots, x^{(m)}\right)$, $x^{(j)} \in R^n$.

Метод k-средних разделяет $m$ наблюдений на $k$ групп (или кластеров) ($k \leq m$) $S = \{S_1, S_2, \dots, S_k\}$, чтобы минимизировать суммарное квадратичное отклонение точек кластеров от центроидов этих кластеров:

$$min\left[\sum_{i=1}^{k}\sum_{x^{(j)} \in S_i} \left|\left|x^{(j)} - \mu_i\right|\right|^2\right], \text{ где } x^{(j)} \in R^n, \mu_i \in R^n,$$

где $\mu_i$ – центроид для кластера $S_i$.

**Алгоритм**

Итак, если мера близости до центроида определена, то разбиение объектов на кластеры сводится к определению центроидов этих кластеров. Число кластеров $k$ задается исследователем заранее.

Рассмотрим первоначальный набор $k$ средних (центроидов) $\mu_1, \dots, \mu_k$ в кластерах $S_1, S_2, \dots, S_k$. На первом этапе центроиды кластеров выбираются случайно или по определенному правилу (например, выбрать центроиды, максимизирующие начальные расстояния между кластерами).

Относим наблюдения к тем кластерам, чье среднее (центроид) к ним ближе всего. Каждое наблюдение принадлежит только к одному кластеру, даже если его можно отнести к двум и более кластерам.

Затем центроид каждого i-го кластера вычисляется по следующему правилу:

$$\mu_i = \frac{1}{S_j} \sum_{x^j \in S_i} x^{(j)}.$$

Таким образом, алгоритм k-средних заключается в перевычислении на каждом шаге центроида для каждого кластера, полученного на предыдущем шаге.

Алгоритм останавливается, когда значения $\mu_i$ не меняются:

$$\mu_i^{mar\ t} = \mu_i^{mar\ t+1}.$$

Важно: неправильный выбор первоначального числа кластеров $k$ может привести к некорректным результатам. Именно поэтому при использовании метода k-средних важно сначала провести проверку подходящего числа кластеров для данного набора данных.

Итак, еще раз подчеркнем некоторые особенности метода k-средних:

- в качестве метрики используется Евклидово расстояние;
- число кластеров заранее не известно и выбирается исследователем заранее;
- качество кластеризации зависит от первоначального разбиения.

**Демонстрация алгоритма**

Действие алгоритма в двумерном случае. Начальные точки выбраны случайно.

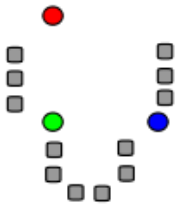


Рис. 23.

Исходные точки и случайно выбранные начальные точки.

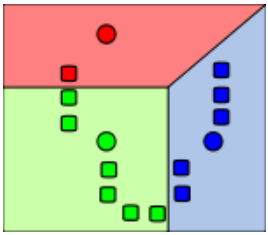


Рис. 24.

Точки, отнесённые к начальным центрам. Разбиение на плоскости – диаграмма Вороного относительно начальных центров.

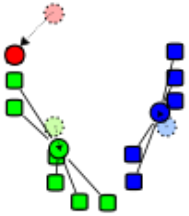


Рис. 25.

Вычисление новых центров кластеров (Ищется центр масс).

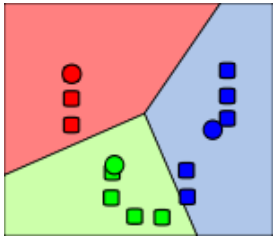


Рис. 26.

Предыдущие шаги повторяются, пока алгоритм не сойдётся.

**Применение для задач глубокого обучения и машинного зрения**

В алгоритмах глубокого обучения метод k-средних иногда применяют не по прямому назначению (классификация разбивкой на кластеры), а для создания так называемых фильтров (ядер свёртки, словарей). Например, для распознавания изображений в алгоритм k-средних подают небольшие случайные кусочки изображений обучающей выборки, допустим, размером 16x16 в виде линейного вектора, каждый элемент которого кодирует яркость своей точки. Количество кластеров k задается большим, например 256. Обученный метод k-средних при определенных условиях вырабатывает при этом центры кластеров (центроиды), которые представляют собой удобные базисы, на которые можно разложить любое входное изображение. Такие «обученные» центроиды в дальнейшем используют в качестве фильтров, например для свёрточной нейронной сети в качестве ядер свёртки или других аналогичных систем машинного зрения. Таким образом осуществляется обучение без учителя при помощи метода k-средних.

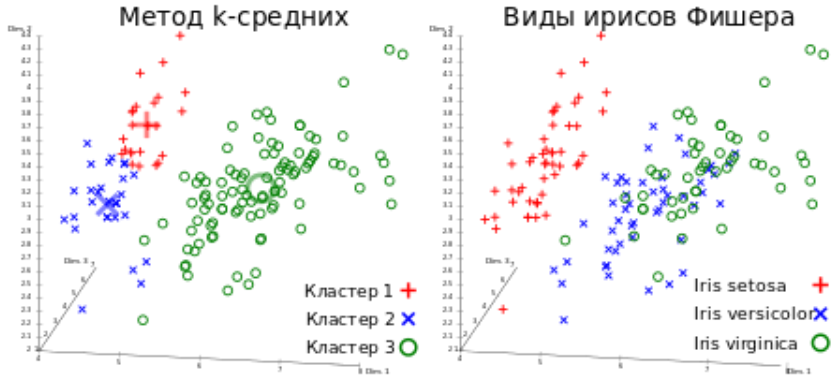



Рис. 27. Результат кластеризации методом k-средних для ирисов Фишера и реальные виды ирисов, визуализированные с помощью ELKI

Центры кластеров отмечены с помощью крупных, полупрозрачных маркеров.

**EM-алгоритм**

EM-алгоритм (англ. Expectation-maximization (EM) algorithm) – алгоритм, используемый в математической статистике для нахождения оценок максимального правдоподобия параметров вероятностных моделей, в случае, когда модель зависит от некоторых скрытых переменных. Каждая итерация алгоритма состоит из двух шагов. На E-шаге (expectation) вычисляется ожидаемое значение функции правдоподобия, при этом скрытые переменные рассматриваются как наблюдаемые. На M-шаге (maximization) вычисляется оценка максимального правдоподобия, таким образом увеличивается ожидаемое правдоподобие, вычисляемое на E-шаге. Затем это значение используется для E-шага на следующей итерации. Алгоритм выполняется до сходимости.

Часто EM-алгоритм используют для разделения смеси гауссиан.

Пусть $X$ – некоторые из значений наблюдаемых переменных, а $T$ – скрытые переменные. Вместе $X$ и $T$ образуют полный набор данных. Вообще, $T$ может быть некоторой подсказкой, которая облегчает решение проблемы в случае, если она известна. Например, если имеется смесь распределений, функция правдоподобия легко выражается через параметры отдельных распределений смеси.

Положим $p$ – плотность вероятности (в непрерывном случае) или функция вероятности (в дискретном случае) полного набора данных с параметрами $\Theta$: $p(X,T|\Theta)$. Эту функцию можно понимать как правдоподобие всей модели, если рассматривать её как функцию параметров $\Theta$. Заметим, что условное распределение скрытой компоненты при некотором наблюдении и фиксированном наборе параметров может быть выражено так:

$$\mathrm{p}(\mathrm{T}|\mathrm{X},\Theta) =$$

$$= \frac{\mathrm{p}(\mathrm{T}|\mathrm{X},\Theta)}{\mathrm{p}(\mathrm{X}|\Theta)} = \frac{\mathrm{p}(\mathrm{X},\mathrm{T}|\Theta)\mathrm{p}(\mathrm{T}|\Theta)}{\int(\Theta\,\check{\mathrm{T}},\Theta)p\left(\check{\mathrm{T}}\middle|\Theta\right)d\check{\mathrm{T}}},$$

используя расширенную формулу Байеса и формулу полной вероятности. Таким образом, нам необходимо знать только распределение наблюдаемой компоненты при фиксированной скрытой $p(X|T,\Theta)$ и вероятности скрытых данных $p(T|\Theta)$

EM-алгоритм итеративно улучшает начальную оценку $\Theta_0$, вычисляя новые значения оценок $\Theta_1$, $\Theta_2$ и так далее. На каждом шаге переход к $\Theta_{n+1}$ от $\Theta_n$ выполняется следующим образом:

$$\Theta_{n+1} = \arg max Q(\Theta),$$

где $Q(\Theta)$ – матожидание логарифма правдоподобия.

Другими словами, мы не можем сразу вычислить точное правдоподобие, но по известным данным ($X$) мы можем найти апостериорную оценку вероятностей для различных значений скрытых переменных $T$. Для каждого набора значений $T$ и параметров $\Theta$ мы можем вычислить матожидание функции правдоподобия по данному набору $X$. Оно зависит от предыдущего значения $\Theta$, потому что это значение влияет на вероятности скрытых переменных $T$.

$Q(\Theta)$ вычисляется следующим образом:

$$Q(\Theta) = E_T\,[\log p\,(X, T|\Theta)|X],$$

то есть это условное матожидание $\log p\,(X, T|\Theta)$ при условии $\Theta$.

Другими словами, $\Theta_{n+1}$ – это значение, максимизирующее (M) условное матожидание (E) логарифма правдоподобия при данных значениях наблюдаемых переменных и предыдущем значении параметров. В непрерывном случае значение $Q(\Theta)$ вычисляется так:

$$\mathrm{Q}(\Theta) = E_T\,[\log p\,(X, T|\Theta)|X] = \int_{\infty}^{-\infty} \mathrm{p}(\mathrm{T}|\mathrm{X}, \Theta_n) \log p\,(X, T|\Theta) dT.$$

## 5.7 Формальная постановка задачи кластеризации

Пусть $X$ – множество объектов, $Y$ – множество номеров (имён, меток) кластеров. Задана функция расстояния между объектами $p(x, x')$. Имеется конечная обучающая выборка объектов $X^m = \{x_1, \ldots, x_m\} \subset X$. Требуется разбить выборку на непересекающиеся подмножества, называемые кластерами, так, чтобы каждый кластер состоял из объектов, близких по метрике $p$, а объекты разных кластеров существенно отличались. При этом каждому объекту $x_i \in X^m$ приписывается номер кластера $y_i$.

Алгоритм кластеризации – это функция $a: X \to Y$, которая любому объекту $x \in X$ ставит в соответствие номер кластера $y \in Y$. Множество $Y$ в некоторых случаях известно заранее, однако чаще ставится задача определить оптимальное число кластеров, с точки зрения того или иного критерия качества кластеризации.

Кластеризация отличается от классификации тем, что метки исходных объектов $y_i$ изначально не заданы, и даже может быть неизвестно само множество $Y$.

Решение задачи кластеризации принципиально неоднозначно, и тому есть несколько причин (как считает ряд авторов):

- не существует однозначно наилучшего критерия качества кластеризации. Известен целый ряд эвристических критериев, а также ряд алгоритмов, не имеющих чётко выраженного критерия, но осуществляющих достаточно разумную кластеризацию «по построению». Все они могут давать разные результаты. Следовательно, для определения качества кластеризации требуется эксперт предметной области, который бы мог оценить осмысленность выделения кластеров;
- число кластеров, как правило, неизвестно заранее и устанавливается в соответствии с некоторым субъективным критерием. Это справедливо только для методов дискриминации, так как в методах кластеризации выделение кластеров идёт за счёт формализованного подхода на основе мер близости;
- результат кластеризации существенно зависит от метрики, выбор которой, как правило, также субъективен и определяется экспертом. Но стоит отметить, что есть ряд рекомендаций к выбору мер близости для различных задач.

## 5.8 Обзор алгоритмов

### *5.8.1 Алгоритмы иерархической кластеризации*

Среди алгоритмов иерархической кластеризации выделяются два основных типа: восходящие и нисходящие алгоритмы. Нисходящие алгоритмы работают по принципу «сверху-вниз»: все объекты помещаются в один кластер, который затем разбивается на все

более мелкие кластеры. Более распространены восходящие алгоритмы, которые в начале работы помещают каждый объект в отдельный кластер, а затем объединяют кластеры во все более крупные, пока все объекты выборки не будут содержаться в одном кластере. Таким образом строится система вложенных разбиений. Результаты таких алгоритмов обычно представляют в виде дерева – дендрограммы. Классический пример такого дерева – классификация животных и растений.

**Дендрограмма**

Под дендрограммой обычно понимается дерево, то есть граф без циклов, построенный по матрице мер близости. Дендрограмма позволяет изобразить взаимные связи между объектами из заданного множества. Для создания дендрограммы требуется матрица сходства (или различия), которая определяет уровень сходства между парами объектов. Чаще используются агломеративные методы.

Далее необходимо выбрать метод построения дендрограммы, который определяет способ пересчёта матрицы сходства (различия) после объединения (или разделения) очередных двух объектов в кластер.

В работах по кластерному анализу описан довольно внушительный ряд способов построения (англ. *Sorting strategies*) дендрограмм:

1. Метод одиночной связи (англ. *Single linkage*). Также известен, как «метод ближайшего соседа».
2. Метод полной связи (англ. *Complete linkage*). Также известен, как «метод дальнего соседа».
3. Метод средней связи (англ. *Pair-group method using arithmetic averages*):
   - невзвешенный (англ. *Unweighted*);
   - взвешенный (англ. *Weighted*).
4. Центроидный метод (англ. *Pair-group method using the centroid average*):
   - невзвешенный;
   - взвешенный (медианный).
5. Метод Уорда (англ. *Ward's method*).

Для первых трёх методов существует общая формула, предложенная А. Н. Колмогоровым для мер сходства:

$$K_{\eta}([i,j],k) = \left[\frac{n_i K(i,k)^{\eta} + (n_i K(i,k)^{\eta}}{n_i + n_j}\right]^{\frac{1}{\eta}}, \quad 1 \le \eta \le +1,$$

где $[i,j]$ – группа из двух объектов (кластеров) $i$и $j$,

$k$ – объект (кластер), с которым ищется сходство указанной группы;

$n_i$ – число элементов в кластере $i$ ;

$n_j$ – число элементов в кластере $j$.

Для расстояний имеется аналогичная формула Ланса-Вильямса.

Центроидный метод используется для пересчёта матрицы расстояний. В качестве расстояния между двумя кластерами в этом методе берётся расстояние между их центрами тяжести.

В методе Уорда в качестве расстояния между кластерами берётся прирост суммы квадратов расстояний объектов до центров кластеров, получаемый в результате их объединения. В отличие от других методов кластерного анализа, для оценки расстояний между кластерами здесь используются методы дисперсионного анализа. На каждом шаге алгоритма объединяются такие два кластера, которые приводят к минимальному увеличению целевой функции, т.е. внутригрупповой суммы квадратов. Этот метод направлен на объединение близко расположенных кластеров.

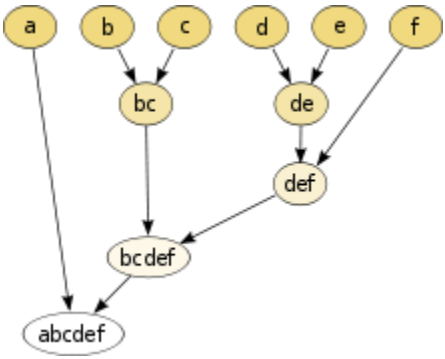



Рис. 28. Дендрограмма

Для вычисления расстояний между кластерами чаще всего пользуются двумя расстояниями: одиночной связью или полной связью (см. обзор мер расстояний между кластерами).

К недостатку иерархических алгоритмов можно отнести систему полных разбиений, которая может являться излишней в контексте решаемой задачи.

### *5.8.2 Алгоритмы квадратичной ошибки*

Задачу кластеризации можно рассматривать как построение оптимального разбиения объектов на группы. При этом оптимальность может быть определена как требование минимизации среднеквадратической ошибки разбиения:

$$e^2(X, L) = \sum_{j=1}^{K} \sum_{i=1}^{n_j} \| \left| x_i^{(j)} - C_j \right| |^2,$$

где $C_j$ – «центр масс» кластера $j$ (точка со средними значениями характеристик для данного кластера).

Алгоритмы квадратичной ошибки относятся к типу плоских алгоритмов. Самым распространенным алгоритмом этой категории является метод k-средних. Этот алгоритм строит заданное число кластеров, расположенных как можно дальше друг от друга. Работа алгоритма делится на несколько этапов:

- случайно выбрать $k$ точек, являющихся начальными «центрами масс» кластеров;
- отнести каждый объект к кластеру с ближайшим «центром масс»;
- пересчитать «центры масс» кластеров согласно их текущему составу.

Если критерий остановки алгоритма не удовлетворен, вернуться к п. 2. В качестве критерия остановки работы алгоритма обычно выбирают минимальное изменение среднеквадратической ошибки. Так же возможно останавливать работу алгоритма, если на шаге 2 не было объектов, переместившихся из кластера в кластер. К недостаткам данного алгоритма можно отнести необходимость задавать количество кластеров для разбиения.

### *5.8.3 Нечеткие алгоритмы*

Наиболее популярным алгоритмом нечеткой кластеризации является алгоритм c-средних (c-means). Он представляет собой модификацию метода k-средних. Шаги работы алгоритма:

- выбрать начальное нечеткое разбиение n объектов на $k$ кластеров путем выбора матрицы принадлежности $U$ размера $n$ x $k$;
- используя матрицу $U$, найти значение критерия нечеткой ошибки:

$$E^2(X,U) = \sum_{i=1}^{N}\sum_{k=1}^{K} U_{ik}\left|\left|x_i^{(k)} - c_k\right|\right|^2,$$

где $c_k$ – «центр масс» нечеткого кластера $k$: $c_k = \sum_{i=1}^{N} U_k x_i$;

- перегруппировать объекты с целью уменьшения этого значения критерия нечеткой ошибки;
- возвращаться в п. 2 до тех пор, пока изменения матрицы $U$ не станут незначительными.

Этот алгоритм может не подойти, если заранее неизвестно число кластеров, либо необходимо однозначно отнести каждый объект к одному кластеру.

### *5.8.4 Алгоритмы, основанные на теории графов*

Суть таких алгоритмов заключается в том, что выборка объектов представляется в виде графа $G$=($V$, $E$), вершинам которого соответствуют объекты, а ребра имеют вес, равный «расстоянию» между объектами. Достоинством графовых алгоритмов кластеризации являются наглядность, относительная простота реализации и возможность внесения различных усовершенствований, основанных на геометрических соображениях. Основными алгоритмами являются алгоритм выделения связных компонент, алгоритм построения минимального покрывающего дерева и алгоритм послойной кластеризации.

### *5.8.5 Алгоритм выделения связных компонент*

В алгоритме выделения связных компонент задается входной параметр $R$ и в графе удаляются все ребра, для которых «расстояния»

больше *R*. Соединенными остаются только наиболее близкие пары объектов. Смысл алгоритма заключается в том, чтобы подобрать такое значение *R*, лежащее в диапазон всех «расстояний», при котором граф «развалится» на несколько связных компонент. Полученные компоненты и есть кластеры.

Для подбора параметра *R* обычно строится гистограмма распределений попарных расстояний. В задачах с хорошо выраженной кластерной структурой данных на гистограмме будет два пика – один соответствует внутрикластерным расстояниям, второй – межкластерным расстояния. Параметр *R* подбирается из зоны минимума между этими пиками. При этом управлять количеством кластеров при помощи порога расстояния довольно затруднительно.

### *5.8.6 Алгоритм минимального покрывающего дерева*

Алгоритм минимального покрывающего дерева сначала строит на графе минимальное покрывающее дерево, а затем последовательно удаляет ребра с наибольшим весом. На рисунке изображено минимальное покрывающее дерево, полученное для девяти объектов.

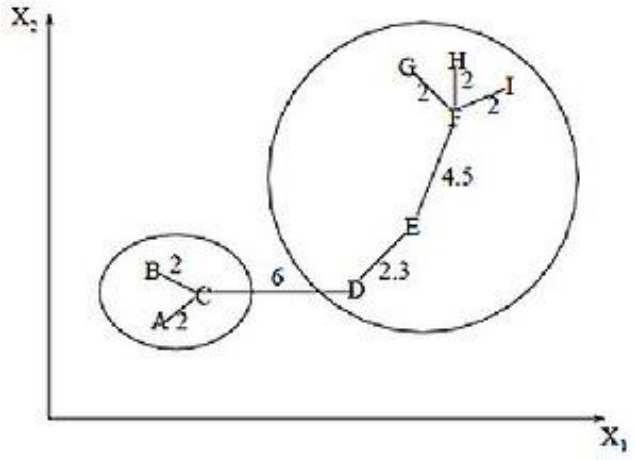



Рис. 29. Минимальное покрывающее дерево, полученное для девяти объектов

Путём удаления связи, помеченной CD, с длиной равной 6 единицам (ребро с максимальным расстоянием), получаем два кластера: {A, B, C} и {D, E, F, G, H, I}. Второй кластер в дальнейшем может быть разделён ещё на два кластера путём удаления ребра EF, которое имеет длину, равную 4,5 единицам.

### *5.8.7 Послойная кластеризация*

Алгоритм послойной кластеризации основан на выделении связных компонент графа на некотором уровне расстояний между объектами (вершинами). Уровень расстояния задается порогом расстояния *c*. Например, если расстояние между объектами $0 \leq \rho(x, x') \leq 1$, то $0 \leq c \leq 1$.

Алгоритм послойной кластеризации формирует последовательность подграфов графа G, которые отражают иерархические связи между кластерами:

$$G^0 \subseteq G^1 \subseteq \ldots \subseteq G^m,$$

где $G^t = (V, E^t)$ – граф на уровне $c^t$;

$E^t = \{e_{ij} \in E: \rho_{ij} \leq c_t\}$;

$c^t$ – t-ый порог расстояния;

m – количество уровней иерархии;

$G^0 = (V, o)$, где o – пустое множество ребер графа, получаемое при $t^0 = 1$;

$G^m = G$, то есть граф объектов без ограничений на расстояние (длину ребер графа), поскольку $t^m=1$.

Посредством изменения порогов расстояния $\{c^0, \ldots, c^m\}$, где $0 = c^0 < c^1 < \ldots < c^m = 1$, возможно контролировать глубину иерархии получаемых кластеров. Таким образом, алгоритм послойной кластеризации способен создавать как плоское разбиение данных, так и иерархическое.

## 5.9 Сравнение алгоритмов

Таблица 5. **Вычислительная сложность алгоритмов**

| Алгоритм кластеризации | Вычислительная сложность |
|---|---|
| Иерархический | $O(n^2)$ |
| k-средних | O(nkl), где k – число кластеров, l – число итераций |
| c-средних | |
| Выделение связных компонент | зависит от алгоритма |
| Минимальное покрывающее дерево | $O(n^2 \log n)$ |
| Послойная кластеризация | O(max(n, m)), где $m < n(n-1)/2$ |

Таблица 6. **Сравнительная таблица алгоритмов**

| Алгоритм кластеризации | Форма кластеров | Входные данные | Результаты |
|---|---|---|---|
| Иерархический | Произвольная | Число кластеров или порог расстояния для усечения иерархии | Бинарное дерево кластеров |
| k-средних | Гиперсфера | Число кластеров | Центры кластеров |
| c-средних | Гиперсфера | Число кластеров, степень нечеткости | Центры кластеров, матрица принадлежности |
| Выделение связных компонент | Произвольная | Порог расстояния R | Древовидная структура кластеров |
| Минимальное покрывающее дерево | Произвольная | Число кластеров или порог расстояния для удаления ребер | Древовидная структура кластеров |
| Послойная кластеризация | Произвольная | Последовательность порогов расстояния | Древовидная структура кластеров с разными уровнями иерархии |

## 5.10 Применение

В биологии кластеризация имеет множество приложений в самых разных областях. Например, в биоинформатике с помощью неё анализируются сложные сети взаимодействующих генов, состоящие порой из сотен или даже тысяч элементов [33]. Кластерный анализ позволяет выделить подсети, узкие места, концентраторы и другие скрытые свойства изучаемой системы, что позволяет в конечном счете узнать вклад каждого гена в формирование изучаемого феномена.

В области экологии широко применяется для выделения пространственно однородных групп организмов, сообществ и т.п. Реже методы кластерного анализа применяются для исследования сообществ во времени. Гетерогенность структуры сообществ приводит к возникновению нетривиальных методов кластерного анализа (например, метод Чекановского).

В общем, стоит отметить, что исторически сложилось так, что в качестве мер близости в биологии чаще используются меры сходства, а не меры различия (расстояния).

При анализе результатов социологических исследований рекомендуется осуществлять анализ методами иерархического агломеративного семейства, а именно методом Уорда, при котором внутри кластеров оптимизируется минимальная дисперсия, в итоге создаются кластеры приблизительно равных размеров. Метод Уорда наиболее удачен для анализа социологических данных. В качестве меры различия лучше квадратичное евклидово расстояние, которое способствует увеличению контрастности кластеров. Главным итогом иерархического кластерного анализа является дендрограмма или «сосульчатая диаграмма». При её интерпретации исследователи сталкиваются с проблемой того же рода, что и толкование результатов факторного анализа – отсутствием однозначных критериев выделения кластеров. В качестве главных рекомендуется использовать два способа – визуальный анализ дендрограммы и сравнение результатов кластеризации, выполненной различными методами.

Визуальный анализ дендрограммы предполагает «обрезание» дерева на оптимальном уровне сходства элементов выборки. «Виноградную ветвь» (терминология Олдендерфера М.С. и

Блэшфилда Р.К.) целесообразно «обрезать» на отметке 5 шкалы Rescaled Distance Cluster Combine, таким образом будет достигнут 80 % уровень сходства. Если выделение кластеров по этой метке затруднено (на ней происходит слияние нескольких мелких кластеров в один крупный), то можно выбрать другую метку. Такая методика предлагается Олдендерфером и Блэшфилдом.

Теперь возникает вопрос устойчивости принятого кластерного решения. По сути, проверка устойчивости кластеризации сводится к проверке её достоверности. Здесь существует эмпирическое правило – устойчивая типология сохраняется при изменении методов кластеризации. Результаты иерархического кластерного анализа можно проверить итеративным кластерным анализом по методу k-средних. Если сравниваемые классификации групп респондентов имеют долю совпадений более 70 % (более 2/3 совпадений), то кластерное решение принимается.

Проверить адекватность решения, не прибегая к помощи другого вида анализа, нельзя. По крайней мере, в теоретическом плане эта проблема не решена. В классической работе Олдендерфера и Блэшфилда «Кластерный анализ» подробно рассматриваются и в итоге отвергаются дополнительные пять методов проверки устойчивости:

- кофенетическая корреляция – не рекомендуется и ограниченна в использовании;
- тесты значимости (дисперсионный анализ) – всегда дают значимый результат;
- методика повторных (случайных) выборок, что, тем не менее, не доказывает обоснованность решения;
- тесты значимости для внешних признаков пригодны только для повторных измерений;
- методы Монте-Карло очень сложны и доступны только опытным математикам.

Применение в информатике.

1. Кластеризация результатов поиска – используется для «интеллектуальной» группировки результатов при поиске файлов, вебсайтов, других объектов, предоставляя пользователю возможность быстрой навигации, выбора заведомо более релевантного подмножества и исключения заведомо менее релевантного – что может

повысить юзабилити интерфейса по сравнению с выводом в виде простого сортированного по релевантности списка:

- Clusty – кластеризующая поисковая машина компании Vivísimo;
- Nigma – российская поисковая система с автоматической кластеризацией результатов;
- Quintura – визуальная кластеризация в виде облака ключевых слов.

2. Сегментация изображений (англ. image segmentation) – кластеризация может быть использована для разбиения цифрового изображения на отдельные области с целью обнаружения границ (англ. edge detection) или распознавания объектов.

3. Интеллектуальный анализ данных (англ. data mining) – кластеризация в Data Mining приобретает ценность тогда, когда она выступает одним из этапов анализа данных, построения законченного аналитического решения. Аналитику часто легче выделить группы схожих объектов, изучить их особенности и построить для каждой группы отдельную модель, чем создавать одну общую модель для всех данных. Таким приемом постоянно пользуются в маркетинге, выделяя группы клиентов, покупателей, товаров и разрабатывая для каждой из них отдельную стратегию.

В работе нужно из иерархических структур (деревьев) выделять отдельные области. Т.е. по сути, необходимо было разрезать исходное дерево на несколько более мелких деревьев. Поскольку ориентированное дерево – это частный случай графа, то естественным образом подходят алгоритмы, основанные на теории графов.

В отличие от полносвязного графа, в ориентированном дереве не все вершины соединены ребрами, при этом общее количество ребер равно $n$–1, где $n$ – число вершин. То есть применительно к узлам дерева, работа алгоритма выделения связных компонент упростится, поскольку удаление любого количества ребер «развалит» дерево на связные компоненты (отдельные деревья). Алгоритм минимального покрывающего дерева в данном случае будет совпадать с алгоритмом выделения связных компонент – путем удаления самых длинных ребер исходное дерево разбивается на несколько деревьев. При этом очевидно, что фаза построения самого минимального покрывающего дерева пропускается.

В случае использования других алгоритмов в них пришлось бы отдельно учитывать наличие связей между объектами, что усложняет алгоритм.

Для достижения наилучшего результата необходимо экспериментировать с выбором мер расстояний, а иногда даже менять алгоритм. Никакого единого решения не существует.

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК


1. Березинская, М.Д. Информационная безопасность современного общества / М.Д. Березинская, А.Ю. Азаров // Информационное общество: состояние, проблемы, перспективы: сб. – 2017. – С. 45-52
2. Машкина, И.В. Управление защитой информации в сегменте корпоративной информационной системы на основе интеллектуальных технологий : дис. д-р. техн. наук: защищена 7.07.09 / Машкина Ирина Владимировна. – Уфа, 2009. – 35 с. – 003474034.
3. Шапот, М. Интеллектуальный анализ данных в системах поддержки принятия решений/ М. Шапот // Открытые системы. – 1998. – № 1. – С. 30-35.
4. Рассел, С. Искусственный интеллект: современный подход / С. Рассел, П. Норвиг. – 2-е изд.: пер. с англ. – Москва : Вильямс, 2016. – 1408 с.
5. Забежайло М.И. Data Mining and Knowledge Discovery in Data Bases: Предметная область, задачи, методы и инструменты / М.И. Забежайло // 6 Нац. конф. с междунар. участием: сб. науч. трудов. Том 2 / Пущино. – 1998. – С. 592–600.
6. Маслова, Н.А. О применении интеллектуального анализа данных для защиты информации корпоративных систем / Н.А. Маслова // Донецк. нац. тех. ун-т. – 2009. – С. 68.
7. Панкратова, Е.С. Автоматическое порождение гипотез в интеллектуальных системах / Е.С. Панкратов, В.К. Финн. – Москва : ЛИБРОКОМ, 2009. – С. 528.
8. Объедков, С.А. Алгоритмы и методы теории решеток и их применение в машинном обучении : дис, канд. тех. наук: 05.13.17 / Объедков Сергей Александрович. – М.:, 2003. – С. 157. – 61 03-5/3517-Х.
9. Барсегян, А.А. Анализ данных и процессов : учебное пособие / А.А. Барсегян, М.С. Куприянов, И.И. Холод, М.Д. Тесс,

С.И. Елизаров. – 3-е изд., перераб. и доп. – Санкт-Петербург : БХВ-Петербург, 2009. — 512 с.
10. Мюллер, А. Введение в машинное обучение с помощью Python : рук. для спец. / А. Мюллер, С. Гвидо. – Москва : Вильямс, 2016-2017. – С.13-37.
11. Аверкин, А.Н., Толковый словарь по искусственному интеллекту / А.Н. Аверкин, М.Г. Гаазе-Рапопорт, Д.А. Поспелов. – Москва : Радио и связь, 1992, – 256 с.
12. Розенблатт, Ф. Принципы нейродинамики: перцептроны и теория механизмов мозга / Ф Розенблатт. – Москва: Мир, 1965. – 478 с.
13. Круглов, В.В. Искусственные нейронные сети. Теория и практика / В.В. Круглов, В.В. Борисов. – Москва : Горячая линия – Телеком, 2001. – 328 с.
14. Осовский С. Нейронные сети для обработки информации / С. Осовский ; пер. с польского И.Д. Рудинского. – Москва : Финансы и статистика, 2004. – 344 с.
15. Фальк, В.Н. Трансдуктивное обучение логистической регрессии в задаче классификации текстов / В.Н. Фальк, И.А. Бочаров, А.Г. Шаграев // Программные продукты и системы. – Москва, 2014. – С. 114–118.
16. Кинелев, В.Г. Многовариантная технология профессиональной ориентации и адаптации обучения / В.Г. Кинелев, Н.М. Кулагин, В.П. Авдеев, Т.М. Киселева, У.П. Фетинина. – Москва, 1998. – С. 44-53.
17. Плотникова, В.С. Проблема правильного формирования инвестиционного портфеля / В.С. Плотникова // Моск. гос. ун-т. – 2006 – С. 48-50.
18. Емельянов, В.В. Теория и практика эволюционного моделирования / В.В. Емельянов, В.В. Курейчик, В.М. Курейчик. – Москва : Физматлит, 2003. – 432 с.
19. Ветров, Д.П. Машинное обучение – состояние и перспективы / Д.П. Ветров // Труды XV Всероссийской научной конференции RCDL’2013 / Ярос. гос. ун-т. – Ярославль, 2013. – С. 21–27.
20. Серафимов, Л.А. Классификация бинарных систем / Л.А. Серафимов, А.В. Фролкова, В.В. Илларионов // Теоретические основы химической технологии. – 2011. – Т.45, № 3 – С.354-358.

21. Бешелев, С.Д. Математико-статистические методы экспертных оценок / С.Д. Бешелев, Ф.Г. Гурвич. – Москва : Статистика, 1974. – 159 с.
22. Прохоров С.А. Прикладной анализ случайных процессов / С.А. Прохоров. – Самара : СНЦ РАН, 2007. – 582 с.
23. Ершов, К.С. Анализ и классификация алгоритмов кластеризации / К.С. Ершов, Т.Н. Романова // Научная электронная библиотека КиберЛеника. – 2016. – С. 274-279.
24. Гринченков, Д.В. Сравнительный анализ алгоритмов интеллектуального анализа данных / Д.В. Гринченков, Ф.Х. Нгуен, Т.Т. Нгуен, Д.А. Горбушин // Моделирование. Теория, методы и средства : материалы 16-й Междунар. науч.-практ. конф. – 2016. – С. 263-266.
25. Гусев, А.Л., Модифицированный метод отношения шансов / А.Л. Гусев, А.А. Окунев // Современная наука: актуальные проблемы теории и практики. Серия: естественные науки и технические науки. – 2018. – № 2. – С. 29-31.
26. Черенков, А.А. Логистическая регрессия – один из инструментов директ-маркетинга / А.А. Черенков // Маркетинг и маркетинговые исследования. – 2003. – № 1. – С. 55-60.
27. Яхъяева, Г.Э. Основы теории нейронных сетей / Г.Э. Яхъяева // Нац. откр. ун-т Интуит. – 2006 – С. 42-50.
28. Осовский, С. Нейронные сети для обработки информации / С. Осовский ; пер. с польского под ред. И.Д. Рудинского. – М.: Финансы и статистика, 2017. – С. 323-412.
29. Рутковская, Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский; пер. с польск. И Д. Рудинского. – 2-е изд., стереотип. – Москва : Горячая линия – Телеком, 2013. – С. 22-90.
30. Филипчик, Е.Ф. «Big data, кластерный анализ и оптимизация в системном анализе» / Е.Ф. Филипчик, Д.Т. Перскевич, О.В. Герман // Наука, техника и образование. – 2017. – № 1(31) – С. 32-35.
31. Кузнецов, Д.Ю. Кластерный анализ и его применение / Д.Ю. Кузнецов, Т.Л. Трошина // Ярославский педагогический вестник. – 2006. – № 4(49) – С.103-107.
32. Махрусе, Н. Современные тенденции методов интеллектуального анализа данных: метод кластеризации / Н. Махрусе // Московский экономический журнал. – 2019. – № 6. – С. 35.

33. Филяк, П.Ю. Сети, большие данные (big data), интеллектуальный анализ данных (data mining) и обеспечение безопасности / П.Ю. Филяк, В.В. Растворов, В.И. Старченко // Вестник МФЮФ. – 2017. – №2 – С. 522-527.

Учебное издание

***Сапрыкин Олег Николаевич***

**ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ**

*Учебное пособие*

Редактор Л.Р. Дмитриенко
Компьютерная верстка Л.Р. Дмитриенко

Подписано в печать 16.12.2020. Формат 60x84 1/16.
Бумага офсетная. Печ. л. 5,0.
Тираж 120 экз. (1-й з-д 1-25). Заказ . Арт. – 4(РЗУ)/2020.

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»
(САМАРСКИЙ УНИВЕРСИТЕТ)
443086, Самара, Московское шоссе, 34.

---

Издательство Самарского университета.
443086, Самара, Московское шоссе, 34.