

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

На правах рукописи



**МОИСЕЕВА Татьяна Александровна**

**МЕТОДЫ ГЕНЕРАЦИИ БАЗ ЗНАНИЙ  
НЕЧЕТКИХ ПРОДУКЦИОННЫХ СИСТЕМ  
С ИСПОЛЬЗОВАНИЕМ ПРОЦЕДУР КЛАСТЕРИЗАЦИИ**

Специальность 1.2.1. Искусственный интеллект и машинное обучение

Диссертация  
на соискание учёной степени кандидата технических наук

Научный руководитель: Леденева Т.М.  
доктор технических наук, профессор

Воронеж – 2025

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	4
ГЛАВА 1. ОБЗОР ПОДХОДОВ К ФОРМИРОВАНИЮ БАЗ ЗНАНИЙ НЕЧЕТКИХ ПРОДУКЦИОННЫХ СИСТЕМ .....	10
1.1.    Нечеткие производственные системы: основные понятия, структура, проблемы разработки .....	10
1.2.    Анализ подходов к формированию баз знаний нечетких производственных систем на основе обучающих данных .....	27
1.3.    Цели и задачи исследования.....	45
Выводы по первой главе .....	49
ГЛАВА 2. ФОРМИРОВАНИЕ БАЗЫ ЗНАНИЙ НА ОСНОВЕ ЭЛЛИПСОИДАЛЬНОЙ КЛАСТЕРИЗАЦИИ .....	50
2.1.    Метод формирования нечетких производственных правил на основе эллипсоидальной кластеризации .....	50
2.2.    Сравнение методов формирования баз знаний на основе эллипсоидальной кластеризации .....	70
Выводы по второй главе .....	77
ГЛАВА 3. НЕЧЕТКИЕ МЕТРИКИ, ОСНОВАННЫЕ НА АДДИТИВНЫХ ГЕНЕРАТОРАХ АРХИМЕДОВЫХ СТРОГИХ ТРЕУГОЛЬНЫХ НОРМ ИЗ КЛАССА РАЦИОНАЛЬНЫХ ФУНКЦИЙ .....	80
3.1.    Понятие и свойства нечеткой метрики .....	80
3.2.    Построение нечетких метрик на основе треугольных норм .....	89
3.3.    Исследование влияния нечетких метрик на решение задачи нечеткой кластеризации .....	96
Выводы по третьей главе .....	107
ГЛАВА 4. ПРОГРАММНЫЙ КОМПЛЕКС «FuzzyLogicCore» И ЕГО АПРОБАЦИЯ ДЛЯ РЕШЕНИЯ ПРАКТИЧЕСКИХ ЗАДАЧ.....	110
4.1.    Описание программного комплекса «FuzzyLogicCore» .....	110
4.2.    Применение программного комплекса «FuzzyLogicCore» для анализа электроэнцефалограмм .....	129

4.3. Применение программного комплекса «FuzzyLogicCore» для решения задачи вставки пропущенных значений.....	143
Выводы по четвертой главе .....	147
ЗАКЛЮЧЕНИЕ .....	148
СПИСОК ЛИТЕРАТУРЫ .....	150
ПРИЛОЖЕНИЕ А. АКТЫ О ВНЕДРЕНИИ.....	171
ПРИЛОЖЕНИЕ Б. СВИДЕТЕЛЬСТВА О ГОСУДАРСТВЕННОЙ РЕГИСТРАЦИИ ПРОГРАММ ДЛЯ ЭВМ .....	174

## ВВЕДЕНИЕ

**Актуальность темы исследования.** Одним из важнейших направлений искусственного интеллекта является создание систем, основанных на знаниях. К ним, в частности, относятся нечеткие производственные системы (НПС), являющиеся обобщением обычных производственных систем. Приложения НПС ориентированы на решение задач управления (нечеткие системы управления, нечеткие регуляторы встроены в огромное количество промышленных изделий), прогнозирования, диагностики, принятия решений в условиях неопределенности. Ядром НПС, «отвечающим» за сферу применения, является база знаний, включающая базу *если-то-правил*, основанную на производственной модели представления знаний. Формирование базы знаний – важнейший этап проектирования, при этом различают два основных подхода к интеграции знаний и данных в НПС. В первом случае для формирования базы знаний привлекаются эксперты, что порождает субъективность и неоднозначность в формулировках правил, невозможность обеспечить полноту знаний о реальной системе. С другой стороны, в настоящее время наблюдается рост объемов информации, описывающей прецеденты в той или иной предметной области, а также полученной в результате наблюдения или экспериментов. Алгоритмы автоматической генерации баз знаний на основе обучающих данных включают несколько групп методов, среди которых кластерный подход признан перспективным для некоторых типов НПС (*evolving fuzzy systems*). Выделяя группировки данных, он позволяет повысить уровень интерпретируемости базы знаний – свойства, которое лежит в основе объяснительной способности интеллектуальных систем. Для кластеризации обучающих данных целесообразно использовать метрические алгоритмы, к преимуществам которых относятся геометрическая интерпретация, возможность выделить нетипичные объекты, возможность осуществить квантизацию пространства признаков, сравнительная простота реализации. Повышение качества обработки данных в процессе кластеризации позволит, в свою очередь, обеспечить

качество базы знаний НПС на начальной стадии ее формирования и определить дальнейшую процедуру работы с ней (оптимизация, редукция, интерполяция правил и др.). Актуальность исследования обусловлена необходимостью совершенствования методов и алгоритмов обработки данных, обеспечивающих генерацию баз знаний НПС с улучшенными свойствами на основе процедур кластеризации при наличии обучающей выборки – важной научной задачи, имеющей значение для инженерии знаний и разработки систем искусственного интеллекта.

**Степень разработанности темы исследования.** Исследованиям в области формирования и оптимизации баз знаний НПС посвящены работы J. Dickerson, B. Kosko, J.C. Bezdek, P. P. Angelov, A. Lemos, H. Genter, E. D. Lughofer, а также отечественных ученых А.С. Катасёва, Д.В. Катасёвой, И.А. Ходашинского, А.Л. Тулупьева, А.А. Сорокина, М.А. Сергиенко и др. С целью совершенствования алгоритмов нечеткой кластеризации I. Kramosil и J. Michálek ввели понятие нечеткой метрики, которое получило развитие в работах A. George, P. Veeramani, V. Gregori, O. Grigorenko, N. M. Ralević. В связи с появлением обобщенных представлений треугольных и конорм в классе рациональных функций появилась возможность построения конкретных представлений нечетких метрик, которые позволяют улучшить качество кластеризации – важного этапа построения базы знаний НПС.

**Цель исследования** заключается в совершенствовании методов генерации баз знаний нечетких производственных систем на основе обучающих данных с использованием кластерных процедур.

Для достижения цели необходимо решить следующие **задачи**:

1. Проанализировать подходы к генерации баз знаний НПС и выявить возможности метрических алгоритмов кластеризации для решения данной проблемы.
2. Разработать процедуры для автоматического формирования баз знаний на основе обучающих данных с использованием эллипсоидальной кластеризации.

3. Предложить новые варианты нечетких метрик и исследовать их свойства.

4. Разработать и протестировать программное обеспечение для проведения вычислительного эксперимента и апробации предложенных подходов.

**Объект исследования** – база знаний в форме совокупности производственных правил. **Предмет исследования** – процедура генерации баз знаний НПС с использованием алгоритмов кластеризации.

**Методы исследования** базируются на принципах инженерии знаний. Теоретические результаты получены с использование методов оптимизации, теории нечетких множеств и методов нечеткого моделирования, методов кластерного анализа. Для программной реализации использовались современные технологии объектно-ориентированного программирования.

**Научная новизна.** В диссертации представлены следующие результаты, характеризующиеся научной новизной:

- метод формирования баз нечетких производственных правил на основе эллипсоидальной кластеризации, отличающийся использованием эллипсоидов минимального объема и позволяющий повысить качество аппроксимации в сравнении с известным подходом, основанным на использовании матриц ковариаций кластеров;
- ограничения на параметры непрерывных архimedовых треугольных норм из класса рациональных функций, обеспечивающие свойство строгости;
- семейство нечетких метрик, впервые полученных на основе аддитивных генераторов непрерывных архimedовых строгих треугольных норм из класса рациональных функций, отличающихся набором настраиваемых параметров, что позволяет учитывать структуру данных при использовании метрических алгоритмов кластеризации;
- модель комплексной оценки качества кластеризации, основанная на использовании функций порядкового взвешенного агрегирования и позволяющая учитывать «нечеткое большинство» значимых значений критериев качества кластеризации;

- алгоритм формирования базы знаний нечеткого классификатора энцефалограмм для асинхронного интерфейса «мозг-компьютер», отличающийся учетом наиболее значимых предикторов, и позволяющий распознавать реальные и мысленные движения верхних конечностей;
- структура и программная реализация приложения для формирования баз правил нечетких продукционных систем на основе метрических алгоритмов кластеризации с возможностью выбора нечетких метрик и критериев качества кластеризации, что обеспечит учет особенностей информационной среды конкретной прикладной задачи.

**Соответствие Паспорту специальности.** Полученные в диссертации научные результаты соответствуют следующим пунктам Паспорта специальности 1.2.1 «Искусственный интеллект и машинное обучение: п. 5 «Методы и технологии поиска, приобретения и использования знаний и закономерностей, в том числе – эмпирических, в системах искусственного интеллекта ...», п. 15 «Математические исследования в области статистики, логики, алгебры, топологии, анализа функций ...».

**Теоретическая и практическая значимость.** Метод формирования баз знаний НПС развивает технологии выявления закономерностей и знаний в системах искусственного интеллекта. Нечеткие метрики, построенные с использованием генераторов непрерывных строгих архimedовых треугольных норм из класса рациональных функций, не только расширяют возможности построения нечетких метрических пространств, но и значительно дополняют инструментарий метрических алгоритмов кластеризации, демонстрируя в рамках исследования превосходство по многим показателям качества кластеризации перед традиционными метриками.

Практическая значимость работы заключается в программной реализации предложенного подхода к решению важной для НПС проблемы автоматического формирования базы знаний на основе обучающего множества, который в сравнении с часто используемым экспертым методом повышает качество и обоснованность

решений в интеллектуальных информационных системах, в которых НПС является ядром.

Теоретические результаты диссертационной работы используются в учебном процессе Воронежского государственного университета; программный комплекс применяется в финансовой компании «ООО ФПК «Альфа», а также для проведения исследований, связанных с разработкой интерфейсов «мозг-компьютер», в Лаборатории медицинской кибернетики Воронежского государственного университета.

**Результаты и положения, выносимые на защиту:**

1. Предложенный метод формирования *если-то-*правил на основе аппроксимации кластеров эллипсоидами минимального объема создает основу для генерации базы знаний с учетом обучающих данных, что повышает уровень объяснительной способности интеллектуальных информационных систем, базирующихся на НПС.

2. На основе аддитивных генераторов строгих непрерывных архимедовых треугольных норм, представимых рациональными функциями, получены нечеткие метрики, которые, с одной стороны, имеют теоретическое значение для построения нечетких метрических пространств, а с другой – обладают практической ценностью для метрических алгоритмов нечеткой кластеризации, проявляя адаптивные свойства за счет настройки параметров.

3. Разработанный многофункциональный программный комплекс, структура которого включает библиотеки метрик и критериев качества кластеризации, программную реализацию некоторых известных алгоритмов кластеризации и всех предложенных алгоритмов и процедур, позволил осуществить объемный вычислительный эксперимент, а также продемонстрировал возможность использования предложенного подхода для построения нечеткого классификатора в системе интерфейса «мозг-компьютер» с целью анализа электроэнцефалограмм.

**Степень достоверности и апробация результатов.** Достоверность результатов исследования основана на корректном использовании математического

аппарата, обосновании выбора алгоритмических решений и их согласованностью с результатами вычислительного эксперимента. Результаты диссертации докладывались и обсуждались на следующих научных конференциях: международная научная конференция «Актуальные проблемы прикладной математики, информатики и механики» (Воронеж, 2021-2023 гг.), International Conferences on Control Systems, международная научная конференция Mathematical Modeling, Automation and Energy Efficiency (Lipetsk, 2022-2024 гг.), межвузовская научная конференция молодых ученых и студентов «Математика, информационные технологии, приложения» (Воронеж, 2024 г.).

**Публикации.** По результатам исследования опубликовано 15 научных работ (4 без соавторов), в том числе 5 статей в журналах из Перечня ВАК, 4 статьи – в изданиях, индексируемых в Scopus, 2 свидетельства о государственной регистрации программ для ЭВМ.

**Объем работы.** Диссертация состоит из введения, четырех глав, заключения, списка использованных источников, включающего 190 наименований, и двух приложений. Объем диссертации составляет 175 страниц, диссертация содержит 55 рисунков, 22 таблицы.

# ГЛАВА 1. ОБЗОР ПОДХОДОВ К ФОРМИРОВАНИЮ БАЗ ЗНАНИЙ НЕЧЕТКИХ ПРОДУКЦИОННЫХ СИСТЕМ

В настоящем разделе рассматриваются принципы построения нечетких моделей и их применение в задачах кластеризации и аппроксимации. Приводится обзор методов генерации баз знаний нечетких продукционных систем на основе экспериментальных данных. На основе проведенного обзора формулируются цели и задачи диссертационной работы.

## 1.1. Нечеткие продукционные системы: основные понятия, структура, проблемы разработки

В современном мире искусственный интеллект играет все более важную роль в различных сферах жизни. Многие страны осознают потенциал искусственного интеллекта и внедряют его в свои национальные стратегии развития. Это связано с тем, что использование искусственного интеллекта способно значительно увеличить эффективность производственных процессов, увеличить ВВП, обеспечить рост мировой экономики, ускорение инноваций во всех областях науки, повышение качества жизни населения, доступности и качества медицинской помощи, качества образования, производительности труда и качества отдыха. Страны, разрабатывающие национальные стратегии его развития, понимают, что искусственный интеллект – ключевой элемент цифровой трансформации и критически важен для будущего экономического и социального развития.

Согласно национальной стратегии развития искусственного интеллекта на период до 2030 года [47], *искусственный интеллект* (ИИ) – комплекс технологических решений, позволяющий имитировать когнитивные функции человека (включая поиск решений без заранее заданного алгоритма) и получать при выполнении конкретных задач результаты, сопоставимые с результатами интеллектуальной деятельности человека или превосходящие их. Разными источниками по-разному оценивается возможный размер международного рынка

ИИ к 2030 году, но их объединяет прогноз роста рынка. Так, например, на сайте *Forbes* [74] оценивается, что к 2030 году размер рынка ИИ достигнет 1339 миллиардов долларов. На сайте [61] приводится диаграмма размера международного рынка ИИ с 2021 по 2030 год по данным Statista (рис. 1.1), причем к 2030 году ожидается рост рынка до 1840 миллиардов долларов.



Рис. 1.1. Размер международного рынка ИИ

Компании активно инвестируют значительные финансовые ресурсы в развитие ИИ, рассматривая его как стратегическое направление, способное повысить эффективность бизнес-процессов, автоматизировать принятие решений и стимулировать инновации. Рост объемов вложений обусловлен высокой конкуренцией на рынке технологий, перспективами внедрения ИИ в различных отраслях, включая здравоохранение, финансовый сектор, промышленность и информационные технологии (рис. 1.2).

Развитие ИИ во многом опирается на прогресс новых информационных технологий. Можно перечислить следующие направления *новых информационных технологий* [26]: инженерия знаний, обработка нечеткой информации и нечеткий логический вывод, мягкие вычисления, нейросетевые технологии, эволюционное

моделирование, многоагентные системы и распределенный искусственный интеллект.



Рис. 1.2. Диаграмма инвестиций компаний в ИИ

*Инженерия знаний* – это совокупность методов, моделей и технических приёмов, предназначенных для создания систем, обеспечивающих решение проблем на основе использования знаний. Фактически под этим термином понимается методология, теория и технологии, охватывающие методы анализа, извлечения, обработки и представления знаний.

*Знания* – вид информации, обладающий свойствами внутренней интерпретируемости, структурированности, связности, наличия семантической метрики и активности. Систематизация, хранение и использование знаний в интеллектуальных информационных системах реализуются посредством баз знаний.

*База знаний* – хранилище единиц знаний, описывающих атрибуты и действия, связанные с объектами предметной области, а также возможные при этом неопределенности.

Можно выделить следующие стратегии работы со знаниями [8]: извлечение знаний; приобретение знаний; формирование знаний. Извлечение знаний – это

процесс целенаправленного взаимодействия между инженером по знаниям и экспертом предметной области, направленный на выявление, анализ и представление явных и неявных экспертных знаний в форме, пригодной для использования в информационных системах. Приобретение знаний характеризуется непосредственным вводом и структурированием знаний экспертом предметной области, что способствует непрерывному развитию и актуализации базы знаний. Формирование знаний представляет собой автоматизированный процесс, осуществляемый на основе методов интеллектуального анализа данных, направленный на выявление закономерностей, моделей и концепций из массивов структурированной или неструктурной информации, для генерации новых знаний, неявно содержащихся в исходных данных.

Модели представления знаний предлагают различные подходы к формализации и структурированию знаний. К основным *моделям представления знаний* относятся: продукционные модели, семантические сети, фреймы, логические модели. В табл. 1.1 приведено сравнение данных моделей.

Каждая из перечисленных моделей имеет свои достоинства и недостатки. Продукционная модель обладает такими достоинствами, как наглядность, высокая модульность, легкость внесения дополнений и изменений и простота механизма логического вывода. В общем виде под продукцией понимается выражение следующего вида:

$$(i); Q; P; A \rightarrow B; N,$$

где  $i$  – имя продукции,  $Q$  – сфера применения продукции,  $P$  – условие применимости ядра продукции, как правило, предикат,  $A \rightarrow B$  – ядро продукции. Элемент  $N$  описывает постусловия продукции. Они актуализируются только в том случае, если ядро продукции реализовалось.

Однако, необходимость учитывать неопределенность входных данных, лингвистическую неопределенность и нелинейные закономерности обуславливают использование аппарата теории нечетких множеств и нечеткой логики [34].

Таблица 1.1. Модели представления знаний

Название	Преимущества	Недостатки
Продукционные модели	Простота интерпретации и понимания; легкость модификации знаний, добавления и удаления правил (модульность); эффективное моделирование знаний эксперта.	Несколько взаимных отношений правил; неоднозначный выбор выполняемой продукции.
Семантические сети	Естественное представление знаний; большие выразительные возможности.	Сложность организации поиска нужного фрагмента сети; результат вывода не гарантирует достоверности результата; громоздкость и неэффективность.
Фреймы	Возможность наследования атрибутов уменьшает дублирование информации; универсальная модель; гибкая возможность реализации вывода с помощью присоединенных процедур.	Отсутствие специального механизма управления выводом; относительно высокая сложность.
Логические модели	Формальность; модульность; единственность теоретического обоснования.	Высокие требования и ограничения к предметной области; сложность выражения эвристических знаний в модели.

Под *нечеткой переменной* подразумевается нечеткое подмножество с именем, которое соответствует его интерпретации. Нечеткая переменная задается тройкой  $\langle \alpha, U, A \rangle$ , где  $\alpha$  – название переменной,  $U$  – универсальное множество,  $A$  – нечеткое подмножество  $U$  с функцией принадлежности  $\mu_A(x)$ , описывающее ограничение на значение переменной.

*Нечеткая величина* – это нечеткая переменная, определенная на множестве действительных чисел  $\mathbb{R}$ . Нечеткое число является типом нечеткой величины.

*Лингвистическая переменная* задается кортежем  $\langle \beta, Term, U, G, M \rangle$ , где  $\beta$  – название переменной;  $Term = \{t_1, t_2, \dots\}$  – терм-множество или множество значений переменной  $\beta$ , причём каждое из них является нечёткой переменной  $t_k$ , заданной на универсальном множестве  $U$  числовой или нечисловой природы;  $G$  – синтаксическое правило, порождающее новые названия значений переменной  $\beta$ ;

$M$  – семантическое правило, которое ставит в соответствие каждой новой нечеткой переменной ее смысл.

*Лингвистической шкалой*  $S$  называется конечное линейно упорядоченное множество термов  $S = \{S_i\}_{i=0,\overline{T}}$ , удовлетворяющих следующим условиям [34]:

1) если  $i < j$ , то  $S_i$  предшествует  $S_j$  ( $S_i \prec S_j$ );

2) отрицание терма определяется правилом  $\text{Neg}(S_i) = S_{T-i+1}$  (для шкалы с четным числом термов) или  $\text{Neg}(S_i) = S_{T-i}$  (для шкалы с нечетным числом термов);

3) объединение (связка «или») термов определяется правилом

$$S_i \vee S_j = \max\{S_i, S_j\} = S_j, \text{ если } S_i \prec S_j;$$

4) пересечение (связка «и») термов определяется правилом

$$S_i \wedge S_j = \min\{S_i, S_j\} = S_i, \text{ если } S_i \prec S_j.$$

Для адаптации к реальным условиям используются нечеткие продукции. Ядро  $A \rightarrow B$  является центральным компонентом нечеткой продукции. Оно может быть представлено в одной из более распространенных форм: *если A, то B*; где  $A$  и  $B$  – некоторые выражения нечеткой логики, которые наиболее часто представляются в форме лингвистических высказываний.

*Лингвистическим высказыванием* называется высказывание одного из следующих типов [34]:

- $\beta$  есть  $\alpha$ , где  $\beta$  – наименование лингвистической переменной,  $a$  – ее значение, которому соответствует некоторый лингвистический терм из базового терм-множества  $T$  лингвистической переменной  $\beta$ ;

- $\beta$  есть  $\Xi(\alpha)$ , где  $\Xi$  – модификатор, соответствующий таким словам, как *очень, более или менее, почти, совсем* и другим, которые задаются процедурой  $G$ ;

- составные высказывания, образованные из высказываний первых двух типов с помощью логических связок *и, или, не, если-то, если-то-иначе, тогда и только тогда*.

Лингвистические высказывания применяются для формирования производственных правил типа *если-то*. В процессе построения базы правил

требуется определить совокупность входных лингвистических переменных, совокупность выходных лингвистических переменных, а также множество производных правил, устанавливающих связи между входными и выходными переменными. В случае, если переменная является лингвистической, необходимо задать для нее лингвистическую шкалу, в рамках которой термы определяются с использованием соответствующих функций принадлежности.

Нечеткие производные правила могут иметь различный тип [32]. Различают лингвистическую модель, модель Такаги-Сугено, реляционные нечеткие модели. Пусть  $L_x = \{A_i\}_{i=1,\dots,n}$  и  $L_y = \{B_i\}_{j=1,\dots,m}$  – лингвистические шкалы входной ( $x$ ) и выходной ( $y$ ) переменных соответственно. Рассмотрим различные типы правил, которые связывают эти переменные.

*Лингвистическая модель* имеет вид:

$$R_i : \text{если } x \text{ есть } A_i, \text{ то } y \text{ есть } B_i,$$

где  $x$ ,  $y$  – входная и выходная лингвистические переменные;  $A_i, B_i$  – термы лингвистических шкал соответствующих переменных.

*Реляционная модель* обобщает лингвистическую модель и имеет вид:

$$R_{ij} : \text{если } x \text{ есть } A_i, \text{ то } y \text{ есть } B_j \text{ со степенью } r_{ij}.$$

Таким образом, каждому значению (терму) входной лингвистической переменной необходимо сопоставить значение (терм) выходной лингвистической переменной со степенью  $r_{ij}$ . Тем самым, на множестве  $L_x \times L_y$  формируется нечеткое отношение с элементами  $r_{ij}$ .

*Модель Такаги-Сугено* (TS-модель) можно рассматривать как комбинацию лингвистической и регрессионной моделей. Она задается в следующем виде:

$$R_i : \text{если } x \text{ есть } A_i, \text{ то } y = f_i(x).$$

Здесь входная переменная  $x$  является лингвистической, а выходная переменная  $y$  является обычной функцией. Предполагается, что при формировании базы правил функции  $f_i$  должны иметь одинаковую структуру. Зачастую

используются линейные функции  $f_i$ , которые можно получить на основе регрессионного анализа.

Если в заключении правила функция принимает значение константы, то данную модель можно интерпретировать как *нечеткий классификатор*.

Совокупность нечетких продукционных правил образует базу правил, которая совместно с базой данных, хранящей информацию о параметрах модели и параметрах правил, составляет базу знаний – ядро нечетких продукционных систем. База знаний на качественном уровне с помощью лингвистических переменных описывает зависимость выходной переменной от входных переменных.

Для работы с нечеткими продукционными системами разработано множество программных сред, библиотек и фреймворков. Можно привести такие примеры, как: MATLAB (Fuzzy Logic Toolbox), Python (Scikit-fuzzy), CLIPS (FuzzyCLIPS), C++ (FuzzyLite), Java (JFuzzyLogic), Wolfram Mathematica, R (sets и FuzzyR), среда GUAJE, LabVIEW (Fuzzy Logic Toolkit). Практически все перечисленное ПО предоставляет средства для моделирования нечетких систем лингвистической модели и модели Такаги-Сугено (в том случае, если поддержка модели Такаги-Сугено отсутствует, это указано в недостатках), но его функциональность различается. Сравнение ПО приведено в табл. 1.2.

Таблица 1.2. Сравнение ПО для нечеткого моделирования

<b>Название</b>	<b>Вид</b>	<b>Преимущества</b>	<b>Недостатки</b>
MATLAB (Fuzzy Logic Toolbox)	Пакет расширения Matlab	Интеграция с MATLAB и Simulink; графический интерфейс; документация; расширяемость; генерация кода; поддержка генерации баз знаний.	Требование лицензии Matlab.
Python (Scikit-fuzzy)	Библиотека Python	Бесплатность и открытый исходный код; интеграция с Python; гибкость.	Отсутствие графического интерфейса; отсутствие готовых решений генерации баз знаний; отсутствие поддержки модели Такаги-Сугено.

Продолжение таблицы 1.2

CLIPS (FuzzyCLIPS)	Программная среда и ее расширение с собственным языком	Бесплатность для обучения и исследовательских целей; гибкость.	Устаревшее ПО; отсутствие графического интерфейса; отсутствие поддержки генерации баз знаний; отсутствие поддержки модели Такаги-Сугено.
C++ (FuzzyLite)	Библиотека C++	Бесплатность и открытый исходный код; интеграция с C++; гибкость; производительность; совместимость с Matlab; графический интерфейс.	Отсутствие готовых решений генерации баз знаний.
Java (JFuzzyLogic)	Библиотека Java	Бесплатность и открытый исходный код; реализует спецификацию FCL; интеграция с Java; гибкость.	Отсутствие графического интерфейса; отсутствие готовых решений генерации баз знаний.
Wolfram Mathematica	Программа	Интеграция с другими средствами Mathematica, предоставляющая широкие возможности; документация; гибкость.	Требование лицензии; отсутствие специализированного GUI для нечетких систем; отсутствие готовых решений генерации баз знаний.
R (sets и FuzzyR)	Пакеты для R	Бесплатность и открытый исходный код; интеграция с R; графический интерфейс; гибкость; частично поддерживает генерацию нечетких систем.	Ограниченнaя документация и интеграция.
Среда GUAJE	Программа	Бесплатность; графический интерфейс; поддержка генерации баз знаний.	Ограниченнaя гибкость.
LabVIEW (Fuzzy Logic Toolkit)	Дополнительный модуль для LabVIEW	Графический интерфейс; интеграция с LabVIEW; гибкость; импорт из других систем.	Требование лицензии LabView; отсутствие готовых решений генерации баз знаний.

В общем случае *нечеткими системами* (НС) называются такие системы, для описания которых используются аппарат теории нечетких множеств и нечеткая логика. *Нечеткая продукционная система* (НПС) является частным случаем НС. НС получили широкое распространение при создании систем управления

техническими объектами, например, поездом [62], квадрокоптером [72], электроприводом постоянного тока [69]; систем прогнозирования и управления технологическими процессами, например, температурой пара в паронагревательной установке [56], микроклиматом в помещении [70]. Помимо задач управления с помощью НС решаются задачи диагностики [46, 54], в том числе медицинской [28], прогнозирования [13, 46], классификации, например, в системах анализа трафика [16] и при распознавании аварийных режимов [12].

На рис. 1.3 приведена типовая структура нечеткой производственной системы.

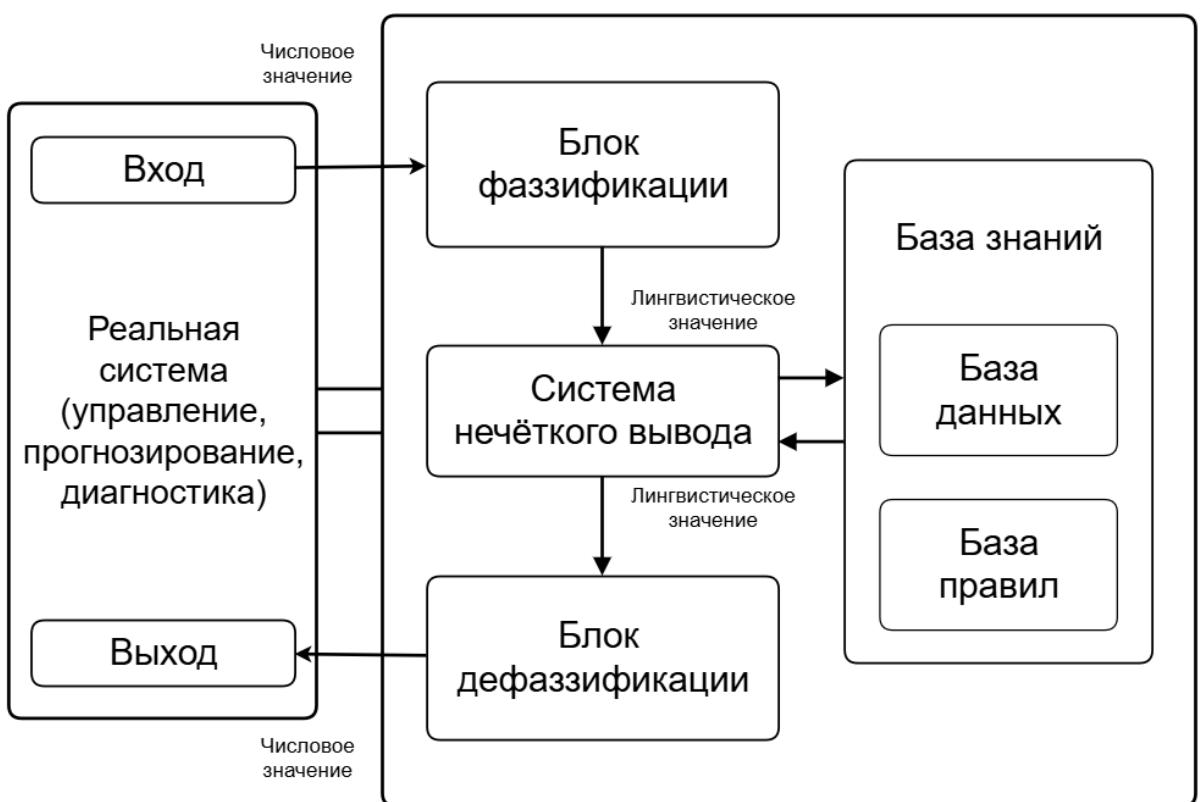


Рис. 1.3. Типовая структура нечеткой производственной системы

Основными компонентами НПС являются:

- блок фазификации;
- механизм нечеткого логического вывода;
- блок дефазификации;
- база знаний.

Ядром базы знаний является база *если-то-правил*, которая приближенно описывает поведение функции, связывающей значения входных переменных и соответствующее значение выходной переменной. Для формирования приближенного описания предполагается, что входные и/или выходная переменные являются лингвистическими и принимают значения в заранее сформированной лингвистической шкале. База данных содержит значения параметров лингвистического представления переменных. При разработке нечетких систем в качестве термов лингвистических шкал, как правило, выбирают треугольные или трапециевидные нечеткие числа, а также гауссовые нечеткие числа.

Блок фазификации позволяет от числового значения переменной перейти к его приближенному представлению в форме нечеткого числа.

Блок дефазификации, получая на вход нечеткое множество в качестве значения входной переменной, позволяет перейти к его числовому представлению с помощью подходящих методов дефазификации.

Механизм нечеткого логического вывода в НПС функционирует на основе обобщенного modus ponens и позволяет определить значение выходной переменной в форме нечеткого множества. Дефазификация этого множества позволяет найти числовое значение выходной переменной.

Основой механизма нечеткого логического вывода является схема правильных рассуждений modus ponens (рис. 1.4).

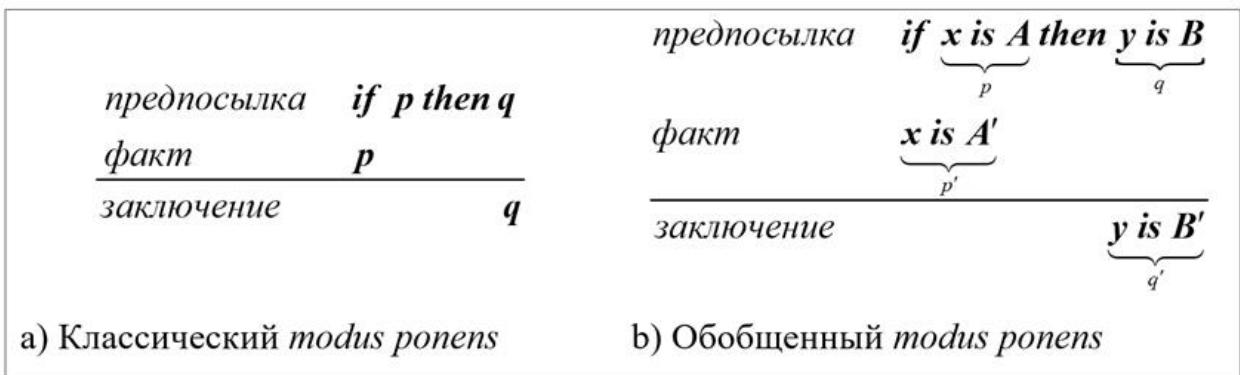


Рис. 1.4. Схемы правильных рассуждений

Значение  $B'$  выходной переменной  $y$ , соответствующее значению  $A'$  входной переменной  $x$ , при наличии правила ***если***  $x$  есть  $A$ , ***то***  $y$  есть  $B$ , определяется по формуле

$$B' = A' \circ (A \rightarrow B)$$

(здесь символом  $\circ$  обозначена операция композиции,  $\rightarrow$  – импликация), которая может быть записана в виде

$$B'(y) = \sup_x T(A'(x), I(A(x), B(y))),$$

где  $\circ$  соответствует  $(\max - T)$ -композиции;  $T$  – треугольная норма;  $I$  – импликация, представляющая продукционное правило  $A \rightarrow B$ ;  $A(x), A'(x), B(y), B'(y)$  – функции принадлежности соответствующих нечетких множеств. В случае, если  $X$  представляет собой множество с конечным числом элементов, выражение принимает вид

$$B'(y) = \max_x T(A'(x), I(A(x), B(y))).$$

Актуальными проблемами разработки НПС являются следующие задачи [52]:

- 1) формализация нечетких логических связок;
- 2) построение лингвистических шкал входных и/или выходных переменных;
- 3) выбор типа продукционных правил и формирование базы знаний;
- 4) оптимизация базы знаний.

Рассмотрим данные проблемы подробнее.

**Формализация нечетких логических связок.** В [64, 140, 182] отмечается, что наибольшее влияние на качество аппроксимации, помимо качества базы правил, оказывает представление нечетких логических операций в системе логического вывода и выбор метода дефазификации. В статье [35] рассматривается схема нечеткого логического вывода, приводятся результаты экспериментального исследования и рекомендации по выбору функционального представления компонент.

Широкое распространение получили следующие операции композиции (табл. 1.3).

Таблица 1.3. Примеры операции композиции

Название	Формула
$(\text{sup}-\text{min})$ -композиция	$\mu_{B^i}(y) = \sup_{x \in X} \left\{ \min \left[ \mu_{A^i}(x), \mu_{A \rightarrow B}(x, y) \right] \right\}$
$(\text{sup}-\text{prod})$ -композиция	$\mu_{B^i}(y) = \sup_{x \in X} \left\{ \mu_{A^i}(x) \cdot \mu_{A \rightarrow B}(x, y) \right\}$
$(\text{max}-\text{min})$ -композиция	$\mu_{B^i}(y) = \max_{x \in X} \left\{ \min \left[ \mu_{A^i}(x), \mu_{A \rightarrow B}(x, y) \right] \right\}$
$(\text{max}-\text{prod})$ -композиция	$\mu_{B^i}(y) = \max_{x \in X} \left\{ \mu_{A^i}(x) \cdot \mu_{A \rightarrow B}(x, y) \right\}$

Поскольку операции композиции и импликации могут быть определены неоднозначно и должны быть определенным образом специфицированы, то выбор конкретных представлений определяет некоторый алгоритм, который реализует нечеткий логический вывод. Вариант реализации механизма логического вывода представим кортежем  $FIS = \langle I, T, Sh, Agg \rangle$ , где  $I$  – нечеткая импликация;  $T$  – треугольная норма;  $Sh$ ,  $Agg$  – соответственно схема и операция агрегирования.

Операция нечеткой импликации играет ключевую роль в нечетких продукционных моделях, устанавливая причинно-следственное отношение между предпосылками и заключениями правил. В настоящее время разработано множество различных вариантов данной операции. В табл. 1.4 приводятся некоторые из них [5].

Таблица 1.4. Примеры операций нечеткой импликации

Название	Формула
Классическая нечеткая импликация (Клине–Даэнса)	$\mu_{A \rightarrow B}(x, y) = \max \{1 - \mu_A(x), \mu_B(y)\}, \mu_A(x) \geq \mu_B(y)$
Нечеткая импликация Заде	$\mu_{A \rightarrow B}(x, y) = \max \{\min \{\mu_A(x), \mu_B(y)\}, 1 - \mu_A(x)\}$
Нечеткая импликация Мамдани	$\mu_{A \rightarrow B}(x, y) = \min \{\mu_A(x), \mu_B(y)\}$
Нечеткая импликация Ларсена	$\mu_{A \rightarrow B}(x, y) = \mu_A(x) \cdot \mu_B(y)$

Совокупность правил  $R_i = A_i \rightarrow B_i$  образует базу правил  $R = \{R_i\}_{i=1..N}$ , поэтому при формировании выходного нечеткого множества  $B'$  важно определиться со

схемами и операциями операции агрегирования. В [140] определены следующие основные схемы агрегирования:

1.  $(\forall i = 1..N (B'_i = A \circ R_i)) \Rightarrow B' = \text{Agg}(B'_1, \dots, B'_N) \Rightarrow y' = \text{defuz}(B').$
2.  $(\forall i = 1..N (B'_i = A \circ R_i)) \Rightarrow y'_i = \text{defuz}(B'_i) \Rightarrow y' = \text{Agg}(y'_1, \dots, y'_N).$
3.  $R = \text{Agg}(R_1, \dots, R_N) \Rightarrow B' = A' \circ R \Rightarrow \text{defuz}(B').$

Функция агрегирования ставит в соответствие векторной оценке скалярную величину, при этом различают конъюнктивную, дизъюнктивную и компромиссную стратегии агрегирования. Отдельным классом функций агрегирования являются OWA-операторы (Ordered Weighted Averaging Operator) [31]. Их ключевая особенность заключается в том, что перед агрегированием компоненты векторной оценки упорядочиваются, а весовые коэффициенты назначаются не отдельным критериям, а их позициям в отсортированном векторе. Это позволяет моделировать различные стратегии агрегирования – от чисто конъюнктивных до дизъюнктивных, а также промежуточные компромиссные варианты.

**Построение лингвистических шкал входных и/или выходных переменных** в нечетких системах включает выбор и оптимизацию множества термов лингвистической шкалы и их соответствующих функций принадлежности. Настройка влияет на точность и интерпретируемость нечеткой системы. *Точность* относится к способности нечеткой модели формировать такое значение выходной переменной при заданных значениях входов, которое в определенном смысле близко к реакции реальной системы и способно адекватно представлять моделируемую систему. *Интерпретируемость* относится к способности нечеткой модели отражать работу системы в понятной для человека форме. При настройке лингвистических шкал важно обеспечить следующие требования [29]:

- к функциям принадлежности: нормальность, выпуклость, непрерывность, ограниченность носителя;
- к структуре лингвистической шкалы: упорядоченность термов, их разумное количество (считается, что работа более чем с 9 термами вызывает у эксперта затруднения, их оптимальное количество -  $7 \pm 2$ ), различимость, полноту

шкалы, равномерность распределения термов, наличие граничных нечётких множеств, естественное положение нуля.

Оптимизация параметров функций принадлежности может проводиться с использованием различных методов, например генетическими алгоритмами [179] или нейронными сетями в случае гибридных нейро-нечетких моделей [123, 166].

**Задача формирования баз знаний** включает в себя задачу формирования лингвистических шкал, подразумевает определение количества правил и структурную организацию термов лингвистических шкал в систему продукционных правил. Решение данных задач в зависимости от выбранных подходов к генерации базы знаний может осуществляться как последовательно, так и параллельно, например, при генерации базы знаний на основе процедуры кластеризации.

При этом к базе правил предъявляются различные требования, основные из которых это: полнота, непротиворечивость, связность и отсутствие избыточности [52] (рис. 1.5).

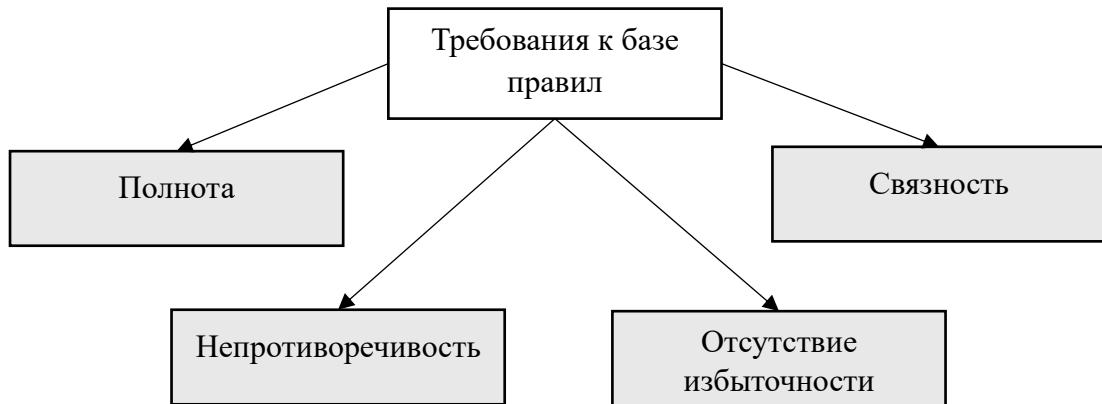


Рис. 1.5. Требования к базе правил

База правил называется *полной*, если каждое значение  $x^*$  входной переменной  $X$  можно связать с некоторым значением  $y^*$  выходной переменной  $Y$ , при этом в зависимости от шкал переменных различают лингвистическую и численную полноту базы правил. Численно полной называется база правил, для

которой каждое четкое входное значение приводит к активизации хотя бы одного правила, и, таким образом, в соответствии с заключением правила формируется числовое значение выходной переменной. Причиной неполноты может быть некорректность разбиения области значений входной переменной, поэтому на решение данной проблемы нужно обращать особое внимание. На рис. 1.6-а изображена полная база правил, на рис. 1.6-б – неполная база правил [81].

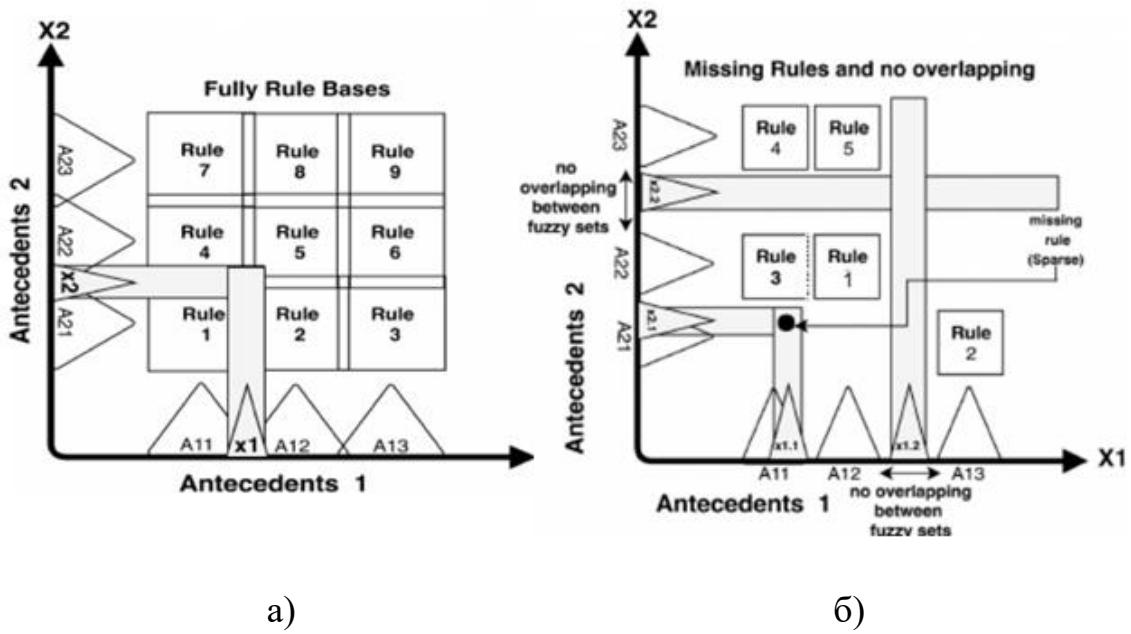


Рис. 1.6. Полная и неполная базы правил

База правил называется *непротиворечивой* (согласованной), если она не содержит несовместные правила, т. е. правила, имеющие одинаковые посылки, но разные заключения. Рассмотрим два правила  $R_i : \text{если } x = A_i, \text{то } y = B_i$  и  $R_j : \text{если } x = A_j, \text{то } y = B_j$ , где  $A_i, A_j$  – термы;  $B_i, B_j$  – термы или числа, причем  $B_i \neq B_j$ . Можно выделить несколько типов противоречивых правил в зависимости от соотношения посылок этих правил при несовпадающих заключениях:  $A_i = A_j$  (тип 1);  $A_i \subset A_j$  (тип 2);  $A_i \cap A_j \neq \emptyset$  (тип 3). Заметим, что здесь вместо  $=$  и  $\subset$  целесообразно рассматривать нечеткие аналоги этих отношений. На рис. 1.7 представлен пример противоречивых правил типа 1.

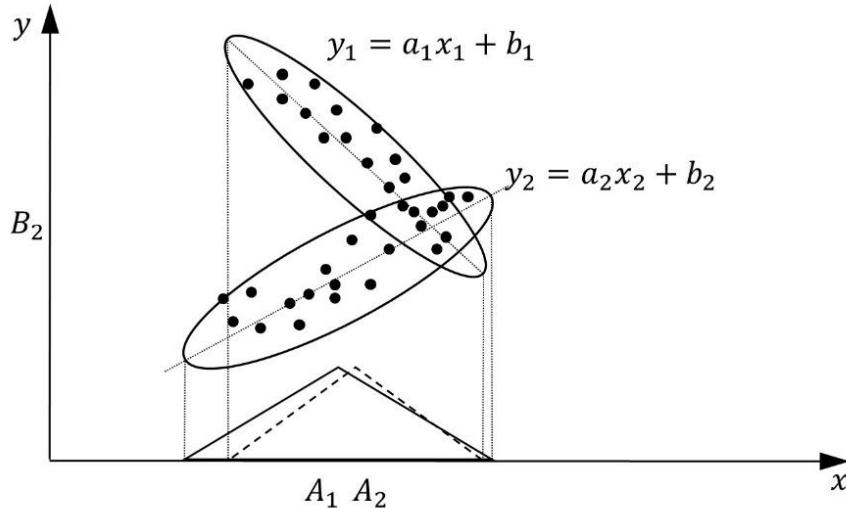


Рис. 1.7. Пример противоречивых правил

Несовместность может быть вызвана, во-первых, ошибкой, допущенной в ходе формирования правил, особенно в случае большого их числа. Другой причиной может стать неоднозначность моделируемой системы, т. е. ситуация, когда измерения входных и выходных данных системы не являются однозначными, и одному входному состоянию  $x^*$  могут соответствовать различные выходные состояния.

База правил называется *связной*, если она не содержит такие правила  $R_i$  и  $R_j$ , что пересечение содержащихся в их заключениях нечетких множеств (термов)  $B_i$  и  $B_j$  является пустым, то есть  $B_i \cap B_j = \emptyset$ . Иными словами, для любого значения у выходной переменной  $Y$  выполняется следующее условие:  $\mu_{B_i}(y) \cdot \mu_{B_j}(y) \neq 0$ . В [52] отмечается, что связность базы правил повышает гладкость поверхности модели, а несвязность обуславливает появление на ней участков резкого подъема или спуска. Требование связности не является абсолютно необходимым, поскольку могут существовать такие процессы или объекты, в которых параметры меняются скачкообразно.

К избыточности базы правил приводит наличие идентичных правил в базе, которые могут появиться либо вследствие ошибки, либо в результате применения

процедуры генерации дополнительных правил, идентичных имеющимся, с целью усиления их заключений. Типы избыточности правил можно описать теми же ситуациями, что и противоречивые правила, предполагая, что заключения этих правил совпадают, то есть  $B_i = B_j = B$ .

**Задача оптимизации базы знаний.** Оптимизация базы знаний нечеткой производственной системы направлена на улучшение ее качества, что включает в себя такие характеристики, как точность и интерпретируемость [29]. Также к критериям качества можно отнести вычислительную эффективность. Основные цели этой оптимизации включают:

- уменьшение избыточности: сокращение количества правил путем удаления дублирующихся или малоинформационных, что снижает объем базы знаний и повышает скорость обработки;
- обеспечение полноты и непротиворечивости: гарантия того, что база знаний охватывает все возможные ситуации без логических противоречий, что повышает надежность системы;
- повышение интерпретируемости: упрощение структуры базы знаний для облегчения понимания и анализа;
- улучшение точности вывода: оптимизация параметров функций принадлежности и структуры правил для повышения точности результатов системы.

## 1.2. Анализ подходов к формированию баз знаний нечетких производственных систем на основе обучающих данных

Подходы к формированию базы знаний можно разделить на три группы: на основе информации от экспертов (например, [60]), на основе наблюдаемых данных (рассматриваются ниже) и гибридные [78], сочетающие два данных подхода.

При использовании подхода к формированию базы знаний на основе информации от экспертов требуется команда как минимум из четырех человек: пользователя (заказчика), эксперта, программиста и инженера по знаниям. В данном случае применяются стратегии извлечения и приобретения знаний. Методы

извлечения знаний подразделяются на две большие группы: коммуникативные и текстологические [8]. Коммуникативные методы извлечения знаний охватывают методы и процедуры контактов инженера по знаниям с экспертом, а текстологические включают методы извлечения знаний из документов и специальной литературы. Среди коммуникативных методов можно перечислить: пассивные, такие как наблюдение, чтение лекций, протокол «мыслей вслух», и активные, например «мозговой штурм», «круглый стол», ролевые игры, анкетирование, интервью, диалог, экспертные игры.

В том случае, если для создания базы знаний привлекается коллектив экспертов, возникает риск формирования противоречивых правил. В данном случае должны использоваться методы устранения противоречивых правил [52, 60]. Отметим, что данная проблема также актуальна и для систем, сгенерированных на основе данных.

Одним из преимуществ разработки нечетких производственных систем с привлечением экспертов является возможность учесть различные аспекты и нюансы предметной области, которые могут быть упущены при автоматизированной разработке. Эксперты способны выделить ключевые факторы, определить взаимосвязи и зависимости между ними, что важно для точного моделирования системы. Благодаря опыту и знаниям экспертов можно создать более эффективную и точную модель, что позволяет повысить качество и надежность системы. Однако, есть и недостатки данного подхода. Привлечение экспертов увеличивает время и стоимость разработки нечеткой производственной системы, так как требуется проводить обширные консультации, обсуждения и анализ. Кроме того, возможно нежелание экспертов раскрывать всю информацию или субъективное влияние их мнения на качество модели. Также в некоторых областях не всегда возможно привлечение экспертов.

В качестве примера правил, полученных с привлечением эксперта, можно рассмотреть правила, полученные в [15]. Здесь была построена экспертная система для диагностики пневмонии с целью предоставления систематических и надежных средств оценки пневмонии путем интеграции различных клинических параметров

в единую диагностическую структуру. В табл. 1.5 приведена часть полученных правил.

Таблица 1.5. Нечеткие правила для диагностики пневмонии

<b>Номер</b>	<b>Правило</b>
1	IF Age IS Young AND Cough IS Asymptomatic AND Shortness of Breath IS Asymptomatic AND Temperature IS Normal AND Chest pain IS Asymptomatic AND Fatigue IS Asymptomatic AND Saturation IS Normal AND Leukocytes IS Normal AND Platelets IS Normal THEN Risk IS Low
2	IF Age IS Middle AND Cough IS Asymptomatic AND Shortness of breath IS Asymptomatic AND Temperature IS Normal AND Chest pain IS Asymptomatic AND Fatigue IS Asymptomatic AND Saturation IS Normal AND Leukocytes IS Normal AND Platelets IS Normal THEN Risk IS Low
3	IF Age IS Elderly AND Cough IS Asymptomatic AND Shortness of breath IS Asymptomatic AND Temperature IS Normal AND Chest pain is asymptomatic AND Fatigue IS Asymptomatic AND Saturation IS Normal AND Leukocytes IS Normal and Platelets IS Normal THEN Risk IS Moderate
...	
54	IF Saturation IS Critical THEN Risk IS Very high
55	IF Temperature IS Elevated THEN Risk IS Moderate
56	IF Chest pain IS Mild THEN Risk IS Moderate
57	IF Fatigue IS Mild THEN Risk IS Moderate

В связи с увеличением объема и доступности данных, описывающих процессы и явления различной природы, актуальной задачей является идентификация НПС на основе наблюдаемых данных. Можно выделить несколько направлений генерации базы знаний на основе обучающих данных:

- алгоритмы сеточного разбиения;
- обучение на основе метаэвристических (в частности, генетических) методов;
- разбиение на основе кластеризации;
- извлечение правил из других моделей, например деревьев решений;

- генерация правил нейросетевыми методами.

В таблице 1.6 представлены ссылки на представителей каждого подхода. На рис. 1.8 приведена классификация подходов к генерации НПС.

Таблица 1.6. Методы генерации баз знаний

<b>Название подхода</b>		<b>Ссылки</b>
Сеточное разбиение		[76, 183]
Метаэвристические алгоритмы	Эволюционные (генетические алгоритмы)	[120, 172, 118, 101, 83, 75, 88, 77]
	Роевые алгоритмы	[114, 142, 126, 75, 127, 21]
	Другие метаэвристические алгоритмы	[2, 49, 66, 67, 181]
На основе кластеризации		[22, 93, 104, 141, 145]
Извлечение правил из других моделей	Ассоциативные правила	[48, 124, 143]
	Из деревьев решений	[84, 116]
Нейросетевые методы		[100, 125, 160, 167, 168, 174, 175, 177; 123, 166]

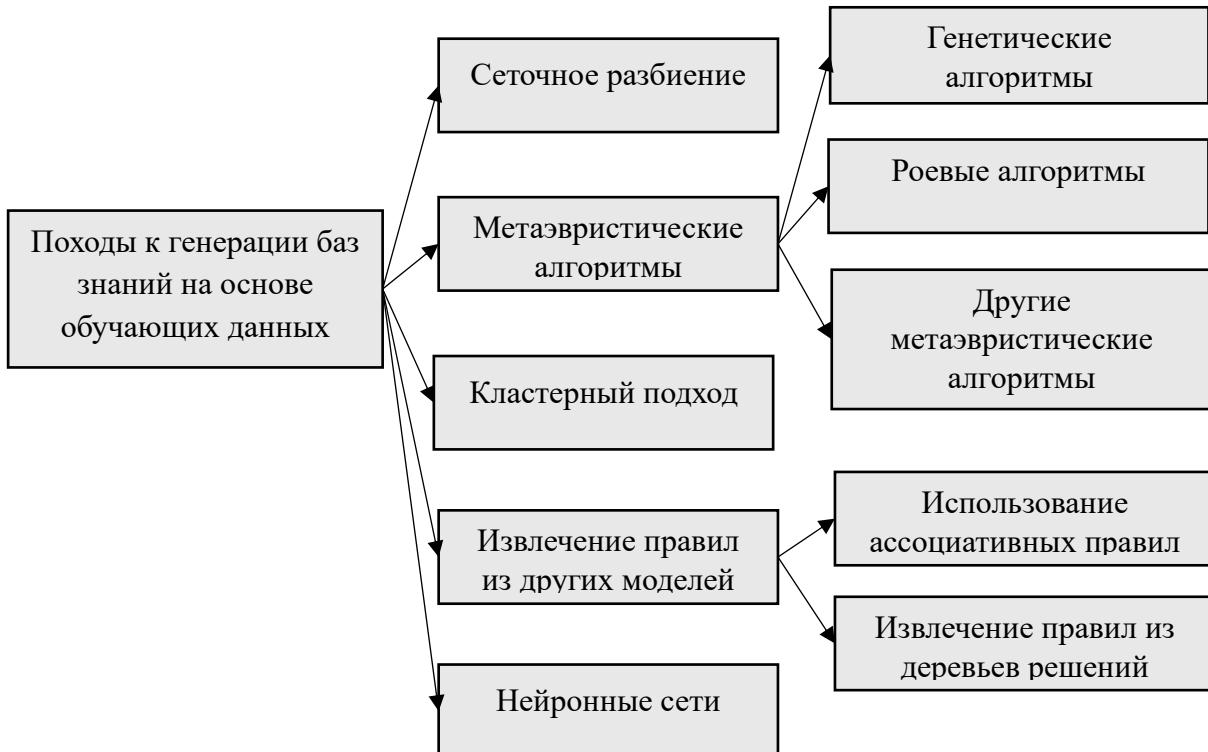


Рис. 1.8. Классификация алгоритмов генерации базы знаний на основе обучающих данных

Рассмотрим часто используемые алгоритмы перечисленных подходов.

В качестве представителей первого направления можно привести такие алгоритмы, как Абе-Лэна и Ванга-Менделя. Рассмотрим алгоритм Абе-Лэна [76]. Пусть неизвестная функция характеризуется одномерным выходом  $y$  и  $n$ -мерным входным вектором  $x$ . Область, на которой определена переменная  $y$ , разделяется на  $m$  интервалов:  $[y_0, y_1], (y_1, y_2], \dots, (y_{m-1}, y_m]$ . Назовем  $i$ -й интервал  $(y_{i-1}, y_i]$  выходным интервалом  $i$ .

Используя заданные входные данные, для которых выходы находятся в выходном интервале  $i$ , рекурсивно определяется область входных значений, соответствующая выходному интервалу  $i$ . А именно, прежде всего находятся области активизации  $A_{ii}$ , которые определяют входную область, соответствующую выходному интервалу  $i$ , посредством вычисления минимального и максимального значений входных данных для каждого выходного интервала. Если область активизации для выходного интервала  $i$  перекрывается с областью активизации для выходного интервала  $j$ , то область перекрытия определяется как область запрещения  $I_{ij}$ . Нечеткое правило в этом случае имеет вид:

*Если  $x$  есть  $A_{ii}$  и  $x$  не  $I_{ij}$ , то  $x$  соответствует выходному интервалу  $i$ .*

Если входные данные для выходных интервалов  $i$  и/или  $j$  находятся внутри области запрещения, то определяются одна или две дополнительные области активизации. В дальнейшем, если области активизации определены и также перекрываются, то далее определяется дополнительная область запрещения. Данный процесс повторяется до тех пор, пока проблема наложения областей не будет решена. Нечеткие правила определяются либо на основе областей активизации, либо на основе областей активизации и соответствующих областей запрещения. Недостатком данного алгоритма является то, что он не решает проблему противоречивых правил.

Алгоритм Ванга-Менделя [183] состоит из нескольких шагов. На первом шаге производится разбиение пространства входных и выходной переменной на регионы: области определения каждой переменной разбиваются на равные

интервалы, и каждому интервалу ставится в соответствие функция принадлежности, в оригинальном варианте – треугольные. На втором шаге происходит генерация нечетких правил, по одному правилу на каждую точку входных данных. Каждая точка относится к тому интервалу, на котором значение функции принадлежности в этой точке достигает максимума. Таким образом происходит генерация первоначальной базы правил. Остальные шаги алгоритма направлены на оптимизацию первоначально сгенерированной базы правил.

Также для генерации базы знаний могут использоваться эволюционные алгоритмы, например, генетические алгоритмы (ГА) [119]. Эволюционные алгоритмы могут использоваться для генерации базы правил на основе уже заданных лингвистических шкал (например алгоритм, описанный в [88], осуществляет селекцию нечетких правил) и для настройки параметров функций принадлежности, либо для обеих задач одновременно (так, в статье [77] используется комбинация генетических алгоритмов с бинарным и вещественным кодированием для поиска функций принадлежности и правил соответственно. При этом инициализация популяции базы правил происходит случайным образом).

Генетический алгоритм – это эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искомых параметров с использованием механизмов, аналогичных естественному отбору в природе. Эволюционные алгоритмы, в частности ГА, хорошо подходят для решения различных задач оптимизации. Было теоретически и практически доказано, что они обеспечивают решение задач, требующих эффективный поиск в сложных пространствах. Благодаря этой причине ГА могут рассматриваться как мощное средство оптимизации, не зависящее от предметной области и применимое к различным задачам, составляющим обучение нечетких классификаторов.

Основные принципы работы простого ГА заключены в схеме в табл. 1.7 [51] (см. также рис. 1.9).

Таблица 1.7. Алгоритм работы простого ГА

Шаг	Действие
0	Выбор метода кодирования хромосом и функции приспособленности.
1	Генерация начальной популяции из $n$ хромосом
2	Вычисление приспособленности для каждой хромосомы
3	Выбор пары хромосом-родителей с помощью одного из способов отбора
4	Проведение кроссинговера двух родителей с вероятностью $P_c$ с производством двух потомков
5	Проведение мутации потомков с вероятностью $P_m$
6	Создание новой популяции из $n$ хромосом
7	Повторение шагов 2–6, пока не будет достигнут критерий окончания процесса

Важнейшим вопросом при использовании ГА является кодирование данных (вид хромосомы). В классическом алгоритме ГА используется бинарное кодирование (и именно для бинарного кодирования сформулирована теорема схемы (Schema Theorem) [119], согласно которой схемы (подмножества хромосом, обладающие общими битовыми шаблонами) с приспособленностью выше средней и низкой дисперсией будут увеличиваться в численности в последующих поколениях. Это указывает на то, что ГА будет постепенно концентрироваться вокруг лучших решений, улучшая их с каждым поколением).

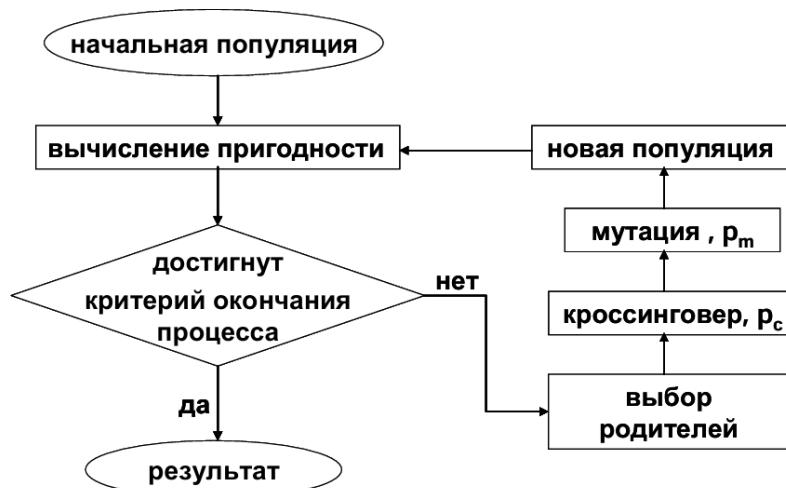


Рис. 1.9. Схема простого ГА

Также среди распространенных можно назвать вещественное кодирование, целочисленное кодирование, символьное кодирование и кодирование в виде деревьев.

Для генерации нечетких правил используются подходы, которые описывают данные с двух различных позиций:

- мичиганский подход [120], предполагающий, что хромосомой является отдельное правило, а целая популяция – базой правил;
- питтсбургский подход [172], в котором каждая хромосома кодирует целую базу правил. Данный подход требует значительных затрат памяти и процессорного времени [122];
- некоторые исследователи выделяют итеративный подход [118]. Так же, как и в Мичиганском подходе, хромосома кодирует одно правило, но, в отличие от Мичиганского подхода, не вся популяция представляет собой базу правил, а происходит отбор лучших правил на каждой итерации.

Кроме того, существуют различные алгоритмы, применяющие генетические алгоритмы на разных этапах генерации нечетких правил. Например, в алгоритме SLAVE (Structural Learning Algorithm in a Vague Environment) [101] генетический алгоритм используется на этапе генерации следующего правила, при этом каждая хромосома кодирует отдельное правило, но, в отличие от Мичиганского подхода, не вся популяция представляет собой базу правил, а результатом работы алгоритма является одно правило, лучше всего представляющее подмножество обучающих данных.

Помимо кодирования информации, важнейшим этапом в применении ГА является выбор функции приспособленности. Функция приспособленности оценивает степень приспособленности данной хромосомы в качестве ответа на поставленную задачу. Одним из удобных вариантов организации функции приспособленности, часто встречающимся на практике, является применение численной степени приспособленности, в определенном смысле показывающей близость некоторого варианта решения к оптимальному. Функция приспособленности строится таким образом, что при оптимальном решении

принимает максимальное или минимальное значение. Стоит заметить, что само значение может оставаться неизвестным, так как разработчику генетического алгоритма необходимо только поведение данной функции. Иными словами, задача поиска оптимального решения на некотором множестве формулируется как задача поиска минимума или максимума функции приспособленности. В Мичиганском подходе фитнесс-функцией определяется эффективность отдельного правила, например, в статье [118] эффективность отдельного правила оценивается как покрытие обучающих данных, а в Питтсбургском подходе фитнесс-функция отражает эффективность закодированной в хромосоме базы правил. При этом не оценивается эффективность отдельного правила, что означает, что эффективные правила могут не попасть в следующее поколение. В статье [83] применяется Питтсбургский подход для генерации нечеткой системы, и в качестве функции приспособленности используется модификация корня из среднеквадратичной ошибки.

К достоинствам ГА можно отнести универсальность, отсутствие требования дифференцируемости целевой функции, поиск глобального максимума и возможность распараллеливания вычислений. К недостаткам – высокую вычислительную сложность, преждевременную сходимость и сложность настройки параметров. Тем не менее, нечеткие системы, полученные с помощью ГА, демонстрируют достаточно высокую точность работы.

Другие метаэвристические алгоритмы, например, метод роя частиц [114] или оптимизация по алгоритму муравьиной колонии [126], алгоритм «стадо криля» [66] по большей части используются для оптимизации и тонкой настройки уже существующей базы правил, однако в статье [75] осуществляется гибридизация генетического алгоритма и алгоритма муравьиной колонии для генерации правил нечеткой системы (на основе уже заданных функций принадлежности). Также в статье [127] алгоритм муравьиной колонии используется для решения задачи идентификации НПС. В статье [21] муравьиный алгоритм используется для нахождения заключений правил.

Другим способом задания разбиения пространства входных и выходных переменных является выполнение проекций подмножеств пространства входов-выходов, определяющих нечеткие правила, на оси координат. Так, в статье [93] рассматриваются правила, имеющие эллипсоидальную форму в пространстве входных и выходных переменных. Далее многомерные подмножества пространства входов и выходов проецируются на одномерные пространства отдельных лингвистических переменных, образуя прототипы термов лингвистической шкалы. Аналогичный подход применяется в статье [22]. Кроме того, подход к генерации правил на основе кластеризации является одним из подходов, применяющихся при создании так называемых развивающихся нечетких систем (evolving fuzzy systems), например, в статьях [104, 141, 145].

Подходы к извлечению правил из нейронных сетей делятся на три группы [1]: декомпозиционные, учитывающие внутреннюю структуру сети; педагогические, которые рассматривают нейросеть как черный ящик; эклектические, сочетающие компоненты декомпозиционного и педагогического подходов. В качестве примера можно привести такие методы декомпозиционного подхода, как экспоненциальный алгоритм извлечения правил КТ [100], полиномиальный алгоритм извлечения правил Цукамото [178] и метод извлечения правил путем индукции дерева решений (CRED) [167]. Рассмотрим алгоритм, появившийся одним из первых – алгоритм КТ. Алгоритм анализирует веса и пороги нейронов, формируя правила для скрытых и выходных слоев, используя входные признаки. Алгоритм выделяет положительные и отрицательные атрибуты, строя подтверждающие и опровергающие правила. Для уменьшения сложности применяются эвристики, которые сокращают пространство поиска, исключая избыточные комбинации. Итоговые правила переписываются, чтобы исключить ссылки на скрытые узлы, и упрощаются. КТ демонстрирует высокую устойчивость к шуму и эффективно обрабатывает сложные задачи классификации и прогнозирования. В качестве примера педагогического подхода можно привести алгоритм извлечения правил, основанный на интервальном анализе [174], получение правил с использованием выборки [125] и алгоритм TREPAN [160], алгоритм извлечения правил двоичного входа и выхода (BIO-RE)

[175] и алгоритм STARE [177]. К эклектическим алгоритмам относятся алгоритмы подхода быстрого поиска, например, FERN [168].

Помимо извлечения правил из нейронных сетей, существуют подходы, осуществляющие генерацию нечетких систем на основе других моделей машинного обучения. Достаточно обширным разделом является осуществление нечеткого логического вывода на основе нечетких ассоциативных правил.

Пусть  $I$  – множество (набор) элементов некоторой природы, транзакция  $\tau$  – это набор элементов из  $I$ :  $\tau \subseteq I$ . Количество элементов в наборе называется размером набора; набор, состоящий из  $k$  элементов, будем называть  $k$ -элементным набором. Ассоциативным правилом называется выражение вида  $X \Rightarrow Y$ , где  $X, Y \subseteq I, X, Y \neq \emptyset, X \cap Y = \emptyset$ . Правило  $X \Rightarrow Y$  можно интерпретировать следующим образом: «каждая транзакция  $\tau$ , которая содержит  $X$ , также содержит  $Y$ » [33].

Определение нечеткого ассоциативного правила базируется на определении нечеткой транзакции. Нечеткая транзакция – это непустые нечеткие множества  $\tau \subseteq I$ . Каждому объекту  $i \in I$  соответствует степень принадлежности  $\tau(i)$  нечеткой транзакции  $\tau$ . При этом  $\tau(I_0)$  - степень принадлежности подмножества элементов  $I_0 \subseteq I$  нечеткой транзакции  $\tau$ , определяемая по формуле:  $\tau(I_0) = \min_{i \in I_0} \tau(i)$ . Нечеткое ассоциативное правило  $X \Rightarrow Y$ , где  $X, Y \subseteq I, X, Y \neq \emptyset, X \cap Y = \emptyset$  существует во множестве транзакций  $T$ , если  $\tau(X) \leq \tau(Y), \forall \tau \in T$ , т. е. степень принадлежности  $X$  больше степени принадлежности  $Y$  для каждой нечеткой транзакции  $\tau$  [158].

В [124] для генерации нечеткого классификатора используются ассоциативные правила, т. е. создание нечеткой системы осуществляется в два этапа: на первом этапе генерируются ассоциативные правила, на втором этапе – с помощью т. н. метода максимальной группировки (*max-group method*) происходит построение нечеткой системы. Преимуществом данного подхода является то, что полученная нечеткая система является полной. В статье [143] описывается генерация нечеткой системы лингвистической модели после предварительной генерации ассоциативных правил с временными ограничениями. В статье [48]

рассматривается построение нечеткой системы, которая на основе нечетких ассоциативных правил осуществляет нечеткий логический вывод по методу Мамдани.

Другим распространенным подходом является генерация НПС на основе деревьев решений. Деревья решений – это непараметрический метод обучения с учителем, используемый для решения задач классификации и регрессии. Цель состоит в том, чтобы создать модель, которая предсказывает значение целевой переменной путем обучения простым правилам принятия решений, выведенным из характеристик данных. Дерево можно рассматривать как кусочно-постоянную аппроксимацию. Пример дерева решений для задачи классификации приведен на рис. 1.10.

В [116] процедура генерации нечетких продукционных правил для нечеткого классификатора достаточна проста: все пути от корня дерева, полученного на первоначальном этапе, до его листьев, соединяются связками «И», а условия в узлах дерева описывают нечеткие трапециевидные функции принадлежности переменных. Меткой класса каждого продукционного правила является метка класса листа, которым окончился путь. Тем не менее, полученные таким образом нечеткие правила, по мнению авторов, недостаточно гибко реагируют на изменения входных переменных, поэтому первоначально полученные правила подвергаются постобработке, а именно модификации функций принадлежности и созданию промежуточных термов.

В [84] описывается генерация нечетких правил для задач классификации и регрессии на основе нечетких деревьев решений, при этом используется тот же принцип, что и в предыдущей статье. После первоначальной генерации набора правил происходит слияние тех правил, в которых заключения одинаковы, для упрощения базы правил.

Необходимо также упомянуть популярные нейро-нечеткие модели. Нечеткой нейронной сетью (ННС) [73] называют четкую нейросеть, которая построена на основе многослойной архитектуры с использованием И-, ИЛИ-нейронов.

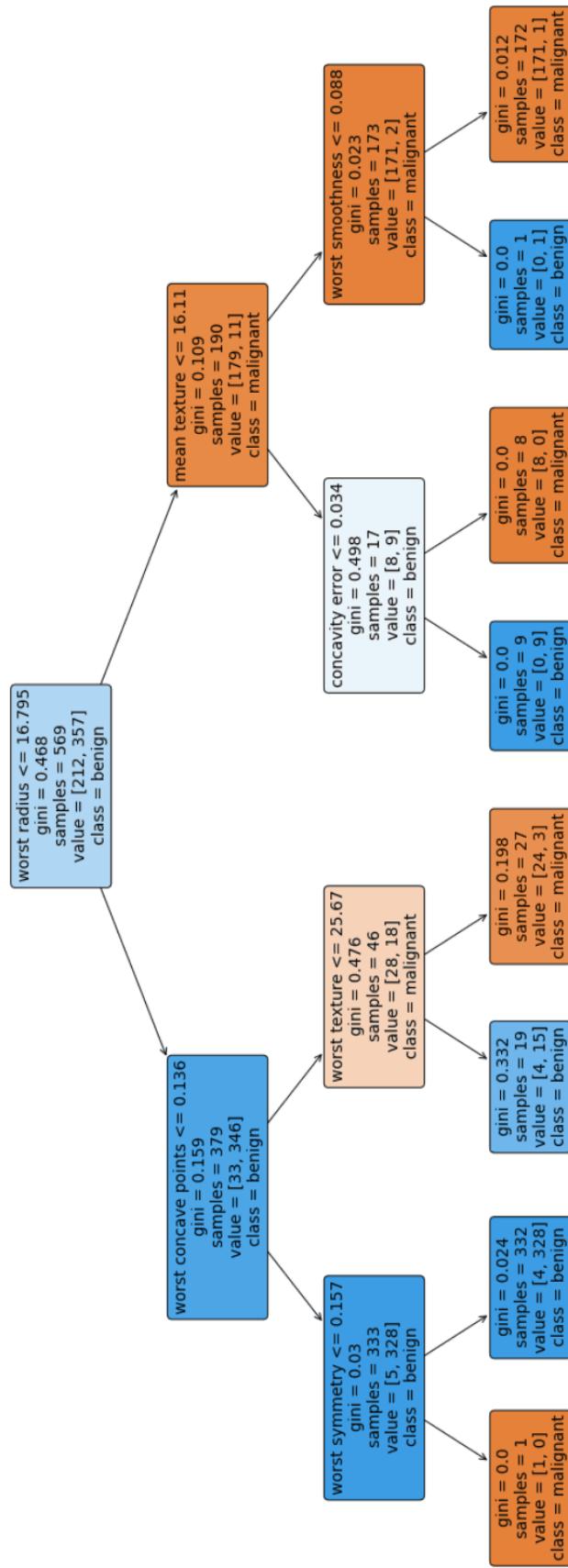


Рис. 1.10. Пример дерева решений

При этом функции принадлежности нечетких чисел сети могут настраиваться с помощью алгоритма обратного распространения ошибки. В качестве примера на рис. 1.11 рассмотрим архитектуру нечеткой нейронной сети класса ANFIS [123], включающую адаптивный слой функций принадлежности и логические И-, ИЛИ-нейроны, «логические» нейроны, моделирующие логические связки.

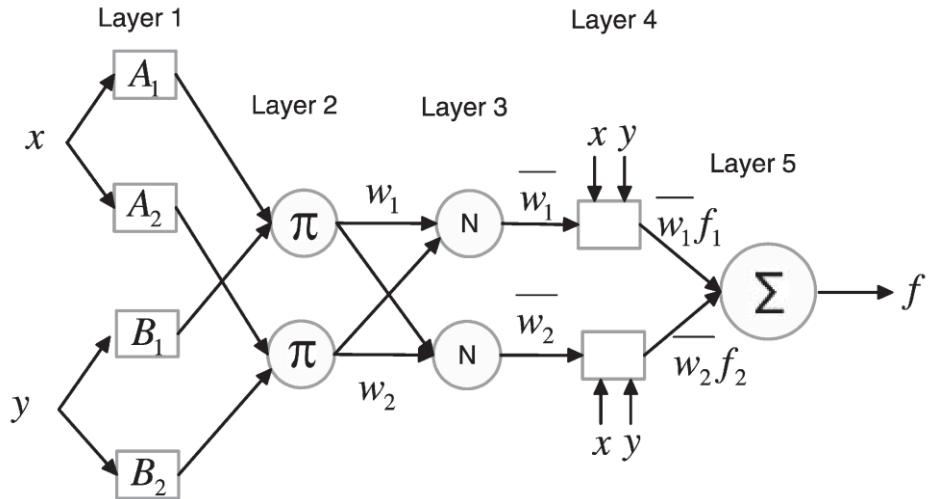


Рис. 1.11. Архитектура ANFIS

Слои ANFIS выполняют следующие функции. Слой 1 представлен радиальными базисными нейронами и моделирует функции принадлежности. Слой 2 — это слой И-нейронов, которые моделируют логическую связку И произведением  $w_i = \mu_{A_i}(x) \mu_{B_i}(y)$ . Слой 3 вычисляет нормированную силу правила:

$$\bar{w}_i = \frac{w_i}{w_1 + w_2}. \quad \text{Слой 4 формирует значение выходной переменной:}$$

$$\bar{w}_i f_i = \bar{w}_i (c_{i1}x + c_{i2}y). \quad \text{Слой 5 выполняет дефазификацию: } f = \bar{w}_1 f_1 + \bar{w}_2 f_2.$$

Гибридная сеть архитектуры ANFIS обучается с помощью алгоритма обратного распространения ошибки.

Нечеткие нейронные сети объединяют преимущества нейронных сетей и нечеткой логики, решая задачи классификации, прогнозирования, управления, аппроксимации сложных функций и применимы для работы с неопределенностью.

К достоинствам нечетких нейросетей относят интерпретируемость, гибкость и качество полученного решения. К недостаткам – сложность настройки, вычислительную сложность и зависимость качества решения от качества и объема обучающей выборки.

Суммарная информация о рассмотренных подходах представлена в табл. 1.8.

В [89, 146, 165, 166] рассматривается оценка качества решения задачи классификации нечеткими системами, созданными с помощью различных подходов. В качестве датасета используется датасет ирисов [82, 98]. Классификация на датасете ирисов является одной из самых популярных и широко используемых задач в машинном обучении.

Таблица 1.8. Описание подходов к генерации базы правил

Название	Описание	Общий принцип генерации	Примеры
Алгоритмы сеточного разбиения	Метод эффективен для низкоразмерных задач, но страдает от экспоненциального роста числа правил при увеличении размерности пространства признаков.	Первоначально области определения входных и/или выходных переменных разбиваются на равные интервалы. Каждый интервал соответствует нечеткому множеству, определяемому функциями принадлежности.	Алгоритмы Абе-Лэна, Ванга-Менделя
Эволюционные (генетические) алгоритмы	Используют популяцию решений, которые эволюционируют через отбор, скрещивание и мутации. Лучшие кандидаты отбираются на основе функции приспособленности, создавая новые поколения, постепенно улучшая решения. Метод эффективен для поиска в большом пространстве решений, но может требовать значительных вычислительных ресурсов и времени.	Хромосомы кодируют отдельные правила/всю базу правил. В качестве целевой функции могут учитываться метрики точности, компактности правил и т. п.	Мичиганский, Питтсбургский подходы. Использование ГА в качестве промежуточного шага вычислений в более сложных алгоритмах.

## Продолжение таблицы 1.8

Разбиение на основе кластеризации	Метод адаптивен к структуре данных, снижает избыточность правил и эффективен для высокоразмерных и сложных задач, но чувствителен к выбору параметров кластеризации. На данном методе основаны развивающиеся нечеткие системы.	Алгоритм группирует входные данные в кластеры с учетом их сходства. Каждый кластер интерпретируется как нечеткое правило, определяемое центром и формой кластера.	
Извлечение правил из нейронных сетей	Являются частью объяснимого ИИ. Метод эффективен для объяснения работы сети, улучшает интерпретируемость, но может требовать сложных вычислений при многослойных сетях.	В зависимости от подхода либо анализируется каждый нейрон и на основе весов и порогов строятся правила, либо нейронная сеть рассматривается как черный ящик и правила извлекаются на основе сопоставления входных данных с выходными.	Декомпозиционный (КТ, Цукимото, через индукцию дерева решений), педагогический (получение правил с использованием выборки, алгоритм Трепана, алгоритм извлечения правил двоичного входа и выхода (BIO-RE) и алгоритм STARE) и эклектический подходы (FERNN)
Извлечение продукционных правил из ассоциативных правил	Ассоциативные правила позволяют выявить скрытые закономерности в данных, отсутствие гарантии получения оптимальной структуры базы правил.	Возможен подход, в котором нечеткий логический вывод Мамдани осуществляется на основе нечетких ассоциативных правил.	
Извлечение правил из деревьев	Склонность деревьев к переобучению, недостаточная гибкость в представлении нечеткости; простая, прямолинейная генерация нечетких правил на основе деревьев решений.	Простая генерация, в ходе которой путь от корня дерева к листу, соединяя узлы связкой «И», используется для генерации нечеткого правила.	

Этот датасет, предложенный Рональдом Фишером в 1936 году, содержит 150 записей, каждая из которых представляет цветок с четырьмя характеристиками: длиной и шириной чашелистиков и лепестков.

Задача состоит в том, чтобы классифицировать цветок на один из трех видов – Setosa, Versicolor или Virginica. Датасет идеально подходит для демонстрации алгоритмов классификации благодаря своей компактности, сбалансированности классов и четкой структуре, что делает его классическим тестовым примером в обучении моделей.

Доли правильных ответов (accuracy) в решении задачи классификации, продемонстрированные нечеткими классификаторами, сгенерированными с помощью различных подходов, представлены в табл. 1.9. В таблице не рассматриваются системы, основанные на извлечении правил из других моделей, так как их точность в большой степени зависит от качества моделей, из которых извлекают правила.

Таблица 1.9. Точность классификации датасета ирисов  
нечеткими классификаторами

Подход	Точность решения	Комментарий
Сеточное разбиение	83.33% [89]	Использовалось адаптивное сеточное разбиение
Генетические алгоритмы	95% [165]	Использовался Мичиганский подход, в качестве функции приспособленности использовалась комбинация критериев покрытия исходных данных правилом, коэффициентов уверенности (определенности) и сложности правила
Разбиение на основе кластеризации	99.81% [146]	В качестве алгоритма кластеризации используются генетические алгоритмы
Нейронечеткие модели	99.98% [166]	Сеть ARTMAP

Как видно из табл. 1.9, алгоритмы на основе кластеризации показывают результаты, сопоставимые с остальными методами или превосходящие их при их относительной простоте и применимости для решения задач с большой размерностью. Поэтому данный подход является перспективным для повышения эффективности процедур и алгоритмов генерации НПС. Кроме того, разбиение на основе кластеризации находит свое активное применение в создании развивающихся НС (evolving fuzzy systems) [113].

Исследованиями в области формирования баз знаний НПС на основе кластеризации занимались такие ученые, как J. Dickerson, B. Kosko, P. P. Angelov, A. Lemos, H. Genther, E. D. Lughofer, И.А. Ходашинский.

Первым этапом данного подхода является кластеризация пространства входных/выходных переменных, которое может осуществляться с привлечением разнообразных методов. Общепринятой классификации методов кластеризации/классификации не существует, но можно выделить несколько групп методов, а именно: разделительные (partitioning) [144], иерархические [156], графовые [187], плотностные [96], модельные [97, 130], эволюционные [169], мягкие [85, 94] методы кластеризации.

Во многих общепринятых классификациях выделяется семейство *метрических алгоритмов*, основанных на понятии расстояния между объектами. При их использовании возникает важная задача выбора метрики. Выбор метрики зависит от многих факторов [7], в том числе от типа исходных данных. Например, при кластеризации текстовой информации часто используется косинусное расстояние, при кластеризации временных рядов – DTW (dynamic time warping), в [36] используется метрика, основанная на логарифме периодограммы. При обработке векторных данных распространенной метрикой является расстояние Евклида, но она имеет свои недостатки, например, чувствительность к выбросам и выявление кластеров в виде сфер. Сравнение некоторых метрик при обработке ЭЭГ осуществлялось в [38]. Частным случаем метрической кластеризации является алгоритм Густавсона-Кесселя [115]. Данный алгоритм в качестве метрики использует расстояние Махalanобиса и позволяет получить кластеры в форме эллипсоидов, ориентированных произвольно. Также известны подходы, позволяющие настраивать метрики в зависимости от данных и особенностей задачи [133]. Другой проблемой, возникающей при использовании метрических алгоритмов кластеризации, является проблема наличия выбросов, или аномальных значений, в данных, поскольку для алгоритмов, не учитывающих влияние выбросов, их наличие может существенно исказить результат. В [7] для задачи классификации отмечается, что, чтобы задача классификации имела решение,

желательно, чтобы данные удовлетворяли так называемой «гипотезе компактности», то есть классы образовывали компактно локализованные подмножества в пространстве объектов. Указанное предположение можно обобщить и на случай кластеризации.

В реальном мире расстояния между объектами измеряются с некоторой погрешностью. Для обработки таких ситуаций было введено понятие нечеткой метрики, где каждому значению расстояния соответствует значение функции принадлежности, определяющей уверенность в том, что объекты находятся на расстоянии меньшем или равном точному значению.

При оценке качества кластеризации используются различные показатели. Часто используемые показатели качества нечеткой кластеризации [3, 14, 85, 185]: *коэффициент энтропии, индекс Кси-Бени, индекс качества разбиения, эффективность разбиения*. Подробное описание данных показателей приведено в главе 3.

### **1.3. Цели и задачи исследования**

Интеллектуальные информационные системы востребованы во многих областях. Повышение уровня интеллектуализации возможно за счет использования моделей представления приближенной и/или качественной информации. Для решения различных задач, включая управление, диагностику, прогнозирование, принятие решений и других используются нечеткие продукционные системы, в которых знания о моделируемом объекте представлены в виде нечетких правил, которые представляют собой условное лингвистическое высказывание вида *если-то* и отражают зависимость выходной переменной от входных переменных на качественном уровне. Основным преимуществом представления знаний в форме продукции является высокая интерпретируемость, что способствует развитию объяснительных способностей интеллектуальных систем и является важным при использовании в тех областях, в которых необходимо объяснение полученного решения, например, в медицине. Ключевым компонентом нечеткой продукционной системы, «отвечающим» за предметную область конкретной прикладной задачи,

является база знаний, которая представляет собой совокупность производственных правил и их параметров. Известно, что ее качество напрямую влияет на точность, интерпретируемость и эффективность работы нечеткой системы, что обосновывает актуальность задачи совершенствования существующих подходов к разработке баз знаний и обеспечения необходимых требований к базе правил. К основным подходам, которые используются на практике для формирования баз знаний, относятся: экспертный подход и подход на основе наблюдаемых данных. В связи с увеличением объема и доступности данных, описывающих процессы и явления различной природы, активным развитием методов машинного обучения актуальной становится задача автоматического и автоматизированного формирования баз знаний нечетких производственных систем с использованием обучающих данных. Можно выделить несколько направлений генерации базы знаний на основе обучающих данных: алгоритмы сеточного разбиения; метаэвристические (в частности, генетические) алгоритмы; разбиение на основе кластеризации; нейросетевой подход. Анализ различных подходов позволил выделить кластерный подход в качестве перспективного для генерации баз знаний, при этом целесообразно использовать метрические алгоритмы, преимуществами которых являются геометрическая интерпретация, возможность выделить нетипичные объекты, возможность осуществить квантизацию пространства признаков, ориентация на широкий класс задач.

Общую постановку задачи можно сформулировать следующим образом: пусть имеется некоторая система  $S$ , для которой выделено множество входов, представленных входными переменными  $\{X_i\}_{i=1,n}$ , и один выход, которому соответствует выходная переменная  $Y$ , т.е. система имеет MISO-структуру. Предположим, что в результате эксперимента или наблюдения сформировано обучающее множество  $\{(x^i, y_i)\}_{i=1,N}$ , где  $x^i \in \mathbb{R}^n$  не содержащее выбросов или аномальных значений. Необходимо, применяя подходящую процедуру кластеризации обучающего множества, выделить кластеры и каждому из них поставить в соответствие производственное правило, при этом модель правила

(лингвистическая, TS-модель, нечеткий классификатор) определяется результатами кластеризации.

При использовании метрических алгоритмов важной является проблема выбора метрики, поскольку он влияет на извлекаемую структуру данных и качество кластеризации.

Цель исследования заключается в совершенствовании методов генерации баз знаний нечетких производственных систем на основе обучающих данных с использованием кластерных процедур. Для достижения указанной цели необходимо решить следующие задачи:

1. Проанализировать подходы к генерации баз знаний НПС и выявить возможности метрических алгоритмов кластеризации для решения данной проблемы.
2. Разработать процедуры для автоматического формирования баз знаний на основе обучающих данных с использованием эллипсоидальной кластеризации.
3. Предложить новые варианты нечетких метрик и исследовать их свойства.
4. Разработать и протестировать программное обеспечение для проведения вычислительного эксперимента и апробации предложенных подходов.

Схема исследования приведена на рис. 1.12.

### Общая характеристика ситуации в сфере инженерии знаний:

- 1) активное использование НПС в качестве ядра интеллектуальных информационных систем для решения различных прикладных задач;
- 2) привлечение экспертов для построения баз знаний;
- 3) наличие коллекций данных в предметных областях, полученных в результате наблюдения или экспериментов;
- 4) интенсивное развитие методов кластерного анализа, ориентированных на обработку больших данных.

### Современные требования к системам, основанным на знаниях:

- 1) повышение уровня интеллектуализации за счет использования представления приближенной и/или качественной информации;
- 2) обоснование новых подходов к построению баз знаний и обеспечение необходимых требований как основы для повышения качества решения прикладных задач;
- 3) развитие объяснительных способностей интеллектуальных систем на основе моделей представления знаний и баз знаний.

**Цель исследования:** совершенствование методов генерации баз знаний нечетких производственных систем на основе кластеризации обучающих данных.

#### Задачи:

1. Проанализировать подходы к генерации баз знаний НС и выявить возможности метрических алгоритмов кластеризации для решения данной проблемы.
2. Разработать процедуры для автоматического формирования баз знаний на основе наблюдаемых данных с использованием эллипсоидальной кластеризации.
3. Предложить новые варианты нечетких метрик и исследовать их свойства.
4. Разработать и протестировать программное обеспечение для проведения вычислительного эксперимента и апробации предложенных подходов.

#### Полученные результаты:

- метод формирования баз нечетких производственных правил на основе эллипсоидальной кластеризации, позволяющий повысить качество аппроксимации в сравнении с известным подходом;
- семейство нечетких параметрических метрик на основе рациональных треугольных норм, позволяющих учитывать структуру данных при использовании метрических алгоритмов кластеризации;
- модель комплексной оценки нечетких метрик, учитывающая «нечеткое большинство» значимых оценок критериев качества кластеризации;
- алгоритм формирования базы знаний нечеткого классификатора энцефалограмм для асинхронного интерфейса «мозг-компьютер»;
- программный комплекс для формирования баз знаний нечетких производственных систем на основе метрических алгоритмов кластеризации с возможностью выбора нечетких метрик.

**Решена научная задача:** Разработка методов и алгоритмов обработки данных, обеспечивающих генерацию баз знаний нечетких производственных систем с улучшенными аппроксимирующими и классифицирующими свойствами на основе предложенных процедур кластеризации при наличии обучающей выборки.

Рис. 1.12. Схема исследования

## **Выводы по первой главе**

1. Анализ использования систем ИИ, основанных на знаниях, позволил выделить НПС как один из важнейших инструментов для решения практических задач широкого спектра. В таких системах свойство интеллектуальности обеспечивается за счет использования моделей представления приближенной и/или качественной информации в форме *если-то* правил. Повышение качества решения прикладных задач возможно на основе разработки новых подходов к построению баз знаний и обеспечения необходимых требований к ним.

2. Рассмотрены различные подходы к генерации баз знаний НПС на основе обучающих данных, проведен их сравнительный анализ и в качестве перспективного выделен кластерный подход, который эффективен в реализации и сопоставим по точности с другими подходами или превосходит их.

3. При использовании кластерного подхода возникает ряд требований к обучающим данным. В частности, данные не должны содержать выбросы, аномальные значения, причем их расположение можно было бы связать с пространственно локализованными областями. Эти требования являются естественными и необходимы в данной задаче для обеспечения высокого качества кластеризации, что позволит, в свою очередь, обеспечить хорошее качество базы знаний НПС.

4. Близость объектов при решении задачи кластеризации эффективно учитывается с помощью функций расстояния, или метрик. В настоящее время существует тенденция при использовании метрических алгоритмов кластеризации использовать такие метрики, которые можно настроить по обучающей выборке, что позволит учитывать структуру и особенности данных. Таким образом, актуальной задачей является разработка подходов к построению новых метрик, содержащих параметры. Именно за счет настройки параметров можно получить такую метрику, которая обеспечит улучшенные свойства метрических алгоритмов.

## ГЛАВА 2. ФОРМИРОВАНИЕ БАЗЫ ЗНАНИЙ НА ОСНОВЕ ЭЛЛИПСОИДАЛЬНОЙ КЛАСТЕРИЗАЦИИ

Построение базы знаний является ключевой задачей при проектировании и реализации нечетких производственных систем. В данном разделе приводится общий метод построения базы знаний, предлагается использовать аппроксимацию кластеров эллипсоидами минимального объема, и представлены результаты вычислительного эксперимента, подтверждающего улучшение точности аппроксимации. Научные результаты, представленные в данной главе, получены в [39, 43, 137, 152].

### **2.1. Метод формирования нечетких производственных правил на основе эллипсоидальной кластеризации**

Анализ, проведенный в первой главе, показал перспективность использования кластерного подхода для генерации баз знаний НПС на основе наблюдаемых данных. Данный подход позволяет получить базу знаний, обеспечивающую высокую точность НПС. Для построения базы знаний с помощью кластеризации предлагается следующая процедура (рис. 2.1):

- 1) предварительная обработка данных;
- 2) кластеризация пространства входных/выходных переменных;
- 3) аппроксимация данных;
- 4) формирование нечетких правил;
- 5) оптимизация модели;
- 6) оценка качества модели.

Рассмотрим более подробно этапы данной процедуры.

Первым этапом является предобработка данных. Результаты исследований алгоритмов кластеризации и построения баз знаний на основе кластеризации показывают, что рассматриваемый метод построения баз знаний НПС чувствителен к наличию выбросов, или аномальных значений, и одним из вариантов решения

проблемы может быть удаление выбросов. Также возможен подход, при котором выбросы описываются дополнительными правилами.



Рис. 2.1. Этапы построения базы знаний на основе процедуры кластеризации

Существуют различные способы поиска аномальных значений [162]. Так как рассматриваемый метод построения баз знаний НПС основан на процедуре кластеризации, целесообразно использовать подход для идентификации выбросов, основанный на нечеткой кластеризации, например, описанный в [80].

Предположим, что результатом работы алгоритма нечеткой кластеризации является матрица нечеткого разбиения  $U$  и матрица  $C$  с центрами кластеров. Вычисление целевой функции происходит по формуле:

$$OF = J(U, c_1, \dots, c_k) = \sum_{i=1}^n \sum_{j=1}^k u_{ij}^m d^2(x_i - c_j),$$

где  $n$  – количество объектов,  $k$  – количество кластеров,  $m$  – экспоненциальный вес, определяющий нечеткость кластеров,  $d$  – выбранная метрика для измерения расстояния между объектами,  $X$  – матрица объектов.

Вычисление выбросов происходит по следующему алгоритму, при этом  $OF$  – значение целевой функции для всего набора данных,  $OF_i$  – значение целевой

функции после удаления  $i$ -й точки,  $DOF_i = OF - OF_i$ ,  $T$  – константа, определяющая порог для идентификации выбросов (в источнике  $T = 1.5$ ).

```

Begin
  SUM = 0
  For each point  $p_i$  in the set
    DO
      Remove  $p_i$  from the set
      Calculate  $OF_i$ 
      Calculate  $DOF_i$ 
       $SUM = SUM + DOF_i$ 
      Return  $p_i$  to the set
    End DO
    AvgDOF =  $SUM / n$ 
    For each point  $p_i$  in the set
      DO
        If  $DOF_i > T \cdot AvgDOF$  then  $p_i$  is outlier
      End DO
    End
  
```

Пример результата работы алгоритма приведен на рис. 2.2 (темно-серым изображены точки данных, идентифицированные как выбросы, светло-серым – остальные точки).

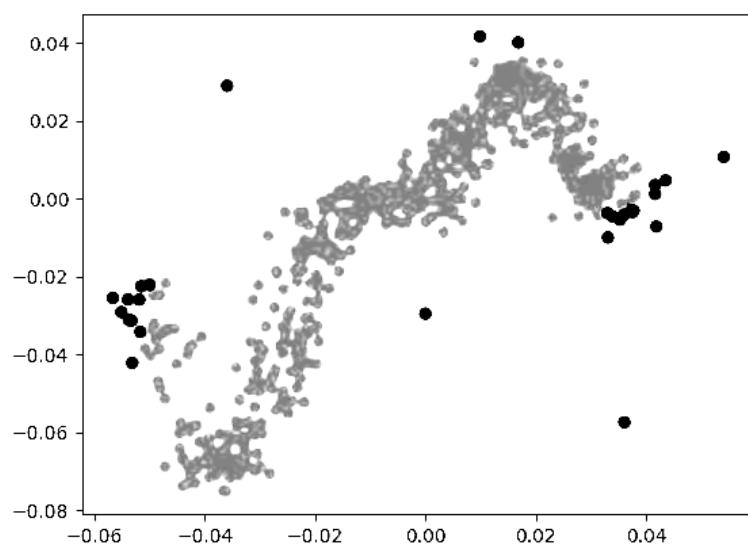


Рис. 2.2. Результат работы алгоритма идентификации выбросов

Выбор метрики также влияет на результат кластеризации. Так, например, евклидова метрика чувствительна к выбросам, тогда как манхэттенское расстояние более устойчиво к выбросам.

Помимо удаления аномальных значений, для работы с данными может потребоваться процедура нормализации [117] для приведения данных к одному масштабу.

*Минмаксная нормализация* (также иногда называется просто *нормализацией*) приводит значения признака  $A$  к отрезку  $[new\_min_A, new\_max_A]$  по формуле:

$$A'_i = \frac{A_i - min_A}{max_A - min_A} (new\_max_A - new\_min_A) + new\_min_A,$$

где  $max_A, min_A$  – исходные границы значений признака,  $new\_max_A, new\_min_A$  – новые границы значений признака. Чаще всего данные приводятся к диапазону  $[0,1]$ .

*Нормализация Z-оценки* (также встречается название *стандартизация*) преобразует значения признака к нулевому среднему и единичному стандартному отклонению по формуле:

$$A'_i = \frac{A_i - \bar{A}}{\sigma_A},$$

где  $\bar{A}$  – среднее значение признака,  $\sigma_A$  – стандартное отклонение значений признака.

Таким образом, при использовании предложенной процедуры можно сформулировать следующие требования к входным данным:

1. Отсутствие выбросов. В данных не должно быть выбросов или аномальных значений, которые значительно отклоняются от остальной массы наблюдений. Для обеспечения качества модели необходимо провести предварительный анализ и обработку выбросов, например, с использованием методов обнаружения выбросов и их последующим удалением.

2. Сопоставимость шкал признаков. Все признаки должны быть приведены к сопоставимым шкалам. Это достигается путём нормализации или стандартизации

данных, что обеспечит одинаковую важность для всех признаков при дальнейшей обработке и применении метрических алгоритмов кластеризации.

3. Для описания данных кластерами данные должны представлять собой *локально плотные* [189] скопления объектов, или *компактно локализованные* подмножества в пространстве объектов [7]. Для оценки локальной плотности существуют различные методы, один из которых, например, оценка плотности методом ближайшего соседа [190]. Локальная плотность в точке  $x_0$  вычисляется по формуле:

$$\hat{f}(x_0) = \frac{k-1}{N \cdot \text{Vol}(\text{Ball}(x_0, d(x_0, x_k)))},$$

где  $k$  – параметр алгоритма (количество рассматриваемых соседей),  $N$  – количество всех точек,  $\text{Vol}(\text{Ball}(x_0, d(x_0, x_k)))$  – объем шара с центром в точке  $x_0$  и радиусом, равным расстоянию от точки  $x_0$  до ее  $k$ -ого ближайшего соседа.

В [189] вводится показатель *относительной локальной плотности* (relative local density), изменяющийся в пределах  $(-0.5, 1)$ ; более высокие значения соответствуют более высокой плотности данных внутри кластеров при низкой плотности между кластерами.

Общий подход к реализации второго шага заключается в регулярном или нерегулярном разбиении входного/выходного пространства. Зачастую используется сеточное равномерное или неравномерное разбиение. Целью применения бессеточных разбиений является сокращение размерности шкал переменных. К таким методам относится использование алгоритмов кластеризации, которые позволяют выделить кластеры, и каждому из них поставить в соответствие производное правило. Могут использоваться метрические алгоритмы, в частности алгоритмы эллипсоидальной кластеризации. Частным случаем эллипсоидальной кластеризации является алгоритм Густавсона-Кесселя [115], который в качестве метрики использует расстояние Махalanобиса и позволяет получить кластеры в форме произвольно ориентированных эллипсоидов. Преимуществом подхода, который заключается в кластеризации обучающего

множества, является возможность проецирования кластеров на оси координат, что позволяет естественным образом сформировать лингвистические шкалы входных/выходных переменных в зависимости от модели продукционных правил. Большинство алгоритмов кластеризации требуют предварительного указания количества кластеров. Для поиска количества кластеров широко используются метод локтя, коэффициент силуэта, а также максимизация (минимизация) показателей качества кластеризации, например, индекса Данна [95], индекса Калински–Харабаша [87], индекса Кси-Бени [185] и др.

После разбиения обучающего множества на кластеры предлагается аппроксимировать их эллипсоидами. Для реализации данного шага можно использовать несколько типов эллипсоидов [4]: эллипсоид главных компонент, эллипсоид минимального объема, эллипсоид с минимальным следом его матрицы, эллипсоид максимального объема, содержащийся в многограннике, эллипсоид Дикина. Широко распространен подход, при котором эллипсоид, описывающий кластер, определяется матрицей ковариаций. В данной работе предлагается использовать эллипсоид минимального объема для осуществления аппроксимации. Рассмотрим теоретическое обоснование данного подхода.

Пусть  $x \in R^n$ ,  $A$  – симметрическая положительно определенная матрица,  $c$  –  $n$ -мерный вектор центра эллипса, тогда определим эллипс выражением

$$E(c, A) = \left\{ x \in R^n : (A^{-1}(x - c), (x - c)) \leq 1 \right\}.$$

Заметим, что данный класс эллипсов характеризуется  $n + \frac{n(n+1)}{2}$

параметрами, при этом  $\frac{n(n+1)}{2}$  параметров задают элементы симметрической матрицы  $A$ , а  $n$  параметров определяют компоненты вектора  $c$ . За счет поворота системы координат ее оси можно совместить с осями эллипса, тогда матрица  $A$  будет диагональной, а неравенство, определяющее эллипс, примет вид

$$\sum_{i=1}^n A_{ii}^{-1} (x_i - c_i)^2 = \sum_{i=1}^n \frac{(x_i - c_i)^2}{d_i^2} \leq 1,$$

где  $d_i$  – полуоси эллипсоида, причем здесь  $A_{ii} = d_i^2$  для  $i = \overline{1, n}$ .

Объем эллипсоида определяется следующим выражением:

$$v_E = \frac{\pi^{n/2} (\det A)^{1/2}}{\Gamma\left(\frac{n}{2} + 1\right)},$$

где  $\Gamma(\cdot)$  – гамма-функция Эйлера.

Известно, что аффинное преобразование пространства  $R^n$  преобразует эллипсоид снова в эллипсоид. Важнейшее прикладное значение эллипсоидов заключается в том, что они используются для аппроксимации произвольных областей. Принципиальная возможность получения аппроксимации множества эллипсоидами экстремального объема опирается на следующее утверждение [11]:

*Теорема 1.* Для любого ограниченного множества  $S$  в  $R^n$  существует единственный эллипсоид  $E^*$  наименьшего объема, содержащий  $S$ .

Формально задачу построения эллипсоида  $E^*$  наименьшего объема можно сформулировать как оптимизационную задачу отыскания такого вектора  $c^*$  и симметрической положительно определенной матрицы  $A^*$  для заданного множества  $S$  в виде

$$\begin{cases} \det A \rightarrow \min \\ S \subset E^* = E(c^*, A^*) \end{cases}$$

Здесь использован тот факт, что объем эллипсоида пропорционален  $\sqrt{\det A}$ .

Пусть в  $R^n$  заданы эллипсоиды  $E(c_1, A_1)$  и  $E(c_2, A_2)$ , тогда можно определить их сумму  $E(c_1, A_1) + E(c_2, A_2)$  как совокупность точек  $x = x^1 + x^2$ , таких что  $x^1 \in E(c_1, A_1)$  и  $x^2 \in E(c_2, A_2)$ . Имеет место следующее утверждение [68].

*Теорема 2.* Параметры эллипсоида  $E(c^*, A^*)$  наименьшего объема, содержащего сумму эллипсоидов  $E(c_1, A_1) + E(c_2, A_2)$ , определяются формулами

$$c^* = c_1 + c_2,$$

$$A^* = (p^{-1} + 1)A_1 + (p + 1)A_2,$$

где  $p$  – единственный положительный корень алгебраического уравнения

$$\sum_{i=1}^n \frac{1}{p + \lambda_i} = \frac{n}{p(p+1)},$$

где  $\lambda_i \geq 0 (i = \overline{1, n})$  – корни характеристического уравнения  $\det(A_1 - \lambda A_2) = 0$ ,

причем каждый корень считается столько раз, какова его кратность.

При обработке реальных данных может случиться, что один из эллипсоидов значительно меньше другого. Пусть, например, длины полуосей эллипсоида  $E(c_1, A_1)$  малы и пропорциональны малому числу  $0 < \varepsilon \ll 1$ , а полуоси эллипсоида  $E(c_2, A_2)$  конечны. Тогда можно положить  $A_1 = \varepsilon^2 A_1^0$ , где  $A_1^0$  – симметрическая неотрицательно определенная матрица. Матрица  $A_2$  второго эллипсоида не зависит от  $\varepsilon$ . Параметры эллипсоида  $E(c^*, A^*)$  наименьшего объема, содержащего сумму  $E(c_1, A_1) + E(c_2, A_2)$  эллипсоидов, когда первый эллипсоид значительно меньше второго, имеют вид [68]:

$$c^* = c_1 + c_2,$$

$$A^* = A_2 + \varepsilon \left( q^{-1} A_1^0 + q A_2 \right),$$

где  $q = \left( n^{-1} \text{Tr}(A_2^{-1} A_1^0) \right)^{1/2}$ , где  $\text{Tr}(\cdot)$  – след соответствующей матрицы.

Таким образом, можно сформулировать следующие свойства эллипсоидов, которые обосновывают их выбор для аппроксимации кластеров:

- а) эллипсоид в  $R^n$  определяется сравнительно небольшим числом параметров;
- б) класс эллипсоидов инвариантен относительно аффинных преобразований;
- в) в общем случае эллипсоиды позволяют получать двустороннюю аппроксимацию произвольных выпуклых множеств [11];
- д) существует эллипсоид минимального объема, который «покрывает» заданное множество точек в  $R^n$ ;
- е) для суммы эллипсоидов существует эллипсоид минимального объема.

Задача нахождения эллипсоида минимального объема подробно рассматривалась в [161, 173, 134, 176]. Авторы [134, 176] для решения задачи построения эллипсоида минимального объема предлагают использовать модифицированный алгоритм Хачияна; в [161] для решения задачи оптимизации предлагается использовать генетический алгоритм смещенного случайного ключа, а в [173] – сочетание метода внутренней точки и метода активного множества.

Рассмотрим задачу построения эллипсоида минимального объема для одного кластера, содержащего некоторое множество точек из  $R^n$ . Пусть  $k$  – количество точек, принадлежащих кластеру;  $P$  – симметрическая положительно определенная матрица, определяющая эллипсоид  $E$  с центром в точке  $c$  объема

$$v_E = w_n / (\det P)^{1/2}, \text{ где } w_n = \frac{\pi^{n/2}}{\Gamma\left(\frac{n}{2} + 1\right)}.$$

Пусть эллипсоид задается формулой

$$E(c, P) = \left\{ x \in R^n : (P(x - c), (x - c)) \leq 1 \right\}.$$

Пренебрегая множителем  $w_n$ , запишем формальную постановку задачи в виде

$$\begin{cases} (\det P^{-1})^{1/2} \rightarrow \min \\ (P(x_i - c), (x_i - c)) \leq 1 \quad (i = \overline{1, k}), \\ P > 0, \end{cases} \quad (2.1)$$

где  $P > 0$  – положительно определенная матрица. Согласно постановке (2.1), требуется определить вектор  $c$  и матрицу  $P$ , которые минимизируют целевую функцию.

Для упрощения (2.1) введем вспомогательную задачу: вложим множество точек  $x_i$  в гиперплоскость  $x_{n+1} = 1$  пространства  $R^{n+1}$ . Обозначим  $q_i = \begin{pmatrix} x_i \\ 1 \end{pmatrix}$ . В новом пространстве будем искать матрицу  $M$ , определяющую эллипсоид с фиксированным центром в нуле. В [134] утверждается, что сечение такого

эллипсоида плоскостью  $x_{n+1} = 1$  и будет решением исходной задачи. Формальная постановка вспомогательной задачи имеет вид:

$$\begin{cases} (\det M^{-1})^{1/2} \rightarrow \min \\ (Mq_i, q_i) \leq 1 \quad (i = \overline{1, k}), \\ M > 0, \end{cases}$$

Преобразуя целевую функцию следующим образом:

$$(\det M^{-1})^{1/2} \rightarrow \min \Leftrightarrow (\det M)^{-1/2} \rightarrow \min \Rightarrow -\ln(\det M) \rightarrow \min,$$

перейдем к задаче

$$\begin{cases} -\ln(\det M) \rightarrow \min \\ 1 - (Mq_i, q_i) \geq 0, i = \overline{1, k}, \\ M > 0. \end{cases} \quad (2.2)$$

Задача (2.2) является задачей выпуклого программирования с линейными относительно  $M$  ограничениями.

Выражения  $q_i^T M q_i$  ( $i = \overline{1, k}$ ) могут быть представлены следующим образом:

$$q^T M q = \sum_{i=1}^n \sum_{j=1}^n m_{ij} q_i q_j = \sum_{i=1}^n \sum_{j=1}^n m_{ij} q q^T_{ji} = \text{Tr}(M q q^T). \quad (2.3)$$

Запишем функцию Лагранжа [53] для задачи (2.2), при этом ограничение  $M > 0$  будем считать нефункциональным и включим его в область определения функции  $L$ .

$$L(M, \lambda) = -\ln(\det M) - \sum_{i=1}^k \lambda_i (1 - q_i^T M q_i), \quad \text{dom } L = \{(M, \lambda) : M > 0, \lambda \in R^k\},$$

где  $\lambda$  – вектор множителей Лагранжа,  $\lambda_i \geq 0$  ( $i = \overline{1, k}$ ).

С учетом (2.3) функция Лагранжа может быть записана в виде

$$L(M, \lambda) = -\ln(\det M) - \sum_{i=1}^k \lambda_i (1 - \text{Tr}(M q q^T)), \quad \text{dom } L = \{(M, \lambda) : M > 0, \lambda \in R^k\}.$$

Рассмотрим некоторые вспомогательные сведения, необходимые для дальнейшего изложения.

В пространстве  $R^{m \times n}$  матриц можно ввести скалярное произведение Фробениуса [99]:

$$\langle A, B \rangle = \text{Tr}(A^T B) = \sum_{i=1}^n \sum_{j=1}^m a_{ij} b_{ij}. \quad (2.4)$$

Пусть  $A$  – фиксированная матрица,  $Y$  – произвольная дифференцируемая матричная функция (согласованные по размерностям, чтобы все операции имели смысл), тогда имеют место следующие производные, определенные в матричном исчислении [148]:

$$\begin{aligned} d\langle A, Y \rangle &= \langle A, dY \rangle, \\ d(\det(Y)) &= \det(Y) \langle Y^{-T}, dY \rangle. \end{aligned}$$

Пусть  $X$  – симметрическая положительно определенная матрица,  $X > 0$ . Найдем производную функции  $f(X) = \ln(\det(X))$ :

$$df(X) = d(\ln(\det(X))) = \frac{d(\det(X))}{\det(X)} = \frac{\det(X) \langle X^{-1}, dX \rangle}{\det(X)} = \langle X^{-1}, dX \rangle,$$

откуда

$$\frac{\partial f(X)}{\partial X} = X^{-1}. \quad (2.5)$$

Найдем производную функции  $f(X) = a^T X a$ , где  $a$  – фиксированный вектор, и, используя (2.4), преобразуем полученное выражение следующим образом:

$$\begin{aligned} df(X) &= d(a^T X a) = d(\text{Tr}(X a a^T)) = d(\text{Tr}(a a^T X)) = d\langle I, a a^T X \rangle = \langle I, d(a a^T X) \rangle = \\ &= \langle I, a a^T dX \rangle = \langle a a^T, dX \rangle, \end{aligned}$$

откуда

$$\frac{\partial f(X)}{\partial X} = a a^T. \quad (2.6)$$

Теперь выпишем условия Куна-Таккера [53], используя формулы (2.5) и (2.6):

$$\begin{cases} \frac{\partial L(M, \lambda)}{\partial M} = \frac{\partial(-\ln(\det M))}{\partial M} - \sum_{i=1}^k \lambda_i \frac{\partial(1 - q_i^T M q_i)}{\partial M} = -M^{-1} + Q \Lambda Q^T = 0 \\ 1 - q_i^T M q_i \geq 0, \\ \lambda_i (1 - q_i^T M q_i) = 0, \quad i = \overline{1, k}, \\ \lambda_i \geq 0, \quad i = \overline{1, k}, \end{cases} \quad (2.7)$$

где  $Q = (q_1, \dots, q_k)$ ,  $\Lambda = \text{diag}(\lambda)$ , откуда

$$M^{-1} = Q \Lambda Q^T. \quad (8)$$

Так как задача (2.2) является задачей выпуклого программирования с линейными ограничениями, то условия Куна-Таккера являются необходимыми и достаточными.

Суммируя  $\lambda_i (1 - q_i^T M q_i) = 0$  по  $i = \overline{1, k}$ , получим

$$\begin{aligned} \sum_{i=1}^k \lambda_i (1 - q_i^T M q_i) &= \sum_{i=1}^k \lambda_i - \sum_{i=1}^k \lambda_i \text{Tr}(M q_i q_i^T) = \sum_{i=1}^k \lambda_i - \text{Tr}\left(\sum_{i=1}^k (\lambda_i q_i q_i^T) Q \Lambda Q^T^{-1}\right) = \\ &= \sum_{i=1}^k \lambda_i - \text{Tr}\left((Q \Lambda Q^T)(Q \Lambda Q^T)^{-1}\right) = \sum_{i=1}^k \lambda_i - \text{Tr}(I) = \sum_{i=1}^k \lambda_i - (n+1) = 0, \end{aligned}$$

где  $I$  – единичная матрица, откуда

$$\sum_{i=1}^k \lambda_i = n+1, \text{ или } \mathbf{1}^T \lambda = n+1.$$

Построим целевую функцию двойственной задачи для задачи (2.2), используя условия (2.7) и (2.8).

$$\begin{aligned} g(\lambda) &= \inf_{M>0} L(M, \lambda) = \inf_{M>0} \left( -\ln(\det M) - \sum_{i=1}^k \lambda_i (1 - q_i^T M q_i) \right) = \\ &= -\ln \det(M) = \ln \det(M^{-1}) = \ln \det(Q \Lambda Q^T). \end{aligned}$$

Двойственная задача имеет следующий вид:

$$\begin{cases} \ln \det(Q \Lambda Q^T) \rightarrow \max \\ \mathbf{1}^T \lambda = n+1, \\ \lambda \geq 0. \end{cases} \quad (2.9)$$

После замены переменных  $u = \frac{\lambda}{n+1}, U = \text{diag}(u)$  в задаче (2.9) получим

$$\begin{cases} \ln \det(QUQ^T) \rightarrow \max \\ \mathbf{1}^T u = 1, \\ u \geq 0. \end{cases} \quad (2.10)$$

Теперь проверим выполнение условия  $M > 0$  в задаче (2.2). Так как обратная к положительно определенной матрице также является положительно определенной, необходимо проверить условие  $Q\Lambda Q^T > 0$ . Зафиксируем произвольный ненулевой вектор  $x \in R^{n+1}$ . По определению, для положительной определённости  $Q\Lambda Q^T$  необходимо, чтобы  $\langle Q\Lambda Q^T x, x \rangle > 0$ . Найдем скалярное произведение

$$\langle Q\Lambda Q^T x, x \rangle = \left\langle \left( \sum_{i=1}^k (\lambda_i q_i q_i^T) \right) x, x \right\rangle = \sum_{i=1}^k \lambda_i \langle q_i q_i^T x, x \rangle = \sum_{i=1}^k \lambda_i \langle q_i, x \rangle^2.$$

Заметим, что точки  $p_i (i = \overline{1, v})$  аффинно зависимы, если существуют

$t_i \in R (i = \overline{1, v})$ , не все равные 0, такие, что  $\sum_{i=1}^v t_i = 0$  и  $\sum_{i=1}^v t_i p_i = 0$ . Если такой

комбинации нет, то точки аффинно независимы. Так как среди точек  $q_i$  содержится аффинно независимое множество, то вектор  $x$  не может быть ортогонален сразу всем векторам  $q_i$ . Значит, если для всех  $i$   $\lambda_i > 0$ , то матрица  $Q\Lambda Q^T$  заведомо является положительно определенной. На самом деле, достаточно, чтобы числа  $\lambda_i$  были положительны для индексов  $i$ , соответствующих аффинно независимому множеству точек  $q_i$ . Будем теперь рассматривать только такие  $\lambda$ , что  $Q\Lambda Q^T > 0$ . Изложенный далее алгоритм, гарантирует выполнение условия  $\lambda > 0$  на каждом шаге.

Пусть  $X$  – матрица, в столбцах которой стоят векторы наблюдаемых данных,  $u$  – решение двойственной задачи (2.10), тогда параметры минимального покрывающего эллипсоида для задачи (2.1) вычисляются по формулам

$$c = Xu,$$

$$P = \frac{1}{n} \left( XUX^T - Xu(Xu)^T \right)^{-1}.$$

Для решения задачи (2.10) воспользуемся алгоритмом Хачияна [134]. В данном алгоритме используются следующие дополнительные обозначения:  $V(u) = QUQ^T$ ,  $e_j$  – нулевой вектор с единичной  $j$ -й координатой,  $EPS$  – требуемая точность. В литературе предлагается различная инициализация вектора решений. Будем использовать в качестве начального приближения вектор  $u = (1/k, \dots, 1/k)$ , определяющий эллипсоид с центром в центроиде всех точек. Так как начальный вектор  $u > 0$ , то гарантируется положительная определенность искомой матрицы и корректность целевой функции. Кроме того, отмечается, что если  $u^i > 0$ , то и  $u^{i+1} > 0$ , что обеспечивает корректность алгоритма.

```

Inputs:  $q_i$ ,  $i = \overline{1, k}$ ;  $EPS$ 
Outputs:  $u$ 
Initialization:  $u := \frac{1}{k} \cdot \mathbf{1}$ 
Do
     $g_i(u) := q_i^T V(u)^{-1} q_i$ ,  $i = \overline{1, k}$ 
     $j := \arg \max_i g_i(u)$ ,  $i = \overline{1, k}$ 
     $\Delta u := e_j - u$ 
     $\varepsilon := \frac{g_j(u) - (n+1)}{n+1}$ 
     $\alpha := \frac{g_j(u) - (n+1)}{(n+1)(g_j(u) - 1)}$ 
     $u := u + \alpha \Delta u$ 
    While  $\varepsilon > EPS$ 
    Return  $u$ 

```

На рис. 2.3 представлены результаты построения покрывающих эллипсов минимальной площади с помощью алгоритма Хачияна.

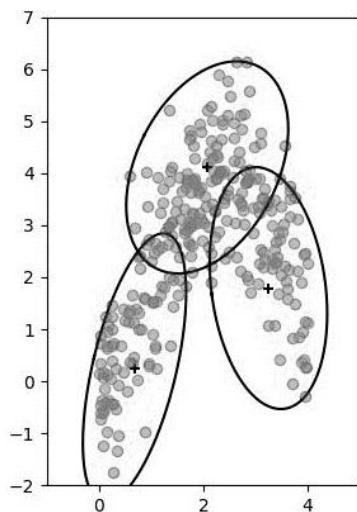


Рис. 2.3. Минимальные покрывающие эллипсы, построенные алгоритмом Хачияна

Следующий этап – получение нечетких правил. После построения эллипсоидов, каждому эллипсоиду можно поставить в соответствие продукционное правило, что позволяет сгенерировать минимально необходимое количество правил. На рис. 2.4 база знаний НПС как бы покрывает график неизвестной функции «заплатками», усредня результат в местах пересечения «заплаток». Если получены такие «заплатки», то их можно объединять, сокращая тем самым количество правил в базе знаний. Функции принадлежности термов лингвистических шкал входных и/или выходной переменных получаются поточечным проецированием матрицы разбиения на оси переменных. Точечно-определенные нечеткие множества затем аппроксимируются подходящей функцией.

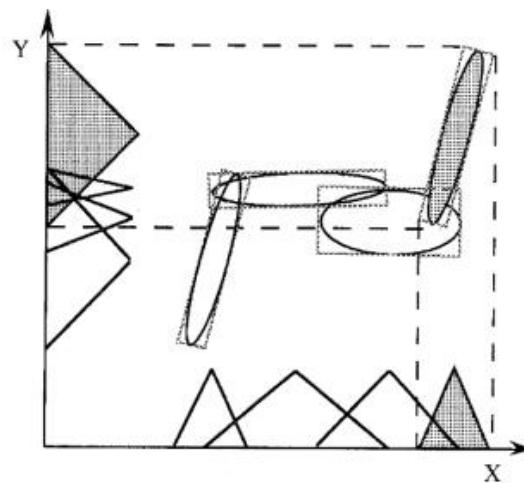


Рис. 2.4. Формирование продукционных правил на основе эллипсоидальной кластеризации

Другой подход [93] заключается в том, что эллипсоид вписывается в  $n$ -мерный параллелепипед, который проецируется на оси переменных. Функция принадлежности задается аналитически с использованием длины проекции и координат центра кластера.

Пусть  $x \in R^n$ ,  $A$  – симметрическая положительно определенная матрица,  $c$  –  $n$ -мерный вектор центра эллипса, тогда определим эллипс в выражением

$$E(c, A) = \left\{ x \in R^n : (A(x - c), (x - c)) \leq \alpha \right\}.$$

Матрица  $A$  представима в виде  $A = P \Lambda P^T$ , где  $\Lambda$  – диагональная матрица с собственными значениями матрицы  $A$ , а  $P$  – матрица, столбцы которой являются соответствующими собственными векторами матрицы  $A$ . Матрица  $P$  является матрицей направляющих косинусов.

Для формирования функций принадлежности термов, соответствующих эллипсу, он вписывается в параллелепипед, который затем проецируется на оси пространства состояний (рис. 2.5). Если  $n$  – количество входов нечеткой системы, а  $p$  – количество выходов, то положим  $q = n + p$ .

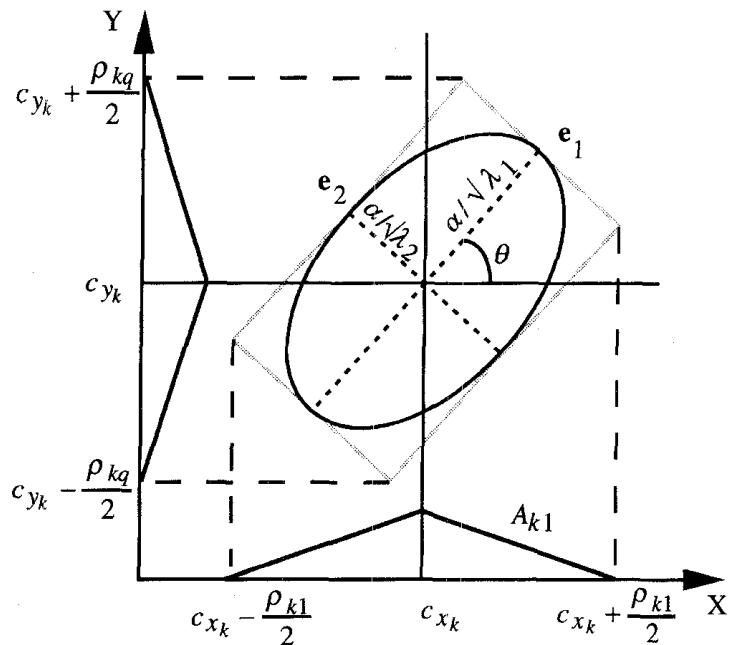


Рис. 2.5. Формирование функций принадлежности термов

Направление косинуса  $\cos \gamma_{ij}$  – это угол между  $j$ -ым собственным вектором и  $i$ -ой осью координат. Проекция параллелепипеда на  $i$ -ую ось отцентрирована на величину  $c_i$  по  $i$ -ой оси и имеет длину

$$\rho_i = 2\alpha \sum_{j=1}^q \frac{|\cos \gamma_{ij}|}{\sqrt{\lambda_j}}.$$

Параметры эллипсоида определяют положение, форму и размер нечеткого правила, значение  $\rho$  определяет носитель треугольного нечеткого числа на  $i$ -ой оси. Ориентация собственных векторов определяет размер проекции. Собственные векторы являются единичными и ортогональными для симметричных положительно определенных матриц [93].

Для  $q = 2$  только одно число ориентирует эллипс, и матрица поворота имеет вид  $P = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$ , тогда проекции определяются следующим образом:

$$\rho_1 = 2\alpha \left( \frac{|\cos \theta|}{\sqrt{\lambda_1}} + \frac{|\sin \theta|}{\sqrt{\lambda_2}} \right), \quad \rho_2 = 2\alpha \left( \frac{|\sin \theta|}{\sqrt{\lambda_1}} + \frac{|\cos \theta|}{\sqrt{\lambda_2}} \right).$$

Заметим, что проекции не используют всю информацию, содержащуюся в эллипсоидальных нечетких «заплатках», такую, как размер и ориентацию эллипсоидов. Улучшение характеристик можно получить путем использования нечетких «заплаток» непосредственно, существенно усложнив при этом вычисления.

Заметим, что в заключениях TS-правил зачастую используются линейные функции. Тем не менее, возможно улучшение модели благодаря использованию функций других типов. В [30] для аппроксимации временных рядов предлагается использовать функции, относящиеся к кривым второго порядка. Основная идея заключается в том, что матрица эллипса  $B$ , полученная в результате кластеризации  $\alpha^2 = (x - c)^T B(x - c)$ , определяет квадратичную форму для каждого кластера. Она

представима в виде  $B = PDP^T$ , где  $D = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$  – матрица, диагональ которой

представлена собственными значениями матрицы  $B$ ,  $P$  – матрица единичных собственных векторов матрицы  $B$ , которая задает угол поворота системы координат  $\theta$ . Пусть  $Y_1, Y_2$  – переменные в новой системе координат, тогда квадратичной форме соответствует каноническое уравнение  $f(Y_1, Y_2) = \lambda_1 Y_1^2 + \lambda_2 Y_2^2$ . На основе значений  $\lambda_1, \lambda_2$  можно определить тип кривой, описывающей кластер, согласно следующим правилам (вырожденные случаи не учитываем):

- a) если  $\lambda_1 > 0, \lambda_2 > 0$ , то кривая представляет собой эллипс;
- b) если  $\lambda_1 \lambda_2 < 0$ , то кривая представляет собой гиперболу;
- c) если  $\lambda_1 = 0$  или  $\lambda_2 = 0$ , то квадратичная форма задает пару параллельных прямых.

После того, как определен тип формы кластера, можно сформировать нечеткое правило. Если кластер описывается прямой, то продукционное правило имеет вид  $R_i : \text{если } x \text{ есть } A_i, \text{то } y = \alpha x + \beta$ . Если кластер описывается гиперболой, то соответствующее правило имеет вид  $R_i : \text{если } x \text{ есть } A_i, \text{то } y = \frac{b}{a} \sqrt{x^2 - a^2}$ , где основной прямоугольник гиперболы ограничен прямыми  $x = \pm a, y = \pm b$ . Если кластер представляет собой эллипс, «сильно вытянутый» вдоль одной из осей, то можно построить четкую или нечеткую функцию регрессии, что позволяет получить правило вида  $R_i : \text{если } x \text{ есть } A_i, \text{то } y = \varphi_i(x)$ , где  $x, y$  – числовые переменные (соответственно входная и выходная), ассоциированные с существенными параметрами кластера;  $\varphi_i(x)$  – функция регрессии для соответствующих переменных. Если кластеру соответствует «объемный» эллипс, то возможна ситуация, когда проецирование на ось выходной переменной не имеет смысла (например, термы накладываются друг на друга), тогда получим продукционное правило  $R_i : \text{если } x \text{ есть } A_i, \text{то } y = C_i$ , где  $C_i$  – метка кластера. Если проекция кластеров на ось выходной переменной позволяет получить лингвистическую шкалу, то получим продукционное правило на основе лингвистической модели.

Для упрощения базы правил, полученной на основе процедур кластеризации, предложена процедура, включающая эвристические правила для преобразования производных правил.

База знаний, полученная в результате процедуры автоматической генерации, как правило, нуждается в корректировке, которая направлена на решение задач структурной и параметрической оптимизации. Упрощенную базу правил можно считать допустимой, если по сравнению с исходной не произошла значительная потеря точности. Использование процедур кластеризации для порождения правил позволяет сформировать разбиение входного пространства и на его основе получить минимальное количество правил, но среди них могут оказаться противоречивые и избыточные правила. Следовательно, необходим анализ посылок и заключений правил с точки зрения корректности лингвистических шкал. Если два правила имеют одинаковые посылки, но разные заключения, то этот факт свидетельствует о наличии конфликтов в базе правил. Возникает вопрос, какое из правил можно оставить, а какое – удалить. Избыточность также проявляется в наличии схожих термов. Схожими считаются термы, значительно перекрывающие друг друга. Для количественной оценки сходства термов используются различные меры сходства нечетких множеств. Различают два подхода к определению сходства – геометрический и теоретико-множественный, причем геометрический основан на использовании расстояния между двумя множествами, и мера сходства определяется по следующей формуле:

$$Sim(A, B) = \frac{1}{1 + D(A, B)},$$

где  $D(A, B)$  – это расстояние между двумя нечеткими множествами  $A$  и  $B$ . Некоторые широко распространенные расстояния для конечного универсального множества носителя  $U$  ( $|U| = n$ ) между нечеткими множествами:

1) расстояние Хэмминга:  $D(A, B) = \sum_{k=1}^n |\mu_A(u_k) - \mu_B(u_k)|;$

2) расстояние Евклида:  $D(A, B) = \sqrt{\sum_{k=1}^n |\mu_A(u_k) - \mu_B(u_k)|^2};$

$$3) \text{ расстояние Минковского: } D(A, B) = \sqrt[p]{\sum_{k=1}^n |\mu_A(u_k) - \mu_B(u_k)|^p}.$$

Также в данной работе в качестве меры сходства предлагается использовать нечеткие метрики, разработанные в главе 3. Если такая оценка превышает заданный порог, то выполняется процедура слияния термов, при этом может быть выбран только один терм, или сконструирован новый терм, в зависимости от носителей схожих термов. Тот случай, когда все термы лингвистической шкалы схожи между собой, означает, что данный признак является неинформативным и может быть исключен из рассмотрения.

Некоторые типовые ситуации, связанные с упрощением базы правил, представлены на рис. 2.6. Заметим, что одним из способов решения перечисленных выше проблем является агрегирование. Так, существуют подходы, основанные на агрегировании конфликтующих правил. Объединение термов в посылках или в заключениях правил выполняется на основе подходящих операций агрегирования. Выбор функции агрегирования должен быть реализован с учетом множества факторов.

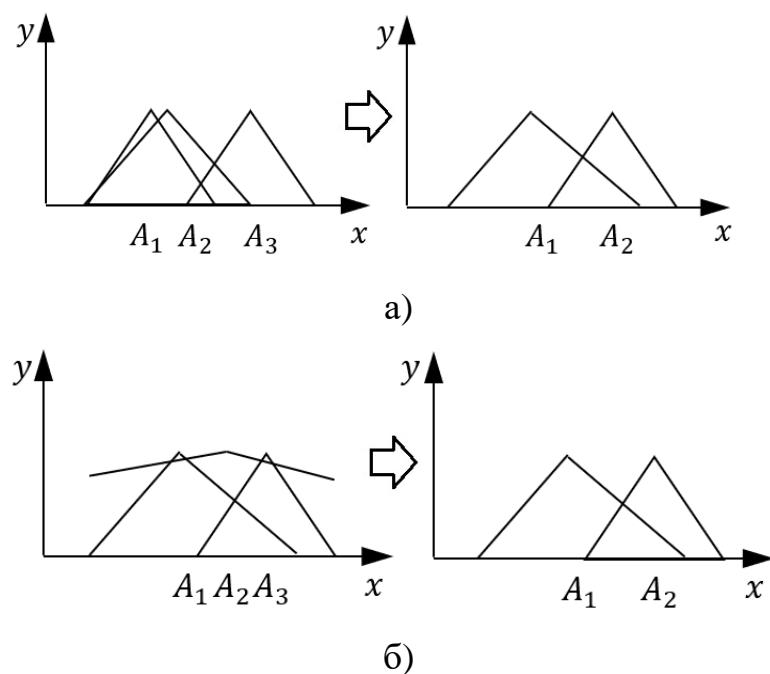


Рис. 2.6. Упрощение правил

## 2.2. Сравнение методов формирования баз знаний на основе эллипсоидальной кластеризации

Для сравнительного анализа методов построения базы знаний с базой правил Такаги-Сугено был проведен вычислительный эксперимент, при этом для обеспечения визуальной оценки результатов рассматривались наборы двумерных данных и строились кластеры в форме эллипсов. Анализ точности осуществлялся на основе величины корня из среднеквадратичной ошибки:

$$RMSE = \sqrt{\frac{1}{M} \sum_{i=1}^M (y_i - y_i^{fuzzy})^2}.$$

Для эмпирического оценивания применялась процедура кросс-валидации, или скользящего контроля, по 5 блокам. Процедура кросс-валидации по  $k$  блокам осуществляется следующим образом: исходные данные разбиваются на  $k$  блоков, равных по размеру. Производится  $k$  итераций. На каждой итерации происходит следующее:

- 1) модель обучается на  $k-1$  части обучающей выборки;
- 2) модель тестируется на той части обучающей выборки, которая не участвовала в обучении.

Таким образом, каждая из  $k$  частей единожды используется для тестирования. На каждой итерации вычисляется ошибка  $RMSE$ , и затем полученные значения усредняются.

В экспериментах использовалось 2000 точек, сгенерированных с помощью следующих тестовых функций (как правило, многоэкстремальных) путем добавления шума:

$$f_1(x) = 3x(x-1)(x-1.9)(x-0.7)(x+1.8), x \in [-2, 2];$$

$$f_2(x) = 10e^{-\left(\frac{x-0.1}{0.25}\right)^2} - 8e^{-\left(\frac{x-0.75}{0.15}\right)^2} - 4e^{-\left(\frac{x-0.8}{0.1}\right)^2}, x \in [-1, 1];$$

$$f_3(x) = 10e^{-\frac{|x|}{0.2}} + e^{-\frac{|x-0.8|}{0.3}} + e^{-\frac{|x+0.6|}{0.1}}, x \in [-1, 1];$$

$$f_4(x) = x + x \sin x, x \in [-7, 5];$$

$$f_5(x) = 8 \sin(10x^2 + 5x + 1), \quad x \in [-0.5, 0.5];$$

$$f_6(x) = 10 \sin(4x + 0.1) + \sin(14x) + \sin(11x - 0.2) + \sin(17x + 0.3), \\ x \in [-0.5, 0.5].$$

Таким образом, формируются двумерные данные, которые будут использоваться для построения системы с одним входом и одним выходом.

На рис. 2.7 изображены тестовые функции с шумом.

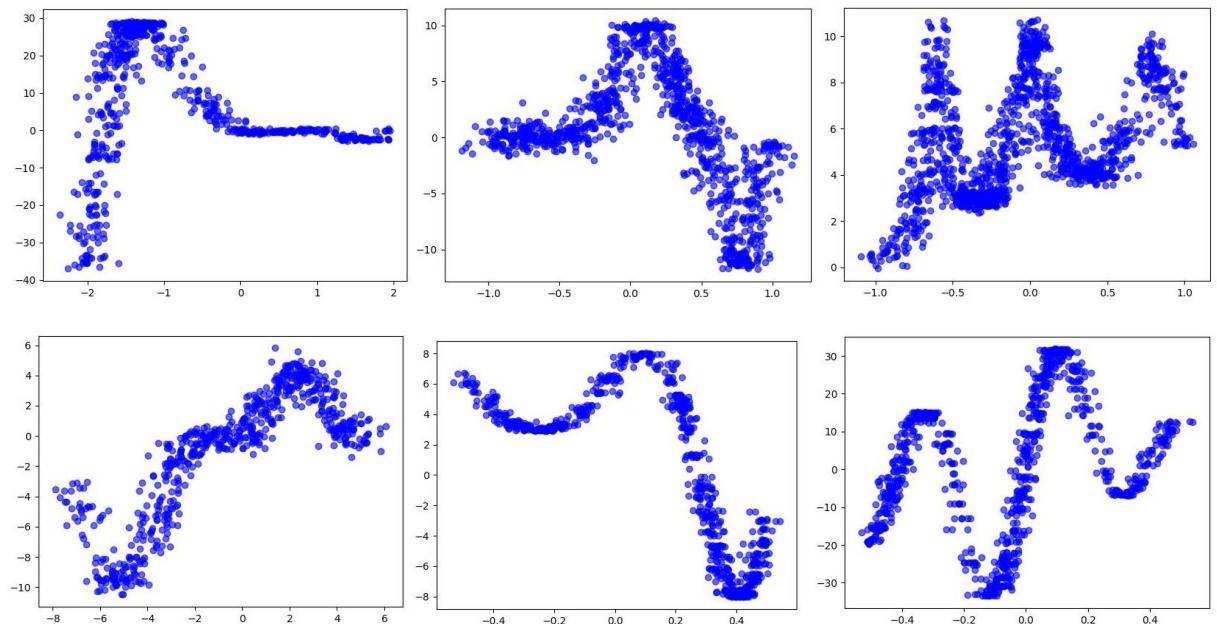


Рис. 2.7. Исходные данные, сгенерированные на основе функций  $f_1 - f_6$   
(слева направо)

Рассмотрим более подробно один из случаев. На рис. 2.8 и 2.9 представлены результаты кластеризации методом Густавсона-Кесселя точек, сгенерированных на основе тестовой функции  $f_4(x) = x + x \sin x, x \in [-7, 5]$ . Эллипсы строились с использованием матриц ковариаций кластеров и минимальных эллипсов соответственно.

Заметим, что расположение кластеров такое, что целесообразно использовать TS-правила, при этом для входной переменной нужно построить лингвистическую шкалу, термы которой используются в посылках правил, а в заключениях правил будут стоять функции, определяемые на основе осей соответствующих эллипсов. В

работе [43] рассматривается построение базы знаний также с правилами лингвистической модели.

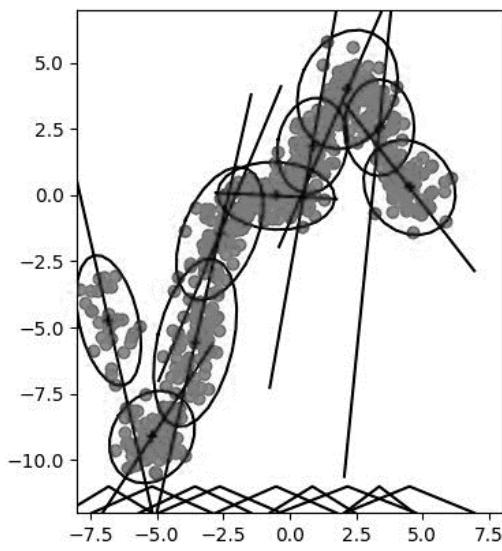


Рис. 2.8. База правил, построенная по матрицам ковариаций, для 9 кластеров

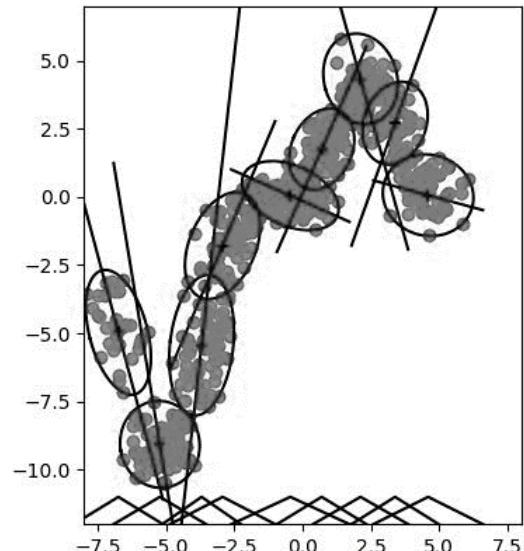


Рис. 2.9. База правил, построенная по минимальным эллипсам, для 9 кластеров

На рис. 2.10 представлена построенная на основе матрицы ковариаций лингвистическая шкала. Здесь термы лингвистической шкалы задаются в форме треугольных нечетких чисел ( $a_i$  – модальное значение;  $l_i, r_i$  – коэффициенты неопределенности, определяющие левую и правую границу соответственно). Важнейшим вопросом является интерпретация полученных термов для входной переменной *величина*. В данном случае считалось, что базовое терм-множество содержит три терма  $T = \{a_1 = \text{малая}, a_2 = \text{средняя}, a_3 = \text{большая}\}$ .

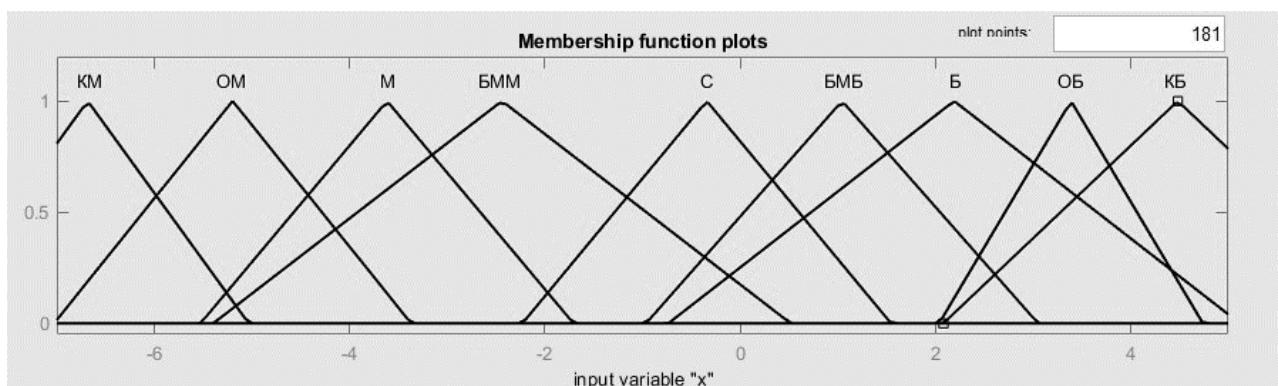


Рис. 2.10. Лингвистическая шкала входной переменной, сгенерированная по матрицам ковариаций

Для порождения новых термов использовались следующие модификаторы: *более или менее* (*БМ*), *очень* (*О*), *крайне* (*К*), что позволило сформировать лингвистическую шкалу для входной переменной в виде

$$\{KM, OM, M, BMM, C, BM\!, B, OB, KB\},$$

где, например, терм *KM* означает *крайне малая*, терм *BMM* – *более или менее малая* и т. д. Функции принадлежности термов в форме треугольных нечетких чисел имеют следующий вид.

$$\begin{aligned} KM &= \begin{pmatrix} -8.33301 \\ -6.68395 \\ -5.03489 \end{pmatrix} & OM &= \begin{pmatrix} -7.02779 \\ -5.19951 \\ -3.37123 \end{pmatrix} & M &= \begin{pmatrix} -5.52715 \\ -3.61744 \\ -1.70773 \end{pmatrix} \\ BMM &= \begin{pmatrix} -5.39600 \\ -2.43992 \\ 0.51615 \end{pmatrix} & C &= \begin{pmatrix} -2.22982 \\ -0.34217 \\ 1.54548 \end{pmatrix} & BM\!B &= \begin{pmatrix} -0.96315 \\ 1.04415 \\ 3.05145 \end{pmatrix} \\ B &= \begin{pmatrix} -0.73421 \\ 2.19217 \\ 5.11857 \end{pmatrix} & OB &= \begin{pmatrix} 2.03946 \\ 3.38994 \\ 4.74043 \end{pmatrix} & KB &= \begin{pmatrix} 2.08290 \\ 4.48389 \\ 6.88487 \end{pmatrix} \end{aligned}$$

База правил, сгенерированная на основе данных, порожденных тестовой функцией  $f_4$ , с использованием матриц ковариаций кластеров (рис. 2.8), имеет следующий вид:

- [1] *если*  $x=KM$ , *то*  $y = -10x - 71.60132$ ;
- [2] *если*  $x=OM$ , *то*  $y = 10x + 42.84763$ ;
- [3] *если*  $x=M$ , *то*  $y = 6.05212x + 16.35037$ ;
- [4] *если*  $x=BMM$ , *то*  $y = 2.08470x + 3.92466$ ;
- [5] *если*  $x=C$ , *то*  $y = 0.08148x + 0.07207$ ;
- [6] *если*  $x=BM\!$ , *то*  $y = 3.13768x - 1.19029$ ;
- [7] *если*  $x=B$ , *то*  $y = 1.34315x + 1.01897$ ;
- [8] *если*  $x=OB$ , *то*  $y = -10x + 36.37498$ ;
- [9] *если*  $x=KB$ , *то*  $y = -1.41547x + 6.62371$ .

На рис. 2.9 кластеры представлены эллипсами минимальной площади, которые получены с помощью алгоритма Хачияна. Здесь также девять кластеров, их проецирование на ось абсцисс позволяет получить соответственно девять термов в форме треугольных нечетких чисел, которые представлены ниже. Их интерпретация будет той же, что и в первом случае, но параметры определяются иначе. Выходная переменная в заключениях правил задается линейной функцией, определяемой на основе соответствующих осей эллипса. На рис. 2.11 представлена построенная на основе эллипсов минимальной площади лингвистическая шкала.

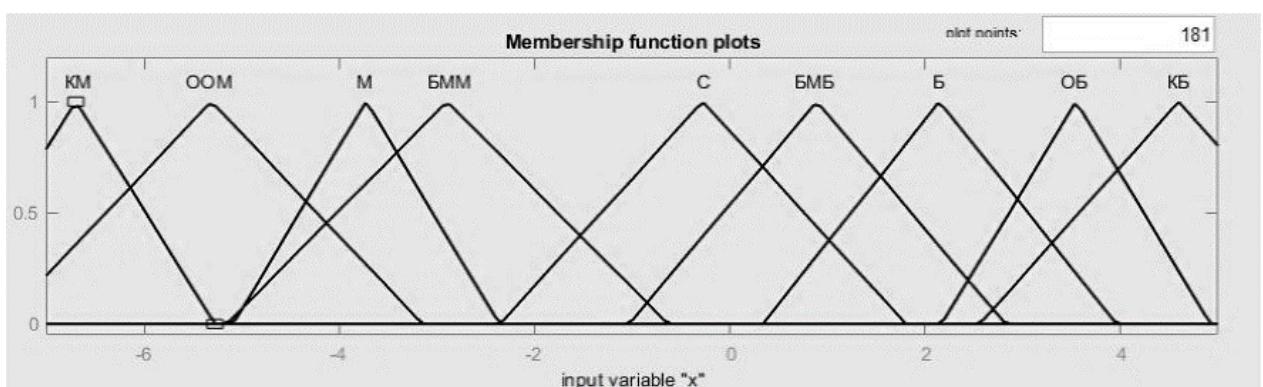


Рис. 2.11. Лингвистическая шкала для входной переменной, сгенерированная по эллипсам минимальной площади

Функции принадлежности термов в форме треугольных нечетких чисел имеют следующий вид.

$$KM = \begin{pmatrix} -8.11680 \\ -6.69428 \\ -5.27176 \end{pmatrix}$$

$$OM = \begin{pmatrix} -7.46619 \\ -5.30839 \\ -3.15059 \end{pmatrix}$$

$$M = \begin{pmatrix} -5.08854 \\ -3.72533 \\ -2.36212 \end{pmatrix}$$

$$BMM = \begin{pmatrix} -5.15348 \\ -2.89479 \\ -0.63610 \end{pmatrix}$$

$$C = \begin{pmatrix} -2.35580 \\ -0.27814 \\ 1.79953 \end{pmatrix}$$

$$BMБ = \begin{pmatrix} -1.03436 \\ 0.89224 \\ 2.81884 \end{pmatrix}$$

$$Б = \begin{pmatrix} 0.33632 \\ 2.14627 \\ 3.95621 \end{pmatrix}$$

$$OB = \begin{pmatrix} 2.16972 \\ 3.54588 \\ 4.92204 \end{pmatrix}$$

$$KB = \begin{pmatrix} 2.55783 \\ 4.59508 \\ 6.63234 \end{pmatrix}$$

База правил включает девять TS-правил следующего вида:

- [1] *если*  $x=KM$ , *то*  $y = -10x - 71.70912$ ;
- [2] *если*  $x=OM$ , *то*  $y = -0.52878x - 11.83678$ ;
- [3] *если*  $x=M$ , *то*  $y = 10x + 31.90432$ ;
- [4] *если*  $x=BMM$ , *то*  $y = 2.13652x + 4.64637$ ;
- [5] *если*  $x=C$ , *то*  $y = -0.63758x - 0.13278$ ;
- [6] *если*  $x=БМБ$ , *то*  $y = 0.92283x + 1.16436$ ;
- [7] *если*  $x=B$ , *то*  $y = 2.43915x + 9.62634$ ;
- [8] *если*  $x=OB$ , *то*  $y = 4.12675x + -12.03818$ ;
- [9] *если*  $x=KB$ , *то*  $y = -0.31180x + 1.49631$ .

В табл. 2.1 приведены величины корня из среднеквадратичной ошибки  $RMSE$  для каждой из тестовых функций для разбиения на 5, 9 и 17 кластеров. Здесь используются следующие обозначения: МК – эллипс на основе матрицы ковариаций, ЭМП – эллипс минимальной площади.

Таблица 2.1. Оценка RMSE

Тестовая функция	5 кластеров		9 кластеров		17 кластеров	
	МК	ЭМП	МК	ЭМП	МК	ЭМП
$f_1$	13.1369	12.5552	11.8493	11.2804	12.8720	12.0148
$f_2$	3.9691	3.8884	3.3668	3.5584	3.3028	3.0823
$f_3$	2.7250	2.6267	2.3193	2.3067	2.2404	2.1668
$f_4$	5.2741	4.4178	1.8016	1.6961	1.7271	1.5669
$f_5$	3.7803	2.4458	2.9305	2.3775	2.5232	1.8707
$f_6$	15.7696	15.4959	15.2297	15.0289	15.2343	14.9577
<b>Среднее RMSE</b>	7.4420	<b>6.9050</b>	6.2495	<b>6.0413</b>	6.3166	<b>5.9432</b>

На основе результатов проведенных экспериментов можно сделать следующие выводы:

1. С увеличением количества кластеров ошибка уменьшается и описание данных становится все более подробным, но, возможно, что за небольшое приращение точности придется заплатить увеличением количества правил в базе, что влияет на быстродействие.

2. При увеличении количества кластеров использование эллипсов минимальной площади в основном дает небольшое улучшение точности при генерации правил типа Такаги-Сугено, но при небольшом количестве кластеров выигрыш оказывается значительным. На рис. 2.12 и 2.13 изображены базы правил, построенные на основе точек, сгенерированных с помощью функции  $f_4$ , для 5 кластеров с использованием матриц ковариаций кластеров и эллипсов минимальной площади. Как видно из рисунков, для меньшего количества кластеров базы правил существенно различаются при использовании различных подходов к построению эллипсов.

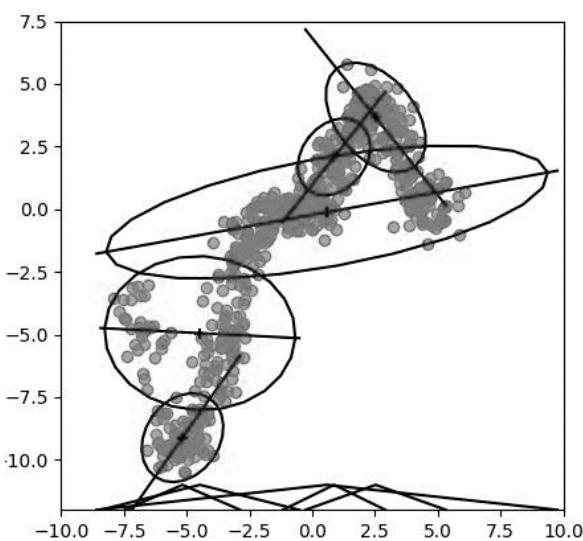


Рис. 2.12. База правил, построенная по матрицам ковариаций, для 5 кластеров

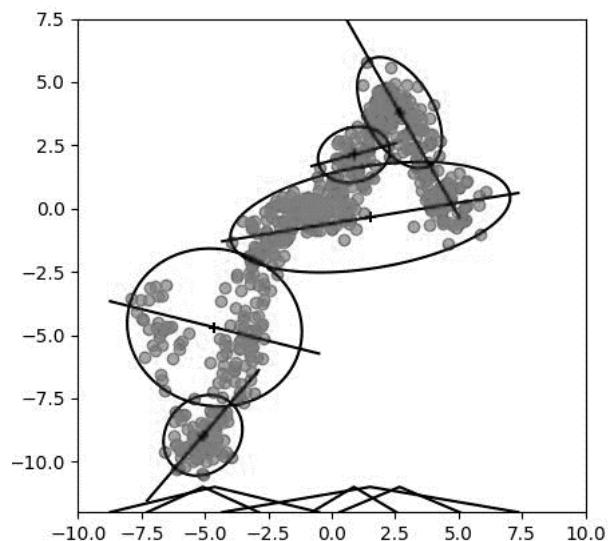


Рис. 2.13. База правил, построенная по минимальным эллипсам, для 5 кластеров

3. Одним из основных свойств базы правил является *интерпретируемость*, которая обеспечивается подходящим количеством правил. Поскольку каждое правило связано с определенным кластером и имеется стратегия на уменьшение количества правил, то необходим обоснованный выбор процедуры кластеризации. Вычислительный эксперимент продемонстрировал преимущества метода

Густавсона-Кесселя по сравнению с методом с-средних. В обоих методах изначально задается количество кластеров. Если данные визуализированы (например, исходные данные – это временные ряды), то эта проблема решаема.

4. Если у полученных эллипсов большая ось значительно превосходит меньшую, то целесообразно использовать правила в виде модели Такаги-Сугено, иначе – лингвистическую модель. Этим выбором обеспечивается улучшение качества аппроксимации.

5. Если получены эллипсы минимальной площади, то можно сократить количество правил за счет сложения соответствующих им эллипсов.

6. Установлено, что функции принадлежности термов, полученные при проецировании эллипсов на ось абсцисс для некоторых тестовых функций, сильно перекрывают друг друга, что ощутимо оказывается на точности аппроксимации (например, этот факт имеет место для функций  $f_1$  и  $f_6$ ). Дальнейшая оптимизация может быть связана с объединением сильно перекрывающихся термов, что требует использования специальных операций над нечеткими множествами, например, параметрических треугольных норм и конорм.

## **Выводы по второй главе**

Процедура формирования базы знаний нечеткой производственной системы на основе наблюдаемых данных с использованием методов кластеризации состоит из нескольких этапов: предобработка данных, кластеризация пространства входных/выходной переменных, аппроксимация данных, формирование нечетких правил, редукция и оптимизация модели, оценка качества модели. Для разбиения пространства входных и выходной переменных используются метрические алгоритмы кластеризации, способные выявить кластеры в форме эллипсоидов. Важной проблемой использования метрических алгоритмов кластеризации является выбор метрики. Кластеры не только отражают структуру данных, но и служат основой для дальнейшей аппроксимации. Эллипсоиды описываются сравнительно небольшим числом параметров, что обуславливает вычислительную эффективность расчетов. Для аппроксимации предлагается использовать

эллипсоиды минимального объема и рассматривается задача нахождения эллипсоида минимального объема, включающего в себя заданное множество точек. Для решения данной задачи используется алгоритм Хачияна. После построения эллипсоидов каждому эллипсоиду ставится в соответствие производственное правило, термы для входных и/или выходной переменных которого получаются путем проецирования эллипсоидов на координатные оси. Для заключений правил модели Такаги-Сугено, как правило, используются линейные функции, но, тем не менее, возможно улучшение модели благодаря использованию функций других типов. База знаний, полученная в результате процедуры автоматической генерации, в большинстве случаев нуждается в корректировке, которая направлена на решение задач структурной и параметрической оптимизации. В частности, среди сгенерированных правил могут оказаться противоречивые и избыточные правила. Для решения этих проблем может быть использована редукция правил на основе мер сходства между термами, для вычисления которых могут применяться различные метрики.

Во второй главе были получены следующие результаты:

1. Проанализированы ограничения метода генерации баз знаний на основе эллипсоидальной кластеризации, а именно: в обучающих данных не должно быть выбросов, они должны иметь сопоставимые по разным признакам масштабы и быть хорошо пространственно локализованными.

2. Представлен общий алгоритм генерации и оптимизации баз производственных правил нечетких систем, основанный на анализе лингвистических шкал входных и/или выходной переменных и позволяющий обеспечить уровень требований как к входной, так и выходной информации.

3. Представлена процедура формирования баз производственных правил на основе эллипсоидальной кластеризации, отличающейся использованием эллипсоидов минимального объема.

4. Проведен вычислительный эксперимент, по результатам которого получено, что предложенная процедура позволяет повысить точность базы правил системы, провести обоснованную редукцию и обеспечить надлежащую

интерпретируемость в отличие от известного подхода, основанного на использовании матриц ковариаций кластеров.

## ГЛАВА 3. НЕЧЕТКИЕ МЕТРИКИ, ОСНОВАННЫЕ НА АДДИТИВНЫХ ГЕНЕРАТОРАХ АРХИМЕДОВЫХ СТРОГИХ ТРЕУГОЛЬНЫХ НОРМ ИЗ КЛАССА РАЦИОНАЛЬНЫХ ФУНКЦИЙ

В данном разделе приводятся определение и свойства нечеткой метрики и рассматривается вопрос построения новых нечетких метрик на основе аддитивных генераторов треугольных норм. Получены ограничения на параметры строгих треугольных норм из класса рациональных функций и семейство построенных на основе их аддитивных генераторов сильных нечетких метрик. Приводятся результаты вычислительного эксперимента, показывающие преимущество использования нечетких метрик в задаче нечеткой кластеризации [27, 139].

### 3.1. Понятие и свойства нечеткой метрики

Пусть  $a$  и  $b$  – объекты заданного множества  $U$ ,  $d : U^2 \rightarrow \mathbb{R}^+ \cup \{0\}$  – функция, которая используется для оценки близости объектов. Рассмотрим следующие условия, на основе которых формируется система аксиом, формализующих требования к  $d$ :

- A1)  $d(a, b) = 0$  тогда и только тогда, когда  $a = b$ ;
- A2)  $d(a, b) = d(b, a)$  для любых  $a, b \in U$ ;
- A3)  $d(a, b) \leq d(a, c) + d(c, b)$  для всех  $a, b, c \in U$ .

Свойство A1) может быть ослаблено, например, следующим образом:

$$A1') \quad d(a, b) = 0 \rightarrow a = b \text{ для любых } a, b \in U \text{ или } A1'') \quad d(a, a) = 0 \text{ для всех } a \in U.$$

Соотношение A3) также допускает менее строгие формы, а именно

$$A3') \quad d(a, b) \leq \rho(d(a, c) + d(c, b)) \text{ или } A3'') \quad d(a, b) \leq \rho \max\{d(a, c), d(c, b)\}.$$

В табл. 3.1 представлены сложившиеся на данный момент определения функций [149], используемых для оценки близости объектов заданного множества

(здесь «+» означает наличие свойства, «-» – отсутствие; либо указана альтернатива основой аксиоме).

Мерой несходства (или несходством) называется симметричное, антирефлексивное отношение [23]. Заметим, что все функции из табл. 3.1 являются антирефлексивными, а большинство симметричны, поэтому, по сути, задают отношения несходства. Дополнением несходства является сходство [23]. Понятия расстояния и метрики зачастую отождествляются [65], поэтому в дальнейшем примем следующее определение: расстоянием (или метрикой) называется мера несходства, которая удовлетворяет аксиоме А3).

Таблица 3.1. Функции для определения близости объектов

Аксиомы	A1)	A2)	A3)
Расстояние	+	+	-
Метрика	+	+	+
Псевдометрика	A1'')	+	+
Преметрика	A1'')	-	-
Квазиметрика	+	-	+
Метаметрика	A1')	+	+
Полуметрика	+	+	A3') или A3'')

Перечислим наиболее известные и широко распространенные функции расстояния [65, 149, 155].

Пусть  $(x_1, \dots, x_n)$  и  $(y_1, \dots, y_n)$  – векторы с числовыми компонентами.

Расстоянием Минковского называется функция

$$d_p(x, y) = \left( \sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}} = \sqrt[p]{|x_1 - y_1|^p + \dots + |x_n - y_n|^p} \quad (3.1)$$

для различных значений  $p \geq 1$ .

В частности, при  $p = 1$ , получим манхэттенское расстояние

$$d_1(x, y) = \sum_{i=1}^n |x_i - y_i| = |x_1 - y_1| + \dots + |x_n - y_n|, \quad (3.2)$$

при  $p = 2$  имеем расстояние Евклида

$$d_2(x, y) = \sqrt{\sum_i^n (x_i - y_i)^2}, \quad (3.3)$$

при  $p \rightarrow \infty$  получается «равномерная» метрика или расстояние Чебышева

$$d(x, y) = \max_{i=1,n} |x_i - y_i|. \quad (3.4)$$

Метрика Канберра при  $x \neq 0$  или  $y \neq 0$  имеет вид

$$d_k(x, y) = \sum_{i=1}^n \frac{|x_i - y_i|}{|x_i| + |y_i|}. \quad (3.5)$$

Расстояние Махalanобиса учитывает взаимодействие признаков

$$d_{M^{-1}}(x, y) = \sqrt{(x - y)^T M^{-1} (x - y)}, \quad (3.6)$$

где  $M = \{m_{ij}\}_{n \times n}$  – ковариационная матрица векторов.

Выбор метрики определяется содержательной постановкой задачи, типом шкал, в которых измеряются признаки; для того, чтобы учесть важность признака, в функцию расстояния вводятся весовые коэффициенты. Если функция расстояния выбрана, то очевидно, что наиболее схожие в содержательном смысле объекты будут представлены близкими точками, и наоборот. Таким образом, качественная неоднородность совокупности данных будет отражена в геометрическом расположении точек в признаковом пространстве. Можно сформулировать следующие особенности, которые должны учитываться при выборе функции расстояния:

- a) недостатком расстояния (3.1) является то, что признаки с наибольшей по мощности шкалой доминируют, поэтому необходима нормализация данных и различные весовые схемы;
- b) расстояние (3.4) стоит использовать, если работа осуществляется с объектами, особенно различающимися по какой-либо одной координате;
- c) расстояние Евклида (3.3) следует применять, если множество данных состоит из «компактных» или «изолированных» кластеров; на величину этого расстояния оказывают влияние отдельные большие выбросы, а также различия между масштабами осей, по координатам которых вычисляется это расстояние;

d) если необходимо учитывать статистическую зависимость между признаками, то целесообразно использовать метрику (3.6); это расстояние используют для получения кластеров в форме эллипсоидов [39].

В условиях неопределенности компоненты векторных оценок объектов могут быть представлены нечеткими числами или, в общем случае, нечеткими множествами. Для обработки такой информации в [132] было введено понятие нечеткой метрики, которое затем, вызвав большой интерес, было развито в работах [90, 106, 109, 151, 184]. Понятие нечеткой метрики и ее свойств оказались полезными при решении различных задач (фильтрация изображений, моделирование многоагентных систем, формирование оценочных процедур для сложных объектов и др.). Известные подходы к определению нечеткой метрики базировались на основе вероятностного определения метрики и метрического пространства. Системный подход порождения нечетких метрик из «обычных» метрик стал возможен благодаря [111]. Он основан на понятии оператора неразличимости [91], представимого с помощью аддитивных генераторов непрерывных архimedовых треугольных норм.

Обобщение классического «неравенства треугольника» при рассмотрении вероятностных метрических пространств привело к понятиям треугольных норм и конорм [128]. Это коммутативные, ассоциативные, монотонные бинарные операции, при этом треугольная норма (*t*-норма)  $T : [0,1]^2 \rightarrow [0,1]$  для всех  $x \in [0,1]$  удовлетворяет условиям  $T(x,1) = T(1,x) = x$ ,  $T(0,0) = 0$  и моделирует операции типа умножения (пересечение нечетких множеств, конъюнкция). Треугольная конорма (*s*-конорма)  $S : [0,1]^2 \rightarrow [0,1]$  моделирует операции типа сложения (объединение нечетких множеств, дизъюнкция), а соответствующие условия имеют вид  $S(x,0) = S(0,x) = x$ ,  $S(1,1) = 1$ . Системы  $\langle [0,1], T, \leq \rangle$  и  $\langle [0,1], S, \leq \rangle$  являются абелевыми полугруппами с нейтральным элементом 0 и 1 соответственно и отношением порядка  $\leq$ .

В [159] введены понятия нечеткой *T*-метрики и нечеткой *S*-метрики.

Пусть  $U$  – произвольное множество,  $T$  – непрерывная треугольная норма,  $M$  – нечеткое множество на  $U \times U \times (0, \infty)$  с функцией принадлежности, удовлетворяющей следующим свойствам для  $\forall x, y, z \in U$  и  $\forall u, v > 0$ :

- a)  $\mu_M(x, y, u) \in [0, 1]$ ;
- b)  $\mu_M(x, y, u) = 1$  тогда и только тогда, когда  $x = y$ ;
- c)  $\mu_M(x, y, u) = \mu_M(y, x, u)$ ;
- d)  $T(\mu_M(x, y, u), \mu_M(y, z, v)) \leq \mu_M(x, z, u + v)$ ;
- e)  $\mu_M(x, y, \cdot) : (0, \infty) \rightarrow [0, 1]$  – непрерывная функция,

тогда  $\mu_M$  называется *нечеткой T-метрикой*, а  $fms = (U, M, T)$  – *нечетким T-метрическим пространством*.

Если при всех  $x, y \in U$  отсутствует зависимость от  $u$ , то есть  $\mu_M(x, y, u) = \mu_M(x, y)$  для любого  $u > 0$ , то нечеткая  $T$ -метрика называется *стационарной*. Если аксиому b) заменить на ослабленную версию  $\forall u > 0 (\mu_M(x, x, u) = 1)$ , то получим определение *нечеткой псевдометрики*, тогда  $fms = (U, M, T)$  – нечеткое псевдометрическое пространство.

Метрика  $\mu_M$  называется *сильной* [159], если для любых  $x, y \in U$  и любого  $u > 0$  имеет место сильная версия неравенства треугольника  $T(\mu_M(x, y, u), \mu_M(y, z, u)) \leq \mu_M(x, z, u)$ .

Заметим, что для сильных метрик из неравенства  $T(\mu_M(x, y, u), \mu_M(y, z, u)) \leq \mu_M(x, z, u)$  (оно выполняется для всех  $x, y \in U$  и любого  $u > 0$ ) следует неравенство  $\max_y T(\mu_M(x, y, u), \mu_M(y, z, u)) \leq \mu_M(x, z, u)$ , которое означает, что  $\mu_M$  удовлетворяет свойству  $(\max-T)$ -транзитивности. Учитывая свойство c) и ослабленную версию b) получим, что  $\mu_M$  можно интерпретировать как нечеткое отношение подобия со специальным типом транзитивности [37]. В [91] отображение с перечисленными свойствами также называется *оператором неразличимости*. Важно различать нечеткие отношения сходства и подобия.

Сходство не обладает свойством транзитивности. Переход от сходства к подобию осуществляется на основе транзитивного замыкания [37], которое определяется через соответствующую композицию.

Пусть  $(T, S)$  – пара двойственных треугольных норм и конорм. Если  $R$  – отношение подобия с  $(\max - T)$ -транзитивностью, то, по аналогии с [23], введем понятие  $(\min - S)$ -расстояния на  $U$  по формуле  $r_R(x, y) = 1 - \mu_R(x, y)$ . Таким образом, задав треугольную норму  $T$ , и функцию принадлежности  $\mu_M$ , можно определить функцию расстояния  $r_M(x, y) = 1 - \mu_M(x, y)$ .

Приведем основные примеры нечетких метрических пространств [111].

Пусть  $(X, d)$  – обычное метрическое пространство, а  $t$ -норма  $T$  – непрерывная, например,  $T(x, y) = \min\{x, y\}$ . Пусть  $M_d$  – нечеткое множество на  $X \times X \times (0, \infty)$ , заданное для  $\forall x, y \in X$  и  $\forall u > 0$  функцией принадлежности вида

$$\mu_{M_d}(x, y, u) = \frac{u}{u + d(x, y)},$$

тогда  $(X, M_d, \min)$  – нечеткое метрическое пространство, а  $\mu_{M_d}$  – стандартная нечеткая метрика, порожденная метрикой  $d$ .

Пусть  $M_e$  – нечеткое множество на  $X \times X \times (0, \infty)$ , заданное для  $\forall x, y \in X$  и  $\forall u > 0$  функцией принадлежности

$$\mu_{M_e}(x, y, u) = e^{-\frac{d(x, y)}{u}},$$

тогда  $(X, M_e, \min)$  – нечеткое метрическое пространство, а  $\mu_{M_e}$  – экспоненциальная нечеткая метрика, порожденная метрикой  $d$ . Множество примеров нечетких метрик рассматривается в [111]. Другие примеры нечетких метрик могут быть найдены, например, в [108, 154].

$$M_{xy}(t) = M(x, y, t) – неубывающая функция при  $\forall x, y \in X$ .$$

Результаты, полученные, например, в [110], позволяют сформулировать некоторые классические теоремы о метрических пространствах для нечетких метрических пространств.

Пусть  $(X, M, *)$ ,  $(Y, N, \diamond)$  – нечеткие метрические пространства.

Отображение  $f : X \rightarrow Y$  называется изометрией, если  $\forall x, y \in X$  и  $\forall t > 0$   $M(x, y, t) = N(f(x), f(y), t)$ . В этом случае, если  $f$  – биективное отображение, то  $(X, M, *)$ ,  $(Y, N, \diamond)$  – изометричные [107]. Изометрия двух пространств означает, что метрические связи между их элементами одни и те же, различной может быть лишь природа их элементов, что с точки зрения теории метрических пространств несущественно [20].

Пусть  $(X, M, *)$  – нечеткое метрическое пространство.

Тогда  $B(x, r, t)$  – открытый шар с центром в точке  $x \in X$  и радиусом  $r, 0 < r < 1, t > 0$ , если  $B(x, r, t) = \{y \in X : M(x, y, t) > 1 - r\}$  [105].

Тогда  $B[x, r, t]$  – замкнутый шар с центром в точке  $x \in X$  и радиусом  $r, 0 < r < 1, t > 0$ , если  $B[x, r, t] = \{y \in X : M(x, y, t) \geq 1 - r\}$  [105].

Множество  $N \subset X$  – ограниченное, если оно содержится целиком в некотором шаре [20].

Точка  $x \in X$  – внутренняя точка  $N$ , если существует  $B(x, r, t)$ , целиком содержащийся в  $N$  [20].

Точка  $x \in X$  – точка прикосновения множества  $N \subset X$ , если любая ее окрестность содержит хотя бы одну точку из  $N$ . Совокупность всех точек прикосновения называется замыканием  $[N]$  этого множества. Операция замыкания – переход от множества  $N$  к его замыканию  $[N]$  [20].

Пусть  $A$  и  $B$  – два множества в  $(X, M, *)$ . Множество  $A$  называется плотным в  $B$ , если  $B \subset [A]$ . Множество  $A$  называется всюду плотным, если его замыкание  $[A]$  совпадает со всем пространством  $(X, M, *)$  [20].

Последовательность  $\{x_n\}$  в  $(X, M, *)$  сходится к  $x$ , если  $\lim_{n \rightarrow \infty} M(x_n, x, t) = 1$  для  $\forall t > 0$  [107].

Множество  $N$ , лежащее в  $(X, M, *)$ , называется замкнутым, если оно совпадает со своим замыканием:  $[N] = N$  [20].

Теорема. Любой замкнутый шар в нечетком метрическом пространстве – замкнутое множество. Доказательство приводится в [105].

Подмножество  $N$  нечеткого метрического пространства  $(X, M, *)$  называется открытым, если  $\forall x \in N \exists r : 0 < r < 1, \exists t > 0 : B(x, r, t) \subseteq N$ , т. е. это множество, все точки которого внутренние [20].

Теорема. Любой открытый шар в нечетком метрическом пространстве является открытым множеством. Доказательство приводится в [105].

Последовательность  $\{x_n\}$  в  $(X, M, *)$  – последовательность Коши, или фундаментальная последовательность, если для любого  $\varepsilon \in [0, 1]$  и  $t > 0$  существует  $n_0 \in N$  такое, что  $M(x_n, x_m, t) > 1 - \varepsilon$  для всех  $n, m \geq n_0$  [105, 107].

Полное нечеткое метрическое пространство – то, в котором любая фундаментальная последовательность сходится [105, 107].

Теорема Бэра. Пусть  $X$  – полное нечеткое метрическое пространство. Тогда пересечение счетного числа плотных множеств также является плотным множеством, или любое полное нечеткое метрическое пространство не может быть представлено как объединение счетного числа нигде не плотных подмножеств. Доказательство приводится в [105].

Пополнение нечеткого метрического пространства  $(X, M)$  – полное метрическое пространство  $(X^*, M^*)$ , такое, что  $(X, M)$  является изометричным подпространству в  $X^*$ , являющимся всюду плотным.  $X$  называется пополняемым, если оно допускает пополнение нечеткого метрического пространства [105].

Теория пополнений нечетких метрических пространств сильно отличается от теории пополнений классических метрических пространств [107].

В [105] показывается, что любая нечеткая метрика  $M$  на  $X$  порождает топологию  $\tau_M$  на  $X$  такую, что ее базой является семейство открытых шаров  $\{B(x, r, t) : x \in X, 0 < r < 1, t > 0\}$ :

$$\tau_M = \{A \subset X : x \in A \Leftrightarrow \exists t > 0, \exists r : 0 < r < 1 : B(x, r, t) \subset A\}$$

Также в [105] доказывается, что любое топологическое пространство  $(X, \tau_M)$  является хаусдорфовым.

Нечеткое метрическое пространство  $(X, M, *)$  называется предкомпактным, если для любого  $r$ ,  $0 < r < 1$ , и всех  $t > 0$  существует конечное подмножество  $A \subset X$ , такое, что  $X = U_{a \in A} A(a, r, t)$  [110].

Нечеткое метрическое пространство является предкомпактным тогда и только тогда, когда каждая последовательность имеет подпоследовательность, которая является последовательностью Коши. Доказательство приводится в [110].

Нечеткое метрическое пространство  $(X, M, *)$  называется компактным, если  $(X, \tau_M)$  является компактным топологическим пространством [110].

Нечеткое метрическое пространство компактно тогда и только тогда, когда оно предкомпактное и полное. Доказательство приводится в [110].

**Принцип сжимающих отображений.** Доказательства обеих теорем приводятся в статье [170].

Пусть  $(X, M, *)$  – полное нечеткое метрическое пространство,  $T$  – отображение пространства  $X$  в себя,  $\varphi : [0, 1] \rightarrow [0, 1]$  строго убывающая и непрерывная слева функция такая, что  $\varphi(\lambda) = 0$  тогда и только тогда, когда  $\lambda = 1, k : (0, \infty) \rightarrow (0, 1)$ . Очевидно, что  $\lim_{\lambda \rightarrow 1^-} \varphi(\lambda) = \varphi(1) = 0$ . Если для  $\forall x, y \in X : x \neq y, \forall t > 0$   $T$  удовлетворяет следующему условию:

$$\varphi(M(Tx, Ty, t)) \leq k(t) \varphi(M(x, y, t))$$

то  $T$  имеет единственную неподвижную точку.

Пусть  $(X, M, *)$  – компактное нечеткое метрическое пространство,  $T$  – непрерывное отображение пространства  $X$  в себя,  $\varphi : [0, 1] \rightarrow [0, 1]$  строго убывающая и непрерывная слева функция такая, что  $\varphi(\lambda) = 0$  тогда и только тогда, когда  $\lambda = 1$ . Если для  $\forall x, y \in X : x \neq y, \forall t > 0$   $T$  удовлетворяет следующему условию:

$$\varphi(M(Tx, Ty, t)) \leq \varphi(M(x, y, t)),$$

то  $T$  имеет единственную неподвижную точку.

### 3.2. Построение нечетких метрик на основе треугольных норм

В [111] предлагаются несколько подходов к конструированию нечеткой метрики. Для начала рассмотрим треугольные нормы, использующиеся в определении нечеткой метрики.

Треугольная норма  $T : [0,1]^2 \rightarrow [0,1]$  – бинарная операция, удовлетворяющая следующим условиям:

- a)  $T(a,b) = T(b,a);$
- b)  $T(T(a,b),c) = T(a,T(b,c));$
- c) если  $a \leq c$  и  $b \leq d$ , то  $T(a,b) \leq T(c,d);$
- d)  $T(a,1) = a, T(0,0) = 0.$

Свойство *архимедовости* треугольных норм связано с нарушением идемпотентности [128]: для всех  $x \in [0,1]$  архимедова треугольная норма  $T$  удовлетворяет условию  $T(x,x) < x$  (subidempotency). Так как треугольные нормы обладают свойством ассоциативности, то имеет место их представление в виде [129]:

$$T(x,y) = t^{(-1)}(t(x) + t(y)), \quad (3.7)$$

где  $t : [0,1] \rightarrow [0,\infty)$  – *аддитивный генератор* t-нормы, под которым подразумевается непрерывная, строго убывающая функция, определенная с точностью до положительной мультипликативной константы и удовлетворяющая условию  $t(1) = 0$ .

В формуле (3.7)  $t^{(-1)}$  – псевдообратная функция, которая определяется следующим образом:

$$t^{(-1)}(x) = \begin{cases} t^{-1}(x), & \text{если } x \in [0, t(0)], \\ 0, & \text{если } x \in (t(0), \infty). \end{cases}$$

Пусть  $T$  – треугольная норма,  $t$  – ее аддитивный генератор. Если  $t(0)=\infty$ , то треугольная норма  $T$  называется строгой; если  $t(0)<\infty$  – нильпотентной [129]. Для строгих  $t$ -норм имеем  $t^{(-1)}(x)=t^{-1}(x)$ .

Перейдем к непосредственно генерации нечетких метрик на основе известных функций расстояния.

Пусть  $U$  – произвольное множество,  $d:U\times U\rightarrow[0,\infty)$  – метрика или псевдометрика на  $U$ ,  $\varphi:(0,\infty)\rightarrow(0,\infty)$  – возрастающая функция,  $T$  – непрерывная архимедова треугольная норма с аддитивным генератором  $t$ , тогда существует нечеткая метрика в форме нечеткого множества  $M$  с функцией принадлежности

$$\mu_M(x,y,u)=t^{(-1)}\left(\frac{d(x,y)}{\varphi(u)}\right), \quad (3.8)$$

при этом в зависимости от свойств функции  $\varphi$  и типа архимедовой треугольной нормы  $T$  получаются различные типы нечетких метрических или псевдометрических пространств. Если  $\varphi$  – непрерывная функция,  $T$  – строгая треугольная норма,  $d$  – псевдометрика или метрика, то  $(U,M,T)$  – соответственно сильное нечеткое псевдометрическое пространство или сильное нечеткое метрическое пространство.

Характеристика основных классов аддитивных генераторов – одна из важнейших проблем функционального представления нечетких операций. Заметим, что единой процедуры для построения аддитивного генератора заданной треугольной нормы или конормы не существует. Однако для некоторых классов треугольных норм и конорм такие подходы предлагаются. В [135, 136, 138] данная проблема решается для треугольных норм и конорм, представимых рациональными функциями, т. е. отношением двух многочленов или, в частном случае, многочленом, в которые переменные входят не выше, чем в первой степени. Установлено, что в качестве аддитивных генераторов рациональных непрерывных

архимедовых треугольных норм (и конорм) могут выступать только функции, общий вид которых основан на дробно-линейной функции:

$$\varphi_1(x) = \frac{ax + b}{cx + d} \text{ (LFAG);}$$

$$\varphi_2(x) = k \cdot \ln \frac{ax + b}{cx + d} \text{ (ln LFAG);}$$

$$\varphi_3(x) = k \cdot \arctg \frac{ax + b}{cx + d} \text{ (arctan LFAG).}$$

Для каждого случая определены ограничения на параметры и построены соответствующие нечеткие операции. Важно, что предложенные параметрические классы обобщают результаты других исследователей, так что известные семейства треугольных норм и конорм являются их частными случаями.

В рамках проведенного исследования были доказаны следующие утверждения, касающиеся аддитивных генераторов строгих треугольных норм.

Утверждение 1. Среди треугольных норм, генераторы которых задаются в форме LFAG, существует единственная строгая треугольная норма

$$T_0(x, y) = \frac{xy}{x + y - xy}, \text{ порожденная генератором } t_\lambda(x) = \frac{\lambda(1-x)}{x}, \text{ где } \lambda > 0.$$

Для  $t_\lambda$  обратная функция имеет вид  $t_\lambda^{-1}(x) = \frac{\lambda}{\lambda + x}$ , где  $\lambda > 0$ .

Исследование аддитивных генераторов в форме ln LFAG позволило установить следующие случаи существования строгой треугольной нормы.

Утверждение 2. Аддитивный генератор  $t_\rho(x) = \frac{1}{1-\rho} \ln \frac{\rho(x-1)+1}{x}$ , где  $\rho < 1$ ,

$\rho \neq 0$  порождает строгую треугольную норму  $T_\rho(x, y) = \frac{xy}{\rho(x+y-xy)+1-\rho}$ .

Для  $t_\rho$  обратная функция имеет вид  $t_\rho^{-1}(x) = \frac{1-\rho}{e^{(1-\rho)x}-\rho}$ .

Утверждение 3. При  $\tau \rightarrow 1$  аддитивный генератор  $t_\tau(x) = -\frac{1}{\tau} \ln(\tau x + 1 - \tau)$ ,

где  $\tau < 1, \tau \neq 0$ , порождает строгую треугольную норму

$$T_\tau(x, y) = \max \left\{ 0, (1 - \tau)(x + y - 1) + \tau xy \right\}.$$

Для генератора  $t_\tau$  обратная функция имеет вид  $t_\tau^{-1}(x) = 1 + \frac{1}{\tau} (e^{-\tau x} - 1)$ .

Утверждение 4. Если хотя бы один из параметров  $\alpha$  и  $\beta$  аддитивного генератора  $t_{\alpha, \beta}(x) = \frac{1}{\beta - \alpha} \ln \frac{\alpha x + 1 - \alpha}{\beta x + 1 - \beta}$ , где  $\alpha, \beta \in (0, 1)$  или  $\beta < 0 < \alpha < 1$  или  $\alpha < 0 < \beta < 1$  принимает значение, близкое к 1, то соответствующая треугольная норма  $T_{\alpha, \beta}(x, y) = \max \left\{ 0, \frac{xy - (1 - \alpha)(1 - \beta)(1 - x)(1 - y)}{1 - \alpha\beta(1 - x)(1 - y)} \right\}$  является строгой.

Обратная функция для  $t_{\alpha, \beta}$  имеет вид

$$t_{\alpha, \beta}^{-1}(x) = \frac{(1 - \alpha) - (1 - \beta)e^{(\beta - \alpha)x}}{\beta e^{(\beta - \alpha)x} - \alpha} = 1 + \frac{1 - e^{(\beta - \alpha)x}}{\alpha - \beta e^{(\beta - \alpha)x}}.$$

На рис. 3.1-3.4 изображены рассмотренные треугольные нормы.

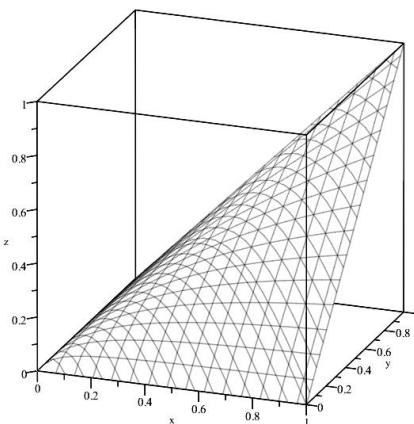


Рис. 3.1. Треугольная норма  $T_0$

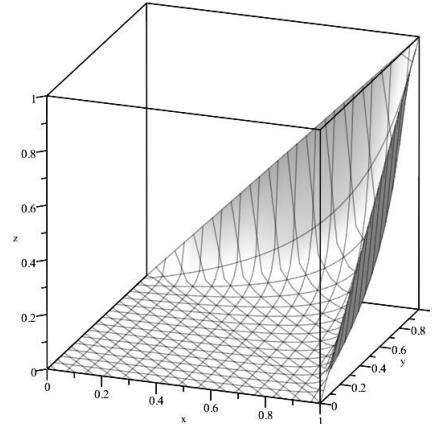


Рис. 3.2. Треугольная норма  $T_\rho$ ,  
 $\rho = -100$

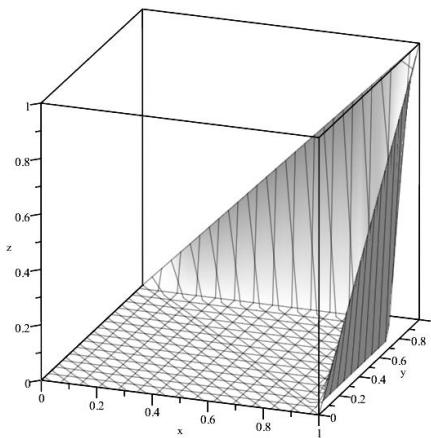


Рис. 3.3. Треугольная норма  $T_\tau$ ,  
 $\tau = -100$

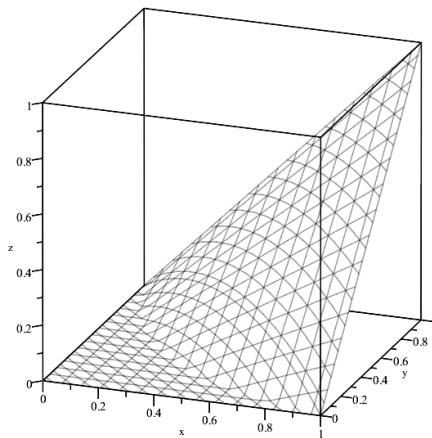


Рис. 3.4. Треугольная норма  $T_{\alpha,\beta}$ ,  
 $\alpha = 0.5, \beta = 0.5$

Таким образом, для сильных архimedовых треугольных норм, относящихся к классу рациональных функций, существует несколько типов аддитивных генераторов, которые могут использоваться для порождения нечетких метрик. Знание аддитивных генераторов строгих треугольных норм позволяет сконструировать множество параметрических метрик на основе формулы (3.8), при этом соответствующие треугольные нормы порождают нечеткие метрические пространства.

Построим нечеткие метрики для рассмотренных выше типов аддитивных генераторов строгих  $t$ -норм в рамках некоторых предположений. В формуле (3.8) в качестве  $d(x, y)$  можно рассматривать следующие известные функции расстояния: расстояние Евклида и обобщающее его расстояние Минковского, манхэттенское расстояние, расстояние Чебышева, метрику Канберра. Поскольку  $\varphi$  – непрерывная возрастающая функция на  $(0, \infty)$ , то можно выбрать  $\varphi(u) = u$ . Результатом данного исследования является следующее

Утверждение 5. В классе рациональных архimedовых строгих треугольных норм нечеткое метрическое пространство  $fms = (U, M, T)$  определяется метриками, которые соответствуют генераторам данных треугольных норм  $T$  и имеют следующий вид:

$$(1) \quad \mu_\lambda(x, y, u) = \frac{u\lambda}{u\lambda + d(x, y)} \text{ при любом } \lambda > 0;$$

$$(2) \quad \mu_{\rho}(x, y, u) = \frac{1 - \rho}{e^{\frac{(1-\rho)d(x,y)}{u}} - \rho} \text{ при } \rho < 1, \rho \neq 0;$$

$$(3) \quad \mu_{\tau}(x, y, u) = 1 + \frac{1}{\tau} \left( e^{-\frac{\tau d(x,y)}{u}} - 1 \right) \text{ при } \tau < 1, \tau \neq 0;$$

$$(4) \quad \mu_{\alpha,\beta}(x, y, u) = \frac{(1-\alpha) - (1-\beta)e^{(\beta-\alpha)\frac{d(x,y)}{u}}}{\beta e^{(\beta-\alpha)\frac{d(x,y)}{u}} - \alpha} \text{ при } \begin{cases} \alpha \rightarrow 1 \\ \beta \rightarrow 1 \end{cases} \text{ или } \max\{\alpha, \beta\} \rightarrow 1,$$

где  $\alpha, \beta \in (0, 1)$  или  $\beta < 0 < \alpha < 1$  или  $\alpha < 0 < \beta < 1$ .

Заметим, что если  $d(x, y) = 0$  (это означает, что  $x = y$ ), то  $\mu(x, y, u) = 1$  для любого  $u$ . В силу симметричности  $d(x, y)$  функции  $\mu$  также будут симметричными. Следовательно, функция принадлежности  $\mu$  каждой из метрик определяет нечеткое бинарное отношение сходства. Согласно [23], дополнение этого отношения является несходством и задает некоторую функцию расстояния – новую метрику. В табл. 3.2 представлены полученные метрики.

Таблица 3.2. Функции от расстояния между объектами для всех типов метрик

Метрика	Функция от расстояния между объектами
Дробно-линейная	$r_{\lambda}(d, u) = \frac{d}{u\lambda + d}$
Логарифмическая-1	$r_{\rho}(d, u) = \frac{e^{\frac{(1-\rho)d}{u}} - 1}{e^{\frac{(1-\rho)d}{u}} - \rho}$
Логарифмическая-2	$r_{\tau}(d, u) = \frac{1 - e^{-\frac{\tau d}{u}}}{\tau}$
Логарифмическая-3	$r_{\alpha,\beta}(d, u) = \frac{e^{(\beta-\alpha)\frac{d}{u}} - 1}{\beta e^{(\beta-\alpha)\frac{d}{u}} - \alpha}$

Заметим, что каждая из функций зависит от  $d \in [0, +\infty)$  и своего параметра. Данные функции являются возрастающими, непрерывными, без экстремумов, с

асимптотой  $y=1$  и нулем в точке 0. При анализе второй производной функции по  $d$  установлено, что для метрики, основанной на LFAG, функция расстояния  $r_\lambda$  не имеет точек перегиба при любых значениях параметров. Для метрики, основанной на логарифме-1, точками, подозрительными на точки перегиба, являются  $d = \frac{u}{\rho-1} \ln\left(-\frac{1}{\rho}\right)$ , где  $\rho < -1, u > 0$ . Метрика, основанная на логарифме-2, не имеет точек перегиба. Метрика, основанная на логарифме-3, имеет подозрительные точки вида  $d = \frac{u}{\alpha-\beta} \ln\left(-\frac{\beta}{\alpha}\right)$ , где  $\begin{cases} \alpha \rightarrow 1 \\ \beta \rightarrow 1 \end{cases}$  или  $\max\{\alpha, \beta\} \rightarrow 1$ , при этом  $\alpha, \beta \in (0, 1)$  или  $\beta < 0 < \alpha < 1$  или  $\alpha < 0 < \beta < 1$ , а также  $\alpha \neq 0, \beta \neq 0, \alpha \neq \beta$ . Проверка достаточного условия существования точки перегиба возможна при конкретных значениях параметров.

Визуализация предложенных метрик позволила выделить несколько характерных поверхностей (рис. 3.5).

Поверхность П1 соответствует функции, которую целесообразно использовать как индикатор при оценке сходства/несходства объектов. Поверхность П3 соответствует ситуации, когда оценка расстояния между объектами формируется с позиции пессимизма, а поверхность П4 – с позиции оптимизма. В случае П3 и П4 возможны значения параметров, когда возникают части поверхности, прилегающие к верхней грани. Это означает, что объекты с соответствующими оценками неразличимы. Наличие точки перегиба свидетельствует о смене позиции – с оптимизма на пессимизм или наоборот. Преимуществом поверхности П2 является то, что она всегда делает все объекты различимыми.

Каждой поверхности можно поставить в соответствие числовую характеристику – уровень различимости/неразличимости по отношению к некоторой базовой функции расстояния. Если за базовую функцию принять евклидово расстояние, то уровень различимости можно вычислить как расстояние между точками базовой поверхности и точками поверхности, образованной заданной функцией расстояния [41].

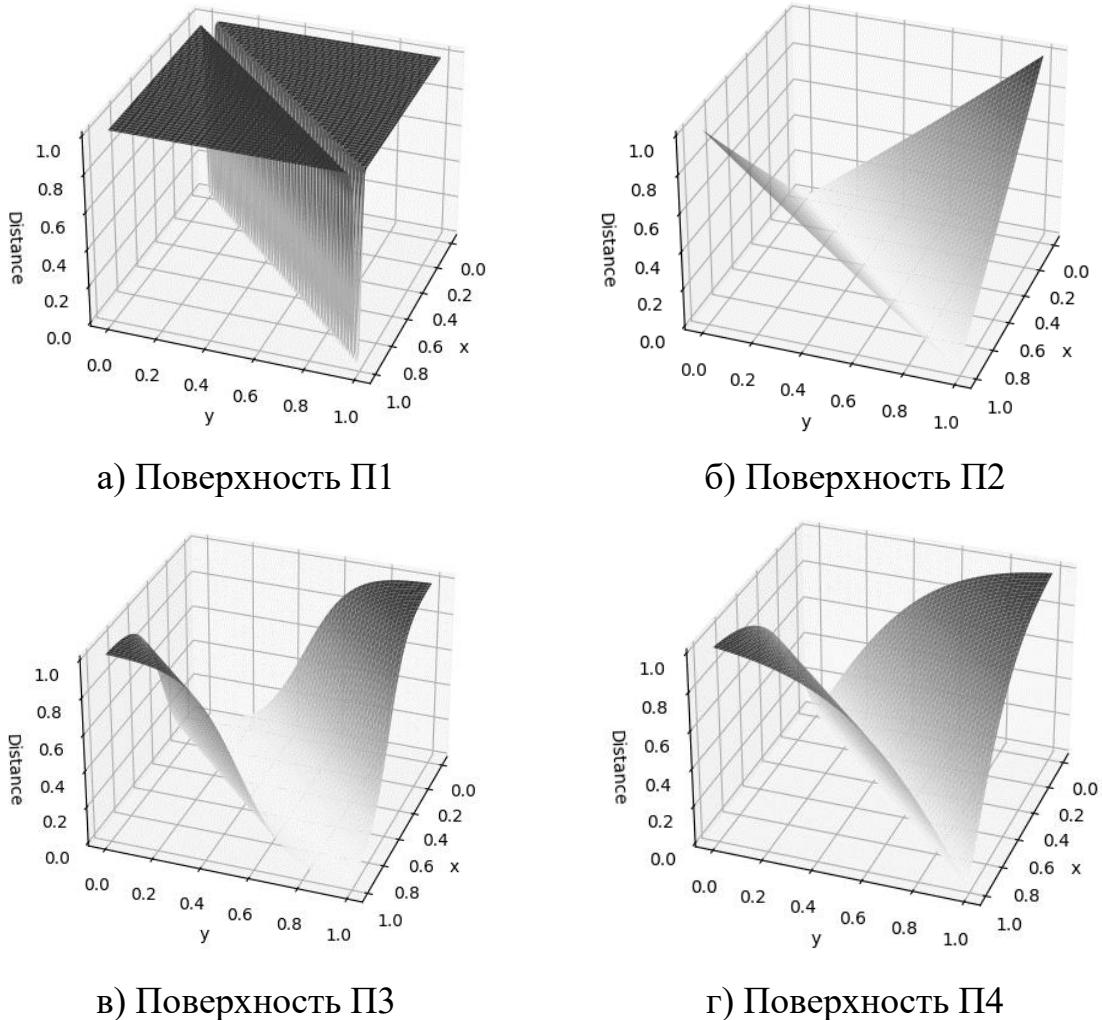


Рис. 3.5. Различные типы поверхностей функции расстояния

### 3.3. Исследование влияния нечетких метрик на решение задачи нечеткой кластеризации

В вычислительном эксперименте рассматривалась задача нечеткой кластеризации, которая заключается в разбиении заданного множества объектов  $X = \{x_i\}_{i=1, N}$  на  $K$  кластеров, при этом объект принадлежит каждому кластеру с определенной функцией принадлежности  $\mu_{ij}$  – степень принадлежности  $i$ -го объекта кластеру  $j$ . В качестве базового метода был выбран алгоритм нечетких k-медоид [157], позволяющий использовать произвольную метрику для получения нечетких кластеров. Входная информация – это матрица расстояний между объектами. Помимо тестируемых метрик использовались известные функции

расстояния Евклида (**E**), Минковского (**M**), Чебышева (**Ч**), а также манхэттенское расстояние (**Mx**) и метрика Канберра (**K**). Результат кластеризации (медоиды  $C = \{c_i\}_{i=1, \overline{K}}$ , а также матрица нечеткого разбиения) оценивался с помощью часто используемых критериев качества кластеризации: *коэффициента энтропии*, *индекса Кси-Бени*, *индекса качества разбиения*, *эффективности разбиения*. Кроме того, производилось приведение кластеров к четкому виду и вычислялся *коэффициент силуэта*.

Вычислительный эксперимент проводился в два этапа: на первом этапе оценивалось влияние метрик на качество кластеризации для разнородных данных, а на втором – для данных, классифицированных по типу. Для оценки качества метрик проводилась кластеризация с заданным изначально количеством кластеров и учитывалось среднее значение 7 показателей качества кластеризации для каждой из метрик: коэффициент энтропии (1), индекс Кси-Бени с метрикой Евклида (2), индекс Кси-Бени с метриками, соответствующими использованным при кластеризации (3), коэффициент силуэта для четких кластеров с метриками, соответствующими использованным при кластеризации (4), коэффициент силуэта для четких кластеров с метрикой Евклида (5), коэффициент разбиения (6), эффективность разбиения с метрикой Евклида (7). В качестве нечеткой метрики использовалась метрика  $r_\rho(d, u)$ .

Рассмотрим более подробно выбранные критерии качества кластеризации.

1. *Энтропия*, известная как количественная оценка степени упорядоченности системы [85], достигает минимума при наибольшей упорядоченности в системе (в случае четкого разбиения энтропия равна нулю), т. е. чем больше степень принадлежности элемента одному кластеру (и меньше степень принадлежности всем остальным кластерам), тем меньше значение энтропии и тем более качественно выполнена кластеризация [3]. *Нормированная энтропия PE* принимает значения в отрезке  $[0,1]$  и определяется формулой

$$PE = -\frac{\sum_{i=1}^N \sum_{j=1}^K \mu_{ij} \ln(\mu_{ij})}{N \ln(K)}.$$

2. Индекс Кси-Бени [185] представляет собой отношение общего разброса к минимальному расстоянию между кластерами и вычисляется по формуле

$$XB = \frac{\sigma(V, C, X)}{N \cdot s(C)},$$

где  $\sigma(V, C, X) = \sum_{i=1}^N \sum_{j=1}^K \mu_{ij} D^2(x_i, c_j)$  – общий разброс;  $s(C) = \min_{i \neq j} \{D^2(c_i, c_j)\}$  –

минимальное расстояние между центрами кластеров;  $D$  – выбранная мера близости между точками данных и центрами кластеров для решаемой задачи. Когда разбиение компактное и «хорошее», значение  $\sigma$  должно быть малым, в то время как значение  $s$  должно быть большим, приводя к уменьшению значения индекса  $XB$ .

3. Коэффициент разбиения [85, 14] определяется формулой

$$P = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^K \mu_{ij}^2.$$

Если каждый объект данных принадлежит только к одному кластеру (в случае четкой кластеризации), то значение индекса максимально и равно 1. Если объекты в равной мере принадлежат всем кластерам и значение принадлежности каждому из них равно  $1/K$ , то значение индекса  $P$  минимально и равно  $1/K$ . Чем больше значение, тем лучше качество разбиения объектов данных на кластеры, т. е. тем лучше выделяются отдельные кластеры данных.

4. Показатель эффективности разбиения определяется по формуле [14]

$$I = \sum_{j=1}^K \sum_{i=1}^N \mu_{ij}^2 (D^2(c_j, \bar{x}) - D^2(x_i, c_j)),$$

причем чем больше его значение, тем лучше выполнена кластеризация.

5. Помимо рассмотренных метрик качества кластеризации, производилось приведение нечеткого разбиения к четкому и вычислялся коэффициент силуэта для четких кластеров по формуле [163]

$$SC = \frac{1}{N} \sum_{i=1}^N SW_i,$$

где  $SW_i = \frac{b_i - a_i}{\max\{a_i, b_i\}}$ ,  $a_i = \frac{\sum_{j \in C_g, i \neq j} D_{ij}}{N_g - 1}$ ,  $b_i = \min_{h \neq g} \left( \frac{\sum_{j \in C_h} D_{ij}}{N_h} \right)$ ,  $N_g$  ( $N_h$ ) – число объектов в  $g$  ( $h$ ) кластере,  $D_{ij}$  – расстояние между объектами  $i$  и  $j$ . Если в кластере один объект, то коэффициент  $SW_i$  для объекта этого кластера считается равным 0. В общем случае коэффициент силуэта принимает значения в отрезке  $[-1, 1]$ , при этом большие значения означают лучшее соответствие объектов кластерам.

Цель вычислительного эксперимента – выявление метрик, дающих наилучшие результаты как по отдельным критериям качества, так и для комплексной оценки, учитывающей все критерии, для данных, имеющих различную структуру.

Вычислительный эксперимент проводился в два этапа: на первом этапе оценивалось влияние метрик на качество кластеризации для разнородных данных, а на втором – для данных, классифицированных по типу. Для оценки качества метрик проводилась кластеризация с заданным изначально количеством кластеров и учитывалось среднее значение 7 показателей качества кластеризации для каждой из метрик: коэффициент энтропии (1), индекс Кси-Бени с метрикой Евклида (2), индекс Кси-Бени с метриками, соответствующими использованным при кластеризации (3), коэффициент силуэта для четких кластеров с метриками, соответствующими использованным при кластеризации (4), коэффициент силуэта для четких кластеров с метрикой Евклида (5), коэффициент разбиения (6), эффективность разбиения с метрикой Евклида (7).

В качестве сравниваемых метрик выступали: евклидово расстояние и семейство метрик на основе  $r_\rho$ , в которых в качестве параметра  $d$  выступали перечисленные ранее известные функции расстояния (Е, М, Ч, Мх, К). Цель вычислительного эксперимента – выявление метрик, дающих наилучшие результаты как по отдельным критериям качества, так и для комплексной оценки, учитывающей все критерии, для данных, имеющих различную структуру.

1. Влияние типа функции расстояния на качество кластеризации относительно критериев (1)-(7) для разнородных данных

В данном эксперименте было сгенерировано 7000 наборов данных, содержащих 4 кластера, которые затем методом  $K$ -медоид подвергались разбиению на заданное число кластеров  $K \in \{2, 3, 4, 5\}$  (параметр алгоритма кластеризации). Кроме того, дополнительно было сгенерировано 7000 наборов данных, содержащих 5 кластеров, для которых строилось разбиение также на 5 кластеров. При этом кластеры генерировались случайным образом, с различной дисперсией и степенью наложения друг на друга. В таблицах 3.3 и 3.4 представлены результаты эксперимента.

Таблица 3.3. Значения оценок качества кластеризации при входных данных, образующих 4 кластера

<b><math>K</math></b>	<b>Метрика</b>	(1)	(2)	(3)	(4)	(5)	(6)	(7)
2	Евкл.	0.3575	<b>0.2624</b>	0.2624	0.6141	0.6141	0.8518	<b>0.0779</b>
	<i>f</i> Е	<b>0.1219</b>	0.2903	<b>0.0966</b>	0.7807	<b>0.6214</b>	<b>0.9533</b>	0.0739
	<i>u</i> К	0.2036	0.3422	0.1851	0.6675	0.5569	0.9136	0.0328
	<i>z</i> Мх	<b>0.1206</b>	0.2918	<b>0.0957</b>	<b>0.7879</b>	0.6188	<b>0.9536</b>	0.0745
	<i>z</i> Ч	0.1242	0.2966	0.1063	0.7774	0.6183	0.9523	0.0715
3	Евкл.	0.2994	<b>0.2713</b>	0.2713	0.6406	0.6406	0.8281	0.1469
	<i>f</i> Е	<b>0.0837</b>	0.2797	<b>0.1162</b>	<b>0.8285</b>	<b>0.6449</b>	<b>0.9516</b>	<b>0.1586</b>
	<i>u</i> К	0.1588	0.4908	0.2098	0.6995	0.5531	0.9041	0.1127
	<i>z</i> Мх	0.0892	0.2805	0.1249	0.8217	0.6420	0.9479	0.1583
	<i>z</i> Ч	0.0873	0.2816	0.1204	0.8252	0.6423	0.9491	0.1583
4	Евкл.	0.3089	0.3930	0.3930	0.5994	0.5994	0.7835	0.1509
	<i>f</i> Е	<b>0.1055</b>	0.3628	<b>0.2512</b>	<b>0.7766</b>	<b>0.6043</b>	<b>0.9207</b>	<b>0.1774</b>
	<i>u</i> К	0.1403	0.7739	0.2590	0.7076	0.5152	0.8985	0.1436
	<i>z</i> Мх	0.1121	<b>0.3627</b>	0.2546	0.7671	0.6012	0.9155	0.1766
	<i>z</i> Ч	0.1092	0.3691	0.2581	0.7723	0.6008	0.9175	0.1768
5	Евкл.	0.3501	0.6179	0.6179	0.5191	0.5191	0.7163	0.1440
	<i>f</i> Е	0.1487	0.5258	0.4909	0.6805	<b>0.5298</b>	0.8674	<b>0.1701</b>
	<i>u</i> К	<b>0.1363</b>	0.9549	<b>0.3216</b>	<b>0.6972</b>	0.4749	<b>0.8876</b>	0.1560
	<i>z</i> Мх	0.1545	<b>0.5201</b>	0.4628	0.6718	0.5239	0.8618	0.1692
	<i>z</i> Ч	0.1523	0.5445	0.5190	0.6772	0.5232	0.8639	0.1693
	<i>y</i> М	0.1517	0.5425	0.5159	0.6775	0.5242	0.8645	0.1694

Таблица 3.4. Значения оценок качества кластеризации для входных данных, образующих 5 кластеров

<i>K</i>	Метрика	(1)	(2)	(3)	(4)	(5)	(6)	(7)
5	Евкл.	0.3337	0.4577	0.4577	0.5610	0.5610	0.7371	0.1550
	<i>f</i> Е	<b>0.1218</b>	<b>0.4147</b>	0.3292	<b>0.7315</b>	<b>0.5667</b>	<b>0.8951</b>	<b>0.1843</b>
	<i>u</i> К	0.1431	0.9568	<b>0.3221</b>	0.6835	0.4733	0.8832	0.1539
	<i>z</i> Мх	0.1296	0.4171	0.3324	0.7200	0.5622	0.8880	0.1830
	<i>z</i> Ч	0.1256	0.4201	0.3309	0.7266	0.5627	0.8913	0.1835
	<i>y</i> М	0.1247	0.4191	0.3313	0.7279	0.5637	0.8922	0.1837

Анализ полученных результатов позволяет сделать следующие выводы.

1) По сравнению с кластеризацией на основе евклидова расстояния (результаты в строке, выделенной серым цветом), кластеризация с нечеткой метрикой на основе различных функций расстояния в подавляющем большинстве случаев дает лучшие значения большинства используемых критериев качества кластеризации.

2) Нечеткая метрика на основе евклидова расстояния дает лучшие значения критериев качества, чем сама евклидова метрика, а, следовательно, можно предположить, что преобразование евклидова расстояния на основе формулы (7) повышает качество кластеризации.

3) Для такой метрики качества, как коэффициент силуэта, вычисленный с использованием евклидовой метрики, после приведения кластеров к четкому виду, нечеткая метрика, основанная на евклидовой функции расстояния, дает лучшие результаты для всех случаев.

Анализ полученных результатов показал, что выбор подходящей метрики кластеризации, а также ее типа, должен осуществляться на основе вида исходных данных и важности отдельных показателей качества кластеризации.

## 2. Влияние характера данных и выбранной метрики на качество кластеризации относительно критериев (1)-(7)

Для оценки качества нечетких метрик проводилась кластеризация при  $K = 4$  для различных типов данных. Всего рассматривалось 3 типа данных: **тип I** –

неперекрывающиеся кластеры с небольшой дисперсией (0.5) (рис. 3.6-а), **тип II** – кластеры с большой дисперсией (1.0) (рис. 3.6-б), которые могут частично перекрываться, и **тип III** – кластеры с большой дисперсией (1.0) (рис. 3.6-в), наложенные друг на друга.

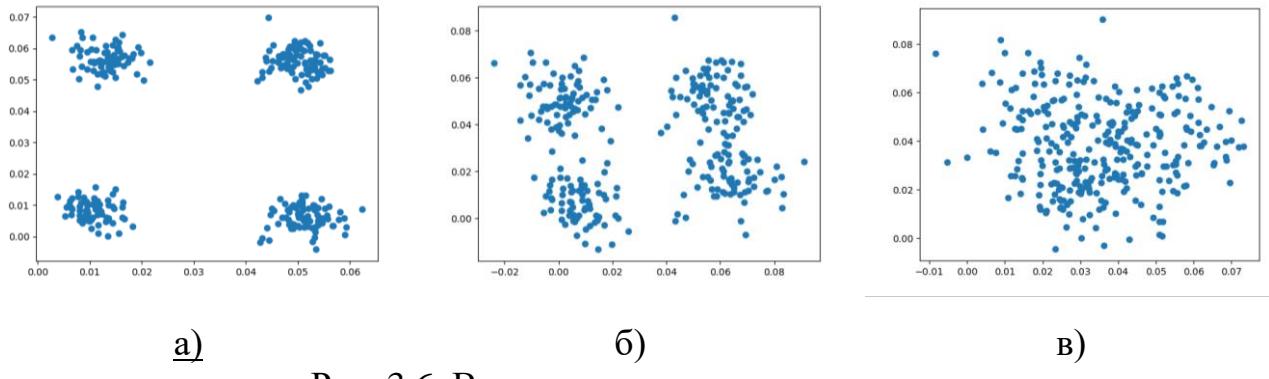


Рис. 3.6. Входные данные разных типов

Для типа I было сгенерировано 200 наборов данных, для типа II – 500, и для типа III – 3500. После этого вычислялось среднее значение показателей качества кластеризации (1)-(7). Значения показателей приведены в таблицах 3.5–3.7.

Таблица 3.5. Показатели качества для кластеров первого типа

Тип	Метрика	(1)	(2)	(3)	(4)	(5)	(6)	(7)
<b>I</b>	Евкл.	0.1222	<b>0.1196</b>	0.1197	0.8364	<b>0.8377</b>	0.9344	0.8513
	<i>f</i> Е	0.0047	0.1226	0.0176	0.9732	<b>0.8377</b>	0.9984	0.9113
	<i>u</i> К	0.0073	0.1230	0.0189	0.9689	<b>0.8377</b>	0.9972	0.9056
	<i>z</i> Мх	0.0085	0.1221	0.0301	0.9611	<b>0.8377</b>	0.9966	0.9141
	<i>z</i> Ч	<b>0.0039</b>	0.1228	<b>0.0138</b>	<b>0.9771</b>	<b>0.8377</b>	<b>0.9987</b>	<b>0.9153</b>
	<i>y</i> М	0.0039	0.1228	0.0140	0.9770	<b>0.8377</b>	0.9987	0.9141

Таблица 3.6. Показатели качества для кластеров второго типа

Тип	Метрика	(1)	(2)	(3)	(4)	(5)	(6)	(7)
<b>II</b>	Евкл.	0.3425	<b>0.2498</b>	0.2498	0.6304	<b>0.6304</b>	0.7735	0.5583
	<i>f</i> Е	0.0646	0.2619	0.0814	0.8531	0.6303	0.9583	0.7154
	<i>u</i> К	0.1061	0.2648	0.0966	0.8082	0.6267	0.9327	0.6804
	<i>z</i> Мх	0.0834	0.2598	0.1292	0.8266	0.6293	0.9452	0.7088
	<i>z</i> Ч	<b>0.0592</b>	0.2640	<b>0.0643</b>	<b>0.8642</b>	0.6296	<b>0.9622</b>	<b>0.7166</b>
	<i>y</i> М	0.0593	0.2641	0.0650	0.8640	0.6297	0.9621	0.7157

Таблица 3.7. Показатели качества для кластеров третьего типа

Тип	Метрика	(1)	(2)	(3)	(4)	(5)	(6)	(7)
III	Евкл.	0.6175	0.5288	0.5288	0.3391	0.3391	0.5475	0.1261
	$f_E$	<b>0.2848</b>	<b>0.4346</b>	0.3117	<b>0.5381</b>	<b>0.3490</b>	<b>0.7864</b>	<b>0.3544</b>
	$u_K$	0.3880	0.4925	<b>0.2844</b>	0.4869	0.3194	0.7281	0.2571
	$z_{Mx}$	0.2956	0.4398	0.3163	0.5257	0.3393	0.7776	0.3455
	$z_{\text{Ч}}$	0.2988	0.4479	0.3305	0.5227	0.3382	0.7756	0.3355
	$y_M$	0.2964	0.4443	0.3261	0.5260	0.3410	0.7776	0.3397

Примеры кластеров для типа данных III, полученных с помощью различных метрик, представлены на рис. 3.7-3.12. Чем светлее цвет точки данных, тем большее значение имеет функция принадлежности в этой точке, точки темного цвета лежат на границах кластеров.

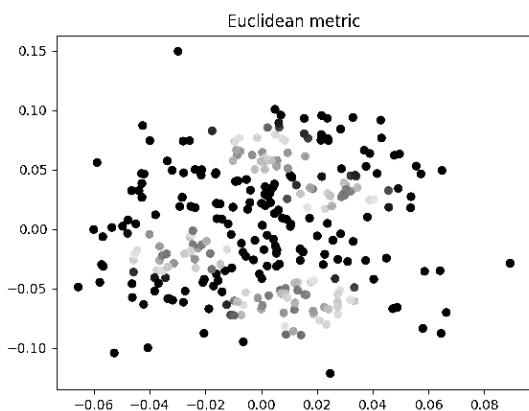


Рис. 3.7. Кластеризация с метрикой Евклида

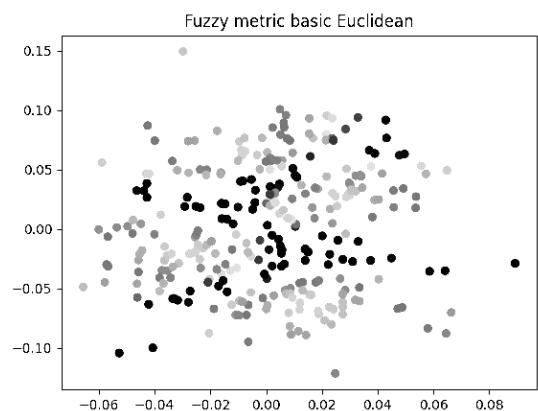


Рис. 3.8. Кластеризация с нечеткой метрикой, основанной на метрике Евклида

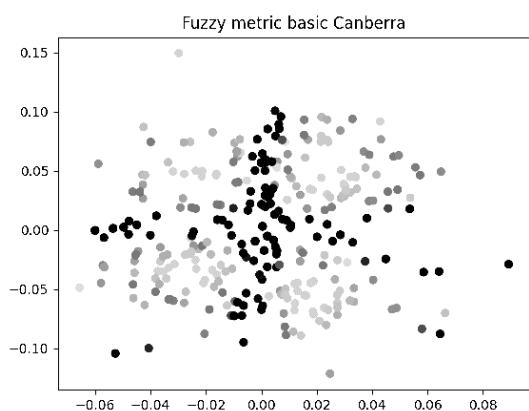


Рис. 3.9. Кластеризация с нечеткой метрикой, основанной на метрике Канберра

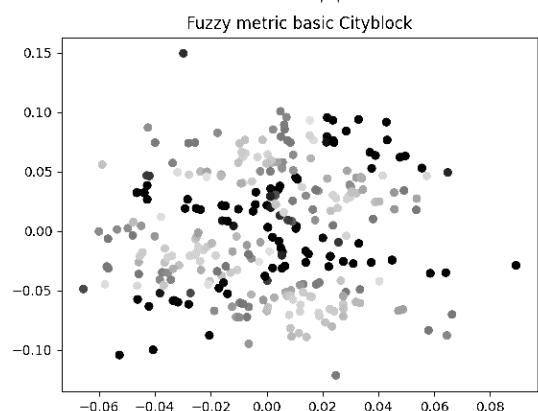


Рис. 3.10. Кластеризация с нечеткой метрикой, основанной на манхэттенском расстоянии

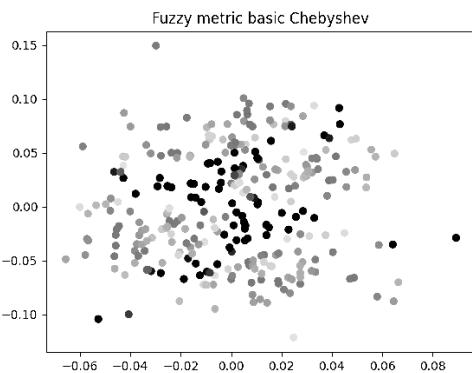


Рис. 3.11. Кластеризация с нечеткой метрикой, основанной на расстоянии Чебышева

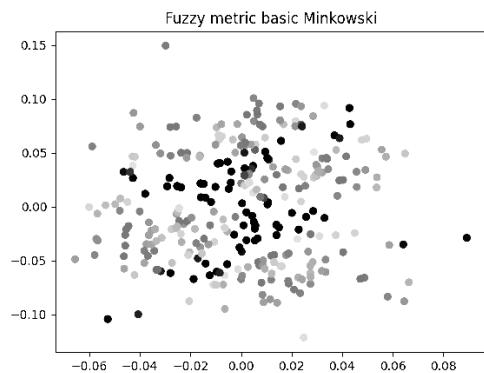


Рис. 3.12. Кластеризация с нечеткой метрикой, основанной на метрике Минковского

Данные иллюстративные примеры показывают, что выбор метрики существенно влияет на результаты кластеризации. В частности, на основе нечетких метрик получаются более «четкие» кластеры с меньшим значением энтропии, что согласуется с результатами экспериментов и подтверждает изложенные выше выводы. Разбиения на рис. 3.8 и рис. 3.10-3.12 визуально имеют только небольшие различия, но и значения критериев качества кластеризации для этих случаев также приблизительно схожи.

Анализ результатов эксперимента позволил сделать следующие выводы.

1. В случае визуально различимых кластеров, возможно, со значительным разбросом внутри, лучшие значения критериев качества кластеризации дает метрика Чебышева, тонко учитывающая разницу между одноименными компонентами векторов данных.

2. Если кластеры не плотные и имеется эффект сильного наложения, то лучшие результаты показывает метрика  $r_\rho$  на основе евклидова расстояния, а, следовательно, она имеет большой потенциал для структурирования данных.

3. Евклидова метрика – лучшая по критериям (2) и (5) для данных I и II типов, когда кластеры, хотя и не плотные, но не пересекаются или пересекаются незначительно.

### 3. Формирование комплексной оценки нечетких метрик по всему набору критериев качества кластеризации

Заметим, что введенные метрики по одним критериям качества могут быть лучше, а по другим хуже остальных. В связи с этим целесообразно использовать комплексные оценки, которые позволяют учитывать частные оценки по всем критериям одновременно. На начальном этапе оценки по критериям необходимо сделать сопоставимыми за счет использования функций нормирования [25]. Если критерий устроен по принципу «чем больше – тем лучше», то используется функция  $x_i^{new} = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}$ , если «чем меньше – тем лучше», то функция

$$x_i^{new} = \frac{x_{\max} - x_i}{x_{\max} - x_{\min}}.$$

На основе анализа критериев установлено, что для критериев 4-7

необходимо использовать первую функцию, а для 1-3 – вторую. В результате нормирования для каждого результата кластеризации будут получены векторные оценки метрик с компонентами из  $[0,1]$ .

В основе «свертки» частных оценок в комплексную лежит использование подходящей функции агрегирования. Самый простой вариант – использование средних [25]. Если необходимо учитывать важность критериев качества кластеризации, то подойдет средневзвешенная средняя. Если все критерии одинаково важны, то, на наш взгляд, перспективным направлением является использование класса порядковых операций агрегирования OWA (Ordered Weighted Averaging Aggregation Operator) [186]. Данный выбор обусловлен наличием ряда числовых характеристик, позволяющих целенаправленно построить процедуру агрегирования с учетом стратегий агрегирования. Пусть  $X = (x_1, \dots, x_n)$  – векторная оценка объекта (в нашем случае это метрика),  $W = (w_1, \dots, w_n)$  – вектор весовых коэффициентов, тогда комплексная оценка имеет вид

$$F_W(X) = \sum_{i=1}^n w_i y_i,$$

где  $Y = (y_1, \dots, y_n)$  – вектор, полученный из вектора  $X$  упорядочением его по невозрастанию.

Заметим, что данный подход отличается от средневзвешенной свертки тем, что весовые коэффициенты не имеют отношения к важности, значимости критериев или показателей, они формализуют различные принципы агрегирования. Обзор операций данного класса представлен в [31]. Например, наиболее известный принцип «нечеткого большинства» можно формализовать с помощью функций квантификаций – непрерывных на  $[0,1]$ , неубывающих функций, удовлетворяющих ограничениям  $Q(0)=0, Q(1)=1$ , тогда на основе выбранной функции  $Q(x)$  однозначно определяется вектор весов. Например, если в качестве функции квантификации выбрать  $Q(x)=x^2$ , то, согласно [31], весовые коэффициенты вычисляются по формулам:  $w_i = \frac{2i-1}{n^2}$ , где  $i = \overline{1, n}$ ,  $n$  – количество весовых коэффициентов. Заметим, что возможны и другие варианты определения весов. Если функция квантификации является строго возрастающей на  $[0,1]$  и дифференцируемой, то, согласно [188], веса определяются по формуле

$$w_i = \frac{Q'(1-y_i)}{\sum_{j=1}^n Q'(1-y_j)} \quad (i = \overline{1, n}),$$

где  $Q'$  – производная функции квантификации.

В отличие от предыдущего случая здесь весовые коэффициенты учитывают еще и значения частных оценок [31]. В рамках проведенного исследования использовались оба подхода, чтобы продемонстрировать важность такого шага процедуры построения комплексной оценки, как выбор функции агрегирования. Комплексные оценки представлены в табл. 3.8. Лучшей считается метрика с максимальным значением комплексной оценки.

Таблица 3.8. Комплексные оценки качества кластеризации

Тип кластеров	Метрика		Комплексная оценка-1	Комплексная оценка-2
1	Евклид		0.0816	0.0
	<i>f</i>	<b>Е</b>	0.7423	0.2375
	<i>u</i>	<b>К</b>	0.6791	0.2131
	<i>z</i>	<b>Mx</b>	0.7410	<b>0.4736</b>
	<i>z</i>	<b>Ч</b>	<b>0.7544</b>	0.0742
	<i>y</i>	<b>M</b>	0.7485	0.0873
2	Евклида		0.0816	0.0
	<i>f</i>	<b>Е</b>	<b>0.7482</b>	0.3520
	<i>u</i>	<b>К</b>	0.4024	0.2606
	<i>z</i>	<b>Mx</b>	0.6515	<b>0.5879</b>
	<i>z</i>	<b>Ч</b>	0.6951	0.1912
	<i>y</i>	<b>M</b>	0.7019	0.1773
3	Евклида		0.0136	0.0351
	<i>f</i>	<b>Е</b>	<b>0.9704</b>	<b>0.8885</b>
	<i>u</i>	<b>К</b>	0.4329	0.3755
	<i>z</i>	<b>Mx</b>	0.8575	0.8017
	<i>z</i>	<b>Ч</b>	0.8112	0.7813
	<i>y</i>	<b>M</b>	0.8529	0.8340

С учетом представленных значений комплексных оценок можно сделать вывод, что

- 1) выбор метрики должен осуществляться на основе анализа исходных данных, подлежащих кластеризации;
- 2) в случае неперекрывающихся кластеров типа I лучшими являются метрики, которые основаны на учете разности между одноименными компонентами исходных векторов;
- 3) чем в большей степени перекрываются кластеры, тем более востребованной становится нечеткая метрика, основанная на евклидовой метрике – так, в соответствии с двумя вариантами расчета комплексной оценки, для перекрывающихся кластеров типа III нечеткая метрика является лучшей.

### Выводы по третьей главе

Метрические алгоритмы кластеризации являются важным инструментом для анализа данных, позволяя группировать объекты на основе выбранных метрик

расстояния. Они широко применяются как предварительный этап для других методов анализа, а также в случае работы с большим объемом информации. При наличии достаточного количества обучающих данных метрические алгоритмы кластеризации могут использоваться для генерации нечетких производственных систем без участия эксперта, позволяя построить разбиение пространства входных/выходных переменных, соответствующее структуре данных. На основе полученного разбиения формируются лингвистические шкалы. При использовании метрических алгоритмов кластеризации важной является проблема выбора метрики, так как выбор метрики влияет на извлекаемую структуру данных и качество кластеризации. Также задача выбора метрики в контексте построения нечетких систем актуальна для оценки схожести термов лингвистических шкал для дальнейшей редукции правил. Параметрические формы метрик играют важную роль в анализе данных, поскольку позволяют адаптировать меры расстояния к специфике рассматриваемой задачи. На практике встречаются ситуации, когда точное измерение расстояния невозможно. Для учета подобных ситуаций применяются различные подходы, в том числе нечеткий. В статье [111] предлагается подход к построению нечетких метрик на основе обычных метрик, что открывает возможности для создания новых нечетких метрик на основе аддитивных генераторов треугольных норм.

К основным результатам третьей главы относятся следующие.

1. Совокупность ограничений на параметры, определяющие аддитивные генераторы строгих треугольных норм, позволяющие генерировать нечеткие метрики, определяемые сильным нечетким псевдометрическим пространством.

2. Совокупность нечетких метрик, полученных на основе аддитивных генераторов архimedовых непрерывных треугольных норм из класса рациональных функций, отличающихся набором настраиваемых параметров, что позволяет учитывать структуру данных при использовании метрических алгоритмов кластеризации.

3. Проведен вычислительный эксперимент, демонстрирующий преимущества использования разработанных нечетких метрик при решении задачи нечеткой кластеризации.

4. Подход к формированию комплексной оценки качества кластеризации, основанный на использовании специальных функций агрегирования и позволяющий учитывать «нечеткое большинство» критериев качества кластеризации.

## ГЛАВА 4. ПРОГРАММНЫЙ КОМПЛЕКС «FuzzyLogicCore» И ЕГО АПРОБАЦИЯ ДЛЯ РЕШЕНИЯ ПРАКТИЧЕСКИХ ЗАДАЧ

### 4.1. Описание программного комплекса «FuzzyLogicCore»

В данной главе приведено описание программного комплекса «FuzzyLogicCore», предназначенного для выполнения автоматизированного формирования баз знаний на основе набора экспериментальных данных для построения моделей нечеткого классификатора, лингвистической и Такаги-Сугено. В данном программном комплексе реализована следующая функциональность:

- алгоритмы кластеризации нечетких с-средних, Густавсона-Кесселя, метод декомпозиционного дерева и алгоритм кластеризации нечетких K-медоид;
- библиотеки обычных и нечетких метрик (реализованы все разработанные в гл. 3 нечеткие метрики);
- алгоритмы построения эллипсоидов на основе матриц ковариаций кластеров и вычисления эллипсоидов минимального объема, включающих все точки каждого кластера;
- построение треугольных функций принадлежности на основе построенных эллипсоидов;
- построение заключений правил Такаги-Сугено с использованием осей эллипсов;
- осуществление нечеткого логического вывода;
- осуществление интерполяции нечетких правил для модели нечеткого классификатора;
- вспомогательный код по загрузке данных, оценке качества решения задачи по различным критериям и визуализации результатов.

Разработанный программный комплекс (ПК) относится к классу прикладного ПО. Для реализации использовались языки Java и Python.

При проектировании ПК были учтены основные требования, предъявляемые к архитектуре программных продуктов в современных условиях:

поддерживаемость, расширяемость, закрытость от изменений, модульность, повторное использование кода.

ПК «FuzzyLogicCore» может применяться в различных учреждениях, в которых используются экспертные нечеткие системы, для построения лингвистических шкал, а также нечетких систем для решения задач классификации и аппроксимации.

### **Технологии разработки программного продукта**

Java является языком, реализующим объектно-ориентированную парадигму. Объектно-ориентированное программирование (ООП) – это парадигма программирования, которая основана на концепции объектов. Объекты представляют собой сущности реального мира или абстрактные конструкции, которые обладают определенными свойствами (атрибутами) и поведением (методами). В отличие от процедурного подхода, где акцент делается на последовательности выполнения операций, в ООП внимание сосредоточено на взаимодействии между объектами.

Перечислим три основных принципа ООП.

1. Инкапсуляция заключается в сокрытии внутренней реализации объекта и предоставлении доступа к его функциональности через четко определенные интерфейсы. Это позволяет отделить внутреннее состояние объекта от внешнего использования, что делает код более гибким и защищенным от случайных изменений. Например, если метод объекта изменяет свое поведение, то при правильной инкапсуляции изменения не затронут другие части программы, использующие этот объект.

2. Наследование – это механизм, позволяющий создавать новые классы на основе уже существующих с возможностью добавления новых свойств и методов либо переопределения старых. Этот принцип способствует повторному использованию кода и упрощает создание иерархий классов. Например, класс "Студент" может наследовать свойства класса "Человек", добавляя специфические атрибуты и методы, такие как номер студенческого билета или курс обучения.

3. Полиморфизм означает возможность объектов разных типов реагировать на одни и те же сообщения по-разному. Это достигается за счет перегрузки методов и виртуальных функций. Полиморфизм позволяет писать обобщенный код, который будет работать с различными типами данных без необходимости явной проверки их типа. Например, можно создать функцию, которая принимает любой объект, реализующий определенный интерфейс, и вызывает у него нужный метод независимо от конкретного типа этого объекта.

Таким образом, ООП представляет собой мощный инструмент для создания сложных систем, обеспечивая структурированность, модульность и удобство сопровождения программного обеспечения. Принципы инкапсуляции, наследования и полиморфизма являются ключевыми элементами этой парадигмы, каждый из которых играет важную роль в разработке качественных приложений. При правильном применении полиморфизма, инкапсуляция и наследование объединяются для создания программной среды, которая поддерживает разработку гораздо более надежных и масштабируемых программ, чем в случае использования модели, ориентированной на процессы.

Для проектирования гибкой архитектуры, удобной в поддержке, были использованы паттерны проектирования, а также обобщения. Паттерн проектирования – описание взаимодействия объектов и классов, адаптированных для решения общей задачи проектирования в конкретном контексте [9]. Паттерн проектирования именует, абстрагирует и идентифицирует ключевые аспекты структуры общего решения, которые и позволяют применить его для создания повторно используемого дизайна. В разработанном ПК используются такие паттерны, как Декоратор, Фасад и Строитель.

1. Паттерн Декоратор относится к группе структурных паттернов. Структурные паттерны определяют принципы объединения классов и объектов в более сложные структуры. Паттерны уровня класса применяют наследование для комбинирования интерфейсов и реализаций, тогда как паттерны уровня объекта предоставляют новую функциональность за счет композиции объектов. Такой подход обеспечивает большую гибкость, позволяя изменять состав объектов во

время выполнения, в отличие от статической композиции классов. При использовании паттерна Декоратор декорируемый объект вкладывается в другой объект, именуемый декоратором, который добавляет новую функциональность перед и/или после вызова методов декорируемого объекта. Паттерн проектирования Декоратор динамически наделяет объект новыми возможностями и является гибкой альтернативой субклассированию в области расширения функциональности. В ПК «FuzzyLogicCore» паттерн Декоратор используется для расширения функциональности механизма нечеткого вывода базы знаний механизмом интерполяции нечетких правил. При этом каждый алгоритм интерполяции нечетких правил инкапсулируется в отдельный класс-обертку над объектом класса базы знаний, а общий интерфейс базы знаний и классов, инкапсулирующих алгоритмы интерполяции, позволяет создавать вложенные структуры, делегирующие вызовы другим алгоритмам в том случае, если по какой-либо причине не был получен результат в результате вызова предыдущего алгоритма. Таким образом, получена гибкая архитектура, позволяющая добавлять новые алгоритмы интерполяции нечетких правил без необходимости изменения существующего кода базы знаний и алгоритмов интерполяции нечетких правил. Паттерн Декоратор имеет схожую структуру с паттерном Цепочка обязанностей. Оба паттерна базируются на принципе рекурсивного выполнения операции через серию связанных объектов. Но есть и несколько важных отличий. Обработчики в Цепочке обязанностей могут выполнять произвольные действия, независимые друг от друга, а также в любой момент прерывать дальнейшую передачу по цепочке. С другой стороны, Декораторы расширяют какое-то определённое действие, не ломая интерфейс базовой операции и не прерывая выполнение остальных декораторов. В данном случае выбор был сделан в пользу паттерна Декоратор ввиду возможного дальнейшего расширения функциональности обработки входных данных.

2. Паттерн Фасад также относится к группе структурных паттернов. Паттерн Фасад предоставляет унифицированный интерфейс, который скрывает сложность взаимодействия с набором интерфейсов внутри подсистемы. Он создает более высокий уровень абстракции, позволяя управлять сложностью и повышать

читаемость кода. Ключевая цель проектирования – минимизировать зависимости между подсистемами и сократить объем их взаимодействия. Один из эффективных способов достижения этой цели – внедрение паттерна Фасада, позволяющего уменьшить связанность компонентов и упрощая поддержку кода. В ПК «FuzzyLogicCore» паттерн Фасад используется для создания простого в использовании интерфейса получения описания кластеров на основе входных данных. Он инкапсулирует множество простых шагов, таких как кластеризация, извлечение точек, принадлежащих кластерам и построение эллипсоидов, охватывающих все точки кластеров, скрывая детали получения необходимой информации от клиента и избавляя от необходимости взаимодействовать с большим количеством объектов. Также в данном случае паттерн Фасад позволяет осуществить повторное использование кода, так как этап получения описания кластеров необходим для построения баз знаний всех трех типов.

3. Паттерн «Строитель» входит в группу порождающих паттернов, основное назначение которых – абстрагировать процесс создания объектов. Эти паттерны особенно полезны в системах, где предпочтение отдается композиции объектов, а не наследованию, что делает код более гибким и расширяемым. В отличие от жестко закодированных вариантов поведения, порождающие паттерны позволяют выделить небольшой набор базовых операций, комбинируя которые можно создавать более сложные структуры. Такой подход помогает снизить зависимость системы от конкретных способов создания, организации и представления объектов, что повышает её адаптируемость к изменениям. Паттерн «Строитель» решает задачу разделения процесса конструирования сложного объекта и его представления. Это означает, что один и тот же процесс создания может привести к разным результатам в зависимости от того, какое представление применяется. Такой подход удобен, когда требуется создавать различные вариации сложных объектов, используя единый алгоритм построения. Кроме того, данный шаблон позволяет создавать сложные объекты пошагово. В ПК «FuzzyLogicCore» паттерн Строитель применяется главным образом для избавления от конструктора класса,

реализующего паттерн Фасад, с большим количеством параметров, так как он позволяет заменить вызов конструктора цепочкой вызовов методов.

В своей основе термин *обобщения* означает параметризованные типы. Класс, интерфейс или метод, который оперирует на параметризованном типе, называется обобщенным [71]. Благодаря использованию обобщений можно создавать классы, интерфейсы и методы, которые будут безопасно работать с разнообразными типами данных. Многие алгоритмы логически одинаковы независимо от того, к какому типу данных они применяются. С помощью обобщений можно определить алгоритм однократно и независимо от конкретного типа данных, после чего применять этот алгоритм к широкому спектру типов данных без каких-либо дополнительных усилий. В ПК «FuzzyLogicCore» обобщения применяются для определения единого алгоритма обработки данных как для решения задачи классификации, так и аппроксимации. Таким образом осуществляется повторное использование кода. Для решения данной задачи определяются два класса, представляющих точки входных данных: `DataPointClassification` и `DataPointRegression`, содержащие соответственно значения независимых переменных с соответствующим классом и массив, содержащий значения независимых и зависимых переменных, и индексы зависимых переменных в нем. Оба класса реализуют единый интерфейс `DataPoint`, предоставляющий доступ к точке входных данных. Далее классы, реализующие одинаковые алгоритмы как для нечеткого классификатора, так и для баз правил лингвистической модели или Такаги-Сугено, параметризуются типом `DataPoint`, что позволяет клиенту решать, с какой задачей он будет работать, без необходимости дублирования кода для каждой задачи.

### **Описание логической структуры программного продукта**

На рис. 4.1 представлена обобщенная архитектура ПК.

На основные модули программного комплекса получены свидетельства о государственной регистрации программ для ЭВМ [57, 58].

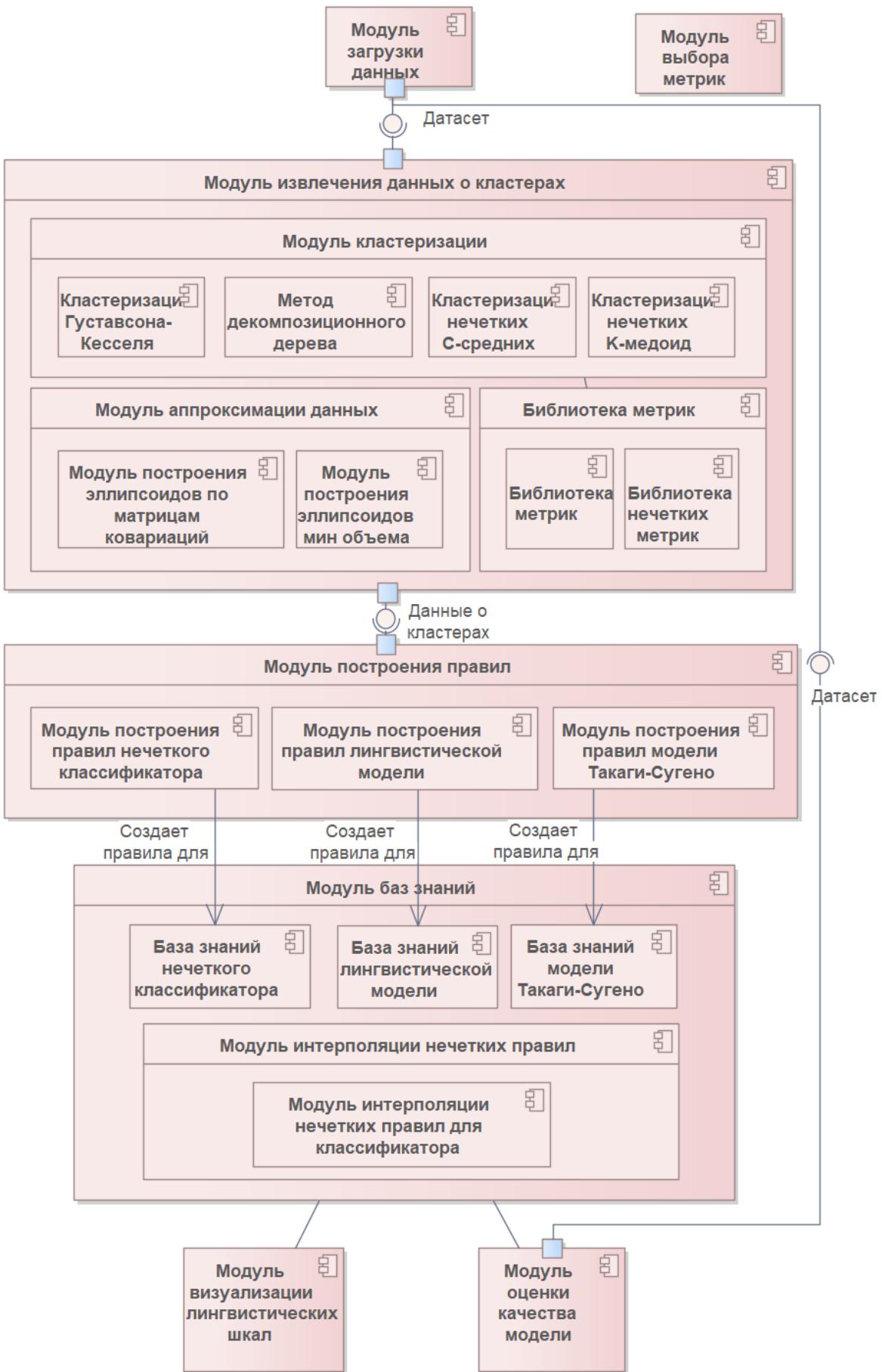


Рис. 4.1. Обобщенная архитектура ПК

На рис. 4.2 представлена UML-диаграмма классов модели данных, использующейся в реализации ПК «FuzzyLogicCore».

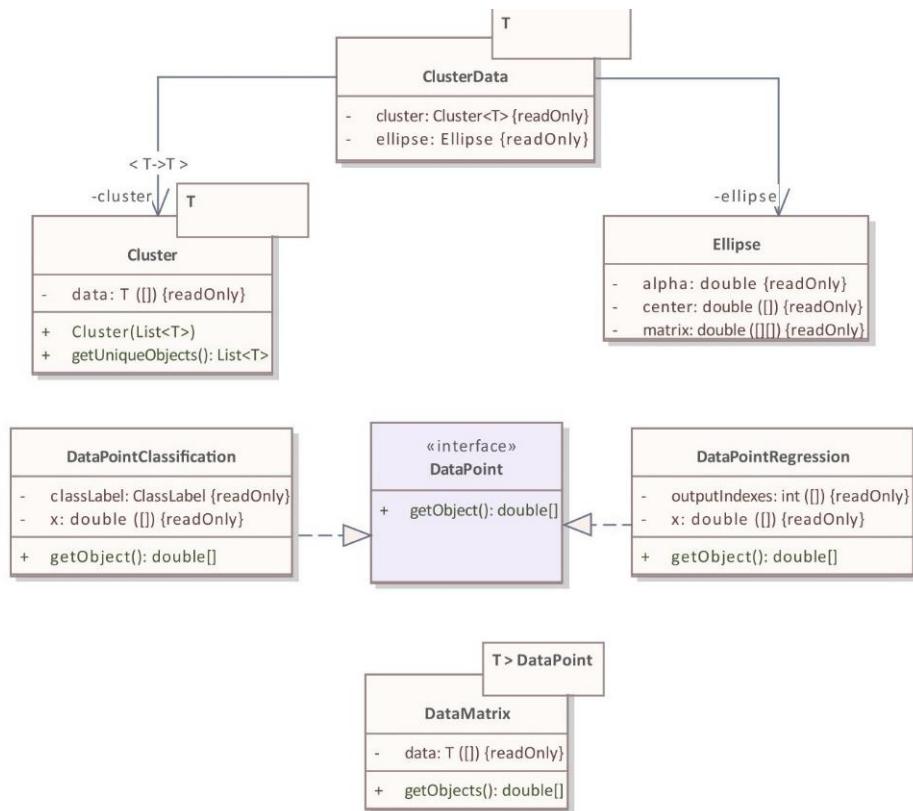


Рис. 4.2. Диаграмма классов модели данных

Классы `DataPointClassification` и `DataPointRegression` представляют собой единичные наблюдения и содержат в себе значения входных и выходных переменных. Оба класса реализуют общий интерфейс `DataPoint`, предоставляющий доступ к координатам точки данных. Класс `DataMatrix` представляет собой набор данных (обучающую или тестовую выборки) и содержит набор точек. Класс параметризуется типом, реализующим или расширяющим интерфейс `DataPoint`.

Входные данные, состоящие из набора точек, в ходе работы программы обрабатываются для разбиения на кластеры и аппроксимации кластеров эллипсоидами. Для представления кластера и эллипсоида используются классы `Cluster` и `Ellipse` соответственно. Класс `Cluster` содержит набор точек данных, принадлежащих данному кластеру, а класс `Ellipse` – значения, определяющие

эллипсоид в пространстве входных/выходных переменных. Класс ClusterData инкапсулирует всю необходимую информацию об одном кластере, имея атрибуты типов Cluster и Ellipse.

На рис. 4.3 представлена UML-диаграмма классов, реализующих непосредственно базу знаний нечеткой системы. Для примера представлена база знаний нечеткого классификатора, лингвистическая модель и Такаги-Сугено имеют аналогичную структуру классов.

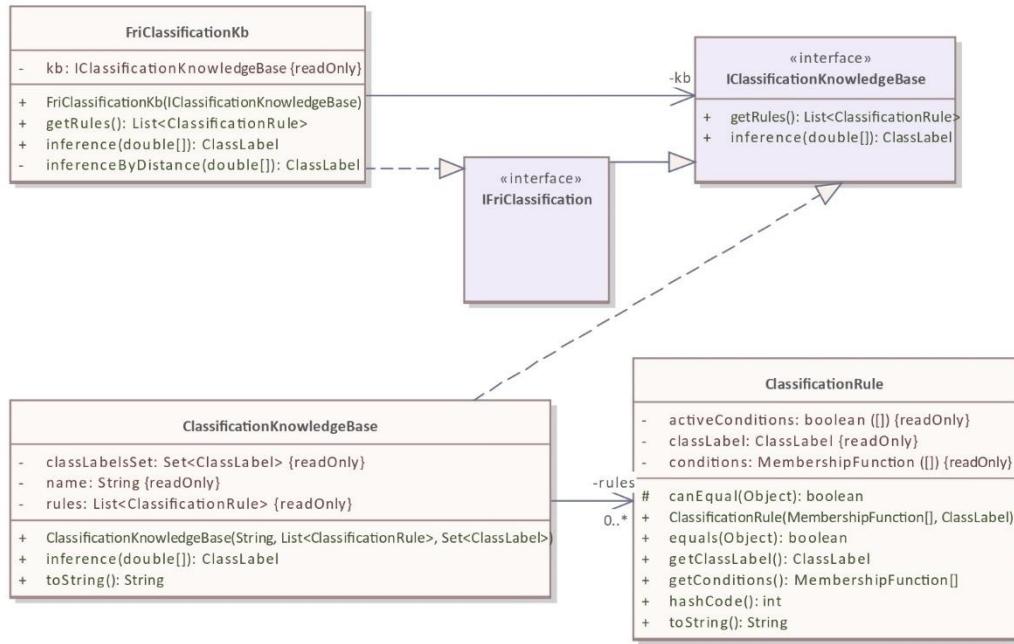


Рис. 4.3. Диаграмма классов нечеткого классификатора

Интерфейс **IClassificationKnowledgeBase** определяет метод **inference**, реализующий логический вывод. Класс **ClassificationKnowledgeBase** предоставляет конкретную реализацию данного интерфейса и содержит в себе набор правил – экземпляры типа **ClassificationRule**. Для реализации интерполяции нечетких правил в модель введен интерфейс-маркер **IFriClassification**, расширяющий интерфейс **IClassificationKnowledgeBase** для совместимости типов. Класс **FriClassificationKb** предоставляет одну из возможных реализаций данного интерфейса. Внутри себя данный класс содержит ссылку на экземпляр **IClassificationKnowledgeBase**, вызывая его методы для обеспечения базовой функциональности и добавляя свою

логику интерполяции нечетких правил. Данная совокупность классов спроектирована таким образом, чтобы обеспечить возможность добавления новых методов интерполяции нечетких правил в виде отдельных классов, не затрагивая код уже существующих методов и самой базы знаний, а также оставляя возможность для добавления дополнительной логики обработки входных данных.

В ПК «FuzzyLogicCore» реализована работа с двумя типами метрик – четкими и нечеткими. При реализации работы с метриками всталась задача создать библиотеку классов, поддерживающую простое расширение новыми метриками без изменения уже существующих классов, а также обеспечить достаточную гибкость в создании метрик, так как необходимо предусмотреть возможность использования различных сочетаний нечетких и четких метрик.

На рис. 4.4 представлена UML-диаграмма классов, реализующих библиотеку четких метрик. Все метрики реализуют общий интерфейс Distance, имеющий метод для вычисления расстояния distance.

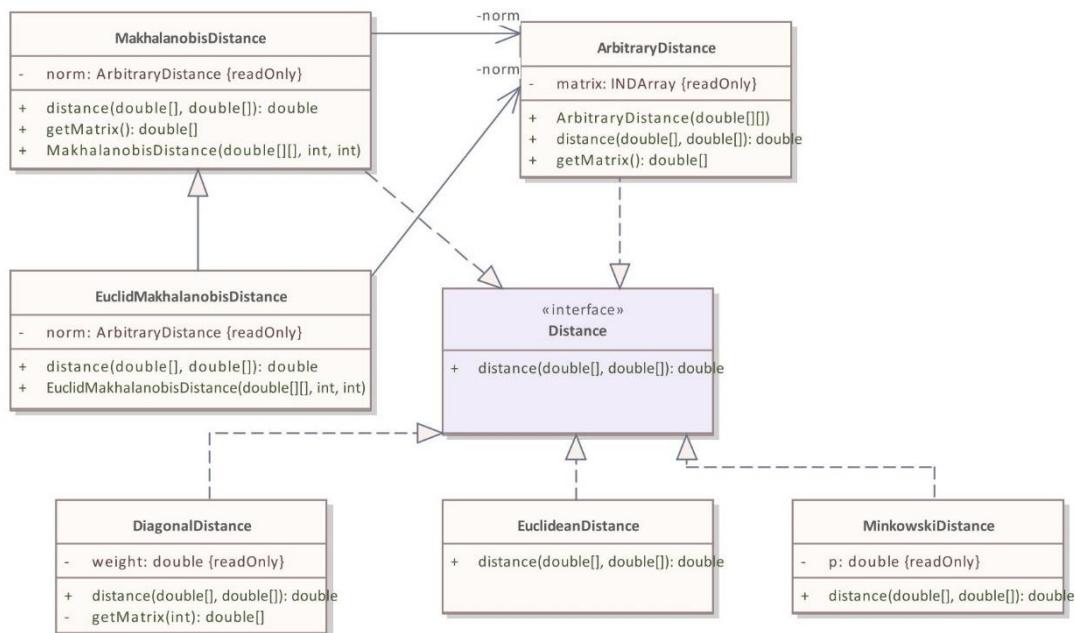


Рис. 4.4. Диаграмма классов библиотеки метрик

На рис. 4.5 представлена UML-диаграмма классов библиотеки нечетких метрик. Все метрики также реализуют интерфейс Distance, а также содержат в себе экземпляр класса NormalizedDistance, который осуществляет вычисление

нормализованного расстояния. Параметры нечетких метрик задаются неизменяемыми атрибутами классов, инициализируемыми в конструкторах классов. Таким образом, обеспечивается возможность выбора произвольной метрики при решении задачи кластеризации.

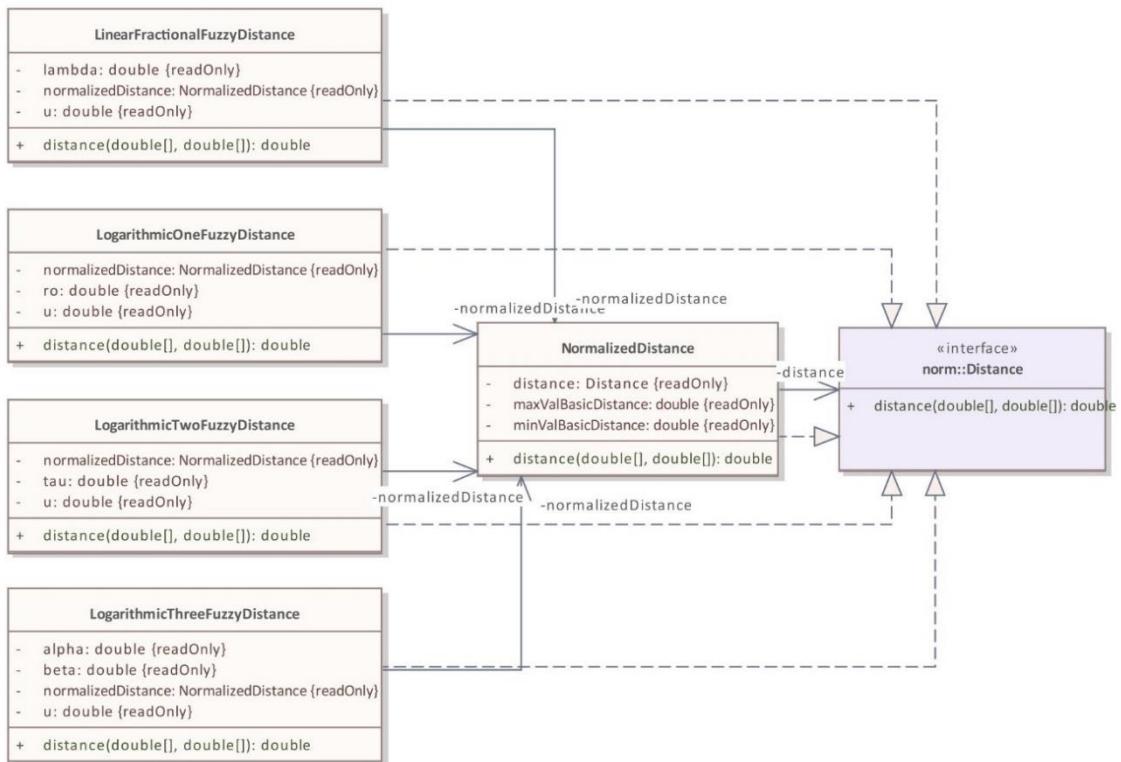


Рис. 4.5. Диаграмма классов библиотеки нечетких метрик

## Методы и средства разработки

Для разработки ПК были использованы следующие технологии и программное обеспечение:

1. Язык разработки – Java.
2. Язык разработки для визуализации – Python.
3. Фреймворк для серверной разработки – SpringBoot.
4. СУБД PostgreSQL.
5. Среда разработки для разработки на Java – IntelliJ IDEA.
6. Среда разработки для разработки на Python – PyCharm.

Выбор языка программирования Java обусловлен следующими преимуществами:

1. Реализация объектно-ориентированной парадигмы. Объектно-ориентированный подход к программированию предоставляет множество инструментов и методов для создания надежных, масштабируемых и удобных в поддержке систем. Одним из ключевых аспектов ООП является использование паттернов проектирования, которые представляют собой проверенные временем решения типичных проблем программирования. Паттерны позволяют стандартизировать подходы к решению задач, делая код более читаемым и поддерживаемым. Использование паттернов проектирования позволяет разработчикам применять лучшие практики и стандарты, повышая качество кода и ускоряя процесс разработки. Высокая производительность. Байт-код Java был тщательно спроектирован, чтобы легко транслироваться прямо в машинный код для достижения очень высокой производительности с использованием оперативного компилятора.

3. Надежность. Для обеспечения надежности Java ограничивает пользователя в ряде ключевых областей, чтобы поиск ошибок осуществлялся на ранней стадии проектирования программ. В то же время Java избавляет пользователя от необходимости беспокоиться о многих наиболее распространенных причинах ошибок при программировании. Основные механизмы достижения этого – это то, что Java является строго типизированным языком, самостоятельно управляет выделением и освобождением памяти, а также предлагает объектно-ориентированную обработку исключений.

4. Простота. Язык Java был спроектирован так, чтобы быть легким в изучении и эффективным в использовании профессиональным программистом.

5. Кроссплатформенность. Кроссплатформенность — это характеристика программного обеспечения, позволяющая ему функционировать на различных аппаратных платформах и операционных системах без необходимости значительной модификации исходного кода. Кроссплатформенные приложения способны выполняться на широком спектре устройств и сред, включая персональные компьютеры, серверы, мобильные устройства и встраиваемые системы. Язык программирования Java реализует кроссплатформенность

посредством использования модели компиляции и исполнения, основанной на виртуальной машине. При компиляции исходного кода Java создается байт-код, который является платформонезависимым представлением программы. Этот байт-код затем исполняется виртуальной машиной Java (JVM), которая абстрагирует особенности конкретной аппаратной платформы и операционной системы.

Выбор языка программирования Python для визуализации обоснован тем, что, помимо простоты и реализации объектно-ориентированной парадигмы он имеет богатую экосистему, в том числе библиотеки, предназначенные для визуализации, что позволяет быстро и качественно визуализировать лингвистические шкалы.

Выбор интегрированных сред разработки IntelliJ IDEA и PyCharm обусловлен тем, что они предоставляют разработчикам удобные функции, упрощая и ускоряя разработку программ. Данные среды поддерживают новейшие возможности языков. В состав сред входит интеллектуальный редактор кода, который обнаруживает ошибки по мере ввода, предлагает варианты автодополнения с учетом контекста, выполняет безопасный рефакторинг и предоставляет много других полезных функций.

### **Используемые технические средства**

Программный комплекс организован в виде веб-сервиса с возможностью его дальнейшего развертывания в облачной инфраструктуре, при этом возможна конфигурация размещения базы данных и самого приложения на разных узлах, а также горизонтальное масштабирование приложения в случае роста нагрузки.

Рекомендуемые системные требования к узлу с приложением:

- операционная система Windows или Linux с установленной виртуальной машиной Java минимум 17 версии;
- оперативная память для работы приложения – 10 Гб (из расчета 30 одновременно активных пользователей), при возрастании нагрузки и/или объема загружаемых данных требуется масштабирование;
- память на жестком диске для запуска и работы приложения – 5 Гб.

Рекомендуемые системные требования к узлу с базой данных:

- оперативная память – 8 Гб;
- память на жестком диске – 30 Гб.

Приложение представляет собой fatJar, запуск программы осуществляется с помощью виртуальной машины Java, параметры приложения указываются в переменных среды.

Объем программы (fatJar без базы данных) составляет ~200 Мб.

### **Интерфейс пользователя**

Для реализации внешнего интерфейса ПК «FuzzyLogicCore» был разработан REST API, предоставляющий доступ как к глобальному процессу генерации базы правил на основе исходных данных, так и возможность реализации отдельных шагов данного процесса, например кластеризации исходных данных с помощью выбранного метода.

REST API (Representational State Transfer Application Programming Interface) – это архитектурный стиль для построения веб-сервисов, который использует HTTP-методы (GET, POST, PUT, DELETE) для взаимодействия с ресурсами. Он основывается на принципах stateless (отсутствие сохранения состояния между запросами), client-server (разделение ответственности между клиентом и сервером) и uniform interface (единообразный интерфейс). REST API обычно представляет ресурсы в виде JSON, позволяя клиентам получать, создавать, обновлять и удалять данные, не завися от реализации сервера. Этот подход обеспечивает гибкость, масштабируемость и простоту интеграции между различными системами, делая REST API популярным выбором для современных веб-приложений и микросервисов.

В интерфейсе данного ПК разделены процессы создания базы правил и базы знаний, что позволяет создать базу знаний не только на основе данных, но и на основе внешнего описания базы правил. Так как процесс формирования базы правил нечеткой системы может выполняться длительное время (в зависимости от использующихся алгоритмов), было принято решение реализовать его в виде

асинхронного метода. На рис. 4.6 представлен метод создания процесса генерации базы правил.

**POST /rbFromDataCreationProcess** Create a new fuzzy system rule base from provided training set

Create a new fuzzy system rule base from provided training set using specified parameters.

**Parameters**

No parameters

**Request body** required

multipart/form-data

model  
string  
clusteringMethod  
string  
ellipsoidBuilder  
string  
metricType  
string  
metricParameters  
object  
trainDataSet  
string(\$binary)

**Responses**

Code	Description	Links
201	Rb creation process started	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "model": "Classifier",
  "status": "Started",
  "errorMessage": "string"
}
```

Рис. 4.6. Метод создания процесса генерации базы правил

Данный метод принимает на вход параметры генерации и обучающую выборку в виде текстового файла, а возвращает созданный объект процесса генерации нечеткой системы. Данный метод может применяться для генерации лингвистической модели, модели Такаги-Сугено и нечеткого классификатора. На рис. 4.7 представлена визуализация спецификации объекта параметров, определяющих особенности формирования базы правил.

```

CreateRbFromDataRequest < {
    model
        < string
    Enum:
        < [ classifier, Mamdani, TS ]
    clusteringMethod
        < string
    Enum:
        < [ GK, DT, FCM, FKM ]
    ellipsoidBuilder
        < string
    Enum:
        < [ cov, minVol, both ]
    metricType
        < string
    Enum:
        < [ crisp, fuzzy ]
    metricParameters
        < {
            oneOf ->
                FuzzyMetric > ...
                CrispMetric > ...
        }
    trainDataSet
        < string($binary)
}

```

Рис. 4.7. Визуализация спецификации объекта параметров запроса создания базы правил

Данный объект параметров содержит включает в себя свойство типа FuzzyMetric или CrispMetric, визуализация спецификации которых представлена на рис. 4.8. Данное свойство содержит параметры соответственно нечеткой метрики или обычной метрики, использующейся для проведения процедуры кластеризации.

На рис. 4.9 представлены остальные методы, обеспечивающие интерфейс создания базы правил на основе данных.

Метод GET /rbFromDataCreationProcess/{id}/load загружает сгенерированную базу правил по идентификатору созданного процесса генерации базы правил в виде текстового файла. Текстовый файл содержит описание базы правил в формате JSON.

Формат JSON (JavaScript Object Notation) представляет собой текстовый формат обмена данными, основанный на синтаксисе JavaScript. Он отличается простотой структуры и высокой степенью читаемости как для человека, так и для машинных систем.

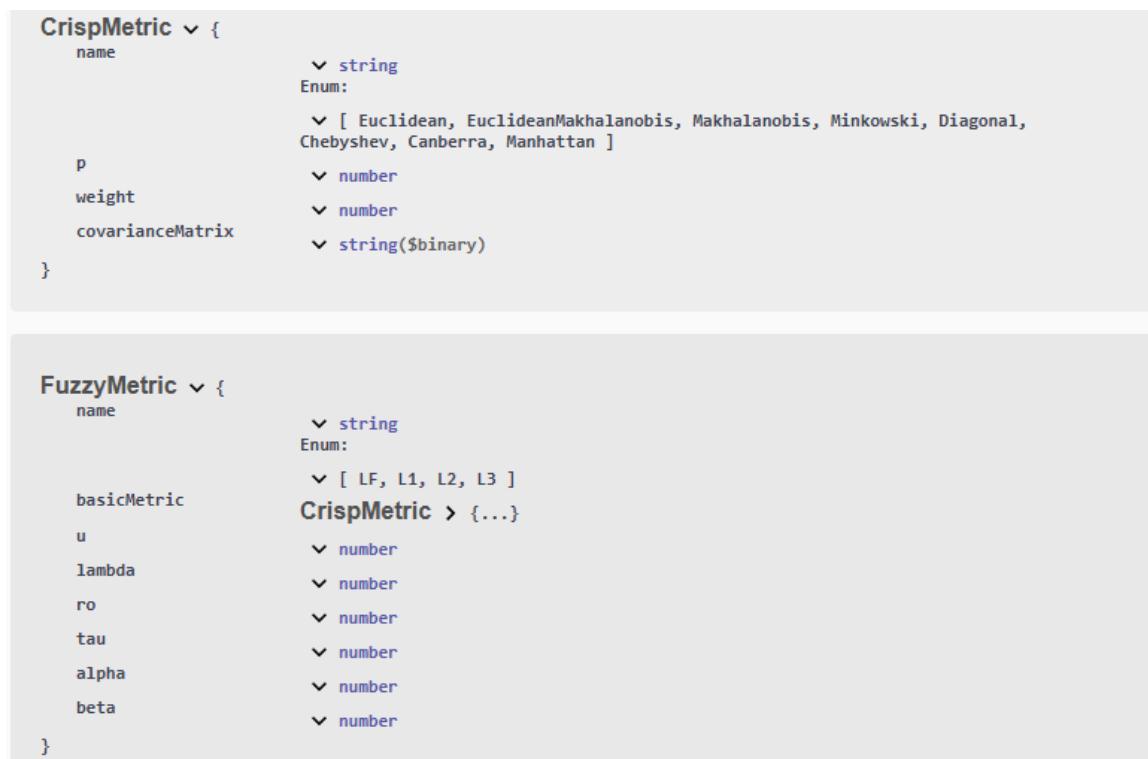


Рис. 4.8. Визуализация спецификации типов CrispMetric и FuzzyMetric

### Rule base generation from data

- GET** /rbFromDataCreationProcess Get a list of rule base creation processes
- POST** /rbFromDataCreationProcess Create a new fuzzy system rule base from provided training set
- GET** /rbFromDataCreationProcess/{id} Get rule base creation process by ID
- POST** /rbFromDataCreationProcess/{id}/stop Stop rule base creation process
- GET** /rbFromDataCreationProcess/{id}/load Load generated rule base

Рис. 4.9. Методы REST API для создания базы правил

В отличие от других форматов, таких как XML, JSON характеризуется более компактным представлением данных, что снижает объем передаваемой информации и ускоряет процессы обработки. Это делает его особенно полезным при разработке веб-приложений и создании API-интерфейсов, где требуется эффективный обмен данными между клиентской и серверной частями системы. Широкая поддержка JSON в различных языках программирования и библиотеках

способствует его популярности среди разработчиков, обеспечивая легкость интеграции и использования данного формата в разнообразных проектах.

На рис. 4.10 приведена визуализация спецификации объектов, с помощью которых описывается база правил нечеткого классификатора.

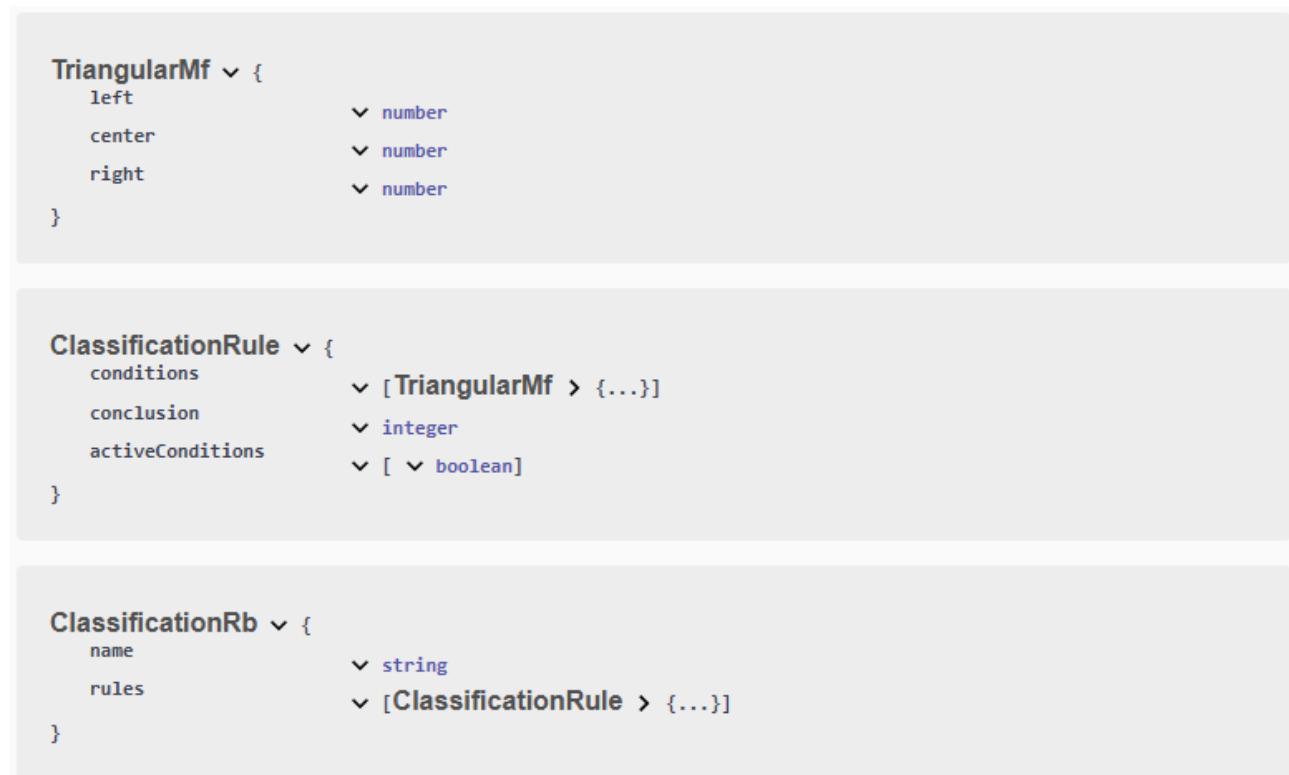


Рис. 4.10. Визуализация спецификации объектов базы правил нечеткого классификатора

Рассмотренная группа методов осуществляет генерацию базы правил на основе обучающих данных. Работа с базами знаний осуществляется группой методов, представленной на рис. 4.11.

Экземпляр базы знаний, готовый к осуществлению логического вывода, создается методом POST /kb. Данный метод принимает на вход текстовый файл, содержащий определение базы правил, и желаемые методы интерполяции нечетких правил и создает экземпляр базы знаний, доступный по сгенерированному идентификатору.

Knowledge base	
GET	/kb Get a list with all knowledge bases
POST	/kb Create a new knowledge base instance from description
GET	/kb/{id} Get a knowledge base instance by id
POST	/kb/{id}/inference Perform fuzzy inference on a provided data
POST	/kb/{id}/test Perform test on a trained model

Рис. 4.11. Методы REST API для работы с базами знаний

Непосредственно нечеткий логический вывод осуществляется вызовом метода POST /kb/{id}/inference. Данный метод принимает на вход текстовый файл с набором векторов, по которым необходимо простроить прогноз (без значений зависимой переменной), и возвращает текстовый файл с полученными в результате логического вывода значениями для каждого входного вектора.

Также реализован метод POST /kb/{id}/test, который осуществляет тестирование модели. На вход он принимает текстовый файл с тестовой выборкой, а возвращает показатели качества модели. Данный метод позволяет оценить качество бинарной классификации или аппроксимации. На рис. 4.12 представлена визуализация спецификации объектов, содержащих показатели качества бинарной классификации и аппроксимации соответственно.

Таким образом, внешний интерфейс системы для генерации базы знаний представлен в виде REST API, что обеспечивает гибкость и удобство взаимодействия с системой через стандартные HTTP-запросы. Данный подход позволяет пользователям выбирать наиболее подходящий тип базы знаний — будь то классификатор, лингвистическая модель или модель Такаги-Сугено — и получать доступ к соответствующим методам и ресурсам для их создания и управления. RESTful архитектура интерфейса делает систему доступной для широкого круга пользователей и интегрируемой с разнообразными внешними

приложениями, способствуя повышению эффективности работы с базой знаний и расширению возможностей её применения.

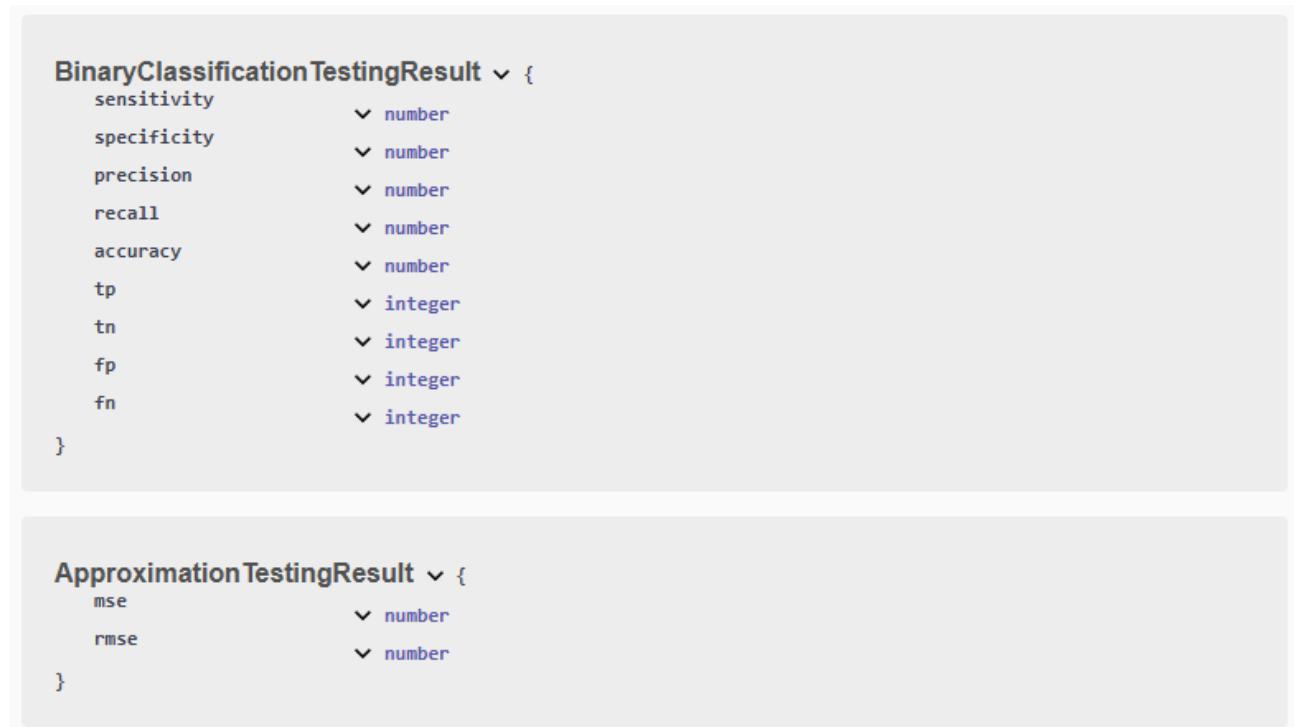


Рис. 4.12. Визуализация спецификации объектов качества модели

## 4.2. Применение программного комплекса «FuzzyLogicCore» для анализа электроэнцефалограмм

Интерфейсы «мозг-компьютер» широко используются в медицинских целях, например, для восстановления двигательных функций у людей с нарушениями движений или параличом. Они также могут быть применены в игровой индустрии, виртуальной реальности и других областях, где управление устройствами с помощью мыслей может быть полезным. Неинвазивные интерфейсы мозг-компьютер позволяют людям с ограниченными возможностями взаимодействовать с миром весьма эффективным способом, открывая перед ними новые возможности и перспективы и обладая большей безопасностью и доступностью по сравнению с инвазивными интерфейсами. Одними из возможных реализаций неинвазивного интерфейса являются подходы, основанные на анализе электроэнцефалограмм (ЭЭГ).

Электроэнцефалография – метод исследования головного мозга, основанный на регистрации его электрических потенциалов. Являясь результатом регистрации сложного процесса, порожденного взаимодействием большого числа микросистем нейронов, нелинейно взаимодействующих друг с другом, электроэнцефалограмма представляет собой запись электрической активности нейронов различных структур головного мозга и несет в себе много полезной информации, отражающей состояние человека. Однако, зачастую полезный сигнал ЭЭГ зашумлен значительным количеством электрических явлений, связанных с фоновой активностью мозга. Одной из актуальных является задача извлечения из ЭЭГ сигналов, связанных с реальным или воображаемым движением человека [10, 17, 24, 50]. Моторные образы [63] – это паттерны мозговой активности, зарегистрированные, например, с помощью ЭЭГ, и отражающие формирование мысленных команд на движение, как реальное, так и воображаемое. Сигналы ЭЭГ во время представления моторных образов могут использоваться в качестве управляющих сигналов интерфейсов «мозг-компьютер» (ИМК). Для распознавания моторных образов, как правило, решается задача классификации ЭЭГ, выполняемая обычно после обучения системы с учетом положения датчиков и индивидуальных особенностей оператора. В [44, 45] для классификации воображаемых движений используется байесовский классификатор, выделение признаков основано на модуляции сенсорно-моторного ритма, или мю-ритма. В [6] используются сверточные нейронные сети, а признаками являются изображения, сгенерированные на основе ЭЭГ. В [55] проводилось исследование различных архитектур нейронных сетей для классификации движений, отмечено, что точность при использовании нейронных сетей для данной задачи составляет около 80%. Также в [79] приводится обзор методов глубокого обучения для решения задачи классификации ЭЭГ. Кроме того, для анализа ЭЭГ применяются классические методы, основанные на частотно-временном, волновом или компонентном анализе [112, 121, 131, 180]. Недостатком данных подходов является то, что их применение зачастую не дает устойчивых результатов ввиду сложности обрабатываемых данных ЭЭГ, высокой вариабельности выборки испытуемых и непосредственно

обучения мозга, который адаптируется к выдаче команд. Для повышения качества методов распознавания моторных образов могут использоваться специальные подходы, например, метод накопления проб путем суммирования нескольких образцов сигналов одного типа движений [59]. Также выбор подходящего способа извлечения признаков играет большую роль в задаче классификации ЭЭГ. Часто для выделения признаков используется общая пространственная модель (CSP) [102, 150, 164]. В статье [103] предлагается использовать комбинацию анализа независимых компонент, вейвлет-преобразования и общей пространственной модели для выделения признаков. Таким образом, возникает необходимость в разработке и апробации новых методов распознавания моторных образов с оценкой возможности обобщения полученных методов на более широкий класс ЭЭГ-феноменов. В данной работе предлагается использовать подход, основанный на расчете взаимной информации между каналами ЭЭГ, и приводятся результаты исследований. В качестве классификатора используется нечеткая система.

Регистрируемый сигнал ЭЭГ представляет собой разность потенциалов между двумя точками на коже головы. На каждый канал регистрации подаются напряжения, отведенные двумя электродами: одно на положительный, другое на отрицательный вход канала усиления. На рис. 4.13 показана схема одного из возможных расположений электродов на поверхности головы человека (по международной системе 10-20, рекомендованной Международной федерацией электроэнцефалографии и клинической нейрофизиологии).

Данные ЭЭГ, как правило, представлены в виде многомерного временного ряда с частотой дискретизации от 100 Гц до 1 кГц, значения которого соответствуют электрическим потенциалам, измеренным на поверхности головы испытуемого (в мкВ).

При воображении движения испытуемым данное явление отражается в сигнале ЭЭГ, при этом необходимо, во-первых, применить комплекс методов, направленных на извлечение полезной информации из зашумленного сигнала, а во-вторых, учесть индивидуальные особенности человека.

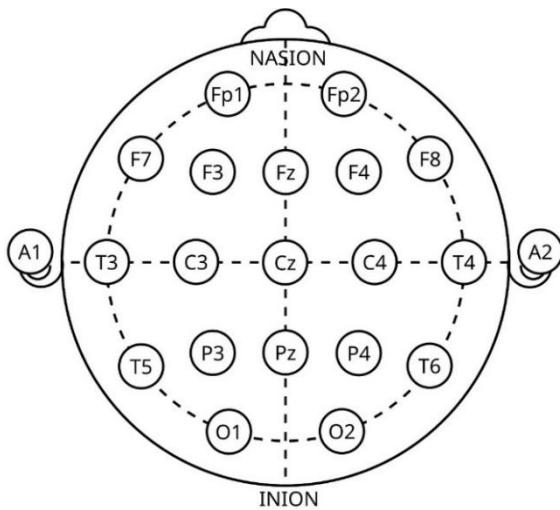


Рис. 4.13. Расположение электродов на поверхности головы человека по международной системе 10-20

Для решения первой задачи используются разнообразные методы предобработки данных, а для решения второй ставится задача обучения с учителем. Обучающие данные представляют собой записи ЭЭГ (многомерные временные ряды), где каждой записи поставлен в соответствие класс – какое движение производилось оператором во время записи ЭЭГ. После обучения классификатора производится тестирование на тестовых данных, где обученный классификатор должен по предъявляемой записи ЭЭГ определить класс – какое движение выполнялось во время записи данной ЭЭГ. В данном исследовании осуществляется распознавание двух классов – отсутствие движения и движение, которое соответствует подъему обеих рук.

Задача распознавания моторных образов является задачей классификации сигналов ЭЭГ, формальная постановка которой имеет следующий вид: пусть задано множество объектов  $X \subseteq M^m$  и множество классов  $Y = \{y_1, y_2\}$ , при этом существует неизвестная целевая зависимость – отображение  $y: X \rightarrow Y$ , чьи значения известны лишь на объектах обучающей выборки  $\{(x_1, y_1), \dots, (x_n, y_n)\}, x \in X, y \in Y$ . Необходимо получить алгоритм, способный классифицировать произвольный объект  $x \in X$ .

В [40] предложен алгоритм для решения данной задачи.

Исходные записи сигналов ЭЭГ представляют собой двумерные матрицы,

столбцы которых соответствуют электродам (так называемым отведениям), а строки – измеренному значению амплитуды сигнала (в мкВ) в определенный момент времени. На рис. 4.14 изображен пример записи ЭЭГ одного канала длительностью 1 с. Для решения задачи классификации необходимо преобразовать каждую запись ЭЭГ, представляющую собой многомерный временной ряд, в вектор признаков, удалив шум и выделив полезную информацию. В частности, сигналы были разбиты на перекрывающиеся окна длительностью 0.5 с. После этого каждый канал был подвергнут фильтрации в широком диапазоне частот от 5.5 до 35 Гц для удаления шума. Далее проводилась процедура децимации и взятие первой производной от амплитуды. Децимация была необходима с целью снижения ресурсоемкости процедуры, критичной для ИМК, работающих в реальном времени.

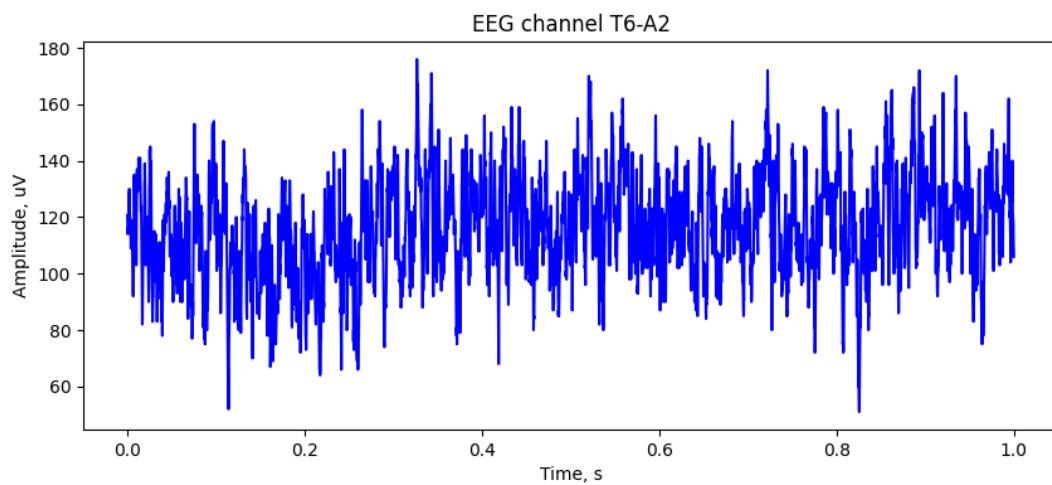


Рис. 4.14. Пример исходной записи ЭЭГ

Одним из основных результатов предложенного подхода является процедура для снижения размерности и построения признакового пространства, которая базируется на вычислении взаимной информации для пар отведений. На ее основе формируется вектор признаков, каждая компонента которого есть значение взаимной информации между конкретной парой отведений. Согласно [19], взаимная информация – функция двух случайных величин, описывающая количество информации, содержащееся в одной случайной величине относительно другой, которая рассчитывается по следующей формуле:

$$I(X, Y) = \iint_{X \times Y} p(x, y) \log_2 \left( \frac{p(x, y)}{p(x)p(y)} \right) dx dy,$$

где  $p(x, y)$  — функция совместной плотности вероятностей,  $p(x), p(y)$  — одномерные плотности вероятностей. При вычислении взаимной информации значения, полученные после препроцессинга амплитуд пары сигналов, используются в качестве независимых реализаций пары зависимых непрерывных случайных величин.

В рамках исследования вектор признаков формируется для каждого испытуемого.

Для решения задачи классификации, которая лежит в основе распознавания реального или воображаемого действия испытуемого, был разработан нечеткий классификатор в форме нечеткой системы производственного типа, ядром которой является база знаний. Данный выбор обусловлен тем, что при качественной базе знаний нечеткая система обладает хорошими аппроксимационными свойствами, а также высоким уровнем интерпретируемости, а, следовательно, объяснительными способностями. Кроме того, после построения базы знаний появляется возможность составить приближенную оценку того, какой вклад вносит каждая пара отведений в формирование ответа классификатора, что предоставляет дополнительную информацию об активности головного мозга каждого испытуемого. Фактически, результаты работы классификатора позволяют уточнить расположение моторных зон головного мозга в аспекте проекции их активности на датчики ЭЭГ.

Цель эксперимента — тестирование предложенного метода предобработки данных ЭЭГ и нечеткого классификатора для задачи распознавания движений, как реальных, так и воображаемых, на основе ЭЭГ.

Для тестирования была создана группа из 30 человек. Осуществлялась запись ЭЭГ с частотой дискретизации 5000 Гц нетренированных испытуемых в двух чередующихся произвольным образом состояниях: состоянии покоя, а также при реальном и воображаемом поднятии обеих рук. Данные были поделены на тестовую и обучающую выборки, объем тестовой выборки составлял 20%.

Входными данными являются записи ЭЭГ: таблицы данных с 29 столбцами, соответствующими отведениям ЭЭГ и каналам со служебной информацией, каждый столбец с данными отведений содержит значения амплитуды сигнала в мкВ. Количество записей в каждом столбце одинаково для одного файла, но от файла к файлу может отличаться, так как велась запись сигналов разной длительности.

В рамках эксперимента устанавливалось влияние на результат распознавания следующих факторов: значение частоты децимации при предобработке, количество бинов при вычислении показателей взаимной информации каналов, были или нет нормализованы данные при предварительной обработке.

Выходными данными нечеткого классификатора является метка класса для каждой записи ЭЭГ (0 соответствует состоянию покоя, 1 – подъему обеих рук).

Для оценки качества классификации использовались такие метрики, как *доля правильных ответов, чувствительность и специфичность*. Доля правильных ответов при отнесении всех объектов тестовой выборки к одному классу, представленному большинством, может дать высокую оценку, несмотря на то что результаты классификации будут некорректны. Исходя из этого, в дополнение к ней выбраны метрики чувствительности и специфичности, показывающие доли результатов по классам. Выбранные метрики рассчитываются по следующим формулам:

$$\text{Доля правильных ответов} = \frac{TP + TN}{TP + TN + FP + FN},$$

$$\text{Чувствительность} = \frac{TP}{TP + FN},$$

$$\text{Специфичность} = \frac{TN}{TN + FP},$$

где  $TP$  – количество истинно положительных результатов,  $TN$  – количество истинно отрицательных результатов,  $FP$  – количество ложно положительных результатов,  $FN$  – количество ложно отрицательных результатов.

Для выполнения эксперимента использовался ПК «FuzzyLogicCore», а также была разработана компонента предобработки данных на языке Python.

Статистический анализ результатов эксперимента осуществлялся пакетом STATISTICA. Компонента предобработки данных осуществляет формирование векторов признаков, а также разбиение данных на обучающую и тестовую выборки. Программа состоит из нескольких модулей и написана на языке Python. Входными данными программы служат текстовые файлы с записями ЭЭГ в формате, описанном ранее, а также соответствующая каждому файлу метка класса, а выходными – текстовые файлы с векторами-признаками и меткой класса, при этом обучающие и тестовые данные содержатся в отдельных файлах.

В табл. 4.1 приведены средние значения показателей качества классификации для реальных и мысленных движений (в таблице приведены средние значения, вычисленные с параметрами, которым соответствуют лучшие значения среднего гармонического чувствительности и специфичности).

Таблица 4.1. Результаты классификации ЭЭГ, средние значения для 30 человек

Показатель качества	Реальные движения	Мысленные движения
Доля правильных ответов, %	74.15	60.06
Чувствительность, %	60.26	68.41
Специфичность, %	72.81	39.56

Необходимо заметить, что для части испытуемых показатель чувствительности равен 1 при показателе специфичности, равном 0, или наоборот. Данная ситуация свидетельствует о том, что построенная модель выдает на каждом объекте константное предсказание и, таким образом, не может считаться удовлетворительной, несмотря на полученный в некоторых случаях показатель доли правильных ответов, больший 50%. Тем не менее, для 72% испытуемых в случае реальных движений и 48% в случае воображаемых предложенный результат отличается от бинарного классификатора, выдающего константные значения.

Рассмотрим более подробно, как работает нечеткий классификатор для отдельного человека. На рис. 4.15 приведен пример визуализации с помощью метода tSNE точек, представляющих векторы признаков, для одного человека.

Разными цветами отмечены точки разных классов. Как можно видеть, выбранный способ извлечения признаков позволяет получить точки, группирующиеся в кластеры по классам.

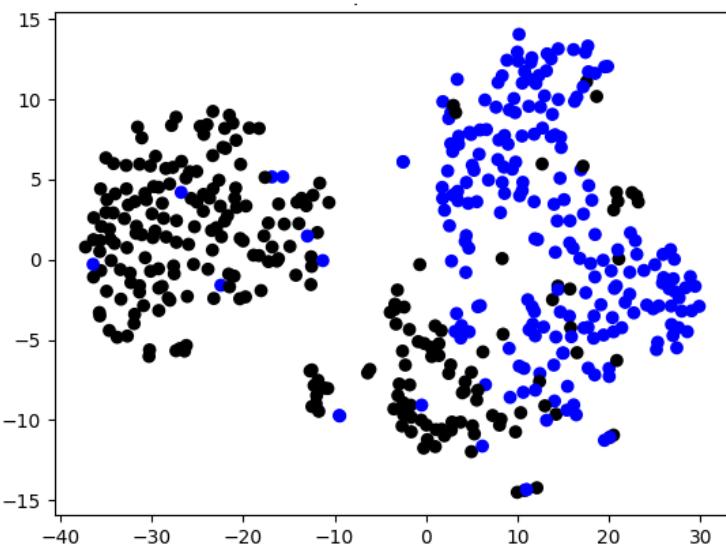


Рис. 4.15. Визуализация с помощью tSNE исходных данных

На рис. 4.16 представлены примеры построенных лингвистических шкал для некоторых переменных.

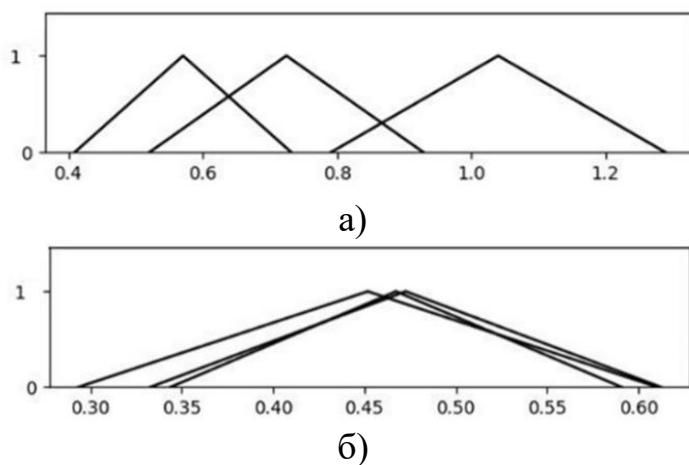


Рис. 4.16. Лингвистические шкалы

На рис. 4.16-а изображена корректная лингвистическая шкала, на рис. 4.16-б термы лингвистической шкалы совпадают друг с другом, в результате чего можно сделать вывод, что данный признак может быть удален из модели.

Приведем в качестве примера базу знаний, полученную для распознавания мысленных движений одного человека. В данном случае лингвистические шкалы каждой переменной содержат три терма:  $T = \{M, C, B\}$  (малый, средний, большой).

Функции принадлежности термов в форме треугольных нечетких чисел имеют вид, представленный в табл. 4.2.

Таблица 4.2. Термы лингвистических переменных

Название переменной	Функции принадлежности		
Взаимная информация электродов F3-F7	$M = \begin{pmatrix} 0.44636 \\ 0.69464 \\ 0.94293 \end{pmatrix}$	$C = \begin{pmatrix} 0.64790 \\ 0.88828 \\ 1.12865 \end{pmatrix}$	$B = \begin{pmatrix} 0.67831 \\ 0.89911 \\ 1.11992 \end{pmatrix}$
Взаимная информация электродов F3-T3	$M = \begin{pmatrix} 0.42819 \\ 0.64292 \\ 0.85765 \end{pmatrix}$	$C = \begin{pmatrix} 0.57762 \\ 0.79144 \\ 1.00526 \end{pmatrix}$	$B = \begin{pmatrix} 0.62860 \\ 0.88307 \\ 1.13753 \end{pmatrix}$
Взаимная информация электродов F3-F4	$M = \begin{pmatrix} 0.41309 \\ 0.66704 \\ 0.92100 \end{pmatrix}$	$C = \begin{pmatrix} 0.53553 \\ 0.80327 \\ 1.07101 \end{pmatrix}$	$B = \begin{pmatrix} 0.67899 \\ 0.90552 \\ 1.13205 \end{pmatrix}$
Взаимная информация электродов F3-F8	$M = \begin{pmatrix} 0.39682 \\ 0.64675 \\ 0.89668 \end{pmatrix}$	$C = \begin{pmatrix} 0.55142 \\ 0.76747 \\ 0.98351 \end{pmatrix}$	$B = \begin{pmatrix} 0.61546 \\ 0.83563 \\ 1.05579 \end{pmatrix}$
Взаимная информация электродов F3-T4	$M = \begin{pmatrix} 0.42152 \\ 0.66686 \\ 0.91221 \end{pmatrix}$	$C = \begin{pmatrix} 0.61787 \\ 0.79950 \\ 0.98113 \end{pmatrix}$	$B = \begin{pmatrix} 0.66394 \\ 0.89831 \\ 1.13267 \end{pmatrix}$
Взаимная информация электродов F7-T3	$M = \begin{pmatrix} 0.53483 \\ 0.77044 \\ 1.00604 \end{pmatrix}$	$C = \begin{pmatrix} 0.62963 \\ 0.87129 \\ 1.11295 \end{pmatrix}$	$B = \begin{pmatrix} 0.71679 \\ 0.94948 \\ 1.18218 \end{pmatrix}$
Взаимная информация электродов F7-F4	$M = \begin{pmatrix} 0.52713 \\ 0.78384 \\ 1.04056 \end{pmatrix}$	$C = \begin{pmatrix} 0.58690 \\ 0.84057 \\ 1.09424 \end{pmatrix}$	$B = \begin{pmatrix} 0.67748 \\ 0.92027 \\ 1.16306 \end{pmatrix}$
Взаимная информация электродов F7-F8	$M = \begin{pmatrix} 0.49135 \\ 0.71641 \\ 0.94147 \end{pmatrix}$	$C = \begin{pmatrix} 0.56189 \\ 0.78359 \\ 1.00529 \end{pmatrix}$	$B = \begin{pmatrix} 0.62177 \\ 0.84964 \\ 1.07751 \end{pmatrix}$

Продолжение таблицы 4.2			
Взаимная информация электродов F7-T4	$M = \begin{pmatrix} 0.55906 \\ 0.74971 \\ 0.94035 \end{pmatrix}$	$C = \begin{pmatrix} 0.65505 \\ 0.84101 \\ 1.02697 \end{pmatrix}$	$B = \begin{pmatrix} 0.68451 \\ 0.91415 \\ 1.14379 \end{pmatrix}$
Взаимная информация электродов T3-F4	$M = \begin{pmatrix} 0.45606 \\ 0.69579 \\ 0.93551 \end{pmatrix}$	$C = \begin{pmatrix} 0.55860 \\ 0.77034 \\ 0.98207 \end{pmatrix}$	$B = \begin{pmatrix} 0.70243 \\ 0.92614 \\ 1.14985 \end{pmatrix}$
Взаимная информация электродов T3-F8	$M = \begin{pmatrix} 0.42876 \\ 0.67111 \\ 0.91347 \end{pmatrix}$	$C = \begin{pmatrix} 0.47225 \\ 0.71230 \\ 0.95236 \end{pmatrix}$	$B = \begin{pmatrix} 0.64522 \\ 0.86636 \\ 1.08750 \end{pmatrix}$
Взаимная информация электродов T3-T4	$M = \begin{pmatrix} 0.43869 \\ 0.67374 \\ 0.90879 \end{pmatrix}$	$C = \begin{pmatrix} 0.54820 \\ 0.76042 \\ 0.97264 \end{pmatrix}$	$B = \begin{pmatrix} 0.69717 \\ 0.91653 \\ 1.13590 \end{pmatrix}$
Взаимная информация электродов F4-F8	$M = \begin{pmatrix} 0.44522 \\ 0.69899 \\ 0.95277 \end{pmatrix}$	$C = \begin{pmatrix} 0.48935 \\ 0.73818 \\ 0.98701 \end{pmatrix}$	$B = \begin{pmatrix} 0.67280 \\ 0.88583 \\ 1.09887 \end{pmatrix}$
Взаимная информация электродов F4-T4	$M = \begin{pmatrix} 0.53114 \\ 0.76252 \\ 0.99391 \end{pmatrix}$	$C = \begin{pmatrix} 0.59334 \\ 0.82005 \\ 1.04677 \end{pmatrix}$	$B = \begin{pmatrix} 0.71765 \\ 0.95691 \\ 1.19618 \end{pmatrix}$
Взаимная информация электродов F8-T4	$M = \begin{pmatrix} 0.4898 \\ 0.70304 \\ 0.91622 \end{pmatrix}$	$C = \begin{pmatrix} 0.54132 \\ 0.75429 \\ 0.96726 \end{pmatrix}$	$B = \begin{pmatrix} 0.67788 \\ 0.87979 \\ 1.08171 \end{pmatrix}$

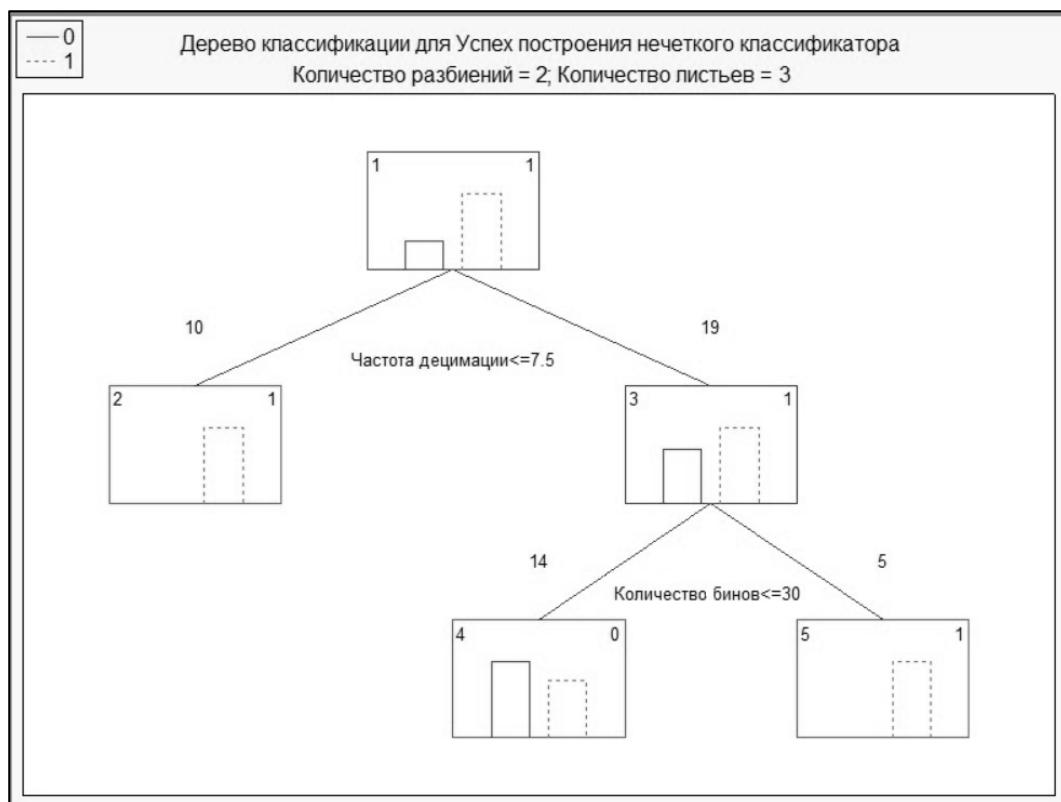
Правила имеют вид:

- [1] ЕСЛИ  $x_1 = M$  И  $x_2 = M$  И  $x_3 = M$  И  $x_4 = M$  И  $x_5 = M$  И  $x_6 = M$  И  $x_7 = M$  И  $x_8 = M$  И  $x_9 = M$  И  $x_{10} = M$  И  $x_{11} = M$  И  $x_{12} = M$  И  $x_{13} = M$  И  $x_{14} = M$ , ТО  $class = 0$ ;
- [2] ЕСЛИ  $x_1 = C$  И  $x_2 = C$  И  $x_3 = C$  И  $x_4 = C$  И  $x_5 = C$  И  $x_6 = C$  И  $x_7 = C$  И  $x_8 = C$  И  $x_9 = C$  И  $x_{10} = C$  И  $x_{11} = C$  И  $x_{12} = C$  И  $x_{13} = C$  И  $x_{14} = C$ , ТО  $class = 1$ ;
- [3] ЕСЛИ  $x_1 = B$  И  $x_2 = B$  И  $x_3 = B$  И  $x_4 = B$  И  $x_5 = B$  И  $x_6 = B$  И  $x_7 = B$  И  $x_8 = B$  И  $x_9 = B$  И  $x_{10} = B$  И  $x_{11} = B$  И  $x_{12} = B$  И  $x_{13} = B$  И  $x_{14} = B$ , ТО  $class = 1$ .

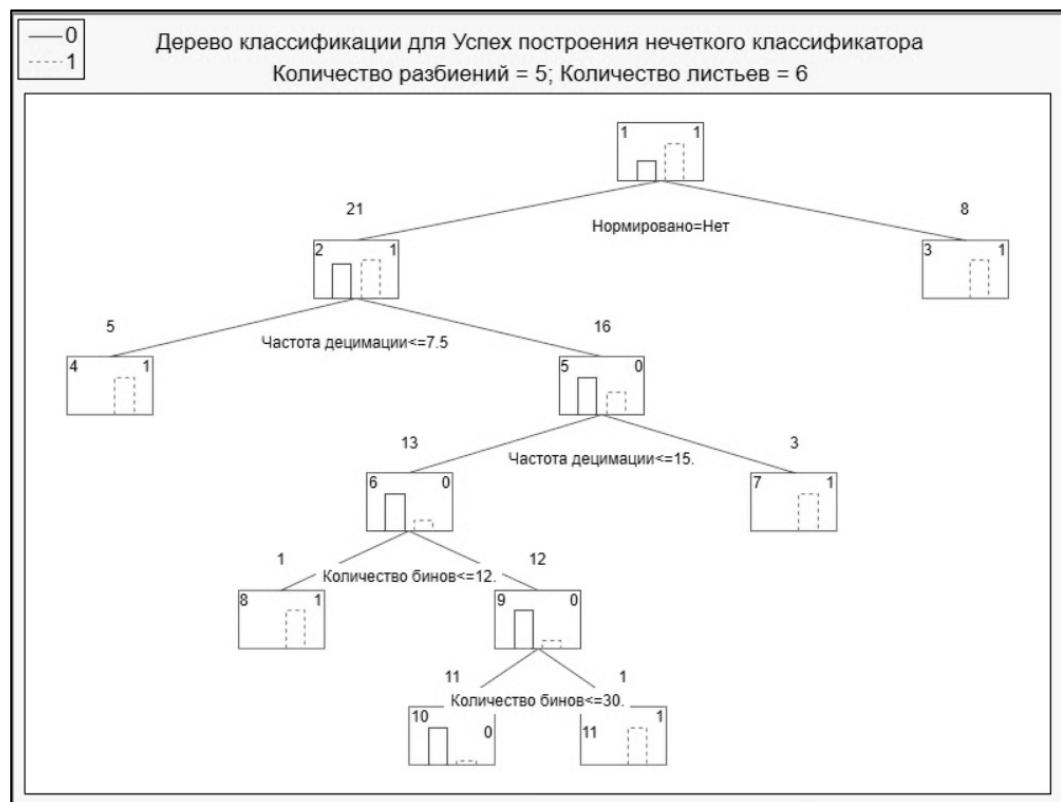
Теперь исследуем подробнее вопрос о возможности построения нечеткого классификатора на основе данных ЭЭГ. Будем считать, что классификатор не построен, если какое-то или оба из значений чувствительности или специфичности равны 0. В противном случае будем считать, что классификатор построен. Невозможность построить классификатор определяется значениями параметров алгоритма и самими данными. На рис. 4 показаны деревья классификации, полученные с помощью пакета STATISTICA, для реального движения (рис. 4.17-а) и воображаемого движения (рис. 4.17-б), в которых независимыми признаками являются параметры алгоритма, а зависимым – бинарное значение, было ли построен нечеткий классификатор.

Точность распознавания для деревьев классификации 73% и 86% для реальных и воображаемых движений соответственно. Как можно видеть на рис. 4.17, для реальных движений полученное дерево характеризуется практически равными долями классов в единственном листе, выдающем значение класса 0, что говорит о высокой неоднородности дерева, в то время как для воображаемых движений дерево демонстрирует хорошее разделение классов.

Таким образом, значимость в построении классификатора самого временного ряда ЭЭГ существенно более весома для реальных движений верхними конечностями. Очевидно, что высокая индивидуальная вариабельность ЭЭГ может служить одной из причин того, что в ряде случаев классификатор не может быть построен. В этом случае требуются дальнейшие исследования для оценки перспективности подходов новой методологии, заключающейся в оценке возможностей построения индивидуально ориентированных классификаторов ЭЭГ применительно к разным задачам. Следует отметить, что именно реальные движения с точки зрения асинхронных интерфейсов «мозг-компьютер» детектируются лучше, чем движения, выполняемые оператором мысленно. Таким образом, худший прогноз построения классификаторов для реальных движений, при лучшей работе самих классификаторов требует более детального изучения. Значимость отдельных параметров алгоритма в шкале 0–100 (0 – наименьшая значимость, 100 – наибольшая значимость) представлена в табл. 4.3.



a)



б)

Рис. 4.17. Деревья классификации возможности построения нечеткого классификатора

Таблица 4.3. Значимость параметров алгоритма для построения классификатора

Параметр	Реальное движение	Воображаемое движение
Частота децимации	81	100
Количество бинов	100	80
Нормализация данных	26	47

Коэффициент корреляции Спирмена для данных об успехе построения классификатора для реальных и воображаемых движений равен 0.03918, что свидетельствует об отсутствии корреляции между этими двумя признаками. Таким образом, успешное построение нечеткого классификатора для реальных движений не гарантирует успешного построения нечеткого классификатора для воображаемых движений, и наоборот.

Можно сформулировать следующие выводы на основе анализа результатов вычислительного эксперимента:

1. После предобработки данных на основе взаимной информации удалось построить нечеткий классификатор в 72% случаев для реальных движений и в 48% для воображаемых, что демонстрирует, известную из литературных источников, большую сложность классификации воображаемых движений, по сравнению с реально выполняемыми пользователем.

2. Среднее значение доли правильных ответов составляет 74.15% для реальных движений и 60.06% для мысленных, что с учетом эксперимента с нетренированными пользователями представляет собой результат, достаточный для первичного построения архитектуры асинхронного ИМК для конкретного пользователя.

3. Установлено, что точность классификации значительно не меняется при сокращении количества используемых электродов до 6 шт., соответствующих височно-теменной области.

4. Наиболее значимыми параметрами алгоритма для построения классификатора для реальных движений является количество бинов при вычислении взаимной информации, а для воображаемых – частота децимации.

5. Корреляция между возможностью построения классификатора для

реальных движений и возможностью построения классификатора для воображаемых движений не наблюдается.

Таким образом, в рамках данного исследования был предложен новый метод для классификации ЭЭГ и проведено его исследование для реальных и мысленных движений. В работе показана принципиальная возможность различия паттернов ЭЭГ реальных и воображаемых движений обеих рук с помощью нечеткого классификатора, анализирующего данные взаимной информации. На основе разбиения алгоритмом Густавсона-Кесселя получены лингвистические шкалы для нечетких переменных, как корректные, так и некорректные, что оставляет возможность для дальнейшей оптимизации базы правил путем модификации некорректных лингвистических шкал и удаления некоторых переменных из базы правил.

Анализ возможности построения классификаторов, при котором качество классификации оценено с помощью трех показателей качества — доли правильных ответов, чувствительности и специфичности, продемонстрировал, что примерно для половины испытуемых удалось построить нечеткий классификатор для воображаемых движений, и для 72% испытуемых удалось построить классификатор для реальных движений. Максимальная точность для отдельного человека составляет 96% при значении чувствительности 92% и значении специфичности 100% для реальных движений. Максимальная точность для отдельного человека для воображаемых движений составляет 71% при значениях чувствительности 78% и специфичности 62%.

#### **4.3. Применение программного комплекса «FuzzyLogicCore» для решения задачи вставки пропущенных значений**

В практических задачах анализа данных выборки часто содержат в себе пропущенные значения. Причины могут быть различными, например, невозможность ответа респондента на конкретный вопрос анкеты, отказ работы датчика для измерений показателя при низких температурах, ошибки в программном обеспечении при записи и обработке данных. За редким исключением алгоритмы

машинного обучения не работают с выборками, имеющими пропущенные значения. Поэтому возникает необходимость перейти к данным, не имеющим пропусков, для дальнейшей работы с ними.

Существуют различные подходы к решению задачи восстановления пропущенных значений, которые различаются по своей природе, области применимости и вычислительной сложности. Так, наиболее простым в реализации является отбрасывание записей. В качестве недостатка данного подхода можно отметить уменьшение количества наблюдаемых данных. Следующий подход – отбрасывание признака. Данный подход применим только в том случае, если признак является неинформативным. Третий подход – внесение недостающих значений (missing data imputation), которое может быть осуществлено несколькими способами. Первый способ заключается в замене недостающих значений заранее определенными константными значениями, что может привести к наличию значений, сильно отличающихся от настоящих. Во втором подходе внесение значений осуществляется с помощью вычисления статистических значений (например, среднего) для каждого признака. Это достаточно распространенный подход, так как статистические значения легко вычислить, и они дают хорошие результаты. И, наконец, отсутствующие значения могут быть заполнены с помощью различных алгоритмов, таких как метод k ближайших соседей (k-nearest neighbors), случайный лес (random forest) [86], линейная регрессия (linear regression), EM-алгоритм (EM-algorithm) [92], метод K-средних (K-means) [147], алгоритм ZET [18] и т. д. В данной работе для внесения недостающих значений предлагается использовать нечеткую систему. Данное предложение основано на теореме о нечеткой аппроксимации (Fuzzy Approximation Theorem, FAT), которая утверждает, что любая математическая система может быть аппроксимирована системой, основанной на нечеткой логике.

Имеется матрица объектов-признаков  $X^{n \times d}$ ,  $n$  – количество объектов,  $d$  – количество признаков. Часть значений матрицы пропущены. Необходимо получить матрицу объектов-признаков без пропущенных значений с целью дальнейшего применения алгоритма машинного обучения.

В [42, 153] предложено использовать НПС для решения данной задачи.

То обстоятельство, что нечеткая система работает с одной входной и одной выходной переменной, делает ее подходящей для восстановления данных, поступающих с датчиков, и имеющих формат «время»-«значение». В качестве экспериментальных данных использовались измерения акселерометра телефона, доступные в [171]. Эксперимент проводился на 1000 точек данных. Данные включают в себя следующие признаки: отметка времени, ускорение по X. Использование данных без пропусков мотивировано тем, что в таком случае можно управлять количеством пропущенных значений, а также информация об истинных значениях пропусков позволяет сравнить восстановленные значения напрямую. Создание пропущенных значений проводилось следующим образом: для каждого значения, входящего в подмножество выбранных признаков, с некоторой вероятностью создается пропуск так, чтобы доля пропусков во всем наборе данных соответствовала заданной. Вычислительный эксперимент проводился для нескольких значений доли пропусков в данных, а именно: 1%, 6% и 30% пропущенных значений.

Для оценки результаты заполнения пропущенных значений, выполненные с помощью нечеткой системы, сравниваются с результатами, выполненными с помощью различных подходов: заполнение средним значением, заполнение методом k ближайших соседей, методом случайного леса, с помощью линейной регрессии, ЕМ-алгоритмом, алгоритмом K-средних и алгоритмом ZET.

Точность рассчитывается на основе корня из среднеквадратичной ошибки. Результаты сравнения приведены в табл. 4.4 и на рис. 4.18.

На основании результатов эксперимента можно сделать следующие выводы:

- 1) С увеличением числа кластеров в нечеткой системе ошибка уменьшается так как описание данных становится более детализированным.
- 2) Доля пропущенных значений сильно влияет на величину ошибки при восстановлении с помощью нечеткой системы.

3) В целом, на низком проценте пропусков данных (1%) заполнение данных с помощью нечеткой системы позволяет получить большую точность чем восстановление данных другими методами.

4) В остальных случаях метод случайного леса показывает лучшие результаты.

Таблица 4.4. Корень из среднеквадратичной ошибки

Подход		Процент пропущенных значений		
		1%	6%	30%
HC	5 кластеров	0.72883	1.21702	1.08404
	9 кластеров	0.62315	1.19603	1.04680
	17 кластеров	<b>0.51899</b>	1.02454	1.03156
Среднее значение		1.09752	1.09703	1.01524
kNN		0.78396	0.69430	0.82206
Случайный лес		0.54502	<b>0.61347</b>	<b>0.69246</b>
Линейная регрессия		1.07149	1.08875	1.00819
EM-алгоритм		1.07149	1.08875	1.00802
K-средние		1.08478	1.04041	0.94379
ZET		0.76613	0.67182	0.78704

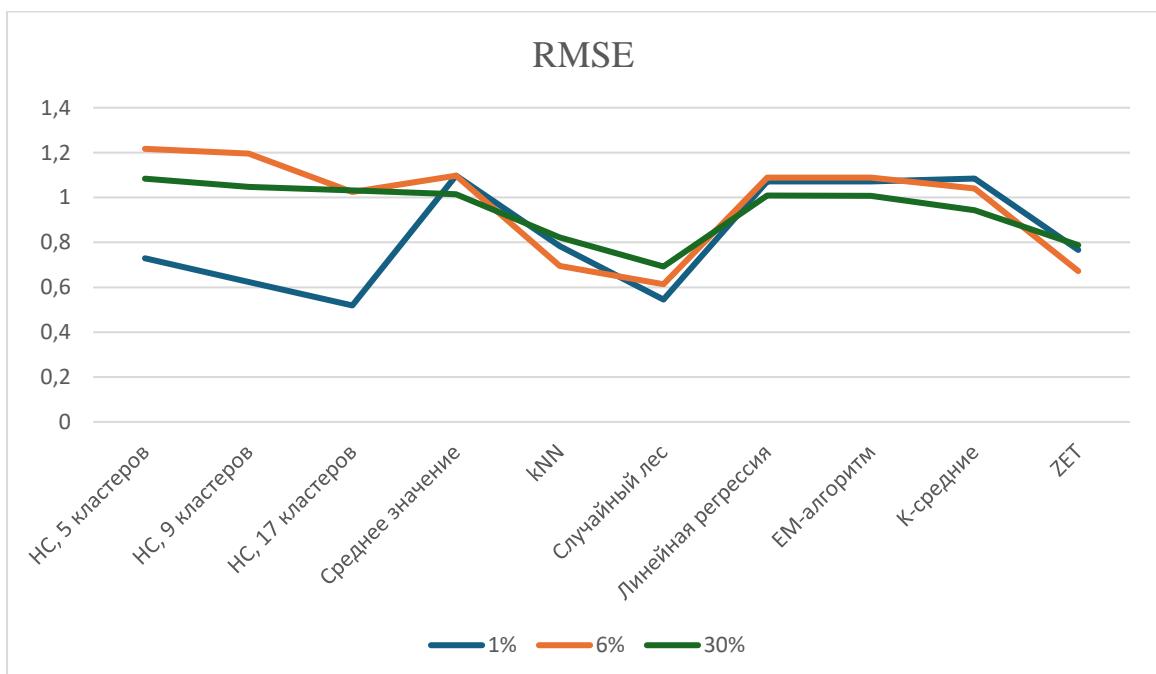


Рис. 4.18. Результаты сравнения алгоритмов восстановления пропусков в данных

## Выводы по четвертой главе

1. Разработаны библиотеки классов, реализующие алгоритмы для работы с предложенными нечеткими метриками, алгоритмами кластеризации, аппроксимации эллипсоидами и интерполяции, функциями принадлежности и осуществления нечеткого логического вывода с использованием разработанных методов на языке Java с применением принципов ООП и паттернов проектирования.

2. Представлен разработанный программный комплекс в виде RESTful-сервиса, обеспечивающий полный цикл работы с нечеткими моделями, а именно: генерацию базы знаний для нечетких классификаторов, лингвистической модели и модели Такаги-Сугено на основе экспериментальных данных, выполнение нечеткого вывода и тестирование моделей, позволяющий интегрировать разработанный функционал в различные системы и приложения, предоставляя гибкий и удобный интерфейс для работы с нечеткой логикой.

3. Проведен вычислительный эксперимент с использованием разработанного ПК для проверки работоспособности алгоритма формирования баз знаний на примере решения задачи классификации ЭЭГ при распознавании движений, как реальных, так и воображаемых для реализации ИМК. В ходе решения также был произведен анализ значимости признаков, влияющих на результат распознавания: значение частоты децимации при предобработке, количество бинов при вычислении показателей взаимной информации каналов, были или нет нормализованы данные при предварительной обработке.

4. Проведен вычислительный эксперимент с использованием разработанного ПК для проверки работоспособности алгоритма формирования баз знаний на примере решения задач восстановления пропущенных значений в данных для дальнейшего применения методов машинного обучения. Предложенный метод сравнивался с рядом распространенных методов восстановления пропущенных значений: заполнение средним значением, kNN, случайный лес, линейная регрессия, EM-алгоритм, К-средних, ZET. Результаты эксперимента показали высокую точность предложенного алгоритма.

## ЗАКЛЮЧЕНИЕ

1. Анализ подходов к построению баз знаний НПС показал, что при наличии наблюдаемых или экспериментальных данных, на основе которых можно сформировать обучающую выборку, для автоматизации данного процесса целесообразно использовать процедуры кластеризации, в том числе метрические алгоритмы.

2. Для автоматического построения баз знаний НПС на основе кластерных процедур предложен метод, основанный на построении эллипсоидов минимального объема (эллипсов минимальной площади), что позволило по сравнению с подходом на основе ковариационных матриц улучшить качество аппроксимации тестовых данных при том же уровне интерпретируемости примерно на 6-7 %.

3. Впервые предложено и исследовано семейство нечетких метрик, базирующихся на аддитивных генераторах непрерывных архimedовых треугольных норм из класса рациональных функций. Установлено, что нечеткие метрики имеют характерный уровень различимости, что важно при разработке рекомендаций по их использованию в задачах кластеризации. Анализ результатов вычислительного эксперимента показал значительное превосходство нечетких метрик перед обычными метриками по большинству показателей качества кластеризации. Предложен подход к формированию комплексной оценки нечетких метрик с учетом этих показателей.

4. Разработан программный комплекс для автоматизированного формирования баз знаний НПС, который был применен для ряда задач, среди которых – построение нечеткого классификатора для интерфейса «мозг-компьютер» с целью анализа электроэнцефалограмм. Для 48% испытуемых удалось построить нечеткий классификатор для воображаемых движений, и для 72% испытуемых был построен классификатор для реальных движений. Кроме того, в процессе исследования были выявлены наиболее значимые признаки для

классификации, что позволило сократить количество электродов в эксперименте до 6.

**Рекомендации по использованию.** Результаты, касающиеся новых нечетких метрик, расширяют теоретическую базу для построения метрических пространств на основе непрерывных рациональных архimedовых треугольных норм. Их целесообразно использовать в метрических алгоритмах кластеризации для управления степенью различности объектов. Процедура формирования базы знаний на основе эллипсоидальной кластеризации может использоваться при разработке интеллектуальных информационных систем различного назначения в ситуации, когда имеются исторические или наблюдаемые данные значительного объема.

**Перспективы** развития проведенного исследования связаны, прежде всего, с оптимизацией сформированной базы правил, а также с интерполяцией нечетких правил, что позволит реализовать процедуру логического вывода с использованием неполной базы правил.

## СПИСОК ЛИТЕРАТУРЫ

1. Аверкин А.Н. Извлечение правил из классических, глубоких нейронных сетей и нейро-нечетких сетей / А.Н. Аверкин // Речевые технологии. – 2020. – № 3-4. – С. 76-85.
2. Бардамова М. Б. Формирование структуры нечеткого классификатора комбинацией алгоритма экстремумов классов и алгоритма «прыгающих лягушек» для несбалансированных данных с двумя классами / М. Б. Бардамова, И. А. Ходашинский // АВТОМЕТРИЯ. – 2021. – Т. 57. – № 4. – С. 54-64.
3. Барсегян А.А. Анализ данных и процессов: учеб. пособие / А.А. Барсегян, М.С. Куприянов, И.И. Холод, М.Д. Тесс, С.И. Елизаров. – 3-е изд., перераб. и доп. – СПб. : БХВ-Петербург, 2009. – 512 с.
4. Бедринцев А.А. Представление данных с помощью экстремальных эллипсоидов / А.А. Бедринцев, В.В. Чепыжков // Информационные технологии и системы. – Калининград, Россия. – 2013. – С. 55-60.
5. Борисов В.В. Нечеткие модели и сети / В.В. Борисов, В.В. Круглов, А.С. Федулов. – 2-е изд., стер. – М. : Горячая линия – Телеком, 2012 . – 284 с.
6. Брагин А. Д. Распознавание моторных образов на электроэнцефалограммах с применением свёрточных нейронных сетей / А. Д. Брагин, В. Г. Спицын // Компьютерная оптика. – 2020. – Т. 44. – № 3. – С. 482-489.
7. Воронцов К. В. Лекции по метрическим алгоритмам классификации [Электронный ресурс]. – Режим доступа: <http://www.ccas.ru/frc/papers/voron04mpc.pdf>.
8. Гавrilova Т. А. Базы знаний интеллектуальных систем / Т. А. Гаврилова, В. Ф. Хорошевский. – СПб : Питер, 2000. – 384 с.
9. Гамма Э. Паттерны объектно-ориентированного проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. – СПб.: Питер, 2020. – 448 с.
10. Гордлеева С. Ю. Управление роботизированным экзоскелетоном на основе технологии «Интерфейс мозг-компьютер» моторно-воображаемого типа / С.

Ю. Гордлеева, М. В. Лукоянов, С. А. Минеев, М. А. Хоружко, В. И. Миронов, А. Я. Каплан, В. Б. Казанцев // Соврем. технол. мед. – 2017. – Т. 9. – №3. – С. 31-38.

11. Грюнбаум Б. Этюды по комбинаторной геометрии и теории выпуклых тел: Пер. с англ. / Б. Грюнбаум. – М. : Наука, 1971. – 349 с.

12. Дикарев П. В. Система распознавания аварийных режимов воздушных линий электропередачи с использованием нечеткой логики / П. В. Дикарев, А. А. Шилин, С. Ю. Юдин // Энерго- и ресурсосбережение: промышленность и транспорт. – 2022. – Т. 38. – № 1. – С. 6-12.

13. Дубенко Ю. В. Нечеткая система определения оптимальных методов для прогнозирования параметров сложных технических систем / Ю. В. Дубенко, Е. Е. Дышкант // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2018. – Т. 47. – № 3. – С. 58-69.

14. Елизаров С. И. Проблема определения количества кластеров при использовании методов разбиения / С. И. Елизаров, М. С. Куприянов // Изв. ВУЗов. Приборостроение. – 2009. – Т. 52. – № 12. – С. 3-8.

15. Еникеева А. И. Разработка на основе нечеткой логики экспертной системы для диагностики пневмонии / А. И. Еникеева, Р. А Бурнашев, Р. Р. Фарахов // Электронные библиотеки. – 2024. – С. 503-532.

16. Ермаков Р. Н. Классификация сетевых протоколов с применением методов машинного обучения и алгоритмов нечеткой логики в системах анализа трафика / Р. Н. Ермаков // Автоматизация, телемеханизация и связь в нефтяной промышленности. – 2019. – Т. 555. – № 10. – С. 39-49.

17. Журавская А. Применение неинвазивного интерфейса «мозг-компьютер» для классификации воображаемых движений нижних конечностей человека / А. Журавская, Л.А. Станкевич // Системный анализ в проектировании и управлении: сборник научных трудов XXV Международной научной и учебно-практической конференции, 2021 г.: [в 3 частях]. – СПб. : ПОЛИТЕХ-ПРЕСС, 2021. – Ч. 3. – С. 146-158.

18. Загоруйко Н. Г. Прикладные методы анализа данных и знаний / Н. Г. Загоруйко. – Новосибирск : Изд-во Ин-та математики, 1999. – 270 с.

19. Зверева Е.Н. Сборник примеров и задач по основам теории информации и кодирования сообщений / Е.Н. Зверева, Е.Г. Лебедько – СПб: НИУ ИТМО, 2014. – 76 с.
20. Колмогоров А. Н. Элементы теории функций и функционального анализа: [Учебник для мат. специальностей ун-тов] / А. Н. Колмогоров, С. В. Фомин. - 4-е изд., перераб. - Москва : Наука, 1976. - 543 с.
21. Кондратенко Ю. П. Генерация баз правил нечетких систем на основе модифицированных муравьиных алгоритмов / Ю. П. Кондратенко, А. В. Козлов // Проблемы управления и информатики. – 2019. – № 2. – С. 59-79.
22. Корышев Н. П. Алгоритм формирования базы правил нечёткого классификатора на основе алгоритма кластеризации k-средних и метаэвристического алгоритма "китов" / Н. П. Корышев, И. А. Ходашинский // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2021. – Т. 24. – № 1. – С. 42-47.
23. Кофман А. Введение в теорию нечетких множеств / А. Кофман; Перевод с фр. В. Б. Кузьмина. – М. : Радио и связь, 1982. – 432 с.
24. Куркин С. А. Использование искусственных нейронных сетей для классификации электрической активности головного мозга в процессе воображения движений у нетренированных испытуемых / С. А. Куркин, Е. Н. Пицик, А. Е. Храмов // Информационно-управляющие системы. – 2019. - № 6. – С. 77-84.
25. Леденева Т.М. Агрегирование информации в оценочных системах / Т.М. Леденева, С.Л. Подвальный // Вестник ВГУ. Серия: Системный анализ и информационные технологии. – 2016. – №4. – С. 155-164.
26. Леденева Т. М. Интеллектуальные информационные системы: Учеб. пособие / Т. М. Леденева. – Воронеж : ВГТУ, 2001. – 136 с.
27. Леденева Т.М. Нечеткие метрики на основе генераторов архимедовых треугольных норм из класса рациональных функций / Т.М. Леденева, Т.А. Моисеева // Искусственный интеллект и принятие решений, 2024. – № 4. – С. 30-44.

28. Леденева Т. М. Нечеткое моделирование медицинских экспертных систем / Т. М. Леденева, С. Л. Подвальный, Р. К. Стрюков, Дегтярев С. В. // Биомедицинская радиоэлектроника, 2016. – № 9. – С. 16-24.
29. Леденева Т. М. О проблеме обеспечения некоторых требований к базе знаний в нечетких системах / Т. М. Леденева, С. А. Моисеев // Вестник ВГУ. Серия: системный анализ и информационные технологии. – 2012. – № 1. – С. 149-156.
30. Леденева Т.М. Об одном подходе к формированию базы знаний для сегментации временных рядов / Т.М. Леденева, М.А. Сергиенко // Вестник ВГУ. Серия: Системный анализ и информационные технологии. – 2017. – № 3. – С. 156-164.
31. Леденева Т.М. Обзор основных классов операторов порядкового взвешенного агрегирования / Т.М. Леденева, И.Н. Левкина // Вестник ВГУ. Серия: Системный анализ и информационные технологии. – 2022. – №1. – С. 5-31.
32. Леденева Т.М. Обзор типов продукционных правил в системах нечеткого логического вывода / Т.М. Леденева, Т.А. Моисеева // Актуальные проблемы прикладной математики, информатики и механики: сборник трудов Международной научно-технической конференции, Воронеж, 2022 г. – С. 1793-1798.
33. Леденева Т. М. Обобщенные ассоциативные правила / Т.М. Леденева, Е.А. Кретов // Вестник Воронежского государственного технического университета. – 2014. – Т. 10. – № 5. – С. 46-49.
34. Леденева Т. М. Основы нечеткого моделирования: учебно-методическое пособие / Т. М. Леденева. – Воронеж : ВГУ, 2022. – 128 с.
35. Леденева Т. М. Особенности реализации механизма нечеткого логического вывода в нечетких системах / Т. М. Леденева, А. Д. Решетников // Международный научно-исследовательский журнал. – 2021. – № 6. – Часть 1. – Т. 108. – С. 107-117.
36. Леденева Т.М. Сравнительный анализ процедур кластеризации для обнаружения аномалий показателей, характеризующих функционирование контактной сети железных дорог / Т.М. Леденева, Т.А. Моисеева // Вестник

Воронежского государственного университета. Серия: Системный анализ и информационные технологии, 2020. – № 2. – С. 115-126.

37. Леденева Т.М. Транзитивность как особое свойство нечетких отношений / Т.М. Леденева, Н.А. Каплиева; Воронеж, ун-т. – Воронеж, 2006. – 51 с. – Деп. В ВИНИТИ 07.12.2006, № 1523-В2006.

38. Моисеева, Т.А. Влияние метрики на выявленную структуру временных задержек сигналов в задачах оценки электрогенеза мозга / Т.А. Моисеева, Я.А. Туровский // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии, 2024. – № 2. – С. 103-112.

39. Моисеева, Т. А. Генерация базы знаний на основе нечеткой кластеризации / Т. А. Моисеева, Т. М. Леденева // Информационные технологии и вычислительные системы. 2023. – № 1. – С. 97-108.

40. Моисеева, Т.А. Нечеткий классификатор электроэнцефалограмм для интерфейса «мозг-компьютер» / Т.А. Моисеева, Я.А. Туровский, Т.М. Леденева // Вестник Воронежского государственного университета. Системный анализ и информационные технологии, 2024. – № 4. – С. 129-142.

41. Моисеева, Т.А. Разработка коэффициента неразличимости для нечеткой метрики / Т.А. Моисеева // Межвузовская научная конференция молодых ученых и студентов «Математика, информационные технологии, приложения», Воронеж, 24-25 апреля 2024 г. – Воронеж, 2024. – С. 573-580.

42. Моисеева, Т. А. Реализация нечеткой системы для решения проблемы обработки пропусков в данных / Т.А. Моисеева // Актуальные проблемы прикладной математики, информатики и механики: сборник трудов Международной научной конференции, Воронеж, 4-6 декабря 2023 г. – Воронеж, 2024. – С. 555-560.

43. Моисеева Т.А. Сравнительный анализ качества баз знаний, построенных на основе процедур кластеризации / Т.А. Моисеева, Т.М. Леденева // Актуальные проблемы прикладной математики, информатики и механики: сборник трудов Международной научной конференции. Воронежский государственный университет. Воронеж, 2023. – С. 510-516.

44. Мокиенко О.А. Воображение движения и его практическое применение / О.А. Мокиенко, Л.А. Черникова, А. А. Фролов, П. Д. Бобров // Журнал высшей нервной деятельности им И. П. Павлова. – 2013. – Т. 63. - №2. – С. 195-204.
45. Мокиенко О.А. Интерфейс мозг-компьютер: первый опыт клинического применения в России / О. А. Мокиенко, Р. Х. Люкманов, Л. А. Черникова, Н. А. Супонева, М. А. Пирадов, А. А. Фролов // Физиология человека. – 2016. – Т. 42. – №1. – С.31-39
46. Нифантов В. М. Диагностика, оценка и прогнозирование технического состояния технологического оборудования при помощи нечеткой экспертной системы в централизованных системах технического обслуживания и ремонта / В. М. Нифантов // Труды Кольского научного центра РАН. – 2019. – Т. 10. – № 9. – С. 187-197.
47. О развитии искусственного интеллекта в Российской Федерации (вместе с «Национальной стратегией развития искусственного интеллекта на период до 2030 года»): Указ Президента РФ от 10.10.2019 № 490 (ред. от 15.02.2024) // Собрание законодательства РФ, 2019, № 41, ст. 5700; 2024, № 8, ст. 1102.
48. Овакимян А. С. Построение Fuzzy-систем принятия решений / А. С. Овакимян, С. Г. Саркисян, А. А. Зироян, М. А. Мартынова // Современная экономика: проблемы и решения. – 2017. – Т. 10. – С. 48-55.
49. Остапенко Р.О. Формирование базы правил нечёткого классификатора с помощью метаэвристического алгоритма «саранчи» / Р.О. Остапенко, И.А. Ходашинский // Доклады ТУСУР. – 2022. – Т. 25. – № 2. – С. 31-36.
50. Павленко Д.В. Применение глубокого обучения в интерфейсах мозг–компьютер для распознавания движений / Д.В. Павленко, Ш.Э. Татарис, В.В. Овчаренко // Программные продукты и системы. – 2024. – Т. 37. – № 2. – С. 164-169.
51. Панченко Т. В. Генетические алгоритмы: учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань : Издательский дом «Астраханский университет», 2007. – 87 с.

52. Пегат А. Нечеткое моделирование и управление: Пер. с англ. / А. Пегат. – М. : БИНОМ. Лаборатория знаний, 2009. – 798 с.
53. Реклейтис Г. Оптимизация в технике: в 2 книгах / Г. Реклейтис, А. Рейвиндрен, К. Рэгсдел ; пер. с англ. В. Я. Алтаева, В. И. Моторина. – М. : Мир, 1986. – Кн. 1. – 320 с.
54. Романов А. В. Система диагностики и прогнозирования остаточного ресурса приводов мобильных роботов на базе нечеткой логики / А. В. Романов, Л. В. Запотичная, И. О. Малышонков, А. С. Урсегова // КИП и автоматика: обслуживание и ремонт. – 2021. – № 2. – С. 59-65.
55. Руннова А.Е. Классификация паттернов двигательной активности на ЭЭГ-данных / А.Е. Руннова, В.Ю. Мусатов, Р.А. Куланин, С.В. Пчелинцева, Т.Ю. Ефремова, В.В. Грубов, Д.В. Лопатин // Вестник российских университетов. Математика. 2017. – Т. 22. – №5-2. – С. 1127-1132.
56. Рябчиков М. Ю. Система управления температурой пара после пароперегревательной установки с применением нечеткой логики для упреждающей компенсации возмущений / М. Ю. Рябчиков, Е. С. Рябчикова, С. А. Филиппов // Мехатроника, автоматизация, управление. – 2021. – Т. 22. – № 4. – С. 181-190.
57. Свидетельство о государственной регистрации программы для ЭВМ 2024661436 Российская Федерация. Программа для генерации нечеткой системы для решения задачи аппроксимации / Т.А. Моисеева; заявитель и правообладатель Федеральное государственное бюджетное образовательное учреждение высшего образования «Воронежский государственный университет». – № 2024661436; заявление 13.05.2024 ; опубл. 17.05.2024.
58. Свидетельство о государственной регистрации программы для ЭВМ 2024661362 Российская Федерация. Программа для выполнения кластеризации методом нечетких к-медоид с использованием нечеткой метрики / Т.А. Моисеева; заявитель и правообладатель Федеральное государственное бюджетное образовательное учреждение высшего образования «Воронежский

государственный университет». – № 2024661362; заявление 13.05.2024 ; опубл. 17.05.2024.

59. Сонькин К.М. Классификация электроэнцефалографических паттернов воображаемых и реальных движений пальцев одной руки методом опорных векторов / К.М. Сонькин, Л.А. Станкевич, Ю.Г. Хоменко, Ж. В. Нагорнова, Н.В. Шемякина // Классификация электроэнцефалографических паттернов воображаемых и реальных движений пальцев одной руки методом опорных векторов // Тихоокеанский медицинский журнал. – 2014. – № 2. – С.30-35.

60. Сорокин А. А. Морфогенез баз правил систем нечеткого вывода в условиях контрадикторных экспертных мнений // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. – 2019. – № 4. – С. 95–105.

61. Статистика искусственного интеллекта (2025) [Электронный ресурс]. – Режим доступа: <https://inclient.ru/ai-stats/>

62. Стельмашенко А. В. Использование систем нечеткого логического вывода для создания алгоритмов систем автоматического ведения поездов / А.В. Стельмашенко, Т.В. Меренкова // В сборнике: Математика и ее приложения в современной науке и практике. Сборник научных статей VI Международной научно-практической конференции. – 2016. – С. 110-115.

63. Столбков Ю. К. Когнитивная двигательная реабилитация: воображение и наблюдение моторных действий / Ю. К. Столбков, Ю. П. Герасименко // Физиология человека. – 2021. – Т. 47. – № 1. – С. 123-132.

64. Татаркин Д. С. Исследование моделей нечеткого логического вывода / Д.С. Татаркин, Т.М. Леденева // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии. – 2006. – №2. – С. 110-118.

65. Хачумов М. В. Расстояния, метрики и кластерный анализ / М. В. Хачумов // Искусственный интеллект и принятие решений. – 2012. – № 1. – С. 81–89.

66. Ходашинский И.А. Алгоритмы «стадо криля» и кусочно-линейной инициализации для построения систем типа Такаги-Сугено / И.А. Ходашинский, К.С. Сарин, И.В. Филимоненко // Автометрия. – 2017. – Т. 53. – № 4. – С. 84-94.
67. Ходашинский И.А. Формирование структуры нечёткого классификатора. Обзор / И.А. Ходашинский // XIX Международная научно-практическая конференция. – 2023. – С. 277-281.
68. Черноусько Ф. Л. Оценивание фазового состояния динамических систем. Метод эллипсоидов / Ф. Л. Черноусько. – М. : Наука: Главная редакция физико-математической литературы, 1988. – 320 с.
69. Черный С. П. Один из подходов к моделированию двухкаскадной нечеткой системы управления электроприводом постоянного тока с двухзонным регулированием скорости / С. П. Черный, В. А. Соловьев, А. В. Бузикаева, С. И. Сухоруков // Электротехнические системы и комплексы. – 2022. – Т. 55. – № 2. – С. 32-39.
70. Шамирбек М. Е. Автоматизированная система управления микроклиматом теплицы на основе нечеткой логики / М. Е. Шамирбек // Приборы и системы. Управление, контроль, диагностика. – 2022. – № 9. – С. 10-18.
71. Шилдт Г. Java. Полное руководство / Шилдт, Г. – 12-е изд. : Пер. с англ. – СПб. : Диалектика, 2023. – 1344 с.
72. Шульгин С. К. Анализ системы управления углом тангажа квадрокоптера на основе нечеткого ПИД-регулятора / С. К. Шульгин, Д. О. Синепольский, Г. В. Ганнота // Вестник Луганского государственного университета имени Владимира Даля. – 2022. – Т. 59. – № 5. – С. 266-269.
73. Ярушкина П. Л. Основы теории нечетких и гибридных систем: учеб. пособие / П. Л. Ярушкина. – М. : Финансы и статистика, 2009. – 320 с.
74. 22 Top AI Statistics And Trends – Forbes Advisor [Electronic resource]. – Mode of access: <https://www.forbes.com/advisor/business/ai-statistics/>
75. Abadeh M. S. Induction of Fuzzy Classification Systems Using Evolutionary ACO-Based Algorithms / M. S. Abadeh, J. Habibi, E. Soroush // First Asia International Conference on Modelling and Simulation. – Phyket, Thailand. – 2007. – Pp. 346-351.

76. Abe S. Fuzzy rules extraction directly from numerical data and its application to pattern classification / S. Abe, M. S. Lan // IEEE Trans. on Fuzzy Systems. – 1995. – Vol. 3. – Pp.18-28.
77. Achiche S. Real/binary like coded genetic algorithm to automatically generate fuzzy knowledge bases / S. Achiche, L. Baron, M. Balazinski // Proceedings of the 4th IEEE Int. Conf. on Control and Automation. – 2003. – Pp. 799-803.
78. Alonso J.M. Designing Highly Interpretable Fuzzy Rule-Based Systems with Integration of Expert and Induced Knowledge / J.M. Alonso, L. Magdalena, S. Guillaume // 12th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU). – 2008. – Pp. 682-689.
79. Altaheri H. Deep learning techniques for classification of electroencephalogram (EEG) motor imagery (MI) signals: A review / H. Altaheri, G. Muhammad, M. Alsulaiman, S. U. Amin, G. A. Altwaijri, W. Abdul, M. A. Bencherif, M. Faisal // Neural Computing and Applications. – 2023. – Vol. 35. – No. 20. – Pp.14681-14722
80. Al-Zoubi M.D. New Outlier Detection Method Based on Fuzzy Clustering / M.D. Al-Zoubi, A. Al-Dahoud, A.A. Yahya // WSEAS Transactions on Information Science and Applications. – 2010. – Vol. 7. – No. 5. – Pp. 681-690.
81. Alzubi M. Fuzzy rule interpolation methods and FRI Toolbox / M. Alzubi, Z. Johanyák, S. Kovács // Journal of Theoretical and Applied Information Technology. – 2018. – Vol. 96. – No. 21. – Pp. 7227-7244.
82. Anderson E. The species problem in Iris / E. Anderson // Annals of the Missouri Botanical Garden. – 1936. – Vol. 23. – No. 3. – Pp. 457–509.
83. Angelov P. P. An evolutionary approach to fuzzy rule-based model synthesis using indices for rules / P. P. Angelov // Fuzzy Sets and Systems. – 2003. – Vol. 137. – No. 3. – Pp. 325-338.
84. Baldwin J.F. Simple Fuzzy Logic Rules Based on Fuzzy Decision Tree for Classification and Prediction Problem / J.F. Baldwin, D. Xie // IFIP International Federation for Information Processing / Eds. Z. Shi, Q. He. – Boston, MA : Springer, 2005. – Vol. 163.

85. Bezdek, J. C. Pattern Recognition with Fuzzy Objective Function Algorithms / Bezdek, J. C. – New York, NY : Springer, 1981. – 272 p.
86. Breiman L. Random forests / L. Breiman // Machine Learning. – 2001. – Vol. 45. – Pp. 5 32.
87. Caliński, T. A dendrite method for cluster analysis / T. Caliński, J. Harabasz // Communications in Statistics. – 1974. – Vol. 3. – No. 1. – Pp. 1-27.
88. Cintra M. E. Fuzzy Rules Generation using Genetic Algorithms with Self-adaptive Selection / M. E. Cintra, H. de Arruda Camargo // 2007 IEEE International Conference on Information Reuse and Integration. – Las Vegas, NV, USA. – 2007. – Pp. 261-266.
89. Daniachew A. A. Fuzzy Rules for Data Set Classification: A Hybrid Approach Using Rough Set and Grid Partitioning / A. A. Daniachew, A. B. Clevon, A. K. Avram, D. T. Chislon // International Journal of Enterprise Modelling. – 2019. – Vol. 13. – No. 3. – Pp. 156-173.
90. De Baets B. Metrics and T-equalities / B. De Baets, R. Mesiar // Journal of Mathematical Analysis and Applications. – 2002. – Vol. 267. – No 2. – Pp. 531-547.
91. Demirci M. Topological properties of the class of generators of an indistinguishability operator / M. Demirci // Fuzzy Sets and Systems. – 2004. – Vol. 143. – No 3. – Pp. 413-426.
92. Dempster A. P. Maximum likelihood from incomplete data via the EM algorithm / A. P. Dempster, N. M. Laird, D. B. Rubin // Journal of the Royal Statistical Society. – 1977. – Vol. 39. – Pp. 1-38.
93. Dickerson J. Fuzzy function approximation with ellipsoidal rules / J. Dickerson, B. Kosko // IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics. – 1996. – Vol. 26. – Pp. 542-560.
94. Dunn J. C. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters / J. C. Dunn // Journal of Cybernetics. – 1973. – Vol. 3. – No. 3. – Pp. 32-57.
95. Dunn J. C. Well-Separated Clusters and Optimal Fuzzy Partitions, / J. C. Dunn // Journal of Cybernetics. – 1974. – Vol. 4. – No. 1. – Pp. 95-104.

96. Ester M. A density-based algorithm for discovering clusters in large spatial databases with noise / M. Ester, H.P. Kriegel, S. Sander, and X. Xu // Proceedings of the 2nd International Conference on Knowledge Discovery and Data Minin. – Menlo Park, CA. – 1996. – Pp. 226-231.
97. Fisher D. Knowledge acquisition via incremental conceptual clustering / D. Fisher // Machine learning. – 1987. – Vol 2. – Pp. 139-172.
98. Fisher R.A. The use of multiple measurements in taxonomic problems / R. A. Fisher // Annals of Eugenics. – 1936. – Vol. 7. – No. 2. – Pp. 179–188.
99. Foucart S. A Mathematical Introduction to Compressive Sensing / S. Foucart, H. Rauhut. – NY: Birkhäuser New York, 2013. – 625 p.
100. Fu L. Rule Generation from Neural Networks / L. Fu // IEEE Transactions on Systems, Man and Cybernetics. – 1994. – Vol. 24. – Pp. 1114-1124.
101. García D. Overview of the SLAVE learning algorithm: A review of its evolution and prospects / D. García, A. González, R. Pérez // International Journal of Computational Intelligence Systems. – 2014. – Vol. 7. – Pp. 1194-1221.
102. Gaur P. A Sliding Window Common Spatial Pattern for Enhancing Motor Imagery Classification in EEG-BCI / P. Gaur, H. Gupta, A. Chowdhury, K. McCreadie, R. B. Pachori, H. Wang // IEEE Transactions on Instrumentation and Measurement. – 2021. – Vol. 70. – Pp. 1-9.
103. Geng X. An improved feature extraction algorithms of EEG signals based on motor imagery brain-computer interface / X. Geng, D. Li, H. Chen, P. Yu, H. Yan, M. Yue // Alexandria Engineering Journal. – 2022. – Vol. 61. – No 6. – Pp. 4807-4820.
104. Genther H. Rule weight generation for a fuzzy classification system based on fuzzy clustering methods / H. Genther, A. Konig, M. Glesner // Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference. – Orlando, FL, USA. – 1994. – Vol. 1. – Pp. 614-617.
105. George A. On Some Results in Fuzzy Metric Spaces / A. George, P. Veeramani // Fuzzy Sets and System. – 1994. – Vol.64. - Issue 64. – Pp. 395-399.
106. George A. On some results of analysis for fuzzy metric spaces / A. George, P. Veeramani // Fuzzy Sets and Systems. – 1997. – Vol. 90. – Pp. 365-368.

107. Gregori V. Characterizing completable fuzzy metric spaces / V. Gregori, S. Romaguera // Fuzzy Sets and Systems. – 2004. – Vol. 144. – Pp. 411-420.
108. Gregori V. Examples of fuzzy metrics and applications / V. Gregori, S. Morillas, A. Sapena // Fuzzy Sets and Systems. – 2011. – Vol. 170. – Pp. 95-111.
109. Gregori V. On completable fuzzy metric spaces / V. Gregori, J.-J. Miñana, S. Morillas // Fuzzy Sets and Systems. – 2015. – Vol. 267. – Pp. 133-139.
110. Gregori V. Some properties of fuzzy metric spaces / V. Gregori, S. Romaguera // Fuzzy Sets and Systems. – 2000. – Vol. 115. - Issue 3. – Pp. 485–489.
111. Grigorenko O. Two new methods to construct fuzzy metrics from metrics / O. Grigorenko, J.-J. Miñana, O. Valero // Fuzzy Sets and Systems. – 2023. – Vol. 467. – Pp. 108483.
112. Grubov V.V. Demonstration of brain noise on human EEG signals in perception of bistable images / V.V. Grubov, A.E. Runnova, M.K. Kurovskay, A.N. Pavlov, A.A. Koronovskii, A.E. Hramov // Proceedings of SPIE. – 2016. – Vol. 9707. – P. 97070Z1-7.
113. Gu X. Autonomous learning for fuzzy systems: a review / X. Gu, J. Han, Q. Shen, P. P. Angelov // Artificial Intelligence Review. – 2023. – Vol. 56. – Pp. 7549–7595.
114. Gu X. Particle Swarm Optimized Autonomous Learning Fuzzy System / X. Gu, Q. Shen, P. P. Angelov // IEEE Transactions on Cybernetics. – 2021. – Vol. 51. – No. 11. – Pp. 5352-5363.
115. Gustafson D. Fuzzy clustering with fuzzy covariance matrix / D. Gustafson, W. Kessel // Proceedings of IEEE CDC. – San Diego, CA, USA. – 1979. – Pp. 761–766.
116. Hall L.O. Generation of fuzzy rules from decision trees / L.O. Hall, P. Lande // Journal of Advanced Computational Intelligence. – 1998. – Vol. 2. – No. 4. – Pp. 128-133.
117. Han J. Data mining: concepts and techniques / J. Han, M. Kamber, J. Pei. – 3rd ed. – Elsevier Morgan Kaufmann, 2012. – 703 p.
118. Herrera F. A learning process for fuzzy control rules using genetic algorithms / F. Herrera, M. Lozano, J.L. Verdegay // Fuzzy Sets and Systems. – 1998. – Vol. 100. – Iss. 1-3. – Pp. 143-158.

119. Holland J.H. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence / J.H. Holland. – Ann Arbor, Michigan : University of Michigan Press, 1975. – 183 p.
120. Holland J. H. Escaping Brittleness: The Possibilities of General-Purpose Learning Algorithms Applied to Parallel Rule-Based Systems / J. H. Holland // Machine Learning: An Artificial Intelligence Approach / Eds. R.S. Michalski, J.G. Carbonell, T.M. Mitchell. - Los Altos, CA: Morgan Kaufmann Publishers, 1986. - Vol. 2. – Pp. 593-623.
121. Hramov A.E. Wavelets in neuroscience / A.E. Hramov, A.A. Koronovskii, V.A. Makarov, V.A. Maksimenko, A.N. Pavlov, E. Sitnikova. – 2nd ed. – Switzerland : Springer, 2021. – 384 p.
122. Ishibuchi H. A hybrid fuzzy genetics-based machine learning algorithm: hybridization of Michigan approach and Pittsburgh approach / H. Ishibuchi, T. Nakashima and T. Kuroda // 1999 IEEE International Conference on Systems, Man, and Cybernetics. – Tokyo, Japan. – 1999. – Vol.1. – Pp. 296-301.
123. Jang J.-S. R. ANFIS: adaptive-network-based fuzzy inference system / J.-S. R. Jang // IEEE Transactions on Systems, Man, and Cybernetics. – 1993. – Vol. 23. – No. 3. – Pp. 665-685.
124. Jia R. Fuzzy System Based on Class Association Rules / R. Jia, Z. Yibo // Fourth International Conference on Fuzzy Systems and Knowledge Discovery. – Haikou, China. – 2007. – Pp. 155-159.
125. Johansson U. Rule extraction from opaque models – a slightly different perspective / U. Johansson, T. Lofstrom, R. Konig, C. Sonstrod, L. Niklasson // 5th International Conference on Machine Learning and Applications. – Orlando, FL, USA. – 2006. – Pp. 22–27.
126. Juang C.-F. Evolutionary interval type-2 fuzzy systems using continuous ant colony optimization algorithms / C.-F. Juang and C.-W. Hung // 2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA). – Hefei, China. – 2016. – Pp. 26-31.

127. Khodashinskii I. A. Identification of fuzzy systems using a continuous ant colony algorithm / I. A. Khodashinskii, P. A. Dudin // Analysis and Synthesis of Signals and Images. – 2012. – Vol. 48. – Pp. 54-61.
128. Klement E.P. Triangular norms. Position paper I: basic analytical and algebraic properties / E.P. Klement, R. Mesiar, E. Pap // Fuzzy Sets and Systems. – 2004. – Vol. 143. – No 1. – Pp. 5-26.
129. Klement E.P. Triangular norms. Position paper III: continuous t-norms / E.P. Klement, R. Mesiar, E. Pap // Fuzzy Sets and Systems. – 2004. – Vol. 145. – No 3. – Pp. 439-454.
130. Kohonen T. Self-organized formation of topologically correct feature maps / T. Kohonen // Biological Cybernetics. – 1982. – Vol. 43. – Pp. 59-69.
131. Koronovskii A.A. Coexistence of intermittencies in the neuronal network of the epileptic brain / A.A. Koronovskii, A. E. Hramov, V.V. Grubov, O.I. Moskalenko, E. Sitnikova, A.N. Pavlov // Physical Review E. – 2016. – Vol. 93. – P. 032220.
132. Kramosil I. Fuzzy Metrics and Statistical Metrics Spaces / I. Kramosil, J. Michálek // Kybernetika. – 1975. – Vol. 11. – No 5. – Pp. 336-344.
133. Kulis B. Metric Learning: A Survey / B. Kulis // Foundations and Trends in Machine Learning. – 2013. – Vol. 5. – No. 4. – Pp 287-364.
134. Kumar P. Minimum-volume enclosing ellipsoids and core sets / P. Kumar, E. Yildirim // Journal of Optimization Theory and Applications. – 2005. – Vol. 126. – No. 1. – Pp. 1-21.
135. Ledeneva T. A parametric Family of Triangular Norms and Conorms with an Additive Generator in the Form of an Arctangent Linear Fractional Function / T. Ledeneva // Computation. – 2023. – Vol. 11. – No 8. – Pp. 155.
136. Ledeneva T. Additive generators of fuzzy operations in the form of linear fractional functions / T. Ledeneva // Fuzzy Sets and Systems. – 2020. – Vol. 386. – Pp. 1-24.
137. Ledeneva T. Generation of a Rule Base Based on Clustering of a Training Set / T. Ledeneva, T. Moiseeva // 2024 6th International Conference on Control Systems,

Mathematical Modeling, Automation and Energy Efficiency (SUMMA), Lipetsk, Russian Federation, 2024. – Pp. 102-107.

138. Ledeneva T. New Family of Triangular Norms for Decreasing Generators in the Form of a Logarithm of a Linear Fractional Function / T. Ledeneva // Fuzzy Sets and Systems. – 2022. – Vol. 427. – Pp. 37-54.

139. Ledeneva T. On One Approach to Constructing a Fuzzy Metric / T. Ledeneva, T. Moiseeva // 5th International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA), Lipetsk, Russian Federation, 2023. – Pp. 175-179.

140. Ledeneva T. Special Aspects of the Fuzzy Inference Mechanism / T. Ledeneva // Proceedings – 2020 2nd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency, SUMMA 2020. – 2020. – Pp. 128-132.

141. Lemos A. Multivariable Gaussian Evolving Fuzzy Modeling System / A. Lemos, W. Caminhas and F. Gomide // IEEE Transactions on Fuzzy Systems. – 2011. – Vol. 19. – No. 1. – Pp. 91-104.

142. Li M. Construction and Optimization of Fuzzy Rule-Based Classifier with a Swarm Intelligent Algorithm / M. Li, Q. Chen, J. Sun // Mathematical Problems in Engineering. – 2020. – Id. 319364.

143. Li Z. G. The Generation of the Fuzzy Control Rules Based on Association Rules with Temporal Constraints / Z. G. Li, F. L. Yang // Applied Mechanics and Materials. – 2013. – Vol. 385-386. – Pp. 931-934.

144. Lloyd S. Least squares quantization in PCM / S. Lloyd // IEEE Transactions on Information Theory. – 1982. – Vol. 28. – No. 2. – Pp. 129-137.

145. Lughofer E. D. FLEXFIS: A Robust Incremental Learning Approach for Evolving Takagi–Sugeno Fuzzy Models / E. D. Lughofer // IEEE Transactions on Fuzzy Systems. – 2008. – Vol. 16. – No. 6. – Pp. 1393-1410.

146. Luong P. An Evolving Fuzzy Classifier for Induction Motor Health Condition Monitoring / P. Luong, W. Wang // Intelligent Control and Automation. – 2019. – Vol. 10. – Pp. 129-141.

147. MacQueen J. Some methods for classification and analysis of multivariate observations / J. MacQueen // Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. – 1967. – Vol. 1. – Pp. 281–297.
148. Magnus J. R. Matrix differential calculus with applications in statistics and econometrics / J. R. Magnus, H. Neudecker. – John Wiley & Sons, 1988. – 504 p.
149. Metric space – Wikipedia [Electronic resource]. – Mode of access: [https://en.wikipedia.org/wiki/Metric\\_space](https://en.wikipedia.org/wiki/Metric_space).
150. Miah A.S.M. EEG classification for MI-BCI using CSP with averaging covariance matrices: An experimental study / A.S.M. Miah, M.R. Islam, M.K.I. Molla // Proceedings of the 2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering IEEE, Rajshahi. – Bangladesh. – 2019. – Pp. 1-5.
151. Miñana J.-J. A duality relationship between fuzzy metrics and metrics / J.-J. Miñana, O. Valero // International Journal of General Systems. – 2018. – Vol. 47. – No 6. – Pp. 593-612.
152. Moiseeva T. Knowledge Base Generation Based on Fuzzy Clustering / T. Moiseeva, T. Ledeneva // Programming and Computer Software, 2023. – Vol. 49. – No. 1. – Suppl. 2. – Pp. 99-107.
153. Moiseeva T. Missing Data Imputation Using Fuzzy System / T. Moiseeva, T. Ledeneva // 4th International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA), Lipetsk, Russian Federation, 2022. – Pp. 350-354.
154. Morillas S. A New Vector Median Filter Based on Fuzzy Metrics / S. Morillas, V. Gregori, G. Peris-Fajarnés, P. Latorre // Image Analysis and Recognition. ICIAR 2005. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg. – 2005. - Vol 3656.
155. Narici L. Topological Vector Space / L. Narici, E. Beckenstein. – 2nd ed. – NY: CRC Press, 2010. – 628 p.
156. Nielsen F. Introduction to HPC with MPI for Data Science / F. Nielsen. – Switzerland : Springer Cham, 2016. – 282 p.

157. Pinheiro D. N. Convex fuzzy k-medoids clustering / D. N. Pinheiro, D. Aloise, S. J. Blanchard // Fuzzy Sets and Systems. – 2020. – Vol. 389. – Pp. 66-92.
158. Prakash R. V. Fuzzy Association Rules and Its Applications / R. V. Prakash, SSVN. Sarma // Advanced Pattern Mining and Multimedia Computing. – February. – 2010.
159. Ralević N. Fuzzy metric space and applications in image processing / N. Ralević, M. Paunović, B. Iricanin // Mathematica Montisnigri. – 2020. – Vol.48. – Pp. 103-117.
160. Rangwala M. Extracting Rules from Artificial Neural Networks Utilizing TREPAN / M. Rangwala, G.R. Weckman // Proceedings of IISE Annual Conference. – Orlando, Florida. – 2006. – 7 p.
161. Raposo A. A. M. A construction of the minimum volume ellipsoid containing a set of points using BRKGA metaheuristic / A. A. M. Raposo, V. M. de Souza, L. R. A. G. Filho // Proceeding Series of the Brazilian Society of Computational and Applied Mathematics. – 2021. – Vol. 8. – №. 1. – Pp. 010339
162. Rehman A.U. Unsupervised outlier detection in multidimensional data / A.U. Rehman, S.B. Belhaouari // Journal of Big Data. – 2021. – Vol. 8. – Article 80.
163. Řezanková H. Different approaches to the silhouette coefficient calculation in cluster validation / H. Řezanková // 21st International Scientific Conference AMSE Applications of Mathematics and Statistics in Economics. – Kutná Hora, Czech Republic. – 2018. – Pp. 259-268.
164. Saha S.K. Motor Imagery EEG Signal Classification Using MWT-CSP for Online BCI Implementation / S. K. Saha, P. K. Sarker, A. Al Shiam, Md. M. Rahoman // International Journal of Computer Science and Information Security. – 2020. – Vol. 18. – Pp. 124-130.
165. Saroj Dr. A Genetic-Fuzzy Algorithm to Discover Fuzzy Classification Rules for Mixed Attributes Datasets / Dr. Saroj, N. Prabhat // International Journal of Computer Applications. – 2011. – Vol. 34. – No. 5. – Pp. 15-22.

166. Sasu L. Function approximation with ARTMAP architectures / L. Sasu, R. Andonie // International Journal of Computers, Communications & Control. – 2012. – Vol. 7. – Pp. 957-967.
167. Sato M. Rule Extraction from Neural Networks via Decision Tree Induction / M. Sato, H. Tsukimoto // Proc. Intern. Joint Conf. on Neural Networks. – Washington, DC. – 2001. – Vol. 3. – Pp. 1870-1875.
168. Setiono R. FERN: An algorithm for fast extraction of rules from neural networks / R. Setiono, W. K. Leow // Applied Intelligence. – 2000. – Vol. 12. – Pp. 15-25.
169. Sheikh R. H. Genetic Algorithm Based Clustering: A Survey / R. H. Sheikh, M. M. Raghuwanshi, A. N. Jaiswal // 2008 First International Conference on Emerging Trends in Engineering and Technology. – Nagpur, India. – 2008. – Pp. 314-319.
170. Shen Y. Fixed point theorems in fuzzy metric spaces / Y. Shen, D. Qiu, W. Chen. // Applied Mathematics Letters. – 2012. – Vol.25. – Pp. 138-141.
171. Smartphone Accelerometer Dataset [Electronic resource]. – Mode of access: <https://github.com/ngjiawaie/Smartphone-Accelerometer-Dataset>
172. Smith S. F. Flexible Learning of Problem Solving Heuristics Through Adaptive Search / S. F. Smith // Proceedings of the 8th International Joint Conference on Artificial Intelligence. – Karlsruhe, FRG. – August 1983. – Pp. 422-425.
173. Sun P. Computation of minimum-volume enclosing ellipsoids / P. Sun, R. Freund // Oper. Res. – 2004. – Vol. 52. – No. 5. – Pp. 690-706.
174. Thrun S. Extracting rules from artificial neural networks with distributed presentations / S. Thrun // Advances in Neural Information Processing Systems / Eds. G. Tesauro, D. Touretzky, T. Leen. – Cambridge, MA : MIT Press, 1995. – Vol. 7.
175. Taha I. Three Techniques for extracting rules from feedforward network / I. Taha, J. Ghosh // Intelligent Engineering Systems Through Artificial Neural Networks. – 1996. – Vol. 6. – Pp. 23-28.
176. Todd M. J. On Khachiyan's algorithm for the computation of minimum volume enclosing ellipsoids / M. J. Todd, E. A. Yıldırım // Discrete Appl. Math. – 2007. – Vol. 155. – Pp. 1731-1744.

177. Towell G. G. Extracting refined rules from knowledge based neural networks / G. G. Towell, J. W. Shavlik // Machine learning. – 1993. – Vol. 13. – No. 1. – Pp. 71-101.
178. Tsukimoto H. Extracting rules from trained neural networks / H. Tsukimoto // IEEE Transactions on Neural Networks. – March 2000. – Vol. 11. – No. 2. – Pp. 377-389.
179. Ullah A. Genetic optimization of fuzzy membership functions for cloud resource provisioning / A. Ullah, J. Li, A. Hussain, Yi. Shen // 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece. – 2016. – Pp. 1-8.
180. van Luijtelaar G. Methods of automated absence seizure detection, interference bystimulation, and possibilities for prediction in genetic absence models / G. van Luijtelaar, A. Lüttjohann, V.V. Makarov, V.A. Maksimenko, A.A. Koronovskii, A.E. Hramov // Journal of Neuroscience Methods. – 2016. – Vol. 260. – Pp. 144-158.
181. Wang GG. A comprehensive review of krill herd algorithm: variants, hybrids and applications / GG. Wang, A.H. Gandomi, A.H. Alavi, D. Gong // Artificial Intelligence Review. – 2019. – Vol. 51. – Pp. 119–148.
182. Wang L.-X. Fuzzy systems are universal approximators / L.-X. Wang // IEEE 1992 Intern. Conf. Fuzzy Systems. – San Diego, 1992. – P. 1163-1170.
183. Wang L. X. Generating Fuzzy Rules by Learning from Examples / L. X. Wang, J. M. Mendel // IEEE Transactions on Systems, Man, and Cybernetics. – 1992. – Vol. 22. – No. 6. – Pp. 1414-1427.
184. Wu X. Answering an open question in fuzzy metric spaces / X. Wu, G. Chen // Fuzzy Sets and Systems. – 2020. – Vol. 390. – Pp. 188-191.
185. Xie X.L. A Validity Measure for Fuzzy Clustering / X.L. Xie, G. Beni // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 1991. – Vol. 13. – No 4. – Pp. 841-847.
186. Yager R.R. Families of OWA operators / R.R. Yager // Fuzzy Sets and Systems. – 2004. – Vol. 59. – No. 2. – Pp. 125-148.

187. Zahn C. T. Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters / C. T. Zahn // IEEE Transactions on Computers. – 1971. – Vol. C-20. – No. 1. – Pp. 68-86.
188. Zarghami M. Revising the OWA operator for multi criteria decision making problem under uncertainty / M. Zarghami, F. Szidarovszky // European Journal of Operational Research. – 2009. – Vol. 198. – Pp. 259–265.
189. Zhang Y. How does Contrastive Learning Organize Images? / Y. Zhang, Y. Lu, Q. Xuan // 2024 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW). – Waikoloa, HI, USA. – 2024. – Pp. 497-506.
190. Zhao P. Analysis of KNN Density Estimation / P. Zhao, L. Lai // IEEE Transactions on Information Theory. – 2022. – Vol. 68. – No. 12. – Pp. 7971-7995.

## ПРИЛОЖЕНИЕ А. АКТЫ О ВНЕДРЕНИИ

УТВЕРЖДАЮ

Декан факультета прикладной математики, информатики и механики

*С.Н. Медведев*  
Медведев С. Н.  
«10» 02 2025 г.



**АКТ**  
**о внедрении (использовании) результатов**  
**кандидатской диссертационной работы**  
**Моисеевой Татьяны Александровны**

Комиссия в составе:

председатель к.ф.-м.н., доц. С.Н. Медведев

члены комиссии: д.т.н., проф. Ю.В. Бондаренко, к.ф.-м.н., доц. Е.М. Аристова

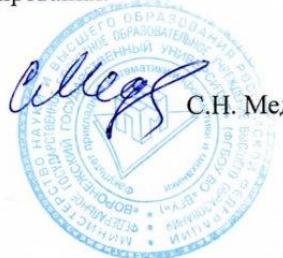
составили настоящий акт о том, что результаты диссертационной работы Моисеевой Татьяны Александровны «Методы генерации баз знаний нечетких производственных систем с использованием процедур кластеризации», представленной на соискание ученой степени кандидата технических наук, использованы в учебном процессе кафедры вычислительной математики и прикладных информационных технологий факультета прикладной математики, информатики и механики при проведении лекционные и практических занятий по дисциплинам «Инженерия знаний и проектирование баз знаний», «Методы нечеткого моделирования», «Интеллектуальные информационные технологии», а также при выполнении выпускных квалификационных работ.

Результаты диссертационной работы Моисеевой Татьяны Александровны:

- метод формирования баз нечетких производственных правил на основе эллипсоидальной кластеризации, отличающийся использованием эллипсоидов минимального объема;
- семейство нечетких метрик, впервые полученных на основе аддитивных генераторов непрерывных архимедовых строгих треугольных норм из класса рациональных функций;
- метод формирования комплексной оценки качества кластеризации, основанный на использовании специальных функций агрегирования.

Председатель комиссии:

Декан факультета прикладной математики, информатики и механики, к.ф-м.н, доцент



С.Н. Медведев

Члены комиссии:

Председатель научно-методического совета факультета ПММ, д.т.н., проф. кафедры математических методов исследования операций

Ю.В. Бондаренко

Зам. зав. кафедрой вычислительной математики и прикладных информационных технологий, к.ф-м.н., доцент

Е.М. Аристова



**Общество с ограниченной ответственностью  
"ФИНАНСОВАЯ ПРОМЫШЛЕННАЯ КОМПАНИЯ "АЛЬФА"**

---

**адрес:** 394036, Воронежская область, город Воронеж, улица Сакко и Ванцетти, дом 61, офис 1  
**tel.:** +7 473 2222 566; +7 952 950 8676    **e-mail:** [fpk-alfa@mail.ru](mailto:fpk-alfa@mail.ru)    **web:** <http://fpk-alfa.ru/>

---

**АКТ о внедрении**

Настоящим подтверждаем, что результаты диссертационного исследования Моисеевой Т.А., а именно: алгоритм построения базы знаний на основе эллипсоидальной кластеризации; модель нечеткого классификатора для классификации электроэнцефалограмм; программный комплекс «FuzzyLogicCore» были использованы при разработке экспертной системы для анализа нейромаркетинговых данных.

Директор

ООО «ФПК «Альфа»

/ А.О. Шамрин



ООО "ФПК "АЛЬФА"

Юр. адрес: 394036, город Воронеж, улица Сакко и Ванцетти, дом 61, офис 1

ОГРН 1133668040923, ИНН 3666187170, КПП 366601001

Р/сч: 40702810202940001344 в ОАО "АЛЬФА-БАНК" г.Москва; БИК: 044525593; Кор/счет: 30101810200000000593

10.02.25

«УТВЕРЖДАЮ»

Лаборатория медицинской кибернетики  
ФГБОУ ВО «Воронежский государственный университет»  
394018, Россия, г. Воронеж, Университетская площадь, 1  
зав. лабораторией Туровский Я.А.

10.02.2025

АКТ

о внедрении результатов диссертационного исследования  
на соискание ученой степени кандидата технических наук  
Моисеевой Татьяны Александровны

Комиссия в составе председателя комиссии:

Туровского Ярослава Александровича, д-ра техн. наук, доцента, зав. лабораторией  
медицинской кибернетики

членов комиссии: Борзунова Сергея Викторовича, канд. физ.-мат. наук, доцента

Вахтина Алексея Александровича, канд. физ.-мат. наук, доцента

составила настоящий акт о том, что научные результаты кандидатской диссертации  
Моисеевой Т.А., включающие

- алгоритмы предобработки данных энцефалограмм,
- алгоритм построения базы знаний нечеткого классификатора энцефалограмм для асинхронного интерфейса «мозг-компьютер»,
- программный комплекс, реализующий перечисленные алгоритмы

используются для информационного обеспечения научно-исследовательской работы,  
проводимой лабораторией по созданию человека-машинных интерфейсов.

Председатель комиссии:

Туровский Я.А.

Члены комиссии: Борзунов С.В.

Вахтин А.А.

10.02.2025

ПРИЛОЖЕНИЕ Б. СВИДЕТЕЛЬСТВА  
О ГОСУДАРСТВЕННОЙ РЕГИСТРАЦИИ ПРОГРАММ ДЛЯ ЭВМ

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2024661362

**Программа для выполнения кластеризации методом  
нечетких к-медоид с использованием нечеткой метрики**

Правообладатель: *федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Воронежский государственный университет» (RU)*

Автор(ы): *Моисеева Татьяна Александровна (RU)*

Заявка № 2024660510

Дата поступления 13 мая 2024 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 17 мая 2024 г.

Руководитель Федеральной службы  
по интеллектуальной собственности

Ю.С. Зубов



РОССИЙСКАЯ ФЕДЕРАЦИЯ



# СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

**№ 2024661436**

## Программа для генерации нечеткой системы для решения задачи аппроксимации

Правообладатель: *федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Воронежский государственный университет» (RU)*

Автор(ы): *Моисеева Татьяна Александровна (RU)*

Заявка № **2024660514**

Дата поступления **13 мая 2024 г.**

Дата государственной регистрации

в Реестре программ для ЭВМ **17 мая 2024 г.**

Руководитель Федеральной службы  
по интеллектуальной собственности

*Ю.С. Зубов*

