2022–2023秋季课程:数据科学与大数据导论

Introduction to Data Science and Big data

Chapter 4: Big Data Analytics Algorithms

曹劲舟 助理教授

深圳技术大学 大数据与互联网学院

caojinzhou@sztu.edu.cn

2022年10月

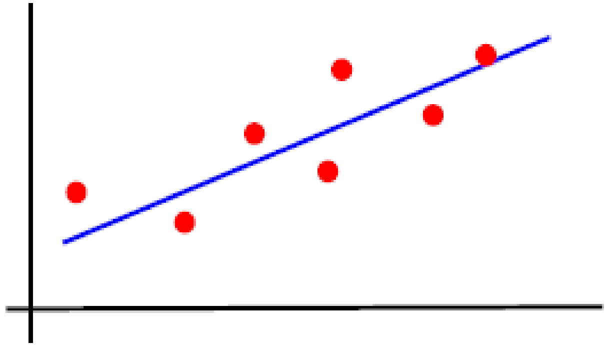
回归：一元线性回归

□回归（Regression）问题

- ■本周的股票价格如何变化?
- ■下礼拜一的气温会是多少?
- ■中国第一季度的GDP增长会是多少?

How **much** or How **Many**?

- ■估计回归的参数，如权重



解决方法：建立模型！

回归：一元线性回归

□什么是模型（Model）

- ■模型是对现实世界的一种“有用”的简化
- ■Model is a useful simplification of reality

Essentially, all models are wrong, but some are **useful**.

-- George Box (1919 - 2013)

□例子：重力公式

- ■$G = mg, g = 9.81$
- ■上述模型简化了以下因素：
- ■不同地区的重力差异
- ■空气阻力
- ■等等

回归：一元线性回归

- 建立模型的三个步骤
- Step（1）选择某种模型
  - 常数模型 – Constant Model
  - 线性回归模型 – Linear Regression Model
  - 更复杂的模型
- Step（2）选择目标函数(损失函数)
  - 均方误差（mean square error, MSE）
  - 平均绝对误差（mean absolute error, MAE）
  - 其它目标函数
- Step（3）拟合模型（model fitting）：优化目标函数
  - 最小化/最大化目标函数

回归：一元线性回归

符号	符号含义
y	**真实**的数据值（如小费） • 第i项数据值表示为y_i • 数据集表示为$\{y_1, y_2, \ldots y_n\}$
$\hat{y}$	**预测**的数据值（如预测的小费） • 第i项数据的预测值表示为$\hat{y}_i$
θ	模型的参数（Parameter）
$\hat{\theta}$	模型的**拟合参数**（fitted parameters） • 我们需要求解的**目标!**

回归：一元线性回归

- 概念辨析：请说出以下两个概念的区别和联系
  - 估计 Estimation
  - 预测 Prediction

- 估计（Estimation）是使用观测到的数据来**拟合参数**

$$\hat{\theta} = f_1(y, x)$$

- 预测（Prediction）是使用拟合的参数来**求解未知的数据**

$$\hat{y}_i = f_2(\hat{\theta}, x_i)$$

回归：一元线性回归

损失函数

- 度量模型预测的优劣，即**真实值y_i与预测值$\hat{y}_i$之间的差异**
- 给定某个数据集，度量**平均损失**，也称目标函数（Objective Function）

$$\frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{y}_i)$$

x	y	$\hat{y}$
x_1	y_1	$\hat{y}_1$
x_2	y_2	$\hat{y}_2$
…	…	…
x_n	y_n	$\hat{y}_n$

- 两种典型的平均损失
  - 均方误差（mean squared error, **MSE**）
  - 平均绝对误差（mean absolute error, **MAE**）
- 模型求解的目标：**最小化平均损失！**

回归：一元线性回归

□两种典型的平均损失：均方误差与平均绝对误差

■均方误差

- 采用平方损失（Squared Loss），也称为L2损失，针对所有数据点求平均

$$L_2(y_i, \hat{y}_i) = (y_i - \hat{y}_i)^2$$

$$\mathrm{MSE}(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

■平均绝对误差

- 采用绝对损失（Absolute Loss），也称为L1损失，针对所有数据点求平均

$$L_1(y_i, \hat{y}_i) = |y_i - \hat{y}_i|$$

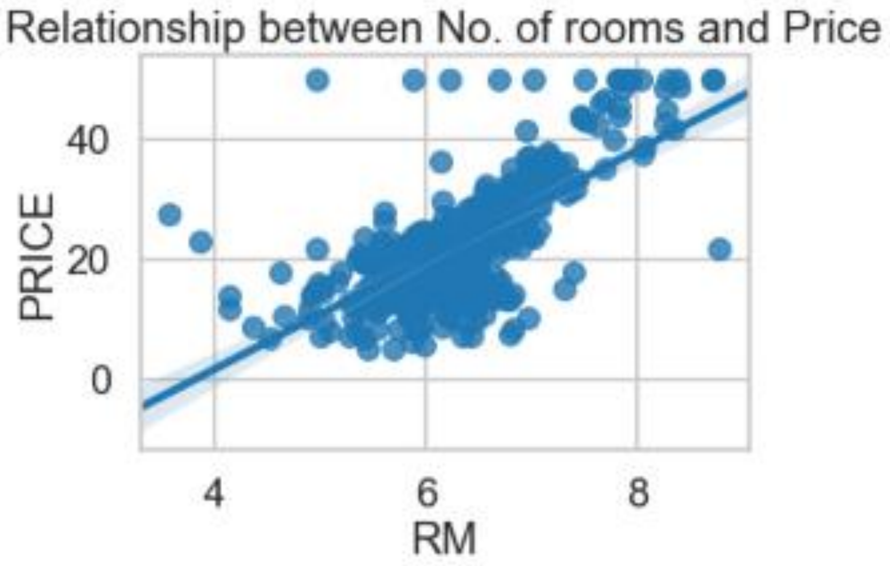
$$\mathrm{MAE}(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

回归：一元线性回归

□一元线性回归

- ■Simple Linear Regression(SLR) or Linear Regression with One Variable
- ■考虑输入变量x

$$\hat{y}_i = \theta_0 + \theta_1 x_i$$



□为了表示方便，将上式写成

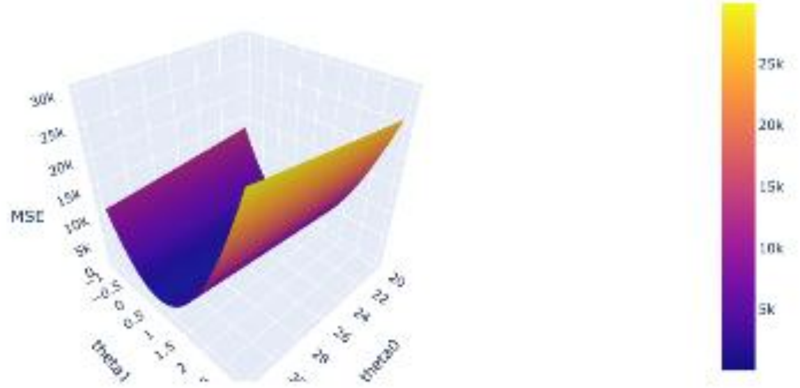
$$\hat{y}_i = a x_i + b$$

□该模型称为简单线性回归模型，简称SLR模型。

- ■例如：右图建立房间数量与房屋价格之间的SLR模型。

回归：一元线性回归

- SLR模型与MSE目标函数
- 给定SLR模型，均方误差MSE可以写为
  - $R(a,b)=\frac{1}{n}\sum_{i=1}^{n}\left(y_i-(\boldsymbol{a}\boldsymbol{x}_{\boldsymbol{i}}+\boldsymbol{b})\right)^2$
- 优化任务：如何计算最优的参数组合
  - $(\hat{a},\hat{b})=\arg\min\limits_{(a,b)}\frac{1}{n}\sum_{i=1}^{n}\left(y_i-(ax_i+b)\right)^2$
- 数学工具：
  - 计算变量(a,b)的一阶偏导
  - **令一阶偏导为0**，从而求解$(\hat{a},\hat{b})$



1，这是损失函数的可视化效果
2，极值点位置，即导数为0的位置

回归：一元线性回归

☐一元线性回归的解

- $\hat{a} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$
- $\hat{b} = \bar{y} - a\bar{x}$

回归：一元线性回归

- 课堂练习（5–10分钟）
- 给定一组训练数据
  - (2, 4)
  - (5, 1)
  - (8, 9)

$$\hat{a}=\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sum_{i=1}^{n}(x_i-\bar{x})^2}$$

$$\hat{b}=\bar{y}-a\bar{x}$$

- 请计算一个简单线性回归模型$\hat{y}=ax+b$的最优参数
  - $\hat{a}=?$
  - $\hat{b}=?$

注：可以使用手机计算器、python辅助计算

回归：一元线性回归

- 课堂练习 答案
- 给定一组训练数据
  - (2, 4)
  - (5, 1)
  - (8, 9)

$$\hat{a}=\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sum_{i=1}^{n}(x_i-\bar{x})^2}$$

$$\hat{b}=\bar{y}-a\bar{x}$$

$\bar{x}$=15/3=5

$\bar{y}$=14/3=4.667

$$\hat{a}=\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sum_{i=1}^{n}(x_i-\bar{x})^2}$$

$$=\frac{(-3)(-0.667)+(0)(-3.667)+(3)(4.333)}{(-3)^2+0^2+3^2}$$

$$=\frac{2.001+12.999}{(9+9)}=\frac{15}{18}=0.8333$$

$$\hat{b}=\bar{y}-a\bar{x}=4.667-0.8333*5=0.5005$$

$$\boldsymbol{y=ax+b=0.8333x+0.5005}$$

回归：多元线性回归

- 多元线性回归（Multiple Linear Regression, MLR）
  - 在简单（一元）线性回归SLR模型基础上添加**更多的独立变量**
- 多元线性回归的一般形式

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots \theta_d x_d = \theta_0 + \sum_{j=1}^{d} \theta_d x_d$$

- 基本概念：
  - 输入变量$x_1, x_2, \dots, x_d$也称：特征（Feature）、解释变量（Explanatory Variable）、回归量（Regressor）
  - 参数$\theta_1, \theta_2, \dots, \theta_d$度量了输入变量对预测值的**权重**
  - 参数θ_0为**截距项**

回归：多元线性回归

- 多元线性回归（ Multiple Linear Regression, MLR ）
  - 在简单（一元）线性回归SLR模型基础上添加更多的独立变量
- 针对每个数据点，添加一个常数特征$x_0 = 1$，得到

$$\hat{y} = \sum_{j=0}^{d} \theta_d x_d$$

- MLR模型举例：波士顿房价数据集
  - 输入变量
    - RM: average number of rooms per dwelling
    - LSTAT: % lower status of the population
  - 输出变量
    - Price: price of house

应该如何建立MLR模型?

回归：多元线性回归

- 向量点积
- 给定两个向量$\boldsymbol{a}$和$\boldsymbol{b}$

$$\boldsymbol{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \qquad \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

- 它们的点积操作定义为

$$\boldsymbol{a} \cdot \boldsymbol{b} = a_1 b_1 + a_2 b_2 + \cdots a_n b_n = \boldsymbol{a}^T \boldsymbol{b}$$

  - 两个向量的点积是一个**标量**，而非向量
  - 点积操作只能定义在两个相同长度的向量上
- 练习：
  - 将$a + bx_i$表示为点积的形式$\begin{bmatrix} a & b \end{bmatrix} \begin{bmatrix} 1 \\ x_i \end{bmatrix}$

回归：多元线性回归

- 多元线性回归（ Multiple Linear Regression, MLR ）

$$\hat{y} = \sum_{j=0}^{d} \theta_d x_d$$

- 引入两个向量：

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_d \end{bmatrix} \quad \boldsymbol{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

$$\Rightarrow \quad \hat{y} = [1 \quad x_1 \quad x_2 \quad \dots \quad x_d] \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_d \end{bmatrix} = \boldsymbol{x}^T \boldsymbol{\theta}$$

- 注：使用粗体表示向量及矩阵

请问上述公式中$\hat{y}$是
A. d*1的向量　B. 标量

回归：多元线性回归

□设计矩阵（Design Matrix）

■给定训练集，我们可以定义设计矩阵

$$\mathbb{X} = \begin{bmatrix} 1 & x_1^1 & x_2^1 & x_3^1 & \cdots & x_d^1 \\ 1 & x_1^2 & x_2^2 & x_3^2 & \cdots & x_d^2 \\ 1 & x_1^3 & x_2^3 & x_3^3 & \cdots & x_d^3 \\ 1 & x_1^4 & x_2^4 & x_3^4 & \cdots & x_d^4 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_1^{\mathrm{n}} & x_2^n & x_3^n & \cdots & x_d^n \end{bmatrix}$$

样本数量

每行代表一个数据实例（数据点）；如数据点1的特征

每列代表一个数据特征（输入变量）如所有点在特征1上的取值

上标：样本编号
下标：维度编号

回归：多元线性回归

□设计矩阵（Design Matrix）

- ■给定训练集，我们可以定义设计矩阵
- ■波士顿房价的例子

	BIAS	RM	LSTAT
0	1	6.575	4.98
1	1	6.421	9.14
2	1	7.185	4.03
3	1	6.998	2.94
4	1	7.147	5.33
...	...	...	...
501	1	6.593	9.67
502	1	6.120	9.08
503	1	6.976	5.64
504	1	6.794	6.48
505	1	6.030	7.88

506 rows × 3 columns

途中对应的设计矩阵维数

A. 506*1 **B. 506*3** C. 3*506 D. 3*1

回归：多元线性回归

□设计矩阵（Design Matrix）

- ■给定训练集，我们可以定义设计矩阵
- ■波士顿房价的例子
- ■基于设计矩阵，MLR模型表示为

$$\hat{\mathbb{Y}} = \mathbb{X}\boldsymbol{\theta}$$

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix}$$

	BIAS	RM	LSTAT
0	1	6.575	4.98
1	1	6.421	9.14
2	1	7.185	4.03
3	1	6.998	2.94
4	1	7.147	5.33
...	...	...	...
501	1	6.593	9.67
502	1	6.120	9.08
503	1	6.976	5.64
504	1	6.794	6.48
505	1	6.030	7.88

506 rows × 3 columns

请分别说出$\hat{\mathbb{Y}}$、$\mathbb{X}$和θ的维数

A. 506*1　B. 506*3　C. 3*506　D. 3*1

回归：多元线性回归

设计矩阵（Design Matrix）

- 基于设计矩阵，MLR模型表示为矩阵形式

$$\hat{\mathbb{Y}} = \mathbb{X}\boldsymbol{\theta}$$

$$\begin{bmatrix} \hat{y}^1 \\ \hat{y}^2 \\ \hat{y}^3 \\ \hat{y}^4 \\ \vdots \\ \hat{y}^{\mathrm{n}} \end{bmatrix} = \begin{bmatrix} 1 & x_1^1 & x_2^1 & x_3^1 & \cdots & x_d^1 \\ 1 & x_1^2 & x_2^2 & x_3^2 & \cdots & x_d^2 \\ 1 & x_1^3 & x_2^3 & x_3^3 & \cdots & x_d^3 \\ 1 & x_1^4 & x_2^4 & x_3^4 & \cdots & x_d^4 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_1^{\mathrm{n}} & x_2^n & x_3^n & \cdots & x_d^n \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \vdots \\ \theta_d \end{bmatrix}$$

$$\hat{y}^{(2)} = \mathbb{X}^{(2)}\boldsymbol{\theta} = \theta_0 + \theta_1 x_1^2 + \cdots + \theta_d x_d^2$$

- 上标表示样本编号，$\hat{y}^2$表示第二个样本的预测的y，为了和平方区分，有时记为$\hat{y}^{(2)}$
- 下标表示分量（第几个变量）

回归：多元线性回归

- **针对单一数据点**
  - 模型表示

$$\hat{y} = \boldsymbol{x}^T\boldsymbol{\theta}$$

    - $\boldsymbol{x}$是长度为$d+1$的向量
    - $\hat{y}$是标量（一个y值）
    - $\boldsymbol{\theta}$是长度为$d+1$的向量
- **针对整个训练集**
  - 模型表示

$$\hat{\mathbb{Y}} = \mathbb{X}\boldsymbol{\theta}$$

    - $\mathbb{X}$是$n \times (d+1)$的矩阵
    - $\mathbb{Y}$是n*1的向量
    - $\boldsymbol{\theta}$是 $(d+1)$*1的向量

注：为了表示方便，在不引起混淆的情况下，我们直接考虑d维

回归：一元回归

- MSE目标函数的矩阵形式
- 给定SLR模型，均方误差MSE可以写为

$$R(a,b)=\frac{1}{n}\sum_{i=1}^{n}\left(y_i-(a+bx_i)\right)^2$$

- 给定MLR模型线性回归的矩阵形式，我们有

$$\begin{aligned}R(\boldsymbol{\theta}) &= \|\mathbb{Y}-\mathbb{X}\boldsymbol{\theta}\|_2^2 \\ &= (y^1-x^1\boldsymbol{\theta})^2+(y^2-x^2\boldsymbol{\theta})^2+\cdots+(y^n-x^n\boldsymbol{\theta})^2\end{aligned}$$

回归：多元线性回归

- 利用几何含义解释MSE目标函数优化
- 令$R(\boldsymbol{\theta})$最小化的条件：
  - 向量$\mathbb{Y}-\mathbb{X}\boldsymbol{\theta}$与设计矩阵$\mathbb{X}$张成的$d$维子空间正交（Orthogonal）

- 根据**正交**的定义，我们得到

$$\mathbb{X}^T(\mathbb{Y}-\mathbb{X}\boldsymbol{\theta})=0$$

$$\Rightarrow \mathbb{X}^T\mathbb{Y}-\mathbb{X}^T\mathbb{X}\boldsymbol{\theta}=0$$

- 根据上式得到最优的参数估计

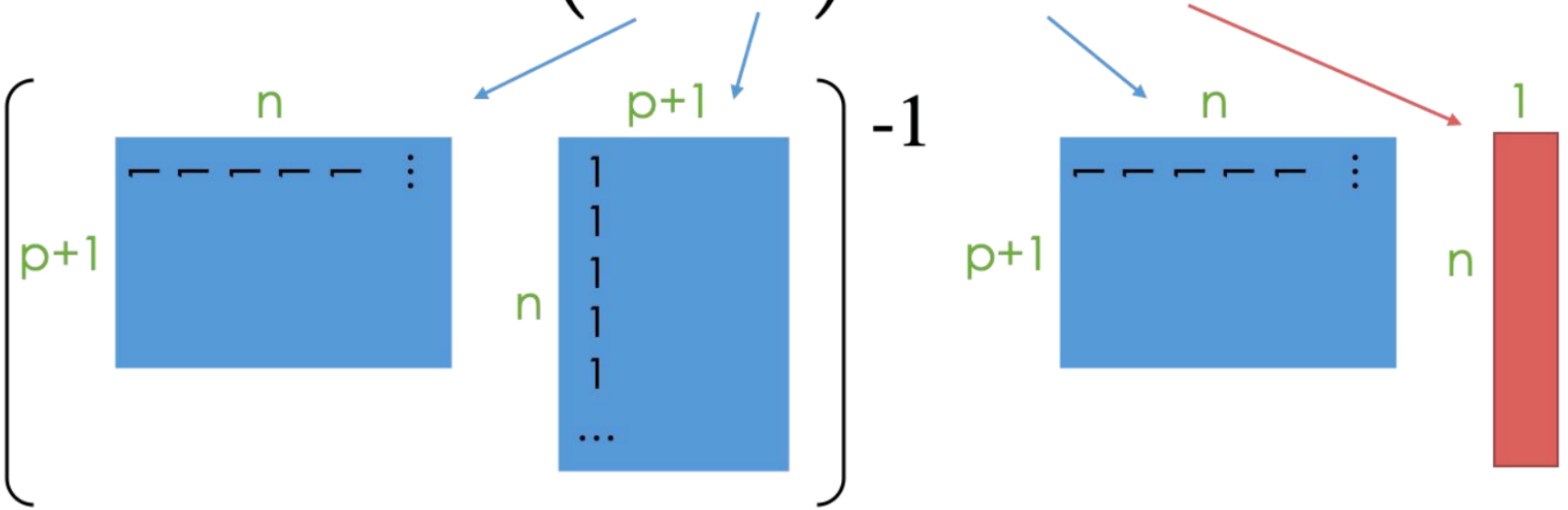
$$\boxed{\hat{\boldsymbol{\theta}}=(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{Y}}$$

回归：多元线性回归

□深入理解最优参数估计：最优的参数估计

- ■当维数d远小于数据量n时

$$\hat{\theta} = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{Y}$$



回归：多元线性回归

- 多元线性回归（Multiple Linear Regression, MLR）
  - 在简单（一元）线性回归SLR模型基础上添加**更多的独立变量**

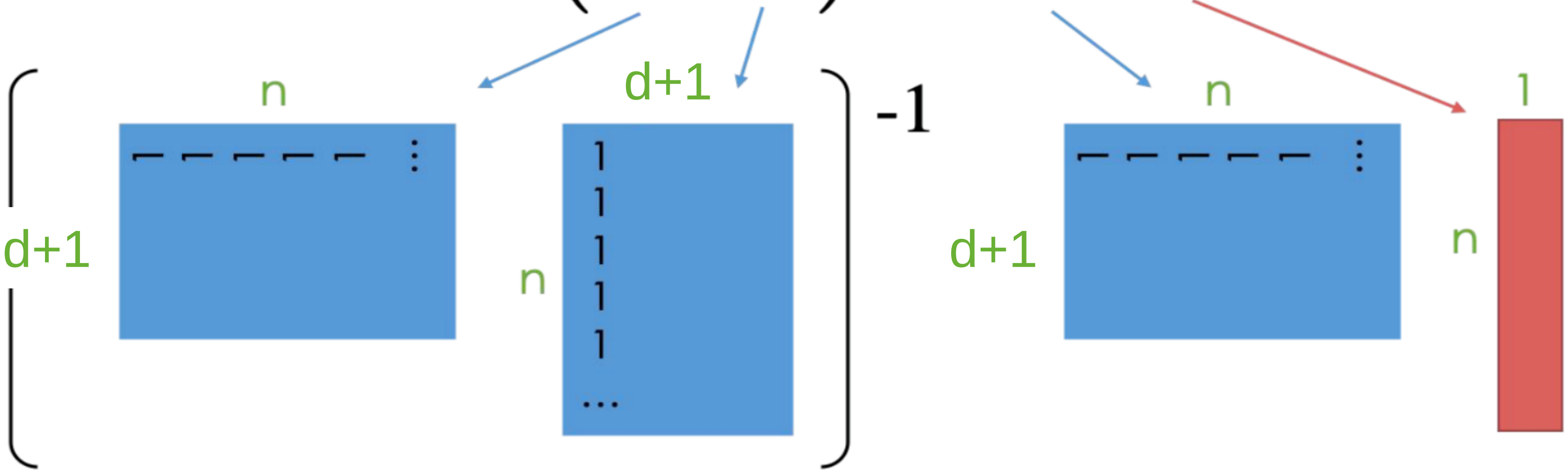
- 思考：
  - 上面为什么强调独立变量?
  - 给定MSE损失函数，SLR模型有唯一解，MLR有唯一解吗?
  - 如果希望MLR满足在MSE损失函数下有唯一解的条件是什么?

因此$\hat{\theta}$有唯一解的条件是，***d*个输入变量彼此线性独立！**

回归：多元线性回归

- 如何求解最优参数估计?
- 方法1：计算解析解

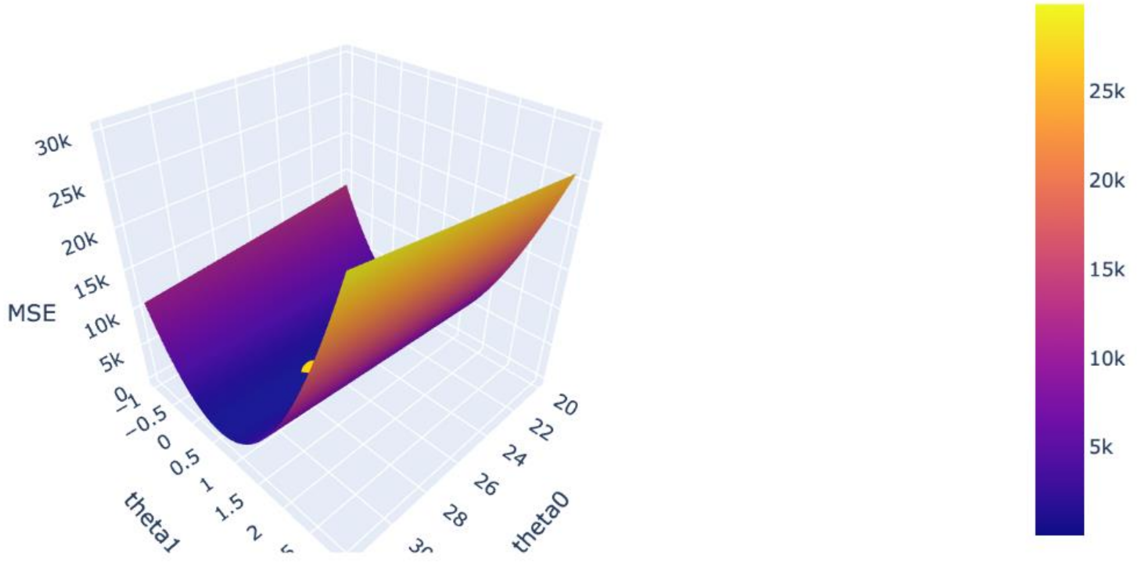
$$\hat{\theta} = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{Y}$$



时间复杂度高！

回归：多元线性回归

- 如何求解最优参数估计?
- 方法2：暴力搜索方法，枚举可能的参数值θ，计算MSE



枚举复杂度高！

回归：多元线性回归

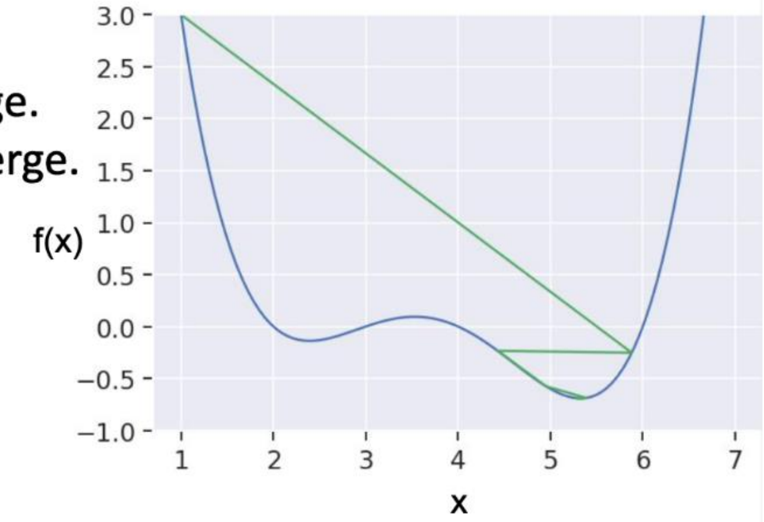
- 如何求解最优参数估计?
- 方法3：梯度下降法（Gradient Decent, GD）

The gradient descent algorithm is shown below:

- alpha is known as the "learning rate".
  - Too large and algorithm fails to converge.
  - Too small and it takes too long to converge.

$$x^{(t+1)} = x^{(t)} - \alpha \frac{d}{dx} f(x)$$

```
def gradient_descent(df, initial_guess, alpha, n):
    guesses = [initial_guess]
    guess = initial_guess
    while len(guesses) < n:
        guess = guess - alpha * df(guess)
        guesses.append(guess)
    return np.array(guesses)
```



回归：多元线性回归

- 一元/多元线性回归的评价：如何评判SLR/MLR两个模型的优劣
- 基本想法：度量观测值y与预测值$\hat{y}$之间的差异
- **均方根误差（Root Mean Squared Error）**

$$\mathrm{RMSE}(y,\hat{y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

  - 均方根误差RMSE是MSE损失函数的平方根
  - 均方根误差RMSE与观测值y与预测值y^的量纲相同
  - 均方根误差RMSE越小，说明模型越准确

回归：多元线性回归

- 拟合优度（Multiple R^2）度量预测值$\hat{y}$对观测值y的**拟合程度**
- 拟合优度R^2定义**$\hat{y}$与y皮尔森相关系数的平方**

$$R^2 = [r(y, \hat{y})]^2$$

- 拟合优度的取值范围在[0, 1]，越高说明模型准确性越好
- 拟合优度的含义：模型在多大程度上解释了观测值的变化

针对包含截距的线性回归模型，拟合优度也可如下计算

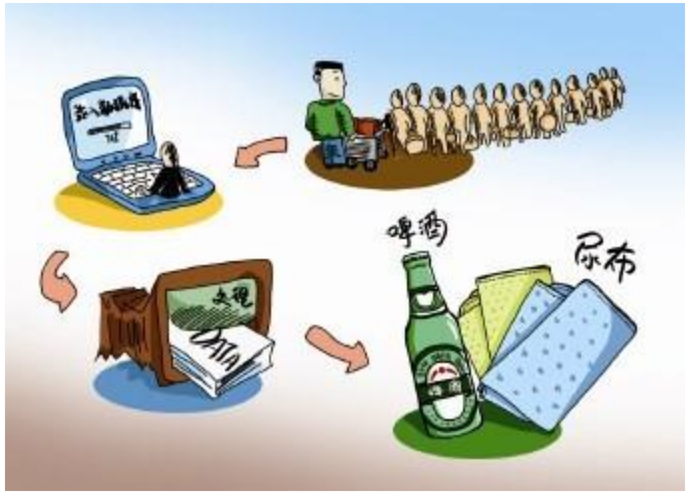
$$R^2 = \frac{\text{variance of fitted values}}{\text{variance of true values}} = \frac{\sigma_{\hat{y}}^2}{\sigma_y^2}$$

关联分析

□数据挖掘任务——关联分析(Association Analysis)

- ■例如："啤酒与尿布"
- ■在一次圣诞节的顾客消费行为分析中，沃尔玛意外发现跟尿布一起购 买最多的商品竟然是啤酒。经过深入分析后，卖场立即对两类商品的 空间距离与价格都进行了调整，结果尿布与啤酒销量双双大增。

萨姆·沃尔顿
沃尔玛公司创始人



轰动一时的啤酒与尿布关联规则

关联规则挖掘

□常用方法 —关联规则挖掘 (Association Rule Mining)

- ■给出事务的集合, 能够发现一些规则：$A => B$
  - ■当事务中某些子项出现时，预测其他子项也出现
- ■例如，从下表中得到一个可能的规则

购买尿布(Diaper)的用户很大可能会购买啤酒(Beer)

−→尿布和啤酒应陈列在一起销售

顾客购物交易数据

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

关联规则挖掘

□关联规则挖掘的基本概念

- ■Itemset（项集）
  - 一个或多个项目(items)的集合
- ■k-itemset：大小为k的项集
  - 例：{Milk, Bread, Diaper}是3项集
- ■Support（支持度）
  - 一个项集在数据中的出现频率
  - 例：$Support$({Milk, Bread, Diaper})=$\frac{2}{5}$
- ■Frequent Itemset（频繁项集）
  - 用户自行设定最小支持度阈值min_sup，支持度大于min_sup的项集称为频繁项集
  - 例：设$min_sup = 0.3$，则{Milk, Bread, Diaper}为频繁项集

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

关联规则挖掘

□关联规则挖掘的基本概念

■Association Rule（关联规则）
- 形如$X \rightarrow Y$的表达式，X, Y均为项集
- 例：{Milk, Diaper} →{Beer}

■Confidence (置信度)
- 度量包含X的事务中同时出现Y的频率
- 例：对于关联规则{Milk, Diaper} →{Beer}
- $confidence$({Milk, Diaper} →{Beer})= $\frac{2}{3}$

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

$$Confidence(A \rightarrow B) = \frac{Support(A \cup B)}{Support(A)}$$

■强关联规则
- 用户自行设定最小置信度阈值min_conf，置信度大于min_conf的规则称为强关联规则
- 例：设$min_conf = 0.5$，则{Milk, Diaper} →{Beer}为强关联规则

随堂练习

- 请依据下表计算出关于早餐的关联规则 {面包}->{豆浆} 的置信度

	买豆浆	不买豆浆	
买面包 不买面包	90 390	30 90	120 480
	480	120	600

随堂练习

- 请依据下表计算出关于早餐的关联规则 {面包}->{豆浆} 的置信度

	买豆浆	不买豆浆	
买面包 不买面包	90 390	30 90	120 480
	480	120	600

买面包的次数=120，

买面包的同时买豆浆的次数=90

置信度=$\frac{90}{120}=\frac{3}{4}$

关联规则挖掘

- 关联规则挖掘的一般步骤
  1. 根据支持度，寻找所有的频繁项集（频繁k项集）
  2. 根据频繁项集，生成频繁规则（长度大于2的频繁k项集）
  3. 根据置信度，过滤筛选规则
- 关联规则挖掘的第一步：如何寻找所有的频繁项集?
  - 暴力解法:
    - 穷举所有可能的项集，删除小于min_sup的项集

$\Rightarrow$ 计算效率低!

频繁项集挖掘

□频繁项集生成的经典算法

- ■APriori算法
- ■DHP算法(课后学习)
- ■FP-Growth算法(课后学习)

APriori算法

□频繁项集挖掘——APriori算法

- ■1994年，IBM研究员Agrawal提出，VLDB
- ■核心思想：广度优先搜索，自底而上遍历，逐步生成候选集与频繁项集
- ■反单调性原理：如果一个项集是频繁的，则它的所有子集一定也是频繁
  - 成立原因： $\forall X, Y: X \subseteq Y \rightarrow support（X）\geq support(Y)$
  - 依据该性质，对于某k+1项集，只要存在一个k项子集不是频繁项集，则可以直接判定该项集不是频繁项集
- ■算法步骤
  - 连接步：从频繁$K-1$项集生成候选K项集
  - 剪枝步：从候选K项集筛选出频繁K项集

APriori算法实例

- A-Priori算法实例
  - 【例】右图为某商店的用户购买记录，共有9个事务，A-Priori假定事务中的项按字典次序存放。

ID	事务
T100	l_1，l_2，l_5
T200	l_2，l_4
T300	l_2，l_3
T400	l_1，l_2，l_4
T500	l_1，l_3
T600	l_2，l_3
T700	l_1，l_3
T800	l_1，l_2，l_3，l_5
T900	l_1，l_2，l_3

APriori算法实例

□A–Priori算法实例

(1) 在算法的第一次迭代，每个项都是候选1项集的集合$C1$的成员。算法简单地扫描所有的事务，对每个项的出现次数计数

ID	事务
T100	l_1，l_2，l_5
T200	l_2，l_4
T300	l_2，l_3
T400	l_1，l_2，l_4
T500	l_1，l_3
T600	l_2，l_3
T700	l_1，l_3
T800	l_1，l_2，l_3，l_5
T900	l_1，l_2，l_3

扫描数据集,对每个候选1项集计算支持度

C_1	支持度
$\{l_1\}$	6
$\{l_2\}$	7
$\{l_3\}$	6
$\{l_4\}$	2
$\{l_5\}$	2

APriori算法实例

A-Priori算法实例

(2) 设最小支持度设为2，可以确定频繁1项集的集合$L1$

比较候选项集支持度与最小支持度阈值

L_1	支持度
$\{l_1\}$	6
$\{l_2\}$	7
$\{l_3\}$	6
$\{l_4\}$	2
$\{l_5\}$	2

APriori算法实例

A-Priori算法实例

(3) 使用$L1 \bowtie L1$产生候选2项集的集合$C2$

由L_1产生候选2项集

C_2
$\{l_1, l_2\}$
$\{l_1, l_3\}$
$\{l_1, l_4\}$
$\{l_1, l_5\}$
$\{l_2, l_3\}$
$\{l_2, l_4\}$
$\{l_2, l_5\}$
$\{l_3, l_4\}$
$\{l_3, l_5\}$
$\{l_4, l_5\}$

APriori算法实例

A-Priori算法实例

(4) 扫描数据集，计算$C2$中每个候选项集的支持度

ID	事务
T100	l_1，l_2，l_5
T200	l_2，l_4
T300	l_2，l_3
T400	l_1，l_2，l_4
T500	l_1，l_3
T600	l_2，l_3
T700	l_1，l_3
T800	l_1，l_2，l_3，l_5
T900	l_1，l_2，l_3

对每个候选2项集计算支持度

C_2	支持度
$\{l_1, l_2\}$	4
$\{l_1, l_3\}$	4
$\{l_1, l_4\}$	1
$\{l_1, l_5\}$	2
$\{l_2, l_3\}$	4
$\{l_2, l_4\}$	2
$\{l_2, l_5\}$	2
$\{l_3, l_4\}$	0
$\{l_3, l_5\}$	1
$\{l_4, l_5\}$	0

APriori算法实例

A-Priori算法实例

(5)最小支持度设为2，确定频繁2项集的集合*L2*

比较候选项集支持度
与最小支持度阈值

L_2	支持度
$\{l_1, l_2\}$	4
$\{l_1, l_3\}$	4
$\{l_1, l_5\}$	2
$\{l_2, l_3\}$	4
$\{l_2, l_4\}$	2
$\{l_2, l_5\}$	2

APriori算法实例

A-Priori算法实例

(6) 使用$L2 \bowtie L2$产生候选3项集的集合$C3$

①连接步： $C_3 = L_2 \bowtie L_2$

$$= \{\{l_1,l_2\},\{l_1,l_3\},\{l_1,l_5\},\{l_2,l_3\},\{l_2,l_4\},\{l_2,l_5\}\}$$

$$\bowtie$$

$$\{\{l_1,l_2\},\{l_1,l_3\},\{l_1,l_5\},\{l_2,l_3\},\{l_2,l_4\},\{l_2,l_5\}\}$$

$$= \{\{l_1,l_2,l_3\},\{l_1,l_2,l_5\},\{l_1,l_3,l_5\},\{l_2,l_3,l_4\},\{l_2,l_3,l_5\},\{l_2,l_4,l_5\}\}$$

L_2	支持度
$\{l_1,l_2\}$	4
$\{l_1,l_3\}$	4
$\{l_1,l_5\}$	2
$\{l_2,l_3\}$	4
$\{l_2,l_4\}$	2
$\{l_2,l_5\}$	2

APriori算法实例

A-Priori算法实例

(6) 使用$L2 \bowtie L2$产生候选3项集的集合$C3$

②剪枝步：反单调性：频繁项集的所有子集必须是频繁的

$\{\{l_1,l_2,l_3\},\{l_1,l_2,l_5\},\{l_1,l_3,l_5\},\{l_2,l_3,l_4\},\{l_2,l_3,l_5\},\{l_2,l_4,l_5\}\}$

- $\{l_1,l_2,l_3\}$的2项子集是 $\{l_1,l_2\}$,$\{l_1,l_3\}$和$\{l_2,l_3\}$

它们都是L_2的元素。因此保留$\{l_1,l_2,l_3\}$在C_3中

- $\{l_1,l_3,l_5\}$的2项子集是$\{l_1,l_3\}$,$\{l_1,l_5\}$和$\{l_3,l_5\}$

$\{l_3,l_5\}$ 不是L_2的元素，因而不是频繁的，由C_3中删除$\{l_1,l_3,l_5\}$

- 以此类推筛选得到C_3

C_3
l_1,l_2,l_3
l_1,l_2,l_5

APriori算法实例

A-Priori算法实例

(7) 扫描数据集，计算$C3$中每个候选项集的支持度

ID	事务
T100	l_1，l_2，l_5
T200	l_2，l_4
T300	l_2，l_3
T400	l_1，l_2，l_4
T500	l_1，l_3
T600	l_2，l_3
T700	l_1，l_3
T800	l_1，l_2，l_3，l_5
T900	l_1，l_2，l_3

对每个候选3项集计算支持度

C_3	支持度
$\{l_1, l_2, l_3\}$	2
$\{l_1, l_2, l_5\}$	2

APriori算法实例

A-Priori算法实例

(8)最小支持度设为2，确定频繁3项集的集合L_3

比较候选项集支持度
与最小支持度阈值

L_3	支持度
$\{l_1, l_2, l_3\}$	2
$\{l_1, l_2, l_5\}$	2

APriori算法实例

A–Priori算法实例

(9) 使用$L_3 \bowtie L_3$产生候选4项集的集合C_4，尽管连接产生结果 $\{l_1, l_2, l_3, l_5\}$这个项集被剪去，因为它的子集$\{l_2, l_3, l_5\}$不是频繁的。则$C_4 = \emptyset$,因此算 法终止，找出了所有的频繁项集如下：

L_1	支持度
$\{l_1\}$	6
$\{l_2\}$	7
$\{l_3\}$	6
$\{l_4\}$	2
$\{l_5\}$	2

L_2	支持度
$\{l_1, l_2\}$	4
$\{l_1, l_3\}$	4
$\{l_1, l_5\}$	2
$\{l_2, l_3\}$	4
$\{l_2, l_4\}$	2
$\{l_2, l_5\}$	2

L_3	支持度
$\{l_1, l_2, l_3\}$	2
$\{l_1, l_2, l_5\}$	2

APriori算法

□APriori算法

- 总结：APriori算法适合用在数据集稀疏，频繁模式较短，支持度较高的场景中
- 不足：难以适用于稠密数据和长频繁模式
  - 可能产生大量的候选集
  - 可能需要重复扫描数据集多次
- 改进方法（课后学习）
  - DHP算法
  - Partition算法
  - Sample算法
  - DIC算法

作业—Apriori算法

- 设$min_sup = 0.5$，给定下图数据，利用Apriori算法，求出所有频繁1项集、频繁2项集和频繁3项集

ID	事务
T100	1,2,3,4
T200	1,2,5
T300	1,2,3,5
T400	2,4,5
T500	1,2,5

关联规则挖掘

□关联规则挖掘的第二步：如何从频繁项集中生成规则?

- ■任务：给定一个频繁项集L, 寻找所有非空子集$f \subset L$ 使得$f \rightarrow L - f$ 满足置信度要求

ABC →D, ABD →C, ACD →B, BCD →A,
A →BCD, B →ACD, C →ABD, D →ABC
AB →CD, AC → BD, AD → BC, BC →AD,
BD →AC, CD →AB

- □ **若$|L| = k$, 则有$2^k - 2$ 种候选的关联规则**(忽略$L \rightarrow \emptyset$和$\emptyset \rightarrow L$)

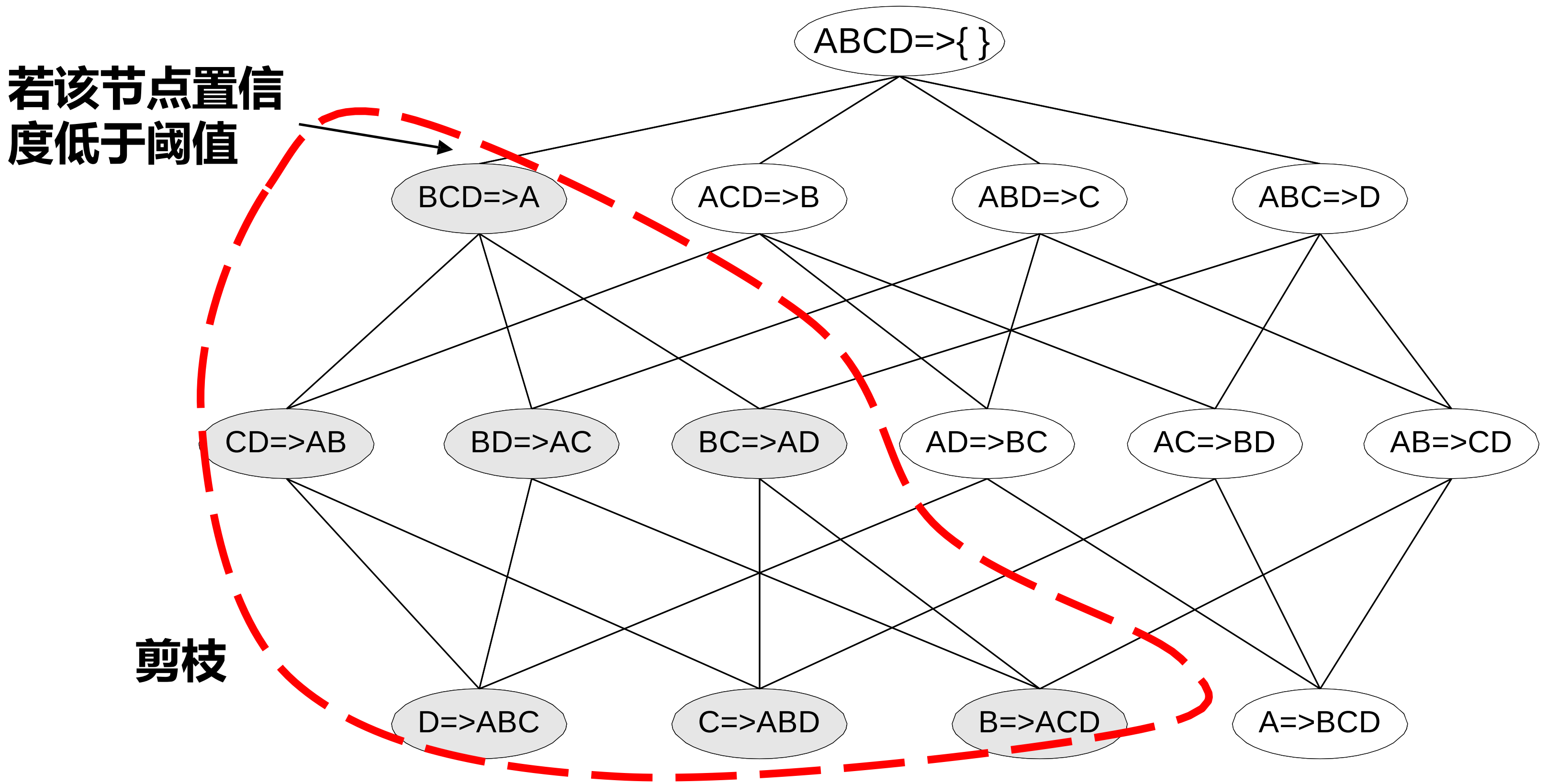
关联规则生成

- 关联规则生成(Rule Generation)
  - 如何高效地从频繁项集中生成规则?
  - 一般而言，置信度不满足反单调性
    - $confidence(\mathrm{ABC} \rightarrow \mathrm{D})$ 可能大于或小于$confidence(\mathrm{AB} \rightarrow \mathrm{D})$
  - 但从同一项集生成的规则满足反单调性（为什么？）
    - 例：L = {A,B,C,D}
    - $confidence(\mathrm{ABC} \rightarrow \mathrm{D})$
    - $confidence(\mathrm{AB} \rightarrow \mathrm{CD})$
    - $confidence(\mathrm{A} \rightarrow \mathrm{BCD})$

关联规则生成

关联规则生成

- 对某个频繁项集，自顶向下生成候选规则
- 若某个父节点置信度较低，其所有子节点无需再判断



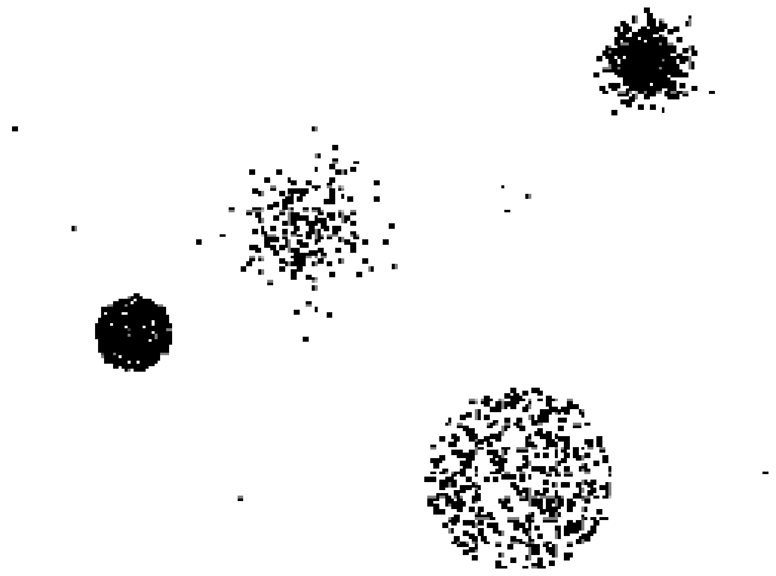
关联规则挖掘前沿：课后学习

- 多维关联规则挖掘
  - 多维的关联规则，如{购买: 电脑} ∧ {年龄∈ [20,30]} → {购买:手机}
- 多层关联规则挖掘
  - {光明牛奶，全麦面包} 的支持度低，抽象化为{牛奶，面包}等高层概念
- 稀有模式挖掘
  - 金融安全领域：普通的交易行为，非正常交易（欺诈交易）
- 负模式挖掘
  - {可口可乐，百事可乐} 支持度高，但 {可口可乐}→ ¬{百事可乐}
- 序列模式挖掘算法
  - 加入事务发生的时间：用户购买顺序的关联分析

Lei Zhang, Ping Luo, Linpeng Tang, Enhong Chen, Qi Liu, Min Wang, Hui Xiong, Occupancy-based Frequent Pattern Mining,ACM Transactions on Knowledge Discovery from Data.

异常检测

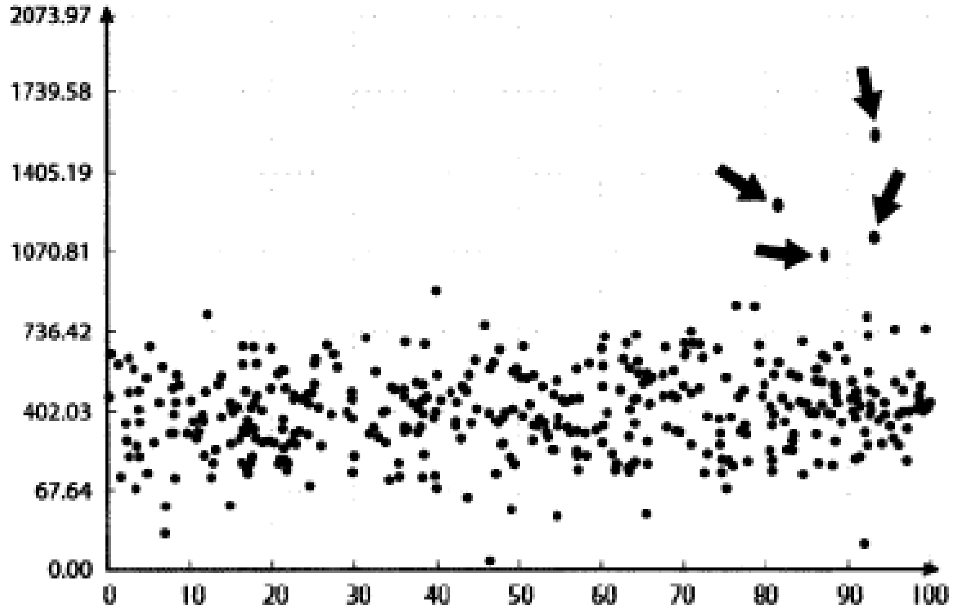
- 异常检测(Anomaly Detection) — 离群点检测
  - 什么是异常/离群点?
    - 与剩余的数据显著不同的数据点
  - 通常情况下异常点是罕见的
    - 成千上万的数据中，可能仅有几条
    - 情境context很重要, 例如, 7月份的温度是0度
  - 异常点的意义：可能是重要的，也可能是有害的
    - 旅游行业：游客的异常点
    - 电商领域：用户的异常交易
    - 金融领域：欺诈交易模式
    - 医疗领域：血压异常，心率异常等
    - 安防领域：飞机航线等
    - 。。。



异常检测

□异常检测：模型分析+后处理确认

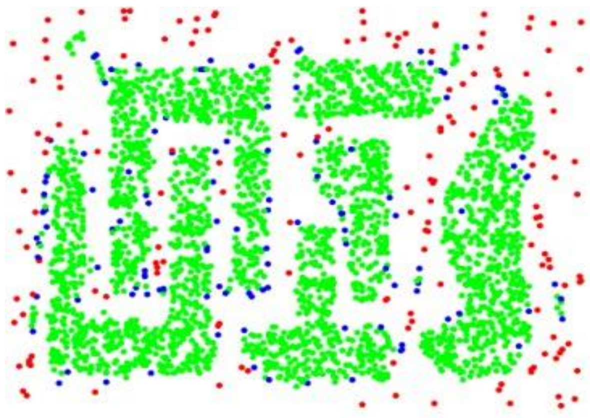
- ■无监督方法
  - 异常是那些不能拟合的点
  - 异常是那些扭曲模型的点
  - 代表方法:
    - 统计方法：数据分布，箱图
    - 聚类（最具代表）
    - 图分析
    - 生成对抗网络
- 监督方法
  - 异常数据通常含有罕见的类别



异常检测

□异常检测的方法

- ■基于邻近度的异常点检测
  - 异常点远离数据（距离度量）
- ■基于密度的异常点检测
  - 例如：DBSCAN算法
- ■模式匹配
  - 模板设计与匹配（例如，网页中正则表达式）
  - 关联规则挖掘算法（稀有模式）
- ■总的来说：与问题相关



Point types:

绿色core，蓝色border，红色noise

Xue Bai, Yun Xiong, Yangyong Zhu, Qi Liu and Zhiyuan Chen. Co-anomaly Event Detection in Multiple Temperature Series. Springer KSEM 2013, pages: 1-14,2013. **(Best Paper Award).**

1 Hengshu Zhu, Hui Xiong, Yong Ge, and Enhong Chen, Discovery of Ranking Fraud for Mobile Apps, IEEE Transactions on Knowledge and Data Engineering (IEEE TKDE).

异常检测

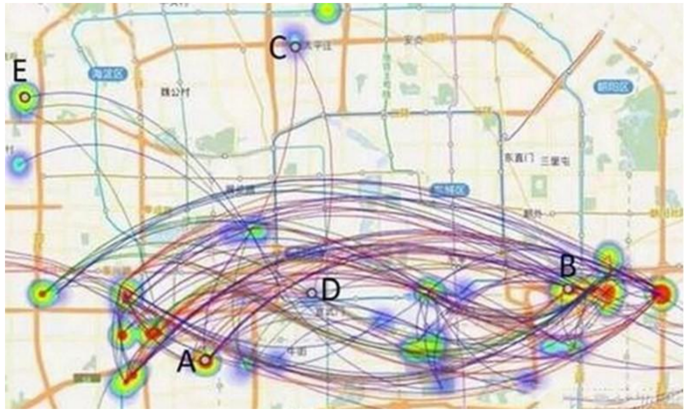
□大数据告诉你：公交车上谁是小偷！

（a）正常出行者，主要在居住地、工作地、途经区域活动

（b）旅游者，频繁访问圆明园、天安门、南锣鼓巷等景点区域。

（c）购物者，主要访问王府井、西单等购物区域。

（d）扒手，他们是一种流浪的模式，没有清晰的目的地，他们频繁地换乘，随机的停留，经常进行短途的出行。他们还（一段时间内）频繁地访问多种功能区：交通枢 纽（例如西直门）、购物区（例如王府井）、景点（例如鼓楼）



Big Data Analytics Algorithms

□数据挖掘/机器学习定义、四类任务及其应用场景

- ■非监督/聚类任务
  - K–Means、DBSCAN、评估方法
- ■监督/分类/回归任务
  - 决策树、K近邻、SVM、集成分类、评估方法
- ■关联分析
  - 支持度和置信度、Apriori算法
- ■异常检测

在设计针对大数据与小数据的挖掘方法时，所用的思想在本质上是一致的。