2020.2.9 第二次读书报告

09018330 孙毅远

一、自己提出的问题的理解

1.为什么第一次遍历时，要产生F_1和L两个集合?

理解：

为了防止漏掉一些元素，按照正常求F_1的方法，可能出现一个$item$的$support$不能满足自己的MIS
但却可以满足别的$item$的MIS的情况，这样组合依然可以产生符合要求的F_2，但是这个$item$并不会出现在F_1中
所以才以满足条件的第一个元素的MIS为基准，生成了一个L去生成F_2
之后只需迭代通过交集产生下一代即可

二、别人提出的问题的理解

1. 为什么在 $MScandidate$ 函数中每次循环中使$|sup(i_{k-1}) - sup(i'_{k-1}| \leq \phi)$就能令一个 $itemset$ 中 $maxsup$ 与$minsup$ 的差值在 ϕ 的范畴内?

理解：

因为新产生的集合$\{i_1, i_2, \cdots, i_{k-1}, i'_{k-1}\}$中只有$i_{k-1}, i'_{k-1}$这一对新组合
其他组合都已经满足了$|sup(i_x) - sup(i_y| \leq \phi$的条件，所以只需判断这一组是否满足即可保证新的集合满足条件

2.为什么对于使用 MIS 的 $item$ 来说，向下封闭性质不再适用

理解：

因为可能因为某一元素的 MIS 较小，去掉这一元素的子集不一定满足剩余的 MIS

3. 在 $MS-Apriori$ 算法中，$level2-candidate-gen(L, \phi)$ 和 $MScandidate-gen(Fk-1, \phi)$ 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

理解：

和自己提的问题部分重合，原因是一样的，因为F_1中不包含那些特殊的非频繁项目
其$support$不满足自身的MIS值，但满足M中排在其之前的项目的MIS值，通过L的产生保证所有可能出现的都被包含

4.为什么只需记录 $c-c[1].count$ 就可以得到所有的 $f-\{a\}.count$

理解：

因为每次合并产生的新集合A中，它的子集中在之前没出现（未统计过）的只有$c-c[1]$的子集
其余的子集在上一轮的$c.count$中已经计算过了

三、读书计划

本周读书进度：2.3-2.6

下周计划：读至3.3

四、读书理解

2.3 Data Formats for Association Rule Mining

- 在应用中，文本文件可以看作 $transaction$ 数据，每个单词看作 $item$
- 离散化

2.4 Mining with Multiple Minimum Supports

- $rare\ item\ problem$：可能有一些 $item$ 本身的 $support$ 很低，但我们需要挖掘到包含它的规则，如果 $support$ 较高就找寻不到；但如果 $support$ 设置太低，就会产生大量冗余数据
- $minimum\ item\ support(MIS)$

  为了解决 $rare\ item\ problem$ 设立的新的阈值，用于寻找到同时包含出现频繁的 $item$ 和出现稀少的 $item$ 的规则

  让用户指定多个最小支持度阈值，每个元素集需要对每一个 $item$ 满足不同的 $minimum\ item\ support$
- 可以将其他项目的支持度阈值设置为大于1，这样可以很快找到只包含某些项目的项集
- 扩展模型

  一条规则的最小支持度采用在规则中出现的所有项目的 MIS 的最小值，因此每个项目都会有一个 MIS 值

  令 $MIS(i)$ 表示第 i 个项目的 MIS 值，一条规则 R 的最小支持度为 R 中所有项目中最低的 MIS 值。
- 需要注意的是向下封闭属性在扩展模型中不再满足，因为可能因为某一元素的 MIS 较小，去掉这一元素的子集不一定满足剩余的 MIS
- 实际上 MIS 值使得一些只包含频繁项目的项集不再是频繁项目集（它们的MIS可能很高），而为了不把同时含有频繁项目和稀有项目的项集作为频繁项目集生成，引入 $Support\ Difference\ Constraint$（支持度差别限制）

  设 $sup(i)$ 为项目 i 在数据集中的真实支持度，对每个项集 s，$Support\ Difference\ Constraint$ 定义为

  $$max_{i\in s}\{sup(i)\}-min_{i\in s}\{sup(i)\}\leqslant\varphi$$

  其中$0\leqslant\varphi\leqslant1$为用户指定的最大支持度差别（$Maximum\ Support\ Difference$）

  这个条件将项集 s 中最大项目支持度和最小项目支持度差限制为 φ，大大减少了最终生成的频繁项目集，且依然满足向下封闭属性
- $MS-Apriori$

  同 $Apriori$ 算法类似，逐级搜索，区别是对 I 中的项目基于 MIS 值进行升序排序

  令 F_k 表示 $k-$ 频繁项目集集合，每个频繁项目集表示为 $\{w[1],w[2],\cdots,w[k]\}$，其中 $MIS(w[1])\leq MIS(w[2])\leq\cdots\leq MIS(w[k])$

2.5 Mining Class Association Rules

- 分类关联规则（$Class\ Association\ Rules, CAR$）

  设 T 为一个含有n个事务的数据集，每个事务标有一个分类($Class$) y。 I 为 T 中所有项目的集合， Y 为所有分类表示($Class\ Labels$)，一个分类关联规则如下：

  $$X\rightarrow y,\text{其中 }X\subseteq I,\text{且 }y\in Y$$

  目标：分类关联规则挖掘指生成完整的满足用户指定的 $minsup$ 和 $minconf$ 的 CAR 集合

  理论上直接使用 $Apriori$ 算法可以做到，但在实际应用中会导致组合爆炸

- 规则项（ $Ruleitem$ ）

  形式为 $(condset, y)$，其中$condset \subseteq I$ 是条件集，$y \in Y$是一个类标。支持计数表示为 $condsupCount$ 是 T 中包含该条件集的事务数量。

  规则项的支持计数为 $rulesupCount$ ，是 T 中包含条件集 $condset$ 且类标为 y 的数量。支持度为 $\frac{rulesupCount}{n}$，置信度为 $\frac{rulesupCount}{condsupCount}$

  同理有频繁规则项，非频繁规则项

- 挖掘算法

2.6 Basic Concepts of Sequential Patterns