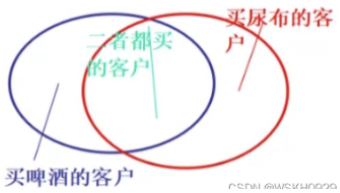
一、关联规则概述

1.1 关联规则引入

下面用一个故事来引出关联规则：

- 在美国，一些年轻的父亲下班后经常要到超市去买婴儿尿布，超市也因此发现了一个规律，在购买婴儿尿布的年轻父亲们中，有30%～40%的人同时要买一些啤酒。超市随后调整了货架的摆放，把尿布和啤酒放在一起，明显增加了销售额。
- 若两个或多个变量的取值之间存在某种规律性，就称为关联
- 关联规则是寻找在同一个事件中出现的不同项的相关性，比如在一次购买活动中所买不同商品的相关性。
- “在购买计算机的顾客中，有30%的人也同时购买了打印机”



关联分析又称关联挖掘，就是在交易数据、关系数据或其他信息载体中，查找存在于项目集合或对象集合之间的频繁模式、关联、相关性或因果结构。属于无监督学习。

关联分析（关联规则学习）：从大规模数据集中寻找物品间的隐含关系被称作关联分析或关联规则学习

1.2 关联规则相关概念介绍

1.2.1 样本、事务、项集、规则

关联规则中的数据集结构一般如下所示：

编号	牛奶	果冻	啤酒	面包	花生酱
T_1	1	1	0	0	1
T_2	0	1	0	1	0
T_3	0	1	1	0	0
T_4	1	1	0	1	0
T_5	1	0	1	0	0
T_6	0	1	1	0	0
T_7	1	0	1	0	0
T_8	1	1	1	0	1
T_9	1	1	1	0	0

样本属性

样本

- 一个样本称为一个“事务”
- 每个事务由多个属性来确定，这里的属性称为“项”
- 多个项组成的集合称为“项集” **如{牛奶、果冻、面包}就可以称为一个项集**

> 关于项集（多个项组成的集合）：
>
> - { 牛奶 } 是 1-项集
> - { 牛奶，果冻 } 是 2-项集；
> - { 啤酒，面包，牛奶 } 是 3-项集

> X==>Y含义（规则）：
>
> - X和Y是项集
> - X称为规则前项
> - Y称为规则后项

> 事务：即样本，一个样本称为一个事务。事务仅包含其涉及到的项目，而不包含项目的具体信息
>
> - 在超级市场的关联规则挖掘问题中事务是顾客一次购物所购买的商品，但事务中并不包括这些商品的具体信息，如商品的数量、价格等

1.2.2 支持度、置信度

> 支持度(support)：一个项集或者规则在所有事务中出现的频率，$\sigma(\mathrm{X})$：表示项集X的支持度计数
>
> - 项集X的支持度：$s(X)=\sigma(\mathrm{X})\div N$
> - 规则X==>Y表示物品集X对物品集Y的支持度，也就是物品集X和物品集Y同时出现的概率
> - 假设某天共有100个顾客到商场买东西，其中30个顾客同时购买了啤酒和尿布，那么上述的关联规则的支持度就是30%

> 置信度(confidence)：确定Y在包含X的事务中出现的频繁程度。$c(X\longrightarrow Y)=\sigma(X\cup Y)\div\sigma(X)$
>
> - 条件概率公式：$P(Y|X)=P(XY)\div P(X)$
> - 置信度反映了关联规则的可信度，即购买了项目集X中的商品的顾客同时也购买了Y中商品的概率
> - 假设购买薯片的顾客中有50%也购买了可乐，则置信度为50%

下面举一个例子，来更深层次的理解支持度和置信度：

交易ID	购买的商品
1	A,B,C
2	A,C
3	A,D
4	B,E,F

(X , Y)==>Z :

- 支持度:交易中包含{X 、 Y 、 Z}的可能性
- 置信度:包含{X 、 Y}的交易中也包含Z的条件概率

计算 A==>C 的支持度和置信度：

- 支持度：即同时购买了商品A和C的顾客的比率 = $2\div 4=50\%$
- 置信度：即在购买了商品A的顾客中，购买了商品C的比率 = $2\div 3=66.7\%$

计算 C==>A 的支持度和置信度：

- 支持度：即同时购买了商品C和A的顾客的比率（其实和A==>C的支持度是一样的） = $2\div 4=50\%$
- 置信度：即在购买了商品C的顾客中，购买了商品A的比率 = $2\div 2=100\%$

我们一般可以用 `X==>Y（支持度，置信度）` 的格式表示规则的支持度和置信度，具体如下所示：

- A==>C（50%，66.7%）
- C==>A（50%，100%）

一般地，我们会**定义最小支持度（minsupport）和最小置信度（minconfidence）**，若规则X==>Y的支持度分别大于等于我们定义的最小支持度和最小置信度，则称关联规则X==>Y为**强关联规则**，否则称为**弱关联规则**。我们通常会把 注意力 放在强关联规则上。

1.2.3 提升度

提升度（lift）：物品集A的出现对物品集B的出现概率发生了多大的变化

- $lift(A ==> B) = confidence(A ==> B) \div support(B) = P(B|A) \div P(B)$
- 假设现在有1000个顾客，其中500人买了茶叶，买茶叶的500人中有450人还买了咖啡。那么可以计算得 $confidence(茶叶 ==> 咖啡) = 450 \div 500 = 90\%$，由此，可能会认为喜欢喝茶的人往往喜欢喝咖啡。但是，如果另外没有购买茶叶的500人中也有450人买了咖啡，同样可以算出置信度90%，得到的结论是不爱喝茶的人往往喜欢喝咖啡。这与前面的结论矛盾了，由此看来，实际上顾客喜不喜欢喝咖啡和他喜不喜欢喝茶几乎没有关系，两者是相互独立的。此时，我们就有提升度这一指标来描述这一现象。
  在这个例子中，$lift(茶叶 ==> 咖啡) = confidence(茶叶 ==> 咖啡) \div support(咖啡) = 90\% \div [(450 + 450) \div 1000] = 1$
- 由此可见，提升度弥补了置信度的这一缺憾，如果提升都等于1，那么X与Y独立，X对Y的出现的可能性没有提升作用。提升度越大（lift > 1），则表明X对Y的提升程度越大，也表明X与Y的关联性越强。

1.2.4 所有指标的公式

measure	definition	interpretation
support	$\text{supp}_T(A \Rightarrow B)$	$P(A \cap B)$
confidence	$\frac{\text{supp}_T[A \Rightarrow B]}{\text{supp}_T[A]}$	$P(B \mid A)$
lift	$\frac{\text{conf}_T[A \Rightarrow B]}{\text{supp}_T[B]}$	$\frac{P(B \mid A)}{P(B)}$
leverage	$\text{supp}_T[A \Rightarrow B] - \text{supp}_T[A]\ \text{supp}_T[B]$	$P(A \cap B) - P(A)\ P(B)$
conviction	$\frac{1-\text{supp}_T[B]}{1-\text{conf}_T[A \Rightarrow B]}$	$\frac{1-P(B)}{1-P(B \mid A)}$

measure	min value, incompatibility	value at independance	max value, logical rule
support	0	$\text{supp}_T(A)\ \text{supp}_T(B)$	$\text{supp}_T(A)$
confidence	0	$\text{supp}_T(B)$	1
lift	0	1	$\frac{1}{\text{supp}_T(B)}$
leverage	$-\text{supp}_T(A)\ \text{supp}_T(B)$	0	$\text{supp}_T(A)\ (1 - \text{supp}_T(B))$
conviction	$1 - \text{supp}_T(B)$	1	∞

1.3 常用关联算法

算法名称	算法描述
Apriori	关联规则最常用也是最经典的挖掘频繁项集的算法，其核心思想是通过连接产生候选项及其支持度然后通过剪枝生成频繁项集。
FP-Tree	针对Apriori算法的固有的多次扫面事务数据集的缺陷，提出的不产生候选频繁项集的方法。Apriori和FP-Tree都是寻找频繁项集的算法。
Eclat算法	Eclat算法是一种深度优先算法，采用垂直数据表示形式，在概念格理论的基础上利用基于前缀的等价关系将搜索空间划分为较小的子空间。
灰色关联法	分析和确定各因素之间的影响程度或是若干个子因素（子序列）对主因素（母序列）的贡献度而进行的一种分析方法。