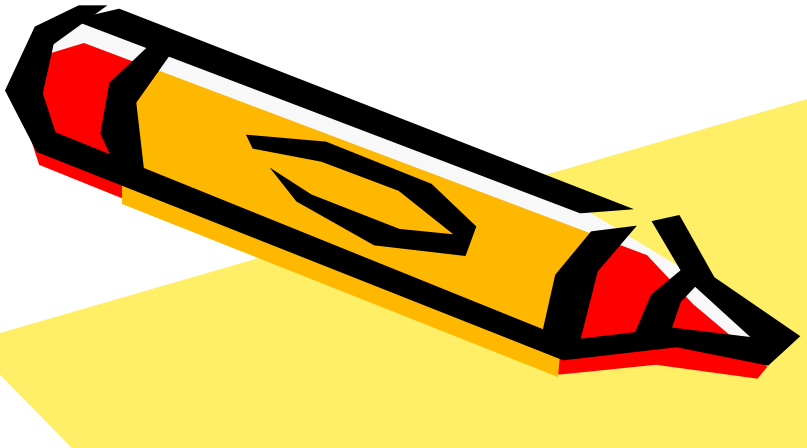


Системы искусственного интеллекта

Бондаренко Иван Юрьевич,
ассистент каф. ПМИ

Автоматическое приобретение знаний из БД

Автоматическое приобретение знаний из баз данных – это методы и технологии выявления компьютером скрытых правил и закономерностей в больших наборах данных.

Синонимы: **Data Mining** («добыча» или «раскопка» данных), **Knowledge Discovery in Databases** (обнаружение знаний в базах данных), **интеллектуальный анализ данных**.

Актуальность автоматического приобретения знаний

В 2002 году, согласно оценке профессоров из ун-та Berkeley, объём информации в мире увеличился на $5 \cdot 10^{18}$ = 5 000 000 000 000 000 000 байт!

Согласно другим оценкам, информация удваивается каждые 2 – 3 года.

В 1989 году большая БД – это БД объёмом 1 мегабайт.

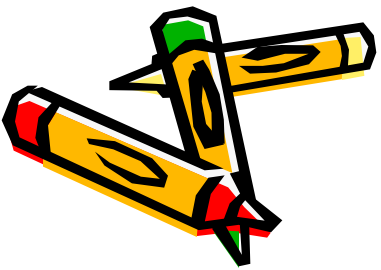
В 2003 году большая БД – это БД объёмом 1 петабайт (примерно в миллион раз больше).

Области применения автоматич. приобретения знаний

1. Розничная торговля
   - анализ покупательской корзины;
   - исследование временных шаблонов;
   - создание прогнозирующих моделей.
2. Банковское дело
   - Выявление мошенничества с кредитными карточками;
   - сегментация клиентов;
   - прогнозирование изменений клиентуры.

Области применения автоматич. приобретения знаний

3. Телекоммуникации
   - анализ записей о подробных характеристиках вызовов;
   - выявление лояльности клиентов.
4. Страхование
   - выявление мошенничества;
   - разработка продуктов;
   - анализ риска.



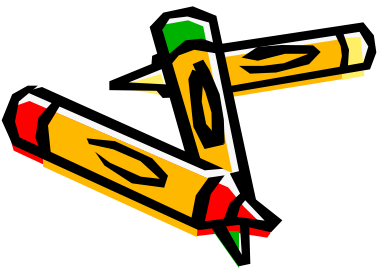
Области применения автоматич. приобретения знаний

5. Другие приложения в бизнесе
   - сегментация рынка;
   - развитие автомобильной промышленности;
   - поощрение часто летающих клиентов.
6. Медицина

   автоматизация создания баз знаний медицинских ЭС (вместо врачей-экспертов – медицинская база данных).

Типы закономерностей, извлекаемых из БД

- Ассоциация;
- Классификация;
- Кластеризация;
- Прогнозирование.



Методы автоматического приобретения знаний

1. Статистические методы
2. Нейронные сети
3. Рассуждения на основе аналогичных случаев
4. Деревья решений
5. Генетические алгоритмы

Статистические методы

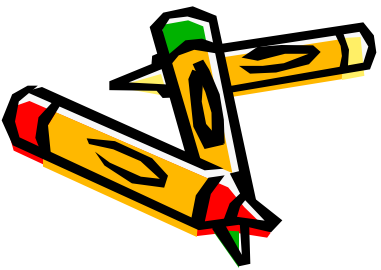
Корреляционный, регрессионный, факторный анализ и др.

Преимущества: классические методы с развитым математическим аппаратом.

Недостатки:

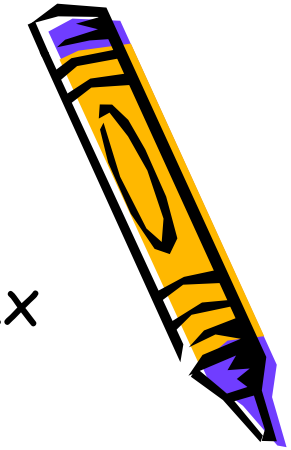
– требуют спец. подготовки пользователя;

– усреднённые характеристики выборки, используемые в статистической парадигме, при исследовании сложных феноменов предметной области часто оказываются фиктивными величинами.



Инструментальные системы: SAS (SAS, США), STATISTICA (StatSoft, США), SPSS Statistics (SPSS, США).

Нейронные сети



Моделируют структуру нервной системы (множество параллельно работающих простых элементов – нейронов – объединённых взвешенными связями).

Преимущества:

– аппроксимация сложных нелинейных зависимостей;

– адаптивность;

– эффективная аппаратная реализуемость.

Недостатки:

– большой объём обучающей выборки;

– плохая интерпретируемость обученной нейронной сети человеком.

Инструментальные системы: BrainMaker (CSS), NeuroShell (Ward Systems Group), OWL (HyperLogic).

Рассуждения на основе аналогичных случаев

Синонимы: Case Based Reasoning, рассуждения по прецедентам, метод ближайшего соседа. **Идея:** для выбора правильного решения в базе находятся близкие аналоги наличной ситуации и выбирается ответ, который был правильным для них.

Преимущества: простота реализации и наглядность результатов анализа.

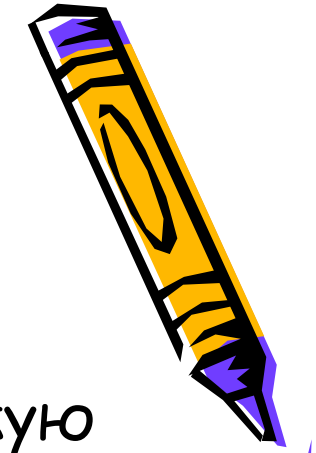
Недостатки:

– не строятся модели или правила, обобщающие предыдущий опыт;

– сложность выбора адекватной меры близости прецедентов.

Инструментальные системы: KATE tools (Acknosoft, Франция), Pattern Recognition Workbench (Unica, США).

Деревья решений



Деревья решений (Decision Trees) – один из самых популярных методов автоматического извлечения знаний. Они создают иерархическую структуру классифицирующих правил типа «ЕСЛИ... ТО...», имеющую вид дерева.

Преимущества: наглядность и понятность.

Недостатки:

– проблема значимости;

– проблема независимости признаков.

Инструментальные системы: See5/C5.0 (RuleQuest, Австралия), Clementine (Integral Solutions, Великобритания), SIPINA (University of Lyon, Франция), IDIS (Information Discovery, США).

Генетические алгоритмы

Моделирование механизма наследственности, изменчивости и отбора в живой природе.

Идея. Создаётся исходный набор (*популяция*) комбинаций элементарных логических высказываний (*хромосом*) и определяются *функции приспособленности* для индивидуумов, заданных хромосомами. Популяция обрабатывается с помощью процедур *скрещивания* и *мутации*. В ходе работы процедур на каждой стадии эволюции получаются популяции со всё более совершенными индивидуумами.

Генетические алгоритмы (окончание)

Преимущества:

– пригодность для поиска в сложном пространстве решений большой размерности;

– эффективная аппаратная реализация.

Недостатки:

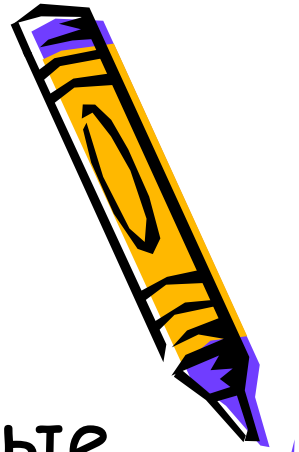
– функции приспособленности и процедуры генетического алгоритма являются эвристическими;

– как и в реальной жизни, эволюцию может «заклинить» на непродуктивной ветви.

Инструментальные системы:

GeneHunter (Ward Systems Group).

Алгоритм индуцирования знаний из БД



Алгоритм генерирует продукционные правила.

В алгоритме используется представление знаний в виде деревьев решений.

Рассмотрим пример.

Пусть необходимо построить базу знаний для получения ответа: «Как поступить, чтобы прибыль росла?».

Исходная база данных, из которой извлекаются знания

ПРИБЫЛЬ	ВОЗРАСТ	КОНКУ-РЕНЦИЯ	ТИП
падает	старый	нет	ПО
падает	средний	есть	ПО
растёт	средний	нет	ЭВМ
падает	старый	нет	ЭВМ
растёт	новый	нет	ЭВМ
растёт	новый	нет	ПО

Окончание на следующем слайде…

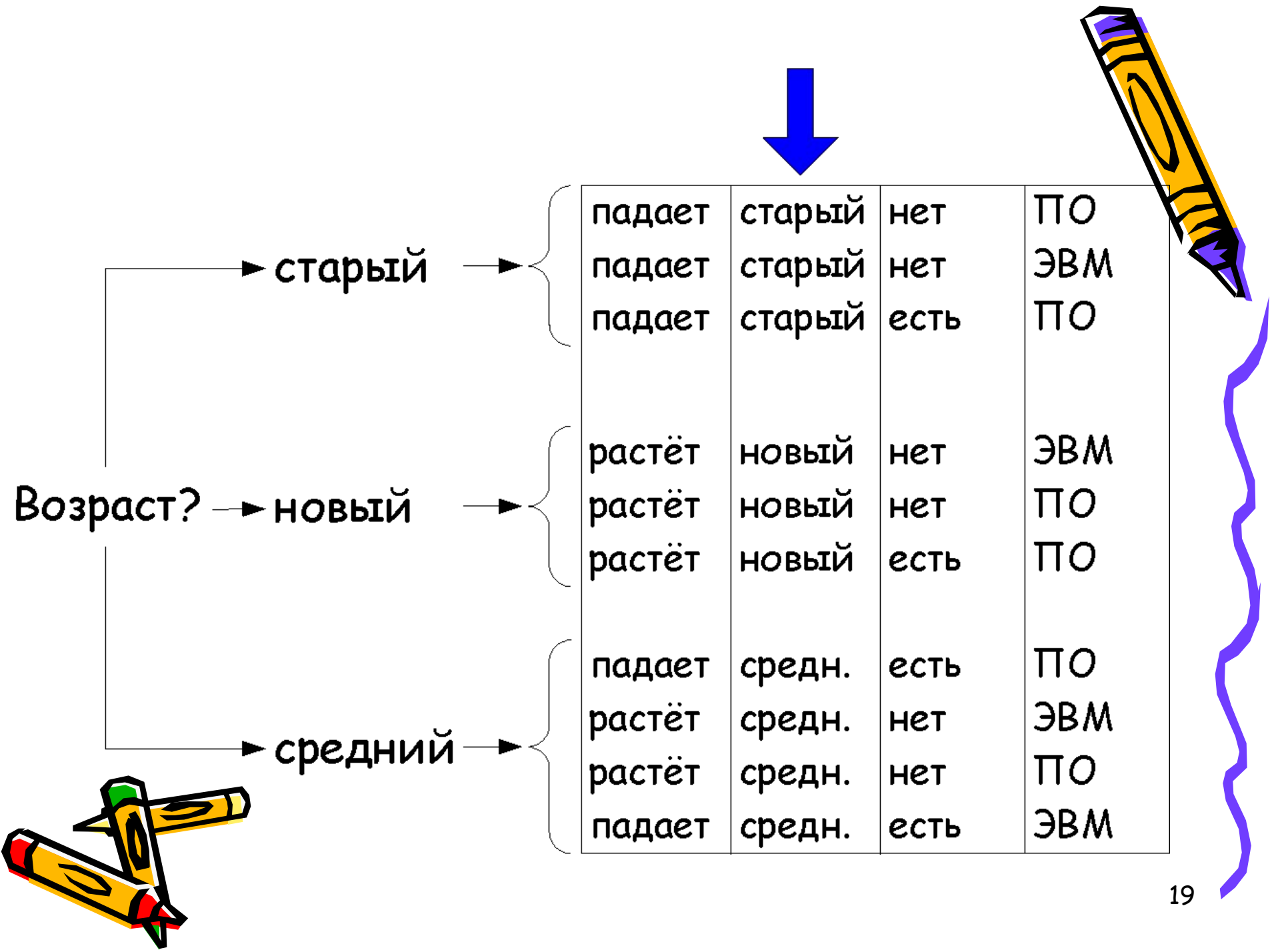
Исходная база данных, из которой извлекаются знания (окончание)

ПРИБЫЛЬ	ВОЗРАСТ	КОНКУ-РЕНЦИЯ	ТИП
растёт	средний	нет	ПО
растёт	новый	есть	ПО
падает	средний	есть	ЭВМ
падает	старый	есть	ПО

Искомый атрибут «Прибыль» будем называть *атрибутом класса*.

Для построения дерева решений нужно взять один из атрибутов таблицы в качестве *основного (корневого) атрибута*. Пусть это будет «Возраст».

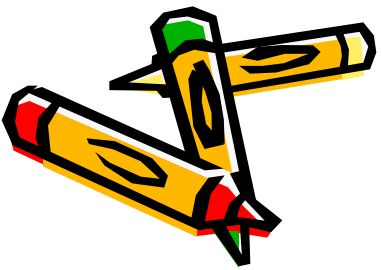
Преобразуем исходную таблицу к следующему виду:


Возраст?
старый
новый
средний
падает старый нет ПО
падает старый нет ЭВМ
падает старый есть ПО
растёт новый нет ЭВМ
растёт новый нет ПО
растёт новый есть ПО
падает средн. есть ПО
растёт средн. нет ЭВМ
растёт средн. нет ПО
падает средн. есть ЭВМ

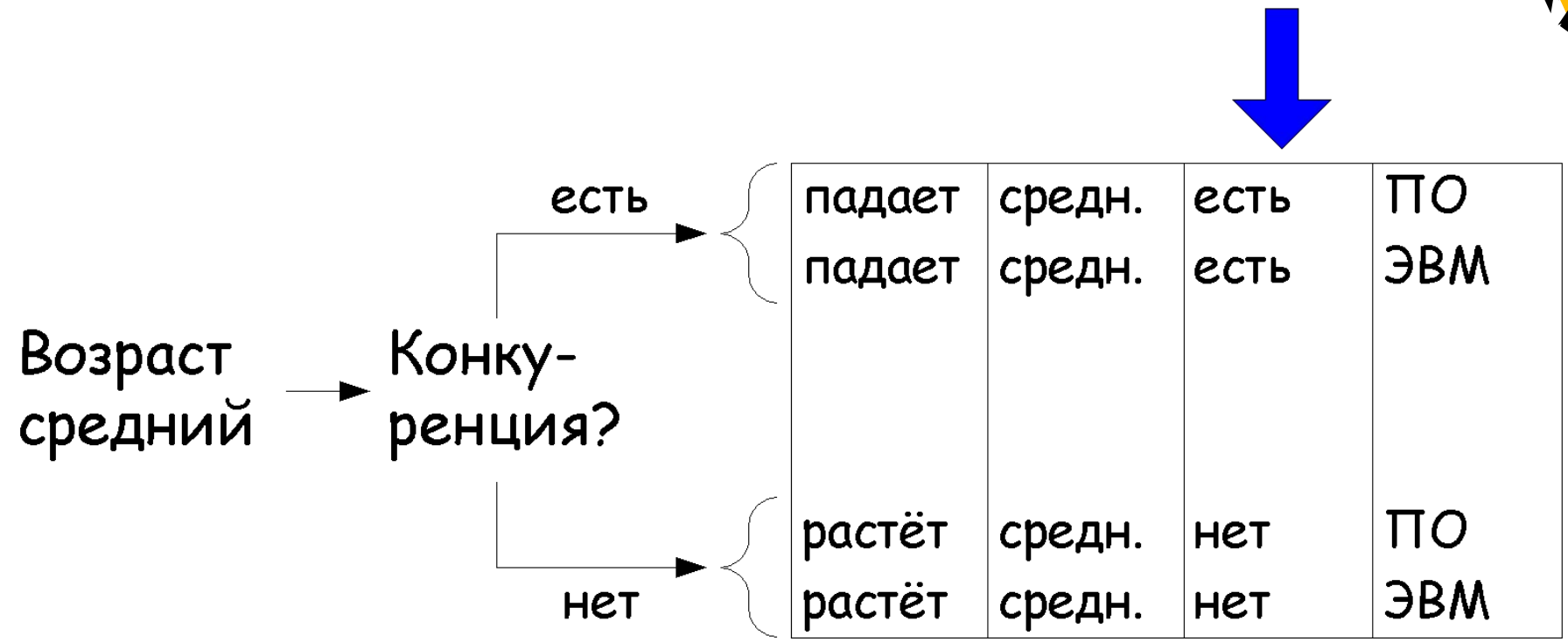
Из таблицы видно, что при значении атрибута *«Возраст»*, равном *«новый»*, прибыль всегда *растёт*, а при значении *«старый»* – *падает*.

В случае же значения *«средний»* такого определённого вывода сделать нельзя.

Поэтому продолжим разбивку таблицы по атрибуту *«Конкуренция»*.



Получим другую таблицу:



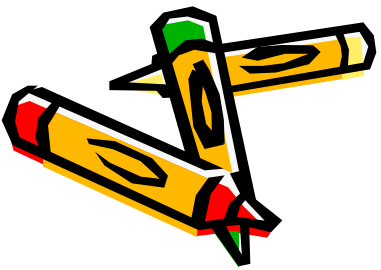
падает	средн.	есть	ПО
падает	средн.	есть	ЭВМ
растёт	средн.	нет	ПО
растёт	средн.	нет	ЭВМ

Поскольку теперь для атрибута класса наше дерево решений выводит однозначный ответ, то дерево решений построено.

Порождаем правила:

1. **ЕСЛИ** Возраст = новый
ТО Прибыль = растёт

2. **ЕСЛИ** Возраст = старый
ТО Прибыль = падает



3. **ЕСЛИ** Возраст = средний
И Конкуренция = нет
ТО Прибыль = растёт

4. **ЕСЛИ** Возраст = средний
И Конкуренция = есть
ТО Прибыль = падает