Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра программного обеспечения
информационных технологий

**Л. В. Серебряная, В. В. Потараев, Е. П. Фадеева**

# *МОДЕЛИ И МЕТОДЫ ОБРАБОТКИ ДАННЫХ В ИНФОРМАЦИОННЫХ СИСТЕМАХ*

*Рекомендовано УМО по образованию в области информатики и радиоэлектроники в качестве учебно-методического пособия для специальностей 1-40 80 05 «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей», 1-40 80 05 «Программная инженерия»*



Минск БГУИР 2019

УДК 519.254(076)
ББК 32.973.3я73
С32

Р е ц е н з е н т ы:

кафедра интеллектуальных систем
Белорусского национального технического университета
(протокол №4 от 15.11.2018);

старший научный сотрудник государственного научного учреждения «Объединенный институт проблем информатики Национальной академии наук Беларуси» кандидат физико-математических наук С. В. Чебаков

**Серебряная, Л. В.**

С32 Модели и методы обработки данных в информационных системах : учеб.-метод. пособие / Л. В. Серебряная, В. В. Потараев, Е. П. Фадеева. – Минск : БГУИР, 2019. – 67 с. : ил.

ISBN 978-985-543-506-9.

Рассмотрены различные типы и особенности функционирования современных информационных систем. Приведены модели и методы работы с данными, используемые в информационных системах. Предложены шесть лабораторных работ, реализация которых позволит практически закрепить знания по курсу «Модели и методы обработки данных в информационных системах».

**УДК 519.254(076)**
**ББК 32.973.3я73**

**ISBN 978-985-543-506-9**

**СОДЕРЖАНИЕ**

# ВВЕДЕНИЕ

Информационная система (ИС) представляет собой коммуникационную систему по сбору, передаче, переработке информации об объекте, снабжающую работников различных рангов информацией для реализации функции управления. Внедрение информационных систем выполняется с целью повышения эффективности производственно-хозяйственной деятельности объекта за счет обработки и хранения информации, автоматизации операций, а также за счет принципиально новых методов управления. Неотъемлемой частью ИС является информационное обеспечение (ИО).

Предметом рассмотрения данного учебно-методического пособия является понятие «информация», а также методы и алгоритмы ее обработки. Информацию можно определить как совокупность сведений, уменьшающих степень неопределенности знания о конкретных событиях, процессах, явлениях и т. п. В зависимости от сферы использования информация может быть самой разной: экономической, технической, генетической и т. д. Информация в системе управления рассматривается как «ресурс управления», имеющий важное стратегическое значение. И хотя информационные ресурсы в значительной степени являются взаимозаменяемыми по отношению к материальным, финансовым или трудовым ресурсам, организационная форма инфоресурсов, объем и качество информации напрямую влияют на эффективность управления и качество решений, принимаемых в информационной системе.

Основное назначение ИО состоит в создании динамической информационной модели объекта, отражающей его состояние в текущий и предшествующий момент времени.

Хранение больших объемов информации практически оправдано только при условии, если ее поиск и обработка осуществляется быстро и выдается она в доступной для понимания форме. Предназначенные для этого программные средства должны учитывать структуру информационного фонда и физические свойства запоминающей среды.

Возможности электронно-вычислительной техники в последние десятилетия значительно расширились: при ощутимом увеличении ее способности хранить большие объемы информации стоимость хранения постоянно сокращается. При этом развитие программных средств, связанных с организацией и поиском данных в «электронных хранилищах», происходило не такими быстрыми темпами. Поэтому специалисты, разрабатывающие программы для информационных систем, находятся в положении, когда они вынуждены все время успевать за новыми возможностями вычислительной техники.

# ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Пользователями ИС являются организационные единицы управления: структурные подразделения, управленческий персонал, исполнители. Содержательную основу ИС составляют функциональные компоненты: модели, методы и алгоритмы формирования управляющей информации. Функциональная структура ИС представляет собой совокупность функциональных компонентов: подсистем, комплексов задач, процедур обработки информации, определяющих последовательность и условия их выполнения.

Внедрение информационных систем производится с целью повышения эффективности производственно-хозяйственной деятельности объекта за счет обработки и хранения информации, автоматизации операций, а также за счет принципиально новых методов управления. Они основаны на моделировании действий специалистов организации в ходе принятия решений, использовании современных средств телекоммуникаций, глобальных и локальных вычислительных сетей.

## 1 Структура и состав информационной системы

Практически все разновидности ИС независимо от сферы их применения включают в себя один и тот же набор компонентов:

- функциональные компоненты;
- компоненты системы обработки данных;
- организационные компоненты.

Под функциональными компонентами понимается система функций управления, взаимосвязанных во времени и пространстве и необходимых для достижения поставленных целей. Декомпозиция ИС по функциональному признаку (рисунок 1) включает в себя выделение ее отдельных частей, называемых функциональными подсистемами. Функциональный признак определяет назначение подсистемы, ее цели, задачи и функции, которые она выполняет. Функциональные подсистемы в большой степени зависят от предметной области ИС.

Практически все ИС независимо от сферы их применения имеют один и тот же набор составных частей (компонентов), называемых видами обеспечения (см. рисунок 1). Принято выделять информационное, программное, техническое, правовое, лингвистическое обеспечение.

*Информационное обеспечение* – это совокупность методов и средств по размещению и организации информации, включающих в себя системы классификации и кодирования, унифицированные системы документации, рационализации документооборота и форм документов, методов создания внутримашинной информационной базы ИС.

*Программное обеспечение* (ПО) – совокупность программных средств для создания и эксплуатации системы обработки данных (СОД) с помощью вычислительной техники. В состав ПО входят базовые и прикладные программные средства. Базовые программные средства служат для автоматизации взаимодействия человека и компьютера, организации типовых процедур

обработки данных и диагностики функционирования технических средств СОД. Прикладные программные средства представляют собой совокупность программных продуктов, предназначенных для автоматизации решения функциональных задач ИС.


Рисунок 1 – Декомпозиция информационной системы

*Техническое обеспечение* представляет собой комплекс технических средств, применяемых для функционирования СОД, и включает в себя устройства, реализующие типовые операции обработки данных как вне компьютера, так и внутри него.

*Правовое обеспечение* – это совокупность правовых норм, регламентирующих создание и функционирование ИС. Правовое обеспечение разработки ИС включает в себя нормативные акты договорных взаимоотношений между заказчиком и разработчиком ИС, правовое регулирование отклонений от договора.

*Лингвистическое обеспечение* – это совокупность языковых средств, используемых на различных стадиях создания и эксплуатации СОД для повышения эффективности разработки и обеспечения общения человека и компьютера.

## 2 Системы классификации данных

Для того чтобы обеспечить эффективную обработку информации на компьютере, ее сначала классифицируют. Классификация информации необходима для ее правильной структуризации и составления в дальнейшем корректной модели базы данных.

Классификация – это разделение заданного множества на подмножества в соответствии с принятыми методами классификации. Система классификации – это совокупность методов и средств, с помощью которых осуществляется разбиение исходного множества данных на пересекающиеся или непересекающиеся подмножества в соответствии с признаками сходства и различия.

Признак, по которому делят множества на подмножества, называют признаком классификации. Ступень классификации – этап разделения заданного множества на подмножества. Число ступеней называют глубиной классификации. После завершения классификации выполняют кодирование – образование и присвоение обозначения объекту классификации, признаку классификации и классификационной группировке. Каждая система классификации характеризуется следующими свойствами: гибкостью, емкостью, степенью заполнения системы.

*Гибкость* системы – это способность допускать включение новых признаков, объектов без разрушения структуры классификатора. Необходимая гибкость определяется временем жизни системы.

*Емкость* системы – это наибольшее количество классификационных группировок, допускаемое в данной системе классификации.

*Степень заполнения* системы определяется как частное от деления фактического количества группировок на величину емкости системы.

Рассмотрим основные типы систем классификации.

***Иерархическая классификация*** подразумевает разделение всего множества объектов на группы, которые формируются в виде отдельных уровней или дерева. Количество уровней иерархии – это глубина классификации.

Первоначальный объем классифицируемых объектов разбивается на подмножества по какому-либо признаку и детализируется на каждой следующей ступени классификации. На рисунке 2 приведена обобщенная схема иерархической классификации.

Характерными особенностями иерархической системы являются:

– возможность использования неограниченного количества признаков классификации;

– подчиненность признаков классификации, которая выражается в разбиении каждой классификационной группировки, образованной по одному признаку, на множество группировок по нижестоящему признаку.

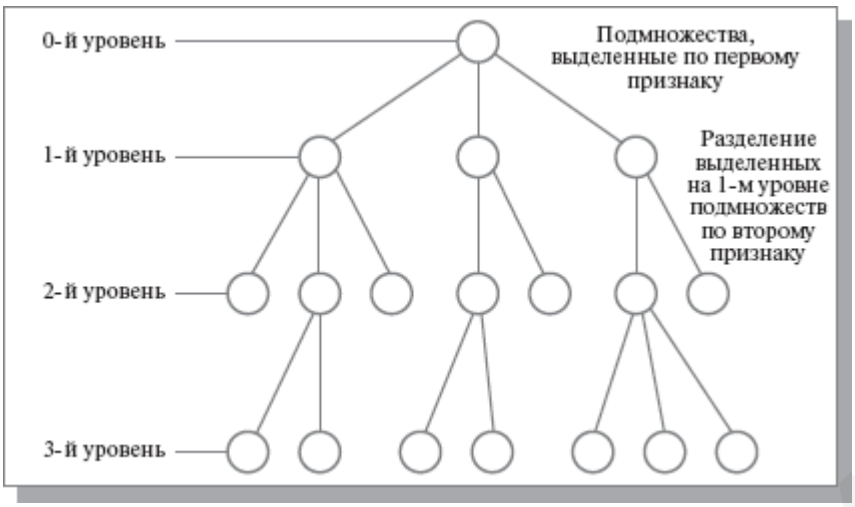



Рисунок 2 – Схема иерархической классификации

Таким образом, классификационные схемы, построенные на основе иерархического принципа, имеют неограниченную емкость, величина которой зависит от глубины классификации и количества объектов классификации, которые можно расположить на каждой ступени. Выбор необходимой глубины классификации зависит от характера объектов классификации и задач, для решения которых предназначен классификатор. При построении иерархической системы классификации сначала выделяется некоторое множество объектов, подлежащее классифицированию, для которого определяются полное множество признаков и их подчиненность друг другу. Затем выполняется разбиение исходного множества объектов на классификационные группировки каждой ступени классификации.

К положительным особенностям иерархической системы относят логичность, простоту построения и удобство обработки.

Серьезным недостатком иерархического метода классификации является жесткость классификационной схемы. Она обусловлена заранее установленным выбором признаков классификации и порядком их использования по ступеням классификации. Это ведет к тому, что при изменении состава объектов классификации, их характеристик или характера решаемых при помощи классификатора задач требуется коренная переработка классификационной схемы. Гибкость данной системы обеспечивается только за счет ввода большой избыточности в ветвях, что приводит к слабому заполнению структуры классификатора. Поэтому при разработке классификаторов следует учитывать, что иерархический метод более предпочтителен для решения неменяющегося комплекса задач, где присутствуют объекты со слабо изменяющимися признаками.

***Многоаспектная классификация*** – это система, которая использует параллельно несколько независимых признаков (аспектов) в качестве основания классификации. Существуют два типа многоаспектных систем: фасетная и дескрипторная. Фасет – это аспект классификации, который используется для образования независимых классификационных группировок.

Дескриптор – ключевое слово, определяющее некоторое понятие, которое формирует описание объекта для его отнесения к классу или группе.

При фасетном методе классификации заранее не создаются жесткая классификационная схема и конечные группировки. Разрабатывается только система таблиц признаков объектов классификации, называемых фасетами. При необходимости создания классификационной группировки для решения конкретной задачи осуществляется выборка необходимых признаков из фасетов и их объединение в определенной последовательности. Общий вид фасетной классификационной схемы представлен на рисунке 3. Внутри фасета значения признаков могут перечисляться в заданном порядке или образовывать иерархическую структуру, если существует подчиненность выделенных признаков.

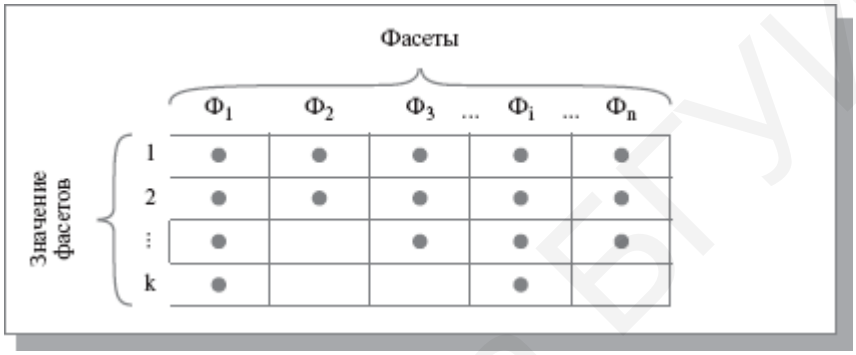



Рисунок 3 – Схема признаков фасетной классификации

Для выполнения многоаспектной дескрипторной классификации выделяется набор ключевых слов. Каждый объект характеризуется некоторой последовательностью ключевых слов и входит в соответствующую группу.

К преимуществам многоаспектной системы относят ее большую емкость и высокую степень гибкости, поскольку при необходимости можно вводить дополнительные фасеты и изменять их место в системе. При изменении характера задач или характеристик объектов классификации разрабатываются новые фасеты или дополняются новыми признаками уже существующие без коренного изменения структуры всего классификатора.

К недостаткам многоаспектной системы можно отнести сложность ее структуры и низкую степень заполнения системы.

В современных классификационных схемах часто одновременно используются оба метода классификации. Это снижает влияние недостатков методов классификации и расширяет возможность использования классификаторов в информационном обеспечении управления.

## 3 Системы кодирования

Кодирование информации – это процедура создания специальных кодов и их использования для уменьшения объема информационной базы.

Система кодирования – это совокупность правил для составления кодов. При работе ИС обычно большую часть ее информации составляет текст.

Поэтому для облегчения работы и исключения избыточности информации широко используются системы кодирования данных, в которых текстовая информации заменяется более короткими кодами. Кодирование информации предусматривает следующие действия:

– приведение к единообразию в обозначениях признаков, характеристик и объектов в целом;

– упорядочение, классификацию и группировку всех номенклатур по определенным сходным признакам;

– выбор системы кодирования и присвоение кодов;

– приведение информации к форме, удобной для обработки с помощью технических средств.

Регистрационные системы кодирования, которые делятся на порядковые и серийные, могут быть использованы без предварительной классификации. В порядковых системах каждый объект получает свой порядковый номер. Недостатком является то, что вновь поступающий объект помещается в конец списка. В серийных системах для каждой группы объектов выделяется группа кодов, обычно с запасом. Можно обеспечить добавление данных в каждую группу.

Классификационные системы кодирования основываются на предварительной классификации информации. Для использования последовательных систем сначала нужно выполнить иерархическую классификацию. Общий код объекта формируется при помощи включения кодов всех предыдущих уровней иерархии. При этом на каждом уровне иерархии может быть использована либо порядковая, либо серийная система кодирования, но чаще – порядковая. Для формирования кода объекта берутся коды всех вышестоящих уровней и дописываются от высшего к низшему. По коду всегда можно судить о принадлежности объекта к определенному классу, виду, подвиду и т. д. Параллельные системы основаны на фасетной классификации, где все объекты распределены по группам в соответствии с признаками классификации, и каждая группа имеет свой код.

При формировании любой базы данных обычно все наименования кодируются и составляются классификаторы или справочники. Кроме наименования, в него могут быть включены дополнительные сведения об объекте. В дальнейшем вместо наименования используется код, и работа выполняется только с ним.

Технология подготовки внемашинного ИО служит основой для формирования внутримашинного ИО. Наиболее важными вопросами для него являются:

– определение состава документов, содержащих информацию, необходимую для решения задач;

– определение структурных единиц информации и их взаимосвязей;

– классификация и кодирование информации, обрабатываемой в задачах;

– разработка инструкций и методик по ведению документов.

Документ, с помощью которого осуществляется формализованное описание информации в ИС, содержащей наименования объектов, классификационных группировок и их кодовые обозначения, называется *классификатором*. Для большей надежности системы классификации и кодирования дополняют защитными кодами, обеспечивающими контроль достоверности информации на входе и на выходе. При разработке классификаторов и систем кодирования рекомендуется соблюдать следующие требования:

1) *выбор кодов минимальной длины.* Это приводит к уменьшению количества ошибок при переносе информации на машинные носители и сокращению трудоемкости их обработки;

2) *логичность и запоминаемость кодов*. Выполнение данного требования помогает при освоении кодов, облегчает кодирование и уменьшает число допускаемых ошибок;

3) *учет особенностей решаемых задач*;

4) *учет существующей системы кодирования и общепринятых обозначений*;

5) *учет перспектив развития*. Коды должны составляться таким образом, чтобы обеспечить возможности изменения и резерв на случай появления новых объектов в системе;

6) *необходимость информационной стыковки*, которая позволяет интегрировать различные объекты ИС и работать с ними сообща.

## 4 Интеллектуальные информационные системы

Развитием обычных информационных систем явились интеллектуальные информационные системы (ИИС). Они воплотили в себе наукоемкие технологии с высоким уровнем автоматизации подготовки информации для принятия решений, а также автоматизации процессов нахождения вариантов решений, опирающихся на полученные данные.

ИИС можно рассматривать как технические и программные системы, ориентированные на решение большого класса задач, называемых неформализованными. Система становится интеллектуальной, если в ней данные заменяются на знания, а алгоритмы функционирования заменяются на методы искусственного интеллекта.

Отличительными особенностями ИИС по сравнению с обычными ИС являются:

– интерфейс с пользователем на естественном языке с применением бизнес-понятий, характерных для прикладной области пользователя;

– способность объяснять свои действия и помогать пользователю в работе с системой;

– представление модели объекта в виде базы знаний в сочетании с возможностью работы с неполной или неточной информацией;

– способность автоматического обнаружения закономерностей бизнеса в ранее накопленных фактах и включения их в базу знаний.

ИИС особенно эффективны в применении к слабоструктурированным задачам, в которых пока отсутствует строгая формализация, и для решения которых применяются эвристические процедуры, позволяющие в большинстве случаев получить решение. Отчасти этим объясняется необычайно широкий диапазон применения ИИС.

Одной из главных проблем обработки знаний, а также создания базы знаний и ИИС в целом является проблема представления знаний. Это задача представления взаимосвязей в конкретной предметной области в форме, понятной системе искусственного интеллекта. Представление знаний – это их формализация и структурирование, с помощью которых отражаются характерные признаки знаний: внутренняя интерпретируемость, структурированность, связность, семантическая метрика и активность. Кроме того, представление знаний – это процесс описания знаний человека о проблемной области посредством выражений на формальном языке, называемом языком представления знаний.

При проектировании модели представления знаний следует учитывать такие факторы, как однородность представления и простота понимания. Однородность представления приводит к упрощению механизма управления логическим выводом и знаниями. Простота понимания предполагает доступность понимания представления знаний и экспертом, и пользователем системы. В противном случае затрудняется приобретение знаний и их оценка.

Способ представления знаний определяет, каким образом знания описываются в памяти компьютера, а также каковы возможности базы знаний. Для того чтобы ЭВМ имела возможность манипулирования знаниями о проблемной области, они должны быть представлены в виде модели.

Модель представления знаний – это способ и результат формального описания знаний в базе знаний. Она должна быть понятной пользователю и обеспечивать однородность представления знаний, за счет чего упрощаются управление знаниями и логический вывод.

Популярными моделями представления знаний являются искусственная нейронная и семантическая сети.

Тематика искусственных нейронных сетей (ИНС) относится к междисциплинарной сфере знаний, связанных с биокибернетикой, электроникой, прикладной математикой, статистикой, автоматикой, медициной и др. ИНС возникли на основе знаний о функционировании нервной системы живых существ. Они представляют собой попытку использования процессов, происходящих в нервных системах, для выработки новых технологических решений. Постоянно расширяется круг задач, для решения которых применяются ИНС. В этот круг входит и распознавание разнообразных образов, отличающихся по природе, сложности и другим признакам.

Современные ИНС по сложности и «интеллекту» постоянно растут и развиваются, демонстрируя ряд ценных свойств:

1 *Обучение*. ИНС могут менять свое поведение в зависимости от внешней среды. После предъявления входных сигналов, возможно вместе с требу-

емыми выходами, они самонастраиваются, чтобы обеспечить ожидаемую реакцию.

2 *Обобщение*. Отклик сети после обучения может быть до некоторой степени нечувствителен к небольшим изменениям входных сигналов. Важно, что ИНС делает обобщение автоматически благодаря своей структуре.

3 *Абстрагирование*. Если, например, предъявить сети несколько искаженных вариантов входного образа, то сеть сама сможет создать на выходе «идеальный» образ, с которым она никогда не встречалась.

Главным недостатком ИНС является длительный процесс их обучения с предварительным поиском объектов для обучения. Зачастую это не позволяет использовать их в системах реального времени.

Решение проблемы распознавания образов с помощью ИНС состоит из двух процедур: обучения и непосредственно самого распознавания незнакомых образов. Процедура поиска решения задачи с помощью сети, прошедшей обучение, оказывается более гибкой, чем использование других вычислительных средств, поскольку ИНС может повышать точность результатов по мере накопления ею опыта и адаптироваться к изменениям.

Семантические сети позволяют выделять смысл текста в виде понятий и связей между ними, образующих граф. Понятия семантической сети записываются в вершинах графа, а отношения между понятиями – это дуги графа. Количество типов отношений в семантической сети определяется ее разработчиком исходя из конкретных целей. Часто используются иерархические семантические сети, в которых отношения образуют древовидную структуру. Также отношения в сетях могут быть разных типов: функциональные, количественные, пространственные, временные, логические и др.

К достоинствам семантических сетей можно отнести следующее:

– универсальность, достигаемая за счет выбора соответствующего набора отношений;

– наглядность системы знаний, представленной графически;

– близость структуры сети, представляющей систему знаний, семантической структуре фраз на естественном языке;

– соответствие современным представлениям об организации долговременной памяти человека.

Недостатки семантических сетей:

– сетевая модель не всегда дает ясное представление о структуре предметной области, поэтому формирование и модификация такой модели могут быть затруднительными;

– сетевые модели представляют собой структуры, для обработки которых необходим специальный аппарат формального вывода;

– проблема поиска решения в семантической сети сводится к задаче поиска ее фрагмента, отражающего поставленный запрос. Это может обуславливать сложность поиска решения в семантических сетях;

– представление, использование и модификация знаний при описании систем реального уровня сложности оказывается трудоемкой процедурой, особенно при наличии множественных отношений между ее понятиями.

Семантические сети используются в системах понимания естественного языка, в вопросно-ответных системах, в других различных предметно-ориентированных системах. Чаще других востребованы сети смешанного типа, т. е. содержащие в зависимости от области применения самые разные типы отношений.

## 5 Генетический алгоритм

Развитие природных систем на протяжении многих веков привлекало внимание ученых. И только изучение эволюционных принципов и генетических основ наследственности позволило им разработать как модели молекулярной эволюции, описывающие динамику изменения молекулярных последовательностей, так и макроэволюционные модели, используемые в экологии, истории и социологии для исследования экосистем и сообществ организмов. Круг задач, решаемых с помощью генетического алгоритма (ГА), очень широк. Ниже перечислены некоторые задачи, для решения которых используются ГА:

- задачи численной оптимизации;
- задачи о кратчайшем пути;
- задачи компоновки;
- составление расписаний;
- аппроксимация функций;
- отбор (фильтрация) данных;
- настройка и обучение искусственной нейронной сети;
- биоинформатика;
- игровые стратегии;
- нелинейная фильтрация;
- развивающиеся агенты/машины.

На рисунке 4 приведена общая схема генетического алгоритма.

Генетический алгоритм работает с популяцией особей, в хромосоме (генотип) каждой из которых закодировано возможное решение задачи (фенотип). В начале работы алгоритма популяция формируется случайным образом (блок «Формирование начальной популяции» на рисунке 4). Для того чтобы оценить качество закодированных решений, используют функцию приспособленности, которая необходима для вычисления приспособленности каждой особи (блок «Оценивание популяции» на рисунке 4). По результатам оценивания особей наиболее приспособленные из них выбираются (блок «Селекция» на рисунке 4) для скрещивания. В результате скрещивания выбранных особей посредством применения генетического оператора кроссинговера создается потомство, генетическая информация которого формируется в результате обмена хромосомной информацией между родительскими осо-

бями (блок «Скрещивание» на рисунке 4). Созданные потомки формируют новую популяцию, причем часть потомков мутирует (используется генетический оператор мутации), что выражается в случайном изменении их генотипов (блок «Мутация» на рисунке 4). Этап, включающий в себя последовательность «Оценивание популяции» – «Селекция» – «Скрещивание» – «Мутация», называется поколением. Эволюция популяции состоит из последовательности таких поколений.

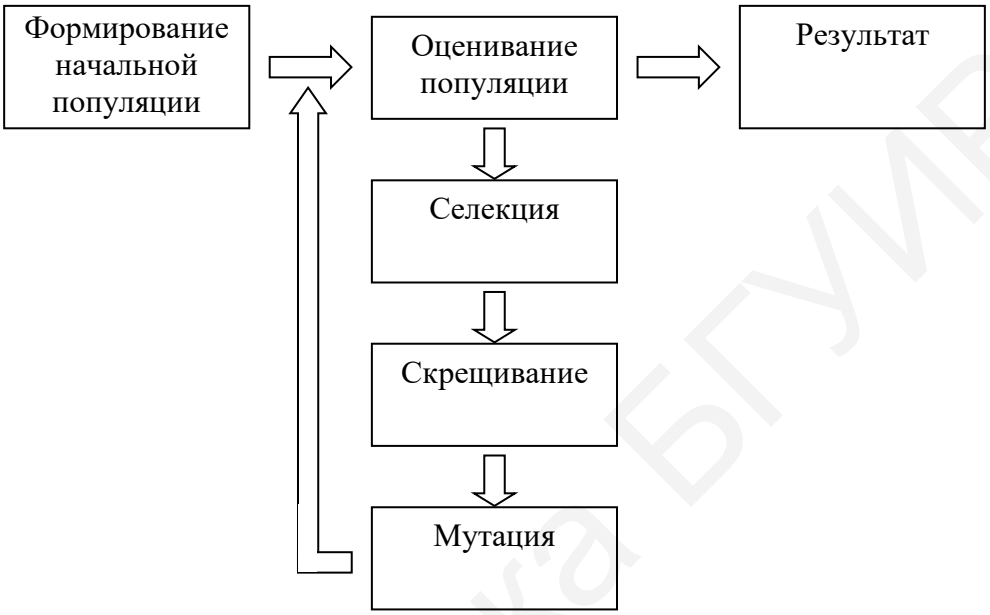



Рисунок 4 – Схема генетического алгоритма

Длительность эволюции может определяться следующими факторами:

– нахождение решения в результате эволюционного поиска;

– ограниченность количества поколений;

– ограниченность количества вычислений функции приспособленности (целевой функции);

– вырождение популяции, когда степень разнородности хромосом в популяции становится меньше допустимого значения.

Рассмотрим параметры и этапы генетического алгоритма.

### 5.1 Кодирование информации и формирование популяции

Выбор способа кодирования является одним из важнейших этапов при использовании эволюционных алгоритмов. В частности, должно выполняться следующее условие: возможность закодировать (с допустимой погрешностью) в хромосоме любую точку из рассматриваемой области пространства поиска. Невыполнение этого условия может привести как к увеличению времени эволюционного поиска, так и к невозможности найти решение поставленной задачи. Как правило, в хромосоме кодируются численные параметры

решения. Для этого возможно использование целочисленного и вещественного кодирования.

В классическом генетическом алгоритме хромосома представляет собой битовую строку, в которой закодированы параметры решения поставленной задачи. На рисунке 5 показан пример кодирования четырех 10-разрядных параметров в 40-разрядной хромосоме.

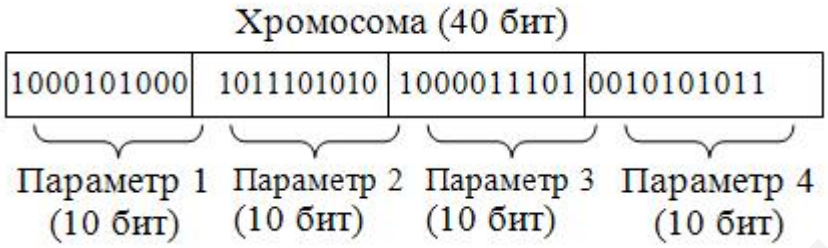



Рисунок 5 – Пример целочисленного кодирования

Обычно считают, что каждому параметру соответствует свой ген. Таким образом, хромосома на рисунке 5 состоит из четырех 10-разрядных генов. Несмотря на то что каждый параметр закодирован в хромосоме целым числом (в виде двоичной последовательности), ему могут быть поставлены в соответствие и вещественные числа. Ниже представлен один из вариантов прямого и обратного преобразования «целочисленный ген → вещественное число».

Часто бывает удобнее кодировать в гене не целое число, а вещественное. Это позволяет избавиться от операций кодирования/декодирования, используемых в целочисленном кодировании, а также увеличить точность найденного решения. Пример вещественного кодирования представлен на рисунке 6.

Как правило, начальная популяция формируется случайным образом. При этом гены инициализируются случайными значениями.


Рисунок 6 – Пример вещественного кодирования

Оценивание популяции необходимо для того, чтобы выявить в ней более приспособленные и менее приспособленные особи. Для подсчета приспособленности каждой особи используется функция приспособленности (целевая функция):

$$f_i = f(G_i),$$

где $G_i$ – хромосома $i$-й особи ($G_i = \{g_{ik} : k = 1, 2, ..., N\}$, где $g_{ik}$ – значение $k$-го гена $i$-й особи; $N$ – количество генов в хромосоме).

В случае использования целочисленного кодирования для вычисления значения функции приспособленности часто бывает необходимо преобразовать закодированные в хромосоме целочисленные значения к вещественным числам. Другими словами,

$$f_i = f(X_i),$$

где $X_i$ – вектор вещественных чисел, соответствующих генам $i$-й хромосомы ($X_i = \{x_{ik} : k = 1, 2, ..., N\}$).

Как правило, использование эволюционного алгоритма подразумевает решение задачи максимизации (минимизации) целевой функции, когда необходимо найти такие значения параметров функции $f$, при которых значение функции максимально (минимально). В соответствии с этим, если решается задача минимизации и $f(G_i) < f(G_j)$, то считают, что $i$-я особь лучше (приспособленнее) $j$-й особи.

В случае задачи максимизации, наоборот, если $f(Gi) > f(Gj)$, то $i$-я особь считается более приспособленной, чем $j$-я особь.

## 5.2 Селекция

Селекция (отбор) необходима, чтобы выбрать более приспособленных особей для скрещивания. Существует множество вариантов селекции, рассмотрим наиболее известные из них.

*Рулеточная селекция*. В данном варианте селекции вероятность $i$-й особи принять участие в скрещивании $p_i$ пропорциональна значению ее приспособленности $f_i$ и равна $p_i = \dfrac{f_i}{\sum_j f_j}$.

Процесс отбора особей для скрещивания напоминает игру в «рулетку». Рулеточный круг делится на сектора, причем площадь $i$-го сектора пропорциональна значению $p_i$. После этого $n$ раз «вращается» рулетка, где $n$ – размер популяции, и по сектору, на котором останавливается рулетка, определяется особь, выбранная для скрещивания.

*Селекция усечением*. При отборе усечением после вычисления значений приспособленности для скрещивания выбираются $ln$ лучших особей, где $l$ – «порог отсечения», $0 < l < 1$, $n$ – размер популяции. Чем меньше значение $l$, тем сильнее давление селекции, т. е. меньше шансы на выживание у плохо приспособленных особей. Обычно выбирают $l$ в интервале от 0,3 до 0,7.

*Турнирный отбор*. В случае использования турнирного отбора для скрещивания, как и при рулеточной селекции, отбираются $n$ особей. Для этого из популяции случайно выбираются $t$ особей, и самая приспособленная из них допускается к скрещиванию. Считается, что формируется турнир из $t$ особей,

$t$ – размер турнира. Эта операция повторяется $n$ раз. Чем больше значение $t$, тем больше давление селекции.

Вариант турнирного отбора, когда $t = 2$, называют бинарным турниром. Типичные значения размера турнира $t$ равны 2, 3, 4, 5.

### 5.3 Скрещивание и формирование нового поколения

Отобранные в результате селекции особи, называемые родительскими, скрещиваются и дают потомство. Хромосомы потомков формируются в процессе обмена генетической информацией между родительскими особями. Для этого применяется оператор кроссинговера. Созданные таким образом потомки составляют популяцию следующего поколения. Будем рассматривать случай, когда из множества родительских особей случайным образом выбираются две особи и скрещиваются с вероятностью $P_C$, в результате чего создаются два потомка. Этот процесс повторяется до тех пор, пока не будет создано $n$ потомков. Вероятность скрещивания $P_C$ является одним из ключевых параметров генетического алгоритма и в большинстве случаев ее значение находится в диапазоне от 0,6 до 1.

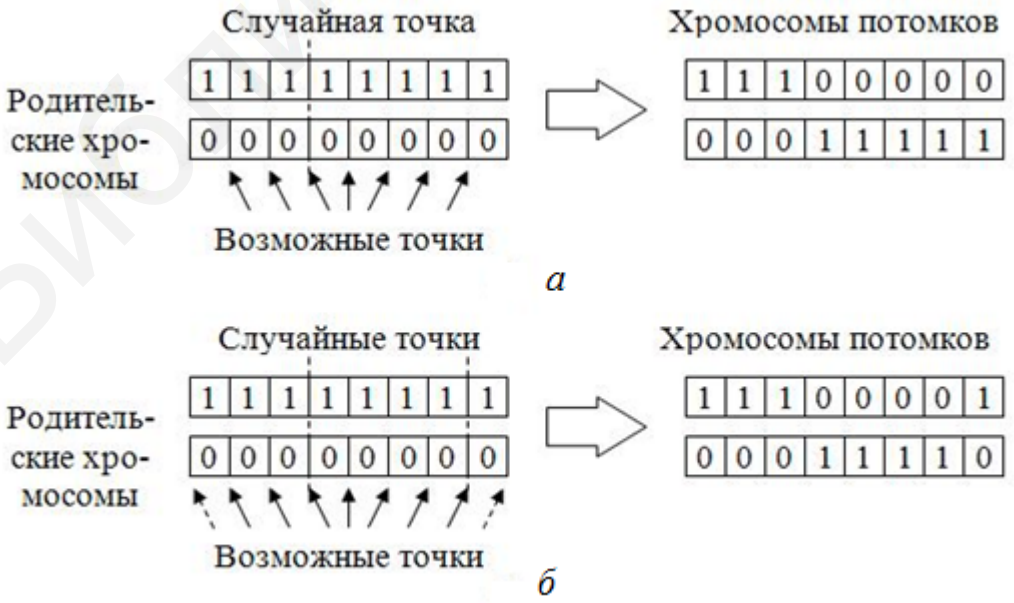
Для кодирования часто используются 1-точечный, 2-точечный и однородный операторы кроссинговера. 1-точечный кроссинговер работает аналогично операции перекреста для хромосом при скрещивании биологических организмов. Для этого выбирается произвольная точка разрыва и для создания потомков производится обмен частями родительских хромосом. Пример работы 1-точечного кроссинговера представлен на рисунке 7, *а*. Для оператора 2-точечного кроссинговера выбираются две случайные точки разрыва, после чего для создания потомков родительские хромосомы обмениваются участками, лежащими между точками разрыва (рисунок 7, *б*). Для 2-точечного оператора кроссинговера начало и конец хромосомы считаются «склеенными», в результате чего одна из точек разрыва может попасть в начало/конец хромосом. В таком случае результат работы 2-точечного кроссинговера будет совпадать с результатом работы 1-точечного кроссинговера. На рисунке 7, *б* точка разрыва в месте склеивания хромосом показана пунктирными стрелками.

При использовании однородного оператора кроссинговера разряды родительских хромосом наследуются независимо друг от друга. Для этого определяют вероятность $p_0$, что $i$-й разряд хромосомы первого родителя попадет к первому потомку, а второго родителя – ко второму потомку. Вероятность противоположного события равна $(1 - p_0)$. Каждый разряд родительских хромосом «разыгрывается» в соответствии со значением $p_0$ между хромосомами потомков. Чаще всего вероятность обоих событий одинакова, т. е. $p_0 = 0{,}5$.

Операторы кроссинговера характеризуются способностью к разрушению родительских хромосом. Кроссинговер для целочисленного кодирования считается более разрушительным, если в результате его применения различие между получившимися хромосомами потомков и хромосомами родителей велико. Это означает, что способность целочисленного кроссинговера к разрушению зависит от того, насколько сильно он «перемешивает» (рекомбинирует) содержимое родительских хромосом. Так, 1-точечный кроссинговер считается слаборазрушающим, а однородный кроссинговер в большинстве случаев является сильноразрушающим оператором. 2-точечный кроссинговер по разрушающей способности занимает промежуточную позицию по отношению к 1-точечному и однородному операторам кроссинговера.

Одновременно со способностью к разрушению оценивается способность к созданию кроссинговером новых особей. Получается, что, разрушая хромосомы родительских особей, кроссинговер может создать новые хромосомы, не встречавшиеся ранее в процессе эволюционного поиска.

В результате скрещивания создаются потомки, которые формируют популяцию следующего поколения. Обновленная таким образом популяция необязательно должна включать в себя одних только особей-потомков. Пусть доля обновляемых особей равна $T$, $0 < T < 1$, тогда в новое поколение попадает $Tn$ потомков, где $n$ – размер популяции, а $(1 - T)n$ особей в новой популяции являются наиболее приспособленными родительскими особями, так называемыми элитными особями. Параметр $T$ отражает разрыв поколений. Использование элитных особей позволяет увеличить скорость сходимости генетического алгоритма.




*а* – 1-точечного; *б* – 2-точечного

Рисунок 7 – Примеры работы кроссинговера

## 5.4 Мутация

Оператор мутации используется для внесения случайных изменений в хромосомы особей. Это позволяет «выбираться» из локальных экстремумов и тем самым эффективнее исследовать пространство поиска. Аналогично оператору кроссинговера работа оператора мутации зависит от вероятности применения мутации $P_M$. Рассмотрим базовые варианты оператора мутации в зависимости от способа представления генетической информации.

*Целочисленное кодирование*. Одним из основных операторов мутации для целочисленного кодирования является битовая мутация. Она изменяет отдельные разряды в хромосоме. Для этого каждый разряд инвертируется с вероятностью $P_M$. В силу того что применение мутации разыгрывается столько раз, сколько разрядов содержится в хромосоме, значение $P_M$ выбирают небольшим, чтобы сильно не разрушать найденные хорошие хромосомы. Один из типичных вариантов: $P_M = L^{-1}$, где $L$ – длина хромосомы в битах. В этом случае каждая хромосома мутирует в среднем один раз.

*Вещественное кодирование*. Оператор мутации для вещественного кодирования изменяет содержимое каждого гена с вероятностью $P_M$, а величина изменения выбирается случайно в некотором диапазоне $[-\xi; +\xi]$, например, [–0,5; 0,5], и может иметь как равномерное, так и любое другое распределение. Для того чтобы избежать сильных изменений содержимого хромосомы в результате мутации, значение вероятности $P_M$ выбирается небольшим. Например, $P_M = N^{-1}$, где $N$ – количество генов в хромосоме. Также возможна адаптивная настройка величины диапазона $2\xi$ изменения значения гена в результате мутации.

## 5.5 Настройка параметров генетического алгоритма

Результат работы ГА сильно зависит от того, каким образом настроены его параметры. Основными параметрами ГА являются:

– длительность эволюции (количество поколений);
– размер популяции;
– интенсивность (давление) селекции;
– тип оператора кроссинговера;
– вероятность кроссинговера $P_C$;
– тип оператора мутации;
– вероятность мутации $P_M$;
– величина разрыва поколений $T$.

Различные параметры влияют на разные аспекты эволюционного поиска, среди которых можно выделить два наиболее общих:

1) исследование пространства поиска;
2) использование найденных «хороших» решений.

Первый аспект отвечает за способности ГА к эффективному поиску решения и характеризует способности алгоритма избегать локальных экстремумов. Второй аспект важен для постепенного улучшения имеющихся результатов от поколения к поколению на основе уже найденных «промежуточных» решений.

# ЛАБОРАТОРНЫЙ ПРАКТИКУМ

# ЛАБОРАТОРНАЯ РАБОТА №1

## ПРИМЕНЕНИЕ ЗАКОНОВ ЗИПФА ДЛЯ ОБРАБОТКИ ТЕКСТОВОЙ ИНФОРМАЦИИ

**Цель работы:** ознакомиться с законами Зипфа и научиться их применять для определения заданных характеристик текстовой информации.

Порядок выполнения работы:
1 Изучить теоретическую часть работы.
2 Реализовать алгоритм проверки первого закона Зипфа.
3 Реализовать алгоритм проверки второго закона Зипфа.
4 Оформить отчет по лабораторной работе.

**Исходные данные:** тексты размером 1–2 страницы на русском, белорусском, английском языках.

**Выходные данные:** вычисленные характеристики Зипфа и наборы ключевых слов для исходных текстов.

Большинство существующих подходов к анализу текстов можно разбить на два класса. К первому классу относятся простые, быстрые, но не очень точные механизмы анализа. Чаще всего эти подходы используют формальные статистические методы, основанные на частоте появления в тексте слов различных тематик. Второй класс формируют достаточно изощренные, дающие хороший результат, но сравнительно медленные подходы, основанные на лингвистических методах. Эффективным же можно считать такой подход, который сочетал бы в себе простоту статистических алгоритмов с достаточно высоким качеством обработки лингвистических методов.

В то же время, как показала практика, для достижения приемлемого качества решения практических задач компьютерного анализа текстовой информации (автоматическое аннотирование, тематическая классификация) не требуется полный грамматический анализ фразы. Достаточно выделить наиболее информативные единицы текста: ключевые слова, словосочетания, предложения и фрагменты. При этом в качестве характеристики информативности удобно выбирать частоту повторения слов в тексте.

Для выделения понятий текста, представляющих слова и словосочетания, может быть применен статистический алгоритм, основанный на анализе частоты встречаемости цепочек слов различной длины и их вхождения друг в друга. Во всех созданных человеком текстах можно выделить статистические закономерности, которые никому не удается обойти. Независимо от текста и языка написания внутренняя структура текста остается неизменной. Она описывается законами Дж. Зипфа, который предположил, что слова с большим количеством букв встречаются в тексте реже коротких слов. Основываясь на этом постулате, Зипф вывел два универсальных закона.

### 1.1 Первый закон Зипфа «ранг – частота»

Если измерить количество вхождений каждого слова в текст и взять только одно значение из каждой группы, имеющей одинаковую частоту, расположить частоты по мере их убывания и пронумеровать (порядковый номер частоты называется рангом частоты – $r$), то наиболее часто встречающиеся слова будут иметь ранг 1, следующие за ними – 2 и т. д. Вероятность $p_i$ встретить произвольно выбранное слово равна отношению количества вхождений этого слова $n_i$ к общему числу слов $n$ в тексте, т. е.

$$p_i = n_i / n. \quad (1)$$

Зипф обнаружил следующую закономерность: произведение вероятности обнаружения слова в тексте на ранг частоты $r$ – есть константа ($C$):

$$C = p_i \cdot r, \text{ или } C = (n_i \cdot r)/n. \quad (2)$$

Это функция типа $y = k/x$, а ее график – равносторонняя гипербола. Следовательно, по первому закону Зипфа, если самое распространенное слово встречается в тексте, например, 100 раз, то следующее по частоте слово с высокой долей вероятности окажется на уровне 50.

Значение константы $C$ в разных языках различно, но внутри одной языковой группы остается неизменно, какой бы текст мы ни взяли. Так, например, для английских текстов константа Зипфа равна приблизительно 0,1.

### 1.2 Второй закон Зипфа «количество – частота»

В первом законе не учтен тот факт, что разные слова могут входить в текст с одинаковой частотой. Зипф установил, что частота и количество слов, входящих в текст с этой частотой, тоже связаны между собой. Если построить график, отложив по оси $X$ частоту вхождения слова, а по оси $Y$ – количество слов с данной частотой, то получившаяся кривая будет сохранять свои параметры для всех без исключения созданных человеком текстов. Как и в предыдущем случае, это утверждение верно в пределах одного языка. Однако и межъязыковые различия невелики. На каком бы языке текст не был написан, форма кривой Зипфа останется неизменной (рисунок 8). Могут немного отличаться лишь коэффициенты, отвечающие за наклон кривой.

Законы Зипфа универсальны. В принципе, они применимы не только к текстам. В аналогичную форму выливается, например, зависимость количества городов от числа проживающих в них жителей. Воспользуемся законами Зипфа для извлечения из текста слов, отражающих его смысл, т. е. ключевых слов. На рисунке 9 показан график зависимости ранга частоты слов от частоты их вхождения в текст.

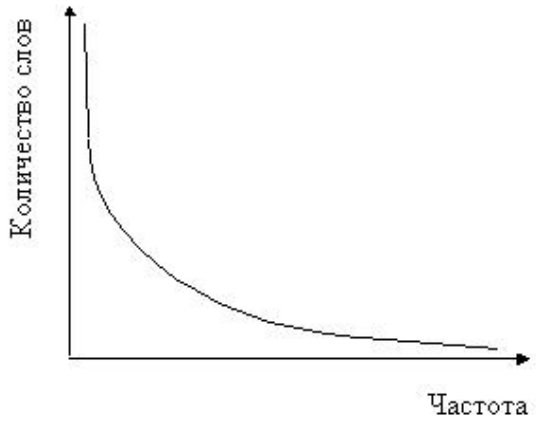



Рисунок 8 – График зависимости частоты вхождения слова от количества слов с данной частотой

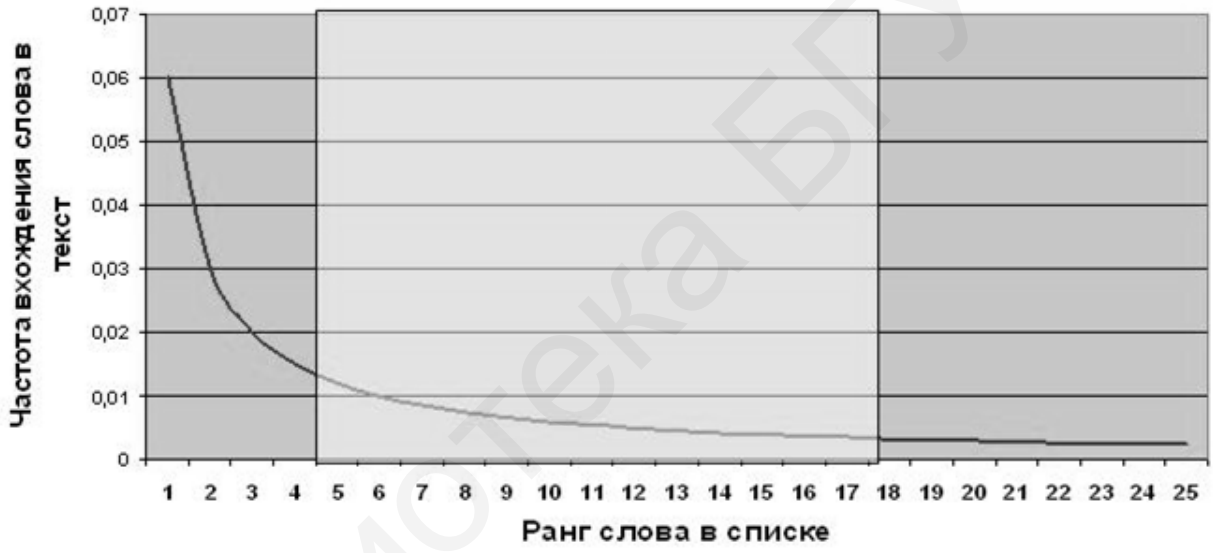



Рисунок 9 – График зависимости ранга от частоты вхождения слова

Исследования показали, что наиболее значимые слова лежат в средней части диаграммы (на рисунке 9 это слова, имеющие ранг от 4 до 17). Это объясняется тем, что слова, которые встречаются слишком часто, в основном оказываются предлогами и местоимениями, в английском языке – артиклями и т. п. Редко встречающиеся слова тоже в большинстве случаев не имеют решающего смыслового значения.

От того как будет выставлен диапазон значимых слов, зависит многое. Если диапазон широкий, то ключевыми словами будут вспомогательные слова; если установить узкий диапазон, то можно потерять смысловые термины.

Выделить ключевые слова помогает предварительное исключение из исследуемого текста некоторых слов, которые априори не могут являться значимыми, и поэтому являются «шумом». Такие слова называются нейтральными, или стоповыми (стоп-словами). Словарь стоп-слов называют

стоп-листом. Например, для английского текста стоп-словами считаются следующие: the, a, an, in, to, of, and, that… и т. д.

### 1.3 Весовые коэффициенты

Каждый из рассматриваемых отдельно взятых документов на практике чаще всего входит в базу данных (коллекцию) наряду с множеством других документов. Если представить всю базу данных как единый документ, к ней можно будет применить те же законы, что и к единичному документу.

В ходе назначения терминов (ключевых слов) коллекции стремятся избавиться от лишних слов (шума) и в то же время поднять рейтинг значимых слов, для чего вводят инверсную частоту термина. *Значение этого параметра тем меньше, чем чаще слово встречается в документах базы данных.* Вычисляют его по формуле

$$\lambda_i = \log(N/N_i), \tag{3}$$

где $\lambda_i$ – инверсная частота термина $i$; $N$ – количество документов в базе данных; $N_i$ – количество документов с термином $i$.

Теперь каждому термину можно присвоить весовой коэффициент, отражающий его значимость:

$$k_{ij} = n_{ij} \cdot \lambda_i, \tag{4}$$

где $k_{ij}$ – вес термина $i$ в документе $j$; $n_{ij}$ – частота термина $i$ в документе $j$; $\lambda_i$ – инверсная частота термина $i$.

В качестве ключевых слов будут выбираться слова, имеющие высокий вес.

Современные способы индексирования не ограничиваются анализом перечисленных параметров текста. Поисковая машина может строить весовые коэффициенты с учетом местоположения термина внутри документа, взаимного расположения терминов, частей речи, морфологических особенностей и т. п. В качестве терминов могут выступать не только отдельные слова, но и словосочетания. Без этих законов Зипфа сегодня не обходится ни одна система автоматического поиска информации. Это обусловлено тем, что математический анализ позволяет машине с высокой точностью без участия человека распознать суть текста.

# ЛАБОРАТОРНАЯ РАБОТА №2

# ПОИСК ТЕКСТОВОЙ ИНФОРМАЦИИ ПО ЗАДАННОМУ НАБОРУ КЛЮЧЕВЫХ СЛОВ

**Цель работы:** реализовать поиск текстовой информации по документу-образцу.

Порядок выполнения работы:

1 Ознакомиться с теоретической частью работы.

2 Реализовать алгоритм поиска текстовой информации по документу-образцу.

3 Оформить отчет по лабораторной работе.

**Исходные данные:** коллекция документов из 10–12 текстов на русском и английском языках. Результат первой лабораторной работы: вектор ключевых слов, построенный по документу-образцу.

**Выходные данные:** документы, найденные в результате работы поисковой машины и соответствующие запросу поиска.

## 2.1 Особенности поиска информации

Накопленные к настоящему времени объемы информации в совокупности с непрерывно увеличивающимися темпами их роста определяют актуальность и значимость исследований в области информационного поиска. Быстрое развитие сетевых технологий, в том числе и Интернета, способствуют значительному увеличению доступных информационных ресурсов и объемов передаваемой информации. Зачастую это разнородная, слабо структурированная и избыточная информация, обладающая высокой динамикой обновления.

При сегодняшних объемах доступной информации решение задач информационного поиска становится не только приоритетным, но и элементарно необходимым условием для обеспечения своевременного доступа к интересующей информации.

Информационный поиск – самостоятельное направление исследований, изучающее вопросы поиска документов, обработки результатов поиска, а также целый ряд смежных вопросов: моделирования, классификации, кластеризации и фильтрации документов, проектирования архитектур поисковых систем и пользовательских интерфейсов, языков запросов и т. д.

Способы поиска можно разделить на две большие группы.

1 Библиографический поиск, или поиск «по каталогу».

Такой вариант поиска обеспечивает нахождение документов по их выходным данным, например, по названию документа, по его тематике, по именам авторов, датам публикаций и т. д. Эти выходные данные составляют рек-

визиты документа. Основой каталога является предварительно заданная модель представления реквизитов, реализованная в виде базы данных, в соответствии с которой обеспечивается запись отдельных элементов реквизитов и последующий поиск по ним.

Основная проблема и недостаток такого варианта поиска – это необходимость выполнения значительного объема работ по предварительной организации, наполнению каталога. Как правило, это ручная классификация на основе привлечения экспертов. Подобный подход позволяет организовать лишь самую малую часть доступных информационных ресурсов.

2 Тематический поиск, или поиск «по тексту».

Этот вариант поиска ориентирован на нахождение документов по их содержанию. Сюда же относится так называемый полнотекстовый поиск. Общая схема такого поиска заключается в формулировании некоторого запроса пользователем относительно содержания документа и отборе из множества доступных документов тех, которые удовлетворяют запросу. Такой вариант поиска удобен прежде всего тем, что нет необходимости в предварительном разделении документов по различным категориям. Особенно это актуально при значительном объеме доступных документов, высокой динамике их обновления или отсутствии некоторых реквизитов, такая ситуация характерна для Интернета.

Основная проблема такого поиска – это сложность однозначной автоматической интерпретации содержания текстов документов и формулировок информационных потребностей пользователей. Сложность интерпретации затрудняет определение соответствия рассматриваемого документа информационным потребностям пользователя. Эти проблемы обусловлены отсутствием какой-либо регулярной структуры у текстовых документов на естественном языке. Такие информационные ресурсы принято называть неструктурированными, или слабоструктурированными. Разработка методов анализа слабоструктурированных информационных ресурсов представляется весьма перспективным и многообещающим направлением исследований в области информационного поиска.

В соответствии с вышеприведенной классификацией способов поиска принято выделять два основных класса информационно-поисковых систем:

1) поисковые каталоги;

2) поисковые системы.

*Поисковые каталоги* в большей степени ориентированы на структурную организацию тематических коллекций с удобной системой ссылок и иерархией документов по тематическим коллекциям. Это позволяет пользователю самостоятельно находить требуемый документ, просматривая структуру каталога, либо использовать механизмы поиска, ориентированные на данный каталог.

*Поисковые системы* работают со слабоструктурированной информацией. Как правило, они используются для поиска документов в больших и динамичных информационных коллекциях, например, в Интернете. Особен-

ностью таких коллекций является отсутствие четко выраженной структурной организации, позволяющей упорядочить и однозначно классифицировать хранящиеся в них документы по тематической направленности.

В рамках данного учебно-методического пособия наибольший интерес представляют поисковые системы, а точнее, используемые в них методы анализа документов. Процесс поиска текстовой информации, реализуемый типичной поисковой системой, включает в себя следующие этапы:

– формализация пользователем поискового запроса (представление пользователем в том или ином виде своих информационных потребностей);

– предварительный отбор документов по формальным признакам наличия интересующей информации (например, наличие в тексте документа одного из слов запроса, если запрос формулируется на естественном языке);

– анализ отобранных документов (лингвистический, статистический);

– оценка соответствия смыслового содержания найденной информации требованиям поискового запроса (ранжирование).

Одним из ключевых понятий, характеризующих выбор того или иного метода анализа текстовой информации, а также реализацию конкретного варианта поиска, является модель поиска. Модель поиска – это сочетание следующих составляющих:

– способа представления документов;

– способа представления поисковых запросов;

– вида критерия релевантности документов.

Всю совокупность представленных на сегодняшний день методов тематического анализа текста можно разделить на две группы:

1) лингвистический анализ;

2) статистический анализ.

Первый ориентирован на извлечение смысла текста по его семантической структуре, второй – по частотному распределению слов в тексте. Однако говорить о принадлежности какого-либо из подходов к конкретной группе можно лишь условно, как правило, в реальных задачах обработки текста приходится использовать сочетание методик из обеих групп.

Лингвистический анализ можно разделить на четыре взаимодополняющих анализа: лексический, морфологический, синтаксический, семантический.

Статистический анализ – это, как правило, частотный анализ в тех или иных его вариациях. Суть такого анализа заключается в подсчете количества повторений слов в тексте и использовании результатов подсчета для конкретных целей. Например, вычисление весовых коэффициентов ключевых слов. Одним из эффективных статистических подходов является способ поиска по документу-образцу.

Документ-образец выступает в качестве одной из форм представления информационных потребностей пользователя. Целью поиска является обнаружение тематически близких документов. Самым простым подходом к решению задачи поиска документов по образцу является использование всех

слов документа-образца в качестве запроса. Однако длина такого запроса может оказаться очень большой, что отрицательно скажется на качестве поиска, т. к. результатом поиска будут все документы, в которых присутствовали данные слова, и таких документов может быть очень много. Это отрицательно скажется как на самой поисковой системе – вычислительные ресурсы и трафик не безграничны, и система может оказаться перегруженной, так и на человеке – просмотр и анализ найденных документов может занять значительное время, редкий пользователь готов к этому. Приемлемым вариантом в данном случае является выделение тематики документа. Под тематикой понимается множество ключевых слов, описывающих с некоторой степенью адекватности содержание документа. Тематика – это приближенное представление документа. Для повышения точности и адекватности описания содержания документа ключевые слова используются с некоторыми весовыми коэффициентами, которые соотносятся с частотой повторений этих слов в тексте. Вопросы выделения тематики и вычисления тематической близости документов по их тематическому представлению во многом и определяют возможность и эффективность поиска по документу-образцу.

## 2.2 Алгоритм поиска

Всю коллекцию документов необходимо организовать так, чтобы можно было легко отыскать в ней нужный материал. База данных должна взаимодействовать с пользовательским запросом. Запросы могут быть простыми, состоящими из одного слова, и сложными – из нескольких слов, связанных логическими операторами. Простой запрос оправдывает свое название. Пользователь вводит слово, машина ищет его в списке терминов и выдает все связанные с термином ссылки. Структура такой базы данных проста. Взаимодействие со сложными запросами требует более изощренной организации.

Рассмотрим последовательность действий для организации поиска:

1 Подбирается текст-источник. Чем четче описаны проблемы в тексте-источнике, тем качественнее и точнее окажется результат.

2 Удаляются из текста стоп-слова.

3 Без учета морфологии слов вычисляется частота вхождения каждого термина.

4 Ранжируются термины в порядке убывания их частоты вхождения.

5 Выбирается диапазон частот из построенного по определенному закону упорядоченного списка. Диапазон выбирается из середины списка. Достаточно взять 10–20 терминов.

6 Составляется запрос, причем отобранные слова располагаются в порядке их следования в списке терминов. Запрос должен восприниматься машиной как слова, связанные логическим оператором ИЛИ.

7 Запрос отправляется поисковой системе.

# ЛАБОРАТОРНАЯ РАБОТА №3

# АВТОМАТИЧЕСКАЯ РУБРИКАЦИЯ ТЕКСТОВОЙ ИНФОРМАЦИИ ПО ОБРАЗЦУ

**Цель работы:** научиться классифицировать по темам текстовую информацию, используя документы-образцы.

Порядок выполнения работы:

1 Ознакомиться с теоретической частью работы.

2 Реализовать алгоритм классификации текстов по рубрикам на основе документов-образцов.

3 Оформить отчет по лабораторной работе.

**Исходные данные:** тексты-образцы, о которых известно, к каким рубрикам они относятся, и тексты, требующие тематической классификации по заданным рубрикам.

**Выходные данные:** список рубрик с принадлежащими им текстовыми документами.

## 3.1 Алгоритм автоматической рубрикации текстов по образцу

Алгоритм автоматической рубрикации текстов по образцу следующий:

1 Применение законов Зипфа для построения векторов признаков (ключевых слов) текстов-образцов, задающих необходимые рубрики.

2 Обучение системы классификации с помощью текстов-образцов. Применение метода персептрона для построения $p$ линейных функций (по числу классов), отделяющих в $r$-мерном пространстве признаков каждый класс от всех остальных.

3 Построение по законам Зипфа наборов ключевых слов для текстов, требующих рубрикации.

4 Рубрикация текстов с помощью разделяющих функций, построенных на втором шаге алгоритма.

Рассмотрим более подробно каждый из шагов алгоритма.

1 Для построения векторов с признаками текстов-образцов создадим словарь терминов. Каждая рубрика может быть задана произвольным количеством текстов-образцов. Пусть векторы признаков всех рубрик, чье количество обозначим $a$, имеют равное количество ключевых слов – $b$. Тогда словарь ключевых слов можно представить следующей динамической структурой (рисунок 10). В вертикальной таблице расположены заголовки классов: от 1 до $a$. Каждый элемент таблицы содержит ссылку на список ключевых слов данного класса. Списки могут быть разной длины в зависимости от количества текстов-образцов, имеющихся для каждой рубрики.

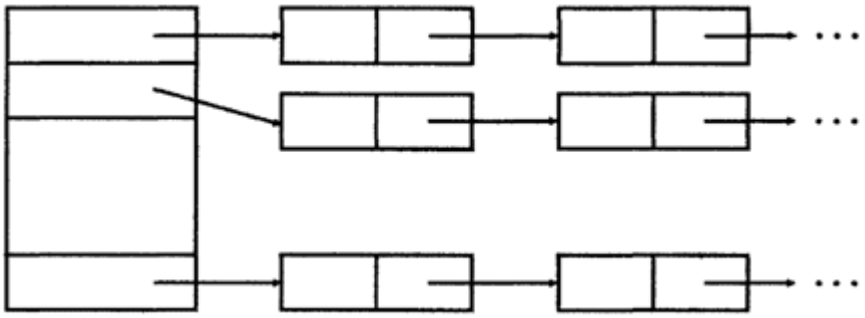


Рисунок 10 – Структура словаря терминов рубрик

Вектор признаков каждого класса представим следующей структурой.

| | | | |
|---|---|---|---|
| $x_{i,j,k}$ | $x_{i,j,k+1}$ | ... | $x_{i,j,b}$ |

Здесь $x$ – это ключевые слова $j$-го текста-образца $i$-й рубрики. Индекс $i$ изменяется от 1 до $a$, индекс $j$ зависит от представленного количества образцов для данной рубрики. Индекс $k$ – это номер ключевого слова в векторе, поэтому он изменяется от 1 до $b$.

На основе векторов признаков всех образцов для всех рубрик определим структуру обобщенного вектора признаков, на базе которого будем строить разделяющие функции. Рассмотрим на примере построение обобщенного вектора ключевых слов. В обобщенный вектор войдут все ключевые слова из всех текстов-образцов.

Пусть имеются следующие исходные данные: $a = 3$, $b = 2$. Рубрики заданы следующими векторами признаков.

| | | | |
|---|---|---|---|
| $x_{1,1,1}$ | $x_{1,1,2}$ | $x_{2,1,1}$ | $x_{2,1,2}$ |

| | |
|---|---|
| $x_{3,1,1}$ | $x_{3,1,2}$ |

| | |
|---|---|
| $x_{1,2,1}$ | $x_{1,2,2}$ |

| | |
|---|---|
| $x_{3,2,1}$ | $x_{3,2,2}$ |

Для первого класса предложены два образца, для второго – один и для третьего – два образца. При этом второй признак образца второго класса совпадает с первым признаком второго образца третьего класса, т. е. $x_{2,1,2} = x_{3,2,1}$.

Тогда структура обобщенного вектора признаков будет следующей:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $x_{1,1,1}$ | $x_{1,1,2}$ | $x_{1,2,1}$ | $x_{1,2,2}$ | $x_{2,1,1}$ | $x_{2,1,2}$ | $x_{3,1,1}$ | $x_{3,1,2}$ | $x_{3,2,1}$ | $x_{3,2,2}$ |

Теперь каждый из пяти имеющихся текстов-образцов будет также описан вектором из десяти элементов. В позиции вектора, соответствующие ключе-

вым словам данного образца, заносятся единицы, в остальные позиции – нули. На рисунке 11 приведены векторы признаков для имеющихся пяти текстов-образцов.

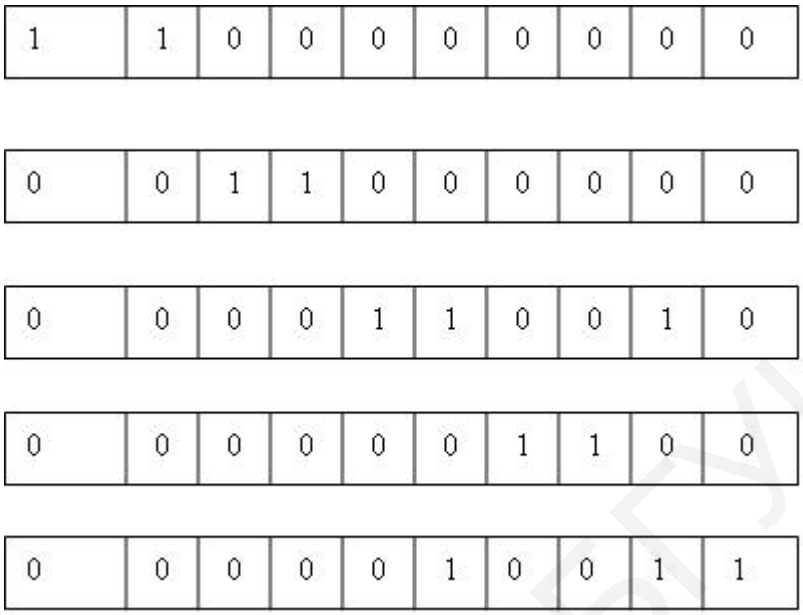



Рисунок 11 – Векторы признаков классов для построения разделяющих функций

2 После того как для каждого текста-образца составлен вектор ключевых слов, как показано на рисунке 11, применим метод персептрона для построения разделяющих функций по числу заданных рубрик (метод персептрона описан в подразделе 3.2).

3 Вектор признаков незнакомого текста имеет такую же структуру и размер, как и вектор признаков текста-образца. Сначала определяется вектор признаков, состоящий из $b$ элементов, незнакомого текста. Затем проверяется, есть ли среди признаков ключевые слова, входящие в обобщенный вектор. Пусть первоначально вектор признаков незнакомого текста имеет вид

| $y_1$ | $y_2$ |
|---|---|

При этом $y_1 = x_{1,1,2}; y_2 = x_{2,1,2} = x_{3,2,1}$. Тогда вектор признаков незнакомого текста примет следующий вид (рисунок 12).

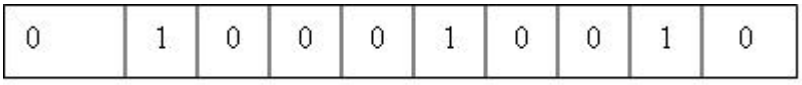



Рисунок 12 – Вектор признаков незнакомого текста

4 Вектор признаков незнакомого текста подставляется в каждую из разделяющих функций, и текст относится к той рубрике, чья функция показала максимальный результат.

### 3.2 Разделение объектов на *N* классов методом персептрона

В настоящее время существует большое множество методов классификации, в том числе и текстовой информации. Однако все методы распознавания можно разделить на две группы. Первая основана на понятии пространства признаков и их обработки в этом пространстве. Вторая – на исследовании конструкции рассматриваемых образов (синтаксическое распознавание).

Для решения поставленной задачи рассмотрим особенности первой группы методов. Для них в качестве основополагающей принята гипотеза о возможности представления образа в виде вектора, принадлежащего множеству *V*. Множество образов представляется в виде множества векторов, состоящего из таких *N* подмножеств, что каждый вектор, отнесенный в результате классификации к *j*-му классу, принадлежит подмножеству $E_j$.

Свойства множества *V* могут быть записаны в виде

$$\bigcup_{i=1}^{N} E_i = V, E_i \cap E_j = 0 (\forall i = j).$$

Задача классификации состоит в отыскании функции *f*, обеспечивающей разделение пространства *V* на требуемые классы:

$$f: V \rightarrow \Pi(V).$$

Процедура классификации заключается в том, чтобы для каждой области $R_l$ найти такую решающую функцию $g_l(x)$, удовлетворяющую следующему условию: если $g_i(\overline{x}) > g_j(\overline{x}), \forall j = 1,2,\ldots,N$, то $\overline{x} \in R_i$, где $N$ – общее количество областей.

Разделяющую функцию часто представляют в виде линейной суммы $g(\overline{x}) = \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \ldots + \omega_n x_n$, где $\omega_i$ – весовые коэффициенты, каждый из которых относится к определенной составляющей. Для удобства записи вводится весовой коэффициент с нулевым индексом $\omega_0$. Это позволяет записать решающую функцию в более компактной форме: $g(\overline{x}_a) = \overline{\omega}\overline{x}_a$, где $\overline{x}_a = \{1, x_1, x_2, \ldots, x_n\}$ – вектор, в число составляющих которого входит дополнительно одна вещественная константа. Ее величину обычно принимают равной единице.

Решающее правило *d* для случая *N* сепарабельных классов ($N > 2$) можно записать следующим образом:

$$d = \begin{cases} c_i, \text{ если } g_i(\overline{x}) = \overline{\omega}_i \overline{x}_a \geq 0, \\ \overline{c}_i, \text{ если } g_i(\overline{x}) < 0, \end{cases}$$

где *C* – множество, состоящее из *N* классов, $C = \{c_1, c_2, \ldots, c_N\}, c_i + \overline{c}_i = C$.

В процессе построения разделяющей функции основная задача заключается в том, чтобы найти весовые коэффициенты вида $\overline{\omega}_i = \{\omega_{0i}, \omega_{1i} \ldots\}$ для каждого конкретного применения.

Рассмотрим один из вариантов применения линейных разделяющих функций для разбиения объектов на $N$ классов.

Существует $M$ решающих функций $d_k(x) = w_k x, k = 1,2,\ldots,M$, таких, что если образ $x$ принадлежит классу $\omega_i$, то $d_i(x) > d_j(x)$ для всех $j \neq i$.

Граница между классами $\omega_i$ и $\omega_j$ определяется теми значениями вектора $x$, при которых имеет место равенство $d_i(x) = d_j(x)$. Поэтому при выводе уравнения разделяющей границы для классов $\omega_i$ и $\omega_j$ значения решающих функций $d_i(x)$ и $d_j(x)$ используются совместно.

Пример расположения разделяющих функций приведен на рисунке 13.

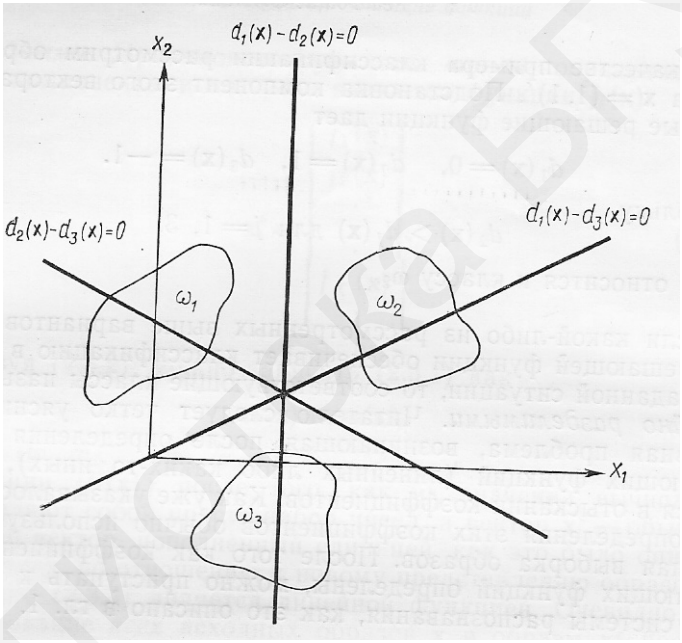



Рисунок 13 – Случай разделения образов на три класса

Для образов, принадлежащих классу $\omega_1$, должны выполняться условия $d_1(x) > d_2(x)$, $d_1(x) > d_3(x)$. В общем случае требуется, чтобы входящие в класс $\omega_i$ образы располагались в положительных зонах поверхностей $d_i(x) - d_j(x) = 0,\ j = 1,2,\ldots,M,\ i \neq j$. Положительная зона границы $d_i(x) - d_j(x) = 0$ совпадает с отрицательной зоной границы $d_j(x) - d_i(x) = 0$.

Пусть в качестве решающих функций выбраны следующие: $d_1(x) = -x_1 + x_2$, $d_2(x) = x_1 + x_2 - 1, d_3(x) = -x_2$. Разделяющие границы для трех классов выглядят при этом так:

$$d_1(x) - d_2(x) = -2x_1 + 1 = 0,$$
$$d_1(x) - d_3(x) = -x_1 + 2x_2 = 0,$$
$$d_2(x) - d_3(x) = x_1 + 2x_2 - 1 = 0.$$

Для того чтобы определить область решений, соответствующую классу $\omega_1$, необходимо выделить область, в которой выполняются неравенства $d_1(x) > d_2(x)$, $d_1(x) > d_3(x)$. Эта область совпадает с положительными зонами для прямых $-2x_1 + 1 = 0$ и $-x_1 + 2x_2 = 0$. Область принятия решения о принадлежности образа классу $\omega_2$ совпадает с положительными зонами для прямых $2x_1 - 1 = 0$ и $x_1 + 2x_2 - 1 = 0$. Область, отвечающая классу $\omega_3$, определяется положительными зонами для прямых $x_1 - 2x_2 = 0$ и $-x_1 - 2x_2 + 1 = 0$.

В качестве примера классификации рассмотрим обработку образа $x$ = (1, 1). Подстановка признаков образа в выбранные решающие функции дает следующие значения:

$$d_1(x) = 0,\ d_2(x) = 1,\ d_3(x) = -1.$$

Поскольку $d_2(x) > d_j(x), j = 1, 3$, образ относится к классу $\omega_2$.

Если какой-либо из рассмотренных вариантов линейной решающей функции обеспечивает классификацию в некоторой заданной ситуации, то соответствующие классы называются *линейно разделимыми.* Основная проблема, возникающая после определения набора решающих функций, заключается в отыскании коэффициентов. Для их определения обычно используется доступная выборка образов. После того как коэффициенты всех решающих функций определены, можно приступать к построению системы распознавания.

Рассмотрим алгоритм – *метод персептрона*, который можно применить для определения решающих функций, когда допускается существование $M$ решающих функций, характеризующихся тем свойством, что при $x \in \omega_i$, где $x$ – объект, $\omega_i$ – класс $d_i(x) > d_j(x)$ для всех $i \neq j$.

Рассмотрим $M$ классов $\omega_1, \omega_2, \ldots, \omega_M$. Пусть на $k$-м шаге процедуры обучения системе предъявляется образ $x(k)$, принадлежащий классу $\omega_i$. Вычисляются значения $M$ решающих функций $d_j[x(k)] = w_j(k)x(k), j = 1,2,\ldots,M$. Затем, если выполняются условия $d_i[x(k)] > d_j[x(k)], j = 1,2,\ldots,M,\ j \neq i$, то векторы весов не изменяются, т. е. $w_j(k+1) = w_j(k), j = 1,2,\ldots,M$.

С другой стороны, допустим, что для некоторого $l$ $d_i[x(k)] \leq d_l[x(k)]$. В этом случае выполняются следующие коррекции весов:

$$\begin{aligned} w_i(k+1) &= w_i(k) + cx(k), \\ w_l(k+1) &= w_l(k) - cx(k), \\ w_j(k+1) &= w_j(k), j = 1,2,\ldots,M; j \neq i, j \neq l, \end{aligned} \qquad (5)$$

где $c$ – положительная константа.

Если классы разделимы, то доказано, что этот алгоритм сходится за конечное число итераций при произвольных начальных векторах. Рассмотрим это на примере.

Даны классы, причем каждый из них содержит один образ: $\omega_1$: {(0, 0)}, $\omega_2$: {(1,1}, $\omega_3$: {( –1, 1)}. Дополним заданные образы одним свободным членом: (0, 0, 1), (1, 1, 1), (–1, 1, 1). Выберем в качестве начальных векторов весов $w_1(1) = w_2(1) = w_3(1) = (0,0,0)$, положим $c$ = 1 и, предъявляя образы в указанном порядке, получим следующее:

$$d_1[x(1)] = w_1(1)x(1) = 0,$$
$$d_2[x(1)] = w_2(1)x(1) = 0,$$
$$d_3[x(1)] = w_3(1)x(1) = 0.$$

Поскольку $x(1) \in \omega_1$ и $d_2[x(1)] = d_3[x(1)] = d_1[x(1)]$, первый весовой вектор увеличивается, а два других уменьшаются согласно соотношениям (5), т. е.

$$w_1(2) = w_1(1) + x(1) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$

$$w_2(2) = w_2(1) - x(1) = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix},$$

$$w_3(2) = w_3(1) - x(1) = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}.$$

Следующий предъявляемый образ $x(2)$ = (1, 1, 1) принадлежит классу $\omega_2$. Для него получаем $w_1(2)x(2) = 1, w_2(2)x(2) = -1, w_3(2)x(2) = -1$. Поскольку все произведения больше либо равны $w_2(2)x(2)$, вводятся коррекции:

$$w_1(3) = w_1(2) - x(2) = \begin{pmatrix} -1 \\ -1 \\ 0 \end{pmatrix},$$

$$w_2(3) = w_2(2) + x(2) = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix},$$

$$w_3(3) = w_3(2) - x(2) = \begin{pmatrix} -1 \\ -1 \\ -2 \end{pmatrix}.$$

Следующий предъявленный образ $x(3) = (-1, 1, 1)$ принадлежит классу $\omega_3$. Для него получаем $w_1(3)x(3) = 0, w_2(3)x(3) = 0, w_3(3)x(3) = -2$. Все эти произведения опять требуют корректировки:

$$w_1(4) = w_1(3) - x(3) = \begin{pmatrix} 0 \\ -2 \\ -1 \end{pmatrix},$$

$$w_2(4) = w_2(3) - x(3) = \begin{pmatrix} 2 \\ 0 \\ -1 \end{pmatrix},$$

$$w_3(4) = w_3(3) + x(3) = \begin{pmatrix} -2 \\ 0 \\ -1 \end{pmatrix}.$$

Поскольку в данном цикле итерации присутствовали ошибки, следует провести новый цикл. Положив $x(4) = x(1)$, $x(5) = x(2)$, $x(6) = x(3)$, получим $w_1(4)x(4) = -1$, $w_2(4)x(4) = -1, w_3(4)x(4) = -1$. Так как образ $x(4)$ принадлежит классу $\omega_1$, то все произведения неверны. Поэтому

$$w_1(5) = w_1(4) + x(4) = \begin{pmatrix} 0 \\ -2 \\ 0 \end{pmatrix},$$

$$w_2(5) = w_2(4) - x(4) = \begin{pmatrix} 2 \\ 0 \\ -2 \end{pmatrix},$$

$$w_3(5) = w_3(4) - x(4) = \begin{pmatrix} -2 \\ 0 \\ -2 \end{pmatrix}.$$

Следующий предъявленный образ $x(5) = (1, 1, 1)$ принадлежит классу $\omega_2$. Соответствующие скалярные произведения равны $w_1(5)x(5) = -2$, $w_2(5)x(5) = 0$, $w_3(5)x(5) = -4$. Образ $x(5)$ классифицирован правильно. Поэтому

$$w_1(6) = w_1(5) = \begin{pmatrix} 0 \\ -2 \\ 0 \end{pmatrix},$$

$$w_2(6) = w_2(5) = \begin{pmatrix} 2 \\ 0 \\ -2 \end{pmatrix},$$

$$w_3(6) = w_3(5) = \begin{pmatrix} -2 \\ 0 \\ -2 \end{pmatrix}.$$

Следующий образ $x(6) = (-1, 1, 1)$ принадлежит классу $\omega_3$, для него получаем $w_1(6)x(6) = -2, w_2(6)x(6) = -4, w_3(6)x(6) = -0$. Этот образ также классифицирован правильно, так что коррекции не нужны, т. е.

$$w_1(7) = w_1(6) = \begin{pmatrix} 0 \\ -2 \\ 0 \end{pmatrix},$$

$$w_2(7) = w_2(6) = \begin{pmatrix} 2 \\ 0 \\ -2 \end{pmatrix},$$

$$w_3(7) = w_3(6) = \begin{pmatrix} -2 \\ 0 \\ -2 \end{pmatrix}.$$

Если продолжить процедуру обучения, рассматривая образы $x(7)$, $x(8)$, $x(9)$, можно убедиться, что в следующем полном цикле никакие коррекции не производятся. Поэтому искомые решающие функции имеют следующий вид:

$$d_1(x) = 0 \cdot x_1 - 2x_2 + 0 = -2x_2,$$
$$d_2(x) = 2x_1 - 0 \cdot x_2 - 2 = 2x_1 - 2,$$
$$d_3(x) = -2x_1 + 0 \cdot x_2 - 2 = -2x_1 - 2.$$

Теперь, получив объект, требующий классификации, необходимо его признаки подставить в каждую из решающих функций и в качестве результата выбрать функцию (класс), на которой будет достигнуто максимальное значение.

# ЛАБОРАТОРНАЯ РАБОТА №4

# МЕТОД АВТОМАТИЧЕСКОГО РЕФЕРИРОВАНИЯ ТЕКСТОВОЙ ИНФОРМАЦИИ

**Цель работы:** научиться строить реферат на основе предложенного текстового документа.

Порядок выполнения работы:
1 Ознакомиться с теоретической частью работы.
2 Реализовать алгоритм автоматического синтеза реферата.
3 Оформить отчет по лабораторной работе.

**Исходные данные:** текстовый документ и величина реферата, указанная в процентах относительно исходного текста.
**Выходные данные:** автоматически построенный реферат.

## 4.1 Автоматическое реферирование и аннотирование текстов

Аннотирование текста заключается в формировании краткого описания его основных тем. Существует два разных подхода к аннотированию. В первом случае выявляется небольшое количество предложений, существующих в тексте, которые наиболее полно отражают основные темы текста. Дополнительно часто выделяются ключевые слова. Во втором случае основные темы текста выявляются как смыслы, и уже эти смыслы выражаются новыми предложениями, новым текстом. Второй вариант в большинстве случаев значительно более предпочтителен, но он и значительно сложнее. Все современные системы аннотирования/реферирования основаны на первом варианте. *SemLP*-технология позволяет реализовать второй вариант в ограниченном виде: автоматический синтез коротких (в несколько слов) простых фраз или предложений. В целом задача аннотирования включает определение тематики документов, выделение ключевых (по темам) слов и фраз с учетом смысла, поиск предложений, содержащих ключевые слова и фразы, и синтез на этой основе фраз и предложений, отражающих основные темы текста.

В рамках первого подхода выделяют три основных направления, которые в современных системах применяются совместно:

– статистические методы, основанные на оценке информативности разных элементов текста по частоте появления, которая служит основным критерием информативности слов, предложений или фраз;

– позиционные методы, которые опираются на предположение о том, что информативность элемента текста зависит от его позиции в документе;

– индикаторные методы, основанные на оценке элементов текста, исходя из наличия в них специальных слов и словосочетаний – маркеров важности, которые характеризуют их содержательную значимость.

После выявления набора ключевых слов по ним строится реферат. Преимущество методов первого подхода заключается в простоте их реализации. Однако выделение текстовых блоков, не учитывающее взаимоотношений между ними, часто приводит к формированию бессвязных рефератов. Некоторые предложения могут оказаться пропущены либо в них могут встречаться слова или фразы, которые невозможно понять без предшествующего пропущенного текста. Попытки решить эту проблему в основном сводятся к исключению таких предложений из рефератов. Реже делаются попытки разрешения ссылок с помощью методов лингвистического анализа.

Краткое изложение содержания первичных документов основывается на выделении из текстов наиболее важной информации и порождении новых текстов, содержательно обобщающих первичные документы. В отличие от частотно-лингвистических методов подход, основанный на базах знаний, опирается на автоматизированный качественный контент-анализ, состоящий, как правило, из трех основных стадий. Первая – сведение исходной текстовой информации к заданному числу фрагментов, которыми являются категории, последовательности и темы. На второй стадии производится поиск регулярных связей между фрагментами, после чего начинается третья стадия – формирование выводов и обобщений. На этой стадии создается структурная аннотация, представляющая содержание текста в виде совокупности концептуально связанных смысловых единиц.

Семантические методы формирования рефератов-изложений предполагают два основных подхода: метод синтаксического разбора предложений и методы, опирающиеся на понимание естественного языка. В первом случае используются деревья разбора текста. Процедуры автоматического реферирования манипулируют непосредственно деревьями, выполняя перегруппировку и сокращение ветвей на основании заданных критериев. Такое упрощение обеспечивает построение реферата, т. е. структурную «выжимку» исходного текста.

Второй подход основывается на системах искусственного интеллекта, в которых также на этапе анализа выполняется синтаксический разбор текста, но синтаксические деревья не порождаются. В этом случае формируются семантические структуры, которые накапливаются в виде концептуальных подграфов в базе знаний. В частности, известны модели, позволяющие производить реферирование текстов на основе психологических ассоциаций сходства и контраста. В базах знаний избыточная и не имеющая прямого отношения к тексту информация устраняется путем отсечения некоторых подграфов. Затем информация подвергается агрегированию методом слияния оставшихся графов или их обобщения. Для выполнения этих преобразований выполняются манипуляции логическими предположениями, выделяются определяющие шаблоны в текстовой базе знаний. В результате преобразования формируется концептуальная структура текста – аннотация, т. е. концептуальные «выжимки» из текста.

Многоуровневое структурирование текста с использованием семантических методов позволяет подходить к решению задачи реферирования путем выполнения следующих операций:

– *удаление малозначащих смысловых единиц.* Преимуществом метода является гарантированное сохранение значащей информации, недостатком – низкая степень сжатия, т. е. сокращения объема реферата по сравнению с первичными документами;

– *сокращение смысловых единиц* – замена их основной лексической единицей, выражающей основной смысл;

– *гибридный способ,* состоящий в уточнении реферата с помощью статистических методов, с использованием семантических классов, особенностей контекста и синонимических связей.

Рассмотрим алгоритм реферирования текстов на базе статистического подхода:

1) согласно законам Зипфа определяются характеристики исходного текста, и строится строка ключевых слов, т. е. терминов текста;

2) фиксируется первое ключевое слово, и в реферат добавляются предложения, содержащие данное слово. При этом сохраняется порядок предложений относительно исходного текста;

3) если достигнут требуемый размер реферата, алгоритм закончен, иначе выбирается следующее ключевое слово, и повторяются шаги 2 и 3;

4) если обработана вся строка ключевых слов, а заданный размер реферата не достигнут, из области значимых слов выбирается первое необработанное слово, а затем выполняется переход на шаг 2.

## ЛАБОРАТОРНАЯ РАБОТА №5

## РЕАЛИЗАЦИЯ ГЕНЕТИЧЕСКОГО АЛГОРИТМА

**Цель работы:** научиться реализовывать генетический алгоритм.

Порядок выполнения работы:

1 Ознакомиться с теоретической частью работы.

2 Применить генетический алгоритм для решения поставленной задачи.

3 Оформить отчет по лабораторной работе.

**Исходные данные:** перечень возможных заданий для реализации генетического алгоритма:

1 Аппроксимировать набор точек линейной функцией $y(x) = a \cdot x + b$. Для этого: а) использовать целочисленное кодирование; б) использовать вещественное кодирование.

2 Аппроксимировать набор точек экспоненциальной функцией $y(x) = a \cdot \exp(b \cdot x)$. Для этого: а) использовать целочисленное кодирование; б) использовать вещественное кодирование.

3 Найти минимум функции $y(x) = x^2 + 4$. Для этого: а) использовать целочисленное кодирование; б) использовать вещественное кодирование.

4 Найти максимум функции $y(x) = 1/x;\ x \in [-4; 0]$. Для этого: а) использовать целочисленное кодирование; б) использовать вещественное кодирование.

5 Найти точку перегиба функции $f(x) = (x - 1{,}5)^3 + 3$. Для этого: а) использовать целочисленное кодирование; б) использовать вещественное кодирование.

6 Найти точку пересечения функции $f(x) = \ln(x + 1) - 2{,}25$, $x > -1$, с осью *OX*. Для этого: а) использовать целочисленное кодирование; б) использовать вещественное кодирование.

7 Сгенерировать с помощью генетического алгоритма слово «МИР».

8. Найти с помощью генетического алгоритма особь, гены которой соответствуют в формате *RGB* фиолетовому цвету (96, 96, 159).

### 5.1 Канонический генетический алгоритм

Рассмотрим пример канонического генетического алгоритма (ГА), который часто используется на практике. Он имеет следующие характеристики:

- целочисленное кодирование;
- все хромосомы в популяции имеют одинаковую длину;
- постоянный размер популяции;
- рулеточная или турнирная селекция;
- одноточечный оператор кроссинговера;
- битовая мутация;
- новое поколение формируется только из особей-потомков (разрыв поколений $T = 1$).

Рассмотрим в качестве примера решение следующей задачи. Требуется найти минимум сферической функции:

$$z = \sum_{i=1}^{n} x_i^2,\ n = 10,\ x_i \in [-5{,}12;\ 5{,}11],\ z \to \min.$$

Параметр $n$ задает количество переменных функции $z$. Необходимо найти такие значения переменных $x_i$, при которых функция $z$ принимает наименьшее значение. Будем использовать общую схему решения.

1 *Определение неизвестных переменных задачи.* По условию задачи необходимо найти значения переменных $x_i$, минимизирующие значение функции $z$, поэтому в хромосоме будем кодировать значения $x_i$. Таким образом, каждый $i$-й ген хромосомы будет соответствовать $i$-й переменной функции $z$.

2 *Задание функции приспособленности.* Будем определять приспособленность особи в зависимости от значения, которое принимает функция *z* при подстановке в нее вектора параметров, соответствующих хромосоме этой особи. Поскольку рассматривается задача минимизации функции z, то будем также считать, что чем меньше значение *z*, тем приспособленнее особь. Приспособленность *i*-й особи $f_i$ будем определять по формуле $f_i = z_i$, где $z_i$ – значение функции *z* в точке, соответствующей *i*-й особи.

3 *Выбор способа кодирования.* В качестве способа представления генетической информации рассмотрим целочисленное кодирование с точностью кодирования параметров 0,01. Тогда в имеющемся по условию задачи диапазоне изменения значений параметров [–5,12; 5,11] можно закодировать (5,12 – (–5,11))/0,01 + 1 = 1024 различных значений переменной. Единица прибавляется, т. к. значение переменной, равное нулю, также учитывается.

Для того чтобы представить 1024 различных значений переменной, достаточно использовать $\log_2 1024 = 10$ бит на каждую переменную. Таким образом, будет использоваться целочисленное кодирование с 10-разрядными генами.

4 *Определение параметров ГА.* Для решения задачи рассмотрим популяцию из 20-ти особей. При отборе особей для скрещивания будем использовать турнирную селекцию с бинарным турниром. В качестве генетических операторов будем использовать одноточечный кроссинговер и битовую мутацию. Вероятности применения операторов скрещивания и мутации установим равными 0,7 и 0,05, соответственно. Новое поколение будем формировать только из особей-потомков, т. е. величина разрыва поколений $T = 1$.

Результат работы генетического алгоритма с выбранными параметрами представлен на рисунке 14. Показаны зависимости изменения среднего $\langle z \rangle$ и наименьшего $z_{\min}$ в популяции значения функции *z* от номера поколения *t*. Данные усреднены по 100 независимым запускам.

По данным рисунка 14 видно, что после 20-го поколения значение $z_{\min}$ колеблется в достаточно большом диапазоне. Из этого следует, что потери хороших особей в результате мутации велики, и следует уменьшить вероятность мутации. Установим значение этого параметра равным $L^{-1} = 0{,}01$, где *L* – длина хромосомы в битах, в данном случае $L = 100$. Результаты работы ГА с измененным значением вероятности мутации показаны на рисунке 15.

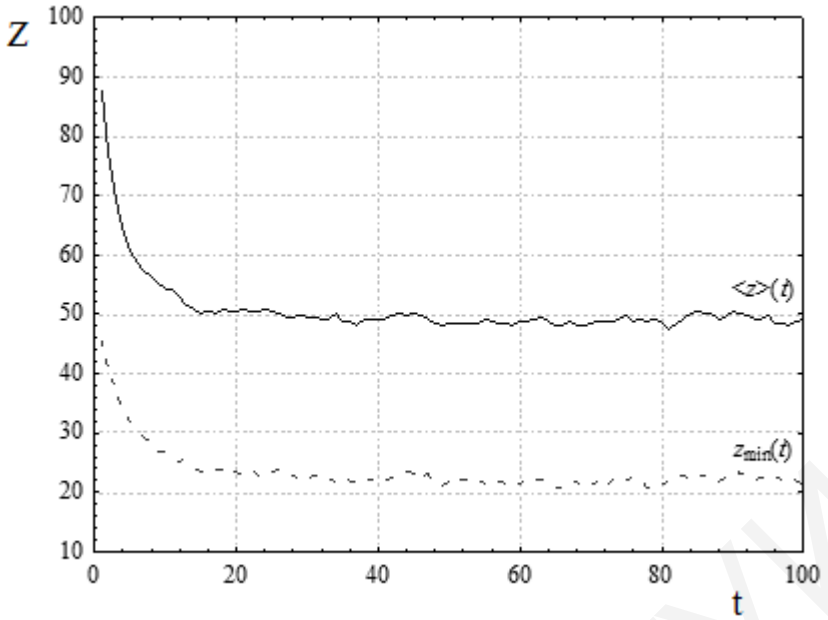



Рисунок 14 – Изменение $z_{min}(t)$ и $<z>(t)$. Популяция из 20-ти особей, турнирный отбор, одноточечный кроссинговер ($P_C$ = 0,7), битовая мутация ($P_M$ = 0,05)

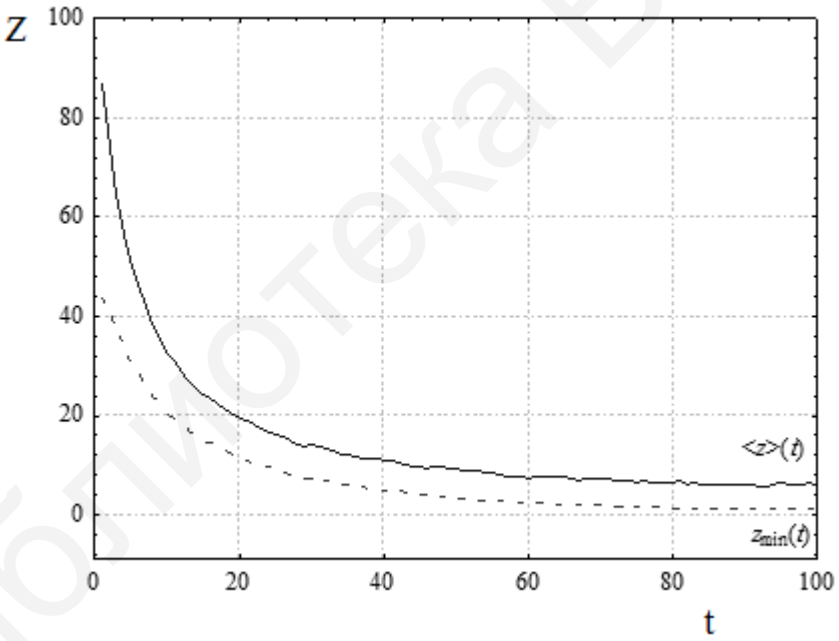



Рисунок 15 – Изменение $z_{min}(t)$ и $<z>(t)$. Популяция из 20-ти особей, турнирный отбор, одноточечный кроссинговер ($P_C$ = 0,7), битовая мутация ($P_M$ = 0,01)

Из сравнения графиков на рисунках 14 и 15 следует, что уменьшение вероятности мутации улучшило результат работы ГА. Также видно, что теперь эволюционный процесс стабилизировался значительно позднее, примерно после 60-го поколения. Усредненное по всем запускам минимальное значение функции $z$, достигнутое за первые 100 поколений, равно ~1,016. Для улучшения результата увеличивается давление селекции путем увеличения размера турнира до 4. Это привело к ускорению эволюционного поиска за счет удаления из популяции особей со средней и плохой приспособленностью.

В результате стабилизация наступила после 40-го поколения, а усредненное по всем запускам минимальное полученное значение функции $z$ равно ~0,013. Наименьшее значение функции $z$ достигается в точке $x_i = 0$, $i = 1, 2, \ldots, 10$ и равно нулю. В случае поиска минимума функции $z$ с точностью 0,01 для ГА решение было найдено в 69 запусках из 100. При этом в среднем было использовано 1698,68 вычислений целевой функции.

Чтобы повысить стабильность результатов, размер популяции увеличивается до 50-ти особей. Полученные кривые $z_{min}(t)$ и $\langle z \rangle(t)$ изображены на рисунке 16. Во всех 100 запусках найден минимум функции $z$ с точностью не меньше 0,01. Среднее количество вычислений целевой функции, использованное для нахождения решения, равно 3145,34.

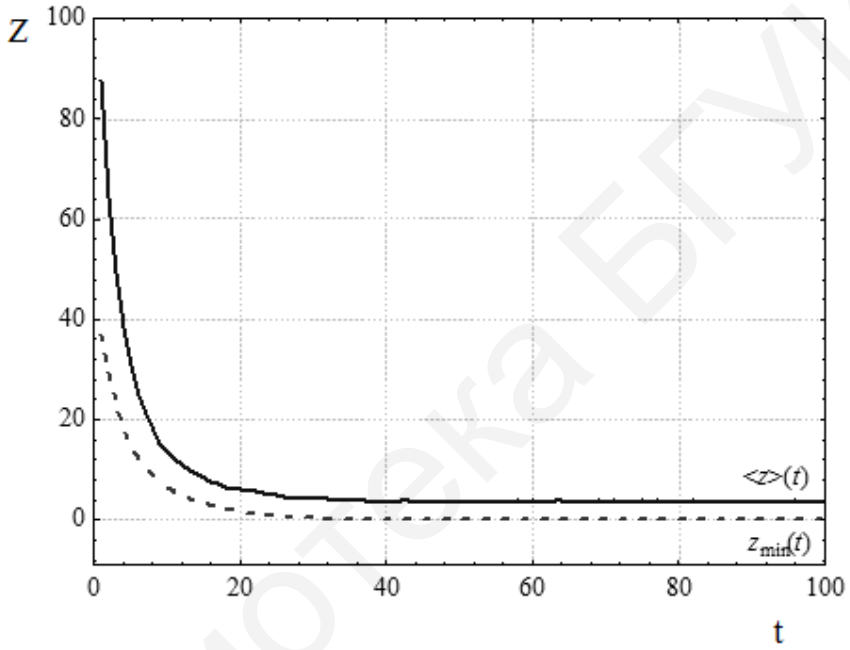



Рисунок 16 – Изменение $z_{min}(t)$ и $\langle z \rangle(t)$. Популяция из 50-ти особей, турнирный отбор, одноточечный кроссинговер ($P_C = 0,7$), битовая мутация ($P_M = 0,01$)

## ЛАБОРАТОРНАЯ РАБОТА №6

## ПРИМЕНЕНИЕ НЕЙРОННОЙ И СЕМАНТИЧЕСКОЙ СЕТЕЙ ДЛЯ ОБРАБОТКИ ТЕКСТОВОЙ ИНФОРМАЦИИ

**Цель работы:** научиться использовать модели нейронной и семантической сетей для работы с текстовой информацией и поиска ответа на вопросы.

Порядок выполнения работы:

1 Ознакомиться с теоретической частью работы.

2 Реализовать алгоритмы ответа на один тип вопроса с использованием нейронной и семантической сетей.

3 Оформить отчет по лабораторной работе.

**Исходные данные:** текстовый документ и вопрос, сформулированный в текстовом виде.

**Выходные данные:** ответ на вопрос в виде текста.

## 6.1 Персептрон и сеть Хопфилда

Решение проблемы распознавания образов с помощью ИНС состоит из двух процедур: обучения и непосредственно самого распознавания незнакомых образов.

Процедура поиска решения задачи с помощью сети, прошедшей обучение, оказывается более гибкой, чем использование других вычислительных средств, поскольку ИНС может повышать точность результатов по мере накопления ею опыта и адаптироваться к происходящим изменениям.

Одной из наиболее популярных моделей ИНС с контролируемым обучением считается ИНС в виде многослойного персептрона (рисунок 17).

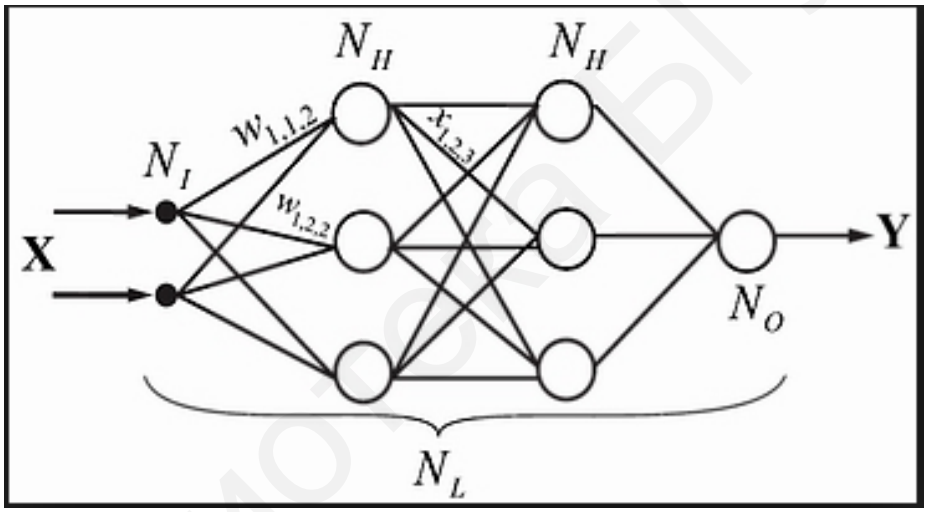



Рисунок 17 – ИНС в виде многослойного персептрона

Нейроны могут объединяться в сети различным образом. Сеть состоит из произвольного количества слоев нейронов, в простейшем случае – однослойная сеть. Первый слой называется сенсорным, или входным, внутренние слои называются скрытыми, или ассоциативными, последний – выходным, или результативным. Количество нейронов в слоях может быть произвольным. Обычно во всех скрытых слоях одинаковое количество нейронов. В каждом слое выполняется нелинейное преобразование линейной комбинации сигналов предыдущего слоя. Следовательно, в тех сетях, где требуется последовательное соединение слоев нейронов друг за другом, необходима нелинейная функция активации. В противном случае многослойность оказывается ненужной, т. к. ее можно заменить эквивалентной однослойной сетью с соответствующими весовыми коэффициентами. Многослойная сеть может формировать на выходе произвольную многомерную функцию при соответ-

ствующем выборе количества слоев, диапазона изменения сигналов и параметров нейронов.

Введем обозначения согласно рисунку 17. Входной слой состоит из $N_I$ нейронов; каждый скрытый слой содержит по $N_H$ нейронов; $N_O$ – количество выходных нейронов; $x$ – вектор входных сигналов сети, $y$ – вектор выходных сигналов.

Входной слой не выполняет никаких вычислений, а лишь распределяет входные сигналы, поэтому иногда его не учитывают, считая количество слоев в сети. Обозначим через $N_L$ полное количество слоев в сети, считая и входной.

Работа многослойного персептрона описывается следующими формулами:

$$NET_{jl} = \sum_{i=1}^{n} w_{ijl} x_{ijl},$$

$$OUT_{jl} = F(NET_{jl} - \theta_{jl}),$$

$$x_{ij(l+1)} = OUT_{il},$$

где $i$ – номер входа; $j$ – номер нейрона в слое; $l$ – номер слоя; $x_{ijl}$ – $i$-й входной сигнал $j$-го нейрона в слое $l$; $w_{ijl}$ – весовой коэффициент $i$-го входа $j$-го нейрона в слое $l$; $NET_{jl}$ – сигнал NET $j$-го нейрона в слое $l$; $F$ – функция активации; $OUT_{jl}$ – выходной сигнал нейрона; $\theta_{jl}$ – пороговый уровень $j$-го нейрона в слое $l$.

Введем еще некоторые обозначения: $w_{jl}$ – вектор-столбец весов для всех входов нейрона $j$ в слое $l$; $W_l$ – матрица весов всех нейронов слоя $l$. В столбцах матрицы расположены векторы $w_{jl}$; $x_{jl}$ – входной вектор-столбец слоя $l$.

В каждом слое рассчитывается нелинейное преобразование от линейной комбинации сигналов предыдущего слоя.

Многослойные персептроны успешно применяются для решения многих сложных задач, в том числе и для семантического анализа текстов. Нейронные сети позволяют преобразовать смысл текста в сложную математическую функцию, представленную в виде графа.

Алгоритм обучения персептрона часто называют алгоритмом обратного распространения ошибки. Целью обучения сети алгоритмом обратного распространения ошибки является такая корректировка ее весов, чтобы от некоторого множества входов можно было перейти к требуемому множеству выходов. Эти множества входов и выходов называются векторами. При обучении предполагается, что для каждого входного вектора существует парный ему целевой вектор, задающий требуемый выход. Вместе они называются обучающей парой. Сеть обучается на многих парах.

Предполагается два прохода по всем слоям сети: прямой и обратный. При прямом проходе входной вектор подается на входной слой нейронной

сети, после чего распространяется по сети от слоя к слою. В результате генерируется набор выходных сигналов, который и является фактической реакцией сети на данный входной образ. Фактически решение задачи – это определение весовых коэффициентов вектора значений. Во время прямого прохода все веса сети фиксированы. Во время обратного прохода все веса настраиваются в соответствии с правилом коррекции ошибок, а именно: фактический выход сети вычитается из желаемого, в результате чего формируется сигнал ошибки. Этот сигнал впоследствии распространяется по сети в направлении, обратном направлению связей между нейронами. Весовые коэффициенты настраиваются с целью максимального приближения выходного сигнала сети к желаемому.

В качестве активационной функции в многослойных персептронах, как правило, используется сигмоидальная активационная функция, в частности, бинарный сигмоид. Для эффективности вычислений желательно, чтобы производная активационной функции легко определялась. Рассмотрим функцию

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

ее производная имеет вид $f'(x) = f(x) \cdot [1 - f(x)]$.

На рисунке 18 приведена функция «Бинарный сигмоид».

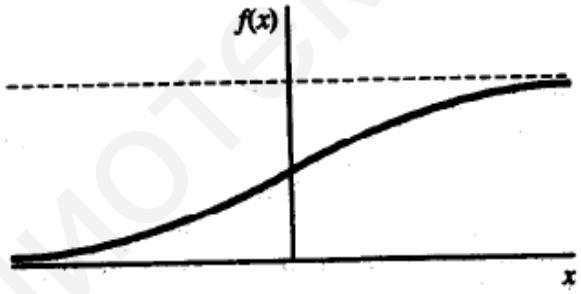



Рисунок 18 – Бинарный сигмоид

Рассмотрим алгоритм обратного распространения ошибки:

1) инициализировать веса маленькими случайными значениями;

2) выбрать очередную обучающую пару из обучающего множества и подать входной вектор на вход сети;

3) вычислить выход сети;

4) вычислить разность между выходом сети и требуемым выходом (целевым вектором обучающей пары);

5) подкорректировать веса сети для минимизации ошибки;

6) повторять шаги с 2 по 5 для каждого обучающего вектора до тех пор, пока ошибка на всем множестве не достигнет приемлемого уровня.

Операции, выполняемые шагами 2 и 3, сходны с теми, которые выполняются при функционировании уже обученной сети, т. е. подается входной вектор и вычисляется получающийся выход. Вычисления выполняются по-

слойно. Шаги 2 и 3 образуют так называемый «проход вперед», т. к. сигнал распространяется по сети от входа к выходу. Шаги 4 и 5 составляют «обратный проход», вычисляемый сигнал ошибки распространяется обратно по сети и используется для настройки весов.

Достоинством алгоритма обратного распространения ошибки является его универсальность с точки зрения применения, т. к. обычно он используется для определения класса объекта, которому соответствует входной сигнал. Кроме того, алгоритм реализует вычислительно эффективный метод обучения многослойного персептрона. При обучении сети нужно заранее определить множество взаимоисключающих классов.

Основным недостатком данного алгоритма является неопределенно долгий процесс обучения. В сложных задачах для обучения сети могут потребоваться дни или даже недели, после чего она может и вообще не обучиться. Разбор доказательства сходимости показывает, что коррекции весов предполагаются бесконечно малыми, что неосуществимо на практике, т. к. ведет к бесконечному времени обучения. Размер шага должен быть конечным. Если размер шага фиксирован и очень мал, то сходимость слишком медленная, если же он фиксирован и слишком велик, то может возникнуть постоянная неустойчивость.

Рассмотрим модель нейронной сети, которая позволяет определить не одну характеристику входного сигнала (класс, к которому он относится), а сразу несколько характеристик – это сеть Хопфилда. Она занимает особое место в ряду ИНС. В ней впервые удалось установить связь между нелинейными динамическими системами и нейронными сетями. Образы памяти сети соответствуют устойчивым предельным точкам (аттракторам) динамической системы. При этом появилась возможность теоретически оценить объем памяти сети Хопфилда, определить область ее параметров, в которой достигается наилучшее функционирование.

В общем случае модель Хопфилда может быть представлена сетью с одним слоем, содержащей произвольные обратные связи, по которым переданное возбуждение возвращается к данному нейрону, и он повторно выполняет свою функцию (рисунок 19).

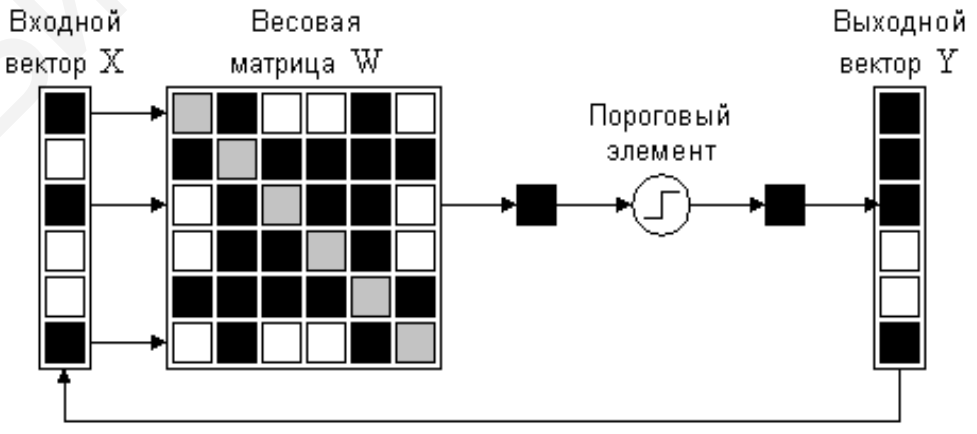



Рисунок 19 – Нейронная сеть Хопфилда

Обратные связи могут вызывать неустойчивости в поведении ИНС, что проявляется в блуждающей смене состояний нейронов, не приводящей к стационарным состояниям. Однако было показано, что сети Хопфилда устойчивы, и их можно определить как динамическую систему с обратной связью, у которой выход одной операции служит входом следующей операции сети. Каждая операция сети называется итерацией. Устойчивость сети подразумевает, что она может сходиться к одной из зафиксированных (неподвижных) точек, которая зависит от исходной точки, выбранной для начальной итерации. Множество неподвижных точек сети Хопфилда – это ее память. В этом случае сеть может действовать как ассоциативная память.

Алгоритм сходимости сети Хопфилда связан с вычислением элементов $y_j, j = 1, 2, \dots, n$ выходного вектора $Y$ по формуле

$$y_j = T(\sum_{i=1}^{n} w_{ij} x_i), \tag{6}$$

где функция активации имеет вид $T(x) = \begin{cases} -1, \text{если } x < 0, \\ 1, \text{если } x > 0, \\ 0, \text{ если } x = 0. \end{cases}$

Если входной вектор $X$ имеет вид $X = (x_1, x_2, x_3, \dots x_n)$, то из него необходимо строить вектор $Y = (y_1, x_2, x_3, \dots, x_n)$, постепенно заменяя $x_j$ на $y_j$ по формуле (6). Полученный вектор $Y$ подается на вход сети в качестве входных данных, после чего вычисляется очередной ($j = j + 1$) элемент вектора $Y$. Процесс продолжается до тех пор, пока вектор $Y$ не перестанет изменяться. Каждый шаг уменьшает величину энергии связей, поэтому обеспечивается сходимость к неподвижной точке (аттрактору), в которой достигается наименьшее значение некоторой функции энергии.

Асинхронная коррекция и нули на диагонали матрицы весовых коэффициентов $W$ гарантируют, что энергетическая функция будет уменьшаться с каждой итерацией. Весовая матрица отличает поведение одной сети Хопфилда от другой.

К недостаткам сети Хопфилда можно отнести ее сравнительно небольшой объем памяти, вследствие чего попытка записи большего числа образов приводит к тому, что нейронная сеть перестает их распознавать. Кроме того, достижение устойчивого состояния не гарантирует правильный ответ сети. Это происходит из-за того, что ИНС может сойтись к так называемым ложным аттракторам.

Для обучения сети Хопфилда составляется обучающая выборка. Это какие-то из характеристик знакомых текстов. При обучении можно рассчитать коэффициенты матрицы связей на основании слов обучающих текстов. В ходе решения задачи на обученной сети входной сигнал после выполнения некоторого количества итераций будет приводить сеть в стационарное состояние, соответствующее образу, на котором проводилось обучение.

Сеть Хопфилда имеет довольно большой диапазон способов применения. Она может быть использована как автоассоциативная память (хранит

образы, но не ассоциирует их друг с другом), как фильтр (для исправления сигнала), а также для решения некоторых задач оптимизации (поиск решений с меньшим значением функции энергии).

При работе с текстами в качестве входов нейронной сети можно использовать некоторое множество слов обучающей выборки, при этом наличие слова в тексте означает единичный сигнал на входе нейронной сети. В таком случае нейронная сеть Хопфилда позволяет восстанавливать текст, который был сохранен в сети. Восстановленный текст может отличаться от текста, поданного на вход сети, на одно или несколько слов.

## 6.2 Применение нейронной сети Хопфилда для ответа на вопрос

Современные информационные системы предназначены для работы в автоматическом режиме с большими объемами текстовой информации. Рассмотрим применение ИНС для поиска ответов на различные вопросы, сформулированные на естественном (русском) языке. В русском языке выделяют пять основных видов вопросов: закрытые, открытые, риторические, переломные, вопросы для обдумывания.

В качестве примера разберем алгоритм поиска ответа на открытый вопрос – это вопрос, требующий разъяснения. Например, «что?», «где?», «как?», «для чего?». Если информационная система умеет отвечать на различные типы вопросов, то пользователь освобождается от части рутинной работы, которую за него сможет выполнять система.

Рассмотрим применение нейронной сети Хопфилда для поиска ответа на вопрос. Пусть каждому входу сети соответствует некоторое слово (или понятие). Наличие слова в предложении может быть обозначено как наличие сигнала на соответствующем входе. Каждому предложению текста можно поставить в соответствие некоторый вектор.

Порядок использования нейронной сети Хопфилда для ответа на вопрос может быть представлен следующим образом:

1) выбрать параметры сети (размерность входного вектора – это количество уникальных слов текста, множество сохраненных в сети векторов – это предложения некоторого текста);

2) рассчитать весовую матрицу на основании параметров;

3) подать на вход сети распознаваемое предложение (вопрос);

4) выполнять перерасчет выходного вектора сети, пока он не станет постоянным. Ответ будет содержаться в получившемся выходном векторе.

Предположим, в тексте есть три предложения: «Дождь идет на улице», «Из-за дождя появились лужи на асфальте», «Завтра будет солнечно». Покажем, что можно построить модель нейронной сети Хопфилда, позволяющую отвечать на следующие виды вопросов (таблица 1).

Таблица 1 – Исходные данные

| **Вид вопроса** | **Что выступает в качестве ответа** | **Признак, который можно использовать для поиска ответа** |
|---|---|---|
| Кто/Что? | Объект, выполняющий действие | Существительное в начальной форме |
| Где? | Указание места | Пространственный предлог + существительное не в начальной форме |
| Как? | Указание, каким образом произведено действие | Наречие образа действия |
| Для чего? | Указание, с какой целью производится действие | Предлог «для» + существительное не в начальной форме |

Для определения того, к какой части речи принадлежит конкретное слово, можно сравнить основу данного слова с основами слов, чья принадлежность уже установлена. Существуют более универсальные и/или точные способы определения частей речи, но предложенные способы достаточны для демонстрации работы данного алгоритма.

Пусть списки заранее известных слов выглядят следующим образом (таблица 2):

Таблица 2 – Списки известных слов

| **Вид списка** | **Содержимое списка** |
|---|---|
| Существительные | Дождь, улица, лужа, асфальт |
| Пространственные предлоги | На (для решения данного примера достаточно знания об одном предлоге) |
| Наречия образа действия | Солнечно (для решения данного примера достаточно знания об одном наречии) |

Для определения основы слова можно использовать один из алгоритмов стемминга, например, алгоритм Портера. Для простоты в качестве алгоритма стемминга применим удаление конечной части слова.

Для существительных в качестве окончаний могут выступать следующие сочетания букв:

а|ев|ов|ие|ье|е|иями|ями|ами|еи|ии|и|ией|ей|ой|ий|й|и|иям|ям|ием|ем|ам|ом|о|у|ах|иях|ях|ы|ь|ию|ью|ю|ия|ья|я.

Так, для слова «улица» в словаре существительных после удаления окончания «а» получаем основу «улиц». Для слова «улице» из предложения «Дождь идет на улице» после удаления окончания «е» получим также основу «улиц». Следовательно, «улице» – это существительное. Аналогичным образом выполняется анализ того, какие суффиксы и окончания могут быть у глаголов.

Для слова «солнечно» в предложении «Завтра будет солнечно» после удаления окончания «о» останется основа «солнечн», которая не совпадает с основой ни одного существительного. Значит, это не существительное. Более того, слово «солнечно» находится в словаре наречий образа действия.

Построим ИНС Хопфилда, позволяющую отвечать на вопрос «Где идет дождь?». Для этого будем использовать следующий алгоритм:

1 Выбрать параметры сети (размерность входного вектора – количество уникальных слов текста, множество сохраненных в сети векторов – предложения некоторого текста).

В качестве слов, которым соответствуют входные сигналы сети, выбираются все уникальные слова предложений, кроме предлогов. Затем применяется алгоритм нахождения основы слова для существительных, чтобы учитывать разные формы существительного как одно и то же слово. Для упрощения глаголы учтем в той же форме, в которой они представлены в тексте. В результате получен следующий ряд слов: *дожд, идет, улиц, из-за, появились, луж, асфальт, завтра, будет, солнечно.*

Поскольку требуется отвечать на вопрос определения места, то в список слов необходимо добавить указания мест. В соответствии с таблицей ответ на вопрос «где?» подразумевает отыскание пространственного предлога и существительного не в начальной форме. Слово «на» есть в словаре пространственных предлогов. Как уже было определено, «улице» – это существительное.

Для определения того, что существительное находится в начальной форме, его можно сравнить со словом в списке существительных. Если совпадений нет, то это не начальная форма. Таким образом, «улице» – это существительное не в начальной форме, а «на + улице» – это сочетание, которое может быть использовано для ответа на вопрос «где?».

Если необходимо обрабатывать существительные во множественном числе, то начальной формой можно также считать основу слова из списка с добавленной буквой «и» либо «ы». Тогда слово «дожди» в тексте будет воспринято как начальная форма.

Изложенным способом во втором предложении можно найти сочетание «на + асфальте» как возможный ответ на поставленный вопрос. Добавив найденные сочетания в список слов, которым соответствуют входные сигналы сети, получаем следующий список: *дожд, идет, улиц, из-за, появились, луж, асфальт, завтра, будет, солнечно, на улице, на асфальте*.

При необходимости того, чтобы сеть умела отвечать на вопрос «как?», в предложениях текста требуется найти наречия образа действия (используя список наречий образа действия). Повторно найденные слова не добавляются в список. Пример полученных векторов предложений представлен на рисунке 20.

2 Рассчитать весовую матрицу на основании параметров.

Весовая матрица для нейронной сети Хопфилда должна быть построена с учетом трех рассмотренных предложений. Наличие слова в предложении обозначается как сигнал «1» на соответствующем входе сети, отсутствие слова – «–1».

У сети 12 входных сигналов. Рассчитаем элементы матрицы 12x12, описывающей связи нейронов, по формуле

$$w_{ij} = \begin{cases} \sum_{k=1}^{N} x_{k_i} x_{kj}, \text{для } i \neq j, \\ 0, \qquad \text{для } i = j. \end{cases}$$

Здесь $k$ – номер запоминаемого вектора. Первый элемент первой строки ($i = j = 1$) равен нулю. Второй элемент первой строки равен сумме произведений 1-го элемента на 2-й элемент для всех трех векторов:

$$1 \cdot 1 + 1 \cdot (-1) + (-1) \cdot (-1) = 1.$$

Также, например, 3-й элемент 5-й строки равен сумме произведений 3-го элемента на 5-й для всех трех векторов:

$$1 \cdot (-1) + (-1) \cdot 1 + (-1) \cdot (-1) = -1.$$

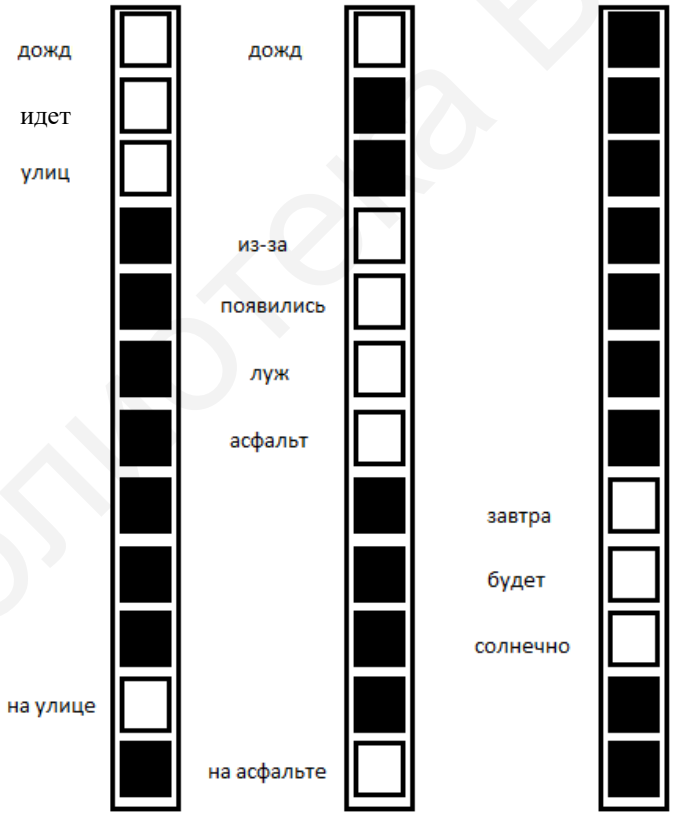



Рисунок 20 – Векторы для заданных предложений

В результате для слов, которые относятся к одному образу (предложению), связь имеет более высокий вес, чем для слов, которые никогда не относятся к одному и тому же образу.

3 Подать на вход сети распознаваемое предложение (вопрос).

Рассмотрим ответ на вопрос «Где идет дождь?» с помощью нейронной сети, основанной на рассчитанной матрице весов связей. Для каждого слова вопроса, имеющего соответствующий входной сигнал сети, на вход подается «1». На остальные входы подается «–1».

Если в вопросе будут слова, для которых нет соответствующих входов сети, то они игнорируются.

4 Выполнить перерасчет выходного вектора сети, пока он не станет постоянным.

В итоге получим следующую матрицу.

| 0 | 1 | 1 | 1 | 1 | 1 | 1 | -3 | -3 | -3 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 3 | –1 | –1 | –1 | –1 | –1 | –1 | –1 | 3 | –1 |
| 1 | 3 | 0 | –1 | –1 | –1 | –1 | –1 | –1 | –1 | 3 | –1 |
| 1 | –1 | –1 | 0 | 3 | 3 | 3 | –1 | –1 | –1 | –1 | 3 |
| 1 | –1 | –1 | 3 | 0 | 3 | 3 | –1 | –1 | –1 | –1 | 3 |
| 1 | –1 | –1 | 3 | 3 | 0 | 3 | –1 | –1 | –1 | –1 | 3 |
| 1 | –1 | –1 | 3 | 3 | 3 | 0 | –1 | –1 | –1 | –1 | 3 |
| –3 | –1 | –1 | –1 | –1 | –1 | –1 | 0 | 3 | 3 | –1 | –1 |
| –3 | –1 | –1 | –1 | –1 | –1 | –1 | 3 | 0 | 3 | –1 | –1 |
| –3 | –1 | –1 | –1 | –1 | –1 | –1 | 3 | 3 | 0 | –1 | –1 |
| 1 | 3 | 3 | –1 | –1 | –1 | –1 | –1 | –1 | –1 | 0 | –1 |
| 1 | –1 | –1 | 3 | 3 | 3 | 3 | –1 | –1 | –1 | –1 | 0 |

Обучение нейронной сети Хопфилда является довольно быстрым процессом, при котором необходимо лишь рассчитать весовую матрицу. Однако ее использование для распознавания образа подразумевает повторение множества итераций, необходимых для учета обратных связей, и в общем случае это может занимать значительное время.

Рассмотрим синхронный режим работы сети, т. е. все сигналы на выходе сети рассчитываются одновременно и не влияют друг на друга в рамках одной итерации. Условием окончания итераций будем считать совпадение текущего выходного вектора с выходным вектором, полученным на одной из предыдущих итераций. Если выходные векторы двух последовательных итераций совпадают, то сеть сошлась к некоторому образу. В случае если полученный в результате итерации выходной вектор совпадает с выходным вектором одной из предыдущих итераций (но не последней итерации), то это означает, что несколько выходных векторов бесконечно сменяют друг друга, и сеть не сумела восстановить образ.

*Итерация 1*. Рассматриваемый входной вектор сети приведет к появлению сигнала на выходе, который рассчитывается по формуле

$$y_i = T\left(\sum_{i=1}^{n} w_{ij} x_i\right),$$

где $T(x) = \text{sign}(x)$, т. е. для получения сигнала на первом выходе надо перемножить первый столбец матрицы на входной вектор:

$$T(0\cdot 1 + 1\cdot 1 + 1\cdot(-1) + 1\cdot(-1) + 1\cdot(-1) + 1\cdot(-1) + 1\cdot(-1) + (-3)\cdot(-1) + (-3)\cdot(-1) + \\ + (-3)\cdot(-1) + 1\cdot(-1) + 1\cdot(-1)) = T(3) = 1.$$

На втором выходе будет

$$T(1\cdot 1 + 0\cdot 1 + 3\cdot(-1) + (-1)\cdot(-1) + (-1)\cdot(-1) + (-1)\cdot(-1) + (-1)\cdot(-1) + (-1)\cdot(-1) + (-1)\cdot(-1) + \\ + (-1)\cdot(-1) + (-1)\cdot(-1) + 3\cdot(-1) + (-1)\cdot(-1)) = T(3) = 1.$$

Аналогично вычисляются значения на остальных выходах. После первой итерации на выходе будет получен вектор, совпадающий с первым предложением.

*Итерация 2.* Для получения значений на выходах сети в результате второй итерации на вход сети подается результат первой итерации.

Для получения сигнала на первом выходе надо перемножить первый столбец матрицы на входной вектор:

$$T(0\cdot 1 + 1\cdot 1 + 1\cdot 1 + 1\cdot(-1) + 1\cdot(-1) + 1\cdot(-1) + 1\cdot(-1) + (-3)\cdot(-1) + (-3)\cdot(-1) + \\ + (-3)\cdot(-1) + 1\cdot 1 + 1\cdot(-1)) = T(7) = 1.$$

На второй итерации на втором выходе будет

$$T(1\cdot 1 + 0\cdot 1 + 3\cdot 1 + (-1)\cdot(-1) + (-1)\cdot(-1) + (-1)\cdot(-1) + (-1)\cdot(-1) + (-1)\cdot(-1) + \\ + (-1)\cdot(-1) + (-1)\cdot(-1) + 3\cdot 1 + (-1)\cdot(-1)) = T(15) = 1.$$

Рассчитав значения на каждом выходе сети, получим вектор, который был получен на предыдущей итерации. Это означает, что на основании вопроса «Где идет дождь?» сеть вернула данный вектор.

5 Ответ будет содержаться в получившемся выходном векторе.

«Где?» – вопрос определения места. Согласно таблице ответом на данный вопрос является «пространственный предлог + существительное не в начальной форме». Среди всех активных сигналов получившегося вектора, только один сигнал соответствует данному сочетанию слов: «на улице». Это означает, что ответом является «на улице». Если бы не было ни одного выходного сигнала, соответствующего данному вопросу, то это бы означало, что ответ не найден. Если таких выходных сигналов несколько, то ответов несколько.

Множество входных и выходных сигналов можно дополнить сигналами, соответствующими формам слов, которые используются в тексте. Например, помимо входа «дожд» добавить вход «дождь», сигнал на котором равен единице только, если в предложении есть слово «дождь» в начальной форме. В таком случае множества выходных сигналов сети может быть до-

статочно, чтобы в ответ на поисковый запрос выдавать сразу все предложение целиком (сигналы «дождь», «идет», «на улице»).

Таким образом, модель нейронной сети может быть использована практически без дополнительных приспособлений для решения задачи ответа на вопрос. Она позволяет сохранить некоторое множество векторов (образов), корректировать их, а также находить недостающие элементы входных образов, т. е. эта модель позволяет дополнять высказывания недостающими словами.

Однако если в вопросе вместо слов текста будут их синонимы, то для получения корректного результата нужно все слова рассматривать в контексте их связей со словами-синонимами. Для этого нужен отдельный этап алгоритма, который будет учитывать синонимичность.

Одной из моделей, которая позволяет описывать различные связи между понятиями, такие, как синонимичность, место, образ действия и т. п., является семантическая сеть.

### 6.3 Поиск ответа на вопрос с помощью семантической сети

Количество типов отношений в семантической сети определяется исходя из ее назначения, и в принципе ничем не ограничено. Рассмотрим задачу ответа на вопрос, сформулированный на естественном языке.

Предположим, есть три предложения: «На стадионе проводится матч», «Из-за матча в городе опустели улицы», «Скоро будет многолюдно».

Покажем, что можно построить модель семантической сети, позволяющую отвечать на виды вопросов, приведенные в таблице 1 (см. подраздел 6.2).

Для определения части речи каждого слова будем использовать тот же подход, что и для ИНС.

Пусть используемые списки слов выглядят следующим образом (таблица 3).

Таблица 3 – Используемые списки слов

| **Вид списка** | **Содержимое списка** |
|---|---|
| Существительные | Матч, стадион, город, улица |
| Глаголы | Проводить, опустеть, быть, будет |
| Пространственные предлоги | В, на (*для решения данного примера достаточно знания об одном предлоге*) |
| Причинные предлоги | Из-за (*для построения более полной модели семантической сети в данном случае необходимо знание об этом предлоге*) |
| Наречия образа действия | Многолюдно (*для решения данного примера достаточно знания об одном наречии*) |
| Словарь синонимов | Матч – игра, улица – дорога |

Слова «быть» и «будет» – разные формы одного слова, но при помощи предложенного подхода это определить нельзя. Поэтому слово «будет» добавлено в список глаголов.

Алгоритм построения семантической сети, используемой для ответа на вопрос, может быть представлен следующим образом:

1 Определить типы связей, необходимые для ответа на поддерживаемые виды вопросов.

2 Перейти к очередному предложению текста, начиная с первого.

3 Добавить в сеть связи «подлежащее – сказуемое». В качестве подлежащего для простоты можно использовать существительное в начальной форме. В качестве сказуемого – глагол.

4 Добавить в сеть типы связей, определенные на шаге 1.

5 Добавить в сеть связи «слово – основа слова». Основу слова можно определить с помощью алгоритма стемминга, как описано выше.

6 Добавить в сеть связи вида «синоним», соединяющие основы слов.

7 Повторить шаги 3–6 для каждого предложения текста.

Рассмотрим работу этого алгоритма на примере трех выбранных предложений.

1 *Определить типы связей, необходимые для ответа на поддерживаемые виды вопросов.*

Пусть сеть будет использоваться для ответа на вопросы «где?», «как?», «из-за чего?». Этим вопросам соответствуют типы связей «место», «образ действия», «причина действия».

2 *Перейти к очередному предложению текста, начиная с первого.*

Первое предложение – «На стадионе проводится матч».

3 *Добавить в сеть связи «подлежащее – сказуемое». В качестве подлежащего для простоты можно использовать существительное в начальной форме. В качестве сказуемого – глагол* (рисунок 21).

Слово «матч» – существительное в начальной форме, «проводится» – глагол. Получаем сеть:

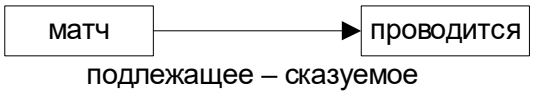



Рисунок 21 – Построение семантической сети. Шаг 3

Если в предложении обнаружено несколько подлежащих или сказуемых, то в сети для данного предложения будет несколько связей «подлежащее – сказуемое».

4 *Добавить в сеть типы связей, определенные на шаге* 1 (рисунок 22):

а) место (соответствует вопросу «где?»). На данном этапе (согласно таблице 1, см. подраздел 6.2) надо найти в предложении пару «пространственный предлог + существительное не в начальной форме». «На» – пространственный предлог, «стадионе» – следующее за ним существительное не в начальной форме (в списке существительных есть начальная форма «стадион»). Добавим

«на стадионе» в качестве обстоятельства места. Лучше связать данное место и с объектом, совершающим действие, и с самим действием;

б) образ действия (соответствует вопросу «как?»). В данном предложении нет наречий образа действия. Сеть не изменяется;

в) причина действия (соответствует вопросу «из-за чего?»). На данном этапе надо найти в предложении пару «причинный предлог + существительное не в начальной форме». В данном предложении причинные предлоги не найдены. Сеть не изменяется.

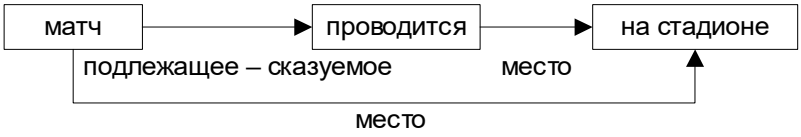



Рисунок 22 – Построение семантической сети. Шаг 4

5 *Добавить в сеть связи «слово – основа слова»* (рисунок 23).

Используя описанный выше алгоритм для существительных и глаголов, получим основы: «на стадионе» (игнорируя предлог) – «стадион», «проводится» – «провод», «матч» – «матч». Слово «матч» совпадает со своей основой. Два узла с совпадающим текстом были бы излишни с точки зрения поставленной задачи ответа на вопрос. Поэтому новый узел и связь не добавляются. Сеть примет следующий вид:

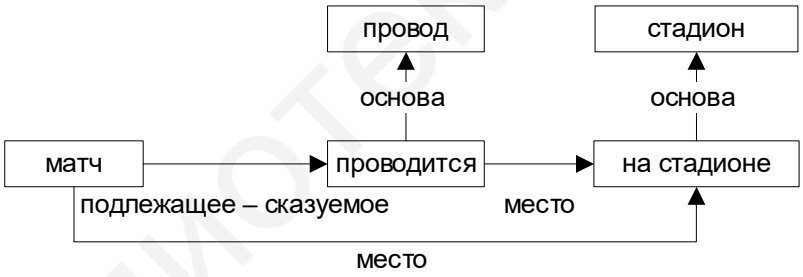



Рисунок 23 – Построение семантической сети. Шаг 5

6 *Добавить в сеть связи вида «синоним», соединяющие основы слов* (рисунок 24).

Используя словарь синонимов, определим, что «матч» и «игра» – синонимы, т. е. узлы, в которых записаны основы этих слов, их можно связать с помощью связи «синоним». Получим следующую сеть:

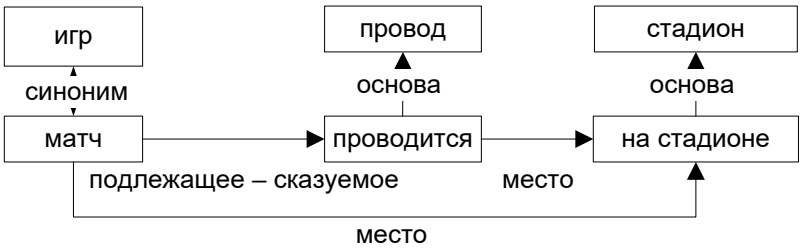



Рисунок 24 – Построение семантической сети. Шаг 6

7 *Повторить шаги 3–6 для каждого предложения текста* (рисунок 25).

Обработав аналогичным образом предложение «Из-за матча в городе опустели улицы», получим сеть:

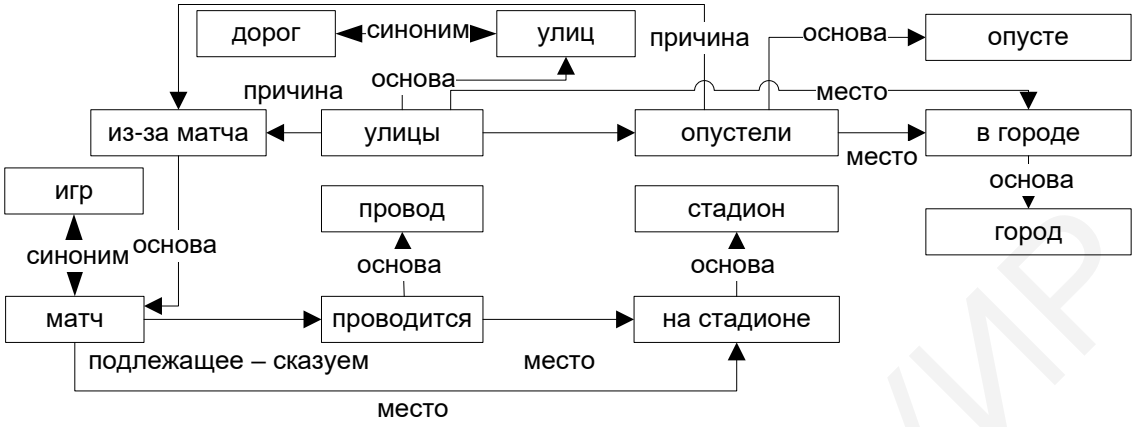



Рисунок 25 – Построение семантической сети. Шаг 7

Здесь подлежащее – улицы, сказуемое – опустели, место – в городе.

В данном предложении нет наречий образа действия. Причина действия («причинный предлог + существительное не в начальной форме») – «из-за матча». Основы: «улицы» – «улиц», «опустели» – «опусте», «из-за матча» – «матч», «в городе» – «город». Синонимы: «улица» – «дорога» (добавляется связь «улиц» – «дорог»).

Аналогичным образом обрабатывается предложение «Скоро будет многолюдно». Существительного из списка в нем нет. В предложение входят глагол «будет» – это сказуемое, наречие образа действия «многолюдно», наречие времени «скоро». Согласно алгоритму в качестве основы слова «будет» определяется часть «буд» и добавляется в сеть. Слова «скоро» нет в списках известных слов, и оно не было добавлено в сеть в результате какого-либо шага алгоритма. Третье предложение не имеет связей с другими частями сети.

На основании трех предложений получается семантическая сеть, представленная на рисунке 26.

Алгоритм применения семантической сети для ответа на вопрос может быть представлен следующим образом:

1 Найти фрагмент сети, связывающий основы подлежащего и сказуемого вопроса. Если подлежащего или сказуемого нет, то фрагмент состоит из одного слова (главный член предложения, который есть в вопросе).

2 Убрать в найденном фрагменте сети синонимы и повторения разных форм одного слова.

3 В зависимости от типа вопроса выбрать дополнительные слова, связанные нужным типом связи с наибольшим количеством слов найденного фрагмента.

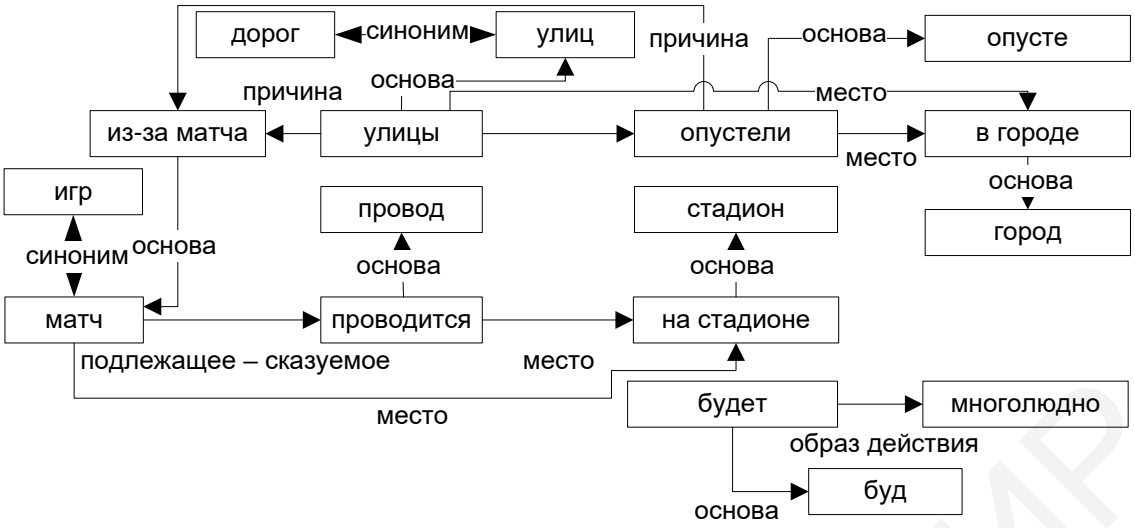



Рисунок 26 – Построенная семантическая сеть

Рассмотрим применение алгоритма для построенной семантической сети.

1 *Найти фрагмент сети, связывающий основы подлежащего и сказуемого вопроса. Если подлежащего или сказуемого нет, то фрагмент состоит из одного слова (главный член предложения, который есть в вопросе).*

Вопрос – «где проводятся игры?». Как описано выше, слово «игры» представляет собой основу существительного из списка с добавленной буквой «ы», поэтому оно воспринимается как начальная форма. Подлежащее – существительное в начальной форме «игры». Основа подлежащего – «игр». Сказуемое – «проводятся», основа которого «провод», есть в сети.

Во время поиска фрагмента сети, соединяющего слова вопроса, неважно, в каком направлении указана связь. Направление связи используется для определения роли связанных слов и не влияет на тот факт, что слова связаны по смыслу. Поэтому семантическую сеть можно рассматривать как неориентированный граф. Применим для ответа на вопрос алгоритм поиска в ширину. В результате будет найден путь «провод» – «проводится» – «матч» – «игр».

2 *Убрать в найденном фрагменте сети синонимы и повторения разных форм одного слова.*

В найденном фрагменте сети дублируются слова: синонимы и разные формы одного слова. Для формирования ответа, понятного пользователю, он строится из слов исходного текста.

Последовательно проанализируем связи между узлами найденного фрагмента сети. В найденном пути первый узел «провод» связан со вторым узлом «проводится» связью «слово – основа слова», поэтому он отбрасывается. У слова «проводится» две связи (с предыдущим и последующим узлами найденного фрагмента), одна из которых – «подлежащее – сказуемое». Слово добавляется в ответ.

У слова «матч» также две связи, одна из которых – «подлежащее – сказуемое», и это слово также добавляется в ответ. Последнее слово «игр»

связано с предыдущим словом только связью «синоним» и поэтому отбрасывается.

В результате получим два слова ответа: «проводится матч».

3 *В зависимости от типа вопроса выбрать дополнительные слова, связанные нужным типом связи с наибольшим количеством слов найденного фрагмента.*

Вопрос «Где проводятся игры?» относится к определению места. Место, связанное со словами найденного фрагмента сети «на стадионе», добавляется в ответ. В результате получен ответ: «Проводится матч на стадионе».

Если для найденного фрагмента сети невозможно определить место, то это означает, что ответ на вопрос не удается найти.

Аналогичным образом можно показать, что описанный алгоритм позволяет найти ответы на другие вопросы:

– «Где опустели улицы?» – «Опустели улицы в городе»;

– «Как будет потом?» – «Будет многолюдно»;

– «Из-за чего улицы опустели?» – «Улицы опустели из-за матча».

Поиск пути между узлами сети может быть реализован с использованием алгоритма поиска в ширину, поиска в глубину или другими способами.

Добавление предложений приводит к расширению сети, но количество сохраняемых предложений не ограничено. Рассмотренные примеры показали, что получение ответа на вопрос при помощи семантической сети является хорошо формализуемой и автоматизируемой процедурой, а ее результат во многом зависит от способа построения сети и точности выделения смысловых связей между словами.

И нейронная, и семантическая сети могут быть использованы для получения ответов на вопросы, сформулированные на естественном языке. Преимуществом нейронной сети можно считать ее универсальность. Она позволяет решать задачи коррекции образов, абстрагирования, классификации и многие другие.

Преимуществом модели семантической сети является ее ориентированность на обработку естественного языка. Она не требует преобразования текстовой информации в другой вид информации, кроме того, модель семантической сети удобна для восприятия человеком. Еще одним преимуществом модели семантической сети является возможность хранить большое количество предложений текста.

# ЗАКЛЮЧЕНИЕ

Учебно-методическое пособие посвящено моделям и методам работы с информацией, обрабатываемой в информационной системе. В работе рассмотрены вопросы, относящиеся к синтаксическому и семантическому аспектам информации.

Поскольку объектом исследования в учебно-методическом пособии является текстовая информация, в нем рассмотрены различные подходы работы с текстами. К первой группе методов относятся простые, быстрые, но не очень точные способы. Чаще всего в этих подходах используются формальные статистические приемы, поэтому они представлены алгоритмами, использующими законы Дж. Зипфа, в которых учитывается частота появления в тексте слов различных тематик (лабораторные работы 1–4). Вторую группу формируют достаточно сложные, показывающие хороший результат, но сравнительно медленные подходы, основанные на лингвистических методах. Они часто применяются в интеллектуальных информационных системах для решения слабоформализованных задач. Все они имеют широкое применение для различных прикладных областей. Ко второй группе можно отнести генетический алгоритм, искусственные нейронные и семантические сети (лабораторные работы 5–6).

Решение задач с помощью ИНС состоит из двух процедур: обучения и непосредственно самого распознавания незнакомых образов. Поиск решения с помощью сети, прошедшей обучение, оказывается более качественным, чем использование других вычислительных средств, поскольку ИНС может повышать точность результатов по мере накопления ею опыта и адаптироваться к изменениям. В основу работы генетического алгоритма также положены принципы существования живой природы, и кроме того, он часто используется в паре с ИНС. Семантические сети позволяют выделять смысл текста в виде понятий и связей между ними, образующих граф. Отношения в сетях могут быть самых разных типов, что делает семантическую сеть универсальной моделью представления данных и знаний. К ее достоинствам относятся наглядность, близость к естественному языку и схожесть по организации с долговременной памятью человека.

Эффективным же можно считать такой подход, который сочетал бы в себе простоту статистических алгоритмов с достаточно высоким качеством обработки лингвистических методов.

В результате изучения дисциплины «Модели и методы обработки данных в информационных системах» с помощью данного учебно-методического пособия студенты смогут лучше разобраться в назначении и особенностях функционирования информационных систем, их видах и принципах работы с информацией, а также получить навыки в применении различных алгоритмов и моделей обработки данных.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ


1 Мельников, В. П. Информационное обеспечение систем управления : учебник для студ. высш. учеб. заведений / В. П. Мельников. – М. : Издательский центр «Академия», 2010.

2 Степанова, Е. Е. Информационное обеспечение управленческой деятельности / Е. Е. Степанова, Н. В. Хмелевская. – М. : Форум, 2010.

3 Долгополова, Е. Е. Информационное обеспечение маркетинга: теория и практика / Е. Е. Долгополова. – Минск : Новое знание, 2009.

4 Серебряная, Л. В. Информационное обеспечение в финансовых структурах : метод. пособие / Л. В. Серебряная. – Минск : БГУИР, 2011.

5 Серебряная, Л. В. Модели и методы классификации текстовой информации на основе искусственной нейронной и семантической сетей / Л. В. Серебряная, В. В. Потараев // Информатика. – 2016. – №4(52).

6 Евгенев, Г. Б. Системология инженерных знаний : учеб. пособие / Г. Б. Евгенев. – М. : Финансы и статистика, 2001.

7 Вендров, А. М. Проектирование программного обеспечения экономических информационных систем : учебник / А. М. Вендров. – М. : Форум, 2000.

8 Крылович, А. В. Эволюция информационных технологий управления / А. В. Крылович // БОСС. – 2000. – №5.

9 Гладков, Л. А. Генетические алгоритмы : учебник / Л. А. Гладков, В. В. Курейчик, В. М. Курейчик. – 2-е изд. – М. : Физматлит, 2010.

10 Рутковская, Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский ; пер. с польского И. Д. Рудинского. – М. : Горячая линия – Телеком, 2007.

11 Искусственная нейронная сеть [Электронный ресурс]. – 2018. – Режим доступа : https://ru.wikipedia.org/wiki.

12 Алгоритм обратного распространения ошибки [Электронный ресурс]. – 2018. – Режим доступа : http://www.aiportal.ru/articles/neural-networks/back-propagation.html.

13 Нейронная сеть Хопфилда и ее применение [Электронный ресурс]. – 2018. – Режим доступа : http://iasa.org.ua /lections/tpr/neuro/hopfield.htm.

14 Семантические сети или сетевые модели знаний [Электронный ресурс]. – 2018. – Режим доступа : http://www.aiportal.ru/articles/knowledge-models/semantic-network.html.

Св. план 2019, поз. 30

*Учебное издание*

**Серебряная** Лия Валентиновна
**Потараев** Виктор Витальевич
**Фадеева** Елена Павловна

# *МОДЕЛИ И МЕТОДЫ ОБРАБОТКИ ДАННЫХ В ИНФОРМАЦИОННЫХ СИСТЕМАХ*

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

Редактор *Е. С. Юрец*
Корректор *Е. Н. Батурчик*
Компьютерная правка, оригинал-макет *М. В. Касабуцкий*

Подписано в печать 22.10.2019. Формат 60×84 1/16. Бумага офсетная. Гарнитура «Таймс».
Отпечатано на ризографе. Усл. печ. л. 4,07. Уч.-изд. л. 4,0. Тираж 50 экз. Заказ 53.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники».
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий №1/238 от 24.03.2014,
№2/113 от 07.04.2014, №3/615 от 07.04.2014.
Ул. П. Бровки, 6, 220013, г. Минск