数据挖掘复习笔记

1.填空与简答

OLTP: Online transaction processing, 联机事务处理

OLAP: Online analytical processing, 联机分析处理

KDD: Knowledge Discovery in Database, 数据库知识发现

BI: Business Intelligence, 商业智能

ETL: Extract-Transform-Load, 用来描述将数据从来源端经过萃取（extract）、转置（transform）、加载（load）至目的端的过程

数据仓库的特征:
1.*面向主题的*:围绕主题组织, 如消费者, 产品, 销售量
2.*集成的*:集成多个的, 异构的数据源
3.*时变的*:数据存储从历史的角度提供信息. 数据仓库中的关键结构都隐式或显式地包含时间元素.
4.*非易失的*:与操作数据库分隔存储, 操作数据库的数据更新不在数据仓库环境出现, 不需要事务处理，数据恢复以及并发控制机制, 一般仅仅有初始装载和数据访问两种操作.

数据仓库模型: 数据仓库基于多维数据模型，以数据立方体的形式对数据进行观察. *什么是数据立方体?*:数据立方体允许从多为对数据建模和观察, 它由维和事实定义. *维表*：如维item (item_name, brand, type)，或维time(day, week, month, quarter, year), *事实表*包含*事实*的名称和度量, 以及每个相关维表的码. *事实*是数值度量的, 比如, 数据仓库sales的事实包含dollars_sold, units_sold, amount_budgeted.

数据仓库的多层结构:
1.*仓库数据服务器*
2.*OLAP服务器*
3.*前端客户层*

数据仓库的视图:
1.*自顶向下视图*
2.*数据源视图*
3.*数据仓库视图*
4.*商务查询视图*

度量的分类:
度量其实就是一个数值函数
书本的说明太复杂, 我尝试用自己的语言组织一下, 如果有错, 立马跟我说, 或者在评论里留一下
1.*分布式的*: 将整体划分为一个个小子集计算再合并, 和直接计算整体是一样的, 比如说sum(), count()
2.*代数的*: 代数度量中有多个分布式的度量作为参数, 比如avg()=sum()/count()
3.*整体的*: 数据必须从整体的数据集中得来, 比如说中位数mid(), 排名rank()等

聚类分析常用的数据结构:
数据结构
1.数据矩阵: p个变量, n个对象, 得到n*p的矩阵

2.相异度矩阵: 存储所有成对的n个对象的临近度, 是一个n*n的单模矩阵

两种学习模型(这个我不是很肯定, 知道的话在评论留一下):
*描述型*和*预测型*, 前者以简洁的方式表达数据中存在的一些有意义的性质, 后者通过对所提供数据集应用特定方法分析所获得的一个或一组数据模型, 并将该模型用于预测未来新数据的有关性质

为什么需要数据预处理及其主要内容:
为什么需要数据预处理: 从现实世界得来的数据可能会不完整, 含噪声, 或者是不一致, 如果不对数据进行预处理, 就会影响分析和预测
数据预处理主要内容:
1.数据清洗: 填充遗失的数据, 平滑噪声数据, 辨识或删除孤立点, 解决不一致性问题
2.数据集成: 对多个数据库，数据立方或文件进行集成
3.数据变换: 规范化与聚集(Normalization and aggregation)
4.数据约简: 得到数据集的压缩表示，它小的多，但能够产生同样的(或几乎同样的)分析结果
5.数据离散化: 特别对数字值而言非常重要

挖掘的知识类型:
1.概念/类描述: 用汇总的,简介的和精确的方式描述给个类和概念
2.挖掘频繁模式, 关联和相关: 从给定的数据集中发现频繁出现的项集模式知识
3.分类和预测: 找出一组能够描述数据集合典型特征的模型(或函数), 以便能够分类识别未知数据的归属或类别,即将未知事例映射到某种离散类别之一
4.聚类分析: 聚类分析数据对象不考虑已知的类标号, 对象根据最大化类内部的相似性, 最小化类之间的相似性的原则进行聚类和分组
5.离群点分析: 发现数据中与数据的一般行为或模型不一致的数据对象
6.演变分析: 描述行为随时间变化的对象的规律和趋势, 并对其建模

常见的OLAP操作:
1.上卷(从城市到国家)
2.下钻(从季度到月)
3.切片(time=Q1)
4.切块(time=Q1 and item="computer" and location="toronto")
5.转轴: 可视化操作, 转动数据的视角, 提供数据的替代表示

关联规则的确定性度量与实用性度量:
1.实用性度量: 支持度(support) 2.确定性度量: 置信度(confidence)

```
normal//例子
computer=>antivirus_software[support=2%, confidence=60%]
//support=2%表示所分析的所有事物中的2%同时购买计算机和杀毒软件,
//confidence=60%表示购买了计算机的顾客60%也购买了杀毒软件
normal
```

数据立方的两种表
RDBMS中, 以维表和事实表两种表的类型, 记录多维数据, 前者记录多维数据的坐标轴, 后者记录多维数据各维度的具体数值, 二者之间通过关系表的外键连接, 共同构成多维数据立方体

```
normal//例子
维表：如维item (item_name, brand, type)，或维time(day, week, month, quarter, year)
事实表包含度量（measures）：如销售额以及每个相关维表的关键字。
normal
```

数据挖掘在互联网, 移动互联网的应用:
这个...自己上知乎看吧

知识发现过程的主要步骤(自己组织的, 有错留个评论)
1.数据预处理: 进行数据清洗, 数据变换和数据规约等
2.建立数据仓库: 建立特定于组织或者企业的数据仓库, 它独立于操作数据库
3.提取与任务相关的数据
4.使用数据挖掘算法进行数据挖掘
5.对挖掘出来的模式进行评估
6.得到知识

OLTP与OLAP的主要区别:
OLTP: 是传统关系数据库的主要任务, 日常操作比如有: 购买, 存货, 财务等
OLAP: 是数据仓库的主要任务, 为数据分析语决策提供支持

为什么需要构建单独隔离的数据仓库:
1.有助于性能, 操作数据库对已知的任务和负载进行设计和优化, 而数据仓库的查询通常是复杂的, 涉及大量的数据组汇总计算, 可能需要特殊的基于多维视图的数据组织,存取方法和实现方法, 操作数据库进行OLAP查询, 可能会大大降低操作任务的性能
2.操作数据库支持多事务的并发处理, 需要加锁和日志等并发控制和恢复机制, 以确保一致性和事务的鲁棒性. 通常, OLAP查询只需要对汇总和聚集数据记录进行只读访问, 如果将并发控制和恢复机制用于这种OLAP操作, 就会危害并行事务的运行, 从而大大降低OLAP系统的吞吐量
3.数据仓库和操作数据库中数据的结构,内容和用法都不相同. 决策支持需要大量的历史记录, 然而操作数据库一般不维护历史记录. 另外, 决策支持需要将来自异种源的数据统一(如聚集和汇总), 产生高质量的,纯净的和集成的数据, 相比之下, 操作数据库只维护详细的原始数据(如事务), 这些数据在进行分析之前需要统一

数据预处理(缺失数据)方法:
数据预处理有*数据清洗*,*数据集成*,*数据变换*和*数据离散化*.
其中, 进行数据清洗的时候, 要处理缺失的数据, 常用的方法有:
1.忽略元组: 除非元组有多个属性缺少值, 否则该方法不是很有效
2.人工填充: 费时费力
3.自动填充:

```
normal使用一个全局常量填充;
该属性的平均值;
使用与给定元组属同一类的所有样本的该属性的平均值;
使用最可能的值: 使用基于推导的方法, 如Bayesian公式或决策树
normal
```

数据仓库的设计模式(我找不到*设计模式*, 只找得到3个模式, 知道的在评论里留一下):
1.星型模式: 一个事实表以及一组与事实表连结的维表
2.雪花模式: 星型模式的变种, 其中某些维表是规范化的, 因而把数据进一步分解到附加的表中
3.事实星座: 多个事实表分享共同的维表, 这种模式可以看作星型模式的集合, 因此成为星系模式或事实星座

三种度量函数的定义(这个和上面的概念是不是有重复???):
1.分布式度量: 是一种可以通过如下方法计算的度量: 将数据集划分为较少的子集, 计算每个子集的度量, 然后合并计算结果, 得到原数据集的度量值, 比如说*sum()*和*count()*;
2.代数度量: 是可以通过应用一个代数函数于一个或多个分布度量计算的度量, 比如*average()*, 因为它可以等于*sum()/count()*
3.整体度量: *必须对整个数据集计算的度量*, 它不能通过将给定数据划分成子集计算再合并来获得, 比如说*medium()(中位数)*;

分箱平滑:
先对数据进行排序, 然后把它们划分到箱;
然后通过箱平均值,箱中值等进行平滑.

无监督离散化(如分箱), 有监督离散化(如基于熵):
什么是*离散化*?: 数据离散化技术可以用来减少给定连续属性值的个数
什么是*无监督*和*有监督*?: 如果离散化过程使用了类信息, 则成为有监督离散化, 否则就是无监督的.
无监督离散化:
1.分箱: 分箱是一种基于箱的指定个数自顶向下的分裂技术. 比如, 通过使用等宽或等频分箱, 然后用箱均值或中位数替换箱中的每个值, 可以将属性值离散化. 应当注意, 上面所说的分箱平滑是用于数据平滑, 这里的分箱用于数据的离散化, 虽然方式都是相同.
2.直方图分析: 直方图将属性A的值划分成不相交的区间(称作桶). 在*等宽*直方图中, 将值分成相等的划分或区间. 在*等频*直方图中, 理想的使得每个划分包含相同个数的数据元组.
有监督离散化: 基于熵: 给定一个样本集合**S**, 如果边界值**T**把**S**划分成两个区间**S**1与**S**2, 则划分后的熵为 I(S,T)=|S1|/|S|*Ent(S1)+|S2|/|S|*Ent(S2), 选择某一边界**T**的准则是: 它使其后划分得到的信息增益(上式)最大. 这个过程队规地用于所得到的划分, 直到满足某个终止条件.

评估分类器准确率的方法:
1.**保持方法和随机子抽样**: *保持*方法指将给定数据随机地划分成两个独立的集合: *训练集*和*检验集*, 训练集用于导出模型, 其准确率用检验集估计. *随机子抽样*是保持方法的一种变形, 它将保持方法重复k次, 总准确率估计取每次迭代准确率的平均值.
2.**交叉确认**: 在*k折交叉确认*中, 初始数据随机划分成k个互不相交的子集或"折" D1,D2,...,Dk, 每个折的大小大致相等. 训练和校验k次. 在第i次迭代, 划分Di用作检验集, 其余的划分一起用来训练模型. 对于分类, 准确率估计是k次迭代正确分类的总数除以初始数据中的元组总数. 对于预测, 误差估计可以用k次迭代的总损失除以初始元组总数来计算. *留一*是k折交叉确认的特殊情况, 其中k设置为初始元组数, 也就是说, 每次只给检验集留出一个样本. 在*分层交叉确认*中, 折被分层, 使得每个折中元组的类分布与在初始数据中的大致相同.
3.**自助法**: 从给定的训练元组中有放回均匀抽样. 也就是说, 每当选中一个元组, 它等可能地被再次选中并再次添加到训练集中. (可以自己去看看.632自助法, p238)

基于规则的分类器:
要弄懂基于规则的分类器, 首先要弄懂什么是规则, 什么是基于规则的分类法.
基于规则的分类法使用一组IF-THEN规则进行分类.
一个**IF-THEN规则**是一个如下形式的表达式: IF条件THEN结论. IF部分成为前件或前提, THEN部分则成为结论.
某规则的**覆盖率**和**准确率**: 假设Covers是规则覆盖的元组数, Corrects是规则正确分类的元组数, Num是总的元组数, 那么**覆盖率**就是:coverage=Covers/Num, **准确率**是: accuracy=Corrects/Covers.
基于规则的分类器:
顺序覆盖算法, 它可以直接从训练数据提取IF-THEN规则(即不必产生决策树). 流行的算法一般有**AQ**, **CN2**和**RIPPER**, 算法的一般决策: 一次**学习一个规则**, 每当**学习到一个规则**, 就删除该规则覆盖的元组, 并对剩下的元组重复该过程. 要明白什么是**学习规则**, 要详细看看P210~P212, 那里有说到如何学习一个规则, 以及哪个规则才是好规则. 十分长, 我就不贴在这里了.

不同类型属性之间的相似性计算:
这个点我只找得到P253的相异度计算, 所以我就贴关于相异度计算的了, 它的概念非常多, 大家要记得详细地看一下书.
1.**区间标度变量**: 是一种粗略线性标度的连续度量. 典型的例子包括重量和高度, 经度和纬度坐标以及气温. 如果要避免域值过大或过小影响聚类结构, 就要进行度量标准化. 区间标度变量描述的对象间的相异度(或相似度)一般基于每对对象间的距离计算. 关于距离计算有很多种方法, 详情要查阅P254~P255, 常用的有欧几里得距离.
2.**二元变量**: 它只有两种状态, 即0(不出现)和1(出现).

	1	0	*sum*
1	*q*	*r*	*q+r*
0	*s*	*t*	*s+t*
sum	*q+s*	*r+t*	*p*

q:对象i和j都为1的变量数目, r是在对象i值为1, 但对象j值为0的变量数目, s,t同理, p=q+r+s+t.
二元变量有对称和非对称的概念, 如果两个状态有相同的权重和价值, 就是对称的, 比如男和女. 对称二元相异度为: d(i,j)=(r+s)/p. 但是如果是非对称情况下, 我们会比较关心其中一个状态, 比如检测癌症或者HIV病毒, 1为阳性, 那么两个变量都为0对我们来说并没有太大价值, 这时相异度就变成了: d(i,j)=(r+s)/(q+r+s).
3.**分类变量**: 其实就是二元变量推广到多个状态的情况, 太无聊了, 我都懒得写了.
4.**序数变量**: 它和区间标量度量有点像, 但我们不关心它的值的大小, 而是它的值的相对次序, 所以如果某个序数变量f有M个状态, 那我们就用变量f的值对应的秩(第几个状态)代替该值, 但是多个序数变量之间的状态数可能是不一样的, 所以我们要再将它们的变量值域映射到[0.0, 1.0], 秩随着改变, 最后用欧几里得距离计算相异度. 详情请阅读P257~P258.
5.**比例标度变量**: 在非线性的刻度(例如指数刻度)取正的度量值.处理这种变量的方法有3种: 将它视为区间标度度量; 进行对数变换; 将它看作序数数据.
处理之后再用欧几里得距离公式计算相异度.

常见的聚类优化目标(最小化聚类间相似性, 最大化聚类内相似性等)及k-center,k-cluster,k-means聚类算法及其时间复杂度:
k-means:

```
normalk: 簇的数目;
D: 包含n个对象的数据集;
输出: k个簇的集合;
方法:
从D中任意选择k个对象作为初始簇中心;
repeat
    根据簇中对象的均值, 将每个对象(再)指派到最相似的簇;
    更新簇均值, 即计算每个簇中对象的均值;
until 不再发生变化
normal
```

k-means的计算复杂度是O(nkt), n是对象的总数, k是簇的个数, t是迭代的次数. **k-cluster**和**k-center**:

```
normalk: 结果簇的个数;
D: 包含n个对象的数据集合;
输出: k个簇的集合;
方法:
从D中任意选择k个对象作为初始化的代表对象或种子;
repeat
    将每个剩余对象指派到最近的代表对象所代表的簇;
    随机选择一个非代表对象o_random; // 即余下的n-k个对象中随机选一个
    for j=1~k
        计算用o_random交换代表对象o_j的总代价S; // S其实就是交换后和交换前绝对误差的差
        if S<0, then
            用o_random替换o_j, 形成新的k个代表对象的集合;
            break;
until 不发生变化
normal
```

由以上代码可知, for语句每次迭代, 都要计算绝对误差, 有k次循环, 所以k-center复杂度为O(tk(n-k)^2), t为迭代次数, k为簇的数目, n为总的对象数.

PageRank与HITS的基本思想,算法,及各自优缺点:
PageRank:
基本思想: 将网页x指向网页y的链接视为x给y的一张投票, 然而PageRank不仅仅考虑网页得票的绝对数目, 它还分析投票者本身的权威性. 来自权威网页的投票能够提升被投票网页的权威性.
算法:

- 网页i的权威性得分 (即i的 PageRank 值，用$P(i)$表示) 定义为所有指向i的网页的PageRank值之和。
- 既然一个网页可能指向多个网页，因此它的值应该被其指向的多个网页所共享。
- 把Web视为一个有向图 $G=(V,E)$，其中V表示顶点集合（即网页集合），一条有向边$(i,j)\in E$当且仅当网页i指向网页j，n为总的网页数。 网页i的$P(i)$定义为: $$P(i)=\sum_{(j,i)\in E}\frac{P(j)}{O_j},$$

O_j是网页j的出边数

改进后:

$$P(i)=(1-d)+d\sum_{(j,i)\in E}\frac{P(j)}{O_j}$$

- 参数 $0<d<1$也称为阻尼因子（ damping factor），在原始文献中设置为 $d=0.85$. 优缺点:
- 防欺骗
  - 网页所有者难以设置其它重要网页指向他自己的网页.
- PageRank 值独立于查询，是一种全局度量.
  - PageRank 值是通过所有网页计算得到并加以存储，而不是提交查询时才计算.
- 缺点： 不能区分全局重要性网页和查询主题重要性网页.

HITS:
权威性网页: 具有很多的入边;
汇集性网页: 具有很多的出边;
HITS有一个非常关键的假设: 一个好的汇集性网页指向许多权威性网页, 一个好的权威性网页被许多好的汇集性网页指向.
具体算法：可利用上面提到的两个基本假设，以及相互增强关系等原则进行多轮迭代计算，每轮迭代计算更新每个页面的两个权值，直到权值稳定不再发生明显的变化为止.
HITS计算这两个权值之前要先收集网页, 它的收集方法是:
将关键词发送给检索系统, 收集回前t个网页, 这t个网页成为root set;
对root set进行扩展, 凡是和root set中的网页有链接的, 无论是指出还是指入, 都加进来, 形成base set.
HITS的优缺点:

- 优点: 根据查询进行排序，可能会返回更相关的权威性和汇集性网页.
- 弱点:
  - 容易被欺骗. 一个网站开发者很容易在自己的网页中加入许多出边.
  - 主题漂移（Topic drift）. 许多扩展网页可能与主题并不相关.
  - 查询效率: 动态抓取网页进行扩展、特征值计算