Web Data Mining 第二次读书报告

2020/2/2-2020/2/10

汇报人：71117415 喻泽弘

读书进度：chapter2.3-chapter2.5

问题与解答

- 个人提出的问题：
  1. 如何理解Fig.2.8的9-11行?

     实际上这三行代码进行的是剪枝的工作，根据数据性质，从candidate itemset中删去不满足条件的集合。主要根据一下两条数理性质：

     - 对于候选集c的任意子集s，如果$c[1] \in s$,即s与c具有相同的minMis，那么如果$s \notin F_{k-1}$,这说明s不是frequent itemset，根据向下闭包性质，那么c也不可能是frequent itemset，因此，可以把c从候选集中删除
     - 对于候选集c的任意子集s，如果$MIS(c[1]) = MIS(c[2])$,即c的最小两个MIS相同，这说明c的任一子集仍然同c一样具有相同的最小MIS，因此c具有向下闭包性质，此时，如果$s \notin F_{k-1}$,这说明s不是frequent itemset，那么c也不可能是frequent itemset，因此，可以把c从候选集中删除
- 别人提出的问题(仅记录了增强我对该章节理解的问题)：
  1. 为什么在MScandidate函数中，每次循环使用$|sup(i_{k-1}) - sup(i^{'}_{k-1})| \leq \phi$就能令一个itemset中的maxSup与minSup的差值在ϕ的范畴内？严谨的数学说明。

     对于该问题，我认为需要分别考虑合并的两个子集。

     对于$f_1 = \{i_1, \dots, i_{k-1}\}$,集合内的任意两元素组合都满足$|sup(a) - sup(b)| \leq \phi$,对于$f_2$同理，因此合并过后的集合$c = \{i_1, \dots, i_{k-1}, i^{'}_{k-1}\}$,此时，仅有$i_{k-1}, i^{'}_{k-1}$这一对组合尚未考虑他们的sup差值是否在$\phi$的范畴内，因此，考虑$|sup(i_{k-1}) - sup(i^{'}_{k-1})| \leq \phi$就能令一个itemset中的maxSup与minSup的差值在ϕ的范畴内。
  2. 算法中为什么记录了$f - \{a\}$就能确保所需的非frequent的condition的count被记录下来?

     这个实际上是一个head item problem，问题出现的原因便是itemset中拥有最小MIS值的元素出现在consequent中。因此$X \to Y$，X.count可能并没有记录，并且X可能不等于$f - \{a\}$，其中a是拥有最小MIS值的元素。那么，我们不妨假设$X = f - \{a, b\}$,对于$f - \{b\}$该集合，显然$f - \{b\}$同f一样具有相同的minMIS，因此根据向下闭包性质，$f - b$仍然是frequent itemset，因此此时记录了$f - \{a, b\}.count$，因此，对于$X = f - \{a, b\}$，这种情况，在之前的循环中便求出了X.count
  3. 在MS-Apriori算法中第二轮搜索与其他轮次搜索的不同的目的是什么?

     可以发现，level-candidate-gen function同MScandidate-gen function传入的第一个参数并不相同，level-candidate-gen传入的是L，而MScandidate-gen传入的是F_{k-1},对于L集合的任意元素a，满足a.count/n>minMIS,这说明，L中包含了的元素不能并不满足自身的MIS，但是它仍然大于minMIS，通过L可能正确的构建F_2,然而仅仅通过F_1是无法构建出F_2的，这便是level-candidate-gen function同MScandidate-gen function不同的目的
  4. MS-Apriori是否需要采用字典序列?

     我认为MS-Apriori并没有采用字典序列，MS-Apriori本身并对item按照MIS的大小顺序进行了排列，因此并不需要再次使用字典序列。同时，我认为Apriori算法的核心在于item具有先后顺序，而不是必须使用字典序列

读书计划

下周打算尽量完成web data mining这本书第三章的阅读，同时，如果学有余力，适当的编写实现代码

读书收获

- 在实际应用中，比如自然语言处理，我们可以把每段文本看出一个transaction，每一个单词视为一个item
- 对于关系表使用Apriori算法中，可能会出现同一attribute具有不同值的问题，以及存在attribute对应的值域过大这一问题。对于值域这个问题，可以通过对数据分段解决，比如身高这一属性，具有多个不同的值，单单看每个不同的值，并没有意义，但是如果把身高数据分为高、中、矮三段，此时数据便有意义了许多。对于同一attribute具有不同值这一问题，我认为，在设计算法的时候，就应该尽量侧重于不同属性之间的关系，而非同属性不同值之间的关系
- 对于实际问题中，为了发掘item之间的关系，有时使用一个support值并不有效，比如transaction中，可能存在support值很高以及support值很低的item，此时，如果将support值设置很高，那么这些低support值的元素将没有意思，如果将support值设置的很低，那么也将发现许多不存在实际意义的关系。因此，此时，我们需要引入多support值来处理这一问题，此时Apriori算法的向下封闭属性将存在问题，我们需要对Apriori算法进行修改是的它适用于Multiple support问题，对于MS-Apriori算法的其他细节，这些细节都是问题列表中的一些问题，因此这里便不再赘述
- 对于MS-Apriori算法中关联规则的产生，某个f的子集可能并不是frequent的，这个子集便是例外集合。为了解决该问题，可以看到之前的算法每次在判别时，都是同时记录去掉第一个元素之后的集合的count值，就是为了使产生关联规则时需要的所有的count值都被记录下来，从而不需要重复扫描数据