1. （必填）自己提出的问题的理解（罗列全部）：
2. **提出的问题1：**

P22第二段，提到为了解决稀有项目问题，一个常见的解决办法是将数据分为几个小块，每个子集只包含那些出现频率比较相近的项目。关于这个分块依据（方法）不是很理解？

讨论后的理解：

书中提到的数据集并不是所有事务的集合，而是说可以将一个事物中的项目按照频率分成几个块，然后针对不同频率的子集给定不同的MIS。这样可以解决有效解决稀有项目问题，在此基础上，还有一种算法是给定每一个项目一个单独的MIS，但在实际操作中，此参数的设置一般由用户设定（需要结合实际运用）。

1. **提出的问题2：**

假定各个项目的MIS都由用户指定，那么参数φ应该怎么考虑呢？为什么要设置该参数呢（即，在满足各自MIS的情况下，支持度相差很大的事物为什么要过滤掉）？P25？

讨论后的理解：

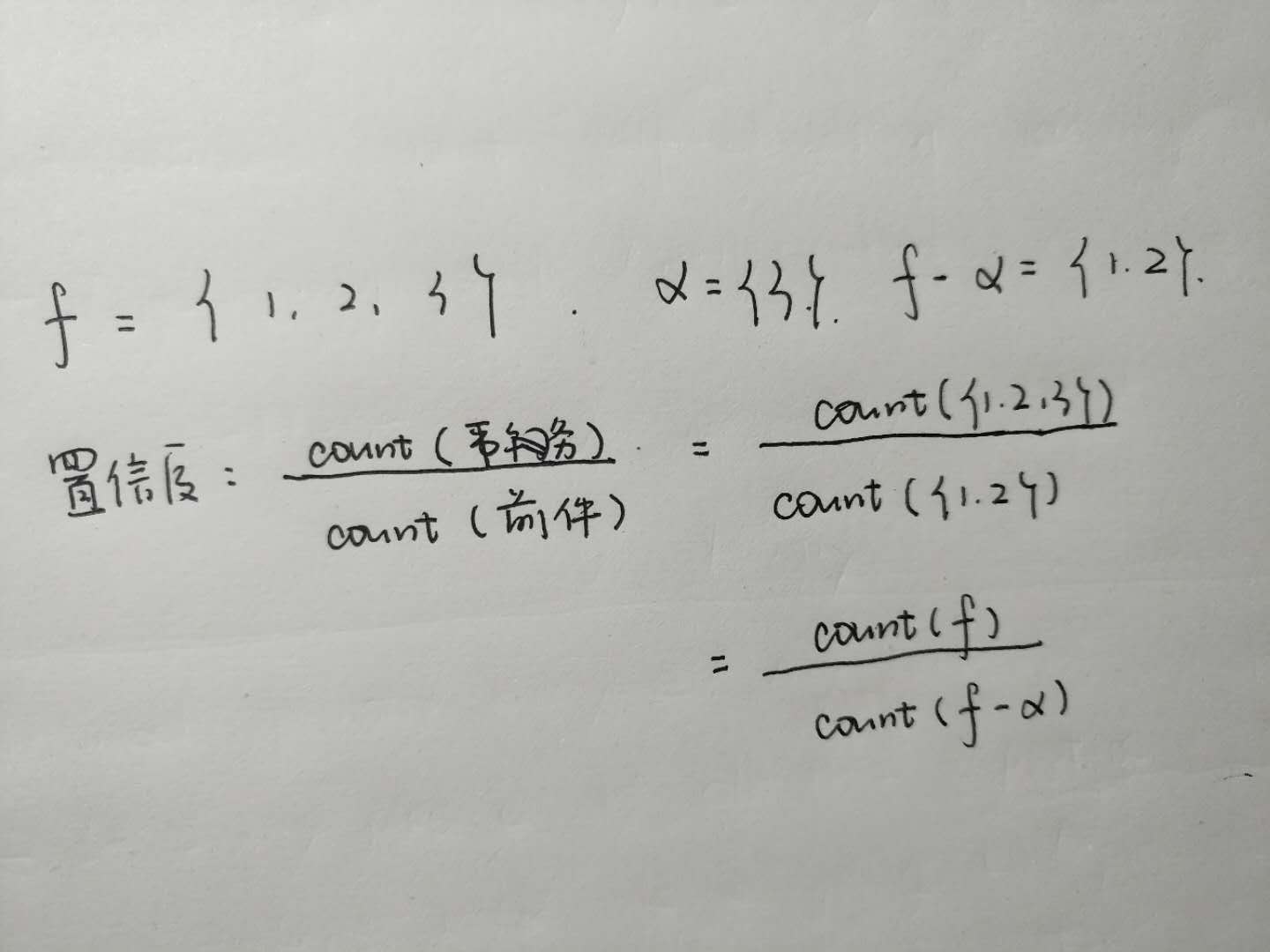
参数φ需要使得收益期望最大化，所以需要对收益函数进行关于φ的建模，然后通过求导取极值的方法找到解。事务中包含支持度相差很大的项目，实际上就是既包含频繁项目又包含稀有项目的事务。这样的事务推出的关系可能没有多大意义，所以可以将其过滤。

1. （必填）别人提出的问题的理解（选择几个问题罗列，并给出理解）：
2. **问题3**：

P19最上方 置信度为什么采用f.count/(f-a).count的形式？

自己的理解：

这是原来的置信度公式的一个变体，推导过程如下：



1. **问题4：**

P16说不需要将整个数据集加载到内存，这是Apriori的特点，但是既然要统计一个事务的置信度，应该没有必要必须把全部的数据集加载进去，可以一条一条读，那么有什么算法是必须要把整个数据集都放进去的嘛，为什么要这么做？

自己的理解：

线性的遍历过程确实不需要将整个数据集全部加载到内存，但是如果需要对原有数据集进行比如结构的转换，比如需要将文件存为树结构、图结构、字典结构，则需要将原始的数据集全部加载到内存，再进行后续的操作。

1. **问题5：**

有没有一些其他的优于Apriori的算法？？

自己的理解：

书上提到FP-growth算法，Apriori算法本质是一个排列组合问题，当数据线性排列时，其算法的时间复杂度是指数级。而FP-growth算法则是将数据集存为在FP树中。构建 FP 树时，首先统计数据集中各个元素出现的频数，将频数小于最小支持度的元素删除，然后将数据集中的各条记录按出现频数排序，剩下的这些元素称为频繁项；接着，用更新后的数据集中的每条记录构建 FP树，同时更新头指针表。头指针表包含所有频繁项及它们的频数，还有每个频繁项指向下一个相同元素的指针。

1. （必填）读书计划

1、本周完成的内容章节：第三章（4.1-4.4）

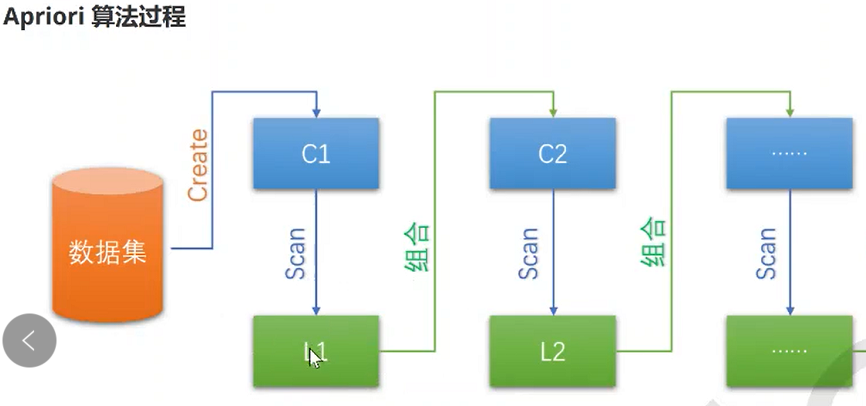
2、下周计划：第四章（4.5-4.9）

四、（选做）读书摘要及理解或伪代码的具体实现（读书摘要、伪代码的具体实现代码等可以写到这个部分）

1、读书摘要及理解（选做）

摘要：本周完成了关联规则相关内容，包括频繁项目集生成的Apriori算法和关联规则生成算法，以及多最小支持度的关联规则挖掘。

**Apriori算法**是用于生成频繁项目集的算法，其原理为向下封闭属性（如果一个项集满足某个最小支持度要求，那么这个项集的任何非空子集都必须满足最小支持度）。算法过程如下：



Create过程为创建项集操作，初始将每一个项目作为一个项集；

Ck表示有k个项目的项集，Lk表示有k个项目的频繁项集；

Scan表示数据集扫描函数，主要是过滤支持度小于最小支持度的项集；

最终可以得到有K个项目的频繁项目集。

**关联规则生成算法**是基于频繁项目集，对于每一个项集，排列组合的方式确定前件和后件作为一条规则，再将不满足最小置信度的规则删去。遍历整个频繁项目集，执行上述操作，最终可以生成关联规则。

**多最小支持度的关联规则挖掘算法**是为了解决“稀有项目问题”而提出的一种算法，与简单的Apriori算法不同的是，针对每一个项都增加了一个MIS(最小支持度)，并且在过滤不满足频繁项集的时候，增加了一个项目支持度差最大值界限参数φ（最大支持度差别），用以过滤掉同时含有频繁项目集和稀有项目集的项目集。

2、伪代码的具体实现(选做)

实现了Apriori算法：

算法准备：



算法实现：

