

Вопросы к экзамену по дисциплине «Интеллектуальные системы и технологии»

Оглавление

1. Искусственный интеллект. Интеллектуальные системы. Понятие, примеры, используемые современные технологии. Классы интеллектуальных систем. Одноагентные и многоагентные интеллектуальные системы.	3
2. Интеллектуальные пространства. Понятие, примеры систем, используемые технологии, повсеместные вычисления, Internet Of Things. Киберфизические системы.	5
3. Семантический Веб. OWL, RDF(S), SPARQL. Эволюция Семантического Веб.	7
4. Онтологии. Понятие, примеры, возможности использования. Проектирование онтологий. Редакторы онтологий.	9
5. Платформа Smart-M3. Протокол SSAP. Модель общего пространства. Системы публикации и подписки.	10
6. Данные, информация и знания. Понятия, отличия, примеры. Виды и типы знаний. Управление знаниями.	12
7. Модели представления знаний. Классификация, описание, преимущества и недостатки семантических моделей.	14
8. Модели представления знаний. Классификация, описание, преимущества и недостатки логических моделей.	17
9. Понятие и свойства интеллектуального агента. Сильное и слабое определения.	19
10. Экспертные системы. Классификация экспертных систем. Понятие, типовые задачи, характерные черты. Роль эксперта в разработке экспертной системы. Преимущества и недостатки.	21
11. Экспертные системы. Структура экспертной системы. Основные этапы разработки. Режимы работы.	24
12. Инженерия знаний. Основные аспекты извлечения знаний. Основные методы практического извлечения знаний.	26
Психологический аспект.	26
Лингвистический аспект.	27
Гносеологический аспект.	27
13. Представление знаний с помощью логики первого порядка. Утверждения, запросы, аксиомы и теоремы.	29
14. Хорновские выражения. Унификация. Алгоритм прямого логического вывода.	30
Хорновские выражения. Унификация. Алгоритм обратного логического вывода.	31
Описательная логика ALC. Синтаксис и семантика.	32
Байесовская сеть. Правила составления байесовской сети. Приближённый вероятностный вывод в байесовских сетях.	33
Теория коэффициентов уверенности. Принцип работы интеллектуальной системы, основанной на коэффициентах уверенности. Вычисление коэффициентов уверенности для конъюнкции, дизъюнкции свидетельств и для нескольких правил.	34
Теория нечётких множеств. Нечёткое множество. Способы представления нечётких множеств. Операции с нечёткими множествами. Нечёткие рассуждения.	36
Деревья решений. Структура дерева решений. Алгоритм построения дерева решений для задачи классификации (ID3).	37
Искусственные нейронные сети. Модель нейрона. Вычислительные возможности одного нейрона. Обучение нейрона.	38
Искусственные нейронные сети. Многослойная сеть. Достоинства и недостатки нейронных сетей. Алгоритм прямого распространения.	41

Искусственные нейронные сети. Многослойная сеть. Алгоритм обратного распространения ошибки.	43
Рекомендующие системы. Назначение и классификация.	44
Контентные рекомендующие системы. Общая архитектура. Представление объектов и профиля пользователя. Достоинства и недостатки.	45
Системы коллаборативной фильтрации. Подход, основанный на оценке сходства. Нормализация, распространенные способы вычисления «сходства». Достоинства и недостатки.	47
Системы коллаборативной фильтрации. Подход, основанный на моделировании латентных факторов. Достоинства и недостатки.	49
Рекомендующие системы. Оценка качества рекомендаций.....	50

1. Искусственный интеллект. Интеллектуальные системы. Понятие, примеры, используемые современные технологии. Классы интеллектуальных систем. Одноагентные и многоагентные интеллектуальные системы.

Искусственный интеллект – направление в информатике, предметом изучения которого является любая интеллектуальная деятельность человека, подчиняющаяся заранее известным законам.

Пример: терминатор Т-800; выделение ключевых слов в тексте (интеллектуальная деятельность, в которой можно определить некие законы, которые можно воплотить в алгоритм и «автоматизировать» = реализовать ИИ).

Интеллектуальные системы - системы для адаптируемого, проактивного и персонализированного решения задач пользователей способные отслеживать текущую ситуацию с целью выявления неких специальных знаний, позволяющих уточнить предложенную задачу и наметить пути ее решения.

Примеры: поисковые системы, к которым можно обращаться с запросами на естественном языке (Google, Яндекс); системы проверки грамматики и правописания при подготовке текстов; системы распознавания текстов; системы распознавания речи.

Интеллектуальные технологии (методы):

- Логические модели;
- Фреймовые модели;
- Семантические модели;
- Нейронные сети.

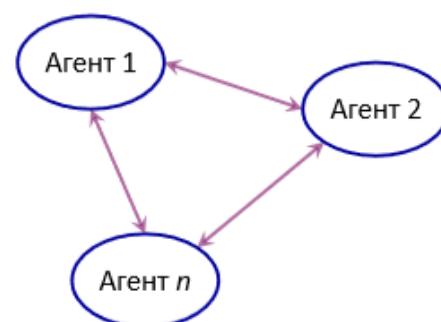
Классы интеллектуальных систем:

- Распознавание естественного языка;
- Интеллектуальные агенты;
- Компьютерное зрение;
- Экспертные системы;
- Автоматическое программирование;
- Нечеткая логика;
- Распознавание речи;
- Робототехника;
- Нейронные вычисления.

«Одноагентные» системы:

- Распознавание естественного языка;
- Компьютерное зрение;
- Экспертные системы;
- Автоматическое программирование;
- Нечеткая логика;
- Распознавание речи;
- Робототехника;
- Нейронные вычисления.

Многоагентные системы



Интеллектуальность во взаимодействии

Агент – это сущность, находящаяся в некоторой среде, от которой она получает информацию, отражающую события в ней происходящие. Сущность интерпретирует эту информацию и исполняет команды, которые воздействуют на среду.

Агент может содержать программные и аппаратные компоненты.

Отсутствие четкого определения мира агентов и присутствие большого количества атрибутов, с ним связанных, а также существование большого разнообразия примеров агентов говорит о том, что агенты это достаточно общая технология, которая аккумулирует в себе несколько различных областей.

2. Интеллектуальные пространства.

Понятие, примеры систем, используемые технологии, повсеместные вычисления, Internet Of Things. Киберфизические системы.

Интеллектуальное пространство

представляет собой сервисориентированную инфраструктуру для возможности обеспечения общего доступа к информации различными устройствами и обладает следующими свойствами:

- Устройства должны быть интегрированы в пространство или динамически соединяться и покидать его.
- Устройства должны осуществлять персонифицированную поддержку пользователя.
- Устройства должны учитывать текущую ситуацию в интеллектуальном пространстве.
- Устройства должны быть адаптивны (реагировать на поведение других устройств и пользователя).
- Устройства должны обеспечивать проактивное поведение (предлагать пользователю полезные в данный момент сервисы без явного запроса от него)

Примеры:

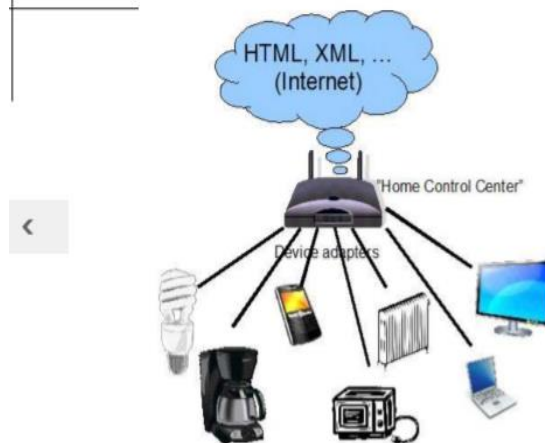
- Умный дом
- Умная конференция
- Умный автомобиль
- Умный город
- Умная больница
- И т.п.

Повсеместные вычисления

Модель человеко-машинного взаимодействия, при котором обработка вовлеченной в интеллектуальную систему информации распределяется и интегрируется в технические устройства, используемые в повседневной деятельности человека. • Различные программно-аппаратные элементы соединенные между собой проводным или беспроводным способом и настолько повсеместны, что остаются незамеченными для человека. • Целью технических устройств является помощь человеку, при этом взаимодействие с таким должно остаться незамеченным

Internet of things – концепция вычислительной сети физических объектов («вещей»), оснащённых встроенными технологиями для взаимодействия друг с другом или с внешней средой, рассматривающая организацию таких сетей как явление, способное перестроить

THE INTERNET OF THINGS



экономические и общественные процессы, исключающее из части действий и операций необходимость участия человека.

Кибер-физические системы – системы, обладающие неразрывной связью между входящими в них вычислительными и физическими элементами – умные системы, охватывающие вычислительные (т.е. аппаратное и программное обеспечение) и эффективно интегрируемые физические компоненты, которые тесно взаимодействуют между собой, чтобы чувствовать изменения состояния реального мира.

Пример: много беспроводных сенсорных сетей контролируют некоторый аспект среды и передают обработанную информацию к центральному узлу; автономные автомобильные системы, медицинский контроль, системы управления техпроцессом, распределенную робототехнику, авионику.



3. Семантический Веб. OWL, RDF(S), SPARQL. Эволюция Семантического Веб.

Семантический веб – это общедоступная глобальная семантическая сеть, формируемая на базе Всемирной паутины путём стандартизации представления информации в виде, пригодном для машинной обработки.

В обычной Всемирной паутине, основанной на HTML-страницах, информация заложена в тексте страниц и предназначена для чтения и понимания человеком.

Семантическая паутина состоит из машинно-читаемых элементов — узлов семантической сети, с опорой на онтологии. Благодаря этому программы-клиенты получают возможность непосредственно получать из интернета утверждения вида «предмет — вид взаимосвязи — другой предмет» и вычислять по ним логические заключения.

Семантическая паутина работает параллельно с обычной Всемирной паутиной и на её основе, используя протокол HTTP и идентификаторы ресурсов URI.

Средства, используемые при описании составляющих Семантического Веб:

- Модели **RDF(S)** и **OWL** для представления знаний предметной области;
- **XML** (eXtensible Markup Language), определяющая синтаксис и структуру инф. в сети Интернет;
- **RDF** (Resource Description Framework) – средство описания ресурсов Семантического Веб;
- **OWL** (Ontology Web Language) – язык описания онтологий, базирующийся на RDF;
- **Protégé** – самый известный на сегодняшний день редактор онтологий, поддерживающий RDF и OWL.

RDFS – язык описания словарей для RDF. RDF Schema определяет классы, свойства и другие ресурсы. RDFS является семантическим расширением RDF.

SPARQL – язык для поиска информации в RDF.

Пример:

```
PREFIX foo: <http://example.com/resources/>
# префиксные объявления
FROM... # источники запроса
SELECT...
# состав результата
WHERE { ... }
# шаблон запроса
ORDER BY...
```

модификаторы запроса

Пример:

```
SELECT ?person
```

```
WHERE {
```

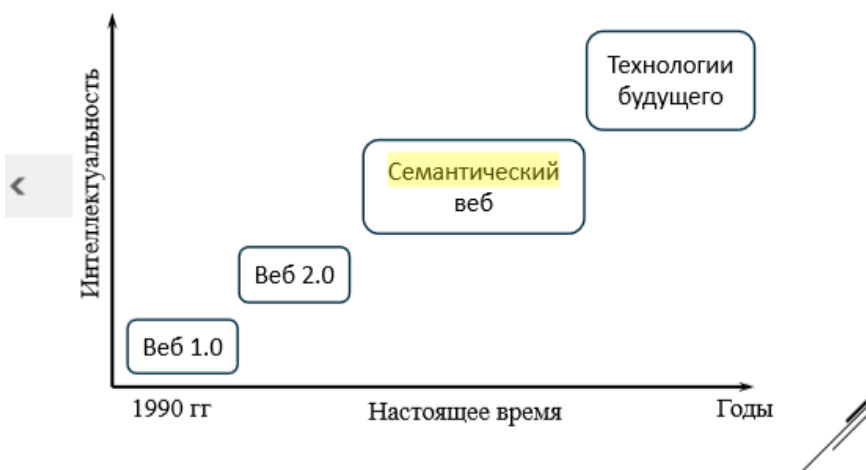
```
    <http://www.foo.com/И.Иванов> <ЖенатНа> ?person
```

```
}
```

OWL предназначен для определения и представления онтологий для Семантического Веб. Язык позволяет описывать предметную область с использованием абстрактных понятий «класс» и «отношения между классами». Он расширяет схему RDFS, позволяя определять сложные взаимосвязи между различными RDFS-классами и более разнообразные ограничения на классы и их свойства.

- OWL Lite: позволяет, описывать таксономию предметной области и простые ограничения. По сравнению с другими разновидностями OWL, поддержка OWL Lite проще для программной реализации.
- OWL DL (аббревиатура DL от англ. descriptive logic): предназначена для пользователей, которым необходима максимальная выразительность (в терминах дескрипционной логики) при сохранении полноты вычислений и разрешимости. Все логические заключения, подразумеваемые той или иной онтологией, будут вычислимыми. Все вычисления завершатся с корректным результатом за конечное время.
- OWL Full: предназначена для использования максимальной выразительности и синтаксической свободы модели RDF. При этом нет гарантий вычислимости.

СЕМАНТИЧЕСКИЙ ВЕБ ЭВОЛЮЦИЯ



4. Онтологии. Понятие, примеры, возможности использования.

Проектирование онтологий. Редакторы онтологий.

Онтология – это подробная спецификация модели предметной области; она включает в себя словарь (т.е. список логических констант и предикатных символов) для описания предметной области и набор логических высказываний, формулирующих существующие в данной проблемной области ограничения и определяющих интерпретацию словаря.

Проектирование онтологий:

- 1) Не существует одной правильной модели предметной области, всегда возможны альтернативные варианты построения онтологии. Лучшее решение всегда зависит от существующего приложения и ожидаемых дополнений.
- 2) Построение онтологий – это обязательно итеративный процесс.
- 3) Концепты онтологии должны быть близки к объектам и их взаимосвязям рассматриваемой предметной области. Это обычно существительные для объектов и глаголы для взаимосвязей из предложений, описывающих предметную область.

Алгоритм онтологического инжиниринга «для чайников» (проф. Т. А. Гаврилова):

- выделение концептов — базовых понятий данной предметной области;
- определение «высоты дерева онтологий» — числа уровней абстракции;
- распределение концептов по уровням;
- построение связей между концептами — определение отношений и взаимодействий базовых понятий;
- консультации с различными специалистами для исключения противоречий и неточностей.

Редакторы онтологий:

- Protégé (<http://protege.stanford.edu/>)
- WebODE (<http://webode.dia.fi.upm.es/webODE/>)
- OntoSaurus (<http://www.isi.edu/isd/ontosaurus.html>)
- WebOnto (<http://kmi.open.ac.uk/projects/webonto/>)
- OilEd (<http://oiled.man.ac.uk/>)
- OntoEdit (<http://ontoserver.aifb.unikarlsruhe.de/ontoedit/>)
- OilEd (<http://oiled.man.ac.uk/>)
- Ontolingua (<http://www-ksl.stanford.edu>)

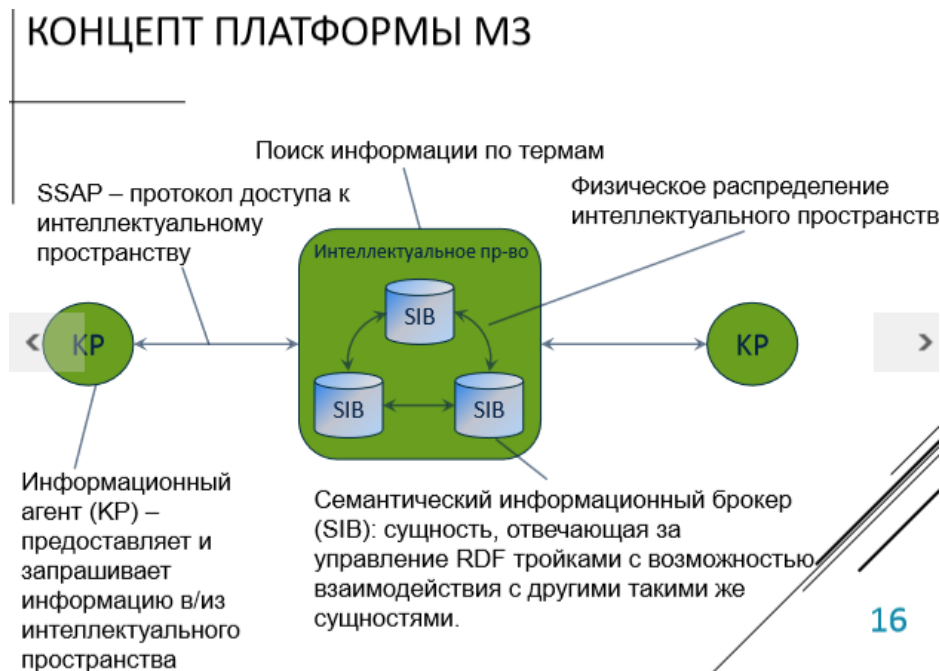
Редактор Protégé

- Изначально создавался для концептуального моделирования в области медицины
- В последнее время используется при создании онтологий для Семантического Веба
- Поддерживаемые языки представления онтологий: DAML+OIL, RDF/RDFS, OWL

5. Платформа Smart-M3. Протокол SSAP. Модель общего пространства.

Системы публикации и подписки.

Платформа Smart-M3 позволяет обеспечивать совместный доступ к информации различных участников интеллектуального пространства, она объединяет в себе идеи распределенных сетевых систем и SemanticWeb



SSAP – протокол доступа к интеллектуальному пространству

Операция	Описание
Join	Подключение к пространству по заданному имени и установка сессии взаимодействия с пространством. В типичном для IoT случае для идентификации также используется IP-адрес и порт сетевой ЭВМ, на которой работает брокер SIB
Leave	Завершение сессии взаимодействия с пространством, т. е. агент KP явно выходит из пространства
Insert	Публикация набора RDF-триплетов (RDF-графа) в информационном хранилище пространства. Выполняется атомарно, независимо от числа триплетов в наборе
Remove	Удаление заданного набора RDF-триплетов (RDF-графа) из информационного хранилища пространства. Выполняется атомарно, независимо от числа триплетов в наборе
Update	Обновление заданного набора RDF-триплетов (RDF-графа) в информационном хранилище пространства. Является атомарной комбинацией последовательного выполнения операций remove и insert
Query	Поисковый запрос на выборку из информационного хранилища пространства. Возможно использование простейших запросов на основе шаблонов триплетов (используется маска типа «*») или языка семантических запросов SPARQL. Возвращается набор найденных RDF-триплетов
Subscribe	Установка сеанса подписки на изменения заданного подмножества информационного содержимого. Интересуемое подмножество специфицируется с помощью шаблонов триплетов или языка SPARQL
Unsubscribe	Завершение сеанса подписки

Модель информационного содержимого в общем пространстве кортежей

- Предложена Дэвидом Гелернтером в 1985 г.

- Определяет хранение и разделяемое использование информационного содержимого в общем пространстве кортежей (от англ. tuplespace).
- Использовалась для параллельного и распределенного программирования.

Модель пространства триплетов

- Предложена Диетером Фенселем.
- Информация в пространстве (триплеты) структурируются в виде RDF-графа.
- Возможность использования пространства как базу знаний.
- Возможность представление знаний предметной области на основе онтологических моделей.
- Возможность вывода нового знания как реакцию на факты, опубликованные другими участниками

Модель CSpace

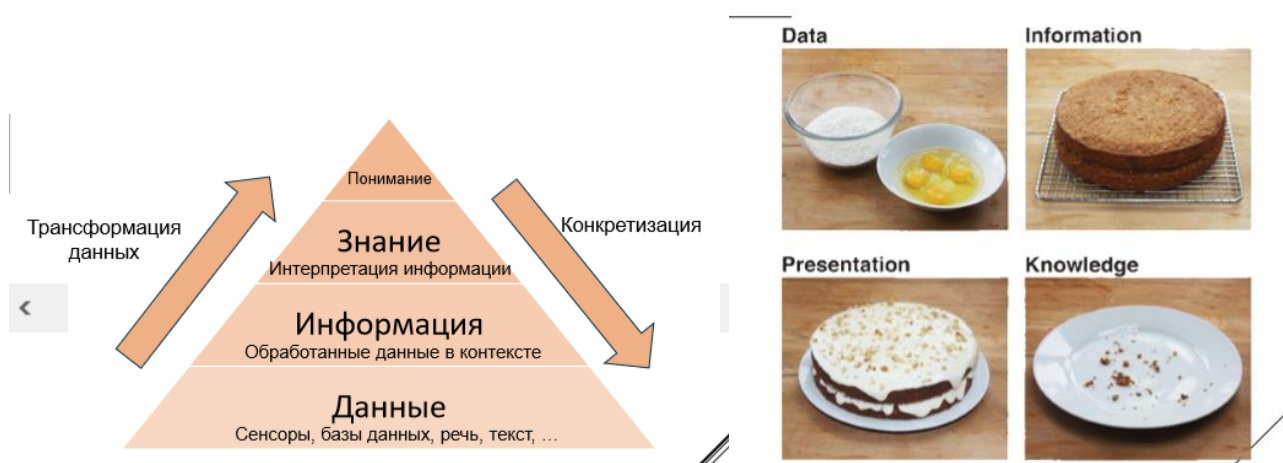
- Расширяет предыдущую операцией подписки.
- Все операции с пространством выполняются как транзакции



6. Данные, информация и знания. Понятия, отличия, примеры. Виды и типы знаний. Управление знаниями.

Данные — это "сырой материал", превращающийся в полезную информацию. Сами по себе исходные данные могут быть совершенно бесполезны, однако большой объем накопленных данных помогает выявить закономерности и тенденции развития.

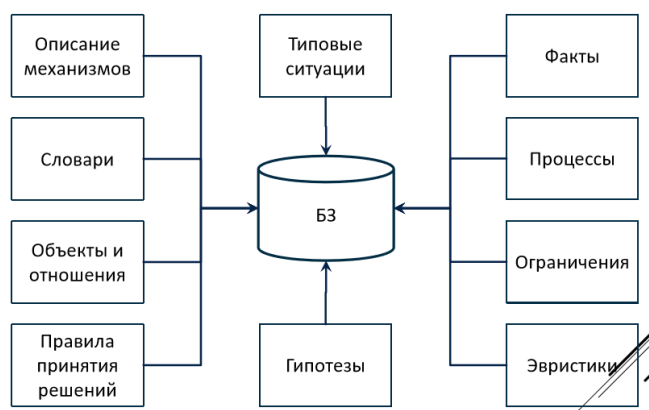
Информация складывается в знания. Простое определение знания — "информация, известная какому-либо человеку". Мудрость — это знания плюс опыт. Знания со временем меняются, мудрость же остается неизменной. Но весь процесс начинается со сбора исходных данных.



Виды знаний:

- Предметные знания – количественные и качественные характеристики конкретных объектов, явлений и их элементов на выбранном представлении.
- Процедурные знания – методы, алгоритмы и программы выполнения различных действий.
- Понятийные знания – совокупность понятий, терминов, свойств, взаимосвязей понятий, используемых в некоторой предметной области.
- Конструктивные знания – знания о возможной структуре и взаимодействии частей различных объектов предметной области

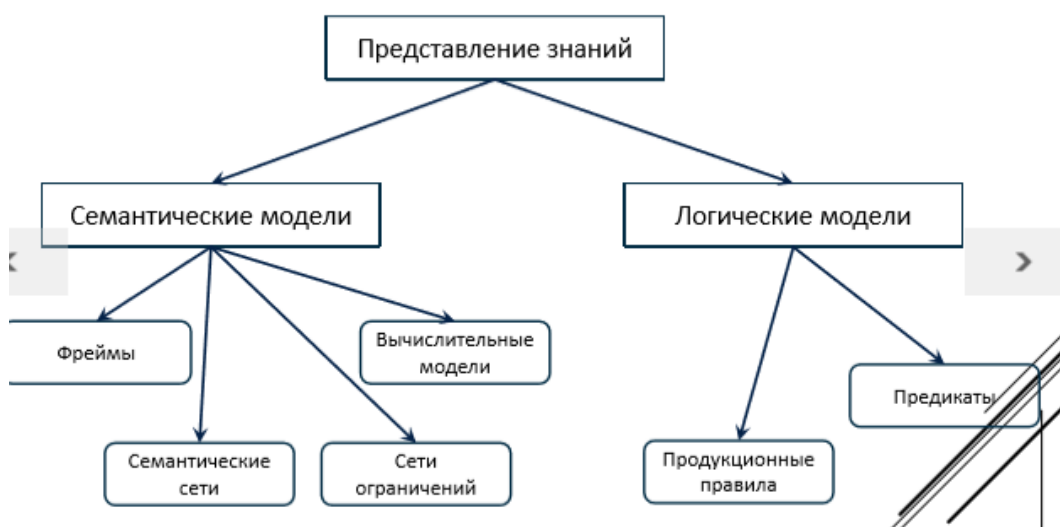
Типы знаний:



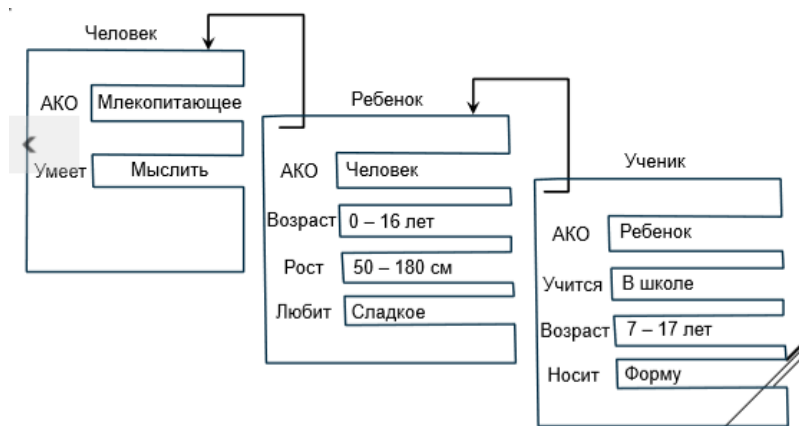
Управление знаниями (менеджемент знаний) – это дисциплина, которая обеспечивает интегрированный подход к созданию, сбору, организации и использованию информационных ресурсов и доступу к ним. Эти ресурсы включают структурированные

БД, текстовую информацию, такую, как документы, и, что наиболее важно, неявные знания и накопленный опыт сотрудников.

7. Модели представления знаний. Классификация, описание, преимущества и недостатки семантических моделей.



Фреймы представляют собой структуры для описания пространственных сцен, типовых объектов, событий, стереотипных ситуаций



Достоинства:

- + Отражает концептуальную основу организации памяти человека [Шенк, Хантер, 1987].
- + Позволяют обеспечить механизм наследования
- + Позволяют формально связывать модульный принцип организации прикладных программных средств с информационными структурами.

Недостатки:

- - Отсутствует возможность интерпретации отношений произвольного типа (только таксономические отношения).
- - Механизмы вывода для фреймового представления слабо развиты.

Семантическая сеть – это ориентированный граф, вершины которого – понятия, а дуги – отношения между ними

По количеству типов отношений

- Однородные – сети с единственным типом отношений.

- Неоднородные – сети с различными типами отношений.

По типам отношений

- Бинарные – сети, в которых отношения связывают два объекта.
- N-арные – сети, в которых есть специальные отношения, связывающие более двух понятий.

Обычно в семантических сетях используются следующие отношения:

- связи типа: «часть -целое» («класс –подкласс», «элемент -множество»);
- функциональные связи;
- количественные связи (больше, меньше, равно и т. п.);
- пространственные (далеко от, близко от, и т. п.);
- временные (раньше, позже и т. п.);
- атрибутивные связи (иметь свойство, иметь значение);
- логические связи (и, или, не);
- Лингвистические связи;
- и др.

Достоинства:

- + Направленность на решение проблемы информационного поиска.
- + Удобная графическая нотация.
- + Универсальность, достигаемая за счет выбора соответствующего набора отношений, позволяющая описать сколь угодно сложную ситуацию, факт или предметную область.

Недостатки:

- - Сетевые модели представляют собой пассивные структуры, для обработки которых необходим специальный аппарат формального вывода.
- - Представление, использование и модификация знаний при описании систем реального уровня сложности оказывается трудоемкой процедурой, особенно при наличии множественных отношений между ее понятиями.

Вычислительная модель, или вычислительная (функциональная) сеть представляет собой двудольный ориентированный граф, включающий вершины двух типов — объекты и операторы (функции). Дуги отражают функциональные связи между операторами и объектами. Дуга, направленная от объекта к оператору, предписывает рассматривать этот объект как аргумент данного оператора, дуга обратной ориентации указывает на то, что объект выступает по отношению к оператору в качестве результата.

Достоинства:

- + Позволяют описать процедуры вычислений одних объектов через другие.
- + Функциональная сеть имеет наглядное представление.

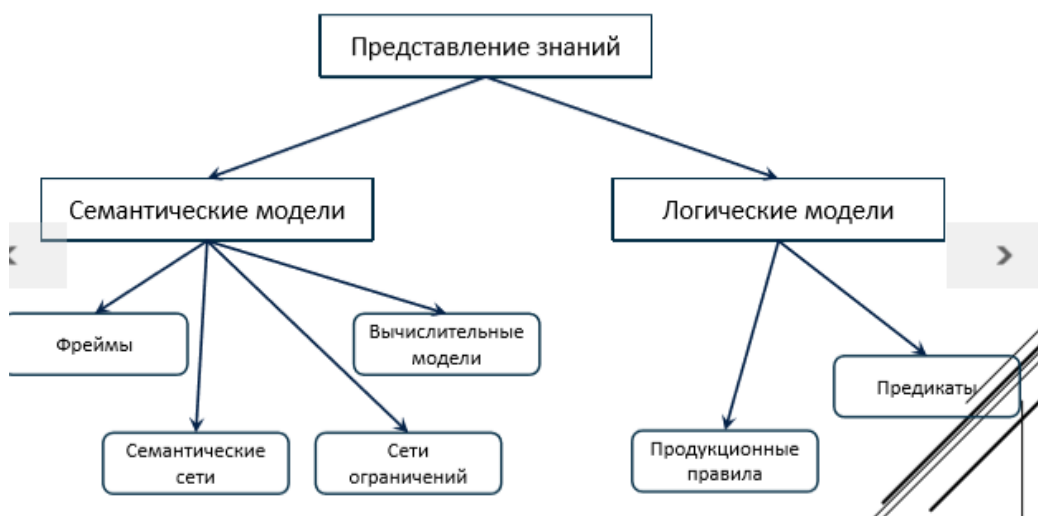
Недостатки:

- - Сложность описания комплексных задач.
- - Может использоваться для узкого круга задач.

Сеть ограничений: (V, dom, C) включает в себя множество переменных $V = \{v_1, \dots, v_i, \dots, v_n\}$, каждая из которых принимает значение в соответствующей области $\text{dom}(i), \text{dom}(j), \dots, \text{dom}(n)$ и множества ограничений C . Каждое ограничение $c(i_1, \dots, i_q)$, ограничивающее подмножество (i_1, \dots, i_q) переменных v определяет, какие из значений переменных согласуются друг с другом.

Позволяют объединить несколько переменных в объект и оперировать терминами ограничения между объектами

8. Модели представления знаний. Классификация, описание, преимущества и недостатки логических моделей.



Продукционная модель

Продукционная модель или модель основанная на правилах позволяет представить знания в виде предложений типа: <Имя правила> ЕСЛИ <условие> ТО <действие> Под условием понимается некоторое предложение-образец, по которому осуществляется поиск в базе знаний, а под действием понимается то, что выполняется при успешном исходе поиска.

Продукционная модель: $i = \langle S; L, A \rightarrow B; Q \rangle$, где S – описание класса ситуаций; L – условие, при котором продукция активизируется; $A \rightarrow B$ – ядро продукции; Q – постусловие продукционного правила.

ЕСЛИ «двигатель не заводится» и «стартер двигателя не работает» ТО «неполадки в системе электропитания стартера»

Предикаты

Информация, необходимая для решения прикладных задач, рассматривается как совокупность фактов и утверждений, которые представляются как формулы в некоторой логике. Знания отображаются совокупностью таких формул, а получение новых знаний сводится к реализации процедур логического вывода. Предикатом называется функция, принимающая два значения (истина или ложь) и предназначенная для выражения свойств объектов или связей между ними. Выражение, в котором утверждается или отрицается наличие каких-либо свойств у объекта, называется высказыванием. ДАТЬ (МИХАИЛ, ВЛАДИМИРУ, КНИГУ);

Достоинства:

- + Хорошо изученный и обоснованный классический аппарат математической логики.

- + Имеют ясную формальную семантику, существование достаточно эффективных процедур вывода.
- + В базах знаний можно хранить лишь множество аксиом, а все остальные знания получать из них по правилам вывода.

Недостатки:

- - С помощью правил, задающих синтаксис языка, нельзя установить истинность или ложность того или иного высказывания, высказывание может быть построено синтаксически правильно, но оказаться совершенно бессмысленным.
- - Отсутствие средств для структурирования используемых элементов и недопустимость противоречий

9. Понятие и свойства интеллектуального агента. Сильное и слабое определения.

Агент – это сущность, находящаяся в некоторой среде, от которой она получает информацию, отражающую события в ней происходящие. Сущность интерпретирует эту информацию и исполняет команды, которые воздействуют на среду.

Агент может содержать программные и аппаратные компоненты.

Отсутствие четкого определения мира агентов и присутствие большого количества атрибутов, с ним связанных, а также существование большого разнообразия примеров агентов говорит о том, агенты это достаточно общая технология, которая аккумулирует в себе несколько различных областей

Под интеллектуальным агентом в слабом смысле понимается программно или аппаратно реализованная система, которая обладает такими свойствами:

- автономность - способность функционировать без вмешательства человека и при этом осуществлять самоконтроль над своими действиями и внутренним состоянием;
- общественное поведение (social ability) - способность функционировать в сообществе с другими агентами, обмениваясь с ними сообщениями с помощью некоторого общепонятного языка коммуникаций;
- реактивность (reactivity) - способность воспринимать состояние среды и своевременно отвечать (реагировать) на те изменения, которые в ней происходят;
- проактивность (pro-activity) - способность агента брать на себя инициативу, т.е. способность генерировать цели и действовать рационально для их достижения, а не только реагировать на внешние события.

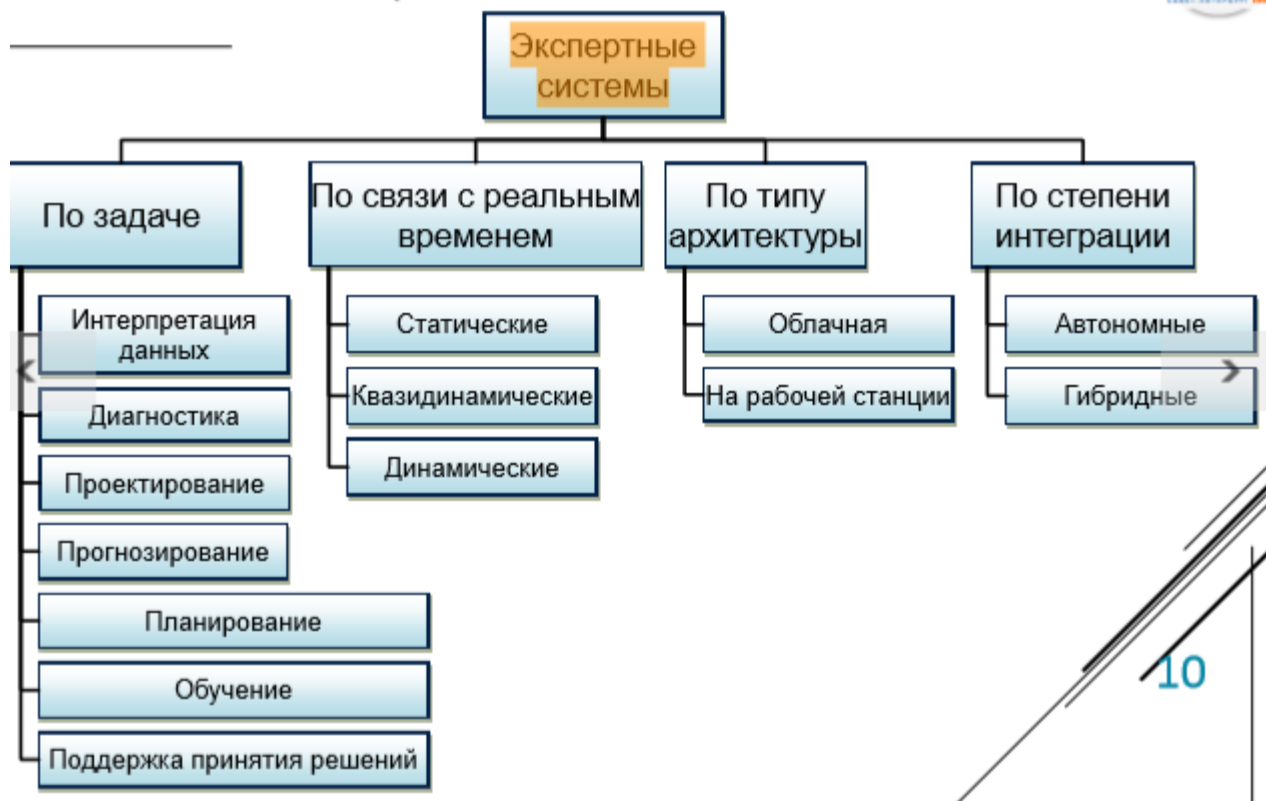
Сильное определение интеллектуального агента подразумевает дополнительно ряд дополнительных свойств:

- знания (knowledge) - это постоянная часть знаний агента о себе, среде и других агентах, т.е. та часть, которая не изменяется в процессе его функционирования;
- убеждения (beliefs, вера) - знания агента о среде, в частности, о других агентах; это те знания, которые могут изменяться во времени и становиться неверными, однако агент может не иметь об этом информации и продолжать оставаться в убеждении, что на них можно основывать свои выводы;
- желания (desires) - это состояния, ситуации, достижение которых по разным причинам является для агента желательным, однако они могут быть противоречивыми и потому агент не ожидает, что все они будут достигнуты;
- намерения (intentions) - это то, что агент или обязан сделать в силу своих обязательств по отношению к другим агентам (ему “это” поручено и он взял эту задачу на себя), или то, что вытекает из его желаний (т.е. непротиворечивое подмножество желаний, выбранное по тем или иным причинам, и которое совместимо с принятыми на себя обязательствами);
- цели (goals) - конкретное множество конечных и промежуточных состояний, достижение которых агент принял в качестве текущей стратегии поведения;

- обязательства по отношению к другим агентам (commitments) задачи, которые агент берет на себя по просьбе (поручению) других агентов в рамках кооперативных целей или целей отдельных агентов в рамках сотрудничества.

10. Экспертные системы. Классификация экспертных систем. Понятие, типовые задачи, характерные черты. Роль эксперта в разработке экспертной системы. Преимущества и недостатки.

КЛАССИФИКАЦИЯ ЭКСПЕРТНЫХ СИСТЕМ



ЭС – это направление исследований в области искусственного интеллекта по созданию вычислительных систем, умеющих принимать решения, схожие с решениями экспертов в заданной предметной области

ЭС может полностью взять на себя функции, выполнение которых обычно требует привлечения опыта человека-специалиста, или играть роль ассистента для лица, принимающего решение.

- ЭС создаются для решения практических задач в некоторых узкоспециализированных областях, где большую роль играют знания «бывалых» специалистов.
- Ценность всей ЭС как законченного продукта на 90% определяется качеством созданной базы знаний
- ЭС является плодом совместной работы экспертов в данной предметной области, инженеров по знаниям и программистов

Типовые задачи ЭС:

- Извлечение информации из первичных данных (потоков данных, баз данных и т.д.);
- Диагностика неисправностей (как в технических системах, так и в человеческом организме);
- Структурный анализ сложных объектов (например, химических соединений);

- Выбор конфигурации сложных многокомпонентных систем(например, распределенных компьютерных систем);
- Планирование последовательности выполнения операций, приводящих к цели (например, выполняемых промышленными роботами).

Характерные черты:

- Четкая ограниченность предметной области.
- Способность принимать решения в условиях неопределенности.
- Способность объяснять ход и результат решения понятным для пользователя способом.
- Четкое разделение декларативных и процедурных знаний (фактов и механизмов вывода).
- Способность пополнять базу знаний.
- Результат выдается в виде конкретных рекомендаций для действий в сложившейся ситуации.
- алгоритм решения не описывается заранее, а строится самой экспертной системой.

РОЛЬ ЭКСПЕРТА В РАЗРАБОТКЕ ЭС НА ПРИМЕРЕ СИСТЕМЫ ДИАГНОСТИКИ ЗАБОЛЕВАНИЙ

- Эксперт предоставляет необходимые знания о тщательно отобранных примерах заболеваний и вариантах лечения
- Эксперт раскрывает список симптомов, которые сопровождают каждое заболевание
- Инженер по знаниям, формализует всю полученную информацию в виде базы знаний

Преимущества:

- Постоянство (человеческая компетенция ослабевает со временем).
- Легкость передачи знаний (Передача знаний от одного человека другому – долгий и дорогой процесс).
- Устойчивость и воспроизводимость результатов (человек легко поддается влиянию внешних факторов, которые непосредственно не связаны с решаемой задачей).
- Стоимость (высококвалифицированные эксперты обходятся компаниям очень дорого)

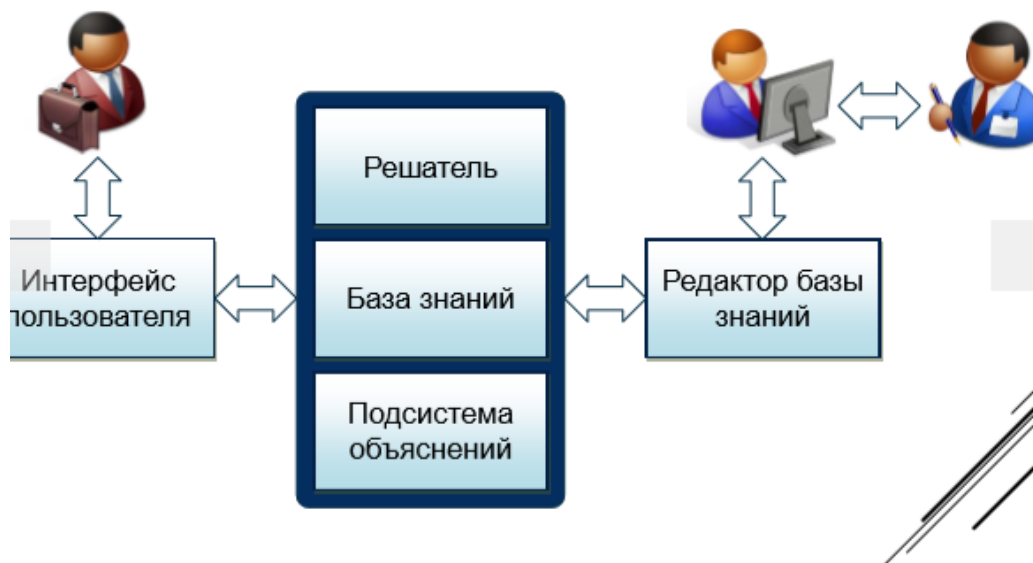
Недостатки:

- Передача экспертным системам «глубоких» знаний о предметной области является большой проблемой. Как правило, это является следствием сложности формализации эвристических знаний экспертов.
- Экспертные системы неспособны предоставить осмысленные объяснения своих рассуждений, как это делает человек. Как правило, экспертные

системы всего лишь описывают последовательность шагов, предпринятых в процессе поиска решения.

- Отладка и тестирование экспертных систем является трудоемким процессом, т.к. требует вовлечение специалистов разного уровня.
- Классические экспертные системы неспособны к самообучению. Для того, чтобы поддерживать экспертные системы в актуальном состоянии необходимо постоянное вмешательство в базу знаний инженеров по знаниям.

11. Экспертные системы. Структура экспертной системы. Основные этапы разработки. Режимы работы.



Этапы разработки ЭС:

- 1) Выбор проблемы
 - 1) Определение проблемной области и задачи
 - 2) Поиск эксперта и коллектива разработчиков
 - 3) Определение подхода к решению проблемы
 - 4) Анализ расходов на разработку и доходов от нее
 - 5) Подготовка плана разработки
- 2) Разработка прототипа
 - 1) Идентификация проблемы, уточнение задачи
 - 2) Извлечение знаний из эксперта
 - 3) Структурирование знаний
 - 4) Формализация знаний с помощью модели представления
 - 5) Реализация прототипа, тестирование
- 6) Доработка
 - 1) Демонстрационный прототип
 - 2) Исследовательский прототип
 - 3) Действующий прототип
 - 4) Промышленная система
 - 5) Коммерческая система
- 7) Оценка
 - 1) Критерии пользователей
 - 2) Критерии приглашенных экспертов
 - 3) Критерии коллектива разработчиков

Режимы работы

- Режим приобретения знаний

Эксперт, используя компонент приобретения знаний, наполняет систему знаниями, которые позволяют экспертной системе в режиме консультаций самостоятельно решать задачи из проблемной области

- Режим консультации

Конечный пользователь осуществляет взаимодействие с системой для получения решения возникшей проблемы. В зависимости от назначения экспертной системы пользователь может не быть специалистом в данной проблемной области (в этом случае он обращается к экспертной системе за результатом, не умея получить его сам), или быть специалистом (в этом случае пользователь может сам получить результат, но он обращается к экспертной системе с целью либо ускорить процесс получения результата, либо возложить на экспертную систему рутинную работу).

12. Инженерия знаний. Основные аспекты извлечения знаний. Основные методы практического извлечения знаний.

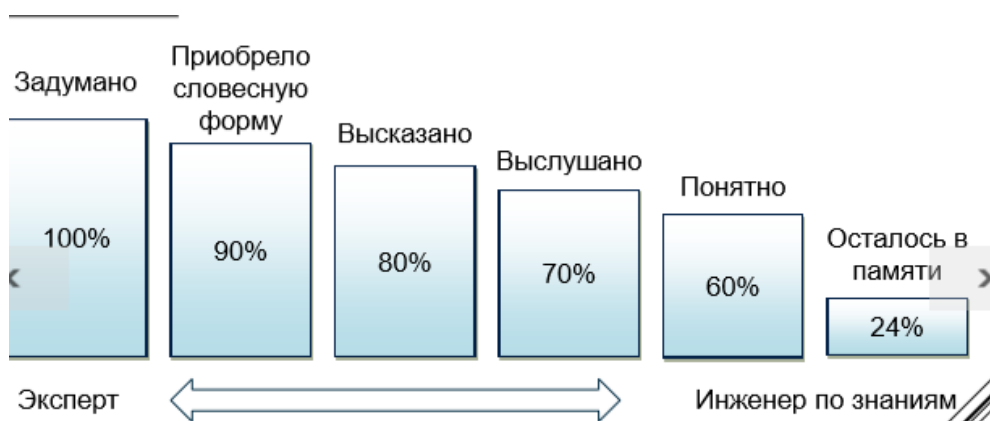
Инженерия знаний – это направление искусственного интеллекта, направленное на извлечение и структурирование знаний экспертов предметной области для последующей разработки баз знаний

Психологический аспект

Психологический аспект является ведущим, он определяет успешность и эффективность взаимодействия инженера по знаниям с экспертами

Уровни общения:

- Уровень манипулирования, при котором один субъект рассматривает другого как средство или помеху по отношению к проекту своей деятельности.
- Уровень «рефлексивной игры», при котором один субъект учитывает «контраргументы» другого, но не признает за ним само ценности.
- Уровень правового общения, при котором субъекты признают право на существование «аргументов» друг друга и пытаются согласовать их хотя бы внешне.
- Уровень нравственного общения, при котором субъекты внутренне принимают общий проект взаимной деятельности



Структурные компоненты модели общения:

- Участники общения (партнеры)
- Средства общения (процедура)
- Предмет общения (знания)

Три слоя психологических проблем:

- Контактный слой. На любой коллективный процесс атмосфера, возникающая в группе участников имеет сильное влияние. Кооперативные, а не конкурентные отношения в коллективе способствуют повышению результативности общения.
- Процедурный слой. Описывает процесс проведения процедуры извлечения знаний (место, время, продолжительность, вспомогательные средства, темп, стиль и т.п.).

В соответствии с исследованиями американского психолога И. Атватера оптимальная дистанция для делового общения 1,2 – 3 м.

- Когнитивный слой. Совокупность критериев предпочтения при решении задач и познании мира, специфическая для каждого человека. Включает в себя исследования познавательных процессов человека с позиции их возможного моделирования.

Лингвистический аспект

Лингвистический аспект касается языковых проблем как основного средства общения в рамках извлечения знаний

- «Общий код»
Сопоставление профессиональной терминологии эксперта и обыденной литературной терминологии инженера по знаниям. «Общий код» объединяет в себе специальную терминологию эксперта, бытовой язык, общенаучные термины и специальные термины из литературы, знакомые инженеру по знаниям.
- Понятийная структура
Особенности формирования понятийной структуры обусловлены установленным постулатом когнитивной психологии о взаимосвязи понятий в памяти человека и наличии семантической сети, объединяющей отдельные термины во фрагменты, фрагменты в сценарии и т.п.
- Словарь пользователя
Необходима разработка специального словаря для пользователя системы, т.к. зачастую профессиональный уровень конечного пользователя не позволяет ему применить специальный язык предметной области в полном объеме.

Гносеологический аспект

Гносеологический аспект объединяет методологические проблемы получения нового знания. Часто, при создании базы знаний эксперт часто впервые формулирует некоторые закономерности



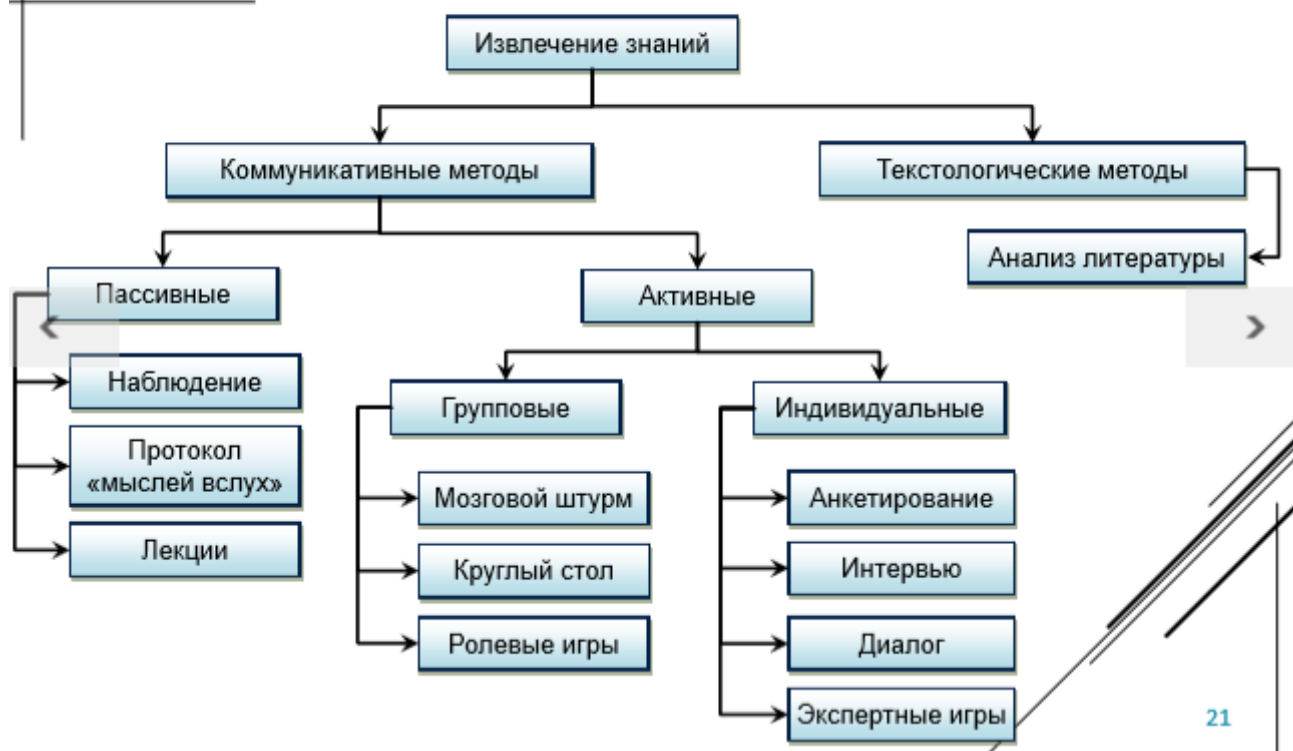
Основные критерии научности нового знания:

- Внутренняя согласованность
Эмпирические знания в голове эксперта обычно являются: модальными (часть закономерностей возможна, часть обязательна и т.д.), противоречивыми (естественное следствие из того, что знания не формализованы в голове эксперта, часто именно противоречия подталкивают эксперта к рассуждениям) и неполными (вытекает из невозможности полного описания предметной области).
- Системность
Ориентирует аналитика на рассмотрение предметной области с позиции закономерностей системного целого и взаимодействия составных его частей.
- Объективность
Процесс познания субъективен, разные эксперты могут дать разную трактовку одного и того же события. Задачей эксперта является понимание закономерностей, а эксперт помогает ему в этом.

- Историзм

Инженер по знаниям должен всегда рассматривать процессы с учетом временных изменений, как связь с прошлым, так и с будущим.

ОСНОВНЫЕ МЕТОДЫ ПРАКТИЧЕСКОГО ИЗВЛЕЧЕНИЯ ЗНАНИЙ



13. Представление знаний с помощью логики первого порядка. Утверждения, запросы, аксиомы и теоремы.

Высказывания вводятся в базу знаний с помощью операции Tell. Такие высказывания называются утверждениями. Например, можно ввести утверждения, что Джон — король и что короли — люди:

- Tell(KB, *King John*)
- Tell(KB, $\forall x \text{King } x \Rightarrow \text{Person}(x)$)

Мы можем задавать вопросы о содержимом базы знаний с использованием операции Ask: Ask(KB, *King John*)

Вопросы, заданные с помощью операции Ask, называются запросами, или целями. На любой запрос, который логически следует из базы знаний, должен быть получен утвердительный ответ:

- Ask(KB, *Person John*) --- True
- Ask(KB, $\exists x \text{Person } x$) --- True

Каждое из этих высказываний может рассматриваться как одна из аксиом в проблемной области родства. Аксиомы предоставляют основную фактическую информацию, на основании которой могут быть получены логическим путем полезные заключения.

Теоремы – высказывания, которые следуют из аксиом.

С логической точки зрения в базе знаний должны содержаться только аксиомы, но не теоремы, поскольку теоремы не увеличивают множество заключений, которые следуют из базы знаний. Но с практической точки зрения важным свойством теорем является то, что они уменьшают вычислительные издержки на логический вывод новых высказываний.

14. Хорновские выражения. Унификация. Алгоритм прямого логического вывода.

Хорновское выражение представляет собой дизъюнкцию литералов, среди которых положительным является не больше чем один.

Например: $P \vee \neg Q \vee \neg R$ – хорновское

$\neg Q \vee \neg R$ – хорновское

$P \vee Q$ – не хорновское

Обобщенное MODUS PONENS

Для атомарных высказываний pi , pi' и q , если существует подстановка θ , такая, что $\text{Subst}(\theta, pi') = \text{Subst}(\theta, pi)$, то для всех i имеет место следующее:

$$\frac{p1', p2', \dots, pn', (p1 \wedge p2 \wedge \dots \wedge pn \Rightarrow q)}{\text{Subst}(\theta, q)}$$

Применение обобщенного М.Р. связано с поиском подстановок, в результате которых различные логические выражения становятся идентичными. Этот процесс называется унификацией и является ключевым компонентом любых алгоритмов вывода в логике первого порядка.

Алгоритм Unify принимает на входе два высказывания и возвращает для них унификатор, если таковой существует: $\text{Unify } p, q = \theta$, где $\text{Subst } \theta, p = \text{Subst } \theta, q$

Идея: рекурсивно исследовать два выражения одновременно, "бок о бок", наряду с этим формируя унификатор, но создавать ситуацию неудачного завершения, если две соответствующие точки в полученных таким образом структурах не совпадают.

Особый случай: если переменная согласуется со сложным термом, необходимо провести проверку того, встречается ли сама эта переменная внутри терма; в случае положительного ответа на данный вопрос согласование оканчивается неудачей, поскольку невозможно сформировать какой-либо совместимый унификатор.

Прямой логический вывод

Идея: В каждой итерации добавлять к базе знаний KB все атомарные высказывания, которые могут быть выведены за один этап из импликационных высказываний и атомарных высказываний, которые уже находятся в базе знаний.

θ – исследуемая подстановка q – голова текущего определенного выражения new - новые высказывания, выводимые в каждой итерации α - запрос - атомарное высказывание

function FOL-FC-Ask (KB, α) **returns** подстановка или значение false

inputs: KB, база знаний – множество определенных выражений первого порядка α , запрос – атомарное высказывание

local variables: new, новые высказывания, выводимые в каждой итерации

repeat until множество new не пусто

new \leftarrow { }

```

for each высказывание  $r$  in KB do  $(p_1 \wedge p_2 \wedge \dots \wedge p_n \Rightarrow q) \leftarrow$ 
Standardize-Apart( $r$ )
    for each подстановка  $\theta$ , такая что  $\text{Subst}(\theta, p_1 \wedge p_2 \wedge \dots \wedge p_n) = \text{Subst}(\theta, p_1' \wedge p_2' \wedge \dots \wedge p_n')$  для
    некоторых  $p_1', p_2', \dots, p_n'$  в базе знаний KB
         $q' \leftarrow \text{Subst}(\theta, q)$ 
        if выражение  $q'$  не является переименованием
        некоторого высказывания, которое уже находится в
        KB, или рассматривается как элемент
        множества new then do добавить  $q'$  к множеству new  $\phi$ 
     $\leftarrow \text{Unify}(q', \alpha)$ 
        if значение  $\phi$  не представляет собой fail then
        return  $\phi$  // (на следующем слайде)
        добавить множество new к базе знаний KB
return false

```

Свойства алгоритма

1. Непротиворечив, поскольку каждый этап представляет собой применение обобщенного М.Р., которое само по себе непротиворечиво. 2. Полон, применительно к базам знаний с определенными выражениями, то есть, способен ответить на любой запрос, ответы на который следуют из БЗ.

Однако в приведенной реализации: • возможна генерация бесконечного числа фактов; • внутренний цикл связан с поиском всех возможных унификаторов, что может быть дорогостоящей операцией; • повторная проверка каждого правила в каждой итерации для определения того, выполняются ли его предпосылки; • может вырабатывать много фактов, не имеющих отношения к текущей цели.

Хорновские выражения. Унификация. Алгоритм обратного логического вывода.

Хорновское выражение представляет собой дизъюнкцию литералов, среди которых положительным является не больше чем один.

Например: $P \vee \neg Q \vee \neg R$ – хорновское

$\neg Q \vee \neg R$ – хорновское

$P \vee Q$ – не хорновское

Обобщенное MODUS PONENS

Для атомарных высказываний p_i, p_i' и q , если существует подстановка θ , такая, что $\text{Subst}(\theta, p_i') = \text{Subst}(\theta, p_i)$, то для всех i имеет место следующее:

$$\frac{p_1', p_2', \dots, p_n', (p_1 \wedge p_2 \wedge \dots \wedge p_n \Rightarrow q)}{\text{Subst}(\theta, q)}$$

Применение обобщенного М.Р. связано с поиском подстановок, в результате которых различные логические выражения становятся идентичными. Этот процесс называется унификацией и является ключевым компонентом любых алгоритмов вывода в логике первого порядка.

Алгоритм Unify принимает на входе два высказывания и возвращает для них унификатор, если таковой существует: $Unify\ p, q = \theta$, где $Subst\ \theta, p = Subst\ \theta, q$
Идея: рекурсивно исследовать два выражения одновременно, "бок о бок", наряду с этим формируя унификатор, но создавать ситуацию неудачного завершения, если две соответствующие точки в полученных таким образом структурах не совпадают.

Особый случай: если переменная согласуется со сложным термом, необходимо провести проверку того, встречается ли сама эта переменная внутри терма; в случае положительного ответа на данный вопрос согласование оканчивается неудачей, поскольку невозможно сформировать какой-либо совместимый унификатор.

Обратный логический вывод

Идея: Алгоритмы обратного логического вывода действуют в обратном направлении, от цели, проходя по цепочке от одного правила к другому, чтобы найти известные факты, которые поддерживают доказательство.

function FOL-BC-Ask(KB, goals, θ) **returns** множество подстановок

inputs: KB, база знаний

goals, список конъюнктов, образующих запрос
(подстановка θ уже применена)

θ , текущая подстановка, первоначально пустая
подстановка { }

local variables: answers, ответы - множество подстановок, первоначально пустое

if список goals пуст **then return** { θ } $q' \leftarrow Subst(\theta, First(goals))$

for each высказывание r in KB, где $Standardize-Apart(r) = (p_1 \wedge p_2 \wedge \dots \wedge p_n \Rightarrow q)$ и $\theta \leftarrow Unify(q, q')$ является выполнимым

new_goals $\leftarrow [p_1, 2, \dots, p_n \mid Rest(goals)]$ answers \leftarrow

FOL-BC-Ask(KB, new_goals, Compose(θ' , θ)) \cup answers

return answers

Описательная логика ALC. Синтаксис и семантика.

Основные задачи логического вывода для описательных логик сводятся к обобщению (проверке того, является ли одна категория подмножеством другой путем сравнения их определений) и классификации (определению принадлежности некоторого объекта к какой-то категории). В некоторых системах предусматривается также проверка непротиворечивости категории.

Описательные логики оперируют понятиями концепт и роль, соответствующими в других разделах математической логики понятиям «одноместный предикат» (или множество, класс) и «двуместный предикат» (или бинарное отношение).

Синтаксис

- Всякий атомарный концепт является концептом;
- выражения \perp и T являются концептами;
- если C есть концепт, то его дополнение $\neg C$ является концептом;
- если C и D есть концепты, то их пересечение $C \sqcap D$ и объединение $C \sqcup D$ являются концептами;
- если C есть концепт, а R есть роль, то выражения $\forall R.C$ и $\exists R.C$ являются концептами.

Семантика

Интерпретация \mathcal{I} состоит из непустого множества $\Delta\mathcal{I}$ (домена) и интерпретирующей функции, которая сопоставляет каждому атомарному концепту A некоторое подмножество $A\mathcal{I}$, а каждой атомарной роли R — некоторое подмножество $R\mathcal{I} \subseteq \Delta\mathcal{I} \times \Delta\mathcal{I}$. Если пара индивидов принадлежит интерпретации некоторой роли R , то есть $(e, d) \in R\mathcal{I}$, то говорят, что индивид d является R -последователем индивида e .

Составные концепты:

- T интерпретируется как весь домен;
- \perp интерпретируется как пустое множество;
- дополнение концепта интерпретируется как дополнение множества;
- пересечение концептов интерпретируется как пересечение множеств;
- объединение концептов интерпретируется как объединение множеств;
- выражение $\forall R.C$ интерпретируется как множество тех индивидов, у которых все R -последователи принадлежат интерпретации концепта C ;
- выражение $\exists R.C$ интерпретируется как множество тех индивидов, у которых имеется R -последователь, принадлежащий интерпретации концепта C .

Байесовская сеть. Правила составления байесовской сети. Приближённый вероятностный вывод в байесовских сетях.

Байесовская сеть — это ориентированный граф, в котором каждая вершина помечена количественной вероятностной информацией.

1. Вершинами сети является множество случайных переменных. Переменные могут быть дискретными или непрерывными.
2. Вершины соединяются попарно ориентированными ребрами; ребра образуют множество ребер. Если стрелка направлена от вершины X к вершине Y , то вершина X называется родительской вершиной вершины Y .
3. Каждая вершина X_i характеризуется распределением условных вероятностей $P(X_i \mid \text{Parents}(X_i))$, которое количественно оценивает влияние родительских вершин на эту вершину.
4. Граф не имеет контуров (Directed Acyclic Graph — DAG)).

Правила составления

Байесовская сеть служит правильным представлением проблемной области, только если каждая вершина в ней условно независима от ее предшественников в

конкретном упорядочении вершин, после того как заданы ее родительские вершины.

Необходимо выбрать для каждой вершины родительские вершины так, чтобы соблюдалось это свойство. Интуитивно ясно, что множество родительских вершин вершины X_i должно включать все такие вершины из множества X_1, \dots, X_{i-1} , которые непосредственно влияют на X_i .

Правильный порядок вершин (от причин к следствиям). Правила должны быть причинными, а не диагностическими.

Приближенный вывод

Например, применение методов выборки (Монте-Карло) для вычисления апостериорных вероятностей. Приближенный алгоритм \Rightarrow точность ответа зависит от количества сформированных выборок.

Идея метода непосредственной выборки: выборка должна формироваться последовательно по каждой переменной, в топологическом порядке.

Распределение вероятностей, из которого берется выборка значения, обуславливается значениями, уже присвоенными родительским переменным.

Ответы вычисляются путем подсчета фактически сформированных выборок.

Теория коэффициентов уверенности. Принцип работы интеллектуальной системы, основанной на коэффициентах уверенности. Вычисление коэффициентов уверенности для конъюнкции, дизъюнкции свидетельств и для нескольких правил.

КОЭФФИЦИЕНТЫ УВЕРЕННОСТИ

...известные также как теория Шортлиффа-Бучанана (Shortliffe-Buchanan), схема Шортлиффа. Впервые применены в экспертной системе медицинской диагностики MYCIN.

Причины: - Выражение экспертами уверенности в тех или иных закономерностях способом, не согласованным со строгой математической теорией - Недостаток достоверных статистических данных (в частности, в области медицинской диагностики)

ЕСЛИ свидетельство E

ТО гипотеза H {cf}

Коэффициент уверенности cf (certainty factor) представляет собой степень уверенности в том, что гипотеза H при условии свидетельства E . Изменяется в пределах от -1 (при соблюдении всех условий существует полная уверенность в ошибочности заключения) до +1 (полная уверенность в правильности заключения). Базовый принцип определения степени уверенности в консеквенте правила при его срабатывании: $cf_{H,E} = cf_E \times cf$

Правило:

ЕСЛИ ЛасточкиЛетаютНизко

ТО БудетДождь { cf=0.85 }

При этом:

$$cf(\text{ЛасточкиЛетаютНизко}) = 0.9$$

Заключаем:

$$cf \text{ БудетДождь, ЛасточкиЛетаютНизко} = 0.9 \times 0.85 = 0.765$$

НЕСКОЛЬКО СВИДЕТЕЛЬСТВ (КОНЪЮНКЦИЯ)

ЕСЛИ E_1 И ... И E_n

ТО $H \{ cf \}$

$$cf H, E_1 \cap E_2 \cap \dots \cap E_n = \min cf E_1, \dots, cf E_n \times cf$$

Правило:

ЕСЛИ ЛасточкиЛетаютНизко И ДымСтелетсяПоЗемле

ТО БудетДождь $\{ cf=0.9 \}$

При этом:

$$cf(\text{ЛасточкиЛетаютНизко}) = 0.9$$

$$cf(\text{ДымСтелетсяПоЗемле}) = 0.8$$

Заключаем:

$$cf \text{ БудетДождь, ЛасточкиЛетаютНизко} \cap \text{ДымСтелетсяПоЗемле} = \min 0.9, 0.8 \times 0.9 = 0.8 \times 0.9 = 0.72$$

НЕСКОЛЬКО СВИДЕТЕЛЬСТВ (ДИЗЪЮНКЦИЯ)

ЕСЛИ E_1 ИЛИ ... ИЛИ E_n

ТО $H \{ cf \}$

$$cf H, E_1 \cup E_2 \cup \dots \cup E_n = \max cf E_1, \dots, cf E_n \times cf$$

Правило:

ЕСЛИ ЛасточкиЛетаютНизко ИЛИ ДымСтелетсяПоЗемле

ТО БудетДождь $\{ cf = 0.9 \}$

При этом:

$$cf(\text{ЛасточкиЛетаютНизко}) = 0.9$$

$$cf(\text{ДымСтелетсяПоЗемле}) = 0.8$$

Заключаем:

$$cf \text{ БудетДождь, ЛасточкиЛетаютНизко} \cup \text{ДымСтелетсяПоЗемле} = \max 0.9, 0.8 \times 0.9 = 0.9 \times 0.9 = 0.81$$

НЕСКОЛЬКО ПРАВИЛ

Два правила с коэффициентами уверенности cf_1 и cf_2 относительно одной гипотезы:

$$cf \text{ } cf_1, cf_2 = \begin{cases} cf_1 + cf_2 \times (1 - cf_1), & \text{если } cf_1 > 0 \text{ и } cf_2 > 0 \\ \frac{cf_1 + cf_2}{1 - \min[cf_1, |cf_2|]}, & \text{если } cf_1 < 0 \text{ или } cf_2 < 0 \\ cf_1 + cf_2 \times (1 - cf_1), & \text{если } cf_1 = 0 \text{ и } cf_2 < 0 \end{cases}$$

Правила:

ЕСЛИ ЛасточкиЛетаютНизко ТО БудетДождь $\{ cf = 0.85 \}$

ЕСЛИ ДымСтелетсяПоЗемле ТО БудетДождь $\{ cf = 0.9 \}$

При этом: $cf(\text{ЛасточкиЛетаютНизко}) = 1.0$

$$cf(\text{ДымСтелетсяПоЗемле}) = 0.7$$

Закключаем:

$$cf(\text{БудетДождь}) = (1 \times 0.85) + (0.7 \times 0.9) \times (1 - (1 \times 0.85)) = 0.9445$$

Теория нечётких множеств. Нечёткое множество. Способы представления нечётких множеств. Операции с нечёткими множествами. Нечёткие рассуждения.

Теория нечётких множеств – один из подходов к определению того, насколько хорошо некоторый объект подходит под расплывчатое описание.

Нечёткая логика – метод формирования рассуждений с помощью логических выражений, описывающих принадлежность элементов к нечётким множествам.

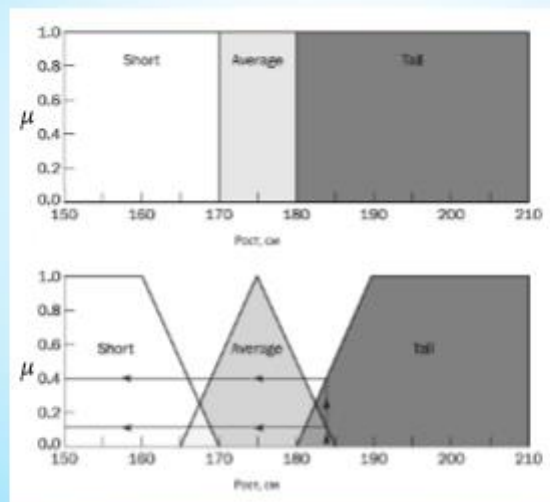
Неопределенность другого плана.

Нечёткое множество A – совокупность упорядоченных пар, составленных из элементов x универсального множества X и соответствующих степеней принадлежности $\mu_A x : A = \{x, \mu_A x \mid x \in X, \mu_A x \in [0,1]\}$ – функция принадлежности (характеристическая функция), показывающая, в какой степени элемент x принадлежит нечёткому множеству A .

ПРИМЕР НЕЧЁТКОГО МНОЖЕСТВА

Универсальное множество – рост человека.

Подмножества – высокие люди (Tall), среднего роста (Average), низкие (Short).



Представление

- Задание функции принадлежности: - сигмоида, гауссиана... (усложняет вычисления)

- линеаризация: $Tall = (0/180, 0.5/185, 1/190)$

Операции

Включение (A содержится в B): $A \subset B \Leftrightarrow \mu_A(x) \leq \mu_B(x)$

Дополнение: $\mu_{\neg A}(x) = 1 - \mu_A(x)$

Пересечение: $\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)]$, где $x \in X$

Объединение: $\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)]$, где $x \in X$

Нечеткие рассуждения

ЕСЛИ x is A3 ЕСЛИ project_funding is adequate
ИЛИ y is B1 ИЛИ project_staffing is small
ТО z is C1 ТО risk is low

ЕСЛИ x is A2 ЕСЛИ project_funding is marginal
И y is B2 ИЛИ project_staffing is large
ТО z is C2 ТО risk is normal

ЕСЛИ x is A1 ЕСЛИ project_funding is small
ТО z is C3 ТО risk is high

Деревья решений. Структура дерева решений. Алгоритм построения дерева решений для задачи классификации (ID3).

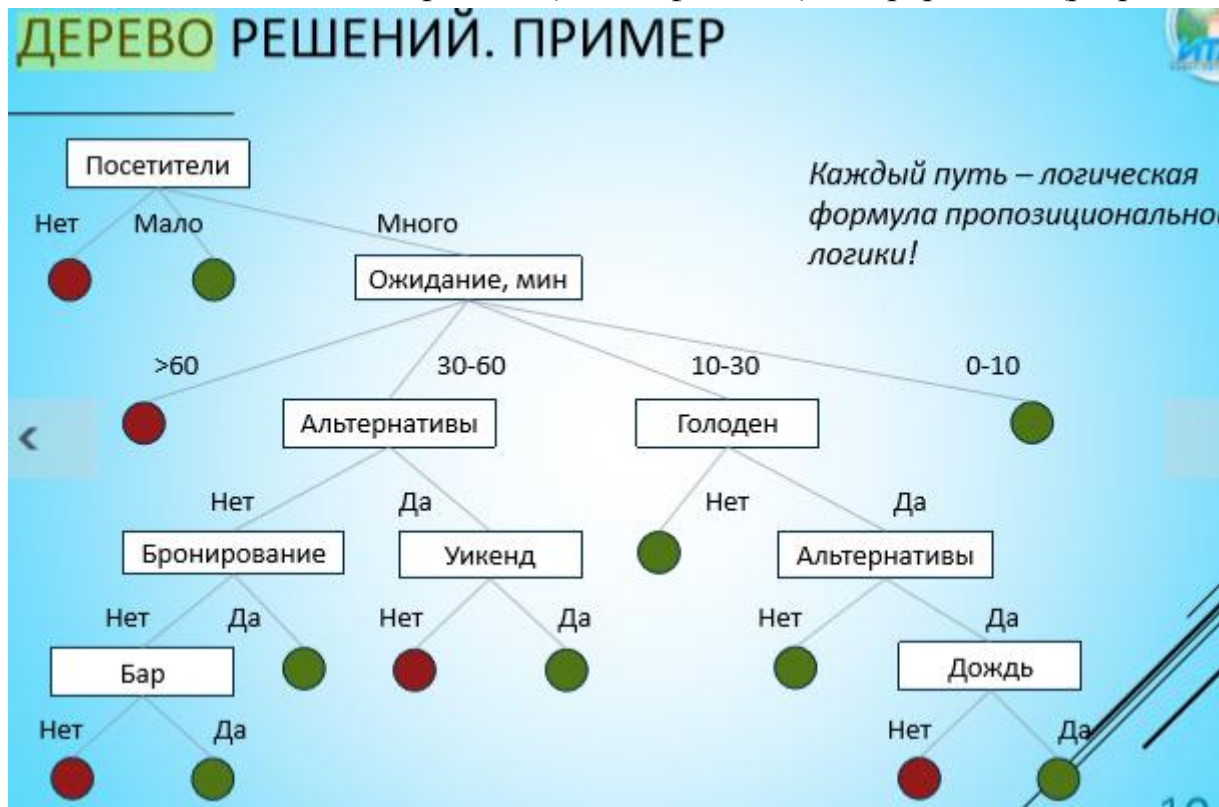
Дерево решений – это дерево, обладающее следующими метками:

- У внутренней вершины – это атрибут.
- У листовой вершины – значение целевой функции.
- У ребра – значения атрибута.

Дерево решений принимает в качестве входных данных объект или ситуацию, описанную с помощью множества атрибутов, и возвращает «решение» – предсказанное выходное значение, соответствующее входным данным.

Входные атрибуты: дискретные или непрерывные.

Выходные значения: дискретные (классификация), непрерывные (регрессия).



Алгоритм

Взять все неиспользованные признаки и посчитать их энтропию относительно тестовых образцов

Выбрать признак, для которого энтропия минимальна (а информационная выгода соответственно максимальна)

Сделать узел дерева, содержащий этот признак

Алгоритм следующий:

ID3(Таблица примеров, Целевой признак, Признаки)

1) Если все примеры положительны, то вернуть узел с меткой «+».

2) Если все примеры отрицательны, то вернуть узел с меткой «-».

3) Если множество признаков пустое, то вернуть узел с меткой, которая больше других встречается в значениях целевого признака в примерах.

4) Иначе:

 A — признак, который лучше всего классифицирует примеры (с максимальной информационной выгодой).

 Создать корень дерева решения; признаком в корне будет являться A.

 Для каждого возможного значения A (v_i):

 Добавить новую ветвь дерева ниже корня с узлом со значением $A = v_i$

 Выделить подмножество $\text{Examples}(v_i)$ примеров, у которых $A = v_i$.

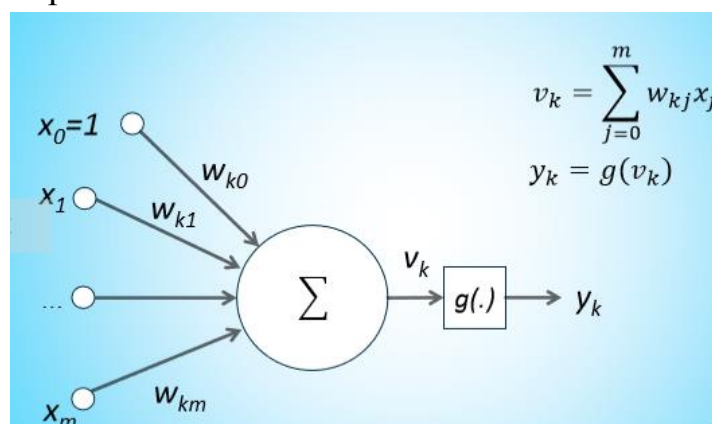
 Если подмножество примеров пусто, то ниже этой новой ветви добавить узел с меткой, которая больше других встречается в значениях целевого признака в примерах.

 Иначе, ниже этой новой ветви добавить поддерево, вызывая рекурсивно $\text{ID3}(\text{Examples}(v_i), \text{Целевой признак}, \text{Признаки})$

5) Вернуть корень.

Искусственные нейронные сети. Модель нейрона. Вычислительные возможности одного нейрона. Обучение нейрона.

Математически нейрон представляет собой взвешенный сумматор, единственный выход которого определяется через его входы и матрицу весов следующим образом:

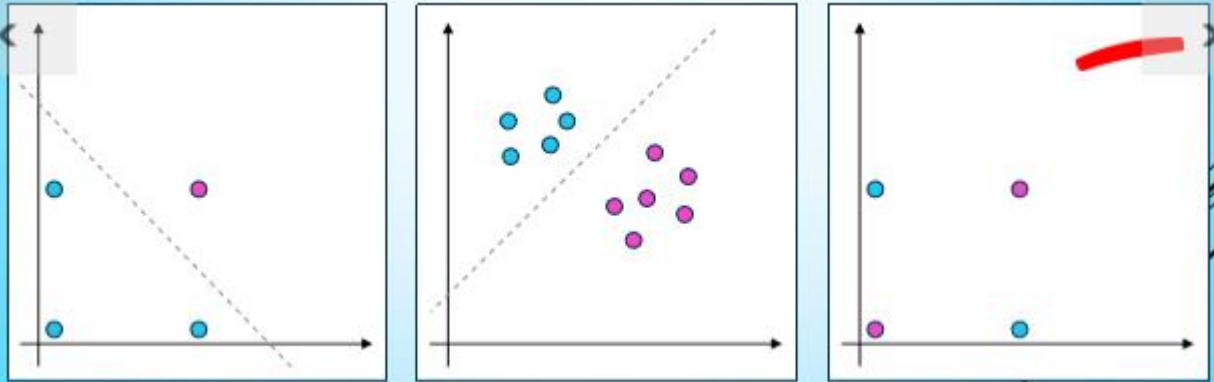


ВЫЧИСЛИТЕЛЬНЫЕ ВОЗМОЖНОСТИ НЕЙРОНОВ



$$y = \begin{cases} 1, & \sum_i w_{ki} x_i > \theta; \\ 0, & \sum_i w_{ki} x_i \leq \theta. \end{cases}$$

Линейная
разделимость



Задача обучения одного нейрона

Дано:

- 1) Обучающее множество $T = \{ (X^{(1)}, y^{(1)}), \dots, (X^{(m)}, y^{(m)}) \}$.
- 2) $X^{(i)} \in \mathbb{R}^n, y^{(i)} \in \{0, 1\}$ (бинарная классификация).

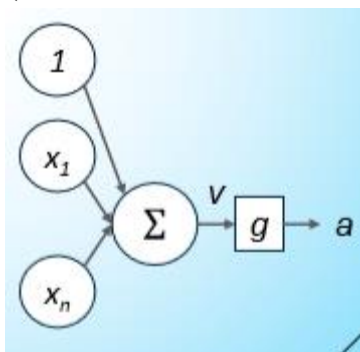
Требуется:

Найти синаптические веса нейрона $w \in \mathbb{R}^{n+1}$, осуществляющего классификацию обучающего множества наилучшим образом.

Допущение: Функция активации – логистическая кривая.

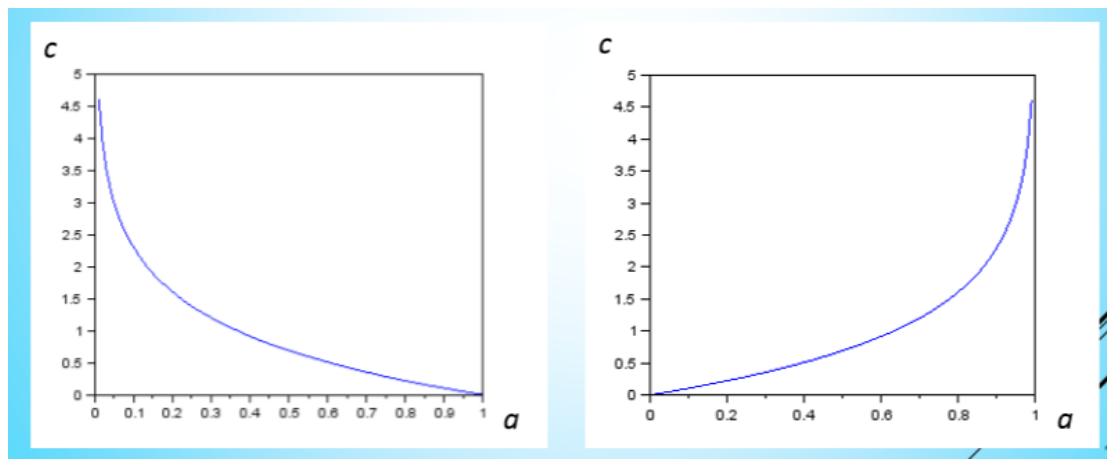
Задача параметрического обучения.

(Та же сама логистическая регрессия)



Пусть $s(a, y)$ – функция стоимости, «штраф», накладываемый на выходное значение a при обработке примера, имеющего «эталонный» ответ y .

$$s(a, y) = \begin{cases} -\log(a), & \text{если } y = 1 \\ -\log(1 - a), & \text{если } y = 0 \end{cases}$$



$$c(a, y) = \begin{cases} -\log(a), & \text{если } y = 1 \\ -\log(1-a), & \text{если } y = 0 \end{cases}$$

$$c(a, y) = -y \log(a) - (1-y) \log(1-a)$$

$$J(w) = \frac{1}{m} \sum_{i=1}^m \left(c(a^{(i)}, y^{(i)}) \right) =$$

$$= -\frac{1}{m} \left[\sum_{i=1}^m (y^{(i)} \log(a^{(i)}) + (1 - y^{(i)}) \log(1 - a^{(i)})) \right]$$

Min J(w)

$$J(w) = -\frac{1}{m} \left[\sum_{i=1}^m (y^{(i)} \log(g_w(x^{(i)})) + (1 - y^{(i)}) \log(1 - g_w(x^{(i)}))) \right]$$

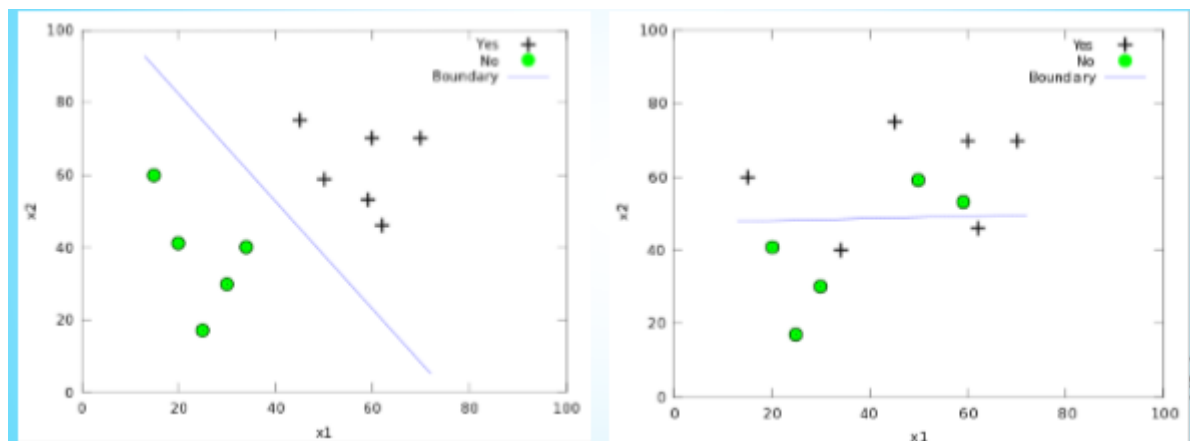
$$\frac{\partial J(w)}{\partial w_j} = \frac{1}{m} \sum_{i=1}^m (g_w(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

Градиентный спуск (например)

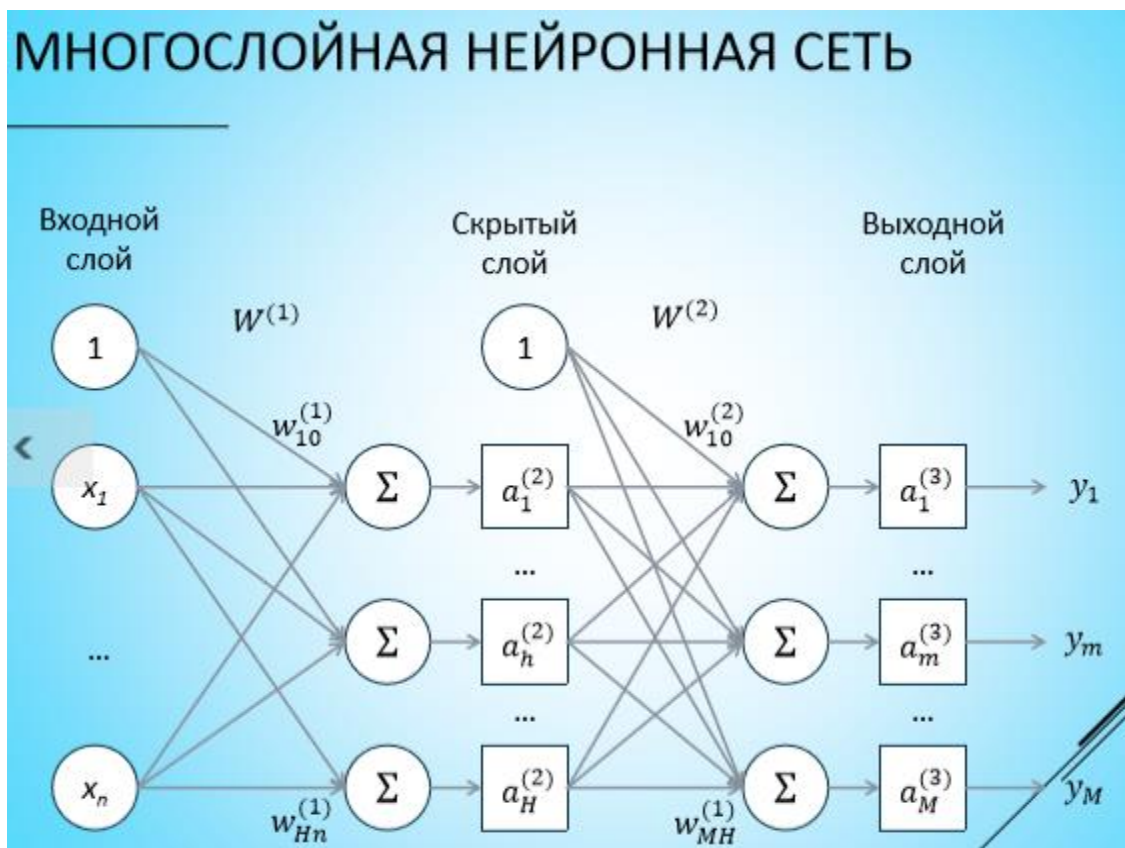
```

w = w0
while (not <условие останова>) do
  nw = w
  for j = 0 .. n do
    nwj = nwj - α  $\frac{1}{m} \sum_{i=1}^m (g_w(x^{(i)}) - y^{(i)}) x_j^{(i)}$ 
  w = nw

```



Искусственные нейронные сети. Многослойная сеть. Достоинства и недостатки нейронных сетей. Алгоритм прямого распространения.



Алгоритм прямого распространения

Исходные данные:

1) нейронная сеть с L слоями; синаптические веса нейронной сети заданы матрицами $W^{(1)}, W^{(2)}, \dots, W^{(L-1)}$; функция активации $g(x)$.

2) входной вектор $x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$

Алгоритм:

1) $a^{(1)} = x$.

2) Для всех $i \in \{1, \dots, L-1\}$ выполнить шаг 3.

3) $a^{(i+1)} = g \left(W^{(i)} \begin{pmatrix} 1 \\ a^{(i)} \end{pmatrix} \right)$

4) $y = a^{(L)}$.

Недостатки

- большинство подходов для проектирования ИНС являются эвристическими и часто не приводят к однозначным решениям;
- для построения модели объекта на основе ИНС требуется выполнение многоцикловой настройки внутренних элементов и связей между ними;
- проблемы, возникающие при подготовке обучающей выборки, связанные с трудностями нахождения достаточного количества обучающих примеров;
- обучение сети в ряде случаев приводит к тупиковым ситуациям;

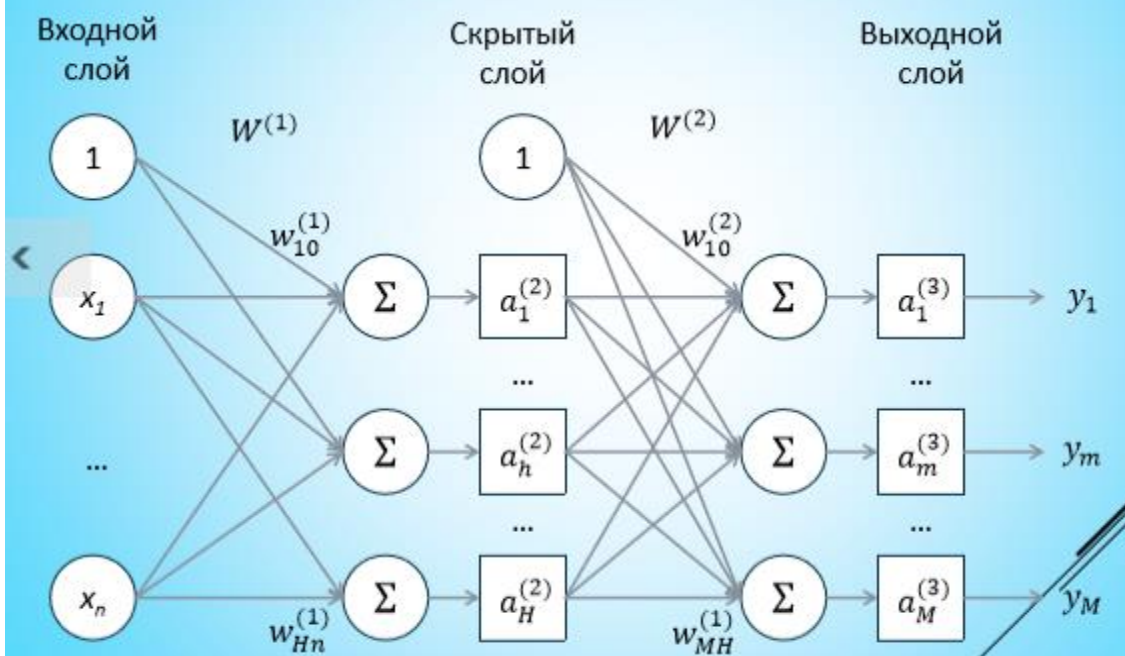
- продолжительные временные затраты на выполнение процедуры обучения зачастую не позволяют применять ИНС в системах реального времени;
- поведение обученной ИНС не всегда может быть однозначно предсказуемо, что увеличивает риск применения ИНС для управления дорогостоящими техническими объектами;
- большинство известных коммерческих продуктов схемотехнической реализации нейронных сетей, выполняются в виде сверхбольших интегральных схем (СБИС), которые сегодня трудно назвать широкодоступными и др.

Достоинства

- А. Нейрокомпьютеры дают стандартный способ решения многих нестандартных задач. И неважно, что специализированная машина лучше решит один класс задач. Важнее, что один нейрокомпьютер решит и эту задачу, и другую, и третью и не надо каждый раз проектировать специализированную ЭВМ, нейрокомпьютер сделает все сам и почти не хуже.
- Б. Вместо программирования обучение. Нейрокомпьютер учится нужно только формировать учебные задачки. Труд программиста замещается новым трудом учителя (может быть, лучше сказать тренера или дрессировщика). Лучше это или хуже? Ни то, ни другое. Программист предписывает машине все детали работы, учитель создает «образовательную среду», к которой приспосабливается нейрокомпьютер. Появляются новые возможности для работы.
- В. Нейрокомпьютеры особенно эффективны там, где нужен аналог человеческой интуиции для распознавания образов (узнавания лиц, чтения рукописных текстов), подготовки аналитических прогнозов, перевода с одного естественного языка на другой и т.п. Именно для таких задач обычно трудно сочинить явный алгоритм.
- Г. Нейронные сети позволяют создать эффективное программное обеспечение для компьютеров с высокой степенью распараллеливания обработки. Проблема эффективного использования параллельных системы хорошо известна многим. Как добиться того, чтобы все элементы одновременно и без лишнего дублирования делали что-то полезное? Создавая математическое обеспечения на базе нейронных сетей, можно для широкого класса задач решить эту проблему.
- Д. Нейросетевые системы «демократичны», они также дружелюбны, как текстовые процессоры, поэтому с ними может работать любой, даже совсем неопытный пользователь

Искусственные нейронные сети. Многослойная сеть. Алгоритм обратного распространения ошибки.

МНОГОСЛОЙНАЯ НЕЙРОННАЯ СЕТЬ



ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ (1)

Для одного нейрона:

$$J(w) = -\frac{1}{m} \left[\sum_{i=1}^m (y^{(i)} \log(g_w(x^{(i)})) + (1 - y^{(i)}) \log(1 - g_w(x^{(i)}))) \right]$$

Для сети:

$$J(w) = -\frac{1}{m} \left[\sum_{i=1}^m \sum_{k=1}^K (y_k^{(i)} \log(g_{w,k}(x^{(i)})) + (1 - y_k^{(i)}) \log(1 - g_{w,k}(x^{(i)}))) \right]$$

(Вычисление градиента функции стоимости)

Рассмотрим один элемент обучающего множества. Выполним для него прямое распространение.

Пусть $\delta_j^{(l)}$ - ошибка элемента j в слое l .

Для элементов выходного слоя нейронной сети ($L=4$):

$$\delta_j^{(4)} = a_j^{(4)} - y_j \quad (\text{или в векторной форме } \delta^{(4)} = a^{(4)} - y)$$

$$\delta^{(3)} = (W^{(3)})^T \delta^{(4)} \circ g'(v^{(3)})$$

$$\delta^{(2)} = (W^{(2)})^T \delta^{(3)} \circ g'(v^{(2)})$$

$\delta^{(1)}$ по понятным причинам нет

$$\frac{\partial J(W)}{\partial w_{ij}^{(l)}} = a_j^{(l)} \delta_i^{(l+1)}$$

42

Схема алгоритма:

repeat

$\Delta_{ij}^{(l)} = 0$ (для всех i, j, l). (накопители для вычисления градиента)

for $i = 1$ **to** m (для каждого примера)

$\leftarrow a^{(1)} = x^{(1)}$

Выполнить прямое распространение по сети,
вычислив $a^{(2)}, \dots, a^{(L)}$

$$\delta^{(L)} = a^{(L)} - y^{(m)}$$

Вычислить $\delta^{(L-1)}, \dots, \delta^{(2)}$

$$\Delta_{ij}^{(l)} = \Delta_{ij}^{(l)} + a_j^{(l)} \delta_i^{(l+1)}$$

$$D_{ij}^{(l)} = \frac{1}{m} \Delta_{ij}^{(l)}$$

$$W_{ij}^{(l)} = W_{ij}^{(l)} - \alpha D_{ij}^{(l)}$$

until достигнут критерий останова

Рекомендующие системы. Назначение и классификация.

Рекомендующие системы (РС) – это класс систем поддержки принятия решений, предназначенных для облегчения процесса выбора из множества вариантов.

- 1) РС предназначены для облегчения выбора из однородного множества вариантов;
- 2) Характер этого выбора и критерии его осуществления являются, во многом, субъективными.

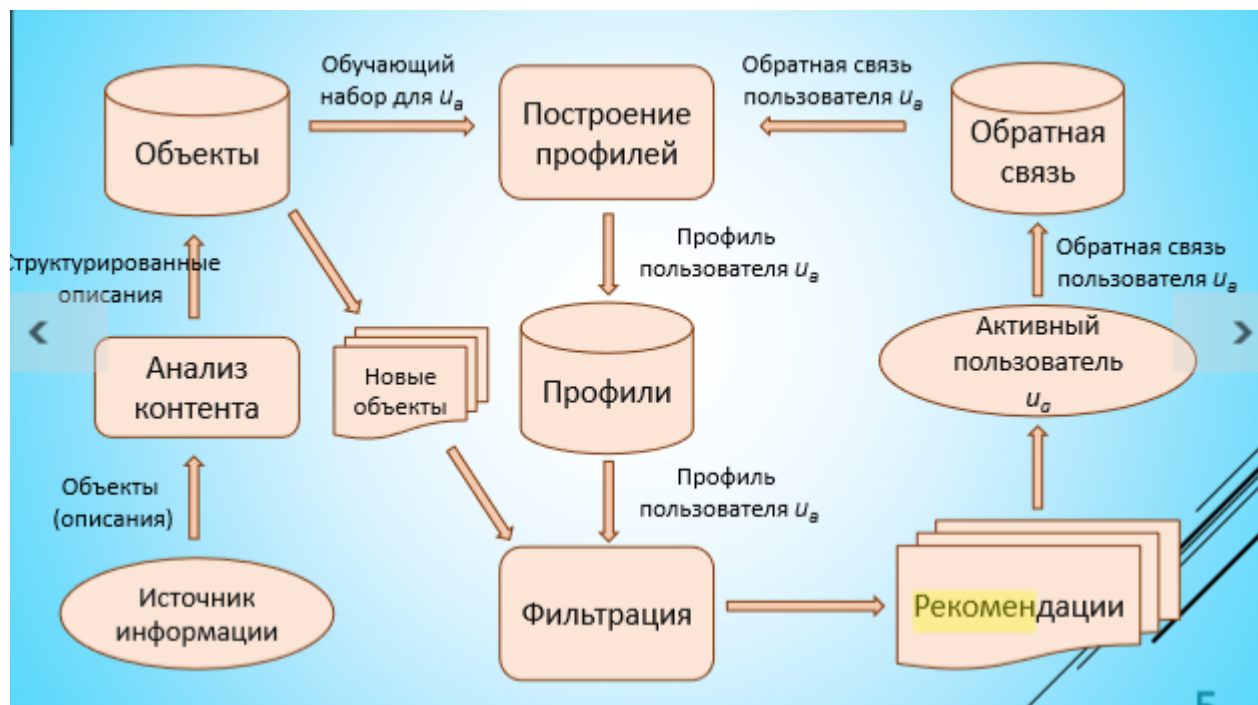
Рекомендующая система, таким образом, нацелена на обобщение и формализацию процесса субъективного выбора.

Классификация

Критерий классификации: вид информации, используемой для формирования рекомендаций.

- 1) Контентные РС (content-based).
- 2) Системы коллаборативной фильтрации (collaborative filtering).
- 3) Демографические (demographic).
- 4) Основанные на знаниях (knowledge-based).
- 5) Социальные (community-based).
- 6) Гибридные (hybrid).

Контентные рекомендующие системы. Общая архитектура. Представление объектов и профиля пользователя. Достоинства и недостатки.



Представление объектов

Зависит от природы объектов, которые собираемся рекомендовать!

Упрощение: работать не с самим содержимым, а с его текстовым описанием.

Широко используются методы информационного поиска (information retrieval): -

- сайты (по содержанию);
- новостные сообщения;
- сообщения электронной почты;
- продукты (по описанию);

Для текстовых объектов/описаний: **векторная модель документа, взвешенная по TF-IDF (и для профилей, и для объектов)**

$D = \{d_1, \dots, d_N\}$ – корпус документов;

$T = \{t_1, \dots, t_n\}$ – словарь (множество слов из корпуса);

Документ представляется в виде $d_j = (w_{1j}, w_{2j}, \dots, w_{nj})$.

$$TFIDF(t_k, d_j) = TF(t_k, d_j) \cdot \log \frac{N}{n_k}$$
$$w_{kj} = \frac{TFIDF(t_k, d_j)}{\sqrt{\sum_{s=1}^{|T|} TFIDF(t_s, d_j)^2}}$$

Вычисление степени сходства между двумя документами (косинусная мера):

$$sim(d_i, d_j) = \frac{\sum_k w_{ki} w_{kj}}{\sqrt{\sum_k w_{ki}^2} \sqrt{\sum_k w_{kj}^2}}$$

Профиль пользователя

Множество объектов T_u , для каждого из которых известно: понравился ли (был ли полезен) пользователю и данный объект: $T_u = \{(X^{(1)}, y^{(1)}), \dots, (X^{(lu)}, y^{(lu)})\}$, $y^{(lu)} \in \{0, 1\}$.

Варианты:

- дерево решений;
- наивный байесов классификатор;
- метод опорных векторов (support vector machine);

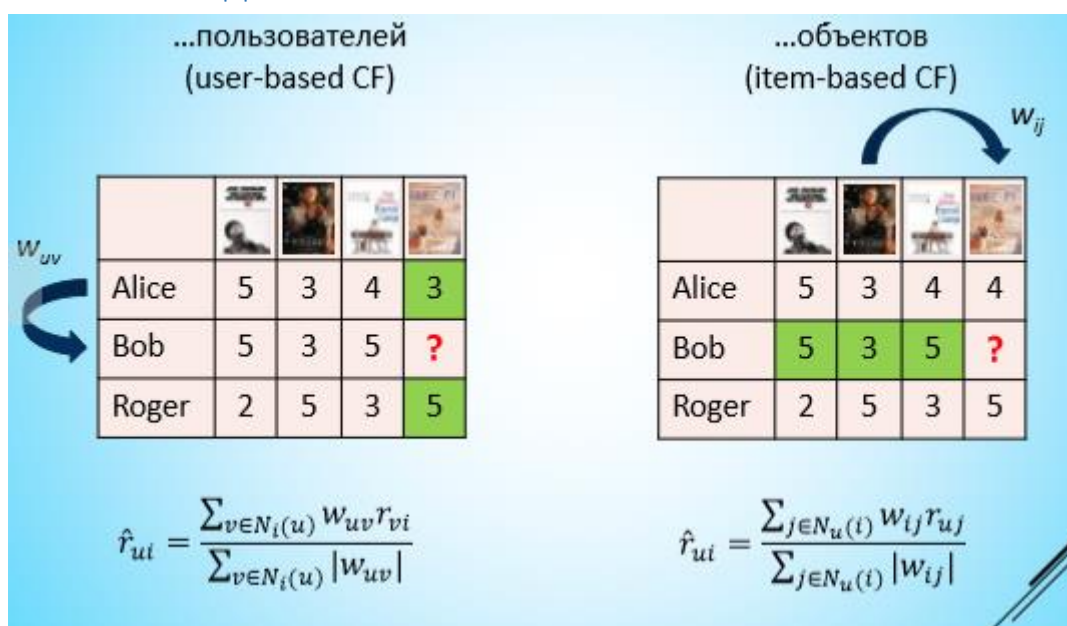
Достоинства:

- отсутствие зависимости от других пользователей
- прозрачность получаемых рекомендаций («похожие объекты»)
- легкость добавления новых объектов

Недостатки:

- ограничения, связанные с анализом контента, сложность при построении моделей объектов
- излишняя специализация (низкие шансы на «неожиданную находку»)
- сложность формирования рекомендаций для нового пользователя

Системы коллаборативной фильтрации. Подход, основанный на оценке сходства. Нормализация, распространенные способы вычисления «сходства». Достоинства и недостатки.



Помимо выбора отношения вида отношения соседства (между объектами или между пользователями), для определения рекомендуемой процедуры необходимо задать:

- 1) Способ нормализации оценок.
- 2) Способ вычисления сходства (объектов или пользователей).
- 3) Способ выбора «ближайших соседей».

Нормализация

Проблема: пользователи могут по-разному использовать шкалу оценок. Например:
Оценки пользователя 1: 10, 10, 8, 9, 10, 10.

Оценки пользователя 2: 8, 8, 5, 6, 7, 8.

Если использовать оценки пользователя 2 в качестве основы для вычисления оценок пользователя 1, то предсказания будут занижены.

$$h(r_{ui}) = r_{ui} - \bar{r}_u$$

Например:





Оценки пользователя 1: 10, 10, 8, 9, 10, 10 \Rightarrow 0.5, 0.5, -1.5, -0.5, 0.5, 0.5.

Оценки пользователя 2: 8, 8, 5, 6, 7, 8 \Rightarrow 1, 1, -2, -1, 0, 1.

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in N_i(u)} w_{uv} (r_{vi} - \bar{r}_v)}{\sum_{v \in N_i(u)} |w_{uv}|}$$

Косинусная мера:

$$cv(u, v) = \frac{\sum_{i \in R(u) \cap R(v)} r_{ui} r_{vi}}{\sqrt{\sum_{i \in R(u)} r_{ui}^2 \sum_{i \in R(v)} r_{vi}^2}}$$





				
Alice	5	3	4	3
Bob	5	3	5	?
Roger	2	5	3	5

$$cv(Alice, Bob) = \frac{5 \cdot 5 + 3 \cdot 3 + 4 \cdot 5}{\sqrt{(5^2 + 3^2 + 4^2 + 3^2) (5^2 + 3^2 + 4^2)}} \approx 0.91525$$

1	0.915	0.853
0.915	1	0.656
0.853	0.656	1

Коэффициент корреляции (Пирсона):

$$pc(u, v) = \frac{\sum_{i \in R(u) \cap R(v)} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in R(u) \cap R(v)} (r_{ui} - \bar{r}_u)^2 \sum_{i \in R(u) \cap R(v)} (r_{vi} - \bar{r}_v)^2}}$$

				
Alice	5	3	4	3
Bob	5	3	5	?
Roger	2	5	3	5

$$cv(Alice, Bob) \approx \frac{2 \cdot 0.67 + 0 \cdot (-1.33) + 1 \cdot 0.67}{\sqrt{(2^2 + 0^2 + 1^2) (0.67^2 + (-1.33)^2 + 0.67^2)}} \approx 0.828$$

1	0.828	-0.986
0.828	1	-0.896
-0.986	-0.896	1

Реализация выбора включает ответ на два вопроса:

- 1) определение того, какие степени сходства, которые будут сохраняться рекомендующей системой (user-based – m2, item-based – n2);
 - 2) фактический выбор «ближайших соседей» среди сохраненных.
- Эвристики, применяемые при хранении «соседей»:
- 1) Лучшие N (N >= K).
 - 2) Превышающие определенный заранее заданный порог.
 - 3) Сигнализирующие о сходстве пользователей (а не о различии). (или их комбинация)

Достоинства:

- универсальность, отсутствие необходимости анализа и моделирования контента
- высокая точность рекомендаций
- шансы найти неожиданный незнакомый объект выше, чем в контентных системах

Недостатки:

- варианты проблемы «холодного старта» (новый пользователь, новый объект)
- сложности с объяснением рекомендаций
- возможность мошенничества/атак

Системы коллаборативной фильтрации. Подход, основанный на моделировании латентных факторов. Достоинства и недостатки.

Отображение и пользователей, и объектов в единое пространство латентных факторов размерности f таким образом, чтобы исходные данные (оценки, полученные от пользователей), можно было восстановить с помощью операции скалярного произведения соответствующих векторов в пространстве латентных факторов.

Т.е. переход от множества рейтингов, к множеству векторов следующего вида: - пользователи – $p_u \in \mathbb{R}^f$ (в какой мере пользователю близки (интересны) объекты, обладающие каждым из факторов);

- объекты – $q_i \in \mathbb{R}^f$ (в какой мере объект обладает каждым из факторов).

Оценка объекта i пользователем u представляется как скалярное произведение $q_i^T p_u$.

Идея метода:

1) Найти значения векторов p_u и q_i , используя для этого известные оценки r_{ui} . 2) Использовать найденные значения этих векторов для вычисления предсказаний неизвестных оценок \tilde{r}_{ui} .

(Вариант интерпретации: $R = PQ^T$ - разложение (факторизация) матрицы.)

Достоинства:

- универсальность, отсутствие необходимости анализа и моделирования контента
- высокая точность рекомендаций
- шансы найти неожиданный незнакомый объект выше, чем в контентных системах

Недостатки:

- варианты проблемы «холодного старта» (новый пользователь, новый объект)
- сложности с объяснением рекомендаций
- возможность мошенничества/атак

ОЦЕНКА КАЧЕСТВА РЕКОМЕНДАЦИЙ (1)

Классический способ для систем, предсказывающих оценки:

$$RMSE = \sqrt{\frac{1}{|K|} \sum_{(u,i) \in K} (\hat{r}_{ui} - r_{ui})^2}$$

Классический способ для систем, формирующих список рекомендаций (не предсказывающих оценки):

$$Precision = \frac{\#tp}{\#tp + \#fp}$$

$$Recall = \frac{\#tp}{\#tp + \#fn}$$

	Рекомендован	Не рекомендован
Полезен	tp	fn
Бесполезен	fp	tn

- Измерять нужно то, что соответствует цели
- Цели внедрения рекомендующих систем... только ли удобство пользователей?
- Более сложные и «жизненные» меры качества в новых (экспериментальных) поколениях РС

