关联规则挖掘过程报告

在数据挖掘的知识模式中，关联规则模式是比较重要的一种。关联规则的概念由Agrawal、Imielinski、Swami 提出，是数据中一种简单但很实用的规则。关联规则模式属于描述型模式，发现关联规则的算法属于无监督学习的方法。

* 数据集说明（数据集二）

本数据集是csv文件，

* 数据预处理

为所有属性设置属性名分别为a1、a2、……、a43。在这些属性中，大体分为两种：标称属性和数值属性，其中数值属性不适合进行关联规则挖掘，因此，需要将数值属性转变为yes or no。首先对每个数值属性给定阈值，高于阈值的认定为“yes”，低于阈值的认定为“no”，经过这一步的转换，所有属性都变为布尔型值，适合做关联规则挖掘。

为了进一步转换数据，我们将每一个数据项进行如下转换,比如:

|a1|a2|a3|a4|a5|a6|a7|a8|

|no|yes|no|yes|yes|yes|yes|no|

转换为： t1=[a2,a4, a5,a6, a7],方便Aprior算法的处理。二值矩阵转换成事务型的列表数据，如下：

* 挖掘算法

Apriori核心算法过程如下：

* 过单趟扫描数据库D计算出各个1项集的支持度，得到频繁1项集的集合。
* 连接步：为了生成，预先生成,由2个只有一个项不同的属于的频集做一个（k-2）JOIN运算得到的。
* 剪枝步：由于是的超集，所以可能有些元素不是频繁的。在潜在k项集的某个子集不是中的成员是，则该潜在频繁项集不可能是频繁的可以从中移去。
* 通过 单趟扫描数据库D，计算中各个项集的支持度，将中不满足支持度的项集去掉形成。

　　通过迭代循环，重复步骤2～4，直到有某个r值使得为空，这时算法停止。在剪枝步中的每个元素需在交易数据库中进行验证来决定其是否加入，这里的验证过程 是算法性能的一个瓶颈。这个方法要求多次扫描可能很大的交易数据库。可能产生大量的候选集，以及可能需要重复扫描数据库，是Apriori算法的两大缺点。

- 支持度

支持度(support)=(X,Y).count/T.count, (T是事务总和，(X,Y).count是X、Y同时出现的次数)

- 置信度

置信度(confidence)=(X,Y).count/X.count

- 期望置信度

期望置信度=Y.count/T.count

* 去冗余

总体来说，规则2 是 规则1 的衍生规则，如果规则2 和 规则1 有相同的提升度或者比规则1 更低的提升度，那么规则2 就被认为是冗余的。

- 借用Aprior算法的思想，可以得到两条有用的去除冗余的规则。

- 如果一个集合是频繁项集，则它的所有子集都是频繁项集。举例：假设一个集合{A,B}是频繁项集，即A、B同时出现在一条记录的次数大于等于最小支持度min\_support，则它的子集{A},{B}出现次数必定大于等于min\_support，即它的子集都是频繁项集。

- 如果一个集合不是频繁项集，则它的所有超集都不是频繁项集。举例：假设集合{A}不是频繁项集，即A出现的次数小于min\_support，则它的任何超集如{A,B}出现的次数必定小于min\_support，因此其超集必定也不是频繁项集。

除此之外，我们主要挖掘的是特征都结果的映射，可以把a开头的属性和d开头的属性分开，因为某种程度上来说a开头的属性是特征属性，而d开头的属性是结果属性。这样，我们就可以得到一系列从特征到结果的规则，形如X->Y。

* lift评价

提升度是可信度与期望可信度的比值，提升度大于1表示正相关，小于1表示负相关，等于1表示不相关。lift评价可以弥补置信度、支持度自身的不足，使得评价更为合理。利用这个值可以对关键规则进行排序，在一定程度上也是一个规则优化挑选的过程。