

人工智能概论-机器学习算法

赵亚伟

zhaoyw@ucas.ac.cn

中国科学院大学 大数据分析技术实验室

2018.6.23

目录

- 概述
- 常见机器学习技术
 - 归纳分析
 - 分类分析
 - 聚类分析
 - 异常分析
 - 关联分析
- 机器学习的局限性
- 数据挖掘语言：DMX简介

三路研究大军

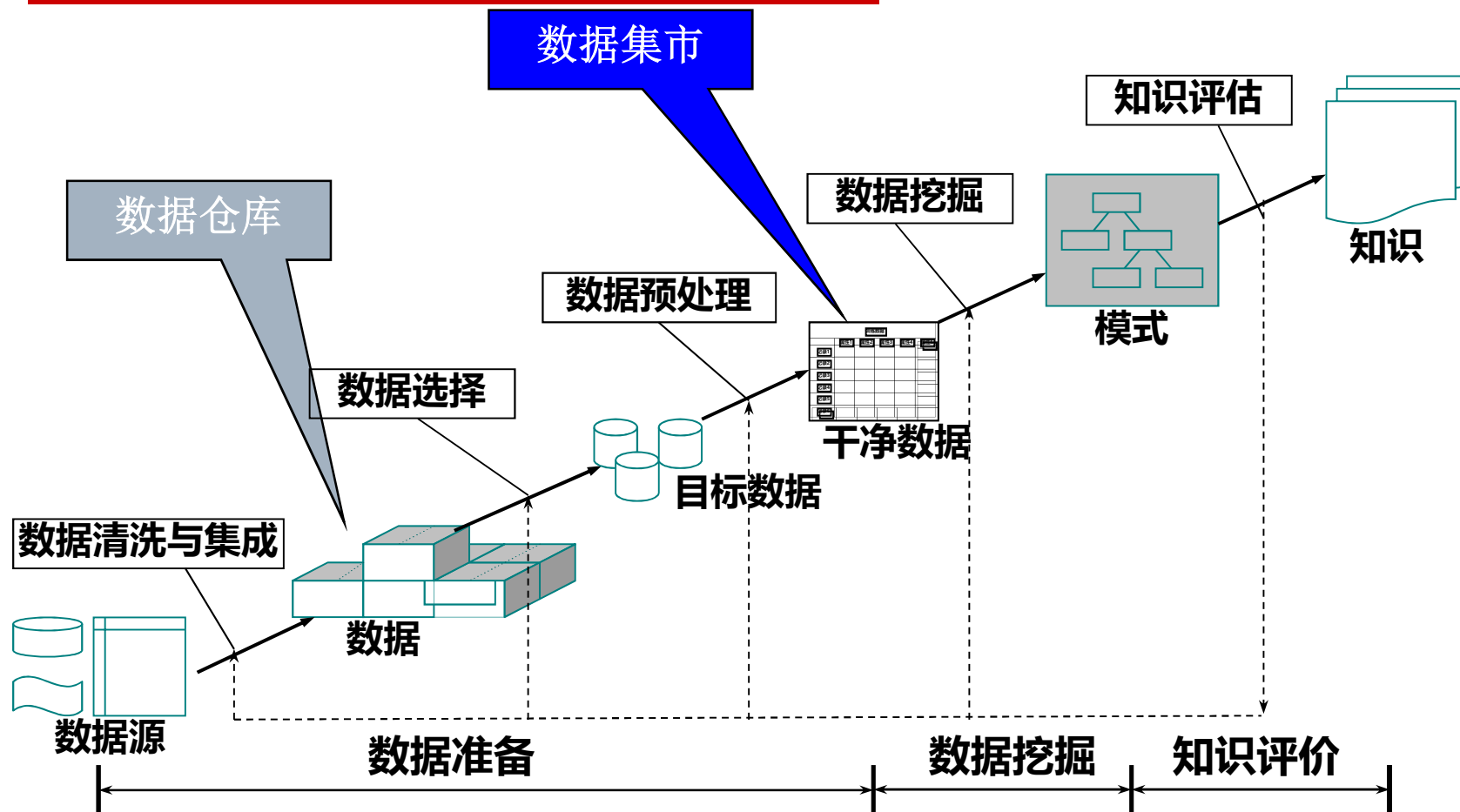
- **AI**：主要是机器学习，提高智能性，**给思想**
- **数据库**：有了数据仓库，何不再深入一步——数据挖掘？**给环境、给方法**
- **统计**：一直做分类、聚类等统计工作，没想到还有这么多的用途！何不占一块地盘？**给方法**

- 在没有和数据库结合之前，**AI**几近穷途末路，统计也几近腐朽，与数据库结合之后，焕发了青春活力
- **原因**：数据库系统是用出来的，**AI**、统计是研究出来的，应用最具活力，还是那句老话“学以致用”、以用促学！

目录

- 概述
- 常见机器学习技术
 - 归纳分析
 - 分类分析
 - 聚类分析
 - 异常分析
 - 关联分析
- 机器学习的局限性
- 数据挖掘语言：DMX简介

机器学习过程模型

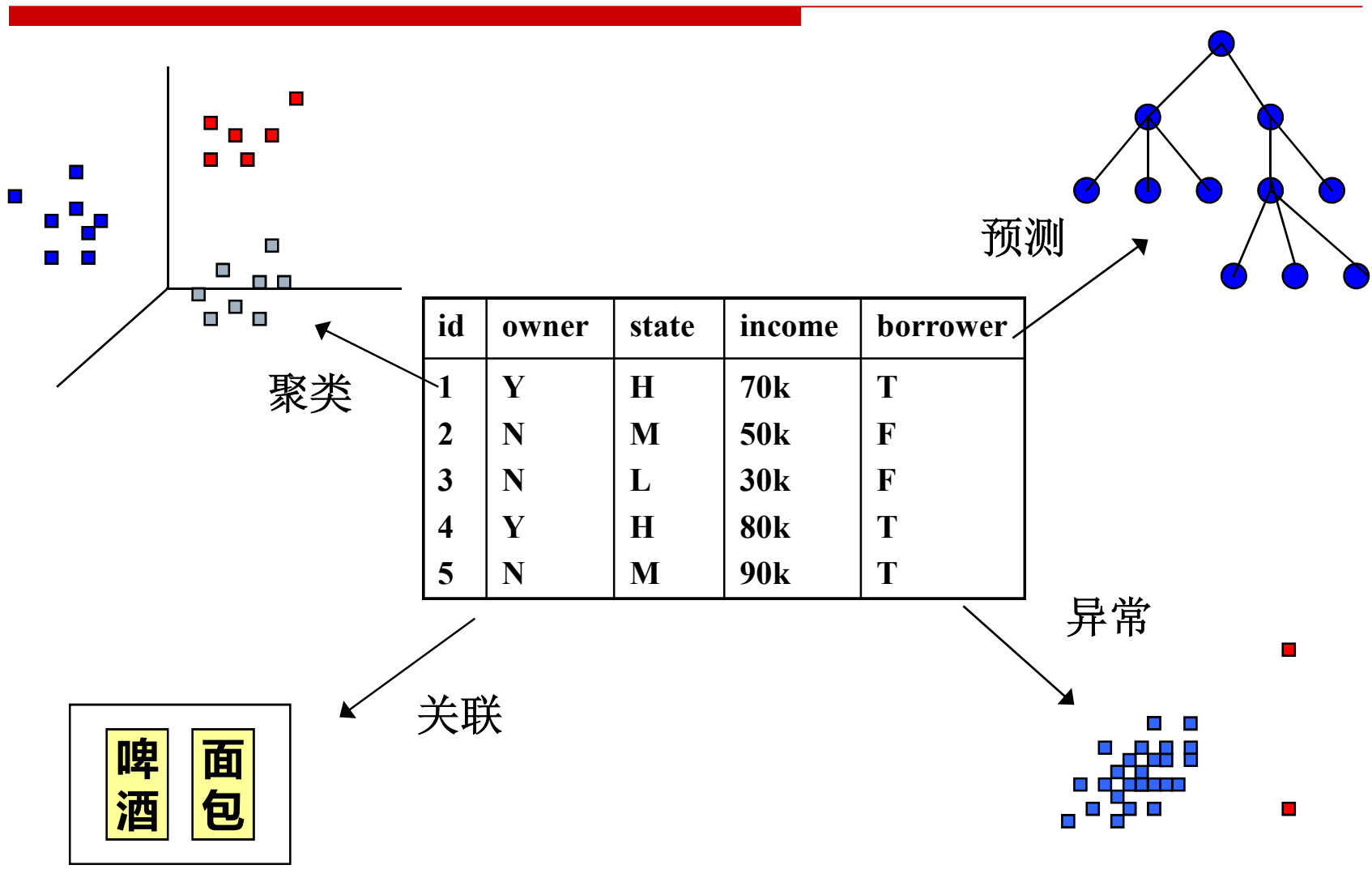


应用

- ❑ **OLTP**: 打工仔用, 看眼前, 现在的库存还有多少件T恤, 要不要进货? 货不足, 进货!
- ❑ **OLAP**: 暴发户老板用, 看过去, 上个月卖了多少钱? 哇, 赚这么多, 下月还这么干!
- ❑ **机器学习**: 有知识的老板用, 看将来, 下个月怎么干? 原来啤酒和尿布还有关系 (沃尔玛的一个传奇故事), 下月放一起卖!

常用机器学习技术

- **归纳分析**: 由特殊到一般(**Cube**, 主因素, 统计)
- **分类分析**: 由已知类别进行分类(决策树, **Bayes**, 神经网络, 关联, **k**-近邻等)
- **预测分析**: 通过回归或贝叶斯模型等实现, 本质是分类
- **聚类分析**: 对数据分类(系统聚类, **k**-平均)
- **异常分析**: 找出异常数据, 本质是聚类
- **购物篮分析**: 发现关联模式(**Apriori**, 聚类)
- **序列模式分析**: 发现高频率模式



目录

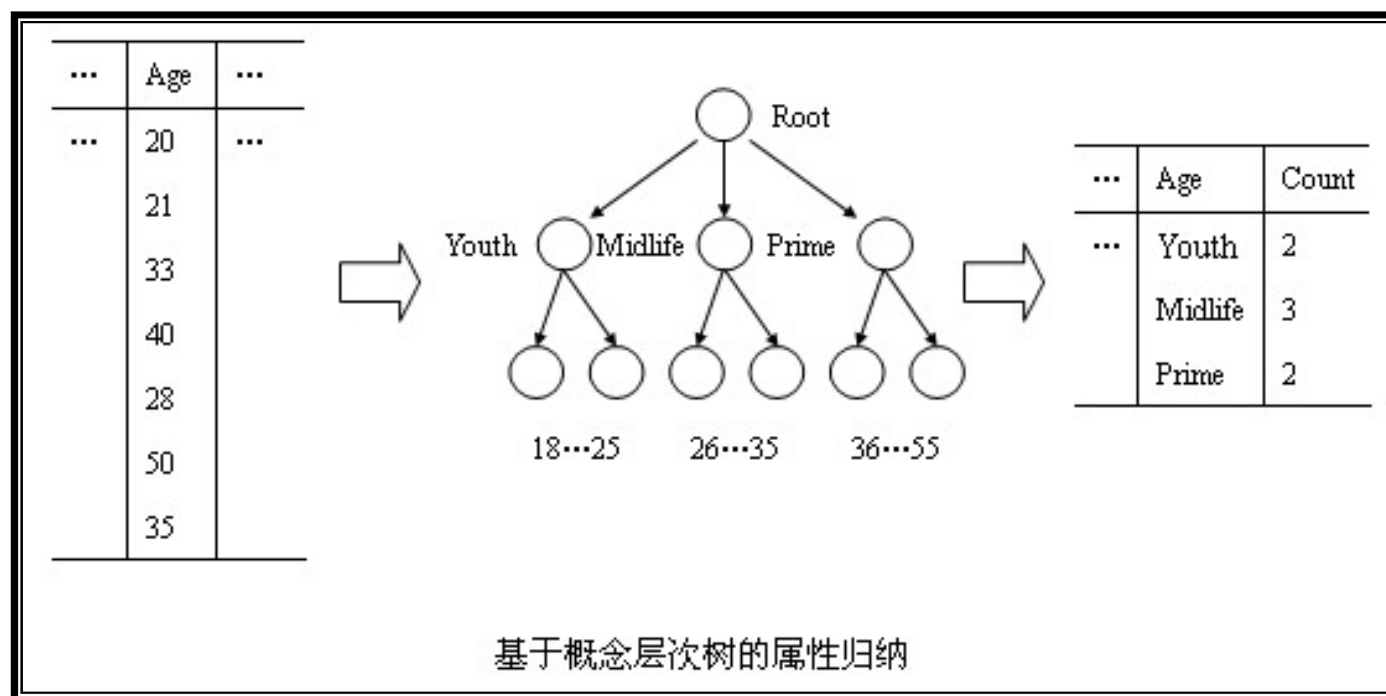
- 概述
- 常见机器学习技术
 - 归纳分析
 - 分类分析
 - 聚类分析
 - 异常分析
 - 关联分析
- 机器学习的局限性
- 数据挖掘语言：DMX简介

归纳分析

- 数据的 **类/概念** 称为**概念描述（模式）**，显然，概念模式是一种对现实世界实体的归纳，发现概念模式的过程称为**归纳分析**。
- 常见类型：
 - **特征化**：汇总，饼图、条图、曲线、**Cube**等
 - **区分**：与一个或多个类比较，类似特征化输出，单输出的比较值，如饼图的一部分表示的是销售增加比例
- 常见的归纳分析方法
 - **基于属性归纳**
 - **属性相关分析**
 - **趋势分析**
 - **分布分析**

例子：基于属性归纳

- 通过概念层次树完成属性泛化，将处理过的数据集中相同内容的数据行进行合并，以获得一个更加泛化的关系表。



目录

- 概述
- 常见机器学习技术
 - 归纳分析
 - 分类分析
 - 聚类分析
 - 异常分析
 - 关联分析
- 机器学习的局限性
- 数据挖掘语言：DMX简介

分类分析

- 假定真理在训练数据中，从其中找出分类规则。
- 找出区分数据的规则（分类模型），以便能够预测未知的数据，这个模型可以称为“分类器”。如何构造分类器？用训练数据集训练一下，故也称有指导的学习。注意：假定真理在训练数据中，训练数据要选好！
- 如何通过训练数据得到分类模型？通过一些算法：
 - 如果规则为决策树，则用决策树算法；
 - 如果规则为贝叶斯模型，则用贝叶斯算法；
 - 如果规则为神经网络，则用神经网络算法；
 - ...

分类器

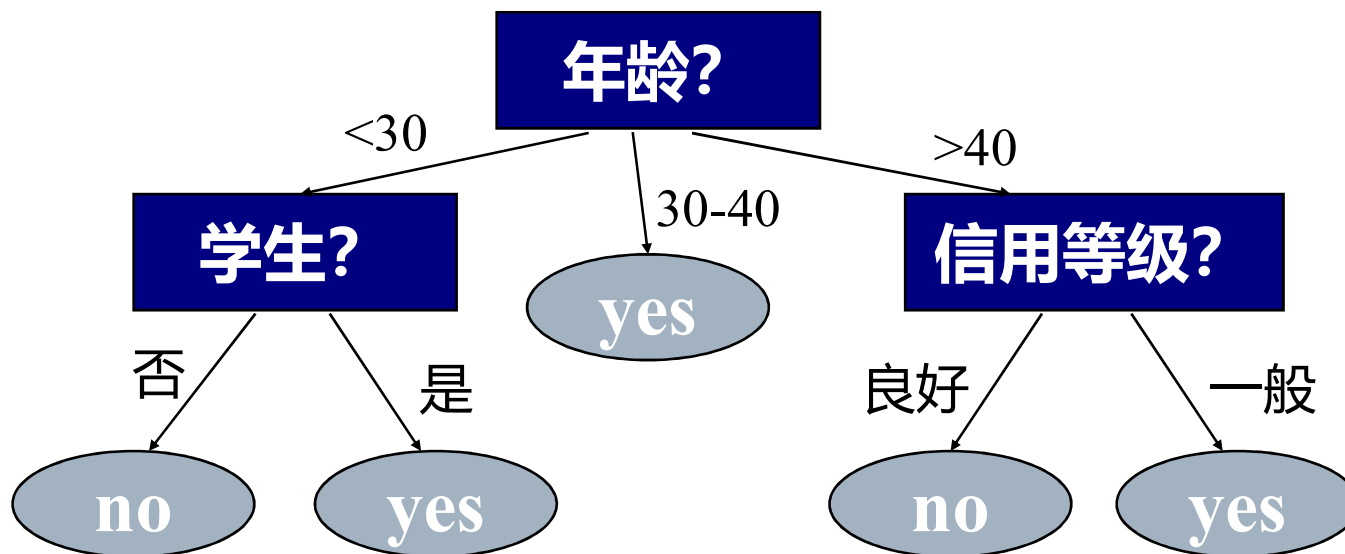
- 常见的分类器：决策树、贝叶斯模型、神经网络、SVM（支持向量机）等
- 不同的分类器，分类效果不同。生活中，选栗子，分大、中、小三种，用手选尺量，用筛子，用称量，...，不同的方法效率效果不同。
- 分类器可以用于预测，如果有了筛子，现在给定一个栗子，就可以确定这个栗子是大的、中的还是小的。
- 如果有了小偷分类器，可以预测来的这个人（未见过的）是不是小偷。😊

构造分类器：分类算法

- 手选尺量分栗子：假设事先有大、中、小栗子3个（**训练数据**），尺子分类量出大（直径**3cm**）、中（直径**2cm**）、小（直径**1cm**），确定**>3cm**的为**大栗子**、**1cm**至**3cm**间的为**中栗子**，**<1cm**为**小栗子**。分类器构建完毕。
- 上述为生活中分类器的算法，分类算法的输入为训练数据，输出为分类器，如决策树、贝叶斯模型、神经网络等。
- 下面举例说明**决策树**与**贝叶斯模型**。

决策树

- 所谓决策树就是一个类似流程图的**树型结构**，其中树的每个内部**节点代表**对一个**属性（取证）**的测试，其**分支**就代表测试的每个**结果**

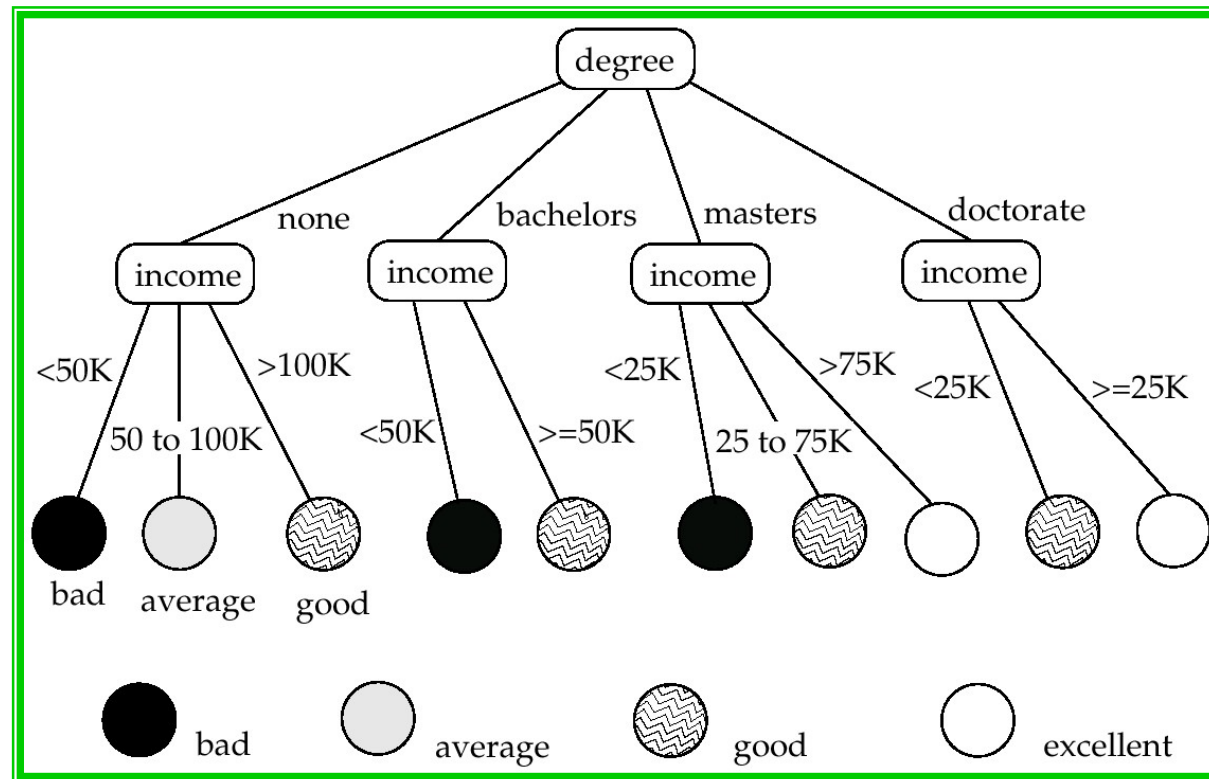


例子：决策树

- 某人申请汽车保险，申请人算哪一类（风险类别）？
- 规则可能如下：
 - $\forall \text{ person } P, P.\text{degree} = \text{masters and } P.\text{income} > 75,000$
 $\Rightarrow P.\text{credit} = \text{excellent}$
 - $\forall \text{ person } P, P.\text{degree} = \text{bachelors and}$
 $(P.\text{income} \geq 25,000 \text{ and } P.\text{income} \leq 75,000)$
 $\Rightarrow P.\text{credit} = \text{good}$
- 规则可能不一定精确，上述规则可表达为决策树
- 这些规则哪儿来的？用训练数据训练出来的，如何通过训练数据得到决策树？分类算法

例子：一棵决策树

□ 可以进行分类，又称分类树



如何构建决策树?

计算公式

- 对给定数据对象进行分类所需要的信息量（根据类别属性）

$$I(s_1, s_2, \dots, s_m) = - \sum_{i=1}^m \frac{s_i}{S} \log_2 \frac{s_i}{S}$$

- 利用属性A划分当前集合所需要的信息熵， $E(A)$ 越小，表示其子集划分结果越纯

$$E(A) = \sum_{j=1}^V \frac{s_{1j} + s_{2j} + \dots + s_{mj}}{S} I(s_{1j}, \dots, s_{mj})$$

- 确定给定子集 S_j 的信息量（继续划分，与第一个同）

$$I(s_{1j}, s_{2j}, \dots, s_{mj}) = - \sum_{i=1}^m p_{ij} \log(p_{ij})$$

$$\text{其中, } p_{ij} = \frac{s_{ij}}{|S_j|}$$

-
- 利用属性A对当前分支节点进行相应样本集合划分所获得的信息增益

$$Gain(A) = I(s_1, s_2, \dots, s_m) - E(A)$$

- 根据 $Gain(A)$ 确定节点，选最大值作为节点
- 重复上述步骤，直至所有属性分析完成
- **基本思想**：上述过程简单地说就是通过熵确定决策树中的各个节点，熵越小越有规律，越靠近根节点，节点定了树就定了。

分类，聚类等操作 都是从无序到有序的过程，是减少熵的过程，信息增益 是减少的熵的量度

例子

- ❑ 某商场顾客数据集
- ❑ 根据该数据集构建决策树
- ❑ 类别（预测）属性为“BAYS_COMPU”

ID	AGE	INCOME	STUDENT	CREDIT_RAT	BAYS_COMPU
1	<30	High	no	Fair	no
2	<30	High	no	Excellent	no
3	30-40	High	no	Fair	yes
4	>40	Medium	no	Fair	yes
5	>40	Low	yes	Fair	yes
6	>40	Low	yes	Excellent	no
7	30-40	Low	yes	Excellent	yes
8	<30	Medium	no	Fair	no
9	<30	High	yes	Fair	yes
10	>40	Medium	yes	Fair	yes
11	<30	Medium	yes	Excellent	yes
12	30-40	Medium	no	Excellent	yes
13	30-40	Medium	yes	Fair	yes
14	>40	Low	no	Excellent	no

计算过程

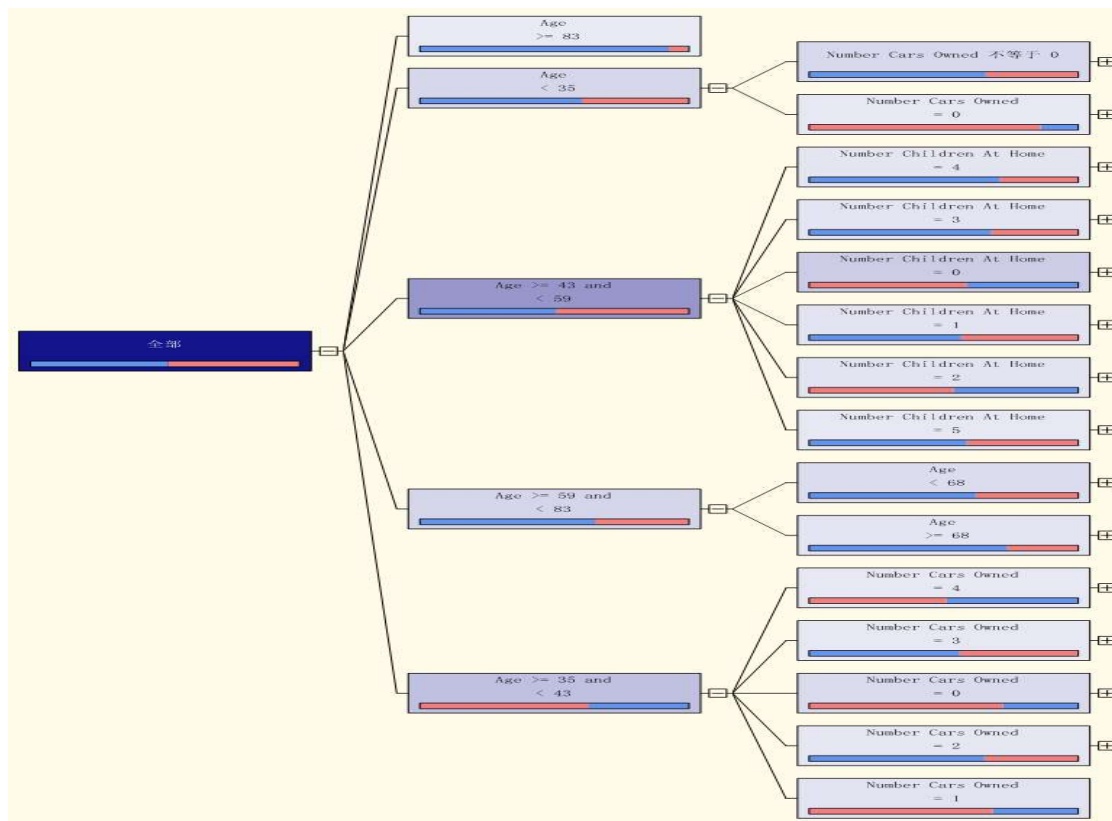
- 样本集合的类别属性为“**BAYS_COMP**”，该属性有两个不同取值，即{yes, no}，因此有两个类别，即 $m=2$ ，设s1对应yes类别，s2对应no类别。s1类别包含9个样本，s2包含5个样本，计算每个属性的信息增益。
- 首先，计算分类所需的信息量：
 - $I(s1,s2) = -9/14\log 9/14 - 5/14\log 5/14 = 0.94$
- 其次，计算每个属性信息熵，假设从“AGE”开始：

-
- 对于AGE=“<30”; $s_{11}=2, s_{21}=3, I(s_{11}, s_{21}) = 0.971$ (AGE=“<30”有2个BAYS_COMPU=‘yes’记录,即 $s_{11}=2$, 有3个, BAYS_COMPU=‘no’记录, $s_{21}=3$)
 - 对于AGE=“30-40”; $s_{12}=4, s_{22}=0, I(s_{12}, s_{22}) = 0$
 - 对于AGE=“>40”; $s_{13}=3, s_{23}=2, I(s_{13}, s_{23}) = 0.971$
 - 第三, 计算按AGE划分所需要的信息熵:
 - $E(\text{AGE}) = 5/14 * I(s_{11}, s_{21}) + 4/14 * I(s_{12}, s_{22}) + 5/14 * I(s_{13}, s_{23}) = 0.694$
 - 第四, 计算这种划分的信息增益:
 - $\text{Gain}(\text{AGE}) = I(s_1, s_2) - E(\text{AGE}) = 0.94 - 0.694 = 0.246$
 - 同样计算 $\text{Gain}(\text{INCOME}) = 0.029, \text{Gain}(\text{STUDENT}) = 0.151, \text{Gain}(\text{CREDIT_RAT}) = 0.048$
 - 由于Gain(AGE)最大, AGE的熵最小, 与BAYS_COMPU的对应最有规律, 因此, AGE为第一个节点
 - 基于AGE继续划分, 重复上述步骤, 直至划分结束

精度问题

- 训练不能**过度**，太精确就不好用了。
- **生活中**，9:00钟上班，9:05前不算迟到，9:05之后可以延长到9:30，算善意迟到，每月有三次机会，超过三次算真正迟到，扣奖金！这已经很人性化了。
- 当属性多、元组多时分类器训练容易过度，需要进行必要的修正，**不十分精确但好用，十分精确但不好用**。
- 分类器的分类效果可以用**精度**计算，用测试数据测一下，100个里面有几个分错了，就可以计算出精度了。
- 所以，分类器不见得所有的都能分对，允许错，但不能经常“善意”，经常“善意”说明有问题。
- **生活中**，警察破案，用侦查手段（分类器），有犯人是冤枉的，分类错误，但可用，若分类过于精确，一丝一毫不能差，漏网之鱼会更多，导致分类器不可用。

SSAS中生成的决策树



**典型的基于决策树的算法有
ID3和C4.5(略)**

贝叶斯分类算法

□ 贝叶斯定理

$$P(A_i | B) = \frac{P(B | A_i)P(A_i)}{P(B)} = \frac{P(B | A_i)P(A_i)}{\sum_{j=1}^n P(B | A_j)P(A_j)}, i = 1, 2, \dots, n$$

条件概率公式

全概率公式

事后概率：B
发生了，A发
生的概率

事前概率

A_i : 类别属性
 B : 属性集（不含 A_i ）
 b_k : 已知属性

$$P(A_i | B) = \frac{(\prod_{k=1}^m P(b_k | A_i))P(A_i)}{P(B)}, i = 1, 2, \dots, n; k = 1, 2, \dots, m$$

各属性相互独立

我们应用的公式

$P(b_k | A_i)$ 表示一条记录的一个已知属性 b_k 的某个取值决定该记录隶属于某一个类别 A_i 的概率。

贝叶斯公式（自学）

- 条件概率公式、全概率公式→贝叶斯公式

- 条件概率公式

$$P(A|B) = \frac{P(AB)}{P(B)}$$

$$\rightarrow P(AB) = P(A|B)P(B)$$

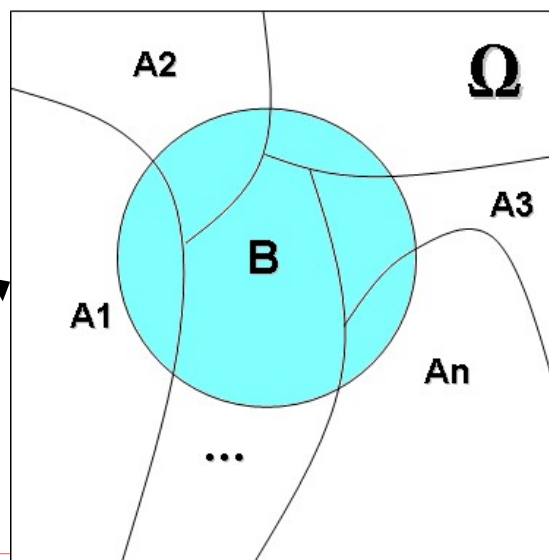
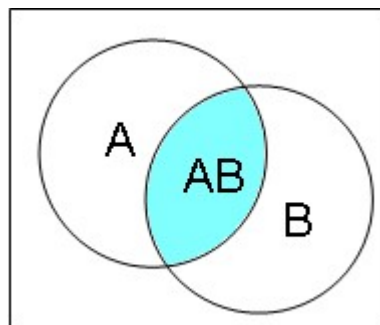
$$\rightarrow P(BA) = P(B|A)P(A)$$

$$\because P(AB) = P(BA)$$

$$\therefore P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- 全概率公式

$$P(B) = \sum_{i=1}^n P(B|A_i)P(A_i)$$



例子

- 已知一些数据，如下表。现有一条记录：

$X=(AGE=<30, INCOME=medium, STUDENT=yes, CREDIT_RAT=fair)$

- 类别（预测）属性为 **BAYS_COMPU**，问：其 X 的 **BAYS_COMPU** 值为 **yes** 还是 **no**？

ID	AGE	INCOME	STUDENT	CREDIT_RAT	BAYS_COMPU
1	<30	High	no	Fair	no
2	<30	High	no	Excellent	no
3	30-40	High	no	Fair	yes
4	>40	Medium	no	Fair	yes
5	>40	Low	yes	Fair	yes
6	>40	Low	yes	Excellent	no
7	30-40	Low	yes	Excellent	yes
8	<30	Medium	no	Fair	no
9	<30	High	yes	Fair	yes
10	>40	Medium	yes	Fair	yes
11	<30	Medium	yes	Excellent	yes
12	30-40	Medium	no	Excellent	yes
13	30-40	Medium	yes	Fair	yes
14	>40	Low	no	Excellent	no

b_k : 已知属性

A_i : 类别属性

计算过程

□ 计算事前概率 $P(A_i)$

$$P(\text{BAYS_COMP}=\text{yes})=9/14$$

$$P(\text{BAYS_COMP}=\text{no})=5/14$$

□ 计算条件概率 $P(b_k|A_i)$

$$P(\text{AGE} = '<30' | \text{BAYS_COMP}=\text{yes})=2/9$$

$$P(\text{AGE} = '<30' | \text{BAYS_COMP}=\text{no})=3/5$$

$$P(\text{INCOME} = \text{'Medium'} | \text{BAYS_COMP}=\text{yes})=5/9$$

$$P(\text{INCOME} = \text{'Medium'} | \text{BAYS_COMP}=\text{no})=1/5$$

同样计算 $\text{STUDENT}=\text{yes}, \text{CREDIT_RAT}=\text{fair}$ 的条件概率（略）

□ 计算 $\Pi(P(b_k|A_i))$

$$P(A|\text{BAYS_COMP}=\text{yes})=2/9*5/9$$

$$P(A|\text{BAYS_COMP}=\text{no})=3/5*1/5$$

□ 计算事后概率 $\Pi(P(b_k|A_i))P(A_i)$ （省略了 $\text{STUDENT}, \text{CREDIT_RAT}$ ，正常应计算）

$$P(A|\text{BAYS_COMP}=\text{yes}) * P(\text{BAYS_COMP}=\text{yes})=2/9*5/9*9/14=0.079$$

$$P(A|\text{BAYS_COMP}=\text{no}) * P(\text{BAYS_COMP}=\text{no})=3/5*1/5*5/14=0.042$$

□ 二者除以相同的 $P(B)$ ，不计算了。

□ 结论：由于 $0.079 > 0.042$ ，因此，X的 BAYS_COMPU 的取值为“yes”

贝叶斯分类器的训练就是计算事前概率、条件概率等值，当一条新的记录出现，可以根据这些取值计算新记录的属性类别（事后概率），实现预测。

目录

- 概述
- 常见机器学习技术
 - 归纳分析
 - 分类分析
 - 聚类分析
 - 异常分析
 - 关联分析
- 机器学习的局限性
- 数据挖掘语言：DMX简介

聚类分析

- 聚类（**clustering**）是一个将数据集划分为若干组（**class**）或类（**cluster**）的过程，并使得同一个组内的数据对象具有较高的相似度；而不同组中的数据对象是不相似的。
- 相似或不相似的描述是基于数据描述属性的取值来确定的，通常利用（各对象间）**距离**来进行表示。
- 对象间的距离是通过**数值型**计算获得，聚类分析针对**数值型数据**，如果是非数值型数据，则需要转化为数值型数据
- 与分类不同，**聚类不需要事先训练**，故也称**无指导学习**
- 常见聚类分析方法：**k-means算法**、**系统聚类法** 等

常用距离计算公式

□ 欧氏距离:

$$d(i, j) = \sqrt{(|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \cdots + |x_{ip} - x_{jp}|^2)}$$

□ **Manhattan** (曼哈顿) 距离:

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{ip} - x_{jp}|$$

□ **Minkowski** (明考斯基) 距离:

$$d(i, j) = (|x_{i1} - x_{j1}|^1 + |x_{i2} - x_{j2}|^2 + \cdots + |x_{ip} - x_{jp}|^q)^{1/q}$$

□ 相关系数：两个 p -维数据对象 i 和 j 的相关系数的计算公式

$$r_{ij} = \frac{\sum_{a=1}^p (x_{ia} - \bar{x}_i)(x_{ja} - \bar{x}_j)}{\sqrt{\sum_{a=1}^p (x_{ia} - \bar{x}_i)^2 \cdot \sum_{a=1}^p (x_{ja} - \bar{x}_j)^2}}, -1 \leq r_{ij} \leq 1$$

$$\bar{x}_i = \frac{1}{p} \sum_{a=1}^p x_{ia}, \bar{x}_j = \frac{1}{p} \sum_{a=1}^p x_{ja}$$

常用的聚类分析算法有 k -means算法和系统聚类法

k-means算法

- **k-means**法是一种常用的聚类算法，该算法中的每一个聚类均用相应聚类中对象的均值来表示，这种方法属于**划分方法**。
- 所谓**划分方法**是指给定一个包含 n 个对象或数据行，将数据集划分为 k 个子集（划分）。其中每个子集均代表一个聚类（ $k \leq n$ ）。也就是说将数据分为 k 组，这些组满足以下要求：
 - （a）每组至少应包含一个对象；
 - （b）每个对象必须只能属于某一组。
 - 需要注意的是后一个要求在一些模糊划分方法中可以放宽。
- 给定需要划分的个数 k ，一个划分方法创建一个初始划分；然后利用循环再定位技术，即通过移动不同划分（组）中的对象来改变划分内容。一个好的划分衡量标准通常就是同一个组中的对象“相近”或彼此相关；而不同组中的对象“较远”或彼此不同。

算法描述

k -means算法描述

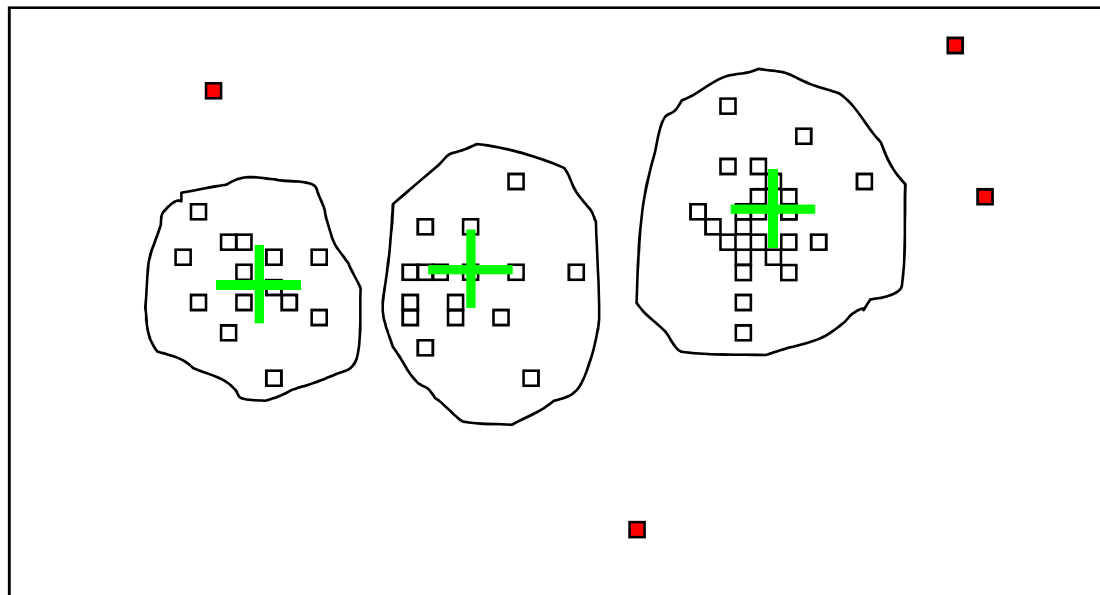
输入：聚类个数 k ，以及包含 n 个数据对象的数据库。

输出：满足距离最小标准的 k 个聚类。

- ① 从 n 个数据对象任意选择 k 个对象作为初始聚类中心；
- ② 循环③到④直到每个聚类不再发生变化为止；
- ③ 根据每个聚类对象的均值（中心对象），计算每个对象与这些中心对象的距离；并根据最小距离重新对相应对象进行划分；
- ④ 重新计算每个（有变化）聚类的均值（中心对象）

例子：k-means算法

- 事先确定要分3类，然后随机选3个做种子，确定其他各点到种子的距离，离哪个近就是哪个类的，计算各类的对象平均值，然后重新确定3类的中心(均值)，不断重复，直至平均值不再变化。



系统聚类法

- ❑ **k-means**法需要事先指定类别的数量，这往往限制了聚类的效果。
- ❑ **系统分类法**是多元统计学的一种方法，采用自然演化的方法进行聚类。
- ❑ **基本思想**：先将每个对象作为一个类别，然后计算相互之间的距离，距离小的合并为一类，以此类推。结果可以是一大类，也可以根据需要调整类别的数量。
- ❑ 参考多元统计学的相关知识。

目录

- 概述
- 常见机器学习技术
 - 归纳分析
 - 分类分析
 - 聚类分析
 - 异常分析
 - 关联分析
- 机器学习的局限性
- 数据挖掘语言：DMX简介

异常分析

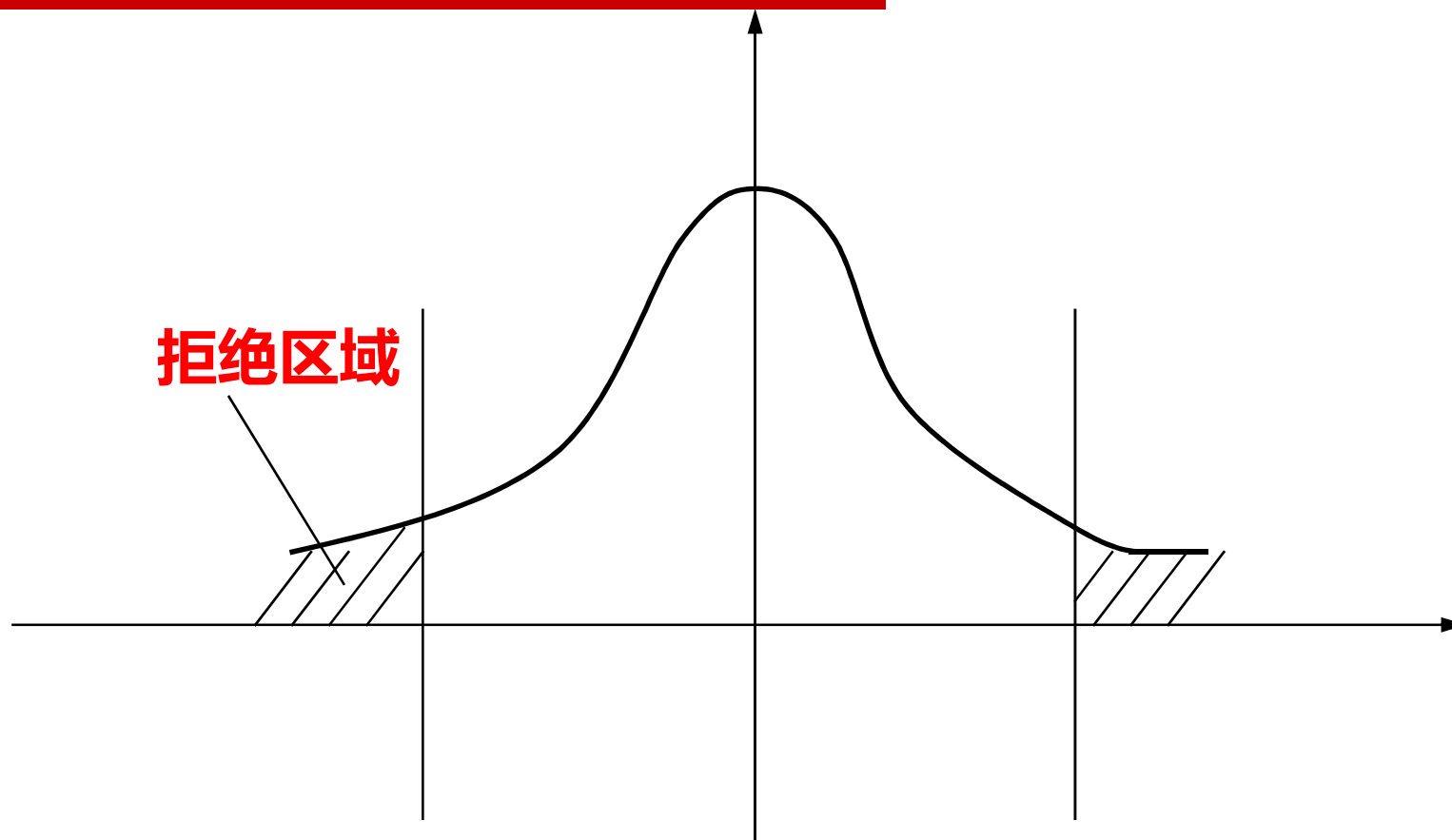
- 常常存在与数据模型或数据一般规律不符合的数据对象，这类与其它数据不一致或非常不同的数据对象就称为异常数据（Outliers）
- 异常数据可能由于测量误差、输入错误或运行错误而造成的，但是，异常数据有时也可能是具有特殊意义的数据。如：欺诈检测
- 异常分析又称为孤立点分析

异常分析的基本方法

- 常见的异常分析有以下三种：
 - 统计类方法
 - 基于距离方法
 - 基于偏差方法

基于统计的异常分析方法

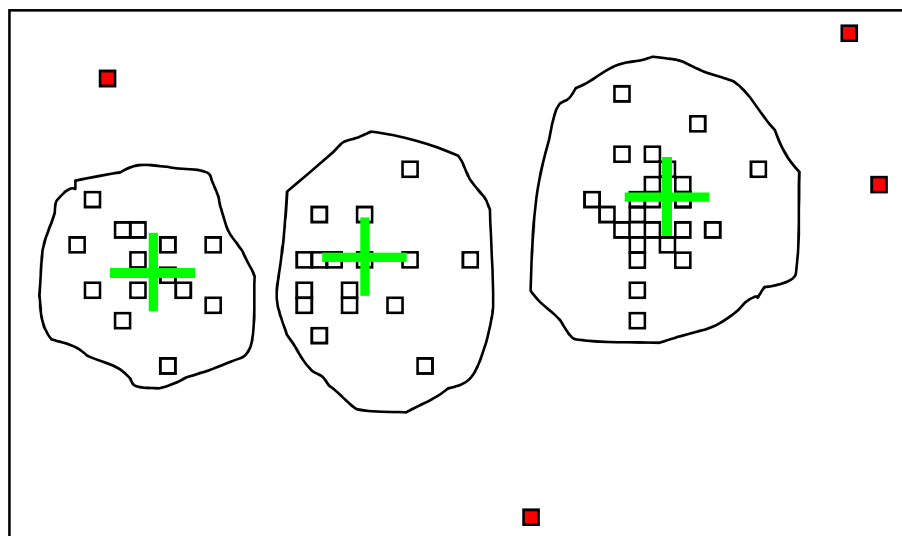
- 基于统计的异常检测方法假设所给定的数据集存在一个分布或**概率模型**（如一个正态分布）；然后根据相应模型并通过不一致性测试来发现异常数据。应用这种测试需要了解数据集参数的有关知识（如数据分布情况）、分布参数知识（如均值和方差），以及所预期的异常数据个数。
- 统计学中的**假设检验**



正态分布下的拒绝区域，为异常数据

基于距离的异常检测方法

- 一个数据集 S 中的一个对象是一个基于距离的异常数据（相对参数 p 和 d ），记为 $DB(p,d)$ ，它表示：若 S 中至少有 p 部分对象落在距离对象大于 d 的位置
- 换句话说将没有足够邻居的对象看成基于距离（检测）的异常数据
- 聚类分析：发现异常数据



目录

- 概述
- 常见机器学习技术
 - 归纳分析
 - 分类分析
 - 聚类分析
 - 异常分析
 - 关联分析
- 机器学习的局限性
- 数据挖掘语言：DMX简介

关联分析

- 发现关联规则，这些规则描述了一些属性值频繁地在给定的数据集中一起出现的条件。
- 关联规则： $X \Rightarrow Y$ ， X 、 Y 表示属性值集合， 即 $A1 \wedge A2 \wedge \dots \wedge A_n \Rightarrow B1 \wedge B2 \wedge \dots \wedge B_n$
- 例子：
 - $\text{age}(X, \text{"20-29"}) \wedge \text{income}(X, \text{"20k-29k"}) \Rightarrow \text{buys}(X, \text{"CD_player"})$ [support=2%, confidence=60%]， 年龄在20-29， 收入在20k-29k之间的会员占总数的2%。 某会员年龄在20-29， 如果收入在20k-29k之间， 则购买CD_player的可能性为60%。
 - **support**： 支持度， 在100个购物小票中同时包含“锤子和钉子”的有15个， 则支持度为15%， $\text{Support}(A \rightarrow B) = P(A \cup B)$
 - **Confidence**： 置信度， 当一个人买了锤子， 那么他有多大可能会钉子？ 这是置信度的问题， $\text{confidence}(A \rightarrow B) = P(B|A)$ ， 买锤子的人有30个， 也买钉子的有15个， 则置信度为50%

例子：Apriori算法

□ 一笔交易数据：

<div>□ 单号：7-215654 机台：15 □ 时间：04-02-12 卡号：207-3145267</div>			
货号	名称	数量	金额
202765	香蕉（公斤）	0.930	3.5
401021	小白菜（W）	0.144	0.4
402345	南北干货（袋）	1	2.8
213818	中排（公斤）	0.824	8.0
251549	五花肉（公斤）	0.398	4.9
236623	双汇王中王（40克*1）	1	6

-
- 单号——TID，如 T100=“7-215654”
 - 货号（或名称）——项ID，如 I1=“202765”

TID	项ID的列表
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

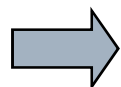
TID	ID列表
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3



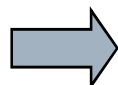
项集	支持度
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2



项集	支持度
{I1,I2}	4
{I1,I3}	4
{I1,I4}	1
{I1,I5}	2
{I2,I3}	4
{I2,I4}	2
{I2,I5}	2
{I3,I4}	0
{I3,I5}	1
{I4,I5}	0



项集	支持度
{I1,I2}	4
{I1,I3}	4
{I1,I5}	2
{I2,I3}	4
{I2,I4}	2
{I2,I5}	2



项集	支持度
{I1,I2,I3}	2
{I1,I2,I5}	2

注：省略了频繁项集的计算过程，
支持度阈值设定为2

□ 规则生成：频繁项集 $l = \{I1, I2, I5\}$ ，其非空真子集为 $\{I1, I2\}$ 、 $\{I1, I5\}$ 、 $\{I2, I5\}$ 、 $\{I1\}$ 、 $\{I2\}$ 、 $\{I5\}$

□ 关联规则计算如下：

- ① $I1 \wedge I2 \rightarrow I5$, confidence = $P(I1 \wedge I2 \wedge I5) / P(I1 \wedge I2) = 2/4 = 50\%$
- ② $I1 \wedge I5 \rightarrow I2$, confidence = $P(I1 \wedge I2 \wedge I5) / P(I1 \wedge I5) = 2/2 = 100\%$
- ③ $I2 \wedge I5 \rightarrow I1$, confidence = $P(I1 \wedge I2 \wedge I5) / P(I2 \wedge I5) = 2/2 = 100\%$
- ④ $I1 \rightarrow I5 \wedge I2$, confidence = $P(I1 \wedge I2 \wedge I5) / P(I1) = 2/6 = 33.3\%$
- ⑤ $I2 \rightarrow I1 \wedge I5$, confidence = $P(I1 \wedge I2 \wedge I5) / P(I2) = 2/7 = 28.6\%$
- ⑥ $I5 \rightarrow I1 \wedge I2$, confidence = $P(I1 \wedge I2 \wedge I5) / P(I5) = 2/2 = 100\%$

□ 如果置信度阈值设置为70%，则只有②、③、⑥三个规则

支持度用于频繁项集计算
置信度用于关联规则计算

用于关联规则分析的著名算法 — Apriori

算法是基于上述基本思想，算法具体描述略

目录

- 概述
- 常见机器学习技术
 - 归纳分析
 - 分类分析
 - 聚类分析
 - 异常分析
 - 关联分析
- 机器学习的局限性
- 数据挖掘语言：DMX简介

机器学习的局限性

- ❑ 机器学习算法不能产生所有有价值的模式（完备性，也称完全性）
- ❑ 机器学习假定数据集中存在有价值的模式，但事实上并非总是这样
- ❑ 机器学习算法不能仅产生有价值的模式

机器学习结果评估

- 机器学习的结果（模式）评价：
 - 易于用户理解
 - 对新数据或测试数据能够确定有效程度
 - 具有潜在价值
 - 新奇的，一个有价值的模式是知识

OLAP和机器学习的比较

□ OLAP可以帮助回答诸如下列问题：

- 过去3年里哪些人是我们最好的客户？
- 过去2年里哪些客户经常拖欠货款？
- 上个季度各区域的销售情况如何？
- 近4个季度哪些销售人员的业绩超过了配额？
- 去年哪些客户转向了其他公司？

□ 机器学习可以回答诸如下列问题：

- 前100个具有最好利润潜力的客户是谁？
- 哪些客户可能存在坏账风险？
- 明年各个地区的预期销售额是多少？
- 哪些商品排放在一起会更容易销售？

目录

- 概述
- 常见机器学习技术
 - 归纳分析
 - 分类分析
 - 聚类分析
 - 异常分析
 - 关联分析
- 机器学习的局限性
- 数据挖掘语言：DMX简介

关于数据挖掘语言

- 相比较SQL、OLAP（MDX）查询语言，数据挖掘语言还是一个比较新的查询语言。
- 1996年由Jiawei Han（韩家炜）等学者提出过一种称为DMQL（Data Mining Query Language）的数据挖掘语言（见《Data Mining Concepts and Techniques》，中文版见《数据挖掘概念与技术》第一版），如果当时算是第一个版本，后来的版本不得而知。DMQL用于DBMiner产品，DBMiner是加拿大Simon Fraser 大学（简称SFU）智能数据库研究所创建，由DBMiner Technology公司产品化的一款知识发现集成系统。

关于数据挖掘语言

- 数据挖掘组织协会（**DMG**）提出的预言模型标记语言 **PMML**（**Predictive Model Markup Language**），利用XML描述和存储数据挖掘模型，是一个已经被W3C所接受的标准。PMML是一种基于XML的语言，用来定义预言模型。
- 上述为研究机构 and 行业协会提出的数据挖掘语言，**DMX**为企业提出的数据挖掘语言，其他还有**JDM**(**Java Data Mining API**)。

这里以DMX为例介绍数据挖掘语言

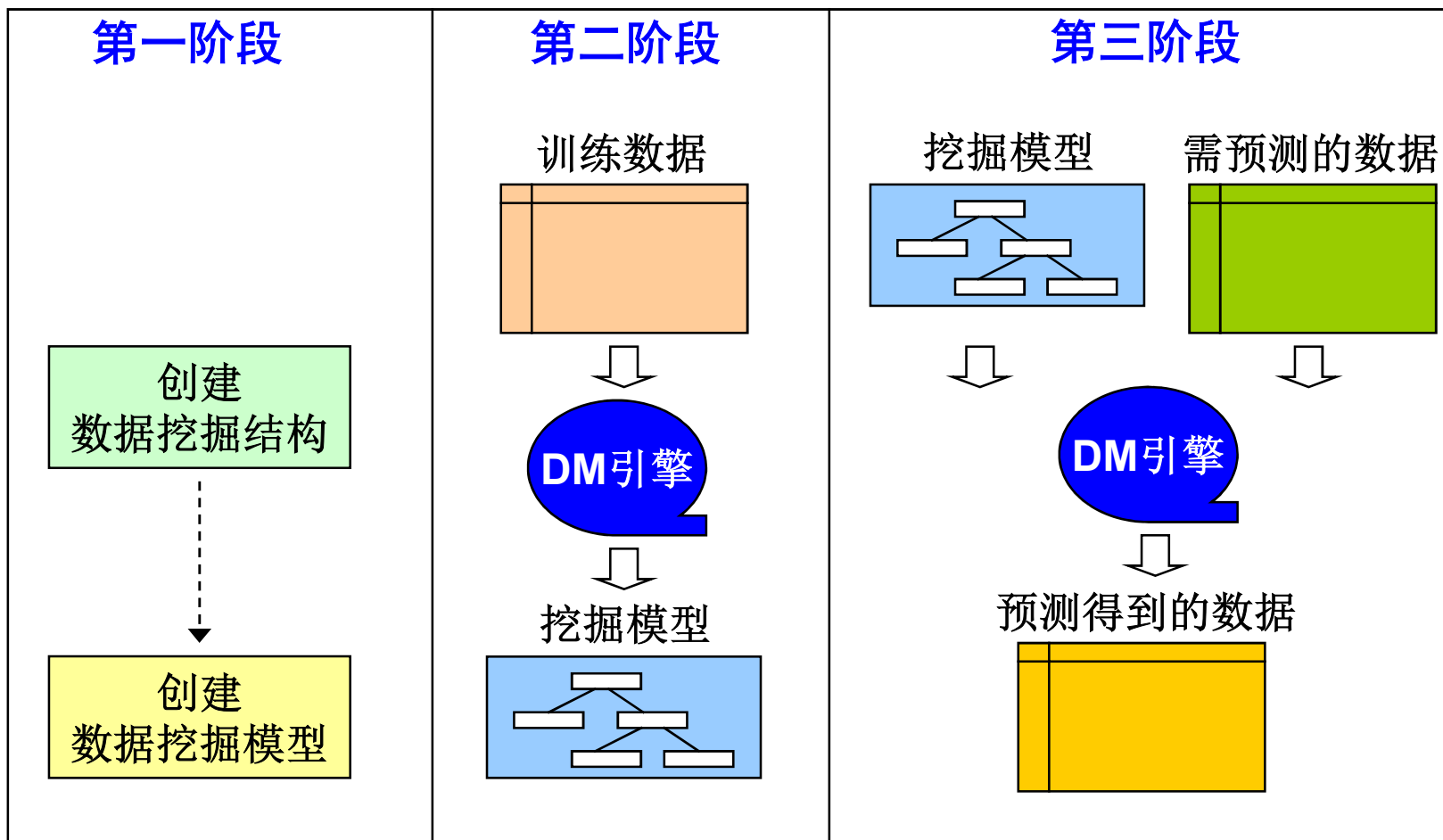
什么是DMX?

- **DMX (Data Mining Extensions)** : 数据挖掘扩展插件, 适用于SQL Server Analysis Service的数据挖掘环境, 也叫DMX数据挖掘语言。
- DMX是**微软的企业标准**语言, 适用于Microsoft的SQL Server产品, 目前支持到2008版本。
- 对应关系
 - **关系型数据库: SQL语言, 国际标准**
 - **多维数据库: MDX语言, 微软企业标准**
 - **数据挖掘领域: DMX语言, 微软企业标准**
- 与MDX一样, DMX也属于微软的OLE (Object Linking and Embedding) DB家族 (OLE DB for DM, 微软2007年7月提出的一个规范), 也是一种专用的**API**。
- **了解: OLE DB**类似ODBC, 但其访问的数据对象更广泛而不仅限于关系数据库。

DMX语言功能

- 可以使用 DMX 语句创建、处理、删除、复制、浏览和预测数据挖掘模型。
- DMX 中有两种类型的语句：
 - 数据定义语句（DDL）：对数据挖掘结构（型）及数据挖掘模型（值）进行定义、创建、删除、复制等操作。
 - 数据操作语句（DML）：对数据挖掘模型进行浏览、训练、预测等操作。

使用DMX进行数据挖掘的步骤



第一阶段：模型定义

□ 1. 创建一个挖掘结构

```
create mining structure Target_mail_Structure  
(  
  CustomerKey long key,  
  Age long continuous,  
  NumberCarsOwned long Discrete,  
  NumberChildrenAtHome long Discrete,  
  Region Text Discrete,  
  YearlyIncome long continuous,  
  BikeBuyer long Discrete  
)
```

□ 2. 根据挖掘结构定义一个挖掘模型

结构

alter mining structure Target_mail_Structure

add mining model Target_mail

模型

(CustomerKey,

Age,

NumberCarsOwned,

NumberChildrenAtHome,

Region,

YearlyIncome,

挖掘算法

BikeBuyer predict

)using Microsoft_Decision_Trees;

第二阶段：模型训练

□ 3. 用训练数据训练挖掘模型

Insert into Target_mail

模型

(CustomerKey,
Age,
NumberCarsOwned,
NumberChildrenAtHome,
Region,
YearlyIncome
BikeBuyer)

函数

用数据仓库Adventure Works
DW中的vTargetMail表中数
据训练

库 (源)

openquery ([Adventure Works DW],
'select CustomerKey, Age, NumberCarsOwned,
NumberChildrenAtHome, Region, YearlyIncome,
BikeBuyer from vTargetMail')

表 (源)

第三阶段：进行预测

- 4. 针对一个待预测数据集进行预测，输出为预测结果

SELECT

t.[ProspectAlternateKey],

[Target_mail].[BikeBuyer],

PredictProbability([Target_mail].[BikeBuyer])

From [Target_mail]

PREDICTION JOIN

OPENQUERY([Adventure Works DW],

'**SELECT** [ProspectAlternateKey], [YearlyIncome],

[NumberChildrenAtHome], [HouseOwnerFlag],

[NumberCarsOwned]

FROM [dbo].[ProspectiveBuyer]'

) **AS t**

ON

[Target_mail].[YearlyIncome] = **t**.[YearlyIncome] **AND**

[Target_mail].[NumberChildrenAtHome] =

t.[NumberChildrenAtHome] **AND**

[Target_mail].[NumberCarsOwned] = **t**.[NumberCarsOwned]

查询结果

消息 结果		
ProspectAlternat...	BikeBuyer	Expression
827	1	0.63383639888...
833	0	0.52113422833...
844	0	0.73569722117...
832	0	0.70812038614...
53313373327	0	0.70812038614...
54107006788	0	0.50529701670...
53315894603	1	0.81957906397...
54037360548	1	0.55200018193...
15732240080	1	0.55200018193...
53469896316	0	0.55956247433...
15737550900	0	0.74929159094...
21596444800	0	0.65150381409...
75533120036	1	0.55200018193...
21534748700	1	0.55200018193...
21585115194	0	0.65150381409...
21562647129	1	0.55200018193...

数据定义语句

- 使用 [CREATE MINING STRUCTURE](#) 语句创建挖掘结构，并使用 [ALTER MINING STRUCTURE](#) 语句在挖掘结构中添加挖掘模型。
- 使用 [CREATE MINING MODEL](#) 语句同时创建挖掘模型和关联的挖掘结构，以生成一个空的数据挖掘模型对象。
- 使用 [EXPORT](#) 语句将挖掘模型和关联的挖掘结构导出到文件中。使用 [IMPORT](#) 语句，将通过 **EXPORT** 语句创建的文件中导入挖掘模型和关联的挖掘结构。
- 使用 [SELECT INTO](#) 语句将现有挖掘模型的结构复制到新模型中，并使用相同的数据为其定型。
- 使用 [DROP MINING MODEL](#) 语句从数据库中完全删除挖掘模型。使用 [DROP MINING STRUCTURE](#) 语句从数据库中完全删除挖掘结构及其关联的挖掘模型。

挖掘结构与挖掘模型

- **挖掘结构**是一种数据结构，定义了生成挖掘模型的数据域（名称，类型）。单个挖掘结构可包含**多个共享**相同域的挖掘模型。
- 用挖掘结构可以定义**挖掘模型**，需要指明应用的挖掘算法及相关参数。
- 如挖掘结构一样，挖掘模型也包含列，但是从结构中继承过来的列。模型可以使用挖掘结构包含的所有列，或使用其中一部分列。
- 理解：挖掘结构类似“**类**”的概念，挖掘模型类似“**实例**”的概念，挖掘结构类似“**模式**”的概念，挖掘模型则类似“**关系**”的概念。但是要注意，挖掘模型定义的属性等于或少于挖掘结构的属性，在模型中可以指定建模标志（如**not null**）及预测子句（如**PREDICT**）。
- 挖掘结构采用语句**CREATE MINING STRUCTURE** 进行定义。挖掘模型可以用**ALTER MINING STRUCTURE** 语句及**ADD MINING MODEL**语句配合使用进行定义。

另一种定义方式

- 定义挖掘模型的另一种定义方式是使用 CREATE MINING MODEL 语句。
- 使用 CREATE MINING MODEL 语句定义挖掘模型的同时也定义了一个结构，结构的名称默认为：模型名_**Structure**
- 该结构也可以复用，但一般情况下考虑可能会基于某个挖掘生成多个用于不同挖掘算法的模型，因此，一般是设计好结构，**先定义结构，后定义模型**。

数据操纵语句

- 使用 **INSERT INTO** 语句为挖掘模型定型（训练）。执行该语句不会将实际源数据插入数据挖掘模型对象。
- 使用**SELECT**语句浏览在模型定型过程中计算的、并在数据挖掘模型中存储的信息，如源数据的统计信息。可以在 **SELECT** 语句中包括下列子句，以扩展其查询能力：
 - **SELECT DISTINCT FROM** <模型>: 某列的唯一值
 - **SELECT FROM** <模型>.**CONTENT** (子句): 内容
 - **SELECT FROM** <模型>.**CASES**: 钻取，定型事例，定义模型需指定可赚取
 - **SELECT FROM** <模型>.**SAMPLE_CASES**: 钻取，样本事例
 - **SELECT FROM** <模型>.**DIMENSION_CONTENT**: 返回与模型的维度用法相关的模型内容
- 使用 **SELECT** 语句的**PREDICTION JOIN**子句，创建基于现有挖掘模型的预测。
- 使用 **DELETE**语句从模型或结构中删除所有定型的数据。

INSERT INTO 语句

挖掘模型

训练数据

- 语法（简化：便于理解）

- Insert into <model> <Dataset>

- **Insert into** 语句的表层含义是向一个**挖掘模型**model插入了一个**数据集**dataset，其本质意义是对模型进行训练，处理指定的数据挖掘对象。
- 经过训练处理后的模型，查看可知，挖掘模型是以表的形式存储的，尽管表达的是树型结构。
- **Insert into** 语句不仅能够处理模型还可处理结构，其功能较多，可参考相关技术文献。
- 训练后的模型具备了**预测能力**。

SELECT语句

- SELECT 语句是DMX中非常复杂的语句，主要作用包括：
 - 浏览现有挖掘模型的架构行集的内容，浏览模型
 - 根据现有挖掘模型创建预测，创建预测
 - 创建现有挖掘模型的副本，复制模型
- 浏览模型内容
 - **SELECT** <属性列> **FROM** <模型>.**CONTENT**
- 创建预测
 - **SELECT** <select expression list> **FROM** <model>
PREDICTION JOIN <source data query>
ON <join mapping list>
- 创建副本
 - **SELECT INTO** <new model> **USING** <algorithm>
FROM <existing model>

例子：模型内容查看


select

attribute_name,
node_type,node_caption,
[children_cardinality],
node_description,
marginal_rule

from

Target_mail.CONTENT

函数



The screenshot shows a database query window with a SQL query in the top pane and a results table in the bottom pane. The query is: `select attribute_name, node_type,node_caption, [children_cardinality], node_description, marginal_rule from Target_mail.CONTENT`. The results table has 5 columns: `attribute_name`, `node_type`, `node_caption`, `children_cardinality`, and `node_description`. The data rows show various nodes for 'BikeBuyer' with different age and income ranges and their corresponding cardinalities.

attribute_name	node_type	node_caption	children_cardinality	node_description
	1		1	
BikeBuyer	2	全部	6	全部
BikeBuyer	3	Age < 35	2	Age < 35
BikeBuyer	3	Age >= 35 and < 43	5	Age >= 35 and <...
BikeBuyer	3	Age >= 43 and < 59	6	Age >= 43 and <...
BikeBuyer	3	Age >= 59 and < 67	3	Age >= 59 and <...
BikeBuyer	3	Age >= 67 and < 83	2	Age >= 67 and <...
BikeBuyer	4	Age >= 83	0	Age >= 83
BikeBuyer	3	Age < 69	2	Age >= 67 and <...
BikeBuyer	3	Age >= 69	2	Age >= 69 and <...
BikeBuyer	3	YearlyIncome < 42000	2	Age >= 69 and <...
BikeBuyer	3	YearlyIncome >= 42000	3	Age >= 69 and <...
BikeBuyer	4	YearlyIncome < 67600	0	Age >= 69 and <...
BikeBuyer	4	YearlyIncome >= 6760...	0	Age >= 69 and <...
BikeBuyer	4	YearlyIncome >= 80400	0	Age >= 69 and <...

数据类型

- ❑ **DMX** 支持以下挖掘结构列数据类型：
 - **Text**: 文本型（字符、串全包）
 - **Long**: 整型（长、短全包）
 - **Boolean**: 布尔型
 - **Double**: 浮点型（单、双精度全包）
 - **Date**: 日期型
- ❑ 相对**SQL**语言来说，**DMX**支持的数据类型少，但是已经够用。

内容类型

- 定义列数据类型只向算法提供了列数据的类型的信息，而不能有关该数据的行为的信息。
- 内容类型说明了列的行为，常用的如离散的（Discrete）、连续的（Continuous）等。
- DMX还提供以下内容类型
 - KEY：该列唯一地标识一行。
 - KEY SEQUENCE：该列是一个特定类型的键，其中的值表示一个事件序列。这些值是有序值，并且不必按等差排列。
 - KEY TIME：该列是一个特定类型的键，其中的值表示有序并按时间尺度出现的值。
 - ORDERED：该列包含定义有序集的值，有序属性列就内容类型而言是离散的。
 - CYCLICAL：该列包含表示循环有序集的值。例如，一周内顺序编号的七天便是循环有序集，第一天前是第七天。

数据类型支持的内容类型

数据类型	支持的内容类型
Text	Discrete、Discretized、Sequence
Long	Continuous、Cyclical、Discrete、Discretized、Key Sequence、Key Time、Ordered、Sequence、Time
Boolean	Discrete
Double	Continuous、Cyclical、Discrete、Discretized、Key Sequence、Key Time、Ordered、Sequence、Time
Date	Continuous、Discrete、Discretized、Key Time

函数

- ❑ **DMX**语言支持函数操作，可用函数来返回列的预测值及预测正确率。
- ❑ 使用 **DMX** 函数可以执行下列任务：
 - 返回预测。
 - 返回预测的统计信息，如概率和支持率。
 - 筛选查询结果。
 - 重新排序表表达式。
- ❑ **DMX**函数包括**Predict**等20几个。

例子：函数Predict

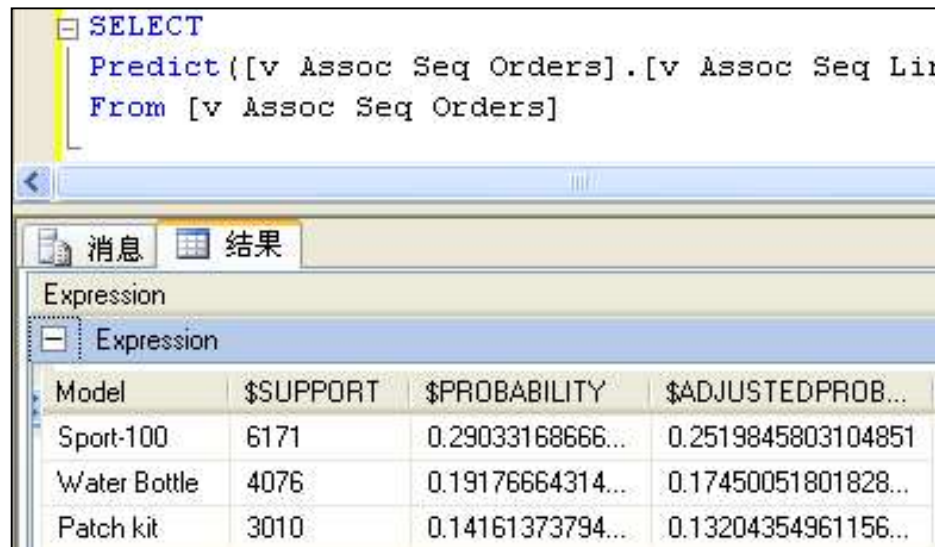
- Predict 函数为指定列返回一个预测值或一组值。

SELECT

Predict([Association].[v Assoc Seq Line
Items],**INCLUDE_STATISTICS**,3)

From [Association]

- 返回 Adventure Works 数据库中3种最可能一起销售的产品



The screenshot shows a SQL query window with the following text:

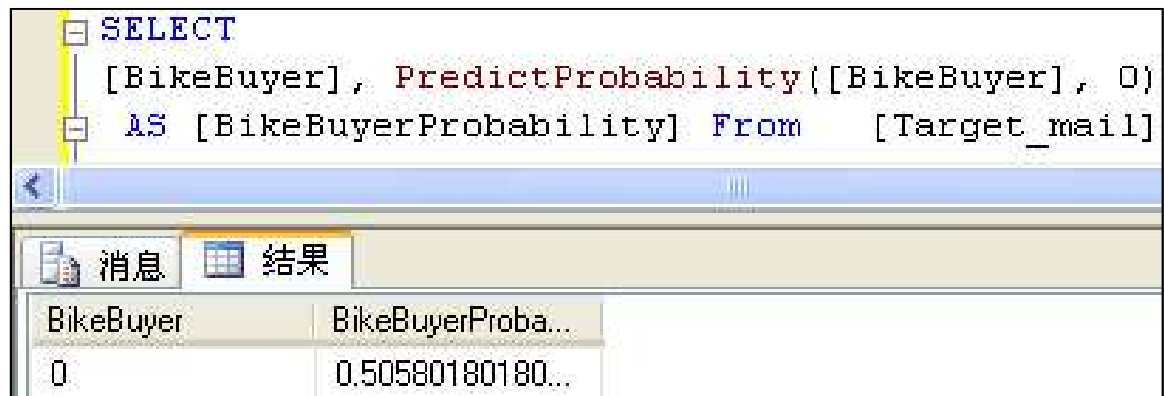
```
SELECT  
Predict([v Assoc Seq Orders].[v Assoc Seq Line  
Items],INCLUDE_STATISTICS,3)  
From [v Assoc Seq Orders]
```

Below the query window, the 'Results' tab is selected, displaying a table with the following data:

Model	\$SUPPORT	\$PROBABILITY	\$ADJUSTEDPROB...
Sport-100	6171	0.29033168666...	0.2519845803104851
Water Bottle	4076	0.19176664314...	0.17450051801828...
Patch kit	3010	0.14161373794...	0.13204354961156...

例子：PredictProbability函数

- 返回指定状态的概率。
- 语法：
 - PredictProbability(<scalar column reference>, [<predicted state>])
- 例子： [BikeBuyer]为0的概率
 - **SELECT**
[BikeBuyer], PredictProbability([BikeBuyer], 0)
AS [BikeBuyerProbability]
From
[Target_mail]



The screenshot shows a SQL query execution window. The query is: `SELECT [BikeBuyer], PredictProbability([BikeBuyer], 0) AS [BikeBuyerProbability] From [Target_mail]`. The results are displayed in a table with two columns: 'BikeBuyer' and 'BikeBuyerProbability'. The first row shows the value '0' for 'BikeBuyer' and '0.50580180180...' for 'BikeBuyerProbability'.

BikeBuyer	BikeBuyerProbability
0	0.50580180180...

挖掘算法

- 数据挖掘算法是创建挖掘模型的必要条件。若要创建模型，算法将首先分析一组数据，查找特定模式和趋势。然后，算法将使用此分析的结果来定义挖掘模型的**参数**。
- SSAS内嵌了以下算法
 - **决策树算法、Naive Bayes 算法、神经网络算法**：分类
 - **时序算法**：回归算法，基于数据集中的其他属性预测一个或多个连续变量，如利润或亏损。
 - **聚类分析算法**：将数据划分为组或分类，这些组或分类的项具有相似属性。
 - **关联算法**：查找数据集中的不同属性之间的相关性。购物篮分析。
 - **顺序聚类分析算法**：顺序分析算法，汇总数据中的常见顺序或事件，如 **Web** 路径流。

AS支持的算法

任务	可使用的 Microsoft 算法
预测离散属性。例如，预测目标邮件活动的收件人是否会购买某个产品。	Microsoft 决策树算法 Microsoft Naive Bayes 算法 Microsoft 聚类分析算法 Microsoft 神经网络算法 (SSAS)
预测连续属性。例如，预测下一年的销量。	Microsoft 决策树算法 Microsoft 时序算法
预测顺序。例如，执行公司网站的点击流分析。	Microsoft 顺序分析和聚类分析算法
查找交易中的常见项的组。例如，使用市场篮分析来建议客户购买其他产品。	Microsoft 关联算法 Microsoft 决策树算法
查找相似项的组。例如，将人口统计数据分割为组以便更好地理解属性之间的关系。	Microsoft 聚类分析算法 Microsoft 顺序分析和聚类分析算法

算法扩展

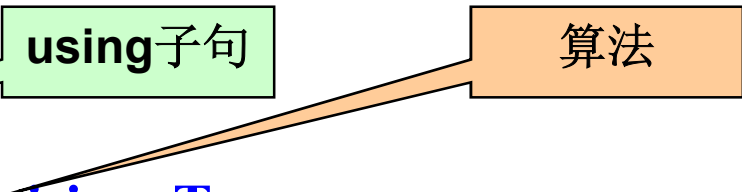
- 为特定的业务任务选择正确的算法很有挑战性。可以使用不同的算法来执行同样的业务任务，每个算法会生成不同的结果，而某些算法还会生成多种类型的结果。
- 微软提供的算法不一定能够满足所有需求，有时需要第三方算法，存在这样的需求。
- **Microsoft SQL Server 2005 Analysis Services (SSAS)**除了其所提供的算法外，还可以应用第三方的算法，即提供了算法的可扩展性。
- 只要第三方算法遵守特定的标准，就可以像使用 **Microsoft** 算法一样在 **Analysis Services** 中使用。
- 第三方算法也称插件算法，以“插件”的方式引进AS中。插件算法具有 SSAS 提供的算法的所有功能。

算法应用

❑ 创建挖掘模型的使用**using**语句指定使用的算法:

❑ 如:

```
alter mining structure Target_mail_Structure
add mining model Target_mail
(CustomerKey,
 Age,
 NumberCarsOwned,
 NumberChildrenAtHome,
 Region,
 YearlyIncome,
 BikeBuyer predict
)using Microsoft_Ddecision_Trees;
```



❑ 微软提供算法使用**Microsoft_***的方式, 如
Microsoft_Ddecision_Trees

小结

- ❑ DMX是微软提供的一种数据挖掘语言，本质是API，是微软的**企业标准**。
- ❑ DMX是一种非常复杂的查询语言，学习DMX需要具备数据挖掘的基础知识，并对相关算法有充分的认识。
- ❑ 使用DMX可以完成**挖掘结构及模型的定义**，**模型训练及预测**。
- ❑ DMX需要**数据挖掘算法、训练数据集、预测数据集**的支持。