

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Левков Сергей Александрович
Должность: Ректор
Дата подписания: 30.05.2024 16:15:11
Уникальный программный ключ:
0ec96352bebea6f8385fb9c27c7d4c35a083708b

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«КАМЧАТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «КамчатГТУ»)**

Кафедра «Системы управления»

О.В. Мандрикова

**СОВРЕМЕННЫЕ ПРОБЛЕМЫ НАУКИ В ОБЛАСТИ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ**

*программа курса и методические указания к изучению
дисциплины для обучающихся*
по направлению подготовки 19.06.01 Информатика и вычислительная техника
(уровень подготовки кадров высшей квалификации)

направленность (профиль)
«Математическое моделирование, численные методы и комплексы программ»

Петропавловск-Камчатский,
2019

Мандрикова Оксана Викторовна

Современные проблемы науки в области информационных технологий. Программа курса и методические указания к изучению дисциплины для обучающихся направления подготовки 19.06.01 «Информатика и вычислительная техника», направленность (профиль) подготовки «Математическое моделирование, численные методы и комплексы программ». – Петропавловск-Камчатский: КамчатГТУ, 2019. – 72 с.

Программа курса и методические указания к изучению дисциплины для обучающихся направления подготовки 19.06.01 «Информатика и вычислительная техника» составлены в соответствии с требованиями федеральных государственных образовательных стандартов высшего образования (уровень подготовки кадров высшей квалификации).

Программа курса и методические указания к изучению дисциплины «Математическое моделирование, численные методы и комплексы программ» рассмотрены и утверждены на заседании НТС (протокол № 1 от 11.09.2019 г.).

СОДЕРЖАНИЕ

1. ЦЕЛЬ И ЗАДАЧИ ДИСЦИПЛИНЫ, ЕЕ МЕСТО В УЧЕБНОМ ПРОЦЕССЕ.....	4
1.1. Краткая характеристика дисциплины.....	4
1.2. Цель и задачи дисциплины.....	4
2. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ	6
2.1. Содержание лекционных занятий.....	6
2.2. Содержание практических занятий	66
3. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ	69
3.1. Методические рекомендации по изучению курса.....	69
3.2. Методические рекомендации по подготовке к практическим занятиям	70
3.3. Методические рекомендации по подготовке к кандидатскому экзамену.....	71
4. УЧЕБНО – МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ	71
4.1. Основная литература.....	71
4.2. Дополнительная литература	71
4.3. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»	72

1. ЦЕЛЬ И ЗАДАЧИ ДИСЦИПЛИНЫ, ЕЕ МЕСТО В УЧЕБНОМ ПРОЦЕССЕ

1.1. Краткая характеристика дисциплины

Курс посвящен ознакомлению с современными проблемами и методами в области информационных технологий и систем и их применении в области математического моделирования, численных методов и комплексов программ. Рассмотрено текущее состояние современных научных достижений в области информационных технологий и систем. Изложены необходимые сведения для формирования общих принципов разработки и анализа информационных систем. Представлено применение современных методов в области разработки информационных систем.

1.2. Цель и задачи дисциплины

Целью освоения дисциплины «Современные проблемы науки в области информационных технологий» является формирование основ математического представления и преобразований информационных данных в современных информационных системах регистрации, накопления, обработки и представления данных.

Основные *задачи* курса:

- развить знания в области современного состояния исследований и разработок информационных систем и технологий;
- научить способности проведения анализа существующих мировых проблем, способов их решения и перспективных направлений развития в области информационных технологий;
- научить применять современные математические знания к исследованию реальных процессов и решению профессиональных задач;
- развить способность выделения основных тенденций в области эффективного использования информационных ресурсов в науке, образовании, промышленности др.
- выработать умение самостоятельно расширять математические знания и проводить математический анализ инженерных задач.

Методика преподавания данной дисциплины предполагает чтение лекций , проведение практических занятий , текущих консультаций , самостоятельную работу по изучаемым темам, а также по отдельным специфическим проблемам дисциплины.

Лекции основываются на изучении наиболее важных концептуальных вопросов , связанных с темой раздела и темой лекции . В ее начале очень кратко объясняются концептуальные положения и ключевые понятия. Затем подробно раскрываются отдельные вопросы лекции, история их изучения, основная суть. В конце дается краткое обобщение представленного на лекции материала.

Целью проведения *практических занятий* является закрепление знаний обучающихся, полученных ими в ходе изучения дисциплины на лекциях и самостоятельно, а также формирования определенных профессиональных навыков и умений в области общей и прикладной гидробиологии . Практические занятия проводятся в форме опроса по заданной тематике . Учащимся предлагается возможность обсудить за - данную тему с точки зрения использования полученных знаний на пра ктическом опыте при проведении диссертационного исследования. Привести примеры из опыта собствен - ного исследования.

В процессе изучения дисциплины предусмотрена *самостоятельная внеаудиторная работа* обучающегося в форме осуществления информационного поиска материалов для выполнения самостоятельной работы и его анализа . Контроль за выполнением само -

стоятельных заданий осуществляется в ходе опроса, тестирования, текущих консультаций.

В результате изучения дисциплины *обучающийся должен:*

знать:

- современные достижения в сфере научно-технической информации по тематике исследования;;
- методику проведения исследований с применением современных информационных технологий в области математического моделирования и численных методов при решении научных и технических проблем;
- методики проведения исследований с применением математических пакетов и программ в области профессиональной деятельности;

уметь:

- анализировать и оценивать современные научные достижения в сфере математического моделирования и численных методов;
- проводить численные исследования с применением современных информационных технологий при решении научных и технических проблем;;
- методики проведения исследований с применением математических пакетов и программ в области профессиональной деятельности;

владеть:

- навыком получения современной научно-технической информации из электронных баз данных и научной литературы;
- навыком использования полученных знаний и умений при решении научных и технических задач в области исследования;
- методологией и навыками комплексных исследований в области профессиональной деятельности;

2. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

2.1. Содержание лекционных занятий

Раздел 1. Интеллектуальные информационные системы и технологии

Тема № 1. Общий анализ современных проблем информационных систем и технологий в области искусственного интеллекта

Материалы лекций

Искусственный интеллект - это возможность решения задач, которые до сих пор не удавалось решить человеку, машинным способом с помощью программных средств. Основной проблемой искусственного интеллекта является разработка методов представления и обработки знаний.

К программам искусственного интеллекта относятся:

1. игровые программы (стохастические, компьютерные игры);
2. естественно-языковые программы - машинный перевод, генерация текстов, обработка речи;
3. распознающие программы - распознавание почерков, изображений, карт;
4. программы создания и анализа графики, живописи, музыкальных произведений.

Выделяются следующие направления искусственного интеллекта:

1. экспертные системы;
2. нейронные сети;
3. естественно-языковые системы;
4. эволюционные методы и генетические алгоритмы;
5. нечеткие множества;
6. системы извлечения знаний.

История развития систем искусственного интеллекта начинается с середины 50-х годов и связана с уровнем развития вычислительной техники. Условно можно выделить 7 этапов развития искусственного интеллекта, каждый из которых связывается с определённым уровнем развития искусственного интеллекта и парадигмой, реализуемой в конкретной системе.

Этап 1 (50-е годы) (Нейрон и нейронные сети)

Он связан с появлением первых машин последовательного действия, с очень небольшими, по сегодняшним меркам, ресурсными возможностями по памяти, быстродействию и классам решаемых задач. Это были задачи сугубо вычислительного характера, для которых были известны схемы решений и которые можно описать на некотором формальном языке. К этому же классу относятся и задачи на адаптацию.

Этап 2 (60-е годы) (Эвристический поиск)

В «интеллект» машины добавились механизмы поиска, сортировки, простейшие операции по обобщению информации, не зависящие от смысла обрабатываемых данных. Это стало новой точкой отсчета в развитии и понимании задач автоматизации деятельности человека.

Этап 3 (70-е годы) (Представление знаний)

Учеными была осознана важность знаний (по объему и содержанию) для синтеза интересных алгоритмов решения задач. При этом имелись в виду знания, с которыми математика не умела работать, т.е. опытные знания, не носящие строгого формального характера и описываемые обычно в декларативной форме. Это знания специалистов в различных областях деятельности, врачей, химиков, исследователей и т.п. Такие знания получили название экспертных знаний, и соответственно системы, работающие на основе экспертных знаний, стали называться системами-консультантами или экспертными системами

Этап 4 (80-е годы) (Обучающие машины)

Четвертый этап развития ИИ стал прорывным. С появлением экспертных систем в мире начался принципиально новый этап развития интеллектуальных технологий – эра интеллектуальных систем - консультантов, которые предлагали варианты решений, обосновывали их, способны были к обучению и к развитию, общались с человеком на привычном для него, хотя и ограниченном, естественном языке.

Этап 5 (90-е годы) (Автоматизированные обрабатывающие центры)

Усложнение систем связи и решаемых задач потребовало качественно нового уровня «интеллектуальности» обеспечивающих программных систем, таких систем, как защита от несанкционированного доступа, информационная безопасность ресурсов, защита от нападений, смысловой анализ и поиск информации в сетях и т.п. И новой парадигмой создания перспективных систем защиты всех видов стали интеллектуальные системы. Именно они позволяют создавать гибкие среды, в рамках которых обеспечивается решение всех необходимых задач.

Этап 6 (2000-е годы) (Робототехника)

Область применения роботов достаточно широка и простирается от автономных газонокосилок и пылесосов до современных образцов военной и космической техники. Модели оборудованы навигационной системой и всевозможными периферийными датчиками.

Этап 7 (год 2008) (Сингулярность)

Создание искусственного интеллекта и самовоспроизводящихся машин, интеграция человека с вычислительными машинами, либо значительное скачкообразное увеличение возможностей человеческого мозга за счёт биотехнологий. По некоторым прогнозам, технологическая сингулярность может наступить уже около 2030 года. Сторонники теории технологической сингулярности считают, что если возникнет принципиально отличный от человеческого разум (постчеловек), дальнейшую судьбу цивилизации невозможно предсказать, опираясь на человеческое (социальное) поведение.

Любая предметная область характеризуется своим набором понятий и связей между ними, своими законами, связывающими между собой объекты данной предметной области, своими процессами, событиями. И конечно, каждая предметная область имеет свои, специфические методы решения задач. Знания о предметной области и способах решения в ней задач весьма разнообразны. Возможны различные классификации этих знаний.

В общем случае знания подразделяются на:

- Процедурные знания описывают последовательности действий, которые могут использоваться при решении задач. Это, например, программы для ЭВМ, словесные записи алгоритмов, инструкция по сборке некоторого изделия.

- Декларативные знания - это все знания, не являющиеся процедурными, например статьи в толковых словарях и энциклопедиях, формулировки законов в физике, химии и других науках и т.п. В отличие от процедурных знаний, отвечающих на вопрос: «Как сделать X?», декларативные знания отвечают, скорее, на вопросы: «Что есть X?» или «Какие связи имеются между X и Y?», «Почему X?» и т.д.

Языки представления знаний можно разделить на типы по формальным моделям представления знаний, которые лежат в их основе:

- логическая,
- сетевая,
- фреймовая,
- продукционная.

В начале 80-х годов в исследованиях по искусственному интеллекту сформировалось самостоятельное направление, получившее название "экспертные системы" (ЭС). Основным назначением ЭС является разработка программных средств, которые при решении задач, трудных для человека, получают результаты, не уступающие по качеству и эффективности решения, решениям, получаемым человеком-экспертом. ЭС используются для решения так называемых неформализованных задач, общим для которых является то, что:

- задачи не могут быть заданы в числовой форме;
- цели нельзя выразить в терминах точно определенной целевой функции;
- не существует алгоритмического решения задачи;
- если алгоритмическое решение есть, то его нельзя использовать из-за
- ограниченности ресурсов (время, память).

Кроме того неформализованные задачи обладают ошибочностью, неполнотой, неоднозначностью и противоречивостью как исходных данных, так и знаний о решаемой задаче.

Экспертная система - это программное средство, использующее экспертные знания для обеспечения высокоэффективного решения неформализованных задач в узкой предметной области. Основу ЭС составляет база знаний (БЗ) о предметной области, которая накапливается в процессе построения и эксплуатации ЭС. Накопление и организация знаний - важнейшее свойство всех ЭС.

Знания являются явными и доступными, что отличает ЭС от традиционных программ, и определяет их основные свойства, такие, как:

1) Применение для решения проблем высококачественного опыта, который представляет уровень мышления наиболее квалифицированных экспертов в данной области, что ведет к решениям творческим, точным и эффективным.

2) Наличие прогностических возможностей, при которых ЭС выдает ответы не только для конкретной ситуации, но и показывает, как изменяются эти ответы в новых ситуациях, с возможностью подробного объяснения каким образом новая ситуация привела к изменениям.

3) Обеспечение такого нового качества, как институциональная память, за счет входящей в состав ЭС базы знаний, которая разработана в ходе взаимодействий со специалистами организации, и представляет собой текущую политику этой группы людей. Этот набор знаний становится сводом квалифицированных мнений и постоянно

обновляемым справочником наилучших стратегий и методов, используемых персоналом. Ведущие специалисты уходят, но их опыт остается.

4) Возможность использования ЭС для обучения и тренировки руководящих работников, обеспечивая новых служащих обширным багажом опыта и стратегий, по которым можно изучать рекомендуемую политику и методы.

Экспертная система - это программное средство, использующее знания экспертов, для высокоэффективного решения задач в интересующей пользователя предметной области. Она называется системой, а не просто программой, так как содержит базу знаний, решатель проблемы и компоненту поддержки. Последняя из них помогает пользователю взаимодействовать с основной программой.

Эксперт - это человек, способный ясно выражать свои мысли и пользующийся репутацией специалиста, умеющего находить правильные решения проблем в конкретной предметной области. Эксперт использует свои приемы и ухищрения, чтобы сделать поиск решения более эффективным, и ЭС моделирует все его стратегии.

Инженер знаний - человек, как правило, имеющий познания в информатике и искусственном интеллекте и знающий, как надо строить ЭС. Инженер знаний опрашивает экспертов, организует знания, решает, каким образом они должны быть представлены в ЭС, и может помочь программисту в написании программ.

Средство построения ЭС - это программное средство, используемое инженером знаний или программистом для построения ЭС. Этот инструмент отличается от обычных языков программирования тем, что обеспечивает удобные способы представления сложных высокоуровневых понятий.

Пользователь - это человек, который использует уже построенную ЭС. Так, пользователем может быть юрист, использующий ее для квалификации конкретного случая; студент, которому ЭС помогает изучать информатику и т. д. Термин пользователь несколько неоднозначен. Обычно он обозначает конечного пользователя. При этом пользователем может быть:

- создатель инструмента, отлаживающий средство построения ЭС;
- инженер знаний, уточняющий существующие в ЭС знания;
- эксперт, добавляющий в систему новые знания;
- клерк, заносящий в систему текущую информацию.

Сферы применения ИИ достаточно широки и охватывают как привычные слуху технологии, так и появляющиеся новые направления, далекие от массового применения, иначе говоря, это весь спектр решений, от пылесосов до космических станций. Можно разделить все их разнообразие по критерию ключевых точек развития.

ИИ — это не монолитная предметная область. Более того, некоторые технологические направления ИИ фигурируют как новые подотрасли экономики и обособленные сущности, одновременно обслуживая большинство сфер в экономике. Развитие применения использования ИИ ведет к адаптации технологий в классических отраслях экономики по всей цепочке создания ценности и преобразует их, приводя к алгоритмизированию практически всего функционала, от логистики до управления компанией.

Автоматизированная информационная система (АИС) представляет собой совокупность информации, экономико–математических методов и моделей, технических,

программных, технологических средств и специалистов, предназначенную для обработки информации и принятия управленческих решений.

Создание АИС способствует повышению эффективности производства экономического объекта и обеспечивает качество управления. Наибольшая эффективность АИС достигается при оптимизации планов работ предприятия, фирм и отраслей, быстрой выработке оперативных решений, четком маневрировании материальными и финансовыми ресурсами.

Выделяют следующие недостатки информационных систем:

Чувствительность системы к неправильным действиям. Все бизнес-процессы должны быть идеально отлажены. Любая неправильная информация или ее отсутствие ведет к кардинальным ошибкам в работе системы и, как следствие, высокому риску принятия неверного решения.

Дороговизна решения. Оценивая стоимость проекта, не надо забывать, что помимо стоимости лицензий и услуг консультантов по внедрению, существуют значительные затраты на перестройку всех бизнес-процессов и затраты, связанные с огромными усилиями всех вовлеченных в процесс менеджеров и специалистов компании. Последняя составляющая с трудом поддается хотя бы приблизительной оценке.

Проблемы понимания. Отсутствие понимания, что такое ИС, какими они должны быть в применении к деятельности и стратегическим задачам компании.

Проблемы организации. Отсутствие должной формализованной системы правил той деятельности предприятия, которая влияет на развитие компании. Зачастую, работа многих участков предприятия при всей ее успешности представляет собой творческий процесс.

Проблемы управления. Отсутствие формализованной системы управления компанией. В частности:

- отсутствие правил принятия решений,
- отсутствие способов контроля качества работы как предприятия в целом, так и его подразделений,
- отсутствие понятных инструментов для принятия управленческих решений.

Проблемы автоматизации. Отсутствие программных инструментов и соответствующих специалистов, способных реализовать видение конкретного управленца и специфику компании. Проблемы поддержки автоматизированной информационной системы управления состоят в необходимости регулярной модернизации системы с целью более полного удовлетворения нужд в информации принятой системы управления предприятием.

Проблемы поддержки программного обеспечения, автоматизирующего информационную систему, связаны с:

- техническими сбоями и поломками ПО,
- потребностью предприятия в обновлении версий ПО,
- необходимостью постоянного обучения новых сотрудников – пользователей ПО,
- необходимостью регулярной модернизации ПО с целью более полного удовлетворения нужд принятой системы управления.

Проблемы развития автоматизированной системы управления. Необходимость преодолевать рассогласование автоматизированного решения с изменяющейся системой управления предприятием, постоянно возникающее как результат его развития. Это

означает, что управленцы на разных уровнях начинают принимать управленческие решения по-новому, и действующая система учета не обеспечивает их необходимой информацией в должном объеме.

Вопросы для самоконтроля:

1. Искусственный интеллект.
2. Типы знаний.
3. Проблемы извлечения знаний.
4. Ограничения информационных систем.

Литература: [1]; [2]; [5]; [6]; [10]; [11]; [12]; [13].

Тема № 2. Искусственный интеллект: процедуры, задачи и средства автоматизации

Методы и инструменты управления формализованными знаниями компании сегодня практически полностью базируются на информационных технологиях (ИТ). Формализованные знания могут быть зафиксированы и на бумажных носителях, но для повышения степени удобства и гарантированности сохранения имеет смысл максимально перевести формализованные знания в электронный вид и применять ИТ для совершения операций над ними. Современные ИТ, построенные на основе систем обработки данных и систем искусственного интеллекта, играют важную роль в развитии компании. Среди таких ИТ выделяются следующие инструменты, применяемые для задач управления знаниями:

- специализированные базы данных (так называемые "базы знаний"), хранящие опыт и знания сотрудников компании, – в этих базах может храниться описание различных профессиональных ситуаций и правил поведения в них, возникавшие ранее в компании проблемы и успешные пути их решения. В зависимости от вида хранимых ситуаций такие базы называют базами "лучших практик" (best practice), "выученных уроков" (lessons learned), "совершенных ошибок" и т.д.;

- системы управления документооборотом, которые обеспечивают публикацию, хранение, индексирование документов, а также быстрый поиск и доступ к электронным документам, содержащим знания;

- системы поиска информации, т.е. средства информационного поиска по запросу сотрудника, – такие системы позволяют сотруднику создавать запрос на естественном для него языке, при этом поиск ведется по всем имеющимся в компании базам данных и знаний;

- средства оперативного представления данных (OLAP – online analytical processing) и системы добычи данных и текстов (data mining и text mining), которые позволяют находить новые значимые закономерности (т.е. новые знания) из обширных массивов данных, находящихся в специализированных информационных хранилищах компании. Эти системы основываются на методах статистического моделирования, математической статистики, нейронных сетей, генетических алгоритмов и др.;

– средства, поддерживающие принятие решений (decision support system) и построенные на методах искусственного интеллекта: экспертные системы, системы описания типовых ситуаций т.д.

Модели представления знаний обычно делят на логические (формальные) и эвристические (формализованные). В основе логических моделей представления знаний лежит понятие формальной системы (теории). Примерами формальных теорий могут служить исчисление предикатов и любая конкретная система продукций. В логических моделях, как правило, используется исчисление предикатов первого порядка, дополненное рядом эвристических стратегий. Эти методы являются системами дедуктивного типа, т.е. в них используется модель получения вывода из заданной системы посылок с помощью фиксированной системы правил вывода. Дальнейшим развитием предикатных систем являются системы индуктивного типа, в которых правила вывода порождаются системой на основе обработки конечного числа обучающих примеров.

В логических моделях представления знаний отношения, существующие между отдельными единицами знаний, выражаются только с помощью тех небогатых средств, которые предоставляются синтаксическими правилами используемой формальной системы.

В отличие от формальных моделей эвристические модели имеют разнообразный набор средств, передающих специфические особенности той или иной проблемной области. Именно поэтому эвристические модели превосходят логические как по возможности адекватно представить проблемную среду, так и по эффективности используемых правил вывода. К эвристическим моделям, используемым в экспертных системах, можно отнести сетевые, фреймовые, продукционные и объектно-ориентированные модели. Следует отметить, что продукционные модели, используемые для представления знаний в экспертных системах, отличаются от формальных продукционных систем тем, что они используют более сложные конструкции правил, а также содержат эвристическую информацию о специфике проблемной среды, выражаемую часто в виде семантических структур.

К основным моделям представления знаний для экспертных систем относятся:

- логические;
- функциональные;
- продукционные;
- семантические сети;
- фреймы.

Продукционная модель, или модель, основанная на правилах, позволяет представить знания в виде предложений типа: ЕСЛИ А ТО В. Импликация может истолковываться в обычном логическом смысле как знак логического следования В из истинного А. Возможны и другие интерпретации продукции, например А описывает некоторое условие, необходимое, чтобы можно было совершить действие В. Если в памяти системы хранится некоторый набор продукций, то они образуют систему продукций. В системе продукций должны быть заданы специальные процедуры управления продукциями, с помощью которых происходит актуализация продукций и выполнение той или иной продукции из числа актуализированных.

В состав системы продукций входит база правил (продукций), глобальная база данных и система управления. База правил – это область памяти, которая содержит совокупность

знаний в форме правил вида ЕСЛИ – ТО. Глобальная база данных – область памяти, содержащая фактические данные (факты). Система управления формирует заключения, используя базу правил и базу данных. Существуют два способа формирования заключений – прямые выводы и обратные выводы.

В прямых выводах выбирается один из элементов данных, содержащихся в базе данных, и если при сопоставлении этот элемент согласуется с левой частью правила (посылкой), то из правила выводится соответствующее заключение и помещается в базу данных или выполняется действие, определяемое правилом, и соответствующим образом изменяется содержимое базы данных.

В обратных выводах процесс начинается от поставленной цели. Если эта цель согласуется с правой частью правила (заключением), то посылка правила принимается за подцель или гипотезу. Этот процесс повторяется до тех пор, пока не будет получено совпадение подцели с данными. При большом числе продукций в продукционной модели усложняется проверка непротиворечивости системы продукций, т.е. множества правил. Поэтому число продукций, с которыми работают современные системы ИИ, как правило, не превышают тысячи.

Семантическая сеть- это ориентированный граф, вершины которого - понятия, а дуги - отношения между ними. Термин семантическая означает смысловая, а сама семантика - это наука, устанавливающая отношения между символами и объектами, которые они обозначают, т.е. наука, определяющая смысл знаков. Понятиями обычно выступают абстрактные или конкретные объекты, а отношения - это связи типа: "это" ("is"), "имеет часть" ("has part"), "принадлежит", "любит". Характерной особенностью семантических сетей является обязательное наличие трех типов отношений:

- класс - элемент класса;
- свойство - значение;
- пример элемента класса.

Семантические сети, используемые для описания естественных языков, используют дуги типа "агент", "объект", "источник", "приемник", "время", "место", "цель". Агент - предмет, являющийся инициатором действия. Объект - предмет, подвергающийся действию. Источник - размещение предмета перед действием. Приемник - размещение предмета после действия. Время - момент выполнения действия. Место - место проведения действия. Цель - действие другого события.

Проблема поиска решения в базе знаний типа семантической сети сводится к задаче поиска фрагмента сети, соответствующего некоторой подсети, соответствующей поставленному вопросу.

Основная идея фреймового подхода к представлению знаний заключается в том, что все, что касается понятия или ситуации, не «размывается по сети», а представляется во фрейме. Фреймом называется структура для описания понятия или ситуации, состоящая из характеристик этой ситуации и их значений. Фрейм можно рассматривать как фрагмент семантической сети, предназначенный для описания понятий со всей совокупностью присущих им свойств.

Различают фреймы-образцы, или прототипы, хранящиеся в базе знаний, и фреймы-экземпляры, которые создаются для отображения реальных ситуаций на основе поступающих данных. Модель фрейма является достаточно универсальной, поскольку позволяет отобразить все многообразие знаний о мире через:

- фреймы-структуры, для обозначения объектов и понятий (заем, залог, вексель);
- фреймы-роли (менеджер, кассир, клиент);
- фреймы-сценарии (банкротство, собрание акционеров, празднование именин);
- фреймы-ситуации (тревога, авария, рабочий режим устройства) и др.

Нейронные сети (НС) - искусственные, многослойные высоко параллельные (т. е. с большим числом независимо параллельно работающих элементов) логические структуры, составленные из формальных нейронов. Начало теории нейронных сетей и нейрокомпьютеров положила работа американских нейрофизиологов У. Мак-Каллока и У. Питтса «Логическое исчисление идей, относящихся к нервной деятельности» (1943), в которой они предложили математическую модель биологического нейрона. Среди основополагающих работ следует выделить модель Д. Хэбба, который в 1949 г. предложил закон обучения, явившийся стартовой точкой для алгоритмов обучения искусственных нейронных сетей. На дальнейшее развитие теории нейронной сети существенное влияние оказала монография американского нейрофизиолога Ф. Розенблатта «Принципы нейродинамики», в которой он подробно описал схему перцептрона (устройства, моделирующего процесс восприятия информации человеческим мозгом). Его идеи получили развитие в научных работах многих авторов. В 1985–86 гг. теория нейронных сетей получила «технологический импульс», вызванный возможностью моделирования нейронных сетей на появившихся в то время доступных и высокопроизводительных персональных компьютерах. Теория нейронной сети продолжает достаточно активно развиваться в начале 21 века.

Потенциальными областями применения искусственных нейронных сетей являются те, где человеческий интеллект малоэффективен, а традиционные вычисления трудоёмки или физически неадекватны (т. е. не отражают или плохо отражают реальные физические процессы и объекты). Актуальность применения нейронных сетей (т. е. нейрокомпьютеров) многократно возрастает, когда появляется необходимость решения плохо формализованных задач. Основные области применения нейронных сетей: автоматизация процесса классификации, автоматизация прогнозирования, автоматизация процесса распознавания, автоматизация процесса принятия решений; управление, кодирование и декодирование информации; аппроксимация зависимостей и др.

С помощью нейронных сетей успешно решается важная задача в области телекоммуникаций – проектирование и оптимизация сетей связи (нахождение оптимального пути трафика между узлами). Кроме управления маршрутизацией потоков, нейронные сети используются для получения эффективных решений в области проектирования новых телекоммуникационных сетей. Распознавание речи – одна из наиболее популярных областей применения нейронных сетей.

Вопросы для самоконтроля:

1. Способы представления и управления знаниями.
2. Методы представления знаний.
3. Нейронные сети.

Литература: [1]; [2]; [5]; [6]; [8]; [9]; [10]; [11]; [12]; [13].

Материалы лекций

DataMining (discovery-driven datamining) дословно переводится как "добыча" или "раскопка данных". В основу современной технологии Data Mining положена концепция шаблонов (паттернов), отражающих фрагменты многоаспектных взаимоотношений в данных. Эти шаблоны представляют собой закономерности, свойственные выборкам данных, которые могут быть компактно выражены в понятной человеку форме. Поиск закономерностей производится методами, неограниченными рамками априорных предположений о структуре выборки и виде распределений значений анализируемых показателей. Важное положение Data Mining среди других информационных технологий – нетривиальность разыскиваемых закономерностей. Это означает, что найденные закономерности должны отражать неочевидные, неожиданные регулярности в данных составляющие, так называемые скрытые знания.

Рассмотрим основные методы Data Mining и их приложения в конкретных системах.

1. Статистические методы.

К базовым методам Data Mining традиционно причисляют все подходы, использующие элементы теории статистики. Последние версии почти всех известных статистических пакетов включают наряду с традиционными статистическими методами также элементы Data Mining. В качестве примеров наиболее мощных и распространенных статистических пакетов можно назвать SAS (компания SAS Institute), SPSS (компания SPSS), STATGRAPHICS (компания Manugistics), STATISTICA, STADIA и другие.

2. Полный и ограниченный перебор.

К базовым методам Data Mining принято относить также алгоритмы, основанные на переборе. Простой перебор всех исследуемых объектов требует $O(2N)$ операций, где N – количество объектов. Следовательно, с увеличением количества данных объем вычислений растет экспоненциально, что при большом объеме делает решение любой задачи таким методом практически невозможным. Наиболее ярким современным представителем реализации этого подхода является система Wiz Why компании WizSoft.

3. Нечеткая логика.

Основным способом исследования задач анализа данных является их отображение на формализованный язык и последующий анализ полученной модели. Неопределенность по объему отсутствующей информации у системного аналитика можно разделить на три большие группы:

- 1) неизвестность,
- 2) неполнота (недостаточность, неадекватность),
- 3) недостоверность.

Недостоверность бывает физической (источником ее является внешняя среда) и лингвистической (возникает в результате словесного обобщения и обуславливается необходимостью описания бесконечного числа ситуаций ограниченным числом слов за ограниченное время). Основной сферой применения нечеткой логики и во многом остается управление.

4. Генетические алгоритмы.

Генетические алгоритмы относятся к числу универсальных методов оптимизации, позволяющих решать задачи различных типов (комбинаторные, общие задачи с ограничениями и без ограничений) и различной степени сложности. Одним из наиболее востребованных приложений генетического алгоритма в области Data Mining является поиск наиболее оптимальной модели (поиск алгоритма, соответствующего специфике конкретной области). Эти алгоритмы удобны тем, что их легко распараллеливать. Пример системы, построенной на основе технологии Data Mining с применением генетических алгоритмов: система Gene Hunter фирмы Ward Systems Group.

5. Нейронные сети.

Нейронные сети – это класс моделей, основанных на биологической аналогии с мозгом человека и предназначенных для решения разнообразных задач анализа данных после прохождения этапа, так называемого обучения на имеющихся данных. При применении этого метода, прежде всего, встает вопрос выбора конкретной архитектуры сети (числа "слоев" и количества "нейронов" в каждом из них). Размер и структура сети должны соответствовать существу исследуемого явления. Примеры нейросетевых систем: BrainMaker (CSS), NeuroShell (WardSystemsGroup), OWL (HyperLogic).

6. Деревья решений.

Деревья решения являются одним из наиболее популярных подходов к решению задачи классификации. Они создают иерархическую структуру классифицирующих правил типа "если-то" (if-then), имеющую вид дерева. Для принятия решения, к какому классу отнести некоторый объект или ситуацию, требуется ответить на вопросы, стоящие в узлах этого дерева, начиная с его корня. Большинство систем, построенных на основе технологии Data Mining, используют именно этот метод. Самыми известными являются See5/C5.0 (RuleQuest, Австралия), Clementine (IntegralSolutions, Великобритания), SIPINA (University of Lyon, Франция), IDIS (Information Discovery, США), KnowledgeSeeker (ANGOSS, Канада).

Методы Data Mining помогают решить многие задачи, с которыми сталкивается аналитик. Из них основными являются: задача классификации и регрессии, задача поиска ассоциативных правил и задача кластеризации. Далее приведено краткое описание основных задач анализа данных.

Задача классификации и регрессии. Задача классификации сводится к определению класса объекта по его характеристикам. Необходимо заметить, что в этой задаче множество классов, к которым может быть отнесен объект, известно заранее. Задача регрессии подобно задаче классификации позволяет определить по известным характеристикам объекта значение некоторого его параметра. В отличие от задач классификации значением параметра является не конечное множество классов, а множество действительных чисел.

Задача поиска ассоциативных правил. При поиске ассоциативных правил целью является нахождение частных зависимостей (или ассоциаций) между объектами или событиями. Найденные зависимости представляются в виде правил и могут быть использованы как для лучшего понимания природы анализируемых данных, так и для предсказания появления событий.

Задача кластеризации заключается в поиске независимых групп (кластеров) и их характеристик во всем множестве анализируемых данных. Решение этой задачи помогает

лучше понять данные. Кроме того, группировка однородных объектов позволяет сократить их число, а следовательно, и облегчить анализ.

Перечисленные задачи по назначению делятся на описательные и предсказательные. Сфера применения Data Mining ничем не ограничена она везде, где имеются какие-либо данные. Но в первую очередь методы Data Mining сегодня заинтересовали коммерческие предприятия, развертывающие проекты на основе информационных хранилищ данных. Кратко приведем некоторые приложения Data Mining:

1. Электронная коммерция.
2. Розничная торговля.
3. Промышленное производство.
4. Медицина.
5. Банковское дело.
6. Страховой бизнес.
7. Телекоммуникации.
8. Молекулярная генетика и геновая инженерия.
9. Прикладная химия.

Машинное обучение (Machine Learning) — обширный подраздел искусственного интеллекта, изучающий методы построения алгоритмов, способных обучаться. Различают два типа обучения. Обучение по прецедентам, или индуктивное обучение, основано на выявлении общих закономерностей по частным эмпирическим данным. Дедуктивное обучение предполагает формализацию знаний экспертов и их перенос в компьютер в виде базы знаний. Дедуктивное обучение принято относить к области экспертных систем, поэтому термины машинное обучение и обучение по прецедентам можно считать синонимами. Машинное обучение находится на стыке математической статистики, методов оптимизации и классических математических дисциплин, но имеет также и собственную специфику, связанную с проблемами вычислительной эффективности и переобучения. Многие методы индуктивного обучения разрабатывались как альтернатива классическим статистическим подходам. Многие методы тесно связаны с извлечением информации и интеллектуальным анализом данных (Data Mining). Наиболее теоретические разделы машинного обучения объединены в отдельное направление, теорию вычислительного обучения (Computational Learning Theory, COLT). Машинное обучение — не только математическая, но и практическая, инженерная дисциплина. Чистая теория, как правило, не приводит сразу к методам и алгоритмам, применимым на практике. Чтобы заставить их хорошо работать, приходится изобретать дополнительные эвристики, компенсирующие несоответствие сделанных в теории предположений условиям реальных задач. Практически ни одно исследование в машинном обучении не обходится без эксперимента на модельных или реальных данных, подтверждающего практическую работоспособность метода.

Искусственная нейронная сеть (ИНС) — вычислительная нелинейная модель, в основе которой лежит нейронная структура мозга, способная обучаться выполнению задач классификации, предсказания, принятия решений, визуализации и некоторых других только благодаря рассмотрению примеров.

Любая архитектура ИНС состоит из искусственных нейронов — элементов обработки, имеющих структуру 3 связанных друг с другом слоев: входным, состоящим из одного или более слоев скрытым и выходным.

Входной слой состоит из входных нейронов, которые передают информацию в скрытый слой. Скрытый слой в свою очередь передает информацию в выходной. Каждый нейрон имеет входы с весами — синапсами, функцию активации, определяющую выходную информацию при заданной входной, и один выход. Синапсы — регулируемые параметры, конвертирующие нейронную сеть в параметризованную систему.

Взвешенная сумма со входов — активационный сигнал — проходит через функцию активации для вывода данных из нейрона. Есть несколько видов функции активации: линейная, ступенчатая, сигмоидная, тангенциальная, выпрямительная (Rectified linear unit, ReLu).

Обучение (или тренировка) — процесс оптимизации весов, в котором минимизируется ошибка предсказания, и сеть достигает требуемого уровня точности. Наиболее используемый метод для определения вклада в ошибку каждого нейрона — обратное распространение ошибки, с помощью которого вычисляют градиент. Это одна из модификаций метода градиентного спуска.

С помощью дополнительных скрытых слоев возможно сделать систему более гибкой и мощной. ИНС с многими скрытыми слоями называются глубокими нейронными сетями (deep neural network, DNN); они создают сложные нелинейные связи.

Рассмотрим популярные архитектуры нейронных сетей, которые хорошо показали себя в задачах NLP и рекомендуются к использованию.

1. Многослойный перцептрон

Многослойный перцептрон состоит из 3 или более слоев. Он использует нелинейную функцию активации, часто тангенциальную или логистическую, которая позволяет классифицировать линейно неразделимые данные. Каждый узел в слое соединен с каждым узлом в последующем слое, что делает сеть полностью связанной. Такая архитектура находит применение в задачах распознавания речи и машинном переводе.

2. Сверточная нейронная сеть

Сверточная нейронная сеть (Convolutional neural network, CNN) содержит один или более объединенных или соединенных сверточных слоев. CNN использует вариацию многослойного перцептрона, рассмотренного выше. Сверточные слои используют операцию свертки для входных данных и передают результат в следующий слой. Эта операция позволяет сети быть глубже с меньшим количеством параметров.

Сверточные сети показывают выдающиеся результаты в приложениях к картинкам и речи. В статье Convolutional Neural Networks for Sentence Classification автор описывает процесс и результаты задач классификации текста с помощью CNN. В работе представлена модель на основе word2vec, которая проводит эксперименты, тестируется на нескольких бенчмарках и демонстрирует блестящие результаты.

В работе Text Understanding from Scratch авторы показывают, что сверточная сеть достигает выдающихся результатов даже без знания слов, фраз предложений и любых других синтаксических или семантических структур присущих человеческому языку. Семантический разбор, поиск парафраз, распознавание речи — тоже приложения CNN.

3. Рекурсивная нейронная сеть

Рекурсивная нейронная сеть — тип глубокой нейронной сети, сформированный при применении одних и тех же наборов весов рекурсивно над структурой, чтобы сделать скалярное или структурированное предсказание над входной структурой переменного размера через активацию структуры в топологическом порядке. В простейшей

архитектуре нелинейность, такая как тангенциальная функция активации, и матрица весов, разделяемая всей сетью, используются для объединения узлов в родительские объекты.

4. Рекуррентная нейронная сеть

Рекуррентная нейронная сеть, в отличие от прямой нейронной сети, является вариантом рекурсивной ИНС, в которой связи между нейронами — направленные циклы. Последнее означает, что выходная информация зависит не только от текущего входа, но также от состояний нейрона на предыдущем шаге. Такая память позволяет пользователям решать задачи NLP: распознавание рукописного текста или речи. В статье Natural Language Generation, Paraphrasing and Summarization of User Reviews with Recurrent Neural Networks авторы показывают модель рекуррентной сети, которая генерирует новые предложения и краткое содержание текстового документа.

Siwei Lai, Liheng Xu, Kang Liu, и Jun Zhao в своей работе Recurrent Convolutional Neural Networks for Text Classification создали рекуррентную сверточную нейросеть для классификации текста без рукотворных признаков. Модель сравнивается с существующими методами классификации текста — Bag of Words, Bigrams + LR, SVM, LDA, Tree Kernels, рекурсивными и сверточными сетями. Описанная модель превосходит по качеству традиционные методы для всех используемых датасетов.

5. LSTM

Сеть долгой краткосрочной памяти (Long Short-Term Memory, LSTM) — разновидность архитектуры рекуррентной нейросети, созданная для более точного моделирования временных последовательностей и их долгосрочных зависимостей, чем традиционная рекуррентная сеть. LSTM-сеть не использует функцию активации в рекуррентных компонентах, сохраненные значения не модифицируются, а градиент не стремится исчезнуть во время тренировки. Часто LSTM применяется в блоках по несколько элементов. Эти блоки состоят из 3 или 4 затворов (например, входного, выходного и гейта забывания), которые контролируют построение информационного потока по логистической функции.

В Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling авторы показывают архитектуру глубокой LSTM рекуррентной сети, которая достигает хороших результатов для крупномасштабного акустического моделирования.

В работе Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Recurrent Neural Network представлена модель для автоматической морфологической разметки. Модель показывает точность 97.4 % в задаче разметки. Apple, Amazon, Google, Microsoft и другие компании внедрили в продукты LSTM-сети как фундаментальный элемент.

6. Sequence-to-sequence модель

Часто Sequence-to-sequence модели состоят из двух рекуррентных сетей: кодировщика, который обрабатывает входные данные, и декодера, который осуществляет вывод.

Sequence-to-Sequence модели часто используются в вопросно-ответных системах, чат-ботах и машинном переводе. Такие многослойные ячейки успешно использовались в sequence-to-sequence моделях для перевода в статье Sequence to Sequence Learning with Neural Networks study.

В Paraphrase Detection Using Recursive Autoencoder представлена новая рекурсивная архитектура автокодировщика, в которой представления — вектора в n -мерном семантическом пространстве, где фразы с похожими значением близки друг к другу.

7. Неглубокие (shallow) нейронные сети

Неглубокие модели, как и глубокие нейронные сети, тоже популярные и полезные инструменты. Например, word2vec — группа неглубоких двухслойных моделей, которая используется для создания векторных представлений слов (word embeddings). Представленная в Efficient Estimation of Word Representations in Vector Space, word2vec принимает на входе большой корпус текста и создает векторное пространство. Каждому слову в этом корпусе приписывается соответствующий вектор в этом пространстве. Отличительное свойство — слова из общих текстов в корпусе расположены близко друг к другу в векторном пространстве.

Существуют два концептуальных подхода к обучению нейронных сетей: обучение с учителем и обучение без учителя.

Обучение нейронной сети с учителем предполагает, что для каждого входного вектора из обучающего множества существует требуемое значение выходного вектора, называемого целевым. Эти вектора образуют обучающую пару. Веса сети изменяют до тех пор, пока для каждого входного вектора не будет получен приемлемый уровень отклонения выходного вектора от целевого.

Обучение нейронной сети без учителя является намного более правдоподобной моделью обучения с точки зрения биологических корней искусственных нейронных сетей. Обучающее множество состоит лишь из входных векторов. Алгоритм обучения нейронной сети подстраивает веса сети так, чтобы получались согласованные выходные векторы, т.е. чтобы предъявление достаточно близких входных векторов давало одинаковые выходы.

Распознавание представляет собой информационный процесс, реализуемый некоторым преобразователем информации (интеллектуальным информационным каналом, системой распознавания), имеющим вход и выход. На вход системы подается информация о том, какими признаками обладают предъявляемые объекты. На выходе системы отображается информация о том, к каким классам (обобщенным образам) отнесены распознаваемые объекты. При создании и эксплуатации автоматизированной системы распознавания образов решается ряд задач. Рассмотрим кратко и упрощенно эти задачи.

Задача формализации предметной области.

По сути это задача является задачей кодирования. Составляется список обобщенных классов, к которым могут относиться конкретные реализации объектов, а также список признаков, которыми эти объекты в принципе могут обладать.

Задача формирования обучающей выборки.

Обучающая выборка представляет собой базу данных, содержащую описания конкретных реализаций объектов на языке признаков, дополненную информацией о принадлежности этих объектов к определенным классам распознавания.

Задача обучения системы распознавания.

Обучающая выборка используется для формирования обобщенных образов классов распознавания на основе обобщения информации о том, какими признаками обладают объекты обучающей выборки, относящиеся к этому классу и другим классам.

Задача снижения размерности пространства признаков.

После обучения системы распознавания (получения статистики распределения частот признаков по классам) становится возможным определить для каждого признака его ценность для решения задачи распознавания. После этого наименее ценные признаки могут быть удалены из системы признаков. Затем система распознавания должна быть обучена заново, так как в результате удаления некоторых признаков статистика распределения оставшихся признаков по классам изменяется. Этот процесс может повторяться, т.е. быть итерационным.

Задача распознавания.

Распознаются объекты распознаваемой выборки, которая, в частности, может состоять и из одного объекта. Распознаваемая выборка формируется аналогично обучающей, но не содержит информации о принадлежности объектов к классам, так как именно это и определяется в процессе распознавания. Результатом распознавания каждого объекта является распределение или список всех классов распознавания в порядке убывания степени сходства распознаваемого объекта с ними.

Задача контроля качества распознавания.

После распознавания может быть установлена его адекватность. Для объектов обучающей выборки это может быть сделано сразу, так как для них просто известно, к каким классам они относятся. Для других объектов эта информация может быть получена позже. В любом случае может быть определена фактическая средняя вероятность ошибки по всем классам распознавания, а также вероятность ошибки при отнесении распознаваемого объекта к определенному классу. Результаты распознавания должны интерпретироваться с учетом имеющейся информации о качестве распознавания.

Задача адаптации.

Если в результате выполнения процедуры контроля качества установлено, что оно неудовлетворительное, то описания неправильно распознанных объектов могут быть скопированы из распознаваемой выборки в обучающую, дополнены адекватной классификационной информацией и использованы для переформирования решающих правил, т.е. учтены. Более того, если эти объекты не относятся к уже имеющимся классам распознавания, что и могло быть причиной их неверного распознавания, то этот список может быть расширен. В результате система распознавания адаптируется и начинает адекватно классифицировать эти объекты.

Обратная задача распознавания.

Задача распознавания состоит в том, что для данного объекта по его известным признакам системой устанавливается его принадлежность к некоторому ранее неизвестному классу. В обратной задаче распознавания, наоборот, для данного класса

распознавания системой устанавливается, какие признаки наиболее характерны для объектов данного класса, а какие нет (или какие объекты обучающей выборки относятся к данному классу).

Задачи кластерного и конструктивного анализа.

Кластерами называются такие группы объектов, классов или признаков, что внутри каждого кластера они максимально сходны, а между разными кластерами — максимально различны. Конструктом (в контексте, рассматриваемом в данном разделе) называется система противоположных кластеров. Таким образом, в определенном смысле конструкты есть результат кластерного анализа кластеров. В кластерном анализе количественно измеряется степень сходства и различия объектов (классов, признаков), и эта информация используется для классификации. Результатом кластерного анализа является сама классификация объектов по кластерам. Эта классификация может быть представлена в форме семантических сетей.

Задача когнитивного анализа.

В когнитивном анализе информация о сходстве и различии классов или признаков интересует исследователя сама по себе, а не для того, чтобы использовать ее для классификации, как в кластерном и конструктивном анализе. Если для двух классов распознавания является характерным один и тот же признак, то это вносит вклад в сходство этих двух классов. Если же для одного из классов этот признак является нехарактерным, то это вносит вклад в различие. Если два признака коррелируют друг с другом, то в определенном смысле их можно рассматривать как один признак, а если антикоррелируют, то как различные. С учетом этого обстоятельства наличие различных признаков у разных классов также вносит определенный вклад в их сходство и различие. Результаты когнитивного анализа могут быть представлены в форме когнитивных диаграмм.

Вопросы для самоконтроля:

1. Data Mining.
2. Модели представления знаний в Data Mining.
3. Методы Data Mining и их свойства.
4. Аппарат искусственных нейронных сетей.
5. Автоматизация в задачах распознавания.

Литература: [1]; [2]; [5]; [6]; [10]; [11]; [12]; [13].

Тема № 4. Оценка адекватности и эффективности разработанных интеллектуальных средств и технологий. Теория принятия решений.

Материалы лекций

Необходимость в методах моделирования возникает при решении каких-либо конкретных задач. В тех случаях, когда решение задачи базируется на законах физики,

химии и других фундаментальных областей знаний или когда задача может быть поставлена в терминах конкретного класса прикладных задач, для которого разработан соответствующий математический аппарат, применять термин «проблема принятия решения» нет необходимости. Потребность в этом термине возникает в тех случаях, когда задача настолько усложняется, что для ее постановки и решения не может быть сразу определен подходящий аппарат формализации, когда процесс постановки задачи требует участия специалистов различных областей знаний. Это приводит к тому, что постановка задачи становится проблемой, для решения которой нужно разрабатывать специальные подходы, приемы, методы. В таких случаях возникает необходимость определить область проблемы принятия решения (проблемную ситуацию); выявить факторы, влияющие на ее решение; подобрать приемы и методы, которые позволят сформулировать или поставить задачу таким образом, чтобы решение было принято.

Для принятия решения нужно получить выражение, связывающее цель со средствами ее достижения с помощью вводимых критериев оценки достижимости цели и оценки средств. Такие выражения получили в параллельно возникавших прикладных направлениях различные названия: критерий функционирования; критерий, или показатель, эффективности; целевая, или критериальная, функция; функция цели и т.п.

При постановке задачи могут быть учтены не только обязательные, основные требования, отражаемые с помощью критерия, но и дополнительные требования, которые могут выступать в качестве ограничений. Тогда для решения задачи формируют комплекс соотношений, включающий наряду с основным выражением, связывающим цель со средствами, соотношения-неравенства, к правилу, неравенства (хотя в принципе могут быть и равенствами).

Выражения, связывающие цель со средствами ее достижения, могут представлять собой не только простые соотношения, но и более сложные, составные критерии (показатели) аддитивного или мультипликативного вида. Конечно в этом случае могут возникнуть вычислительные сложности, при преодолении которых может потребоваться вновь обратиться к постановке задачи. Однако полученное формализованное ее представление позволяет в дальнейшем применять и формализованные методы анализа проблемной ситуации.

Функция потерь — функция, которая в теории статистических решений характеризует потери при неправильном принятии решений на основе наблюдаемых данных. Если решается задача оценки параметра сигнала на фоне помех, то функция потерь является мерой расхождения между истинным значением оцениваемого параметра и оценкой параметра.

Статистические оценки могут быть неслучайными (нерандомизированными) и случайными (рандомизированными). Нерандомизированные оценки возможны только в том случае, если между полученными данными (реализацией) и принимаемым решением существует детерминированная зависимость, то есть неслучайная. Тем не менее, наблюдаемые данные обычно случайны. При этом на основе принятой реализации задаётся вероятность того или иного решения. Выбор принятия решения может тоже быть случайным, но часто такой рандомизации удастся избежать.

Из-за случайности наблюдаемых данных принятое решение (оценка) может не совпадать с истинным значением оцениваемого параметра, который в общем случае может быть векторным. Очевидно, что ошибки зависят от выбранного правила принятия

решений. Качество принимаемых оценок характеризуется функцией потерь. На выбор функции потерь влияют особенности решаемой задачи. Общего правила выбора функции потерь не существует. Чаще всего используются следующие функции потерь:

- простая,
- квадратичная,
- прямоугольная,
- экспоненциальная (функция потерь с насыщением).

Многокритериальная задача оптимизации – математическая модель принятия оптимального решения одновременно по нескольким критериям. Эти критерии могут отражать оценки различных качеств объекта (или процесса), по поводу которого принимаются решения.

Многокритериальные задачи, которые не просто решить исходя только из интуитивных соображений, возникают в разнообразных видах человеческой деятельности. Например, при проектировании компьютера может быть поставлена задача, в рамках которой формируется конфигурация, при которой одновременно достигаются максимальное быстродействие, наибольшая оперативная память и наименьший вес компьютера. При создании электрической машины проектировщик пытается подобрать такие ее параметры, при которых достигается максимум коэффициента полезного действия, при этом расходуется минимальное количество дорогих электротехнической стали и меди.

Математическая постановка многокритериальной задачи (далее МЗ) представляется следующим образом: необходимо определить такой вектор переменных $x = (x_1, x_2, \dots, x_n)$ из множества допустимых решений D , при котором значение векторной функции векторного аргумента $F(x_1, x_2, \dots, x_n) = \{f_1(x_1, x_2, \dots, x_n), f_2(x_1, x_2, \dots, x_n), \dots, f_k(x_1, x_2, \dots, x_n)\}$ (где $f_i(x_1, x_2, \dots, x_n)$ – i -ая скалярная целевая функция, $i=1, k$) достигает своего максимума (или минимума).

Сущность многокритериальной задачи состоит в нахождении такого решения, принадлежащего области допустимых решений (т. е. такого $x \in D$), которое в том или ином смысле максимизирует (или минимизирует) значения всех целевых функций $f_i(x)$, где $i = 1, \dots, k$.

Существование решения, буквально максимизирующего (или минимизирующего) одновременно все целевые функции, является редким исключением. Так как многокритериальная задача не имеет в общем случае строго математического решения, то есть, как правило, невозможно найти решение при котором достигается максимум (минимум) сразу по всем критериям, то приходится прибегать к какому-то соглашению о том, какое решение будет наиболее предпочтительным в заданных условиях. Отсюда следует, что, во-первых, в теории многокритериальных задач понятие оптимальности получает различные и притом нетривиальные толкования, а, во-вторых, то, что многокритериальная задача в общем случае решается с привлечением неформальной субъективной информации того, кто принимает окончательное решение или по терминологии теории принятия решений – лица, принимающего решения (ЛПР).

В зависимости от степени неизвестности предстоящего поведения исходных параметров принятия решений различают условия риска, в которых вероятность наступления отдельных событий, влияющих на конечный результат, может быть установлена с той или иной степенью точности, и условия неопределенности, в которых из-за отсутствия необходимой информации такая вероятность не может быть установлена.

Условиями неопределённости считается ситуация, когда результаты принимаемых решений неизвестны. Неопределённость подразделяется на:

- стохастическую (имеется информация о распределении вероятности на множестве результатов),
- поведенческую (имеется информация о влиянии на результаты поведения участников),
- природную (имеется информация только о возможных результатах и отсутствует о связи между решениями и результатами),
- априорную (нет информации и о возможных результатах).

Теория принятия решений в условиях риска и неопределенности основывается на следующих исходных положениях:

- Объект принятия решения четко детерминирован и по нему известны основные из возможных факторов риска.
- По объекту принятия решения избран показатель, который наилучшим образом характеризует эффективность этого решения.
- По объекту принятия решения избран показатель, характеризующий уровень его риска.
- Имеется конечное количество альтернатив принятия решения.
- Имеется конечное число ситуаций развития события под влиянием изменения факторов риска.
- По каждому сочетанию альтернатив принятия решений и ситуаций развития события может быть определен конечный показатель эффективности решения.
- По каждой из рассматриваемой ситуации возможна или невозможна оценка вероятности ее реализации. Возможность осуществления оценки вероятности разделяет всю систему принимаемых рискованных решений на ранее рассмотренные условия их обоснования («условия риска» или «условия неопределенности»).
- Выбор решения осуществляется по наилучшей из рассматриваемых альтернатив.

Разделение ошибочных решений на ошибки первого и второго рода вызвано тем, что последствия от разного рода ошибочных решений принципиально различаются в части того, что упущенный выигрыш оказывает меньшее влияние на ситуацию, чем реализованный проигрыш. Вместе с тем, классификация ошибок первого и второго рода допустима только в ситуациях, когда ведется точный учёт и анализ рисков.

Проверка статистической гипотезы означает проверку согласования исходных выборочных данных с выдвинутой основной гипотезой. При этом возможно возникновение двух ситуаций – основная гипотеза может подтвердиться, а может и опровергнуться. Следовательно, при проверке статистических гипотез существует вероятность допустить ошибку, приняв или опровергнув верную гипотезу. При проверке статистических гипотез можно допустить ошибки первого или второго рода. Ошибкой первого рода называется ошибка, состоящая в опровержении верной гипотезы. Ошибкой второго рода называется ошибка, состоящая в принятии ложной гипотезы. Уровнем значимости α называется вероятность совершения ошибки первого рода. Значение уровня значимости α обычно задаётся близким к нулю (например, 0,05; 0,01; 0,02 и т. д.), потому что чем меньше значение уровня значимости, тем меньше вероятность совершения ошибки первого рода, состоящую в опровержении верной гипотезы H_0 . Проверка справедливости статистических гипотез осуществляется с помощью различных

статистических критериев. Статистическим критерием называется случайная величина, которая используется с целью проверки нулевой гипотезы. Статистические критерии называются соответственно тому закону распределения, которому они подчиняются, т. е. F-критерий подчиняется распределению Фишера-Снедекора, T-критерий подчиняется распределению Стьюдента, U-критерий подчиняется нормальному распределению. Наблюдаемым значением статистического критерия называется значение критерия, которое рассчитано по выборочной совокупности, подчиняющейся определённому закону распределения. Множество всех возможных значений выбранного статистического критерия делится на два непересекающихся подмножества. Первое подмножество включает в себя те значения критерия, при которых основная гипотеза отвергается, а второе подмножество – те значения критерия, при которых основная гипотеза принимается. Критической областью называется множество возможных значений статистического критерия, при которых основная гипотеза отвергается. Областью принятия гипотезы или областью допустимых значений называется множество возможных значений статистического критерия, при которых основная гипотеза принимается. Если наблюдаемое значение статистического критерия, рассчитанное по данным выборочной совокупности, принадлежит критической области, то основная гипотеза отвергается. Если наблюдаемое значение статистического критерия принадлежит области принятия гипотезы, то основная гипотеза принимается. Критическими точками или квантилями называются точки, разграничивающие критическую область и область принятия гипотезы. Критические области могут быть как односторонними, так и двусторонними.

Часто делают выборку, чтобы определить аргументы против гипотезы относительно популяции (генеральной совокупности). Этот процесс известен как проверка гипотез (проверка статистических гипотез или проверка значимости), он представляет количественную меру аргументов против определенной гипотезы.

Установлено 5 стадий при проверке гипотез: определение нулевой (H_0) и альтернативной гипотезы (H_1) при исследовании. Определение уровня значимости критерия. Отбор необходимых данных из выборки. Вычисление значения статистики критерия, отвечающей H_0 . Вычисление критической области, проверка статистики критерия на предмет попадания в критическую область. Интерпретация достигнутого уровня значимости p и результатов. Определение нулевой и альтернативной гипотез, уровня статистической значимости

При проверке значимости гипотезу следует формулировать независимо от используемых при ее проверке данных (до проведения проверки). В таком случае можно получить действительно продуктивный результат. Всегда проверяют нулевую гипотезу (H_0), которая отвергает эффект (например, разница средних равняется нулю) в популяции. Например, при сравнении показателей курения у мужчин и женщин в популяции нулевая гипотеза H_0 означала бы, что показатели курения одинаковые у женщин и мужчин в популяции.

Затем определяют альтернативную гипотезу (H_1), которая принимается, если нулевая гипотеза неверна. Альтернативная гипотеза больше относится к той теории, которую собираются исследовать. Итак, на этом примере альтернативная гипотеза H_1 заключается в утверждении, что показатели курения различны у женщин и мужчин в популяции.

Разницу в показателях курения не уточнили, т.е. не установили, имеют ли в популяции мужчины более высокие или более низкие показатели, чем женщины. Такой подход известен как двусторонний критерий, потому что учитывают любую возможность, он рекомендуется постольку, поскольку редко есть уверенность заранее в направлении какого-либо различия, если таковое существует.

В некоторых случаях можно использовать односторонний критерий для гипотезы H_1 , в котором направление эффекта задано. Его можно применить, например, если рассматривать заболевание, от которого умерли все пациенты, не получившие лечения; новый препарат не мог бы ухудшить положение дел.

Вопросы для самоконтроля:

1. Общая проблема решения.
2. Функция потерь.
3. Принятие решений.
4. Анализ ошибок.
5. Оценка гипотезы.

Литература: [1]; [2]; [23].

Раздел 2. Инфокоммуникационные системы и технологии, проблемы и способы их решения

Тема № 5. Численные методы в задачах автоматизации. Оконные преобразования. Вейвлет-анализ

Материалы лекций

Наиболее распространенными методами поиска функциональных зависимостей являются аппроксимация, интерполяция и экстраполяция.

Аппроксимация - научный метод, состоящий в замене одних объектов другими, в каком-то смысле близкими к исходным, но более простыми. Аппроксимация позволяет исследовать числовые характеристики и качественные свойства объекта, сводя задачу к изучению более простых или более удобных объектов.

Интерполяция - способ нахождения промежуточных значений величины по имеющемуся дискретному набору известных значений.

Экстраполяция - метод научного исследования, состоящий в распространении выводов, полученных из наблюдения над одной частью явления, на другую его часть; научное прогнозирование событий.

Наиболее известный и распространенный метод аппроксимации данных – метод наименьших квадратов. Этот математический метод основан на минимизации суммы квадратов отклонений некоторых функций от искомых переменных.

Пусть функция $y=f(x)$ задана таблицей своих значений: $y_i = f(x_i)$, $i=0,1,n$. Требуется найти многочлен фиксированной степени m , для которого среднее квадратичное

отклонение (СКО) $\sigma = \sqrt{\frac{1}{n+1} \sum_{i=0}^n (P_m(x_i) - y_i)^2}$ минимально.

Так как многочлен $P_m(x) = a_0 + a_1x + a_2x^2 + \dots + a_mx^m$ определяется своими коэффициентами, то фактически нужно подобрать набор коэффициентов $a_0, a_1, \dots, a_m$, минимизирующий функцию

$$\Phi(a_0, a_1, \dots, a_m) = \sum_{i=0}^n (P_m(x_i) - y_i)^2 = \sum_{i=0}^n \left(\sum_{j=0}^m a_j x_i^j - y_i \right)^2.$$

Используя необходимое условие экстремума, $\frac{\partial \Phi}{\partial a_k} = 0$, $k=0,1,m$ получаем так называемую нормальную систему метода наименьших квадратов:

$$\sum_{j=0}^m \left(\sum_{i=0}^n x_i^{j+k} \right) a_j = \sum_{i=0}^n y_i x_i^k, \quad k=0,1,m.$$

Полученная система есть система алгебраических уравнений относительно неизвестных $a_0, a_1, \dots, a_m$. Можно показать, что определитель этой системы отличен от нуля, то есть решение существует и единственно. Однако при высоких степенях m система является плохо обусловленной. Поэтому метод наименьших квадратов применяют для нахождения многочленов, степень которых не выше 5. Решение нормальной системы можно найти, например, методом Гаусса.

Запишем нормальную систему наименьших квадратов для двух простых случаев: $m=0$ и $m=2$. При $m=0$ многочлен примет вид: $P_0(x) = a_0$. Для нахождения неизвестного

коэффициента a_0 имеем уравнение: $(n+1)a_0 = \sum_{i=0}^n y_i$. Получаем, что коэффициент a_0 есть среднее арифметическое значений функции в заданных точках.

Если же используется многочлен второй степени $P_2(x) = a_0 + a_1x + a_2x^2$, то нормальная система уравнений примет вид:

$$\begin{cases} (n+1)a_0 + \left(\sum_{i=0}^n x_i\right)a_1 + \left(\sum_{i=0}^n x_i^2\right)a_2 = \sum_{i=0}^n y_i \\ \left(\sum_{i=0}^n x_i\right)a_0 + \left(\sum_{i=0}^n x_i^2\right)a_1 + \left(\sum_{i=0}^n x_i^3\right)a_2 = \sum_{i=0}^n y_i x_i \\ \left(\sum_{i=0}^n x_i^2\right)a_0 + \left(\sum_{i=0}^n x_i^3\right)a_1 + \left(\sum_{i=0}^n x_i^4\right)a_2 = \sum_{i=0}^n y_i x_i^2 \end{cases}$$

За критерий погрешности результата МНК обычно берется среднеквадратичное отклонение по узлам приближения

$$S = \frac{1}{n} \sum_{i=1}^n (P(x_i) - y_i)^2$$

Интерполяция.

Существует много различных подходов к интерполяции данных, рассмотрим основные из них: линейную интерполяцию и интерполяцию полиномами.

Линейная интерполяция — интерполяция алгебраическим двучленом $P_1(x) = ax + b$ функции f , заданной в двух точках x_0 и x_1 отрезка $[a, b]$. В случае, если заданы значения в нескольких точках, функция заменяется кусочно-линейной функцией. То есть заданные точки (x_i, y_i) при $(i=0..n)$ соединяются прямолинейными отрезками, и функция $f(x)$ приближается ломаной с вершинами в данных точках. Уравнения каждого отрезка ломаной в общем случае разные. Поскольку имеется n интервалов (x_{i-1}, x_i) , то для каждого из них в качестве уравнения интерполяционного многочлена используется уравнение прямой, проходящей через две точки. В частности, для i -го интервала можно написать уравнение прямой, проходящей через точки (x_{i-1}, y_{i-1}) и (x_i, y_i) , в виде

$$\frac{y - y_{i-1}}{y_i - y_{i-1}} = \frac{x - x_{i-1}}{x_i - x_{i-1}}$$

Отсюда

$$\begin{aligned} y &= a_i x + b_i, \quad x_{i-1} \leq x \leq x_i \\ a_i &= \frac{y_i - y_{i-1}}{x_i - x_{i-1}}, \quad b_i = y_{i-1} - a_i x_{i-1} \end{aligned}$$

Следовательно, при использовании линейной интерполяции сначала нужно определить интервал, в который попадает значение аргумента x , а затем подставить его в формулу и найти приближенное значение функции в этой точке.

Интерполяция полиномами.

Также можно проинтерполировать данные гладкой, не кусочной функцией. Для этого часто используют так называемые полиномы Лагранжа. Интерполяционный многочлен Лагранжа — многочлен минимальной степени, принимающий данные значения в данном

наборе точек. Для $n+1$ пар чисел $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$, где все x_j различны, существует единственный многочлен $L(x)$ степени не более n , для которого $L(x_j) = y_j$.

В простейшем случае ($n=1$) — это линейный многочлен, график которого — прямая, проходящая через две заданные точки. Лагранж предложил способ вычисления таких многочленов:

$$L(x) = \sum_{i=0}^n y_i l_i(x),$$

где базисные полиномы определяются по формуле:

$$l_i(x) = \prod_{j=0, j \neq i}^n \frac{x-x_j}{x_i-x_j} = \frac{x-x_0}{x_i-x_0} * \dots * \frac{x-x_n}{x_i-x_n}.$$

$l_i(x)$ обладают следующими свойствами:

являются многочленами степени n ,

$$l_i(x_i) = 1,$$

$$l_i(x_j) = 0 \text{ при } j \neq i,$$

Отсюда следует, что $L(x)$, как линейная комбинация $l_i(x)$, может иметь степень не больше n , и $L(x_i) = y_i$.

Экстраполяция.

Экстраполяция – распространение результатов, полученных из наблюдений над одной частью некоторого явления, на другую его часть. Экстраполяция функции – продолжение функции за пределы её области определения, при котором продолженная функция (как правило, аналитическая) принадлежит заданному классу функций. Экстраполяция функций обычно происходит с помощью формул, в которых используется информация о поведении функции в некотором конечном наборе точек (в узлах экстраполяции), принадлежащих её области определения.

Понятие интерполирования функций иногда употребляется в качестве противопоставления понятию экстраполирования, когда конструктивно восстанавливаются (возможно, приближённо) значения функций в областях их определений.

Например, если используется информация о значениях функции, определённой на отрезке $[a; b]$, в узлах x_k из этого отрезка ($k = 0; 1; \dots; n$), то интерполяционный многочлен Лагранжа, поскольку он определён на всей числовой оси, является, в частности, экстраполяцией этой функции вне отрезка $[a; b]$ в классе многочленов степени не выше n .

Численное дифференцирование и интегрирование - вычисление производных и интегралов по приближенным разностным формулам. Если функция задана в виде таблицы или имеет очень сложный вид и алгоритм расчета, то ее можно дифференцировать или интегрировать только численным способом. Для численного дифференцирования и интегрирования существует ряд формул численного приближенного вычисления. Для получения таких формул можно использовать два варианта построения или вывода:

- использование разложения в ряд Тейлора,
- дифференцирование интерполяционных полиномов или аппроксимационных функций.

Рассмотрим 2 формулы разложения в ряд Тейлора:

$$F(x_i+h) = F(x_i) + F'(x_i)*h + F''(x_i)*h^2/2 + F'''(x_i)*h^3/6 + F^{(4)}(x_i)*h^4/24 + \dots$$

$$F(x_i - h) = F(x_i) - F'(x_i)*h + F''(x_i)*h^2/2 - F'''(x_i)*h^3/6 + F^{(4)}(x_i)*h^3/24 - \dots$$

Многообразием в формуле обозначается все последующие (до бесконечности) элементы ряда. Существуют специальные оценки этого остатка. В частности в матанализе доказывается, что остаток формулы Тейлора всегда можно представить как $O(h^n)$. Это значит, что в формуле оставлена максимальная степень $n-1$.

Замечание. Выражение $O(h^n)$ является асимптотической оценкой роста или уменьшения некоей величины, которая зависит от h . Практически можно считать, что это произведение неизвестной константы и $h^n \Rightarrow O(h^n) = \text{const} * h^n$. Смысл такого выражения в том, что при стремлении h к бесконечности величина будет изменяться со скоростью пропорциональной h^n . Таким образом, асимптотические оценки позволяют сравнивать скорости роста или уменьшения различных формул.

Теперь наши формулы будут иметь вид:

$$F(x_i+h) = F(x_i) + F'(x_i)*h + F''(x_i)*h^2/2 + F'''(x_i)*h^3/6 + F^{(4)}(x_i)*h^3/24 + O(h^4) \quad (A)$$

$$F(x_i - h) = F(x_i) - F'(x_i)*h + F''(x_i)*h^2/2 - F'''(x_i)*h^3/6 + F^{(4)}(x_i)*h^3/24 + O(h^4) \quad (B)$$

$O(h^n)$ является так же оценкой погрешности для формулы.

Говорят, что формула имеет порядок погрешности n , если последний элемент в формуле $O(h^n)$. В нашем случае мы ограничились 4 порядком погрешности.

Теперь, используя эти разложения, построим формулы для разностных производных. Если в формуле (A) оставить только 2-ю степень в разложении и выразить из этой формулы производную, то мы получим:

$$F(x_i+h) = F(x_i) + F'(x_i)*h + O(h^2) \Rightarrow F'(x_i) = (F(x_i+h) - F(x_i))/h + O(h^1)$$

Это выражение определяет формулу правой разностной производной $\frac{f(x_i+h) - f(x_i)}{h}$, имеющей 1-й порядок точности. Аналогично из формулы (B) получим левую разностную

производную $\frac{f(x_i) - f(x_i-h)}{h}$, имеющую так же 1-й порядок точности.

Среднее между этими производными – центральная разностная производная, имеет 2-й порядок точности и формулу $\frac{f(x_i+h) - f(x_i-h)}{2h}$ (это легко доказать, если вычесть из (A) (B) и разделить разность на $2h$).

Для получения формулы 2-й производной нужно сложить (A) и (B) и разделить на h^2 . Получим формулу второй разностной производной $\frac{f(x_i-h) + f(x_i+h) - 2f(x_i)}{h^2}$, которая имеет 2-й порядок точности.

Замечание. Использование формул с более высоким порядком точности более выгодно. Это можно пояснить простым примером. Пусть есть 3 формулы, имеющие 1,2,3 порядки точности соответственно. Уменьшим шаг h в 10 раз. Тогда погрешность 1-й формулы уменьшится в 10 раз, 2-й в 100 раз, а 3-й в 1000 раз. Таким образом, достичь заданной погрешности легче при использовании 3-й формулы.

Получить формулы численного дифференцирования можно и другим способом. Если записать формулу для полинома Лагранжа и продифференцировать ее, то можно получить различные формулы разностных производных.

При этом существует некоторое правило (которое обычно верно, но в некоторых случаях не соблюдается) – порядок погрешности при дифференцировании интерполяционного полинома уменьшается на 1. Как известно, формула оценки

$$\omega_n(x) = \sum_{k=0}^n (x - x_k)$$

погрешности полинома Лагранжа, пропорциональна полиному. Если считать, что $(x - x_k) = h$, то оценка погрешности интерполирования является $O(h^{n+1})$. Таким образом, порядок точности интерполяционного полинома на 1 больше его порядка. После дифференцирования порядок точности уменьшится на 1 $\Rightarrow$ для получения формулы аналогичной правой разностной нужно дифференцировать полином 1-го порядка, для центральной разностной полином 2-го порядка, а для 2-й разностной уже полином 3-го порядка (так как формула дифференцируется дважды).

В общем случае задача численного интегрирования сводится к нахождению

$$I = \int_a^b f(x) dx$$

интеграла с помощью приближенных формул. Численные методы при решении задачи интегрирования используются в следующих случаях:

а) Функция задаётся в виде таблицы значений или её значение вычисляется, как решение другой задачи (функция задаётся неявным видом).

б) Функция задана явно, но первообразную не возможно записать в явном виде.

Методы, используемые для решения задачи численного интегрирования можно разделить на однократные и многократные.

Однократный метод реализуется путем замены интегрируемой функции на ее аппроксимацию, которая имеет явную первообразную. Это дает возможность указать явную формулу для вычисления определенного интеграла.

Многократный метод - используем однократный метод, но весь отрезок интегрирования от a до b делим на n частей с шагом $h = (b-a)/n$ с помощью точек $x_i = a + ih$.

На каждом участке $[x_{i-1}, x_i]$ интеграл вычисляется однократным методом, а затем используя свойство аддитивности интегралов общий интеграл вычисляется как сумма интегралов на отдельных участках:

$$I = \int_a^b f(x) dx = \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} f(x) dx$$

Таким образом, основа формулы многократного метода – формула метода однократного. Смысл использования многократного метода – увеличение точности вычисления. Как правило формулы однократных методов таковы, что их погрешность быстро уменьшается при уменьшении отрезка интегрирования.

Одним из основных вариантов получения формул однократного метода является метод интегрирования интерполяционных полиномов. Если использовать полином Лагранжа, то получаются формулы, которые принято называть **формулами Ньютона-Котеса**:

$$\int_a^b L_n(x) dx = (b-a) \sum_{i=1}^n f_i C_i \quad - \text{формула Ньютона – Котеса.}$$

$$C_i = \frac{1}{n \prod_{j \neq i} (i - j)} *$$

$$* \int_0^n \prod_{i \neq j} (x' - j) dx'$$

- коэффициенты Котеса. Для получения этого выражения для коэффициентов необходимо подставить в интеграл формулу полинома Лагранжа для равноотстоящих узлов. Затем делается замена переменных $x' = n*(x-a)/(b-a) \Rightarrow$ и находим выражения, которые необходимо подставить в интеграл $(x_i - x_j) = (b-a)*(i-j)/n$, $(x - x_j) = (b-a)*(x' - j)/n$, $dx' = n*dx/(b-a)$.

Численные методы решения экстремальных задач.

Пусть $f(x)$ - функция, определенная на некотором множестве $X (x \in X)$. Будем рассматривать задачу минимизации функции $f(x)$. Любая задача максимизации функции $f(x)$ на X равносильна задаче минимизации функции $-f(x)$ на том же множестве X . Поэтому можно ограничиться лишь изучением задач минимизации.

Пусть $f(x)$ кусочно-непрерывная и кусочно-гладкая функция на отрезке $[a, b]$ ($[a, b] \in X$). Это значит, что на $[a, b]$ может существовать лишь конечное число точек, в которых функция $f(x)$ либо терпит разрыв первого рода, либо непрерывна, но не имеет производной. Тогда точками экстремума функции $f(x)$ на $[a, b]$ могут быть лишь те точки, в которых выполняется одно из следующих условий: 1) $f(x)$ терпит разрыв; 2) $f(x)$ непрерывна, но производная не существует; 3) производная $f'(x)$ существует и равна нулю; 4) $x = a$ или $x = b$. Такие точки принято называть точками подозрительными на экстремум. Поиск точек экстремума функции начинают с нахождения всех точек, подозрительных на экстремум. После того, как такие точки найдены, проводят дополнительное исследование и отбирают среди них те, которые являются точками локального минимума (максимума).

Чтобы найти глобальный минимум (максимум) функции $f(x)$ на $[a, b]$, нужно перебрать все точки локального минимума (максимума) на $[a, b]$ и среди них выбрать точку с наименьшим (наибольшим) значением функции, если таковая существует (если вместо $[a, b]$ имеем дело с $\mathbb{R}$, то следует изучить поведение функции при $x \rightarrow \infty$ или $x \rightarrow -\infty$).

В общем случае функция $f(x)$ ($x \in X$) может иметь несколько экстремумов (минимумов или максимумов). Задача поиска экстремумов сводится к их локализации и уточнению значений X и $f(x)$ в точке экстремума. Все рассмотренные ниже численные методы предполагают, что локализация экстремумов каким-либо образом произведена (например, графически или аналитически) и задача численных методов будет состоять в уточнении полученных результатов с заданной точностью ε . В дальнейшем для функций одной переменной под экстремумом будем подразумевать минимум $f(x)$. Будем считать, что $x \in [a, b]$, где a и b границы интервала поиска. В пределах отрезка $[a, b]$ функция $f(x)$

необязательно непрерывная, могут существовать разрывы первого рода. Достаточно чтобы функция $f(x)$ на отрезке $[a, b]$ была унимодальной, то есть, содержащей на указанном отрезке один минимум.

Метод равномерного поиска.

Этот метод основан на том, что переменной x присваиваются значения $x_i = x_0 + ih$ с шагом $h = \text{const}$ (шагом поиска), где $i=0,1,2,\dots$ и вычисляются значения $f(x)$ в соседних точках. Если $f(x_{i+1}) \leq f(x_i)$, то переменной x дается новое приращение. Как только становится $f(x_{i+1}) > f(x_i)$, поиск останавливается и предпоследняя точка считается ответом.

Выбор x_0 (начального значения переменной x) определяется пользователем. Шаг поиска - фактическая погрешность определения результата. При поиске решения на отрезке, обычно в качестве начального приближения берут один из его концов, а при изменении переменной x предусматривается проверка на выход ее за границу отрезка.

Метод поразрядного приближения. Является разновидностью метода равномерного поиска и реализуется следующим образом:

1. Задаем начальное приближение $x = x_0$ слева от минимума функции $f(x)$ и вычисляем $f(x_0)$, задаем начальный шаг поиска h (выбирается вычислителем), точность ε определения результата поиска (для переменной x), берем $i=0$.

2. Задаем $x_{i+1} = x_i + h$ и вычисляем $f(x_{i+1})$.

3. Проверяем условие $f(x_{i+1}) \leq f(x_i)$, если оно выполняется, то идем к пункту 2, увеличивая i на 1.

4. Проверяем условие $|h| \geq \varepsilon$. Если оно выполняется, полагаем $h = -h/10$, увеличиваем i на 1 и идем к пункту 2, т.е. обеспечиваем поиск минимума в другом направлении с шагом $h/10$.

5. Выводим на печать полученное значение для переменной x и функции $f(x)$.

Метод деления отрезка пополам (или метод дихотомии).

Поиск минимума $f(x)$ на отрезке $[a, b]$ на каждом шаге начинается с выбора двух точек $x_1 = \frac{a+b-\delta}{2}$ и $x_2 = \frac{a+b+\delta}{2}$, где $\delta > 0$ -постоянная, являющаяся параметром метода.

Величина δ выбирается вычислителем и может определяться целесообразным количеством верных десятичных знаков при задании аргумента x . Точки x_1 и x_2 расположены симметрично на $[a, b]$ относительно его середины и при малых δ делят его почти пополам. Уточняем положение экстремума с заданной точностью ε .

Метод реализуется следующим алгоритмом:

1. Проверяем условие $|b-a| < \varepsilon$. Если условие выполняется, идем к пункту 6.
2. Делим интервал поиска $[a, b]$ точками $x_1 = \frac{a+b-\delta}{2}$ и $x_2 = \frac{a+b+\delta}{2}$.
3. Для значений x_1 и x_2 вычисляем $f(x_1)$ и $f(x_2)$.
4. Проверяем условие $f(x_2) < f(x_1)$. Если оно выполняется, полагаем $a = x_1$ и идем к пункту 1.
5. Полагаем $b = x_2$ и идем к пункту 1.
6. Выводим на печать $x = \frac{a+b}{2}$ и $f(x)$.

Системой дифференциальных уравнений называется система вида

$$\begin{cases} \frac{dy_1}{dx} = f_1(x, y_1 \dots y_n) \\ \frac{dy_2}{dx} = f_2(x, y_1 \dots y_n) \\ \dots\dots\dots \\ \frac{dy_n}{dx} = f_n(x, y_1 \dots y_n) \end{cases}$$

или

$$\frac{dy_i}{dx} = f_i(x, y_1 \dots y_n), i = \overline{1, n}$$

где x - независимый аргумент,

y_i - зависимая функция, $i = \overline{1, n}$,

$y_i|_{x=x_0} = y_{i0}$ - начальные условия.

Функции $y_i(x)$, при подстановке которой система уравнений обращается в тождество, называется решением системой дифференциальных уравнений.

Численные методы решения систем дифференциальных уравнений.

1. Метод Эйлера.

$$y_{ij+1} = y_{ij} + hf_i(x_i, y_{1j} \dots y_{nj})$$

$$i = \overline{1, n}$$

j - номер шага.

$$x_{j+1} = x_j + h$$

2. Модифицированный метод Эйлера.

$$k_{i1} = h * f_i(x_j, y_{1j} \dots y_{nj})$$

$$k_{i1} = h * f_i(x_j + h, y_{1j} + k_{i1} \dots y_{nj} + k_{i2})$$

$$y_{ij+1} = y_{ij} + (k_{i1} + k_{i2})/2$$

$$x_{j+1} = x_j + h$$

3. Метод Рунге-Кутта четвертого порядка.

$$\begin{aligned}
k_{i1} &= hf_i(x_j, y_{1j} \dots y_{nj}) \\
k_{i2} &= hf_i(x_j + h/2, y_{2j} + k_{i1}/2, \dots, y_{nj} + k_{n1}/2) \\
k_{i3} &= hf_i(x_j + h/2, y_{2j} + k_{i2}/2, \dots, y_{nj} + k_{n2}/2) \\
k_{i4} &= hf_i(x_j + h, y_{1j} + k_{i2}, \dots, y_{nj} + k_{n3}) \\
y_{ij+1} &= y_{ij} + (k_{i1} + 2k_{i2} + 2k_{i3} + k_{i4})/6 \\
x_{j+1} &= x_j + h
\end{aligned}$$

Спектральный анализ - один из методов обработки сигналов, который позволяет охарактеризовать частотный состав измеряемого сигнала. Преобразование Фурье является математической основой, которая связывает временной или пространственный сигнал (или же некоторую модель этого сигнала) с его представлением в частотной области. Важную роль в спектральном анализе играют методы статистики, поскольку сигналы, как правило, имеют случайный характер или зашумлены при распространении или измерении. Если бы основные статистические характеристики сигнала были точно известны, или их можно было определить по конечному интервалу этого сигнала, то спектральный анализ представлял бы собой отрасль "точной науки". Однако, в действительности по отрезку сигнала можно получить только оценку его спектра. Различие между спектральными оценками, получаемыми в результате обработки одного и того же отрезка сигнала разными методами, можно объяснить различием допущений, принятых относительно данных, различными способами усреднения и т.п. Если априори характеристики сигнала не известны, нельзя сказать какие из оценок лучше.

Преобразование Фурье - математическая основа спектрального анализа. Кратко обсудим разные виды преобразования Фурье.

Начнем с преобразования Фурье непрерывного во времени сигнала

$$X(f) = \int_{-\infty}^{+\infty} x(t) \exp(-j 2 \pi f t) dt \equiv \mathcal{F}\{x(t)\}$$

которое идентифицирует частоты и амплитуды тех комплексных синусоид (экспонент), на которые разлагается некоторое произвольное колебание.

Обратное преобразование

$$x(t) = \int_{-\infty}^{+\infty} X(f) \exp(j 2 \pi f t) df \equiv \mathcal{F}^{-1}\{X(f)\}$$

Существование прямого и обратного преобразования Фурье определяется рядом условий. Достаточное - абсолютная интегрируемость сигнала

$$\int_{-\infty}^{+\infty} |x(t)| dt < \infty$$

Менее ограничительное достаточное условие - конечность энергии сигнала

$$\int_{-\infty}^{+\infty} |x(t)|^2 dt < \infty$$

Приведем ряд основных свойств преобразования Фурье и функций, заметив, что прямоугольное окно определяется выражением

$$w(x) = \begin{cases} 1, & |x| < 1; \\ 1/2, & |x| = 1; \\ 0, & |x| > 1, \end{cases}$$

а функция sinc - выражением

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}.$$

Функция отсчетов во временной области определяется выражением

$$\Psi_T(x) = \sum_{n=-\infty}^{n=\infty} \delta(x - nT).$$

Таблица 1. Основные свойства преобразования Фурье и функции

Свойство, функция	Функция	Преобразование
Линейность	$ag(t) + bh(t)$	$aG(f) + bH(f)$
Сдвиг по времени	$h(t - t_0)$	$H(f) \exp(-j2\pi f t_0)$
Сдвиг по частоте (модуляция)	$h(t) \exp(j2\pi f_0 t)$	$H(f - f_0)$
Масштабирование	$(1/ a)h(t/a)$	$H(af)$
Теорема свертки во временной области	$g(t)*h(t)$	$G(f)H(f)$
Теорема свертки в частотной области	$g(t)h(t)$	$G(f)*H(f)$
Функция окна	$A_w(t/T)$	$2AT \text{sinc}(2Tf)$
Функция sinc	$2AF \text{sinc}(2Ft)$	$A_w(f/F)$
Импульсная функция	$\delta(t)$	1
Функция отсчетов	$\Psi_T(f)$	$F\Psi_F(f), F=1/T$

Вейвлет-функция Ψ имеет нулевое среднее значение, а вейвлет-преобразование функции $f \in L^2(R)$ ($L^2(R)$ -пространство Лебега)

$$Wf(s,u) = \int f(t) \frac{1}{\sqrt{s}} \Psi\left(\frac{t-u}{s}\right) dt,$$

где $s, u \in R, s \neq 0$, t - время, u - параметр сдвига, s - параметр масштаба, R - множество действительных чисел,

измеряет изменение функции f в окрестности точки u , размер которой пропорционален s . При стремлении масштаба s к нулю вейвлет-коэффициенты характеризуют свойства функции f в окрестности u .

Если вейвлет-функция Ψ имеет l нулевых моментов, т.е.

$$\int_{-\infty}^{+\infty} t^k \Psi(t) dt = 0, k = 0, \bar{l} - 1,$$

и n производных, то для функции f , удовлетворяющей равномерному условию Липшица α , $\alpha \leq l$, на $[a, b]$, существует $A > 0$ такое, что

$$\forall (s, u) \in [a, b] \|Wf(s, u)\| \leq As^{\alpha+1/2}. \quad (1)$$

Обратно, если f ограничена и $Wf(s, u)$ удовлетворяет (1) для нецелого $\alpha < l$, тогда f удовлетворяет равномерному условию Липшица α на $[a + \varepsilon, b - \varepsilon]$ при любом $\varepsilon > 0$.

Неравенство (1) является условием асимптотического убывания абсолютных значений вейвлет-коэффициентов $\|Wf(s, u)\|$ при $s \rightarrow 0$. Таким образом, убывание амплитуды вейвлет-преобразования в зависимости от масштаба связано с гладкостью Липшица функции.

Если функция $f(t)$ имеет сложную структуру, характеризовать гладкость f в некоторой точке $t = v$ бывает достаточно трудно, потому что f может иметь различные виды особенностей, присутствующих в окрестностях точки v . Будем считать, что функция f в окрестности точки v имеет локальную особенность, если $\|Wf(s, u)\|$ не удовлетворяет условию (1) в окрестности точки v .

Обработка дискретных последовательностей данных базируется на дискретной схеме вейвлет-разложения. Полагают, неполная выборка параметров преобразования полностью определяет функцию, если она может быть полностью восстановлена. Теория вейвлетов имеет условия точного восстановления функции, которые накладывают ограничения на используемые семейства вейвлет-функций. Семейства таких функций определяют дискретные вейвлеты, схему разложения по таким базисам называют дискретным вейвлет-преобразованием.

Если Ψ порождает базис Рисса с нормой выбора $u_0 = 1$, то Ψ называется R -функцией: $\Psi_{j,n}(t) := 2^{j/2} \Psi(2^j t - n)$, j - параметр масштаба, n - параметр сдвига, и дискретное вейвлет-преобразование определяется следующим образом:

$$f(t) = \sum_{j,n \in \mathbb{Z}} c_{j,n} \Psi_{j,n}(t),$$

где $c_{j,n} = \langle f, \Psi_{j,n} \rangle$, $\mathbb{Z}$ - множество целых чисел. Эти бесконечные ряды называются вейвлет-рядами и сходятся в $L^2(R)$.

Вейвлет-образ природного временного ряда.

Формальная модель природного временного ряда $f(t)$ имеет вид:

$$f(t) = f_1(t) + f_2(t) + \dots + f_s(t),$$

где $f_j(t)$ - разномасштабные ортогональные компоненты временного ряда, имеющие различную структуру.

Поскольку компоненты $f_j(t)$ имеют различную структуру и, учитывая локальный характер содержащихся в них локальных особенностей, их разномасштабность и разнообразие по форме, наиболее подходящим пространством для их представления является вейвлет-пространство:

$$f_j(t) = \sum_n c_{j,n} \Psi_{j,n}(t),$$

где $f_j(t) \in L^2(R)$, $\{\Psi_{j,n}(t)\}_{j,n \in Z}$ – ортогональный вейвлет-базис пространства $L^2(R)$. Здесь вейвлет-коэффициенты $c_{j,n} = \langle f, \Psi_{j,n} \rangle$, рассматриваются как результат отображения $f(t)$ в пространство с разрешением j , который определяется свойствами используемой вейвлет-функции Ψ . В зависимости от задачи возможны различные способы формирования критериев выбора вейвлет-функции Ψ . Решая задачу аппроксимации функции в базисе ортогональных функций в качестве основных критериев логично определить:

- минимизацию числа аппроксимирующих слагаемых;
- минимизацию погрешности аппроксимации.

С целью возможности построения адаптивных аппроксимирующих вейвлет-схем, будем использовать нелинейные преобразования: $f_M(t)$ – проекция $f(t)$ на M векторов, индексы которых содержатся в I_M (I_M – множество индексов):

$$f_M(t) = \sum_{j,n \in I_M} \langle f, \Psi_{j,n} \rangle \Psi_{j,n}(t).$$

В этом случае приближение функции f выполняется M векторами, зависящими от ее структуры.

Погрешность аппроксимации есть сумма оставшихся коэффициентов:

$$\varepsilon[M] = \|f - f_M\|^2 = \sum_{j,n \notin I_M} |\langle f, \Psi_{j,n} \rangle|^2.$$

Чтобы минимизировать эту погрешность, индексы в I_M должны соответствовать M векторам, имеющим скалярные произведения с наибольшей амплитудой $|\langle f, \Psi_{j,n} \rangle|$. Поэтому минимизация числа аппроксимирующих слагаемых и минимизация погрешности аппроксимации могут быть достигнуты выбором базиса

$$\beta = \{\Psi_{j,n}\}_{j,n \in I_M},$$

обеспечивающего как можно большее число пренебрежимо малых коэффициентов $c_{j,n} = \langle f, \Psi_{j,n} \rangle$.

Предполагая, что компонента

$$e(t) = \sum_{j,n \notin I_M} \langle f, \Psi_{j,n} \rangle \Psi_{j,n}(t)$$

является следствием влияния шумового фактора, и что она является некоррелированной и аддитивной

$$y(t) = f(t) + e(t),$$

на основе получаем представление в вейвлет-пространстве случайного временного ряда:

$$y(t) = \sum_{j,n \in I_M} \langle f, \Psi_{j,n} \rangle \Psi_{j,n}(t) + \sum_{j,n \notin I_M} \langle f, \Psi_{j,n} \rangle \Psi_{j,n}(t) = \sum_{j,n \in I_M} c_{j,n} \Psi_{j,n}(t) + e(t). \quad (2)$$

Каждая $\Psi_{j,n}$ определяет свой частотно-временной прямоугольник в частотно-временной плоскости с временным n и масштабным j параметрами. Важность такого представления состоит в том, что различные составляющие временного ряда рассортировываются и хранятся в различных пространствах W_j с разрешением j :

$$L^2(R) = \sum_{j \in \mathbb{Z}} W_j := \dots \sqcup W_{-1} \sqcup W_0 \sqcup W_1 \sqcup \dots, \text{ где } W_j := \text{clos}_{L^2(R)}(\Psi_{j,n}; n \in \mathbb{Z}).$$

При этом наименьший анализируемый масштаб j ограничен шагом выборки дискретного временного ряда.

Имея представление временного ряда в виде (2), операция выделения локальных особенностей временного ряда может быть построена на основе свойства (1) путем определения наибольших значений функции $\left| \langle f, \Psi_{j,n} \rangle \right|$ на малых масштабах j . Реализуем ее на основе проверки условия

$$\left| \langle f, \Psi_{j,n} \rangle \right| \leq T_j,$$

где T_j – пороговое значение, определяющее наличие в компоненте временного ряда масштаба j локальной особенности.

Путем выбора порогового значения T' мы можем реализовать выделение и идентификацию особенности в природном временном ряде.

В задачах аппроксимации временного ряда и выделения и идентификации локальных особенностей определяющими при выборе вейвлет-функции будут следующие ее характеристики: гладкость, размер носителя, число нулевых моментов. Покажем это.

Число нулевых моментов.

Вейвлет Ψ имеет l нулевых моментов, т.е.

$$\int_{-\infty}^{+\infty} t^k \Psi(t) dt = 0, k = \overline{0, l-1}.$$

Число нулевых моментов m характеризует способность вейвлет-функции выявлять особенность вида $\alpha \leq l$. Также Ψ ортогональна любому многочлену степени $l-1$, т.е. «пропускает» полиномы степени выше $l-1$. Поэтому, в окрестностях, где f гладкая функция, большое число нулевых моментов Ψ обеспечит большое число пренебрежимо малых вейвлет-коэффициентов. В этом случае мы получим аппроксимирующую схему с минимальным числом слагаемых при заданном уровне точности аппроксимации.

Размер носителя.

Вейвлет-преобразование порождает искусственные “скачки” на краях функции f , находящие отражения в коэффициентах разложения. Поскольку размер окрестности на масштабе j , содержащей возникающий при этом краевой эффект, зависит от размера носителя вейвлет-функции Ψ и определяется по формуле $h_j = 2^j * q$ где q - размер носителя используемой базисной функции. Получаем, чем меньше размер носителя q , тем меньшую погрешность мы имеем на краях f .

Если f имеет локальную особенность в некоторой точке v и если v находится внутри носителя $\Psi_{j,n}$, то, вейвлет-коэффициенты имеют большую амплитуду. Чем больше носитель, тем большее число таких коэффициентов на каждом масштабном уровне мы получаем. Чтобы минимизировать их число, мы должны использовать вейвлет-функции с наименьшим размером носителя Ψ .

Гладкость вейвлета.

Аналогично числу нулевых моментов, гладкость вейвлет-функции характеризует её способность выявлять особенность вида $\alpha \leq l$. Что касается погрешности аппроксимации, гладкость вейвлет-функции Ψ оказывает в основном «косметическое» влияние на нее: если Ψ – гладкая, то и погрешность будет гладкая.

Таким образом, при выборе вейвлет-функции мы приходим к выбору между числом нулевых моментов и размером носителя. Для ортогональных вейвлет-функций Добеши доказала, что если вейвлет Ψ имеет l нулевых моментов, то его наименьший носитель равен $2l-1$. Поэтому, если функция f имеет несколько локальных особенностей и очень гладкая между этими особенностями, необходимо использовать вейвлет с большим числом нулевых моментов; если число особенностей нарастает, лучше уменьшить размер носителя ценой уменьшения числа нулевых моментов. Ортогональные вейвлеты Добеши с компактным носителем – это единственное семейство базисных вейвлет-функций, которые имеют минимальный размер носителя при заданном числе нулевых моментов.

Вопросы для самоконтроля:

1. Интерполяция и аппроксимация.
2. Численное дифференцирование и интегрирование.
3. Численные методы решения систем дифференциальных уравнений.
4. Вейвлет-анализ.
5. Выбор вейвлет-базиса.

Литература: [1]; [2]; [5]; [7]; [9]; [14].

Тема № 6. Информационно-управляющие системы. Мобильные технологии

Материалы лекций

Типы производственных процессов. Принято разделение по типу производственных процессов на: дискретное и непрерывное, или процессное.

Непрерывное производство (Process-Costing) / процессное - это совокупность непрерывных технологических процессов, организованных в виде производственной линии, участка, цеха или предприятия в целом; диктуется характером технологии. К ним относятся предприятия пищевой, химической, фармацевтической, целлюлозно-бумажной, металлургической промышленности, энергетика, и др. Непрерывное производство характеризуется тем, что продукция в этом случае воспринимается как единой целое, и не имеет смысла раскладывать его на составные единицы в процессе производства.

Дискретное производство — тип производства, в котором исходный материал (сырье) при переработке в исходный продукт претерпевает более одного передела с прерыванием технологического процесса. К этому типу относится и позаказное (Job-Order-Costing) производство, характеризующееся тем, что предприятие изготавливает различные виды продуктов, которые могут быть выражены в количестве штук или подразделены на более или менее мелкие серии. Каждый продукт или серия продукции может быть выделена по

свойственным им признакам. Дискретный тип производства превалирует в машиностроении, приборостроении, легкой промышленности, на предприятиях по выпуску мебели, упаковок.

Причина классификации производственных процессов по признаку позаказное-непрерывное заключается в том, что это предопределяет технику измерения производственных издержек, применяемую предприятием. При позаказном производстве можно измерять количество материала и рабочего времени, затраченных на изготовление отдельного продукта или серии. При непрерывном производстве невозможно измерить затраты на отдельные продукты, и потому расчет направлен на исчисление затрат за период.

Под автоматизацией управления понимается применение программных средств, которые используются для решения задач управления на всех этапах и во всех сферах деятельности компании.

Информационно-управляющая структура производственного предприятия. Информационно-управляющая структура производственного предприятия (по данным MESA International, Manufacturing Enterprise Solutions Association — Международная ассоциация производителей систем управления производством, www.mesa.org) представляется в виде пирамиды и дает схематическое представление о структуре информационной системы автоматизации промышленного предприятия, разделенной на четыре уровня:

- на первом уровне находятся АСУТП;
- на втором уровне MES-системы;
- на третьем уровне — ERP-системы;
- на верхнем уровне находятся OLAP-системы.

АСУТП — автоматизированные системы управления технологическими процессами;

MES — (Manufacturing Execution System) — исполнительная система производства, автоматизированная система управления производством, информационно-вычислительная система. Системы такого класса решают задачи синхронизации, координируют, анализируют и оптимизируют выпуск продукции в рамках какого-либо производства в режиме реального времени.

ERP - (Enterprise Resource Planning) — Система планирования ресурсов предприятия. Основное назначение ERP — управление финансовой и хозяйственной деятельностью предприятия. ERP-система работает на самом верхнем уровне в иерархической лестнице систем управления, она затрагивает основные аспекты всех элементов производственной и торговой деятельности предприятия.

OLAP — (On-Line Analytic Processing) — Оперативный многомерный анализ данных. Аналитическая обработка в реальном времени, технология обработки информации, включающая составление и динамическую публикацию отчетов и документов. Используется аналитиками для быстрой обработки сложных запросов к базе данных. Служит для подготовки бизнес-отчетов по продажам, маркетингу, в целях управления, т.н. — data mining — добыча данных (способ анализа информации в базе данных целью отыскивания аномалий и трендов без выявления смыслового значения записей).

Рассматривая данную пирамиду, можно представить себе передачу информации по всем ступеням иерархии системы. Из производственной зоны (АСУТП) информация поступает к MES-системам, проходит стадию обработки, а затем уже обработанная

информация поступает к MES-системам, проходит стадию обработки, а затем уже обработанная информация поступает в ERP-системы, и далее — на уровень высшего менеджмента предприятия (OLAP).

ERP-системы имеют модульное устройство. Существующие в настоящее время ERP-системы можно разделить на два класса:

- системы, предназначенные для управления бизнесом (то есть процессами, в которые вовлечен продукт, уже прошедший производственную фазу);
- системы, рассчитанные на управление производством.

Модель управления предприятием может базироваться на различных концепциях в зависимости от специфики деятельности организации и принятой стратегии:

EAM (Enterprise Asset Management) — система, предназначенная для управления основными фондами. Когда производственные средства эксплуатируются в течение достаточно продолжительного срока и регулярно подвергаются ремонту, определяющим становится прогнозирование производственной загрузки и амортизации. Оба названных фактора непосредственно влияют на техническое оснащение производства и затраты, необходимые для осуществления технического обслуживания и проведения капитального ремонта;

BSC (Balanced Scorecard) — так называемая сбалансированная система показателей. Позволяет оценивать деятельность подразделений корпорации одновременно в нескольких измерениях;

ABC (Activity Based Costing) - процессно-ориентированный метод распределения накладных издержек. Основой данной технологии является то, что накладные издержки включаются в себестоимость продукции по мере их возникновения, а не локализируются после завершения производства или реализации по видам продуктов. Крайне важным процессом является отслеживание физического переноса накладных издержек, при котором внимание акцентируется на источниках возникновения затрат и их обоснованности;

EVA (Economic Valued Added) - система управления, основанная на определении и учете экономической добавленной стоимости. Ориентирована на сравнение экономического эффекта от вложения средств в тот или иной перспективный проект с отдачей от альтернативных вложений.

MES-системы много моложе ERP-систем.

Организация ISA определила стандарты, определяющие структуру MES-приложений и их интеграцию в IT-архитектуру компании, независимо от поставщика MES-системы. Стандарт ISA S95 «Enterprise-Control System Integration» определяет уровни модели, описывающей взаимодействия между ERP, MES и уровнем автоматизации производства. Стандарт поддерживают ведущие поставщики MES-систем. Стандарт ISA S88 «General and Site Recipe Models and Representation» определяет модели для batch (рецептурных) задач в таких отраслях промышленности, как пищевая, фармацевтическая, химическая.

Наряду с приведенным выше определением системы MES приведем следующие: Согласно определению AMR Research — термин MES (Manufacturing Execution System) — это высокоавтоматизированная система контроля производственных процессов и управления ими. По определению APICS (American Production and Inventory Control Society) MES — это информационная и коммуникационная система производственной среды предприятия.

Большинство экспертов считают, что управление производством опирается на решение четырех задач:

- формирование объемно-номенклатурного плана;
- баланс производственных мощностей;
- составление производственных расписаний;
- группирование оборудования и детали-сборочных единиц.

Первые две задачи достаточно эффективно решаются с помощью ERP, а две последние — MES.

Интеграция между этими системами направлена на установление связей между процессами планирования производства, имеющими место и на уровне предприятия, и на уровне производства. Она также предусматривает возможность передачи детальных производственных планов, сводок и отчетов в существующие на предприятии информационные системы.

MIS — (Management Information System) — Система информационного обеспечения управления. Средства MIS позволяют интегрировать информацию, необходимую для принятия управленческих решений.

SCM — (Supply Chain Management) — Управление цепочками поставок. Технологии SCM автоматизируют весь цикл: «поставщик» — транспорт- производство — хранение — распространение — заказчики», что дает возможность предприятию доставлять необходимый товар в нужное место точно в срок и с минимальными издержками.

CRM — (Customer Relationship Management) — Управление взаимоотношениями с клиентами. Технологии CRM автоматизируют все этапы работы с клиентами: от рекламной кампании и первого контакта с потенциальным заказчиком до послепродажной поддержки и гарантийного обслуживания.

SSM — (Sales & Service Management) — Управление сбытом и обслуживанием.

P/PE — (Product and Process Engineering) — Проектирование производственных процессов и продуктов.

e-business — (электронный бизнес) — Средства электронного бизнеса позволяют взаимодействовать с поставщиками и заказчиками через Интернет. Это особо важно в условиях нашей страны, когда многие участники бизнес-операций географически удалены друг от друга.

SCADA — (Supervisory Control and Data Acquisition) — диспетчерское управление и сбор данных.

Под термином SCADA понимают инструментальную программу для разработки программного обеспечения систем управления технологическими процессами в реальном времени (АСУ ТП) и удаленного сбора данных (телемеханика). Реже термин SCADA-система используют для обозначения программно-аппаратного комплекса сбора данных (телемеханического комплекса).

HMI — (Human Machine Interface) — человеко-машинный интерфейс. Отображение информации на экране монитора в понятной для человека форме.

SCADA-системы позволяют разрабатывать АСУ ТП в клиент-серверной или в распределенной архитектуре (DCS — Distributed Control System — распределенная система управления).

Термин SCADA эволюционировал вместе с развитием технологий автоматизации и управления технологическими процессами. В 80-е годы под SCADA-системами понимали

любые программно-аппаратные комплексы сбора данных реального времени. В 90-х годах термин SCADA больше используется для обозначения только программной части АСУ ТП. В некоторых странах синонимом SCADA является HMI, хотя данный термин сужает функциональные возможности SCADA-систем.

В России больше распространены SCADA-системы отечественного производства.

BI - (Business Intelligence) — Бизнес Аналитика — Под этим понятием чаще всего подразумевают программное обеспечение, созданное для помощи менеджеру в анализе информации о своей компании и её окружении. Это ПО, специально разработанное, чтобы помочь понять, что является движущей силой предприятия, и предсказать влияние текущих решений на будущее. BI включает аналитику, сбор информации для принятия решений и аналитические технологии. Современные BI-системы способны обеспечивать поиск в базах данных, используя технологии наподобие нейронных сетей и деревьев решений, с поиском сильных связей и примеров, которые человеку практически невозможно обнаружить, и преподносят их руководству, чтобы они могли выбрать верный курс.

OLAP (от англ. OnLineAnalyticalProcessing — оперативная аналитическая обработка данных, также: аналитическая обработка данных в реальном времени, интерактивная аналитическая обработка данных) — подход к аналитической обработке данных, базирующийся на их многомерном иерархическом представлении, являющийся частью более широкой области информационных технологий — бизнес-аналитики (BI).

OLAP-системы редко используются как средство непосредственного хранения и модификации данных (за исключением некоторых простых и маломасштабных систем бюджетирования, учета и анализа продаж и т. п.), так как большинство данных, используемых в OLAP для анализа, генерируются в других информационных системах (ERP, CRM, HRM и т. д.).

При этом, с одной стороны, специфичные для OLAP-систем требования к данным обычно подразумевают хранение данных в специальных оптимизированных под типовые задачи OLAP структурах, с другой стороны, непосредственное извлечение данных из существующих систем в процессе анализа привело бы к существенному падению их производительности.

Следовательно, важным требованием является обеспечение максимально гибкой связки импорта-экспорта между существующими системами, выступающими в качестве источника данных и OLAP-системой, а также OLAP-системой и внешними приложениями анализа данных и отчетности.

При этом такая связка должна удовлетворять очевидным требованиям поддержки импорта-экспорта из нескольких источников данных, осуществления процедур очистки и трансформации данных, унификации используемых классификаторов и справочников. Кроме того, к этим требованиям добавляется необходимость учёта различных циклов обновления данных в существующих информационных системах и унификации требуемого уровня детализации данных. Сложность и многогранность этой проблемы привела к появлению концепции хранилищ данных, и, в узком смысле, к выделению отдельного класса утилит конвертации и преобразования данных — ETL (ExtractTransformLoad).

Хранение активных данных в многомерной БД. В этом случае данные OLAP хранятся в многомерных СУБД, использующих оптимизированные для такого типа

данных конструкции. Обычно многомерные СУБД поддерживают и все типовые для OLAP операции, включая агрегацию по требуемым уровням иерархии и так далее. Этот тип хранения данных в каком-то смысле можно назвать классическим для OLAP. Для него, впрочем, в полной мере необходимы все шаги по предварительной подготовке данных. Обычно данные многомерной СУБД хранятся на диске, однако, в некоторых случаях, для ускорения обработки данных такие системы позволяют хранить данные в оперативной памяти. Для тех же целей иногда применяется и хранение в БД заранее рассчитанных агрегатных значений и прочих расчётных величин. Многомерные СУБД, полностью поддерживающие многопользовательский доступ с конкурирующими транзакциями чтения и записи достаточно редки, обычным режимом для таких СУБД является однопользовательский с доступом на запись при многопользовательском на чтение, либо многопользовательский только на чтение. Среди условных недостатков, характерных для некоторых реализаций многомерных СУБД и базирующихся на них OLAP-систем можно отметить их подверженность непредсказуемому с пользовательской точки зрения росту объёмов занимаемого БД места. Этот эффект вызван желанием максимально уменьшить время реакции системы, диктующим хранить заранее рассчитанные значения агрегатных показателей и иных величин в БД, что вызывает нелинейный рост объёма хранящейся в БД информации с добавлением в неё новых значений данных или измерений. Степень проявления этой проблемы, а также связанных с ней проблем эффективного хранения разреженных кубов данных, определяется качеством применяемых подходов и алгоритмов конкретных реализаций OLAP-систем.

Хранение активных данных в реляционной БД. Могут храниться данные OLAP и в традиционной РСУБД. В большинстве случаев этот подход используется при попытке «безболезненной» интеграции OLAP с существующими учётными системами, либо базирующимися на РСУБД хранилищами данных. Вместе с тем, этот подход требует от РСУБД для обеспечения эффективного выполнения требований FASMI-теста (в частности, обеспечения минимального времени реакции системы) некоторых дополнительных возможностей. Обычно данные OLAP хранятся в денормализованном виде, а часть заранее рассчитанных агрегатов и значений хранится в специальных таблицах. При хранении же в нормализованном виде эффективность РСУБД в качестве метода хранения активных данных снижается. Проблема выбора эффективных подходов и алгоритмов хранения предрасчитанных данных также актуальна для OLAP-систем, базирующихся на РСУБД, поэтому производители таких систем обычно акцентируют внимание на достоинствах применяемых подходов. В целом считается, что базирующиеся на РСУБД OLAP-системы медленнее систем, базирующихся на многомерных СУБД, в том числе за счет менее эффективных для задач OLAP структур хранения данных, однако на практике это зависит от особенностей конкретной системы. Среди достоинств хранения данных в РСУБД обычно называют большую масштабируемость таких систем.

Хранение активных данных в «плоских» файлах. Этот подход предполагает хранение порций данных в обычных файлах. Обычно он используется как дополнение к одному из двух основных подходов с целью ускорения работы за счет кэширования актуальных данных на диске или в оперативной памяти клиентского ПК.

Гибридный подход к хранению данных. Большинство производителей OLAP-систем, продвигающих свои комплексные решения, часто включающие помимо собственно OLAP-системы СУБД, инструменты ETL (Extract Transform Load) и отчетности, в настоящее время используют гибридный подход к организации хранения активных данных системы, распределяя их тем или иным образом между РСУБД и специализированным хранилищем, а также между дисковыми структурами и кэшированием в оперативной памяти. Так как эффективность такого решения зависит от конкретных подходов и алгоритмов, применяемых производителем для определения того, какие данные и где хранить, то поспешно делать выводы о изначально большей эффективности таких решений как класса без оценки конкретных особенностей рассматриваемой системы.

На сегодняшний день, к *мобильным информационным технологиям* можно отнести:

- GSM и UMTS – это стандарты связи,
- WAP – протокол, по которому можно получить доступ в Интернет с мобильного телефона,
- GPRS и EDGE – технологии передачи данных,
- Wi-Fi – мобильные беспроводные сети Интернет,
- GPS – спутниковая система навигации,
- WiMAX – телекоммуникационная технология мобильной связи, работающая по принципу Wi-Fi и позволяющая получить доступ в Интернет
- новое поколение связи – сеть 5G.

GSM (от названия группы Groupe Special Mobile, позже переименован в Global System for Mobile Communications) (русс.СПС-900) — глобальный стандарт цифровой мобильной сотовой связи, с разделением каналов по времени (TDMA) и частоте (FDMA). Разработан под эгидой Европейского института стандартизации электросвязи (ETSI) в конце 1980-х годов.

GSM относится к сетям второго поколения (2 Generation) (1G — аналоговая сотовая связь, 2G — цифровая сотовая связь, 3G — широкополосная цифровая сотовая связь, коммутируемая многоцелевыми компьютерными сетями, в том числе Интернет).

Мобильные телефоны выпускаются с поддержкой 4 частот: 850 МГц, 900 МГц, 1800 МГц, 1900 МГц.

В зависимости от количества диапазонов, телефоны подразделяются на классы и вариацию частот в зависимости от региона использования.

- Однодиапазонные — телефон может работать в одной полосе частот. В настоящее время не выпускаются, но существует возможность ручного выбора определённого диапазона частот в некоторых моделях телефонов, например Motorola C115, или с помощью инженерного меню телефона.

- Двухдиапазонные (Dual Band) — для Европы, Азии, Африки, Австралии 900/1800 и 850/1900 для Америки и Канады.

- Трёхдиапазонные (Tri Band) — для Европы, Азии, Африки, Австралии 900/1800/1900 и 850/1800/1900 для Америки и Канады.

- Четырёхдиапазонные (Quad Band) — поддерживают все диапазоны 850/900/1800/1900.

В стандарте GSM применяется GMSK-модуляция с величиной нормированной полосы $BT = 0,3$, где B — ширина полосы фильтра по уровню минус 3 дБ, T — длительность одного бита цифрового сообщения.

UMTS (англ. Universal Mobile Telecommunications System — Универсальная Мобильная Телекоммуникационная Система) — технология сотовой связи, разработана Европейским Институтом Стандартов Телекоммуникаций (ETSI) для внедрения 3G в Европе. В качестве способа передачи данных через воздушное пространство используется технология W-CDMA, стандартизованная в соответствии с проектом 3GPP ответ европейских учёных и производителей на требование IMT-2000, опубликованное Международным союзом электросвязи как набор минимальных критериев сети сотовой связи третьего поколения.

С целью отличия от конкурирующих решений UMTS также часто называют 3GSM с целью подчеркнуть принадлежность технологии к сетям 3G и его преимущество в разработках с сетями стандарта GSM.

Wireless Application Protocol (WAP) (англ. Wireless Application Protocol — беспроводной протокол передачи данных). Протокол создан специально для сетей GSM, где нужно устанавливать связь портативных устройств (мобильный телефон, КПК, пейджеры, устройства двусторонней радиосвязи, смартфоны, и другие терминалы) с сетью Интернет. WAP возник в результате слияния двух сетевых технологий: беспроводной цифровой передачи данных и сети Интернет. С помощью WAP пользователь мобильного устройства может загружать из сети Интернет любые цифровые данные. Параллельно с WAP, для возможности отображать мобильный контент на монохромных (а позже и четырёх- и восьмицветовых) экранах мобильных устройств, был создан WML по стилю написания похожий на HTML, но гораздо более облегчённый и специализированный для мобильных устройств с низким уровнем поддерживаемых технологий.

GPRS (англ. General Packet Radio Service — «пакетная радиосвязь общего пользования») — надстройка над технологией мобильной связи GSM, осуществляющая пакетную передачу данных. GPRS позволяет пользователю сети сотовой связи производить обмен данными с другими устройствами в сети GSM и с внешними сетями, в том числе Интернет. GPRS предполагает тарификацию по объёму переданной/полученной информации, а не по времени, проведённому онлайн.

При использовании GPRS информация собирается в пакеты и передаётся через неиспользуемые в данный момент голосовые каналы. Такая технология предполагает более эффективное использование ресурсов сети GSM. При этом, что именно является приоритетом передачи — голосовой трафик или передача данных — выбирается оператором связи. Федеральная тройка в России использует безусловный приоритет голосового трафика перед данными, поэтому скорость передачи зависит не только от возможностей оборудования, но и от загрузки сети. Возможность использования сразу нескольких каналов обеспечивает достаточно высокие скорости передачи данных, теоретический максимум при всех занятых таймслотах TDMA составляет 171,2 кбит/с. Существуют различные классы GPRS, различающиеся скоростью передачи данных и возможностью совмещения передачи данных с одновременным голосовым вызовом.

EDGE (EGPRS) (англ. Enhanced Data rates for GSM Evolution) — цифровая технология беспроводной передачи данных для мобильной связи, которая функционирует как надстройка над 2G и 2.5G (GPRS)-сетями. Эта технология работает в TDMA- и GSM-сетях. Для поддержки EDGE в сети GSM требуются определённые модификации и усовершенствования. EDGE был впервые представлен в 2003 году в Северной Америке.

EDGE обеспечивает передачу данных со скоростью до 474 кбит/с в режиме пакетной коммутации (8 тайм-слотов $\times$ 59,2 кбит на схеме кодирования MCS-9) соответствуя, таким образом, требованиям ITU к сетям 3G. Данная технология была принята ITU как часть семейства IMT-2000 стандартов 3G. Она также расширяет технологию передачи данных с коммутацией каналов HSCSD, увеличивая пропускную способность этого сервиса.

Wi-Fi — торговая марка Wi-Fi Alliance для беспроводных сетей на базе стандарта IEEE 802.11. Под аббревиатурой Wi-Fi (от английского словосочетания Wireless Fidelity, которое можно дословно перевести как «беспроводное качество» или «беспроводная точность») в настоящее время развивается целое семейство стандартов передачи цифровых потоков данных по радиоканалам.

Любое оборудование, соответствующее стандарту IEEE 802.11, может быть протестировано в Wi-Fi Alliance и получить соответствующий сертификат и право нанесения логотипа Wi-Fi.

Обычно схема Wi-Fi сети содержит не менее одной точки доступа и не менее одного клиента. Также возможно подключение двух клиентов в режиме точка-точка (Ad-hoc), когда точка доступа не используется, а клиенты соединяются посредством сетевых адаптеров «напрямую». Точка доступа передаёт свой идентификатор сети (SSID (англ.)русск.) с помощью специальных сигнальных пакетов на скорости 0,1 Мбит/с каждые 100 мс. Поэтому 0,1 Мбит/с — наименьшая скорость передачи данных для Wi-Fi. Зная SSID сети, клиент может выяснить, возможно ли подключение к данной точке доступа. При попадании в зону действия двух точек доступа с идентичными SSID приёмник может выбирать между ними на основании данных об уровне сигнала. Стандарт Wi-Fi даёт клиенту полную свободу при выборе критериев для соединения. Более подробно принцип работы описан в официальном тексте стандарта.

GPS (англ. Global Positioning System — система глобального позиционирования, читается Джи Пи Эс) — спутниковая система навигации, обеспечивающая измерение расстояния, времени и определяющая местоположение во всемирной системе координат WGS 84. Позволяет в любом месте Земли (исключая приполярные области), почти при любой погоде, а также в околоземном космическом пространстве определять местоположение и скорость объектов. Система разработана, реализована и эксплуатируется Министерством обороны США, при этом в настоящее время доступна для использования для гражданских целей — нужен только навигатор или другой аппарат (например, смартфон) с GPS-приёмником.

GPS состоит из трёх основных сегментов: космического, управляющего и пользовательского. Спутники GPS транслируют сигнал из космоса, и все приёмники GPS используют этот сигнал для вычисления своего положения в пространстве по трём координатам в режиме реального времени.

Космический сегмент состоит из 32 спутников, вращающихся на средней орбите Земли.

Управляющий сегмент представляет собой главную управляющую станцию и несколько дополнительных станций, а также наземные антенны и станции мониторинга, ресурсы некоторых из упомянутых являются общими с другими проектами.

Пользовательский сегмент представлен приемниками GPS, находящихся в ведении государственных институтов, и сотнями миллионов устройств, владельцами которых являются обычные пользователи.

WiMAX (англ. Worldwide Interoperability for Microwave Access) — телекоммуникационная технология, разработанная с целью предоставления универсальной беспроводной связи на больших расстояниях для широкого спектра устройств (от рабочих станций и портативных компьютеров до мобильных телефонов). Основана на стандарте IEEE 802.16, который также называют Wireless MAN (WiMAX следует считать жаргонным названием, так как это не технология, а название форума, на котором Wireless MAN и был согласован).

Название «WiMAX» было создано WiMAX Forum — организацией, которая была основана в июне 2001 года с целью продвижения и развития технологии WiMAX. Форум описывает WiMAX как «основанную на стандарте технологию, предоставляющую высокоскоростной беспроводной доступ к сети, альтернативный выделенным линиям и DSL». Максимальная скорость — до 1 Гбит/сек на ячейку.

4G (от англ. fourth generation — четвертое поколение) — поколение мобильной связи с повышенными требованиями. К четвертому поколению принято относить перспективные технологии, позволяющие осуществлять передачу данных со скоростью, превышающей 100 Мбит/с - подвижным и 1 Гбит/с - стационарным абонентам.

Технологии LTE Advanced (LTE-A) и WiMAX 2 (WMAN-Advanced, IEEE 802.16m) были официально признаны беспроводными стандартами связи четвертого поколения 4G (IMT-Advanced) Международным союзом электросвязи на конференции в Женеве в 2012 году.

Мобильные устройства:

- Смартфоны
- Планшеты
- Ноутбуки
- Умные часы и прочее

Смартфон (англ. smartphone — умный телефон) — телефон, дополненный функциональностью персонального компьютера.

Хотя в мобильных телефонах практически всегда были дополнительные функции (калькулятор, календарь), со временем выпускались все более и более интеллектуальные модели, для подчеркивания возросшего функционала и вычислительной мощности таких моделей ввели термин «смартфон». В эру роста популярности КПК стали выпускаться КПК с функциями мобильного телефона, такие устройства были названы коммуникаторами. В настоящее время разделение на смартфоны и коммуникаторы не актуально, оба термина обозначают одно и то же.

Смартфоны отличаются от обычных мобильных телефонов наличием достаточно развитой операционной системы, открытой для разработки программного обеспечения сторонними разработчиками (операционная система обычных мобильных телефонов закрыта для сторонних разработчиков). Установка дополнительных приложений

позволяет значительно улучшить функциональность смартфонов по сравнению с обычными мобильными телефонами.

Мировая статистика использования смартфонов показывает абсолютное преобладание всего двух мобильных операционных систем. Так, по данным портала statista.com, во втором квартале 2018 OS Android была установлена на 88% всех используемых смартфонов, а iOS – на 11.9%. Данные портала netmarketshare.com, в свою очередь, показывают на апрель 2019 для OS Android – 69.63%, а для iOS - 28.50%.

По состоянию на март 2019 в Google Play Store находилось более 2 600 000 приложений. В Apple App Store, по данным на июль 2018 – 2 450 220 приложений. В течение последних лет количество доступных приложений растет на сотни тысяч в год. По прогнозам [statista](http://statista.com) – к 2020 году объем рынка мобильных приложений приблизится к 190 млрд \$. При таком, постоянно растущем количестве конкурентов, перед разработчиками мобильных приложений встает вопрос - какой подход использовать в разработке, чтобы новые, конкурентоспособные приложения:

- разрабатывались быстро;
- получались качественными и надежными;
- легко обновлялись и поддерживались;
- легко задействовали все необходимые возможности платформы.

Фактически, рынок заставляет разработчика делать выбор между разработкой кроссплатформенных приложений и разработкой нативных приложений. Рассмотрим подробнее, что представляет из себя каждый из указанных подходов.

Нативные приложения (от англ. native - родной) разрабатываются под конкретную аппаратно-программную платформу и пишутся на языках, созданных для данной платформы. И iOS, и Android имеют свои SDK (от англ. software development kit — набор средств разработки) и свой стек технологий, завязанные на определенный язык программирования. Например, родными языками для Android являются Java и Kotlin, для iOS, соответственно - Swift и Objective-C.

Нативные приложения создаются специально для запуска на целевой платформе - с поддержкой всех нативных технологий и аппаратных возможностей конкретной платформы.

Как следует из названия, *кроссплатформенность* подразумевает создание приложений, которые могут работать в различных операционных системах. После написания кода приложения его можно развернуть на разных устройствах и платформах, не беспокоясь о проблемах несовместимости. Это универсальный подход, который широко используется для экономии времени и денег на разработку. Часто для этого используются специализированные кроссплатформенные фреймворки.

Примером такой разработки является применение фреймворка Xamarin для создания приложений, работающих не только на Windows. Благодаря использованию Mono (открытая реализация платформы .Net), проекты, написанные на C#, успешно запускаются на Unix-like системах – iOS, Android, Linux.

Наиболее часто применяемым инструментом разработки кроссплатформенных мобильных приложений на конец 2017 года был Cordova – 39.89%. Вторым по частоте применения инструментом является Unity – 30.93%. Третьим – Adobe Flash с 10.39%. Следом идут Cocos2D – 9.37%, Xamarin – 4.5%, Appcelerator – 3.79%, Corona – 2.68%, React Native – 1.85%.

По данным портала appfigures.com на начало 2018 года количество приложений, присутствующих на обеих популярных платформах, было вполне ощутимым: 450 тысяч приложений на обеих платформах. Это более 28% приложений в Apple App store и 14% в Google Play Store. Это выглядит достаточно весомой частью, чтобы задуматься об присутствии на обеих платформах и попытке экономии используя кроссплатформенную разработку.

По данным того же портала, многие уже существующие приложения расширяют свой рынок, выходя, со временем, на другой платформе. При чем чаще приложения выходят дополнительно на Android, выпускаясь изначально под iOS.

Плюсы и минусы кроссплатформенных приложений.

Как следует из названия, кроссплатформенность влечет за собой создание приложений, которые могут работать в различных операционных системах. После написания кода приложения его можно развернуть на разных устройствах и платформах, не беспокоясь о проблемах несовместимости. Это универсальный подход, который широко используется для экономии времени и денег.

Вот некоторые преимущества и недостатки использования кроссплатформенного подхода в разработке мобильных приложений.

Плюсы кроссплатформенных приложений.

Один код доступен для повторного использования на других платформах

Основным преимуществом кроссплатформенной разработки мобильных приложений является тот факт, что один и тот же код может использоваться на разных мобильных платформах. В отличие от разработки нативного приложения, для кроссплатформенного приложения не требуется использование отдельного технического стека для каждой операционной системы.

Повторное использование кода позволяет легко развертывать приложение на другой платформе, так как возможности приложения, реализованные на одной платформе, будут работать и на других платформах.

Разработка кроссплатформенных приложений экономически эффективна

Одна команда может реализовать нужную идею сразу на всех платформах, используя единый технологический стек. Это приводит к меньшим затратам ресурсов.

Простое и быстрое развертывание

Разработчикам кроссплатформенных приложений не нужно изучать несколько технологических стеков различных платформ перед созданием своих приложений, им нужно хорошо освоить один стек разработки и особенности его применения.

Поскольку нет необходимости создавать разные кодовые базы, начальное развертывание на целевых платформах происходит намного быстрее.

Кроме того, будущие изменения в приложении могут выполняться одновременно, без внесения индивидуальных изменений на каждой платформе.

Кроссплатформенные приложения покрывают более широкую аудиторию

Кроссплатформенные приложения предлагают разработчикам больше возможностей для охвата более широкой аудитории, поскольку такие приложения достигают пользователей всех типов и мобильных устройств, независимо от их операционной системы. Это значительно рентабельнее для бизнеса, чем присутствие только на одной платформе.

Кроссплатформенные приложения допускают одинаковый интерфейс и UX. Тогда как производительность важна для любого мобильного приложения, его внешний вид (UI) и ощущения (UX) так же важны. Использование единой общей команды разработчиков и единого кода позволяет компаниям использовать одинаковый внешний вид приложения на всех платформах. То есть один и тот же пользовательский интерфейс и UX будет одинаково выглядеть на всех платформах.

Недостатки кроссплатформенной разработки приложений.

Кроссплатформенные приложения не являются такими гибкими, как нативные приложения.

Хотя задачи приложения будут реализовываться на всех платформах, скорее всего вы не сможете адаптировать готовое приложение для использования максимальных возможностей каждой из платформ. Работа с унифицированным стеком технологий не обеспечит такой же гибкости настройки и оптимизации, как применение стека технологий, индивидуального для каждой ОС.

Кроссплатформенные приложения не работают так же хорошо, как нативные приложения.

Использование одного универсального стека технологий приносит в жертву гибкость. Однако потеря гибкости в разработке будет означать потерю возможности улучшить производительность. Поскольку кроссплатформенные приложения отказываются от некоторой гибкости, эти приложения не будут работать так же хорошо, как нативные приложения.

Возможное несоответствие UI в различных платформах.

Внешний вид интерфейса приложения и правильная настройка UI для соответствия функционала в обеих системах может доставить проблем. К примеру, у каждой системы имеются свои требования к дизайну элементов UI. В определенных случаях эти требования могут оказаться взаимоисключающими.

Отправка кроссплатформенных приложений в соответствующие Магазины приложений может иметь сложности.

Механизм добавления приложения, являющегося кроссплатформенным, в Apple App Store и в Google Play Store будет отличаться. Требования этих магазинов приложений к представленным у них продуктам различны. Прохождение всех проверок и выполнение всех правил для соответствия обоим магазинам будут вызывать определенные сложности.

В мобильных облачных вычислениях используются облачные вычисления для доставки приложений на мобильные устройства. Эти мобильные приложения могут удаленно развертываться с помощью быстрых и гибких инструментов разработки. В мобильном облаке приложения можно быстро разрабатывать и перерабатывать с помощью облачных служб. Они могут доставляться на множество различных устройств с разными операционными системами, вычислительными задачами и хранилищами данных. Таким образом, пользователи получают доступ к приложениям, которые в противном случае не поддерживались бы.

Основные функции.

Упрощение быстрой разработки и совместного использования ресурсов мобильных приложений.

Поддержка различных устройств и подходов к разработке.

Повышение надежности благодаря резервному копированию и хранению информации в облаке.

Приложения используют меньше ресурсов, поскольку поддерживаются облаком.

Мобильные приложения подключаются к службам, предоставляемым архитектурой API.

Вопросы для самоконтроля:

1. Информационно-управляющие системы.
2. OLAP-технологии.
3. Мобильные технологии.
4. Кросс-платформенность в мобильных технологиях.

Литература: [1]; [2]; [4]; [9]; [12]; [15]; [16]; [17]; [18]; [19]; [20]; [21]; [22].

Тема № 7. Проблемы больших данных и способы их решения

Материалы лекций

Получение полезной информации из больших данных основано на их анализе. Существует множество разнообразных методик анализа массивов данных, в основе которых лежит инструментарий, заимствованный из статистики и информатики (например, машинное обучение). Чем более объемный и диверсифицируемый массив подвергается анализу, тем более точные и релевантные данные удастся получить на выходе.

Основные методики анализа больших данных.

A/B testing. Методика, в которой контрольная выборка поочередно сравнивается с другими. Тем самым удастся выявить оптимальную комбинацию показателей для достижения, например, наилучшей ответной реакции потребителей на маркетинговое предложение. Большие данные позволяют провести огромное количество итераций и таким образом получить статистически достоверный результат.

Association rule learning. Набор методик для выявления взаимосвязей, т.е. ассоциативных правил, между переменными величинами в больших массивах данных. Используется в data mining.

Classification. Набор методик, которые позволяют предсказать поведение потребителей в определенном сегменте рынка (принятие решений о покупке, отток, объем потребления и проч.). Используется в data mining.

Cluster analysis. Статистический метод классификации объектов по группам за счет выявления наперед не известных общих признаков. Используется в data mining.

Crowdsourcing. Методика сбора данных из большого количества источников.

Data fusion and data integration. Набор методик, который позволяет анализировать комментарии пользователей социальных сетей и сопоставлять с результатами продаж в режиме реального времени.

Data mining. Набор методик, который позволяет определить наиболее восприимчивые для продвигаемого продукта или услуги категории потребителей,

выявить особенности наиболее успешных работников, предсказать поведенческую модель потребителей.

Ensemble learning. В этом методе задействуется множество предикативных моделей за счет чего повышается качество сделанных прогнозов.

Genetic algorithms. В этой методике возможные решения представляют в виде `хромосом`, которые могут комбинироваться и мутировать. Как и в процессе естественной эволюции, выживает наиболее приспособленная особь.

Machine learning. Направление в информатике (исторически за ним закрепилось название `искусственный интеллект`), которое преследует цель создания алгоритмов самообучения на основе анализа эмпирических данных.

Natural language processing (NLP). Набор заимствованных из информатики и лингвистики методик распознавания естественного языка человека.

Network analysis. Набор методик анализа связей между узлами в сетях. Применительно к социальным сетям позволяет анализировать взаимосвязи между отдельными пользователями, компаниями, сообществами и т.п.

Optimization. Набор численных методов для редизайна сложных систем и процессов для улучшения одного или нескольких показателей. Помогает в принятии стратегических решений, например, состава выводимой на рынок продуктовой линейки, проведении инвестиционного анализа и проч.

Pattern recognition. Набор методик с элементами самообучения для предсказания поведенческой модели потребителей.

Predictive modeling. Набор методик, которые позволяют создать математическую модель наперед заданного вероятного сценария развития событий. Например, анализ базы данных CRM-системы на предмет возможных условий, которые подтолкнут абоненты сменить провайдера.

Regression. Набор статистических методов для выявления закономерности между изменением зависимой переменной и одной или несколькими независимыми. Часто применяется для прогнозирования и предсказаний. Используется в data mining.

Sentiment analysis. В основе методик оценки настроений потребителей лежат технологии распознавания естественного языка человека. Они позволяют вычленив из общего информационного потока сообщения, связанные с интересующим предметом (например, потребительским продуктом). Далее оценить полярность суждения (позитивное или негативное), степень эмоциональности и проч.

Signal processing. Заимствованный из радиотехники набор методик, который преследует цель распознавания сигнала на фоне шума и его дальнейшего анализа.

Spatial analysis. Набор отчасти заимствованных из статистики методик анализа пространственных данных – топологии местности, географических координат, геометрии объектов. Источником больших данных в этом случае часто выступают геоинформационные системы (ГИС).

Statistics. Наука о сборе, организации и интерпретации данных, включая разработку опросников и проведение экспериментов. Статистические методы часто применяются для оценочных суждений о взаимосвязях между теми или иными событиями.

Supervised learning. Набор основанных на технологиях машинного обучения методик, которые позволяют выявить функциональные взаимосвязи в анализируемых массивах данных.

Simulation. Моделирование поведения сложных систем часто используется для прогнозирования, предсказания и проработки различных сценариев при планировании.

Time series analysis. Набор заимствованных из статистики и цифровой обработки сигналов методов анализа повторяющихся с течением времени последовательностей данных. Одни из очевидных применений – отслеживание рынка ценных бумаг или заболеваемости пациентов.

Unsupervised learning. Набор основанных на технологиях машинного обучения методик, которые позволяют выявить скрытые функциональные взаимосвязи в анализируемых массивах данных. Имеет общие черты с Cluster Analysis.

В 2004 году Google опубликовал статью, посвященную процессу MapReduce. Каркас MapReduce использует модель *параллельной обработки* очень большого объема данных. В рамках MapReduce запросы разделяются и распределяются по параллельным узлам, при этом они обрабатываются параллельно (этап распределения (Map)). Затем осуществляется сбор и доставка результатов (этап предварительного преобразования (Reduce)). Этот каркас был весьма эффективен, поэтому другие компании также захотели скопировать алгоритм. Каркас MapReduce был использован в проекте Apache открытым исходным кодом, носящем название Hadoop. Последние исследования показали, что многоуровневая архитектура может успешно использоваться для анализа больших данных. Архитектура распределенной параллельной обработки распределяет данные по многочисленным узлам, которые, производя, соответственно, параллельную обработку, генерируют результат гораздо быстрее, чем другие системы. В рамках такой архитектуры данные передаются в параллельные СУБД с использованием каркасов MapReduce и Hadoop. При этом задействуются фронтальные серверы приложений.

Основная идея технологии *Облачных вычислений*. Облачных сервисов заключается в получении через интернет полностью готового к использованию сервиса в нужном объеме в каждый конкретный период времени.

Примеры массово используемых облачных сервисов:

- Электронная почта (публичные сервисы с веб-доступом),
- Виртуальная АТС,
- Видеоконференцсвязь (для проведения вебинаров, видеопрезентаций, видеосовещаний),
- Управление проектами/рабочими группами,
- Управление взаимоотношениями с клиентами -CRM (часто интегрируется с функциями Виртуальной АТС и электронной почты),
- Виртуальный офис (набор сервисов по созданию/обработке документов и электронных таблиц в сочетании с функциями документооборота, электронной почты, виртуальной АТС),
- Складской и бухгалтерский учет,
- Хостинг (виртуальные серверные ресурсы для запуска произвольных приложений, например, хостинг сайтов).

Облачная модель предоставления сервисов быстро и активно развивается в последние годы и большинство функций по обработке, управлению и хранению информации уже сейчас доступны в форме облачных сервисов и число их продолжает бурно расти.

Обычно выделяют следующие модели предоставления Облачных сервисов:

- IaaS - Infrastructure as a Service. Инфраструктура как услуга (IaaS - Infrastructure as a Service) – предоставление вычислительных мощностей в форме виртуальной инфраструктуры с заданной конфигурацией, требуемого объема и оснащенной системным программным обеспечением (например: хостинг).

- PaaS - Platform as a Service. Платформа как услуга (PaaS - Platform as a Service) – предоставление платформы для разработки, тестирования, развертывания и поддержки приложений как услуги (пример: Windows Azure от Microsoft).

- SaaS – Software as a Service. Программное обеспечение как услуга (SaaS – Software as a Service) – предоставление программного обеспечения как услуги (например: Виртуальный офис Office 365).

- DaaS (Desktop-as-a-Service) переводится как «рабочий стол как услуга». Модель DaaS — это логическое продолжение SaaS. Здесь сервисом является не определенное программное обеспечение, а рабочее место, которое готово к использованию и снабжено всеми необходимыми средствами.

Независимо от типа сервиса потребитель всегда приобретает нужные ему функции в готовом виде, как услуги с требуемым объемом и на заданное время. Единственная забота пользователя – наличие интернет-канала достаточной пропускной способности и надежности.

Основные выгоды потребителя облачного сервиса:

- Экономия времени и средств на создании собственной инфраструктуры (на оборудовании, программном обеспечении, помещении для размещения),

- Экономия средств на приобретении и управлении лицензиями программного обеспечения,

- Экономия средств на обслуживании оборудования и программного обеспечения,

- Отсутствие убытков от простоев (например: потери от не полной средней загрузки приобретенных вычислительных мощностей, сезонных простоев, от использования лишь небольшой доли функциональных возможностей приобретённого в собственность программного обеспечения или от частичного использования находящихся в собственности лицензий),

- Отсутствие рисков потери инвестиций в оборудование и программное обеспечение для случаев репрофилирования бизнеса,

- Масштабирование потребляемых облачных мощностей от минимально необходимых до, практически, неограниченных по мере изменения текущих потребностей,

- Сравнительно низкая стоимость облачных сервисов в расчете на 1-го пользователя в сочетании с возможностью ежемесячной/поквартальной оплаты,

- Оплата за фактическое потребление (только за то, что нужно и когда нужно).

Основные выгоды поставщика облачного сервиса:

- Унифицированный набор сервисов упрощает задачи текущего обслуживания, устранения выявляемых недостатков, развития и совершенствования сервисов,

- Массовое предоставление сервисов (в идеале пользователям из различных часовых поясов) позволяет добиться высокого процента утилизации мощностей дата центра, на базе которого развернуты облачные сервисы,

- Полностью автоматизированные процедуры регистрации пользователей, получения доступа к услугам и их оплаты, позволяют радикально минимизировать штат собственных сотрудников, занимающихся индивидуальным взаимодействием с клиентами.

Все это позволяет минимизировать себестоимость облачных услуг, сделать их очень привлекательными по стоимости для конечных потребителей и обеспечить хорошую доходность и устойчивость

Дополнительные технологические преимущества и экономические выгоды можно получать при интеграции сервисов частных и публичных облаков (Гибридные облака), а также при грамотном их сочетании с традиционными моделями применения информационных технологий.

Вопросы для самоконтроля:

1. Работа с большими данными.
2. Параллельная обработка данных.
3. Облачные вычисления и их модели.

Литература: [1]; [9]; [14]; [15]; [16]; [21].

Тема № 8. Принципы проведения вычислительного эксперимента

Материалы лекций

Исходным пунктом исследования (проектирования) системы (объекта, явления, процесса) с помощью математического моделирования является цель исследования системы. В соответствии с этой целью производится постановка задачи моделирования и план проведения вычислительного эксперимента. Данный метод позволяет решать различные задачи, постановка которых тесно связана с выбором и разработкой математической модели. Строится содержательная модель (расчётная схема) в рамках соответствующей области науки на основе словесно- смыслового описания, которая может содержать и некоторые предположения. Во многих областях для описания системы разработаны специальные приёмы, методы и символы наглядного представления. На этом этапе завершается идеализация системы, задаются границы, формулируются допущения моделирования, определяются главные и второстепенные факторы и параметры системы, оговариваются рамки применимости моделей. Указываются учитываемые свойства системы, и оценка их влияния на её исследуемые характеристики. Формулируются законы (принципы, аналогии), которым подчиняется система. При этом часто одними и теми же моделями могут описываться разные по природе объекты, подчиняющиеся разным фундаментальным законам, и наоборот, данному закону могут отвечать принципиально разные модели (например, линейные и нелинейные). Задаются начальные условия (состояние). На этом этапе формулирования

задачи записываются основные уравнения системы, начальные и граничные условия. Кроме основных уравнений (соотношений), как правило выражающих общие законы сохранения; часто необходимо бывает записать дополнительные соотношения, выражающие общие свойства сред, определяющие коэффициенты основных уравнений (диффузии, электропроводности, вязкости, теплоемкости и т. д.). К ним относятся уравнения состояния. Все соотношения являются функциями состояния среды, характеристики которой необходимо знать с достаточной точностью, что иногда заставляет проводить дополнительные вычислительные эксперименты для их определения. Корректность и полнота учёта существенных с точки зрения поставленной задачи свойств системы является необходимым условием получения адекватных моделей. Может быть построена не одна, а несколько вложенных или взаимодополняющих моделей с разной степенью адекватности (или других свойств), что позволяет говорить об иерархии моделей, упорядоченных по сложности и полноте. Часто строятся упрощённые или оценочные модели. При исследовании систем часто требуется совокупность моделей отражающих различные функциональные аспекты и иерархические уровни поведения, состояния или структуры системы. Определяется оптимальная стратегия и план эксперимента - процедура выбора последовательности расчета и значений параметров для получения требуемых результатов.

Следующим этапом будет собственно разработка математической модели (ММ) системы. На основе содержательной модели формируется математическая задача. На этом этапе решаются важные задачи выбора типа модели и ее конкретного вида в соответствии с теми требованиями и критериями, которые предъявляются к ММ, исходя из постановки проблемы. Здесь используются свои методы, позволяющие предварительно исследовать ММ. К ним относятся также и классические аналитические методы как-то: качественный анализ задачи (исследование особых точек), асимптотические методы, понижение размерности задачи путем осреднения уравнений, метод разделения переменных, линеаризация. Здесь уже ставится (формулируется) чисто математическая задача. Определяется ее корректность, т. е. выясняется характер зависимости от входных данных для того, чтобы определить возможности и метод работы с этой моделью. На этом этапе часто проводят качественные решения задачи, т.е. определяют особые точки, асимптотики и т.п., находят частные решения для специальных случаев, рассматриваются предельные случаи и определяются различные ограничения. Полученные точные решения (иногда построенные на основе упрощённых моделей) при предварительном исследовании ММ часто используются и в дальнейшем для определения качества вычислительных алгоритмов (по близости полученных результатов), построенных для решения полной задачи моделирования.

Необходимо отметить такую важную особенность ММ, как инвариантность, т. е. возможность описания различных систем одной и той же ММ. Это даёт возможность использования универсального программного обеспечения. Для некоторых типовых содержательных моделей существуют банки ММ, прежде всего простых элементов, облегчающие моделирование во многих областях науки и техники.

Часто создаётся множество возможных ММ одного и того же объекта (явления). Причем, эти ММ могут иметь различный тип, например, детерминированный или

вероятностный. А также составляются иерархии моделей по степени полноты или сложности для системы.

Для этого этапа также важно отметить, что часто приходится строить еще и информационную модель, содержащую структуру входной, выходной и промежуточной информации и способы ее преобразований, особенно актуально это для экономичности ММ при большом объеме данных. Необходимо соблюдать соответствие между содержательным и математическим аспектами ММ, а также соответствие между точностью математического решения и степенью приближений и допущений при построении ММ. При построении ММ важно также соблюдать принцип баланса точности, который требует соизмеримости различных типов погрешностей: погрешности модели, погрешности задания исходных данных, погрешности численного метода, погрешности представления (округления) данных.

Четвёртым этапом является разработка или выбор методов или алгоритмов решения сформулированной математической задачи, т. е. вычислительного или моделирующего алгоритма. На этом этапе производится преобразование математической модели в форму, пригодную для последующего счета на ЭВМ, т. е. представляет собой описание последовательности элементарных действий. Фактически получается цепочка алгебраических формул и логических условий, устанавливающих необходимую последовательность действий. Получается сложная иерархическая структура, которая может перестраиваться в процессе расчетов в зависимости от промежуточных результатов. В большинстве случаев для одной и той же математической задачи можно предложить много различных вычислительных алгоритмов (методов), и следовательно, они будут неравнозначны по своим качествам. Поэтому необходимо уметь отличать их, т. е. оценивать их по каким-то критериям. Эти вопросы служат предметом рассмотрения теории численных методов, математической дисциплины, интенсивно развивающейся в настоящее время, имеющей целью - построение и исследование эффективных вычислительных методов (эффективность или экономичность - решение поставленной задачи с заданной точностью за минимальное количество действий). Существует оценка эффективности алгоритма - количество операций умножения в алгоритме. Это особенно важно в связи с "многовариантным" характером вычислительного эксперимента, т. е., как правило, в ходе вычислительного эксперимента проводится большое количество однотипных расчетов, обусловленных большим количеством входных параметров и шириной диапазона их изменения, влияющих на решение задачи, особенно, если это задача оптимизации.

Разработка вычислительного алгоритма, как правило, включает в себя два шага: аппроксимацию исходной системы дифференциальных уравнений системой алгебраических уравнений (построение разностной схемы) и построение метода решения полученных алгебраических уравнений. Т. е. первый шаг представляет собой замену непрерывной среды ее дискретным эквивалентом - разностной сеткой по времени и пространству. Таким образом, производится замена непрерывной области определения задачи дискретным множеством точек (узлов) сетки. При этом необходимо соблюдать ряд условий. Так при дискретизации задачи сплошной среды требуется, чтобы полученная дискретная модель правильно отражала основные свойства сплошной среды. В первую очередь она должна удовлетворять принципу консервативности (удовлетворять законам сохранения массы импульса, полной

энергии). Развитие принципа консервативности привело к понятию полной консервативности: при конечных величинах шагов сетки помимо основных законов сохранения соблюдаются также балансы в отдельных видах энергии (кинетической, тепловой, электрической и т.д.). Практика показала высокую эффективность полностью консервативных схем.

Разностная схема представляет собой, вообще говоря, систему нелинейных алгебраических уравнений, для решения которой применяются различные итерационные методы. Разработка экономичных методов решения таких систем является серьезной проблемой теории численных методов.

Необходимо также отметить, что преобразование собственно математической модели в вычислительный алгоритм требует проверки на соответствие. Эта проверка в принципе может быть проведена формальными методами, так как преобразуются два формальных объекта - ММ и алгоритм.

После разработки эффективного алгоритма следует этап его программной реализации. Этот этап определяется организацией вычислений и технологией программирования. Новая технология строится на основе модульной (блочной) структуры математической модели и алгоритма. Это обусловлено “многовариантностью” и “многомодельностью” вычислительного эксперимента, проявляющихся в многократных изменениях структуры программы (для варьирования параметров задач) и изменениях ММ (уточнениях). На этом этапе производится модульный анализ всей задачи, т. е. определение состава математических модулей и их взаимосвязей в различных схемах решения задач из рассматриваемого класса. Здесь же устанавливается структура программных модулей, общая архитектура пакета прикладных программ (ППП), характер информационных потоков, операционная среда и требования к пользовательским интерфейсам.

Программное обеспечение (ПО) должно создаваться и функционировать с учетом возможности пополнения, совершенствования и обновления его компонент (принцип развития). Языковые и информационные средства программирования и связи между компонентами должны обеспечивать их совместное функционирование и сохранять открытую структуру (принцип совместимости). При этом необходимо стремиться к использованию унифицированного и стандартного ПО и созданию единого информационно совместимого программного комплекса.

Этот этап включает разработку программных модулей, их автономную отладку, комплексную отладку, тестирование и документирование. Качество и производительность разработки здесь определяется дисциплиной и стилем программирования, полнотой состава и эффективностью инструментальных средств. По этому важно использование средств автоматизации программирования, использование ППП и библиотек программных компонентов, использование визуального программирования и объектно-ориентированного программирования. Большое значение на этом этапе имеет выбор языка и среды программирования, который в значительной степени определяется целью и задачами моделирования.

Следующим этапом является этап проведения практических расчётов на ЭВМ. Этот этап связан с подготовкой исходных данных, проведением вычислительных сеансов, выводом и обработкой результатов расчётов. Здесь особое значение имеет оперативность выполнения работ и эффективность использования вычислительных

ресурсов, что определяется качеством и полнотой ПО, как прикладного, так и системного, и уровнем организации его эксплуатации.

Завершающим этапом является этап анализа и интерпретации результатов и принятия решения о ходе ВЭ. Здесь на основе сравнения численного решения с данными натурного эксперимента на реальном объекте, или данными физического моделирования делаются выводы об адекватности модели и целесообразности продолжения расчётов по разработанным программам, или об их модернизации, или модификации алгоритмов, или об изменении ММ.

Вопросы для самоконтроля:

1. Этапы вычислительного эксперимента.
2. Преимущества вычислительного эксперимента.
3. Математическая модель.

Литература: [1]; [2]; [5]; [6]; [7]; [8]; [9]; [12]; [14]

Тема № 9. Алгоритмические языки и пакеты прикладных программ.

Материалы лекций

Высокоуровневый язык программирования — язык программирования, разработанный для быстроты и удобства использования программистом. Основная черта высокоуровневых языков — это абстракция, то есть введение смысловых конструкций, кратко описывающих такие структуры данных и операции над ними, описания которых на машинном коде (или другом низкоуровневом языке программирования) очень длинны и сложны для понимания.

Так, высокоуровневые языки стремятся не только облегчить решение сложных программных задач, но и упростить портирование программного обеспечения. Использование разнообразных трансляторов и интерпретаторов обеспечивает связь программ, написанных при помощи языков высокого уровня, с различными операционными системами и оборудованием, в то время как их исходный код остаётся, в идеале, неизменным.

Такого рода оторванность высокоуровневых языков от аппаратной реализации компьютера помимо множества плюсов имеет и минусы. В частности, она не позволяет создавать простые и точные инструкции к используемому оборудованию. Программы, написанные на языках высокого уровня, проще для понимания программистом, но менее эффективны, чем их аналоги, создаваемые при помощи низкоуровневых языков. Одним из следствий этого стало добавление поддержки того или иного языка низкого уровня (язык ассемблера) в ряд современных профессиональных высокоуровневых языков программирования.

Примеры: C++, Visual Basic, Python, Perl, Delphi (Pascal), строковых типов, объектов, операций файлового ввода-вывода и т. п.

Первым языком программирования высокого уровня считается компьютерный язык Plankalkül разработанный немецким инженером Конрадом Цузе ещё в период

1942—1946 гг. Однако, широкое применение высокоуровневых языков началось с возникновением Фортрана и созданием компилятора для этого языка (1957).

Распространено мнение, что программы на языках высокого уровня можно написать один раз и потом использовать на компьютере любого типа. В действительности же это верно только для тех программ, которые мало взаимодействуют с операционной системой, например, выполняют какие-либо вычисления или обработку данных. Большинство же интерактивных (а тем более мультимедийных) программ обращаются к системным вызовам, которые сильно различаются в зависимости от операционной системы. Например, для отображения графики на экране компьютера программы под Microsoft Windows используют функции Windows API, которые недоступны в системах, поддерживающих стандарт программный интерфейс X-сервера.

К настоящему времени создан целый ряд программных библиотек (например, библиотека wxWidgets), скрывающих несоответствия системных вызовов различных операционных систем от прикладных программ. Однако такие библиотеки, как правило, не позволяют полностью использовать все возможности конкретных операционных систем.

Новой тенденцией является появление языков программирования еще более высокого уровня (ультра-высокоуровневых). Такого рода языки характеризуются наличием дополнительных структур и объектов, ориентированных на прикладное использование. Прикладные объекты, в свою очередь, требуют минимальной настройки в виде параметров и моментально готовы к использованию. Использование ультра-высокоуровневых языков программирования снижает временные затраты на разработку программного обеспечения и повышает качество конечного продукта за счет, опять таки, уменьшения объема исходных кодов. Пример: Application XML.

Все программы, работающие на компьютере, можно условно разделить на три категории:

- системные программы;
- прикладные программы (приложения);
- системы программирования.

Системные программы выполняются вместе с прикладными и служат для управления ресурсами компьютера — центральным процессором, памятью, вводом-выводом. Это программы общего пользования, которые предназначены для всех пользователей компьютера. Основные функции системного программного обеспечения:

- управление ресурсами компьютера;
- создание копий используемой информации;
- проверка работоспособности устройств компьютера;
- выдача справочной информации о компьютере и др.

Системное программное обеспечение разрабатывается так, чтобы компьютер мог эффективно выполнять прикладные программы. Среди десятков тысяч системных программ особое место занимают операционные системы, которые обеспечивают управление ресурсами компьютера с целью их эффективного использования. Операционная система — это комплекс взаимосвязанных системных программ, назначение которого — организация взаимодействия пользователя с компьютером и выполнение всех других программ. Операционная система обеспечивает совместное

функционирование всех устройств компьютера и предоставляет пользователю доступ к его ресурсам. Важными классами системных программ являются также программы вспомогательного назначения — утилиты (лат. *utilitas* — польза). Они либо расширяют и дополняют соответствующие возможности операционной системы, либо решают самостоятельные важные задачи.

Системы программирования включают в себя язык программирования, средства для удобства редактирования текстов программ, а также методы отладки программ и различные сервисные функции для облегчения труда программиста.

Интерфейс (*interface*) - правила взаимодействия операционной системы с пользователем или прикладными системами (программами, вычислительными средствами).

Язык программирования - это совокупность набора символов (алфавита) системы, правил пользования (синтаксис) и истолкования конструкций из символов (семантика) для задания алгоритмов с использованием символов естественного языка.

Существуют три вида языков программирования:

- языки низкого уровня (на уровне машинных кодов, язык типа ассемблер) - это машинно-ориентированные языки, поэтому и программы на них занимают значительно меньше места и довольно быстроедействие, но работа на таких языках требует от программиста высокого искусства;

- языки высокого уровня приближены к естественному языку, чаще всего к английскому, имеют интерпретатор, т. е. специальное устройство для перевода текста программы в компьютерные коды. Примеры таких языков: Basic, C, Fortran, Pascal и др.;

- объектно-ориентированные языки, в которых оперирование идет объектами (окно, меню, кнопка, указатель и т. д.), позволяют писать программы с единым интерфейсом, облегчают программирование рутинных операций оформления программы. Примеры: Object Pascal, Visual Basic, Visual FoxPro, Java и др.

К прикладному программному обеспечению относятся компоненты указанные ниже.

Средства проектирования.

1. Системы управления базами данных (СУБД) - специальные программные продукты для создания и работы с базами данных. Базы данных (БД) - это интегрированный и структурированный набор данных, относящихся к какой-либо области или задаче. БД в том или ином виде - неотъемлемая часть большинства практических задач, в первую очередь - информационно-поисковых систем.

2. Системы искусственного интеллекта - раздел информатики, занимающийся вопросами имитации мышления человека с помощью компьютера. Экспертные системы - это системы искусственного интеллекта, созданные для решения задач на основе возможностей компьютера, знаний и опыта квалифицированных специалистов; представляют собой программу для компьютера, с помощью которой возможно получить приемлемое решение в ситуации, когда формальных, абсолютно точных решений получить нельзя (диагностика заболеваний, поиск полезных ископаемых, педагогика, психология, политика).

3. Системы автоматического проектирования (САПР) - комплекс технических и программных средств, позволяющих создавать всю необходимую конструкторскую и технологическую документацию на отдельные изделия, здания, сооружения.

4. Системы электронного документооборота, включающие в себя:

- системы управления документами, которые обеспечивают интеграцию с приложениями, хранение документов, осуществляют поиск документов по атрибутивной или полнотекстовой индексации;
- системы массового ввода документов, которые позволяют осуществлять ввод с клавиатуры и сканирование, включающее в себя чистку, подготовку к распознаванию, выравнивание изображений и распознавание, которое может быть оптическим или интеллектуальным;
- системы автоматизации деловых процессов, предназначенные для моделирования деятельности каждого сотрудника, работающего с электронным документооборотом.

5. Информационное хранилище - это автоматизированная система, которая собирает данные из существующих баз и внешних источников, формирует, хранит и эксплуатирует информацию как единую.

6. Геоинформационная система - это информационная система, обеспечивающая сбор, хранение, обработку, доступ, отображение и расположение пространственно-координатных данных.

Средства использования.

1. Электронные таблицы представляют собой прямоугольный бланк произвольной формы и размера, отображенный на экране ЭВМ. Причем каждая ячейка этого бланка может хранить не только видимое или невидимое значение величины, но и формулу или целую программу для ее расчета; активизация программы в ячейках может происходить автоматически или по какому-либо событию.

2. Графический редактор - комплекс программ, направленный на обработку изображений компьютерной графики. Графика может быть различных видов:

- коммерческая (диаграммы, графики), чаще всего встроена в электронные таблицы;
- иллюстративная (оформление книг, журналов, создание фильмов);
- научная (трехмерное изображение объектов, визуализация научных опытов, графики функций, решение систем уравнений, моделирование экспериментов);
- когнитивная, позволяющая образно представить различные формулы и закономерности для доказательства сложных теорем, открывает новые возможности для познания законов функционирования сознания.

3. Текстовый редактор - комплекс программ, предназначенный для ввода с клавиатуры текста и дальнейшего редактирования (правка, копирование, создание таблиц и списков, вставка рисунков, расположение их на странице, проверка грамотности, автоматическая нумерация страниц, составление оглавления, списка литературы, вывод на печать и другие возможности).

4. Электронная почта - специальный пакет программ для хранения и пересылки сообщений между пользователями ЭВМ.

5. Электронный офис - интегрированный пакет программ, включающий в себя текстовый редактор, электронные таблицы, системы управления базами данных, средства коммуникации, язык программирования, графический редактор и др.

6. Видеоконференция - сетевая технология, позволяющая участвовать в дискуссии одновременно сразу нескольким пользователям и видеть на мониторе своего компьютера взявшего слово участника.

7. Система групповой работы объединяет средства коллективной работы прикладных приложений с электронной почтой, настольными и офисными приложениями, управление электронным документооборотом, планирование, управлением заданиями (моделирование деловых процессов), календарным планированием.

8. Корпоративные информационные системы - это автоматизированные системы управления крупными, территориально рассредоточенными предприятиями, имеющими несколько уровней управления, построенные посредством новейших информационных технологий.

Данный список не может быть полным и окончательным. К прикладным программным средствам использования можно отнести игровые и обучающие программы, различные видеотренажеры, программы цифровой обработки звука и т. д.

Вопросы для самоконтроля:

1. Языки программирования высокого уровня.
2. Типы программного обеспечения.
3. Прикладное программное обеспечение.

Литература: [1]; [2]; [4]; [8]; [9]; [12]; [14]; [15]; [17]; [19]; [20]; [21].

2.2. Содержание практических занятий

Раздел 1. Интеллектуальные информационные системы и технологии

Тема № 1. Общий анализ современных проблем информационных систем и технологий в области искусственного интеллекта

Вопросы для обсуждения:

1. Инженерия знаний. Экспертные системы
 - обобщенная структура экспертных систем;
 - интеллектуальные роботы;
 - системы общения на естественном языке и речевой ввод-вывод;
2. Понятия о прикладных системах искусственного интеллекта
 - методы вывода (прямой и обратный);
 - инструментальные средства проектирования, разработки и отладки;
 - понятие CASE-технологии.

Задание:

Подготовка материала для обсуждения на практическом занятии.

Литература: [1]; [2]; [5]; [6]; [10]; [11]; [12]; [13].

Тема № 2. Искусственный интеллект: процедуры, задачи и средства автоматизации

Вопросы для обсуждения:

1. Языки высокого уровня
 - Java;
 - Python;
 - C++;
2. Среды программирования для создания экспертных систем
 - LOOP;
 - CLISP;
3. Оболочка разработки экспертных системEXSYS;
 - Фреймворк Tensorflow для машинного обучения.

Задание:

Подготовка материала для обсуждения на практическом занятии.

Литература: [1]; [2]; [5]; [6]; [8]; [9]; [10]; [11];[12]; [13].

Тема № 3. Современные технологии и методы создания систем искусственного интеллекта

Вопросы для обсуждения:

1. Искусственные нейронные сети;
2. Деревья решений, символьные правила;
3. Методы ближайшего соседа и k-ближайшего соседа;
4. Метод опорных векторов;
5. Байесовские сети;
6. Иерархические методы кластерного анализа;
7. Методы поиска ассоциативных правил, в том числе алгоритм Apriori;
8. Эволюционное программирование и генетические алгоритмы.

Задание:

Подготовка материала для обсуждения на практическом занятии.

Литература:[1]; [2]; [5]; [6]; [10]; [11];[12]; [13].

Тема № 4. Оценка адекватности и эффективности разработанных интеллектуальных средств и технологий. Теория принятия решений.

Вопросы для обсуждения:

1. Нахождение оптимального решения:
 - многокритериальные задачи;
 - человеко-машинные процедуры;
2. Принятие решений на основе информации об относительной важности критериев:
 - процесс принятия решений;
 - комбинированные методы.

Задание: Подготовка материала для обсуждения на практическом занятии.

Литература:[1]; [2]; [23].

Раздел 2. Инфокоммуникационные системы и технологии, проблемы и способы их решения

Тема № 5. Численные методы в задачах автоматизации. Оконные преобразования. Вейвлет-анализ

Вопросы для обсуждения:

1. Численное решение систем линейных и нелинейных уравнений;
2. Решение дифференциальных уравнений;
3. Методы интерполяции и аппроксимации;
4. Критерии выбора вейвлет-базиса для данных;
5. Спектральные методы анализа данных.
6. Выделение трендовых составляющих в данных.

Задание: Подготовка материала для обсуждения на практическом занятии.

Литература: [1]; [2]; [5]; [7]; [9]; [14].

Тема № 6. Информационно-управляющие системы. Мобильные технологии

Вопросы для обсуждения:

1. Управление технологическими процессами;
2. Оптимальное управление производственными процессами (MES);
3. Системы управления ресурсами (ERP);
4. Аналитические и прогнозирующие системы (OLAP, CRM).

Задание: Подготовка материала для обсуждения на практическом занятии.

Литература: [1]; [2]; [4]; [9]; [12]; [15]; [16]; [17]; [18]; [19]; [20]; [21]; [22].

Тема № 7. Проблемы больших данных и способы их решения

Вопросы для обсуждения:

1. Структурированные и неструктурированные данные и технология in-memory;
2. Вопросы хранения больших данных и хранение их в Hadoop;
3. Извлечение полезной информации из больших данных.

Задание: Подготовка материала для обсуждения на практическом занятии.

Литература:[1]; [9]; [14]; [15]; [16]; [21].

Тема № 8. Принципы проведения вычислительного эксперимента

Вопросы для обсуждения:

1. Варианты математических моделей;
2. Алгоритмизация;
3. Выбор конечной платформы для программы.

Задание: Подготовка материала для обсуждения на практическом занятии.

Литература:[1]; [2]; [5]; [6]; [7]; [8]; [9]; [12]; [14].

Тема № 9. Алгоритмические языки и пакеты прикладных программ

Вопросы для обсуждения:

1. Среда разработки
 - IntelliJ IDEA;
 - Eclipse;
2. Комплексы программ
 - Matlab;
 - Latex.

Задание: Подготовка материала для обсуждения на практическом занятии.

Литература: [1]; [2]; [4]; [8]; [9]; [12]; [14]; [15];[17]; [19]; [20]; [21].

3. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

3.1. Методические рекомендации по изучению курса

В основу отбора тем для изучения курса были положены компетенции, установленные федеральным государственным образовательным стандартом высшего образования направления подготовки 19.06.01 Информатика и вычислительная техника (уровень аспирантуры). Особенностью курса является определенная последовательность рассмотрения тем, которые выбраны для изучения на лекционных и на практических занятиях.

Теоретические и дискуссионные вопросы, выносимые на практические занятия, задания, задачи позволяет закрепить, расширить и углубить знания, полученные на лекционных занятиях.

Такая последовательность позволяет получить соответствующие знания о Современных проблемах науки в области информационных технологий. Все это позволит обучающимся в дальнейшем применять полученные знания на практике в своей профессиональной деятельности . Предложенная последовательность изучения курса позволяет овладеть категориальным аппаратом , навыками приобретения , пополнения и реализации знаний , необходимых исследователю и управленцу в рассматриваемой предметной области и в целом изучить курс в соответствии с требованиями к его освоению.

Целесообразен следующий механизм работы обучающегося:

1. Прежде чем приступить к изучению курса внимательно изучите содержание и структуру данных методических указаний.
2. Перед лекцией прочтите и уясните тему и содержание лекции.
3. Прочтите конспект прослушанной лекции, проработайте рекомендуемую основную

и дополнительную литературу по теме.

4. Изложите свое понимание темы.

5. Выявите дискуссионные вопросы и сформулируйте свою точку зрения на них, аргументируя ее.

6. После ознакомления с теоретическим материалом ответьте на вопросы для самоконтроля.

7. Закрепление материала проводится на практических занятиях или в результате самостоятельной работы. Каждая тема курса должна быть «проработана» обучающимся в той или иной форме.

3.2. Методические рекомендации по подготовке к практическим занятиям

Значительную роль в изучении предмета выполняют практические занятия, которые призваны, прежде всего, закреплять теоретические знания, полученные в ходе прослушивания и запоминания лекционного материала, ознакомления с учебной и научной литературой, а также выполнения самостоятельных заданий. Практические занятия способствуют получению наиболее качественных знаний, помогают углубить навыки самостоятельной работы.

Приступая к подготовке темы практического занятия, необходимо, прежде всего, внимательно ознакомиться с его планом. Затем необходимо изучить соответствующие конспекты лекций, главы учебников и методических пособий, разобрать примеры, ознакомиться с дополнительной литературой (справочниками, энциклопедиями, словарями, диссертационными работами по близкой тематике). К наиболее важным и сложным вопросам темы желательно составлять конспекты ответов. Конспектирование дополнительных источников, особенно освещающих вопросы изучаемой темы НИР также способствует более плодотворному усвоению учебного материала.

Следует готовить все вопросы соответствующего занятия. При этом необходимо уметь давать определения основным понятиям, знать основные положения теории, правила и формулы, предложенные для запоминания. Это помогает понять построение изучаемого материала, выделять в нем основные моменты и уметь видеть связь явлений и их причинно-следственные связи. Ведение записей, особенно сделанных в электронном виде, способствует превращению чтения в активный процесс, мобилизует, наряду со зрительной, и моторную память. Следует помнить: у аспиранта, систематически ведущего записи по мере проработки лекций, самостоятельной подготовки тем, подготовки к практическим занятиям, создается свой индивидуальный фонд подсобных материалов для быстрого повторения прочитанного, для мобилизации накопленных знаний. Особенно важны и полезны записи тогда, когда в них находят отражение мысли, возникшие при самостоятельной работе в ходе подготовки к практическим занятиям.

Подготовка к практическому занятию является важной формой самостоятельной работы аспиранта. Она должна носить систематический и планомерный характер. После лекции аспирант должен познакомиться с планом практических занятий и списком обязательной и дополнительной литературы, которую необходимо прочитать, изучить и законспектировать. Разъяснение по вопросам новой темы аспиранты получают у преподавателя в конце предыдущего практического занятия.

Подготовка к практическому занятию требует, прежде всего, чтения рекомендуемых источников и монографических работ, их реферирования, подготовки докладов и сообщений. Важным этапом в самостоятельной работе аспиранта является повторение материала по конспекту лекции. Одна из главных составляющих внеаудиторной подготовки – работа с книгой. Она предполагает внимательное прочтение, критическое осмысление содержания, обоснование собственной позиции по дискуссионным моментам, постановки интересующих вопросов, которые могут стать предметом обсуждения на практическом занятии.

3.3. Методические рекомендации по подготовке к кандидатскому экзамену

В рамках учебного процесса аспирантуры подготовке к кандидатскому экзамену по специальности принадлежит особо важное место, поскольку он является наиболее значимым показателем квалификации выпускника аспирантуры, уровня самостоятельности научного мышления, эрудиции будущего кандидата наук, преподавателя и исследователя. Именно поэтому экзамен по специальности завершает цикл кандидатских экзаменационных испытаний, а вслед за его сдачей аспирант вступает в стадию написания самой кандидатской диссертации.

Освоение основной программы следует начать с изучения материалов университетских учебников и специальных работ по экологии и гидробиологии. Однако, освоение содержащегося в учебнике материала представляет собой лишь начальную стадию подготовки к экзамену, поскольку он решает качественно иные задачи. Ответ экзаменуемого на этом экзамене должен существенно отличаться от ответа студента и по содержанию, и по внутренней структуре. Обязательным требованием к ответу на любой вопрос программы является характеристика степени его изученности в научной литературе. При этом важно показать не только знание современного состояния изученности того или иного вопроса, но и историю его изучения, ученых, внесших вклад в развитие соответствующей области знаний.

В ходе ответа необходимо выделить наиболее дискуссионные и недостаточно изученные моменты для данной тематики. При ответе на вопрос следует воспользоваться возможностью показать свои знания в области методов проведения подобного рода гидробиологических исследований. В ходе ответа необходимо обратить особое внимание на литературный стиль изложения, правильное понимание объема гидробиологических понятий и научных терминов.

Экзаменационный билет включает в себя три вопроса. Каждый билет составлен при этом таким образом, чтобы проверить знания аспиранта по истории разным разделам программы. Экзаменаторы имеют право задать аспиранту дополнительные вопросы по завершении им ответа, имеющие целью уточнить оставшиеся неясными моменты, а также составить более полное представление об уровне его подготовки. Дополнительные вопросы могут быть связанными с проблематикой вопросов экзаменационного билета, однако члены экзаменационной комиссии имеют также право задать любой вопрос, присутствующий в содержании программы экзамена.

Итоговая оценка складывается из ответов на все вопросы билета, при условии, что все они положительные. Она выносится членами экзаменационной комиссии после совещания и затем доводится до сведения аспиранта.

4. УЧЕБНО – МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ

4.1. Основная литература

1. Сергиенко А.Б. Цифровая обработка сигналов: учебник, 2-е изд.. – СПб.: Питер, 2007. – 751 с
2. Суворова Н.И. Информационное моделирование. Величины, объекты, алгоритмы. – М.: Лаборатория Базовых Знаний, 2002. – 128 с

4.2. Дополнительная литература

3. Рычка И.А. Разработка и стандартизация программных средств и информационных технологий: метод. Указания. – Петропавловск-Камчатский : КамчатГТУ, 2008. – 65 с.
4. Мандрикова О.В. Т.Л Заляев Методы анализа вариаций космических лучей в задачах исследования гелиосферных процессов и выделения спорадических эффектов: монография. – Петропавловск-Камчатский : КамчатГТУ, 2016. – 129 с.

5. Чебраков Ю.В. Методы системного анализа в экспериментальных исследованиях. – СПб: СПбГУ, 2000. – 116 с.
6. Вержбицкий В.М. Основы численных методов: учебник. – М.: Высшая школа, 2002. – 840 с.
7. Гулятьев А.К. MatLab 5.2. Имитационное моделирование в среде Windows. – СПб.: Корона-принт, 1999. – 288 с.
8. Дьяконов В. MATLAB. Обработка сигналов и изображений: спец. справочник / В. Дьяконов, И. Абраменко. – СПб: Питер, 2002. – 608 с.

4.3. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

9. Качала В.В. Основы теории систем и системного анализа, 2-е изд., 2016г. [электронный ресурс]. – Издательство Лань. - ЭБС ЛАНЬ
10. Осипов Г.С. Методы искусственного интеллекта [электронный ресурс]. – Изд-во Физматлит, 2016 г. ЭБС ЛАНЬ
11. Остроух А.В., Николаев А.Б. Интеллектуальные информационные системы и технологии, 2019 г - коллекция "Информатика - Издательство Лань" ЭБС ЛАНЬ.
12. Осипов Г.С. Методы искусственного интеллекта, 2011 г. -коллекция "Информатика - Издательство Физматлит" ЭБС ЛАНЬ.
13. Косарев Е.Л. Методы обработки экспериментальных данных, 2-е изд., 2008 г." - коллекция "Информатика - Издательство "Физматлит" ЭБС ЛАНЬ
14. Биллиг В.А. Параллельные вычисления и многопоточное программирование, 2-е Изд., 2016 г. - коллекция "Информатика - Национальный Открытый Университет ИНТУИТ" ЭБС ЛАНЬ.
15. Богачёв К.Ю. Основы параллельного программирования, 3-е изд., 2015 г.- коллекция "Информатика - Издательство Лаборатория знаний" ЭБС ЛАНЬ.
16. Ипатова Э.Р. Методологии и технологии системного проектирования информационных систем, 2-е изд., 2016 г. - коллекция "Информатика - Издательство ФЛИНТА" ЭБС ЛАНЬ.
17. Казаков Ю.М., Тищенко А.А., Кузьменко А.А., Леонов Ю.А., Леонов Е.А. Методология и технология проектирования информационных систем, 2018 г. – коллекция "Информатика — Издательство "ФЛИНТА" ЭБС ЛАНЬ.
18. Масленникова О.Е., Назарова О.Б. Разработка реляционных баз данных с использованием CASE-средства: - коллекция "Информатика — Издательство "ФЛИНТА" ЭБС ЛАНЬ.
19. Махмутова М.В. Теория и практика разработки баз данных, 2017 г" – коллекция "Информатика - Издательство ФЛИНТА" ЭБС ЛАНЬ.
20. Остроух А.В., Помазанов А.В. Теория проектирования распределенных информационных систем, 2019 г. - коллекция "Информатика — Издательство "Лань" ЭБС ЛАНЬ.
21. Остроух А.В. Суркова Н.Е. Проектирование информационных систем, 2019 г." - коллекция "Информатика — Издательство "Лань" ЭБС ЛАНЬ.
22. Королёв В.Ю., Бенинг В.Е., Шоргин С.Я. Математические основы теории риска, 2-е изд., 2011 г. - коллекция "Математика -Издательство Физматлит" ЭБС ЛАНЬ.