读书报告

09017244
郑健雄

一、自己提出的问题的理解:

1、 提出的问题 1: 为什么对于使用 MIS 的 item 来说，向下闭包性质不再适用。
讨论后的理解: 可以找出一个反例，因为如果删除掉最小 MIS 的 item 的话，其子集虽然满足 support>=最小 MIS 的条件，但是由于子集中没有最小 MIS,其不一定能满足自身现在的最小 MIS.

2、 提出的问题 2: 在 MS-Apriori 算法中, level2-candidate-gen(L, ϕ)和 MScandidate-gen(Fk−1, ϕ）的内容有类似的地方，也有不同的地方，为什么两种 function 的内容有所不同。

讨论后的理解: 这两个函数很明显的是参数不同，2-level 使用了 L，其目标在于产生最基本的频繁集，并且保证最小的 item 在最前面，这是后面一切处理的基础，所以需要另外提出, 此外, level-2 在算法组织上也提升了整体的效率, 省略了没有必要的剪枝操作。而 candidate-gen 的部分则和之前原始算法类似，只是对例外情况进行了处理。

二、别人提出的问题的理解:

3、 问题 3: 为什么在 MScandidate 函数中每次循环中使|sup(ik-1) − sup(i'k−1)| ≤ ϕ 就能令一个 itemset 中 maxsup 与 minsup 的差值在 ϕ 的范畴内?求严谨数学说明。
自己的理解: 实际上这两个项目同属两个不同的频繁集, 而同一个频繁集中的不同项目，一定满足阈值的约束，所以在两个项目集合并的时候，只有这两个新的项目之间没有检验过阈值，其他部分都被检验过了，所以没有检验的必要。

4、 问题 4:
反证法证明 2.4.3 的 lemma 的时候，为什么

occur when $a \in X$. Since $a \in X$ and $X \subset f$, a must have the lowest MIS value in X and X must be a frequent itemset, which is ensured by the MS-Apriori algorithm. Hence, the support count of X is recorded. Since f is a

a 属于 X 那么 X 就一定是一个 frequent itemset。
自己的理解: 实际上 X 是我们生成规则时使用频繁集的一个子集，而频繁集本身是满足最小 MIS 的, 那么根据 2.2 的公式说明, 其子集的 support 一定大于等于其全集的 support, 那么其也一定满足最小 MIS,而因为其包含最小 MIS,所以符合成为频繁集的条件，confidence 的关系则可以根据 confidence 的公式说明子集的 confidence 满足条件。

三、读书计划

1、本周完成的内容章节: 3.3 4.1-4.3
2、下周计划: 第五章 5.1

四、读书摘要及理解

1、读书摘要及理解

第二章:

在 association rule 的挖掘中，学习了 support 和 confidence 的概念，同时学习了 Apriori 算法的内容，其可以限定一个 minsup 和 minconf，挖掘出有价值的 rule。但是由于 rare item 问题存在，使得那些重要但是频率很低的 item 的相关规则难以被挖掘，所以又引入了

MIS 的概念和 maximum support difference,分别用于学习低频率规则以及限制 support 差异。对 Apriori 算法进行了衍生，并且通过将最小 MIS item 放在开始，处理了向下闭包性质不满足问题，产生了 MS-Apriori 算法。其与 Apriori 算法十分类似。

之后学习了 class association rule 也就是 CAR mining,其挖掘算法 CAR-Apriori 和 Apriori 十分类似，只是将 class 本身作为一个单独类进行考虑，而不是一个 item，且产生的结果只能是一个 class，所以在细节上有所不同。其也可以引入 MIS，实际上是我们学习的基础 Apriori 和 MS-Apriori 算法的延申。

第三章:
1. 监督学习基本概念：第三章的主要内容是讲监督学习，监督学习的目标是根据已有的数据产生一个分类器，使其可以对未来的数据进行分类。监督学习的一般流程是将数据集分为训练集和测试集，使用机器学习算法通过训练集构筑模型，再通过测试集对其准确性进行评估。
可以认为对于给定的数据集 D,给定的工作 T 以及给定的测试标准 M,如果计算机系统通过学习数据集 D,其工作 T 在测试标准 M 下有所改善，则称之为进行了学习。
2. 决策树模型：决策树模型一般会寻找能够让分类结果更为纯净或者说错误情况最小的分类属性进行分类，使得子节点的不纯净程度尽可能下降，在决策树模型中使用信息增益的概念来理解。决策树模型的构建是一个递归的过程。
一般我们使用 C4.5 算法进行决策树构建，其引入了信息论中的熵概念来表征不确定度或者说不纯粹度。具体寻找分类属性使用了信息增益或者信息增益比来计算分类前后的信息增加量。可以说 Impurity Function 在决策树模型中是最重要的部分，它直接决定了决策树模型的效率和准确度。
决策树模型一般处理的是属性值分散的属性，对于连续数字属性，其可以使用二分或者区间的方法将其作为一种离散属性来操作。此外，由于决策树容易出现过拟合的情况，所以也提出了一系列处理问题的方法。
3 分类器的评价方法：在拥有一个分类器模型之后，我们需要对其进行评价，此处给出了一系列评价准则和评价参数。
评价方法给出了三种，分别是：Holdout Set，Multiple Random Sampling，Cross-Validation。第一种方法在可用数据比较多的情况下适用，而剩下的方法在处理小数据集的时候比较有效。
分类参数给出了 Precision, Recall, F-score and Breakeven Point 四种，precision 表达了在所有分类为正的数据中正确分类数目的比例, 而 recall 在实际正类数据中正确分类数目的比例，一般而言，p 和 r 之间有一种负相关关系。F-score and Breakeven Point 是对上面两种参数的延申。
第四章:
1. 无监督学习：第三章的监督学习，使用了已经被标注好的数据，这样模型可以构建属性和目标分类之间的关联关系，也可以理解为人为监督地进行学习。而在无监督学习中，提供给机器地是比较原始地数据，机器并不知道它们可以被分成什么类别，无监督学习的目标使根据数据之间的相似性，将其分为多个聚类，聚类内部的数据之间彼此相似，而不同聚类之间的数据有着比较明显的差别。
聚类有着比较广泛的用途，比如销售商品问题上，如果希望满足所有客户的需求，那么就给每一个人量身定做商品，但是这样根本无法盈利，而如果完全不考虑客户需求，那么只生产一种商品，则这个商品不一定被市场欢迎。一种理想的办法是，收集客户的数据，根据其相

似性分为几类客户，再对不同类别的客户销售符合其需求的商品，这样可以确保利润最大化。为了研究聚类的相似性，我们需要引入 distance function 来计算其在多维空间中的距离。一般可以使用向量距离。

2. K-means 算法：k-means 算法是应用最广泛的聚类算法之一，其效率很高且有着较好的准确性。根据其名字，其会根据聚类内部的均值产生总共 k 个聚类。

其整体思路很简单，首先随机抽取 k 个样本点作为原始中心，然后计算其他样本点到这 k 个点的距离，每个样本点和其距离最近的中心加入到一个聚类，形成新的聚类，在这个新的聚类中重新根据均值计算中心，重复上面的流程，直到停止标准达到，比如说中心收敛或者样本点不再重新分配以及 SSE 减少程度最小。

算法整体简单并且高效，但是也存在缺点。首先，它只能针对可以定义均值的数据点进行聚类，如果说出现一个属性，其值不是数字，而是对或错，那么就无法定义均值的概念，处理这个问题的方法是使用 k-modes 这种变体方法。

此外，该方法对于噪音点以及初始分配的原始中心十分敏感，其会对结果造成很大影响，一般处理噪声可以采用采样小样本的方法进行处理，原始中心问题可以多次尝试或者手动选择的方法。另外，其对于不规则形状数据无法进行准确地聚类。

3. 聚类的表示方法：比较常见的有三种方法，第一种就是使用聚类中心来表示聚类，它符合聚类的定义，并且很好分析。另外一种是将每一个聚类作为一个 class，而将分好的数据点作为数据集进行监督学习，这样的好处在于可以产生一个很好理解的分类模型。其问题在于有可能出现分类过多的问题，比如将一个聚类划分为了好几个不同的类。还有一种是使用聚类中最频繁值作为中心，其一般用在 k-modes 方法中。

而对于任意形状数据的表示来说，其在高维空间中是很难表示的。