考前重点

数据分析概念

侠义的数据分析是传统的数学统计方法，广义的数据分析在侠义的基础上还包括数据挖掘

用适当的分析方法对收集来的大量数据进行分析，提取有用信息和形成结论而对数据加以详细研究和概括总结

1 数据分析的基本步骤有哪些？每个步骤的主要工作

1. 明确目的和思路
2. 数据收集：抽取、建立数据库
3. 数据处理：清洗、集成、归约、转换
4. 数据分析：数据统计、数据挖掘
5. 数据展现：图标、表格、文字
6. 报告撰写

2 关于大数据的4V理论是什么？

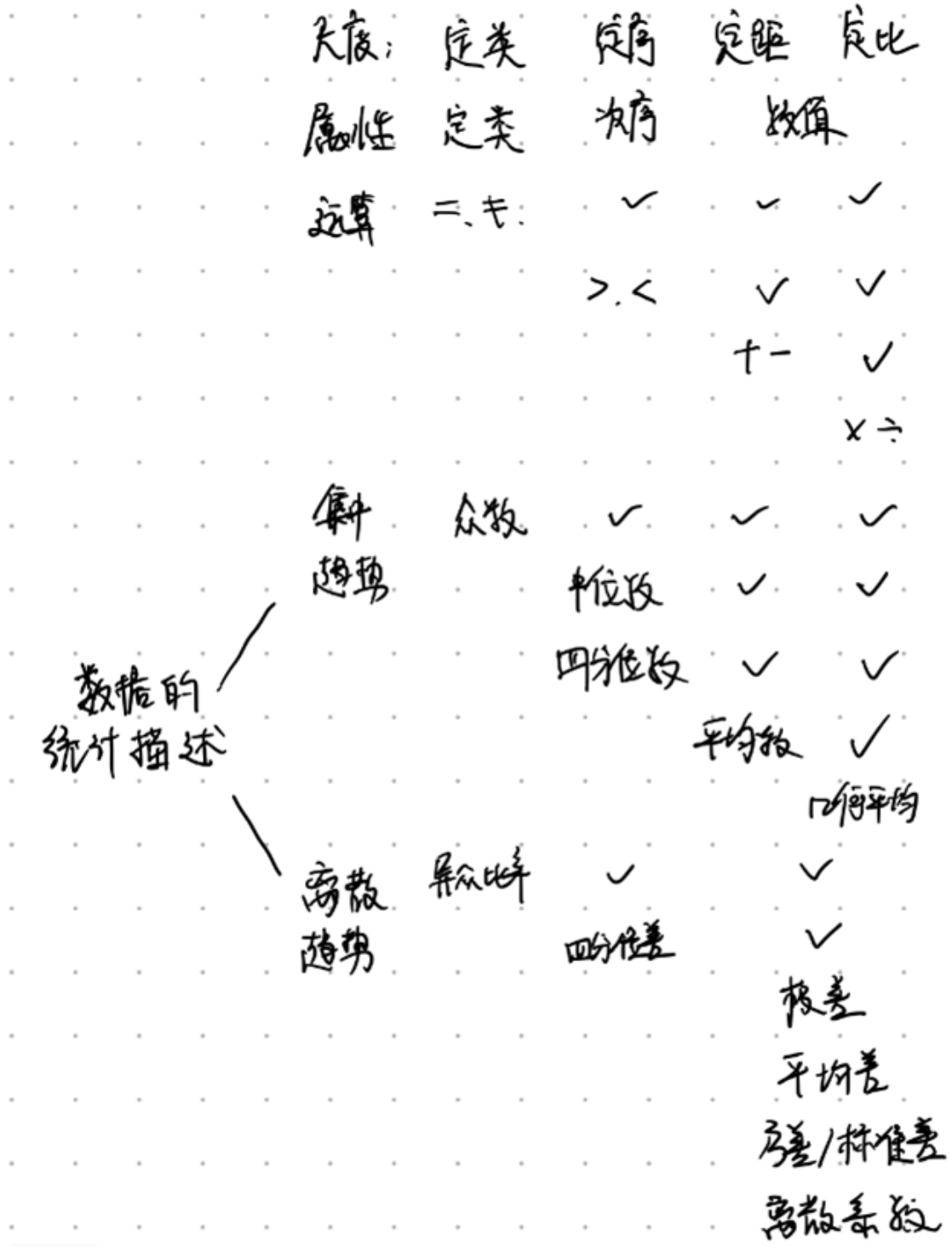
- volume 大量化
- velocity 速度化：处理速度快
- variety 多样化：数据类型繁多
- value 价值密度低

大数据的思维方式

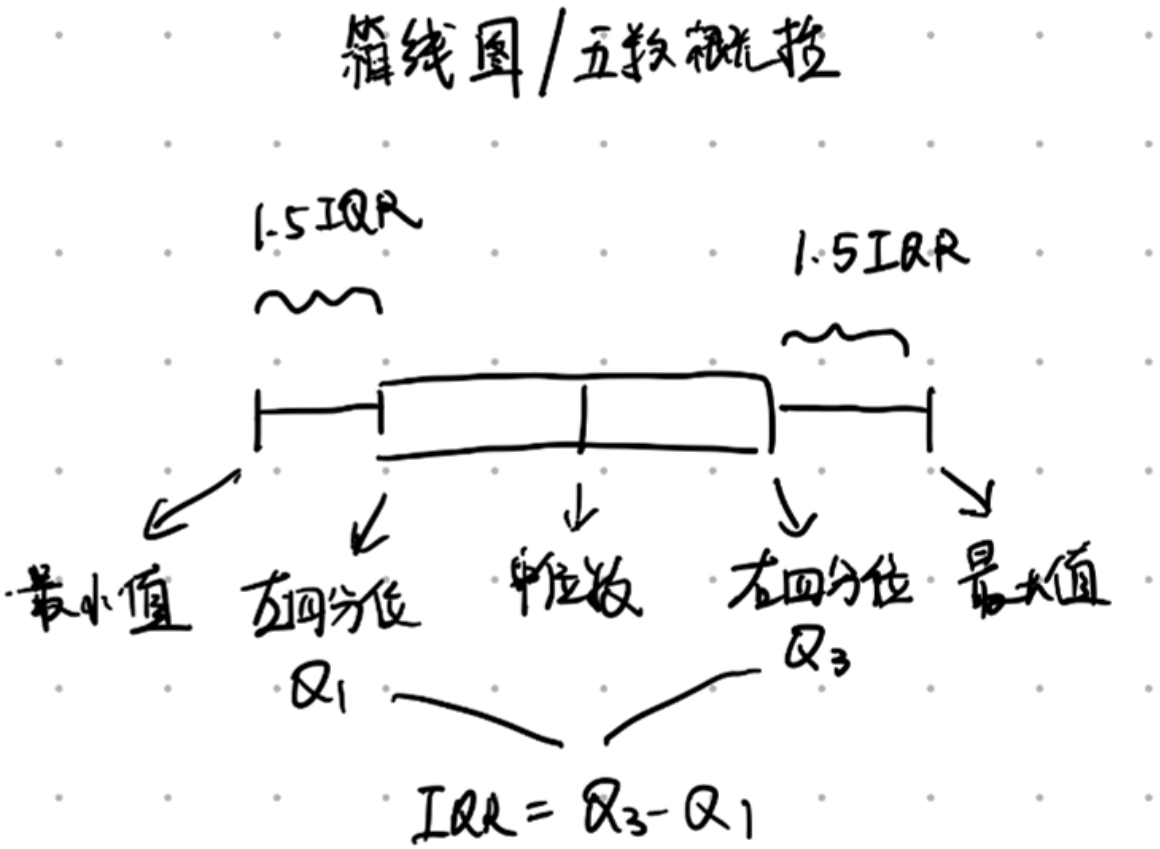
- 全样而非抽样
- 效率而非精确
- 相关而非因果
- 以数据为中心

四种尺度的计算比较

3 四种基本度量尺度适用的集中趋势和离散度量方法有哪些？

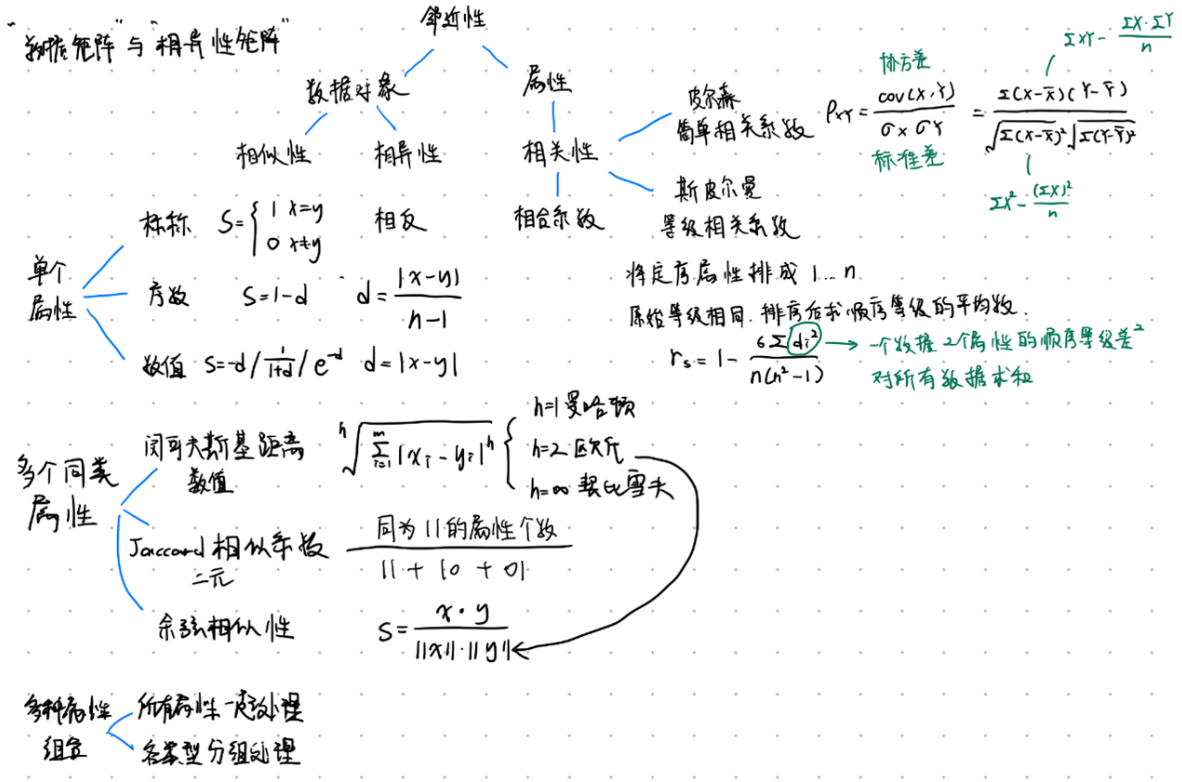


数据展示的五数概况和箱线表达



4 数据对象的相似性有哪些方法

5 数据属性的相关性有哪些方法（斯皮尔曼等级相关系数，皮尔森）



6 数据预处理的主要任务有哪些？每个任务要解决的问题主要有哪些？

- 数据清理：处理缺失数据，噪音处理，处理数据不一致
- 数据集成：模式匹配，实体识别，数据冗余，冲突检测
- 数据归约：维度规约，数值规约，离散化与概念分层，数据立方体聚集，数据压缩

- 数据变换：泛化，光滑，聚集，属性构造，规范化

7 脏数据主要有哪几种？产生的主要原因是什么？

- 不完全数据
  数据收集时未包含
  数据收集和数据分析时的不同考虑
  人/硬件/软件问题
- 噪音数据
  来源于收集、录入、变换
- 不一致数据
  不同的数据源
  违反函数依赖

8 缺失值的处理方法有哪些？

- 忽略元组：用于缺少类别标签的时候
- 全局常量填充：如 “unknown"
- 属性的中心度量填充
  对称分布的数据：均值
  倾斜分布的数据：中位数
- 同类属性的中心度量
  与给定元组属于同一类的所有样本度量值
  适用于分类数据挖掘
- 最可能的值：基于归纳推理的方法
  - 热卡填补法：寻找一个与它最相似的对象，用该相似对象的值进行填充
    常使用相关系数矩阵来确定该相似对象
  - 其他方法：最近距离决定填补法、回归填补法、多重填补方法、K-最近邻法、有序最近邻法、基于贝叶斯等

9 什么是噪音数据？产生的原因有哪些？

- 噪声：测量值相对于真实值产生偏差
- 孤立点：不符合模型的数据
- 引起噪声数据的原因
  错误的数据收集工具
  数据录入问题
  数据传输问题
  技术限制
  不一致的命名习惯

10 噪声数据的检测和处理方法有哪些？

噪声数据的检测方法：

- 简单统计分析
  对属性值规定范围，范围之外的值是不合理的
- 3σ原则
  样本距离平均值大于3σ的对象
- 使用距离检测多元离群点
  若不服从正态分布，计算远离平均距离多少倍的标准差
- 基于模型检测
  建立数据模型，模型不能完美拟合的对象

    - 聚类：将对象分组为不同的簇，找出并清除那些落在簇之外的孤立点
    - 回归：利用拟合函数对数据进行平滑及除去噪声
- 基于密度
  点的局部密度显著低于它大部分近邻的密度
- 计算机和人工检查相结合

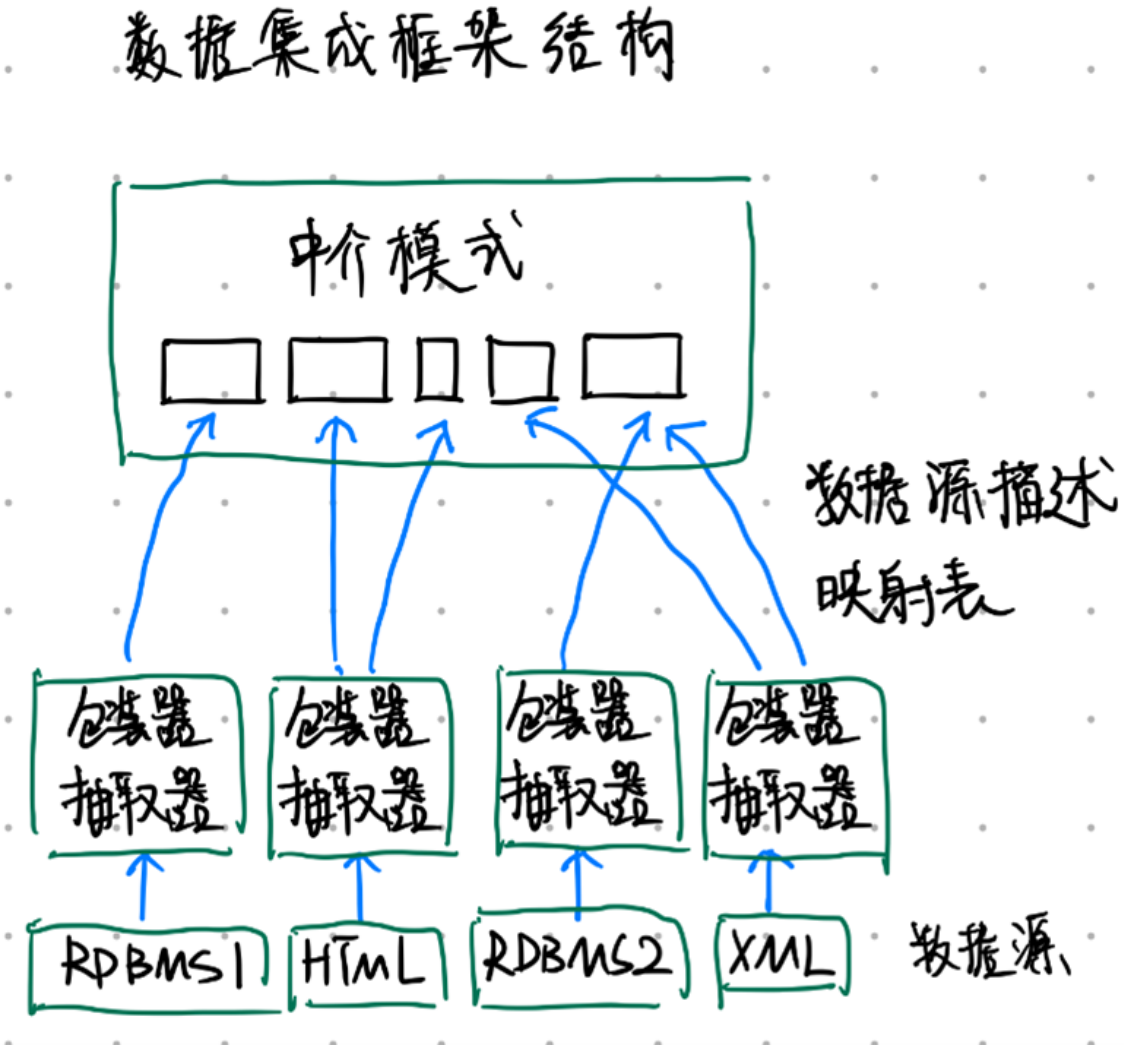
噪声数据的处理方法：

- 不处理
- 删除含有异常值的记录
- 将异常值视为缺失值，使用缺失值处理方法来处理
- **分箱**
    1. 把待处理的数据按照一定规则放进箱子中
        - 等宽度分箱：大小相等的n个区间
          区间宽度 =（属性最大值-属性最低值）/ 数据个数
        - 等频/等深分箱：含样本数目相同的n个区间
    2. 对箱子中的数据进行平滑处理
        - 箱平均值
        - 箱中值
        - 箱边界值

11 什么叫数据集成？集成的框架结构？分类？数据集成解决的主要问题有哪些？例子

定义：将互相关联的分布式异构数据源集成到一起，维护数据源整体上的数据一致性，使用户无需关心如何实现对异构数据源的访问，使用恰当方式访问集成数据即可

框架结构：



分类：虚拟集成、实际存储集成

- 虚拟集成：数据还是保存在原来的数据源中，只在需要查询时才被访问
  联邦数据库，中间件集成
- 实际存储集成：将数据从不同的数据源加载并存储到一个物理数据库
  数据仓库

数据集成解决的主要问题：

- **模式集成与模式匹配**
  - 模式匹配：建立两个不同模式中元素间的对应关系
  - 模式集成：通过匹配和映射方法将局部模式融合为一个统一的全局模式
- **实体识别：**识别出描述同一真实世界实体的元组集合，提取有意义的实体信息
  - 实体集识别：匹配多个数据源在现实世界中的等价实体集
- **数据冗余**
  - 属性冗余
    - 属性重复：同一属性在一个表中出现多次，只是字段命名不一致
    - 属性相关冗余：一个属性可以由另外一个属性推出
    - 计算相关性系数和协方差
  - 元组重复
    - 重复检测：识别数据集中，指向现实世界同一实体的重复记录
    - 去重：清除、合并
      清除：只把一条记录看成是正确的，其他记录则是含有错误信息的重复记录
      合并：把每一条检测出的重复记录看成是数据源的一部分，对这些记录进行合并，产生一条更完整的新纪录

- **冲突检测**
  - 物理冲突：不同系统使用的数据处理技术不一致
  - 字段冲突：在描述同一特征时，字段结构、命名、类型、长度、精度、单位等出现不一致
  - 记录冲突：描述同一个对象的不同记录之间有不同关键字、数据不一致
  - 表冲突：
    - 命名冲突：表名重复等
    - 结构冲突：表达相同概念的不同表具有不同的字段集
    - 关系冲突：例如A系统中，两表之间是父子关系，B系统中两表是等价关系
- 共享语义冲突

12 什么叫数据归约？主要有哪几类归约问题?

数据归约：从原有的庞大数据集获取精简的数据集，并且维持原有数据集的完整性

- 维度规约：特征选择，特征提取
- 数值规约：参数方法 (回归、对数线性模型)，非参数法 (直方图、聚类、抽样)
- 离散化与概念分层
- 数据立方体聚集
- 数据压缩

13 维度归约有哪两类技术？有什么区别?

- 特征选择：从原始特征中选出和任务相关的特征
- 特征提取：通过线性或非线性组合，将原始特征转化为新的特征

14 什么是数据离散化和概念分层?

- 离散化：将连续属性的值域划分为若干区间，消减属性的取值个数，转化为离散属性
- 概念分层：递归离散化属性，产生属性值分层划分；定义由底层概念集到高层概念集的映射，用较高层次的概念替换低层次的概念
  - 数值型：**分箱**、层次聚类、自然划分分段 **3-4-5规则**
  - 标称型：属性值的个数、语义

15 数据规范化/标准化的方法有哪些？形式，有什么作用?

- Z-Score / 标准化：x' = (x - 平均值) / 标准差
- Min-Max / 归一化：x' = (x - min) / (max - min)

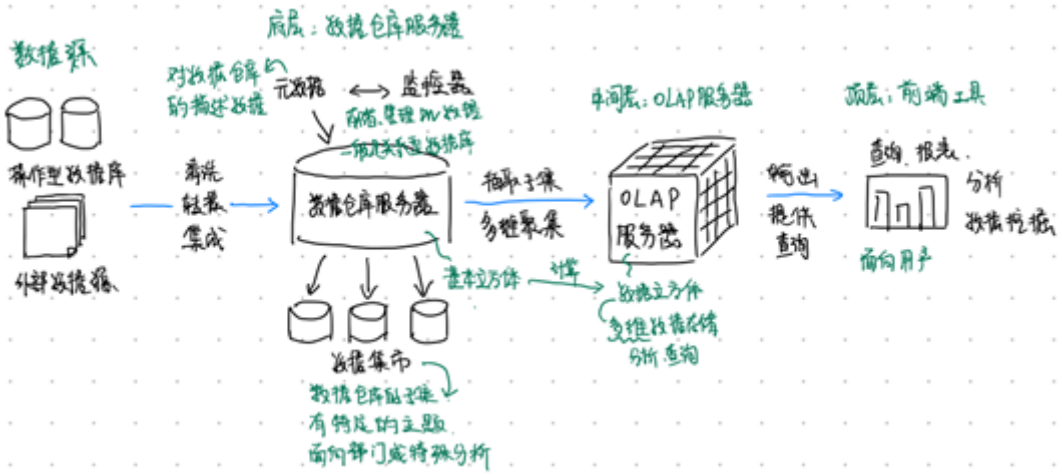
16 数据仓库的主要特征是什么，对每一特征给予简要解释

- 面向主题的：数据仓库以业务主题为中心组织数据，一个主题通常与多个操作型数据库相关，提供特定主题的视图
- 集成的：将来自不同数据源的数据，通过抽取、清洗、转换、整合到数据仓库中，保证数据一致性，并定期进行动态的集成刷新
- 反应历史变化的：数据仓库存储历史数据，可以利用历史数据进行查询和分析；数据仓库的关键数据结构也都隐式或显式地包含时间元素
- 相对稳定的：只需要定期的加载和刷新，加载后的数据极少更新；操作多为查询，很少修改和删除

17 数据仓库的作用

数据仓库提供用户用于决策支持的当前和历史数据，把操作形数据集成到统一的环境中以提供决策型数据访问，让用户更快更方便查询所需要的信息，提供决策支持

18 典型的数据仓库体系结构，各层简要说明



19 数据库与数据仓库系统在设计上的差别

系统	数据库	数据仓库
需求目标	面向事务型处理和应用	面向决策分析
数据来源	外部获得 关注数据的安全性和完整性	大部分从内部获得，一部分从外部获得 关注数据的一致性
处理类型	增删改查	复杂查询

数据仓库的**“数据驱动”**系统设计方法：尽可能利用已有的数据和代码，识别出当前系统设计与已做工作的共同性

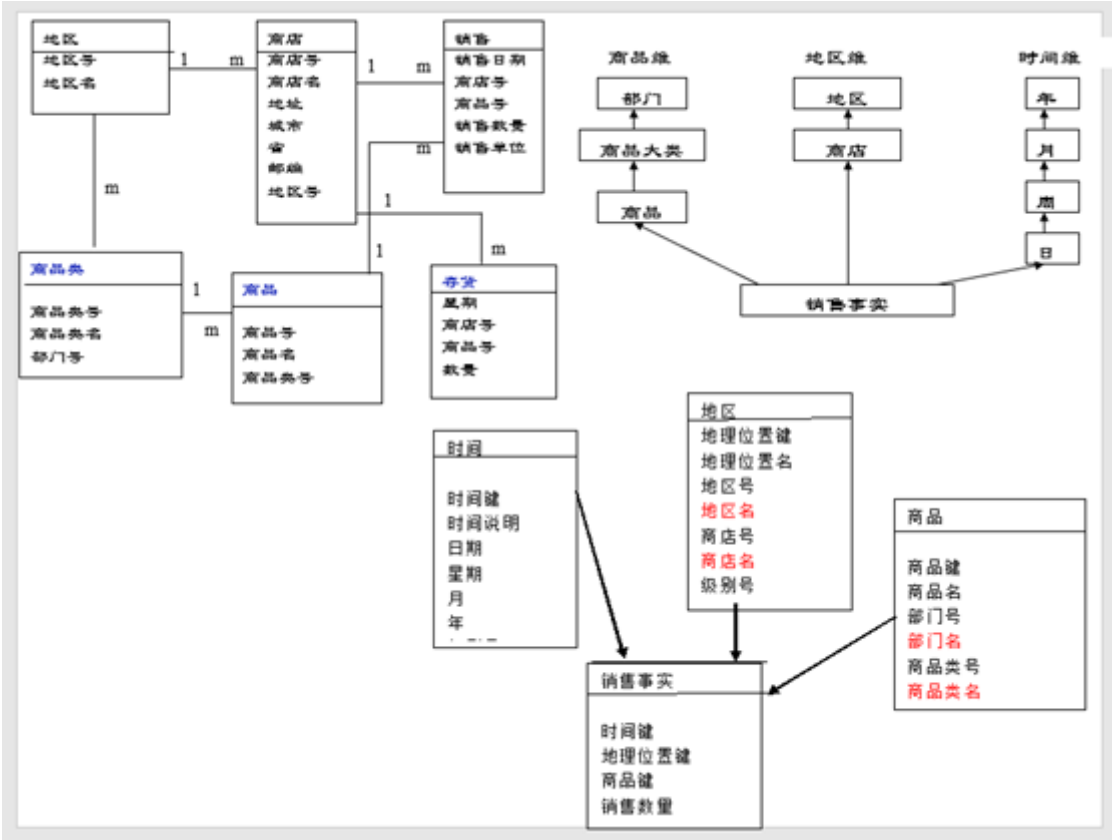
20 数据仓库设计的过程有哪些

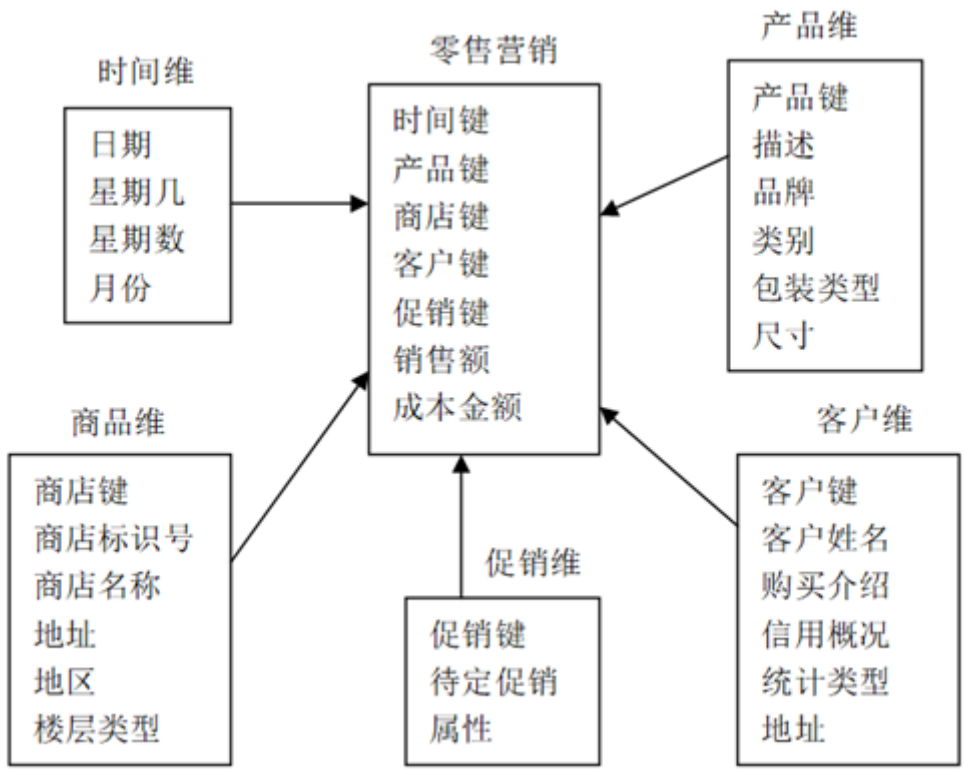
1. 规划与确定需求
2. 概念模型、逻辑模型、物理模型
3. 设计体系结构
4. 元数据设计

21 模型设计（概念——逻辑；星型模型；粒度选择）

1. 概念模型设计：
    - 确定系统边界
    - 确定主题域
2. 逻辑模型设计：
    - 概念模型到逻辑模型的转换：确定维度表、事实表
    - 粒度层次划分：早期细节、当前细节、轻度综合、高度综合
    - 关系模式定义：表名、属性、键等
    - 定义记录系统：数据仓库数据与源数据的对照记录
3. 物理模型设计：为逻辑模型设计的数据模型确定存储结构和存取方法
    - 确定数存储结构
    - 确定索引
    - 确定存放位置
    - 确定存储分配

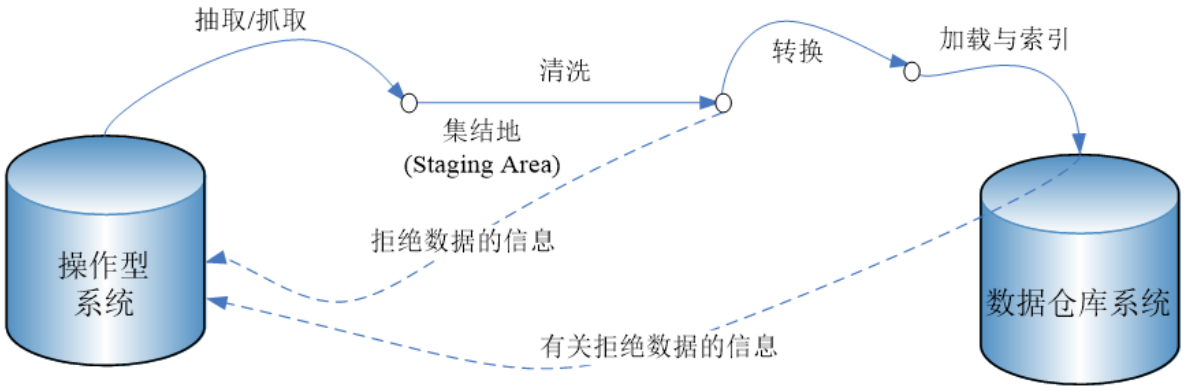
画逻辑模型(星型)





22 ETL的内容

- 数据抽取
- 数据清洗
- 数据转换：不一致数据转换、数据粒度转换、符合业务规则
- 数据装载：抽取、转换、清洗后的数据装载到数据仓库系统中
  更新元数据，注意这里要记录从操作DBMS到DW的映射关系



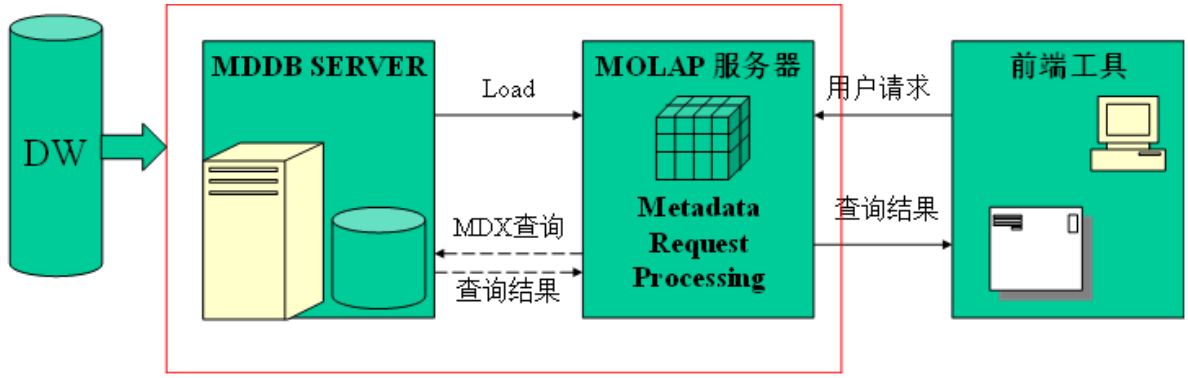
23 五种OLAP的操作，并说明每种的具体内容

- 上卷 roll up：概括数据
  沿着一个维的概念分层向上攀上，或通过维归约、对数据立方进行聚集
- 下钻 drill down：上卷逆操作
  高层概括→底层概括，不太详细→更详细的数据
  给数据增加细节，可以添加新的维到立方体
- 切片和切块 slice and dice：投影和选择
  按二维切片、三维切块，可以得到想要的数据
- 转轴pivot（或旋转 rotate)：转换立方体的视角 (维的角度
- 其他操作
  钻过 drill across：执行涉及多个事实表的查询
  钻透 drill through：钻透立方体的底层，到后端关系表，使用SQL

24 MOLAP和ROLAP的体系结构，工作原理

MOLAP

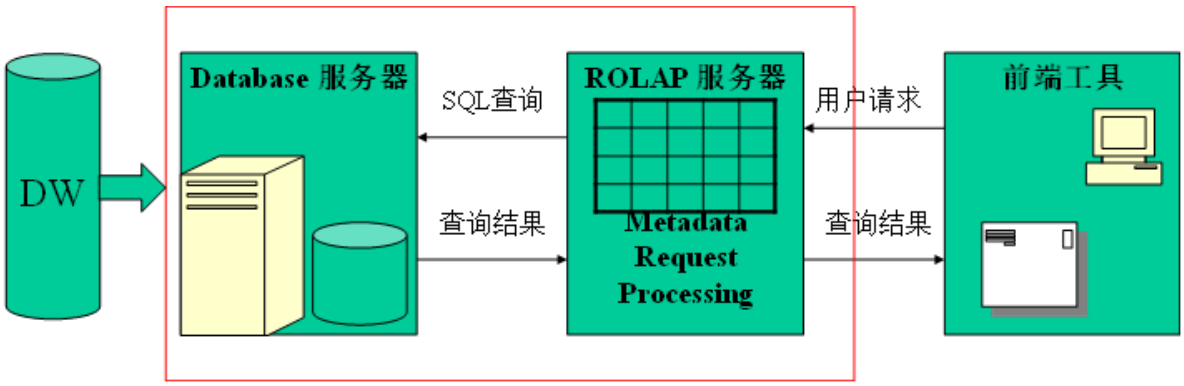
- 利用**多维数组**来存放和管理数据，数据存储在多维数据库 (Multidimensional Database)中
- 维的属性值映射成多维数组的下标
- 事先计算汇总数据，作为多维数组的值存储在数组单元中
- 用户的OLAP操作可以直接映射到多维数据库，不需要通过SQL访问



- 客户端提交分析需求到OLAP服务器，
- OLAP服务器检索MDDB数据库，得到结果返回给用户
- 优点：速度快、空间小、预处理程度高
- 缺点：灵活性、适应能力差

ROLAP

- 利用**关系数据库**来存放和管理数据，数据存储在关系数据库服务器中
- 物化使用频率高、计算工作量大的视图
- 关系数据库系统可以针对OLAP做优化



- ROLAP服务器将用户请求翻译成SQL请求，交给关系数据库服务器进行处理
- 关系表形式的查询结果 → 多维视图形式，返回用户
- 优点：关系表适合处理大量数据；更灵活、适应性好
- 缺点：查询多维数据会涉及到大量表连接运算，查询速度较慢
- 改进：采用**星型**或**雪花**组织数据，维表和事实表使用关系表存储

描述	MOLAP	ROLAP
存取速度	快	响应时间较长
存储容量	受操作系统大小限制	传统关系数据库存储，基本无限制
多维计算能力	很强	不太好，要进行表连接
维度变化适应性	需要重新建立多维数据库	适应性好
数据变化的适应性	需要重新计算(甚至重构)	适应性好
软硬件平台适应性	相对较差	适应性好

数据组织形式

RDB数据组织
关系数据库

产品名称	地区	销售量
冰箱	东北	50
冰箱	西北	60
冰箱	华北	100
彩电	东北	40
彩电	西北	70
彩电	华北	80
空调	东北	90
空调	西北	120
空调	华北	140

MDDB数据组织
多维数据库

	东北	西北	华北
冰箱	50	60	100
彩电	40	70	80
空调	90	120	140

关系表中综合数据的存放

产品名称	地区	销售量
冰箱	东北	50
冰箱	西北	60
冰箱	华北	100
冰箱	总和	210
彩电	东北	40
彩电	西北	70
彩电	华北	80
彩电	总和	190
空调	东北	90
空调	西北	120
空调	华北	140
空调	总和	350
总和	东北	180
总和	西北	250
总和	华北	320
总和	总和	750

MDDB中综合数据的存放

	东北	西北	华北	总和
冰箱	50	60	100	210
彩电	40	70	80	190
空调	90	120	140	350
总和	180	250	320	750

25 什么叫数据立方体的预计算？为什么要进行预计算？面临的问题有哪些？有哪些策略？

- 预计算：在查询之间，对不同综合程度的细节数据进行综合，并存储起来
- 原因：方体个数多，要确保快速的查询响应时间
- 问题：实时计算工作量大，全部事先计算存储量大，要进行取舍
- 预计算策略：
    - **不物化：**不预先计算任何“非基本”方体
    - **部分物化：**在所有方体集中，有选择地物化适当子集
      冰山立方体：只存放count大于某个最小支持度阈值的立方体单元
    - **全物化：**预先计算所有的方体

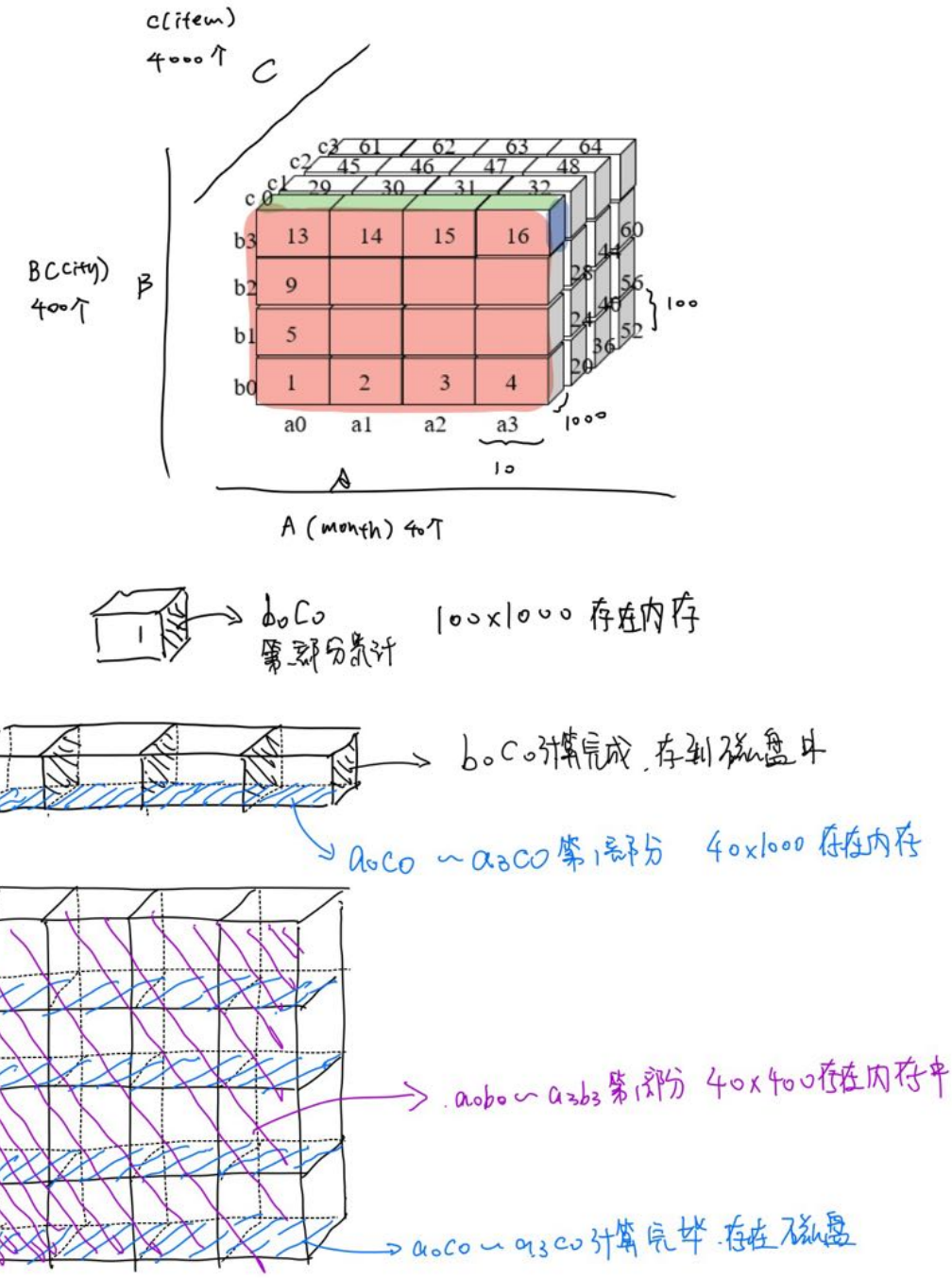
26 完整数据立方体的预计算方法

多路数组聚集

使用多维数组作为基本数据结构，计算完全数据立方体

1. 将数组分成一个装入内存的子方块
2. 根据不同维的基数，优化访问立方体单元的次序，使每个单元被访问的次数最小化

将最小的平面放在内存中，对最大的平面每次只取并计算一块



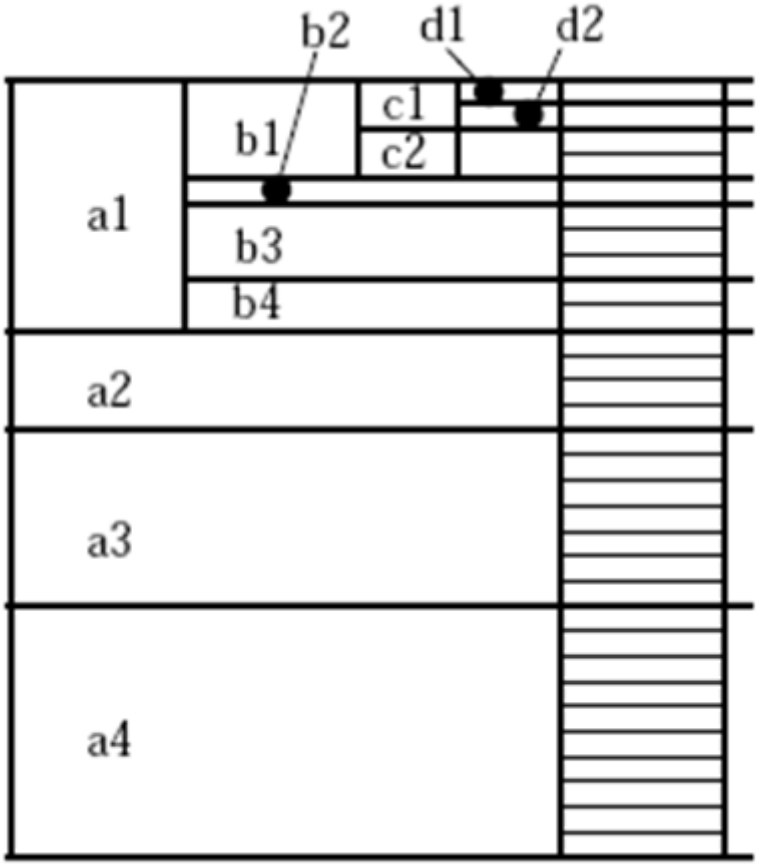
根据1到64的扫描次序，在块内存中保存所有相关的2-D平面所需的最小存储为：
40×400（用于整个AB平面）＋40×1000（用于AC平面一行）＋100×1000（用于BC平面一块)＝156，000
如图，BC为最大的面，每次只在内存中保留一块；AC为次大的面，每次在内存中保留一行；AB为最小的面，在内存中保留一正面

自底向上冰山立方体计算

从顶点立方体逐步计算到基本立方体，用于计算稀疏冰山立方体

把数据立方体中的所有元组存放在一个关系表中，假设有ABCD四个维

1. 对基本表中的所有元组进行聚集，计算出ALL定点立方体
2. 对第一维A的每个取值，把基本表划分成多个分片
3. 针对每个维A的分片，对它进行聚集，创建元组来表示划分
4. 对每个划分进行计数，判断是否满足最小支持度，不满足则减枝
5. 如果满足冰山条件，输出该划分的聚集元组，并在该划分上对下一个维度B进行BUC递归调用



应当以维基数的递减顺序进行划分：基数越大，不同值越多，剪枝的机会越大

27 什么叫数据泛化

将数据库中与任务相关的数据集，从较低概念层抽象到较高概念层

主要方法：

- 数据立方体：OLAP使用的泛化方法
  从数据仓库子集构造的数据集合，组织成 [一组维度 + 度量值] 定义的多维数据结构
- 概念描述：面向属性的泛化方法

28 面向属性的泛化方法有哪两种方法及规则

- 属性删除：对于获取到的相关数据，如果某个属性下具有大量不同值，符合以下情况，可进行属性删除
  - 该属性上没有定义相关的概念分层
  - 该属性的较高层概念已由其他属性表示
- 属性泛化：对于获取到的相关数据，如果某个属性下具有大量不同值，且该属性上存在概念分层定义，根据概念分层对该属性进行数据泛化

泛化阈值控制：不同元组的最大个数

29 频繁模式挖掘相关概念（关联规则，支持度，置信度）

- 关联规则挖掘：发现大量数据中项集之间的关联联系
  如果两项或多项属性之间存在关联，一个属性可以依据其他属性值进行预测
- 关联规则的蕴涵式：$A \Rightarrow B$ [支持度 support，置信度 confidence]
  A与B相交为空
- 支持度 = |同时包含A和B的事务| / |所有事务|
- 置信度 = |同时包含A和B的事务| / |包含A的事务|
- 强规则：同时满足最小支持度和最小置信度阈值

30 关联规则挖掘的步骤

1. 找出所有所有的频繁项集F：满足最小支持度
2. 由频繁项集产生强关联规则：满足最小置信度

31 Apriori方法（原理，例子）

见ipad

32 FP-TREE（原理，例子）

见ipad

33 为什么进行关联规则的主观性测试？有哪些指标及其特点

- 强关联规则不一定有趣：只有用户能够确定规则是否有趣，并且这种判断是主观的， 因用户而异
- 客观兴趣度量可以清除无关规则，不向用户提供

关联规则的兴趣度度量

- 客观度量：支持度、置信度
- 主观度量：用户判断
    - 是出人意料的
    - 是可以利用该规则做出某些行动的

33 序列挖掘的相关概念

见ipad

34 apriori-all算法（原理，例子）

见ipad

35 GSP算法（原理，例子）

见ipad

37 过拟合的主要原因及其解决方法

过拟合的原因：

- 数据量太小
- 训练集和测试集特征分布不一致
- 噪音数据干扰过大
- 过渡训练
- 模型太复杂

解决方案：

- 模型层面
    - 调小模型复杂度
    - 在损失函数中加入正则项来惩罚模型的参数
- 数据层面
    - 从数据源中获取更多数据
    - 扩充数据，比如通过图像平移、翻转等扩大数据数量
- 训练层面
    - 提前终止迭代

38 关于基于混淆矩阵的几个主要指标及其作用

- 每一列代表预测类别

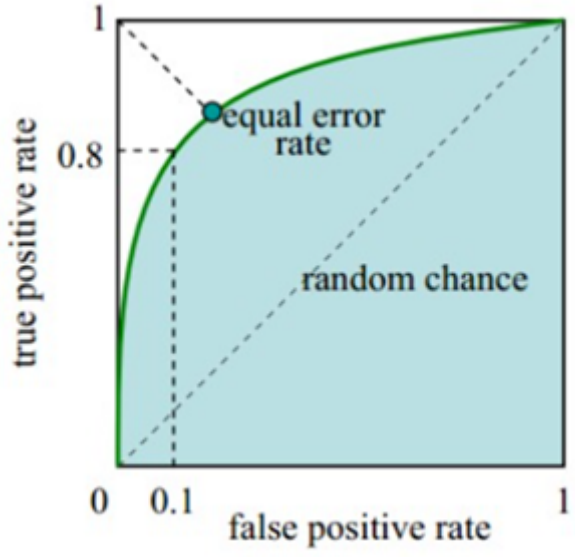
- 每一行代表真实归属类别

二分类问题的混淆矩阵		预测结果类		
		+	-	总
实际类	+	tp	fn	tp+fn
	-	fp	tn	fp+tn
	总	tp+fp	fn+tn	tp+fn+fp+tn

- 准确率：预测正确 / all
  错误率：预测错误 / all
- 真正率/召回率/敏感度 = tp / tp + fn
  正确预测为正 / 实际为正这一行的所有数据
- 真负率：tn / tn + fp
  正确预测为负 / 实际为负这一行的所有数据
- 精度/精确率/查准率 = tp / tp+fp
  正确预测为正 / 预测为正的这一列所有数据

39 ROC曲线及其特点，计算

- 纵轴：TPR True Postive Rate 真正率/召回率 TP/(TP+FN)
  横轴：FPR False Postive Rate 假正率 FP/(FP+TN)
  真正确/那一行；假正确/那一行
- 取定一个分类阈值时，算出对应的TPR、FPR，并在上图a找到对应点
  当取完不同阈值，可以得到样本对应的ROC曲线
- 当阈值从1到0慢慢移动时，假正会越来越多：因为预测成positive的“门槛降低”，被错误预测的就会增多
  TPR越高，FPR越低（ROC曲线越陡），模型性能越好



- ROC曲线上的关键点：
  - (TPR = 0, FPR = 0)：每个实例都预测为negative
  - (TPR = 1, FPR = 1)：每个实例都预测为positive
  - (TPR = 1, FPR = 0)：理想模型

- AUC：Roc曲线下方的面积 (Area under Curve)
    - [0, 1]
      理想情况：面积= 1
      随机猜测：面积 = 0.5

40 划分聚类基本思想和原理，k-means, K-medoid算法（原理，例子）
41 层次聚类基本思想和原理，AGNES, DIANA算法（原理，例子）
42 BIRCH算法相关概念，基本思想，例子
43 Chameleon基本思想和步骤
44 密度聚类相关概念（邻域，密度可达等）
45 DB-SCAN算法
46 OPTICS算法原理，例子
47 CLIQUE算法基本思想

48 什么是离群点？离群点挖掘有什么意义？主要有哪几类方法

- 离群点：在样本空间中，与其他样本点的一般行为或特征不一致
- 离群点挖掘的意义：
  少量的数据可能蕴含着重要的研究价值
  分析数据并及时发现异常，从而避免损失
- 检测离群点的方法：
    - 基于统计的方法
    - 基于距离的方法
    - 基于密度的方法
    - 基于聚类的方法

49 基于距离和密度的离群点发现方法（相关概念，原理，例子）
50 基于聚类的离群点发现方法（原理，例子）

51 基于物品的协同推荐算法（原理，例子）
52 基于用户的协同推荐算法（原理，例子）
53 基于内容的推荐算法（原理）

- 根据推荐物品或内容的元数据，发现物品或者内容的相关性，然后基于用户以往的喜好记录，推荐给用户相似的物品。
- 为每个物品构建一个物品的属性资料
  将自然语言描述的Item Profiles转换成0,1矩阵
- 为每个用户构建一个用户的喜好资料（User Profile）