**2017年4月**

**第十届“认证杯”数学中国**

**数学建模网络挑战赛**

**承 诺 书**

我们仔细阅读了第十届“认证杯”数学中国数学建模网络挑战赛的竞赛规则。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的, 如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们接受相应处理结果。

我们允许数学中国网站([www.madio.net](http://www.madio.net))公布论文，以供网友之间学习交流，数学中国网站以非商业目的的论文交流不需要提前取得我们的同意。

**我们的参赛队号为：1510**

**参赛队员 (签名) ：**

队员1：

队员2：

队员3：

**参赛队教练员 (签名)：**

**参赛队伍组别：本科组**

**第十届“认证杯”数学中国**

**数学建模网络挑战赛**

**编 号 专 用 页**

参赛队伍的参赛队号：1510

竞赛统一编号（由竞赛组委会送至评委团前编号）：

竞赛评阅编号（由竞赛评委团评阅前进行编号）：

参赛密码 

**（由组委会填写）**

**2017年第十届“认证杯”数学中国**

**数学建模网络挑战赛第一阶段论文**

题 目 （**C题**）移动端考研产品的预测与评估模型

关 键 词 FP-growth算法 关联规则学习 定价模型 马尔可夫模型

摘要

**针对问题一**：

根据题目要求，本文通过阅读问卷之后发现：第16问（您倾向于通过哪种渠道进行考研学习？）可以将参与问卷调查者分为倾向线下渠道（含辅导班、讲座、自学）、倾向线上PC端、倾向线上移动端三种。因此，根据题目要求本文分析倾向线上移动端的人所选择的其他问题的选项的关联性，即可分析出主要因素。

对于各个问题间的关联信息的分析，适合使用数据挖掘的关联规则学习，所以需要对数据进行二元化处理。因为问卷数量较大，使用FP-growth算法处理数据从而得到结论，影响移动端考研产品发展的主要因素有：1、对“伴随着移动互联网时代的来临，使用移动端进行考研学习的人数将呈现大幅度增长趋势”的观点的认同。2、倾向于选择考研课程辅导。3、时间精力有限，能安排在考研学习的时间较少，有随时随地使用考研产品的需求和期望。4、期望考研产品能够满足灵活安排时间的需求。5、刚进入大学的新生，对未来考研有一定的规划。

**针对问题二**: 本文将问题二分成两个问进行求解：

（1）根据题目要求，本文从产品的价格与销售量的关系出发，结合微观经济学中的产品定价法，建立移动端同类考研产品的定价模型对产品做出合理的区间价格估计。

首先本文假设在完全市场经济的条件下，同时考虑产品收益、价格与销售量的关系产品数量限制等多种因素构建出多产品定价决策模型。

然后本文通过查找数据，根据市场热度选取了三个移动端考研APP：考研帮、研途宝、口袋题库。结合这三个移动考研产品的实际市场价格（年会员价格）、销售量（购买会员数量）关系代入所建立的定价模型，根据所求得同类产品的最优定价估计出该类产品的合理区间。本文通过建模求解可得，三个考研APP（考研帮、口袋题库、研途宝）年会员的最优定价，最终得到区间价格：[22.1021，92.1492] (单位/元)。

（2）题目要求，本文利用MATLAB对附件2中的数据进行数据挖掘和分析可得到可供用户选择的三种考研学习渠道:线上PC学习、线上移动端学习、线下学习。根据所挖掘分析得到的数据：首先可得到三种学习渠道的初始分布比率为：30%、 17%、53%。再利用马尔可夫预测得到移动端考研产品在某一时期、某一地区在整个考研教育产品市场当中的潜在市场占有率预期分布。最后得到结果：线上移动端的潜在市场占有率为23%。

**针对问题三**：

根据题目要求，本文利用问题二中使用的马尔可夫预测模型在具体市场环境下的应用，通过使用问题一的结论，依据所选定的城市发展水平和用户人群分析来预测考研移动端市场的发展趋势，依据具体的数据预测其未来几年的市场占有率，上海市经济发达、移动互联网普及率高且用户有更多碎片化学习的需要，潜在市场占有率在三年后能达到50%以上，得到结论：移动端产品的投放能获得良好收益。

**Summary**

**For question one**:

In this paper, after reading the questionnaire found: Question 16 (you tend to pass through which channels for graduate study?) Can participate in the survey respondents are divided into the trend of the channel (including remedial classes, lectures, self), tend to online PC , Tend to move on the line three. Therefore, the main factors can be analyzed by analyzing the relevance of the options of other issues chosen by the person who is inclined to the mobile on the line according to the subject requirements.

For the analysis of the relevant information between the various issues, suitable for the use of data mining association rules to learn, so the need for data processing. Because the number of questionnaires is large, the FP-growth algorithm is used to process the data to get the following conclusion.

**For question two**:In this paper, the problem is divided into two questions to solve:

(1) Based on the relationship between the price and the sales volume of the product, this paper puts forward a reasonable interval price estimation of the product based on the product pricing method in the microeconomics and the pricing model of the similar research products.First of all, this paper assumes that under the condition of complete market economy, the product pricing decision model is constructed by considering the factors such as product yield, price and sales quantity and so on.Then this article through the search data, according to the market heat selected three mobile end PubMed APP: Kaoyan Bang, research treasure, pocket question bank. (The annual member price) and the sales volume (the number of purchased members) are substituted into the established pricing model, and the reasonable range of the products is estimated according to the optimal pricing of the similar products. The In this paper, through the modeling solution available, the three examinations APP (Kaoyan Bang, pocket exam, research treasure) annual members of the optimal pricing are: 0, so it can be estimated:The price of the annual membership of APP products is: [22.1021,92.1492] (unit / yuan).

(2) In this paper, we use MATLAB to data and analyze data in Annex 2 to get three kinds of entrance examination channels available for users: online PC learning, online mobile learning, offline learning. According to the data obtained by digging analysis:First, the initial distribution ratio of the three learning channels: online PC-side learning accounted for 30%, online mobile side accounted for 17%, offline learning accounted for 53%.In this paper, we use the Markov forecast to obtain the expected distribution of the potential market share in the whole market of Kaoyan education products in a certain period of time. As shown in the table,Finally, the potential market share of online mobile is 23%.

**In view of the problem three**: In this paper, we use the Markov forecasting model used in Problem 2 to apply it in specific market environment. By using the conclusion of problem 1, we can forecast the development trend of Kaoyan mobile market according to the selected urban development level and user population analysis. Specific data to predict its market share in the next few years, Shanghai's economy developed, mobile Internet penetration and users have more fragmented learning needs, the potential market share in three years to reach more than 50%, get conclusions : The delivery of mobile products can be a good income.

# 一、问题重述

1.1 问题背景

2017年的全国硕士研究生招生考试共有201万人报名参加，比去年增加了24万名考生，增加了13.56%。移动互联网时代的到来，使得许多考研教学活动转移到了手机等移动端，同时出现了一系列依托于移动互联网的产品，如教学app，手机题库，单词本，错题本或依托于现有移动端视频平台的直播课程等。

1.2 问题提出

国内某知名考研网站为了深入了解移动端考研产品的市场占有率和发展趋势，开展了网上问卷调查，问卷格式如附件1所示。共收集有效问卷38182份，从中随机抽取出10000份样本形成了附件2。请你建立合理的数学模型解决如下问题。

1. 请依据附件2中的数据进行数据挖掘，找出影响移动端考研产品发展的主要因素。

2. 请估计移动端考研产品的合理价格区间，预测移动端考研产品的潜在市场占有率。

3. 请选择一个高校相对较多的城市作为研究对象，充分考虑经济、社会情况和考研教育的特点，评价移动端产品投放的可行性。

# 

# 二、问题分析

2.1 问题一 ：

问题一要求根据附件2中的数据进行数据挖掘，找出影响移动端考研产品发展的主要因素。根据题意，本文通过附件2中各个问题进行关联信息的分析，使用数据挖掘的关联规则学习，所以需要对数据进行二元化处理。因为问卷数量较大，使用FP-growth算法处理数据从而得到结论。

2.2 问题二：

本文针对问题二中的两个问分别进行问题分析。

根据题意，移动端考研产品包括了：教学APP、手机题库、单词本、错题本或依托于现有移动端视频平台的直播课程等。本文从产品的价格与销售量的关系出发，结合微观经济学中的产品定价法，建立移动端同类考研产品的定价模型对产品做出合理的区间价格估计。

已知市场占有率是指产品在某时期的实际销售占整个行业的实际销售百分比，即市场占有率=（该品牌实际销售数量/行业实际销售数量）\*100%，因此通过查找网络数据，本文利用马尔可夫预测得到移动端考研产品在某一时期、某一地区在整个考研教育产品市场当中的潜在市场占有率。

2.3 问题三：

根据题意，问题三是从问题一和问题二的结论出发，通过选择教育相对发达、高校众多的地区，在已有数据的基础上，在对该地考研教育业的市场分析的基础上，使用马尔科夫模型，对未来几年移动考研app的市场占有率进行预测，并据此评价移动端产品投放的可行性。

# 三、模型假设

3.1问题一的假设

假设参与问卷调查者填写的问卷均符合自身实际情况

3.2问题二的假设

假设下载移动端考研APP的用户都购买了会员。

假设所建立的定价模型在完全市场经济条件下运行。

# 四、符号说明

4.1问题一符号说明：

S：支持度 c:置信度

:支持度计数 

4.2问题二符号说明：

、: 产品的价格水平 、：产品的销售量

、： 预期销售收入

：交叉弹性 ：需求价格弹性

：价格变动系数 ：产品的商品数量上限

、：销售量与价格水平的函数关系

：产品的销售量对该产品价格的灵敏系数

：状态到状态的转移次数，

：转移状态矩阵 系统从状态状态到其他状态的次数

# 五、模型的建立与求解

5.1问题一模型的建立与求解

5.1.1问题分析与模型准备

本文使用关联规则挖掘算法，对附件2的数据进行挖掘分析，找出各个因素之间的关联程度，再进行评估优化，找出影响移动端考研产品发展的主要因素。

附件2的大部分数据项（问卷问题）属于多选项问题，而且有多选问题，并且含有可自行填写的其他选项，在使用关联规则挖掘算法之前，需要对附件2的数据进行预处理，将分类属性和对称二元属性转换成“项”。通过为每个不同的属性-值对创建一个新的项即可实现这种转换。例如，“您所在的城市”是一个分类属性，可以用三个二元项取代：{您所在的城市 = 北上广深等一线城市}，{您所在的城市 = 二线城市}，{您所在的城市 = 三线城市}。类似的，“性别”是一个对称二元属性，可以转换成两个二元项：{性别 = 男}，{性别 = 女}。对于自行填写的其他选项，本文统一处理为“其他”。

为了方便算法对数据的处理，再进一步进行编码，将数据简化表示为 p + 数据所属的问卷题号 + 数据所属问题的第几个选项，首字母p代表Problem：{您所在的城市 = p3\_1}，{您所在的城市 = p3\_2}，{您所在的城市 = p3\_3}。简化后的数据具有较强的可读性，并方便算法进行处理。

如图1显示最后将清洗后的数据导出的结果(部分)。

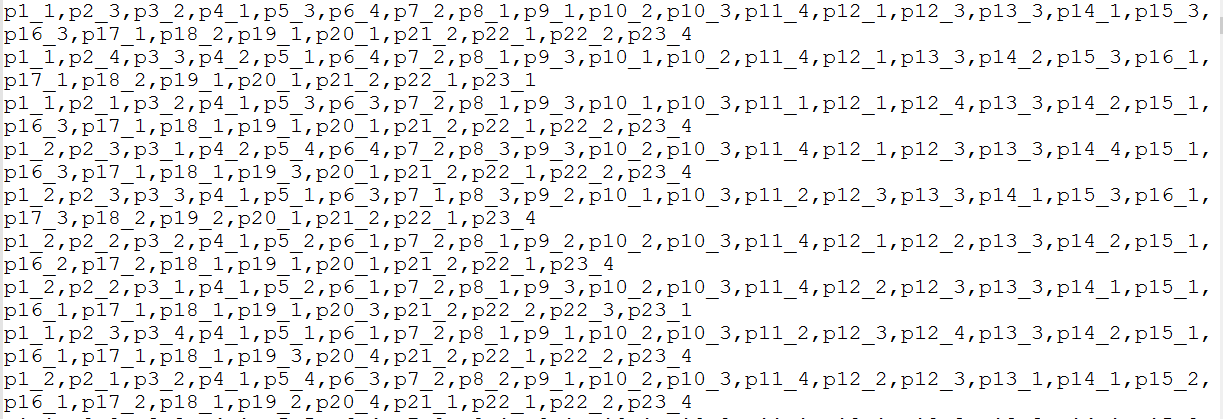


图1：数据预处理后的格式

5.1.2 算法的概念说明与模型的建立

5.1.2.1关联分析

关联分析是一种用于发现隐藏在大型数据集中的有意义的联系，所发现的联系可以用关联规则或频繁项集的形式来表示。例如，可以提取出这样的规则。

由于问卷的问题16是“您倾向于通过哪种渠道进行考研学习？”，只需要找出所有和“您倾向于通过哪种渠道进行考研学习？= 线上移动端”这样的二元项在同一个频繁项集的所有关联规则即可求解问题一。

5.1.2.2项集和支持度计数

设是问卷数据中所有项的集合，而是所有事务的集合。每个事务包含的项集都是的子集。在关联分析中，包含0个或多个项的集合被称为项集。如果一个项集包含k个项，则称它为k-项集。

附录2中的每一个用户填写的问卷视为一个事务。

事务的宽度定义为事务中出现项的个数。如果项集是事务的子集，则称包含项集。

项集的一个重要性质是它的支持度计数，即包含特定项集的事务个数。数学上，项集的支持度计数可以表示为：



其中，符号表示集合中元素的个数。

例如，在预处理后的数据集中，项集{p1\_2,p2\_3,p3\_1,p4\_2,p5\_4}的支持度计数为6，因为只有6个事务同时包含这三个项。

5.1.2.3关联规则

关联规则是形如的表达式，其中和是不相交的项集，即。关联规则的强度可以用它的支持度（support）和置信度(confidence)度量。支持度确定规则可以用于给定数据集的频繁程度，而置信度确定在包含的事务中出现的频繁程度。支持度（s）和置信度（c）这两种度量的形式定义如下：





5.1.3模型的求解

5.1.3.1步骤一：去除数据的交叉支持模式，减少倾斜支持度分布造成的影响。

问卷中大部分问题的非二元选项在二元化后并非都能等价影响算法分析过程，例如在这10000份问卷中统计第5问的选项占比饼图，如图2所示：

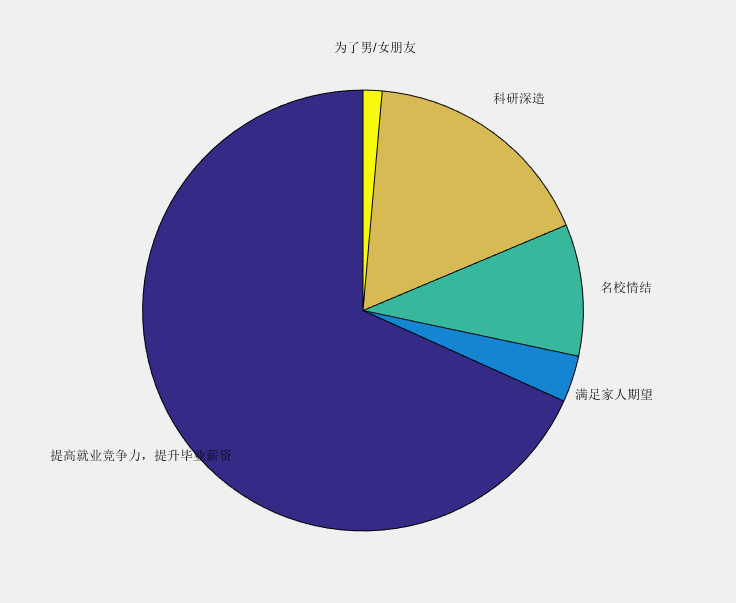


图2：问卷第5问各选项占比饼图

可看出除了第一项外，选择其它选项的人数远小于选择第一项的人数。对这种数据，难以选择合适的支持度阈值去挖掘它，如果阈值太高，可能会遗漏较低支持度项的关联规则；如果支持度阈值太低，将会导致提取出的关联规则的数量大幅度增加，这也导致算法运算过程中所需的运算与存储资源大幅度增大，除此之外，可能会导致提取出大量的高频率项与低频率项相关联的虚假关联规则，这样的模式就是交叉支持模式，这些模式的相关性太弱，虚假的可能性较大。

所谓交叉支持模式是指：交叉支持模式是一个项集，它的支持度比率



小于用户指定的阈值。

支持度不足以消除交叉支持模式，虽然可以采用较高的支持的阈值来消除交叉支持模式，但是这样会损失一些强关联但支持度未达到阈值的关联规则。置信度剪裁也不足以消除交叉支持模式，因为交叉支持模式提取的规则的置信度可能很高。因此通过设置h置信度来确保项集中的项之间是强关联的。

本文采取适当的h置信度阈值来消除交叉支持模式，避免倾斜支持度分布对关联规则数据挖掘算法带来的影响。

关联规则挖掘算法使用FP Growth算法，此算法使用FP树组织数据，从中提取频繁项集。

5.1.3.2预估最小值支持度计数

首先需要确定需要提取的项集需要满足的最小支持度计数阈值。为了找到合理的大致的最小支持度计数，本文处理了最小支持度计数参数在600至9000区间内的情况，得到了各参数下输出项条数条数和含有p16\_3的项的条数，见表1：

表1：不同最小支持度计数产生不同的输出项条数

|  |  |  |
| --- | --- | --- |
| 最小支持度计数（输入参数） | 输出项条数 | 含有p16\_3的项的条数 |
| 600 | 536731 | 7273 |
| 700 | 342229 | 3741 |
| 800 | 228913 | 2071 |
| 900 | 158856 | 1142 |
| 1000 | 113670 | 684 |
| 2000 | 9784 | 4 |
| 3000 | 1802 | 0 |
| 4000 | 459 | 0 |
| 5000 | 117 | 0 |
| 6000 | 43 | 0 |
| 7000 | 14 | 0 |
| 8000 | 3 | 0 |
| 9000 | 0 | 0 |

通过MATLAB绘图可直观感受到输出条数随着最小支持度计数改变的变化，呈指数变化，且输出项条数远远大于含有p16\_3的项的条数，如图3所示：

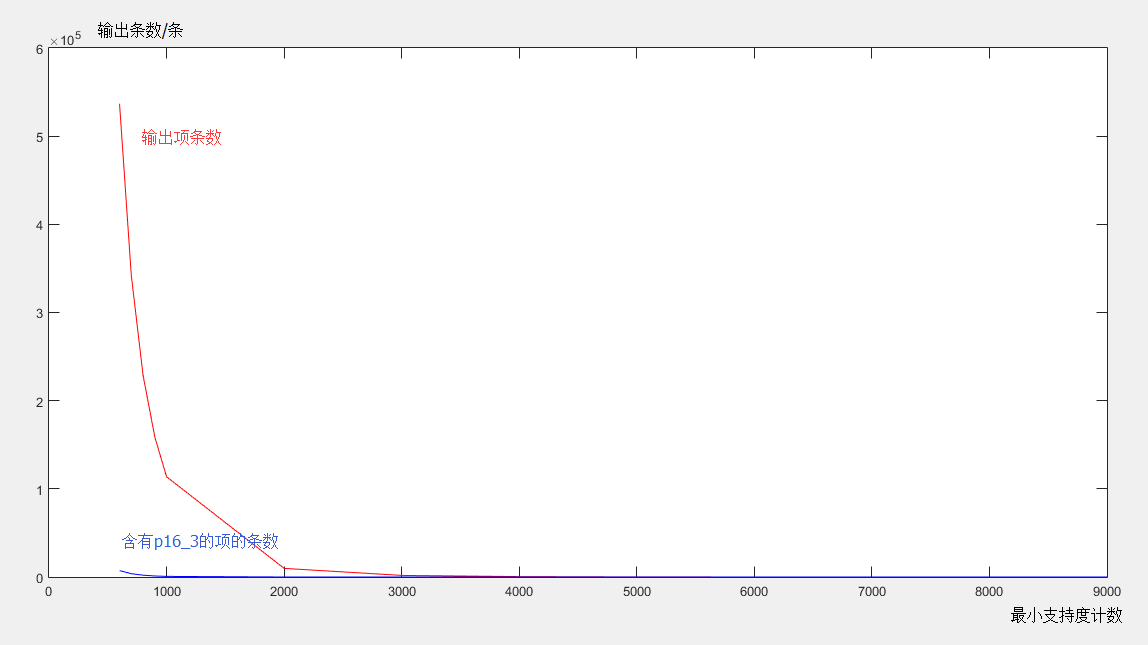


图3：输出条数随着最小支持度技术的改变的变化

为了能更直观细致地感受到不同最小支持度计数的变化对输出条数的较小变化，本文去除最小支持度计数为600时的数据，因为目的是选择合适的最小支持度计数参数，能够输出理想规模的含有p16\_3的项，便于分析处理，所以去除含有p16\_3的项的条数为0时的数据，并对修改后的数据对输出项条数绘图,如图4：

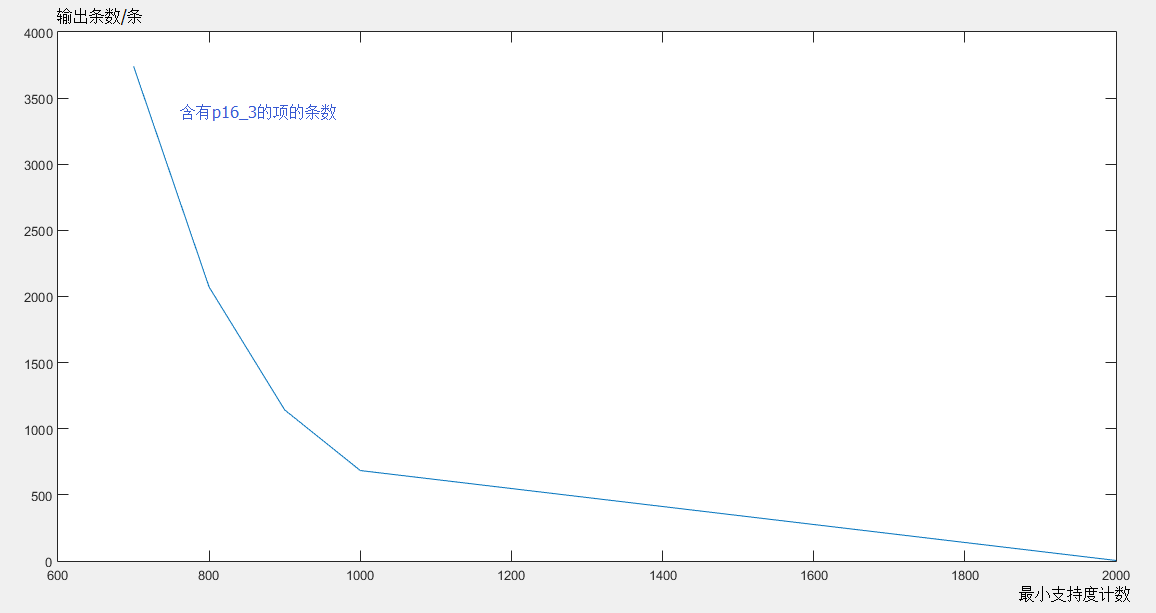


图4：输出条数随着最小支持度技术的改变的变化

当最小支持度计数小于800时，含有p16\_3的项的条数开始指数式递增，而当最小支持度计数大于1200时，含有p16\_3的项的条数变化较小。为了方便数据分析处理，以及提取出足够的有价值的数据，本文期望得到含有p16\_3的项的条数为1000条，所以选取最小支持度计数参数为920。

根据算法输出的结果，当最小支持度计数参数为920时，含有p16\_3的项的条数为1032条，符合本文的期望。输出的数据如下图5所示（详细代码参见附录）

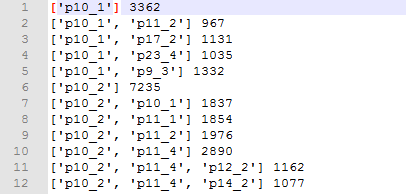


图5：最小支持度计数参数为920时输出的结果（局部）

每行方括号内的元素组成一个关联规则，行末的数字代表这一项的支持度计数。

5.1.3.3 FP树及其构造方法

5.1.3.3.1 FP树的生成

FP树是一种输入数据的压缩表示，它通过逐个读入事务，并把每个事务映射到FP树中的一条路径来构造。由于不用的事务可能会有若干个相同的项，因此它们的路径可能部分重叠。路径重叠的越多，使用FP树结构获得的压缩效果越好。

初始FP树仅包含一个根节点，用符号null标记。随后，用如下方法扩充FP树：

（1）扫描一次数据集，确定每个项的支持度计数。丢弃非频繁项，而将频繁项按照支持度的递减排序。

（2）算法继续上述过程，直到每个事务都映射到FP树的一条路径。

5.1.3.3.2 FP增长算法的频繁项集产生

FP增长（FP-growth）是一种以自底向上上式探索树，由FP树产生频繁项集的算法。给定树，算法首先查找以最后一个节点结尾的频繁项集，接下来依次向上发现。

由于每一个事物都映射到FP树中的一条路径，因而通过仅考察包含特定结点的路径，就可以发现以该结点结尾的频繁项集。使用与特定结点相关联的指针，可以快速访问这些路径。

下面考虑如何发现所有以特定节点e结尾的频繁项集的任务。

（1）第一步收集包含e结点的所有路径。这些初始的路径称为前缀路径。

（2）通过把与结点e相关联的支持度计数相加得到e的支持度计数，将其与最小支持度比较，从而得知它是否是频繁项集。

（3）如果它是频繁项集，继续向上递归，通过更新前缀路径中的支持度计数和删除非频繁的项来构建条件FP树。由于子问题是不相交的，因此FP增长不会产生任何重复的项集。

此外，与结点相关联的支持度计数允许算法在产生相同的后缀项时进行支持度计数。

伪代码如下(详细代码参见附录)：

**procedure** **FP\_growth**(Tree, a)

**if** Tree 含单个路径P **then{**

**for**路径P中结点的每个组合（记作b）

产生模式b U a，其支持度support = b 中结点的最小支持度；

**} else {**

**for each**a i 在Tree的头部(按照支持度由低到高顺序进行扫描){

产生一个模式b = a i U a，其支持度support = a i.support；

构造b的条件模式基，然后构造b的条件FP-树Treeb；

**if**Treeb 不为空 then

调用 FP\_growth (Treeb, b)；

**}**

**}**

5.1.3.4.用MATLAB对频繁项集根据h置信度消除交叉支持模式，如图6所示：

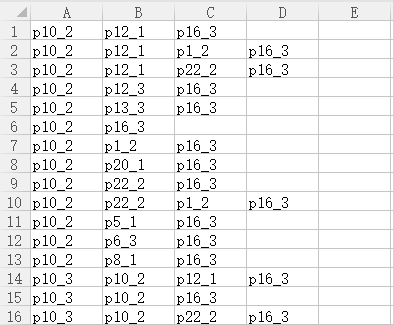


图6：对频繁项集根据h置信度消除交叉支持模式后的数据

本文设（h置信度）为0.3，可得到222条数据。

5.1.4求解结果及分析

对得到的关联规则统计支持度计数，并排序，选取支持度计数较大的几个强关联规则。这一关联规则的支持度计数越大，其支持度和置信度也越高，关联规则的关联性越强，也越可靠，如下图7所示：

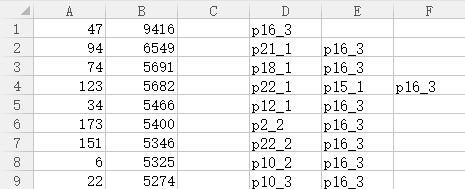


图7：统计得到的关联规则的支持度计数并排序

第一项规则因为只有一个属性（p16\_3自身），所以排除。由输出并排序后的关联性强的几项结果结合问卷以及现实经验可以分析出，影响移动端考研产品发展的主要因素有：

1. 对“伴随着移动互联网时代的来临，使用移动端进行考研学习的人数将呈现大幅度增长趋势”的观点的认同
2. 倾向于选择考研课程辅导
3. 时间精力有限，能安排在考研学习的时间较少，有随时随地使用考研产品的需求和期望
4. 期望考研产品能够满足灵活安排时间的需求
5. 刚进入大学的新生，对未来考研有一定的规划

5.2问题二模型的建立与求解

5.2.1 建模准备

本文从产品的价格与销售量的关系出发，结合微观经济学中的产品定价法，建立移动端同类考研产品的定价模型对产品做出合理的区间价格估计，同时利用马尔可夫预测得到移动端考研产品在某一时期、某一地区在整个考研教育产品市场当中的潜在市场占有率。

为了对移动端考研类产品做出合理的区间价格估计，通过查找相关产品的体验报告，本文例举出了三种用户量较大的移动端考研APP产品（考研帮APP、口袋题库APP、研途宝APP）的年会员购买价格以及在安卓应用市场的的下载量数据，如下表2所示：

表2：三款移动端考研APP产品的年会员购买价格以及下载量

|  |  |  |  |
| --- | --- | --- | --- |
|  | 考研帮 | 口袋题库 | 研途宝 |
| 会员价格（元/年） | 29.9 | 66 | 99 |
| 应用宝下载量/次 | 400000 | 70000 | 26000 |
| 豌豆荚下载量/次 | 320000 | 78000 | 80000 |
| 总下载量/次 | 720000 | 148000 | 106000 |

利用所建立的需求相互联系的产品数量——销售系统组合定价决策模型结合上表中的数据对产品的价格区间进行估计。

5.2.2定价模型的建立与求解

5.2.2.1定价模型的建立

本文首先假设在完全市场经济条件下运行，同时对产品数量——销售系统中同类产品进行定价。它以产品在一段时间内取得的投资收益最大为目标，在制定某产品价格时，充分考虑到该价格与该产品的销售量，其他群同类产品价格对该产品销售量的影响的互相联系，同时又考虑到产品数量限制的多产品定价决策模型，为企业科学定价具有重要的指导意义。

首先假设n种同类的移动端考研产品在价格水平时的销售量分别是，相应的预期销售收入分别为，其中，i=1,2,…,n。而在价格水平分别为时的销售量和预期销售收入分别为和，其中，。另外由于考虑到多产品的需求价格弹性和交叉弹性的影响，它们分别用和来表示，

 （1）

 （2）

由上式（1），（2）可以推得

= （3）

 （4）

另外假设销售量与价格之间有函数关系，而当商品由价格调整为时有，于是有

=

结合（3）和（4）式，，其中为价格变动系数，从而有（5）

因而，

（6）

故调价后的总预计销售收入为

 (7)

另一方面，本文再次考虑一下该产品价格与该产品销售量之间的关系和n中同类产品它们的价格对彼此销售量的交叉影响。某种产品价格与该产品销售量之间按的关系可以用该产品的需求曲线来刻画，在一些情况下，需求曲线可以近似看成一条直线，此时，需求曲线的函数式可以写成，其中，参数是该产品的销售量对该产品价格的灵敏系数，表示该品牌价格变动的一个单位引起该品牌销售量的变化量。在产品进入成熟期后，同类产品相互竞争，销售量对价格的灵敏系数变大，本文研究适度竞争情况下多产品的最优定价，假定销售量多价格的灵敏度系数参数为。

由于n种同类产品它们的价格对彼此销售量交叉影响，一般而言，在某一阶段，某一产品的降价（提价）会增加（减少）该产品这一阶段的销量，但会减少（增加）其他同类产品这一阶段的销量。若将所有其他产品的价格对该产品销量的交叉影响予以考虑，并假定这种交叉影响是线性的。则根据文献可假定产品的销售量与该产品的价格和其他产品价格之间如下关系

（8）

其中，是产品的销量对产品的价格变动的灵敏系数，表示竞争产品的价格变动一个单位引起产品的销量的变化量。

又由于考虑商品数量因素，每种产品调价后的销量应该小于该种产品商品数量的限制条件

即为：

 (9)

综合（6）、（7）、（8）、（9）式，得到需求相互；联系的商品数量——销售系统组合定价决策模型



 （10）

5.2.2.2定价模型的求解

由上述可知，本文所研究的定价问题是一个目标函数为二次函数而约束条件为线性函数的二次规划问题，根据二次规划问题最优解的存在条件：由于可行解集是空间的有界闭集，而二次函数是这个有界闭集上的连续函数，从而这个二次规划问题存在最优解。将上述二次规划化为标准形式：



 (11)

对于该二次规划模型，求解算法本文采用二次规划问题的有效集法来进行，通过求解有限个登时2约束二次规划问题来得到一般约束二次规划问题的最优解。

根据5.3.2.1表2中的数据画出考研帮、口袋题库、研途宝三个移动端考研APP的会员购买价格与下载量之间的关系，如图8所示：

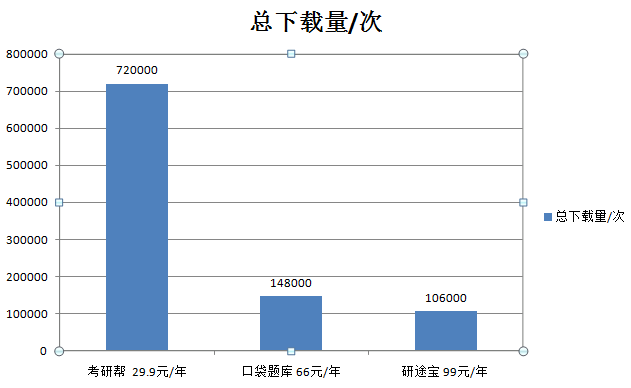


图8：三款移动端考研APP的会员购买价格与总下载量关系

根据图8分析得到：假设在不考虑其他因素影响价格的条件下，会员购买价格越高的APP其下载量越低，因此本文可以假设下载APP的用户量都会并且已经购买了APP会员，即APP的下载量等于APP会员购买量。综上所述，这三种考研APP之间存在着一定的竞争关系，且APP的会员购买量（即APP的需求量）与APP的会员购买价格存在着相互影响的关系。

根据上述结论以及表格数据，针对本文所选择的三种考研APP：考研帮、口袋题库、研途宝，分别得到价格水平：，，；销售量：，，。利用本文所构建的定价模型可以求解得到三种考研APP需求对价格的灵敏系数参数：，，；，，；，， 。同时，价格变动系数为已知常数，每种产品的商品数量上限 也为已知常数，，，；。

这时来求解三种同类产品在一段时期内取得的投资收益最大问题。将这些系数参数带入模型后可知，目标函数是一个严格凸函数，而二次规划问题（11）是一个严格凸函数规划问题。这时求解严格的凸二次规划全局的问题便转换为求满足条件的点的问题。利用求解二次规划的有效集法，构造初始可行点。用MATLAB计算获得最优解。因此三种移动端考研APP的最优定价分别为：，，，由此可得移动端考研APP的区间价格是：[22.1021，92.1492]。同理，通过查找其他移动端考研产品的数据带入所建立的多产品定价决策模型可求得合理的产品价格区间。

5.2.3马尔可夫模型的建立与求解

5.2.3.1 模型准备

产品的市场占有率是企业战略管理和营销学中的一个重要概念，即某一产品在某一时期、某一地区的实际销售占整个行业的实际销售百分比。 本文考虑到实际市场当中的复杂多变，实际市场占有率易受到经济、社会属性、区域文化等多重因素的影响，具有一定的随机性，因此本文根据随机过程理论建立出马尔可夫模型来分析移动端考研产品在整个考研教育类产品市场中的状态分布，并且预测出该类产品在对应产品市场当中的占有率。

本文利用MATLAB通过对题目所提供的附件2中题目11、题目17、题目19的市场调查结果进行数据挖掘和分析，通过附件2中答卷内容总结出可供用户选择的三种学习渠道：线上PC端学习、线上移动端学习、线下学习，通过MATLAB编程分别得出选择这三类学习渠道人群的初始分布比率如表3所示：

表3：三类学习渠道人群的初始分布比率

|  |  |
| --- | --- |
| 学习渠道 | 比率 |
| 线上PC端学习 | 30% |
| 线上移动端 | 17% |
| 线下学习 | 53% |

本文首先将只选择了线下学习的数据直接划分到线下学习那一类，为了对模型进行简化，在对人群学习渠道的分类过程中删减掉了线上、线下学习两种都报和都不报的数据，然后为了判别出剩下数据所选择的具体线上学习渠道，即线上PC端学习或线上移动端学习，通过比较余下数据中平均每天花费在PC端学习的时长与花费在移动端学习的时长长短，将花费时间较长的那一类确定为被调查人所选择的具体线上学习渠道。

根据上述分析，本文定义：

（1）当PC端学习时长≥移动端学习时长，记为A类（即选择PC端学习）。

（2）当移动端学习时长≥PC端学习时长，记为B类（即选择移动端学习）。

（3）线下学习，记为C类。

根据上述的数据挖掘和分析计算可得出A类、B类、C类三种考研学习渠道在整个考研市场的预期分布表，如下表4所示：

表4：考研市场的预期分布

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 考研市场预期分布 | | | | | 合计 | |
| 来  自 |  | A | B | C | |  |
| A | 395 | 406 | 750 | | 1551 |
| B | 123 | 526 | 267 | | 916 |
| C | 174 | 289 | 2317 | | 2780 |
| 合计 |  | 692 | 1221 | 3334 | | 5247 |

**5.2.3.2**模型的建立与求解

本文记是由状态到状态的转移次数，行和是系统从状态状态到其他状态的次数，和的统计数据见表。一步转移状态矩阵的估计值是，根据上表（）数据计算可得一步转移状态矩阵的估计：



由一步转移矩阵可以推得一个稳定的市场占有率，当市场出现平衡时：



由此可得

经整理，并加上条件，得



上方程组是三个变量四个方程的方程组，在前三个方程中只有二个是独立的，任意删去一个，从剩下的三个方程中，可求出唯一解：

=0.098 ，=0.231 ，=0.671

由此得到线上移动端的潜在市场占有率是23%。

5.3问题三的模型的建立与求解

5.3.1 研究对象的选择与产品分析

研究对象的选择：根据题目要求：从经济、高校数量、社会属性、考研教育特点等几个方面考察来评价移动端产品的投放可行性，因此本文选择全国的经济中心、高校众多、每年的研究生统考报名人数逐年上升的上海市作为研究对象。

产品优势分析：

（1）本文结合学生学习的实际情况对移动端产品的适用性与实用性进行分析：学生群体的消费能力较高，考研教育市场份额巨大，随着国家经济的发展，科学教育水平的不断提升，移动互联网的兴起带动考研移动app市场进一步扩大。考研教育虽然属于高等教育，但更多的是面向应试并不能帮助学生学习到新的知识和方法，尤其针对理工科学生，学习成果很难速成，主要依靠平时学习的积累，但是学生参加考研辅导班能够在老师的帮助下理清思路、合理备考，帮助学习考研所考查的原理、概念、方法等套路性的内容，同时帮助学生选择自己理想的学校和专业。移动端产品的出现能够帮助学生随时随地的进行学习，有效的利用了时间，更有助于知识的积累与反复学习。

（2）考研教育面向的主要是没有经济收入的大学生，线下面授班收费价格相对于一般线上学习支付的价格较为昂贵，因此线上学习的方式容易对学生造成沉重的经济负担，同时也不利于学生自主地规划学习时间。

（3）根据问题一的建模与求解可知，由于智能手机的普及，考研app帮助学生合理利用了碎片化时间，随时随地的观看考研课程，提高了学习利用率，受到广泛的喜爱，因此人们的消费需求逐渐从PC端转换到移动端，同时认同使用移动app进行考研辅导的人数也在逐渐的增加。

5.3.2 产品可投放性评价

本文利用问题二所建立的马尔可夫预测模型，在对上海市考研教育市场进行了市场调研的基础上，本文对三年后上海市移动端考研app在整个考研教育市场中的市场占有率进行了预测，根据产品的市场占有率评估出该产品的投放可行性。

通过市场调研，得到现有考研教育市场的初始占有率分布，如表5所示：

表5：考研教育市场的初始占有率分布

|  |  |
| --- | --- |
| 学习渠道 | 占有率 |
| A类=线上PC端 | 34% |
| B类=线上移动端 | 19% |
| C类=线下学习 | 47% |

根据所查找到的数据可得出A类：、B类、C类三种考研学习渠道在整个考研市场的预期分布表，如下表6所示：

。

表6：考研市场的预期分布

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 考研市场预期分布 | | | | | 合计 | |
| 来  自 |  | A | B | C | |  |
| A | 140 | 132 | 68 | | 340 |
| B | 27 | 133 | 30 | | 190 |
| C | 32 | 149 | 289 | | 470 |
| 合计 |  | 199 | 414 | 387 | | 1000 |

计算得一步状态转移矩阵的估计为



设表示预测对象3年以后的市场占有率，初始分布则为，市场占有率的预测模型为

 （1-6）

由第一步可得：，因此本文可预测出任意时期A、B、C的市场占有率。

则三年以后的预测值为：

由一步转移矩阵可以推得一个稳定的市场占有率。由此得到线上移动端的潜在市场占有率是51.81%，推出移动端产品投放的可行性为51.81%。

# 六、模型的评价与推广

1、优点  
本文建立了定价模型和马尔可夫模型对问题进行求解，结合相关的影响因素，能很好的解决实际问题。  
本文所用算法，效率好，精度高，解决实际问题方便快捷。  
本文建立的模型完整，恰当，实用性强，便于推广。  
2、缺点   
本文模型假设比较理想化，具有特殊性。  
本文模型求解过程例举较少，降低了结果的说服性。  
3、模型的推广  
本文模型除了用于估计产品价格区间外还可以用于对产品进行最优定价、产品的市场分析，帮助制定精细的销售策略，有效提高了企业收益。

# 七、参考文献

[1]过晓芳，王宇平，同类产品的定价模型研究，技术经济与管理研究 2007年3期

# 八、附录

1.

p1\_pre.m：将分类属性与对称二元属性转化为“项”

clc, clear

%将分类属性与对称二元属性转化为“项”，以csv格式存储

%data即附件二，但对含有可自定义“其他”选项内容进行手动过滤，避免读取处理时产生类型错误

[NUM, TEXT, RAW] = xlsread('data', 1, 'B2:X10001');

p = '';

for i = 1:10000

if strcmp(RAW(i, 1), '男')

p = [p, 'p1\_1,'];

else

p = [p, 'p1\_2,'];

end

if strcmp(RAW(i, 2), '18—20岁')

p = [p, 'p2\_1,'];

elseif strcmp(RAW(i, 2), '20—23岁')

p = [p, 'p2\_2,'];

elseif strcmp(RAW(i, 2), '23—25岁')

p = [p, 'p2\_3,'];

else

p = [p, 'p2\_4,'];

end

if strcmp(RAW(i, 3), '北上广深等一线城市')

p = [p, 'p3\_1,'];

elseif strcmp(RAW(i, 3), '二线城市')

p = [p, 'p3\_2,'];

elseif strcmp(RAW(i, 3), '三线城市')

p = [p, 'p3\_3,'];

else

p = [p, 'p3\_4,'];

end

if strcmp(RAW(i, 4), '第一次')

p = [p, 'p4\_1,'];

elseif strcmp(RAW(i, 4), '第二次')

p = [p, 'p4\_2,'];

elseif strcmp(RAW(i, 4), '第三次')

p = [p, 'p4\_3,'];

else

p = [p, 'p4\_4,'];

end

if strcmp(RAW(i, 5), '提高就业竞争力，提升毕业薪资')

p = [p, 'p5\_1,'];

elseif strcmp(RAW(i, 5), '满足家人期望')

p = [p, 'p5\_2,'];

elseif strcmp(RAW(i, 5), '名校情结')

p = [p, 'p5\_3,'];

elseif strcmp(RAW(i, 5), '科研深造')

p = [p, 'p5\_4,'];

elseif strcmp(RAW(i, 5), '为了男/女朋友')

p = [p, 'p5\_5,'];

else

p = [p, 'p5\_6,'];

end

if strcmp(RAW(i, 6), '本校本专业')

p = [p, 'p6\_1,'];

elseif strcmp(RAW(i, 6), '本校跨专业')

p = [p, 'p6\_2,'];

elseif strcmp(RAW(i, 6), '跨校本专业')

p = [p, 'p6\_3,'];

else

p = [p, 'p6\_4,'];

end

if strcmp(RAW(i, 7), '线下渠道')

p = [p, 'p7\_1,'];

else

p = [p, 'p7\_2,'];

end

if strcmp(RAW(i, 8), '通过学长学姐介绍')

p = [p, 'p8\_1,'];

elseif strcmp(RAW(i, 8), '图书馆查阅资料')

p = [p, 'p8\_2,'];

else

p = [p, 'p8\_3,'];

end

if strcmp(RAW(i, 9), '考研网站及贴吧社区')

p = [p, 'p9\_1,'];

elseif strcmp(RAW(i, 9), '考研资讯App')

p = [p, 'p9\_2,'];

elseif strcmp(RAW(i, 9), '报考院校官网')

p = [p, 'p9\_3,'];

else

p = [p, 'p9\_4,'];

end

str = regexp(RAW(i, 10), '，', 'split');

[line, len] = size(str{1});

for t = 1:len

if strcmp(str{1}{t}, '权威可信')

p = [p, 'p10\_1,'];

elseif strcmp(str{1}{t}, '全面及时')

p = [p, 'p10\_2,'];

elseif strcmp(str{1}{t}, '方便快捷')

p = [p, 'p10\_3,'];

else

p = [p, 'p10\_4,'];

end

end

if strcmp(RAW(i, 11), '线上班')

p = [p, 'p11\_1,'];

elseif strcmp(RAW(i, 11), '线下班')

p = [p, 'p11\_2,'];

elseif strcmp(RAW(i, 11), '都报')

p = [p, 'p11\_3,'];

else

p = [p, 'p11\_4,'];

end

str = regexp(RAW(i, 12), '，', 'split');

[line, len] = size(str{1});

for t = 1:len

if strcmp(str{1}{t}, '时间安排灵活')

p = [p, 'p12\_1,'];

elseif strcmp(str{1}{t}, '无需往返课堂')

p = [p, 'p12\_2,'];

elseif strcmp(str{1}{t}, '节省交通成本')

elseif strcmp(str{1}{t}, '可重复性学习')

p = [p, 'p12\_3,'];

elseif strcmp(str{1}{t}, '价格便宜')

p = [p, 'p12\_4,'];

else

p = [p, 'p12\_5,'];

end

end

if strcmp(RAW(i, 13), '0—3个月')

p = [p, 'p13\_1,'];

elseif strcmp(RAW(i, 13), '3—6个月')

p = [p, 'p13\_2,'];

elseif strcmp(RAW(i, 13), '6—12个月')

p = [p, 'p13\_3,'];

else

p = [p, 'p13\_4,'];

end

if strcmp(RAW(i, 14), '自习室')

p = [p, 'p14\_1,'];

elseif strcmp(RAW(i, 14), '图书馆')

p = [p, 'p14\_2,'];

elseif strcmp(RAW(i, 14), '学校宿舍')

p = [p, 'p14\_3,'];

elseif strcmp(RAW(i, 14), '家里')

p = [p, 'p14\_4,'];

else

p = [p, 'p14\_5,'];

end

if strcmp(RAW(i, 15), '3-6小时')

p = [p, 'p15\_1,'];

elseif strcmp(RAW(i, 15), '6-9小时')

p = [p, 'p15\_2,'];

elseif strcmp(RAW(i, 15), '9-12小时')

p = [p, 'p15\_3,'];

else

p = [p, 'p15\_4,'];

end

if strcmp(RAW(i, 16), '线下渠道（含辅导班、讲座、自学）')

p = [p, 'p16\_1,'];

elseif strcmp(RAW(i, 16), '线上PC端')

p = [p, 'p16\_2,'];

else

p = [p, 'p16\_3,'];

end

if strcmp(RAW(i, 17), '0-1小时')

p = [p, 'p17\_1,'];

elseif strcmp(RAW(i, 17), '1-2小时')

p = [p, 'p17\_2,'];

elseif strcmp(RAW(i, 17), '2—3小时')

p = [p, 'p17\_3,'];

elseif strcmp(RAW(i, 17), '3—4小时')

p = [p, 'p17\_4,'];

else

p = [p, 'p17\_5,'];

end

if strcmp(RAW(i, 18), '考研课程辅导')

p = [p, 'p18\_1,'];

elseif strcmp(RAW(i, 18), '试题练习')

p = [p, 'p18\_2,'];

else

p = [p, 'p18\_3,'];

end

if strcmp(RAW(i, 19), '0-1小时')

p = [p, 'p19\_1,'];

elseif strcmp(RAW(i, 19), '1-2小时')

p = [p, 'p19\_2,'];

elseif strcmp(RAW(i, 19), '2-3小时')

p = [p, 'p19\_3,'];

elseif strcmp(RAW(i, 19), '3-4小时')

p = [p, 'p19\_4,'];

else

p = [p, 'p19\_5,'];

end

if strcmp(RAW(i, 20), '录播课程类产品 ')

p = [p, 'p20\_1,'];

elseif strcmp(RAW(i, 20), '直播课程类产品')

p = [p, 'p20\_2,'];

elseif strcmp(RAW(i, 20), '题库/练习类产品')

p = [p, 'p20\_3,'];

elseif strcmp(RAW(i, 20), '单词/词典类产品')

p = [p, 'p20\_4,'];

elseif strcmp(RAW(i, 20), '考点类产品')

p = [p, 'p20\_5,'];

else

p = [p, 'p20\_6,'];

end

if strcmp(RAW(i, 21), '不认同')

p = [p, 'p21\_1,'];

else

p = [p, 'p21\_2,'];

end

str = regexp(RAW(i, 22), '，', 'split');

[line, len] = size(str{1});

for t = 1:len

if strcmp(str{1}{t}, '随时随地')

p = [p, 'p22\_1,'];

elseif strcmp(str{1}{t}, '时间碎片化')

p = [p, 'p22\_2,'];

elseif strcmp(str{1}{t}, '学习利用率高')

elseif strcmp(str{1}{t}, '个性化推送功能更强')

p = [p, 'p22\_3,'];

elseif strcmp(str{1}{t}, '及时提醒功能更强')

p = [p, 'p22\_4,'];

else

p = [p, 'p22\_5,'];

end

end

if strcmp(RAW(i, 23), '更方便找到报考院校学长学姐获取相关经验')

p = [p, 'p23\_1'];

elseif strcmp(RAW(i, 23), '搭建研友交流互助平台')

p = [p, 'p23\_2'];

elseif strcmp(RAW(i, 23), '增加微讲座、微课程方便碎片化学习')

p = [p, 'p23\_3'];

elseif strcmp(RAW(i, 23), '集资讯、答疑、课程学习于一体')

p = [p, 'p23\_4'];

else

p = [p, 'p23\_5'];

end

p = [p, char(10)];

end

fid = fopen('p1.csv','w');

for i = 1:size(p, 1)

fprintf(fid, '%s\n', p(i,:));

end

fclose(fid);

**2、**

p1\_h\_conf.m：对支持度为920输出的数据去除交叉支持模式

clc, clear

%对支持度为920输出的数据去除交叉支持模式

[NUM, TEXT, RAW] = xlsread('data', 1, 'B2:X10001');

p = zeros(23, 6); %最多只有6个可选项

mol = 0;

conf = 0;

for i = 1:10000

if strcmp(RAW(i, 1), '男')

p(1, 1) = p(1, 1) + 1;

else

p(1, 2) = p(1, 2) + 1;

end

if strcmp(RAW(i, 2), '18—20岁')

p(2, 1) = p(2, 1) + 1;

elseif strcmp(RAW(i, 2), '20—23岁')

p(2, 2) = p(2, 2) + 1;

elseif strcmp(RAW(i, 2), '23—25岁')

p(2, 3) = p(2, 3) + 1;

else

p(2, 4) = p(2, 4) + 1;

end

if strcmp(RAW(i, 3), '北上广深等一线城市')

p(3, 1) = p(3, 1) + 1;

elseif strcmp(RAW(i, 3), '二线城市')

p(3, 2) = p(3, 2) + 1;

elseif strcmp(RAW(i, 3), '三线城市')

p(3, 3) = p(3, 3) + 1;

else

p(3, 4) = p(3, 4) + 1;

end

if strcmp(RAW(i, 4), '第一次')

p(4, 1) = p(4, 1) + 1;

elseif strcmp(RAW(i, 4), '第二次')

p(4, 2) = p(4, 2) + 1;

elseif strcmp(RAW(i, 4), '第三次')

p(4, 3) = p(4, 3) + 1;

else

p(4, 4) = p(4, 4) + 1;

end

if strcmp(RAW(i, 5), '提高就业竞争力，提升毕业薪资')

p(5, 1) = p(5, 1) + 1;

elseif strcmp(RAW(i, 5), '满足家人期望')

p(5, 2) = p(5, 2) + 1;

elseif strcmp(RAW(i, 5), '名校情结')

p(5, 3) = p(5, 3) + 1;

elseif strcmp(RAW(i, 5), '科研深造')

p(5, 4) = p(5, 4) + 1;

elseif strcmp(RAW(i, 5), '为了男/女朋友')

p(5, 5) = p(5, 5) + 1;

else

p(6, 5) = p(6, 5) + 1;

end

if strcmp(RAW(i, 6), '本校本专业')

p(6, 1) = p(6, 1) + 1;

elseif strcmp(RAW(i, 6), '本校跨专业')

p(6, 2) = p(6, 2) + 1;

elseif strcmp(RAW(i, 6), '跨校本专业')

p(6, 3) = p(6, 3) + 1;

else

p(6, 4) = p(6, 4) + 1;

end

if strcmp(RAW(i, 7), '线下渠道')

p(7, 1) = p(7, 1) + 1;

else

p(7, 2) = p(7, 2) + 1;

end

if strcmp(RAW(i, 8), '通过学长学姐介绍')

p(8, 1) = p(8, 1) + 1;

elseif strcmp(RAW(i, 8), '图书馆查阅资料')

p(8, 2) = p(8, 2) + 1;

else

p(8, 3) = p(8, 3) + 1;

end

if strcmp(RAW(i, 9), '考研网站及贴吧社区')

p(9, 1) = p(9, 1) + 1;

elseif strcmp(RAW(i, 9), '考研资讯App')

p(9, 2) = p(9, 2) + 1;

elseif strcmp(RAW(i, 9), '报考院校官网')

p(9, 3) = p(9, 3) + 1;

else

p(9, 4) = p(9, 4) + 1;

end

str = regexp(RAW(i, 10), '，', 'split');

[line, len] = size(str{1});

for t = 1:len

if strcmp(str{1}{t}, '权威可信')

p(10, 1) = p(10, 1) + 1;

elseif strcmp(str{1}{t}, '全面及时')

p(10, 2) = p(10, 2) + 1;

elseif strcmp(str{1}{t}, '方便快捷')

p(10, 3) = p(10, 3) + 1;

else

p(10, 4) = p(10, 4) + 1;

end

end

if strcmp(RAW(i, 11), '线上班')

p(11, 1) = p(11, 1) + 1;

elseif strcmp(RAW(i, 11), '线下班')

p(11, 2) = p(11, 2) + 1;

elseif strcmp(RAW(i, 11), '都报')

p(11, 3) = p(11, 3) + 1;

else

p(11, 4) = p(11, 4) + 1;

end

str = regexp(RAW(i, 12), '，', 'split');

[line, len] = size(str{1});

for t = 1:len

if strcmp(str{1}{t}, '时间安排灵活')

p(12, 1) = p(12, 1) + 1;

elseif strcmp(str{1}{t}, '无需往返课堂')

p(12, 2) = p(12, 2) + 1;

elseif strcmp(str{1}{t}, '节省交通成本')

elseif strcmp(str{1}{t}, '可重复性学习')

p(12, 3) = p(12, 3) + 1;

elseif strcmp(str{1}{t}, '价格便宜')

p(12, 4) = p(12, 4) + 1;

else

p(12, 5) = p(12, 5) + 1;

end

end

if strcmp(RAW(i, 13), '0—3个月')

p(13, 1) = p(13, 1) + 1;

elseif strcmp(RAW(i, 13), '3—6个月')

p(13, 2) = p(13, 2) + 1;

elseif strcmp(RAW(i, 13), '6—12个月')

p(13, 3) = p(13, 3) + 1;

else

p(13, 4) = p(13, 4) + 1;

end

if strcmp(RAW(i, 14), '自习室')

p(14, 1) = p(14, 1) + 1;

elseif strcmp(RAW(i, 14), '图书馆')

p(14, 2) = p(14, 2) + 1;

elseif strcmp(RAW(i, 14), '学校宿舍')

p(14, 3) = p(14, 3) + 1;

elseif strcmp(RAW(i, 14), '家里')

p(14, 4) = p(14, 4) + 1;

else

p(14, 5) = p(14, 5) + 1;

end

if strcmp(RAW(i, 15), '3-6小时')

p(15, 1) = p(15, 1) + 1;

elseif strcmp(RAW(i, 15), '6-9小时')

p(15, 2) = p(15, 2) + 1;

elseif strcmp(RAW(i, 15), '9-12小时')

p(15, 3) = p(15, 3) + 1;

else

p(15, 4) = p(15, 4) + 1;

end

if strcmp(RAW(i, 16), '线下渠道（含辅导班、讲座、自学）')

p(16, 1) = p(16, 1) + 1;

elseif strcmp(RAW(i, 16), '线上PC端')

p(16, 2) = p(16, 2) + 1;

else

p(16, 3) = p(16, 3) + 1;

end

if strcmp(RAW(i, 17), '0-1小时')

p(17, 1) = p(17, 1) + 1;

elseif strcmp(RAW(i, 17), '1-2小时')

p(17, 2) = p(17, 2) + 1;

elseif strcmp(RAW(i, 17), '2—3小时')

p(17, 3) = p(17, 3) + 1;

elseif strcmp(RAW(i, 17), '3—4小时')

p(17, 4) = p(17, 4) + 1;

else

p(17, 5) = p(17, 5) + 1;

end

if strcmp(RAW(i, 18), '考研课程辅导')

p(18, 1) = p(18, 1) + 1;

elseif strcmp(RAW(i, 18), '试题练习')

p(18, 2) = p(18, 2) + 1;

else

p(18, 3) = p(18, 3) + 1;

end

if strcmp(RAW(i, 19), '0-1小时')

p(19, 1) = p(19, 1) + 1;

elseif strcmp(RAW(i, 19), '1-2小时')

p(19, 2) = p(19, 2) + 1;

elseif strcmp(RAW(i, 19), '2-3小时')

p(19, 3) = p(19, 3) + 1;

elseif strcmp(RAW(i, 19), '3-4小时')

p(19, 4) = p(19, 4) + 1;

else

p(19, 5) = p(19, 5) + 1;

end

if strcmp(RAW(i, 20), '录播课程类产品 ')

p(20, 1) = p(20, 1) + 1;

elseif strcmp(RAW(i, 20), '直播课程类产品')

p(20, 2) = p(20, 2) + 1;

elseif strcmp(RAW(i, 20), '题库/练习类产品')

p(20, 3) = p(20, 3) + 1;

elseif strcmp(RAW(i, 20), '单词/词典类产品')

p(20, 4) = p(20, 4) + 1;

elseif strcmp(RAW(i, 20), '考点类产品')

p(20, 5) = p(20, 5) + 1;

else

p(20, 6) = p(20, 6) + 1;

end

if strcmp(RAW(i, 21), '不认同')

p(21, 2) = p(21, 2) + 1;

else

p(21, 1) = p(21, 1) + 1;

end

str = regexp(RAW(i, 22), '，', 'split');

[line, len] = size(str{1});

for t = 1:len

if strcmp(str{1}{t}, '随时随地')

p(22, 1) = p(22, 1) + 1;

elseif strcmp(str{1}{t}, '时间碎片化')

p(22, 2) = p(22, 2) + 1;

elseif strcmp(str{1}{t}, '学习利用率高')

elseif strcmp(str{1}{t}, '个性化推送功能更强')

p(22, 3) = p(22, 3) + 1;

elseif strcmp(str{1}{t}, '及时提醒功能更强')

p(22, 4) = p(22, 4) + 1;

else

p(22, 5) = p(22, 5) + 1;

end

end

if strcmp(RAW(i, 23), '更方便找到报考院校学长学姐获取相关经验')

p(23, 1) = p(23, 1) + 1;

elseif strcmp(RAW(i, 23), '搭建研友交流互助平台')

p(23, 2) = p(23, 2) + 1;

elseif strcmp(RAW(i, 23), '增加微讲座、微课程方便碎片化学习')

p(23, 3) = p(23, 3) + 1;

elseif strcmp(RAW(i, 23), '集资讯、答疑、课程学习于一体')

p(23, 4) = p(23, 4) + 1;

else

p(23, 5) = p(23, 5) + 1;

end

end

[NUM, TEXT, RAW] = xlsread('920', 1, 'A1:G1032');

cou = 0;

data = '';

for x = 1:1032

max\_m = 0;

min\_m = 10000000000;

for y = 1:7

st = RAW(x, y);

if isnan(st{1}) == 0

str = regexp(RAW(x, y), '\_', 'split');

[line, len] = size(str{1});

str{1}{1}(1) = [];

max\_m = max(max\_m, p(str2double(str{1}{1}), str2double(str{1}{2})));

min\_m = min(min\_m, p(str2double(str{1}{1}), str2double(str{1}{2})));

end

end

if ( min\_m / 10000 ) / ( max\_m / 10000 ) >= 0.30

for y = 1:7

st = RAW(x, y);

if isnan(st{1}) == 0

str = RAW(x, y);

data = [data, ' ', str{1}];

end

end

data = [data, char(10)];

cou = cou + 1;

end

end

fid = fopen('p1\_h\_data.csv','w');

for i = 1:size(data, 1)

fprintf(fid, '%s\n', data(i,:));

end

fclose(fid);

['OK! H Confidence pass count: ', num2str(cou)]

**3、**

p1\_final.m：对去除交叉支持模式后产生的222条数据统计支持度

clc, clear

%对去除交叉支持模式后产生的222条数据统计支持度，输出为p1\_final.xls

[NUM, TEXT, RAW] = xlsread('data', 1, 'B2:X10001');

[N1, T1, R1] = xlsread('p1\_h\_data', 1, 'A1:E222');

mat = zeros(222, 2);

for ct = 1:222

support = 0;

for i = 1:10000

p = zeros(23, 6);

if strcmp(RAW(i, 1), '男')

p(1, 1) = p(1, 1) + 1;

else

p(1, 2) = p(1, 2) + 1;

end

if strcmp(RAW(i, 2), '18—20岁')

p(2, 1) = p(2, 1) + 1;

elseif strcmp(RAW(i, 2), '20—23岁')

p(2, 2) = p(2, 2) + 1;

elseif strcmp(RAW(i, 2), '23—25岁')

p(2, 3) = p(2, 3) + 1;

else

p(2, 4) = p(2, 4) + 1;

end

if strcmp(RAW(i, 3), '北上广深等一线城市')

p(3, 1) = p(3, 1) + 1;

elseif strcmp(RAW(i, 3), '二线城市')

p(3, 2) = p(3, 2) + 1;

elseif strcmp(RAW(i, 3), '三线城市')

p(3, 3) = p(3, 3) + 1;

else

p(3, 4) = p(3, 4) + 1;

end

if strcmp(RAW(i, 4), '第一次')

p(4, 1) = p(4, 1) + 1;

elseif strcmp(RAW(i, 4), '第二次')

p(4, 2) = p(4, 2) + 1;

elseif strcmp(RAW(i, 4), '第三次')

p(4, 3) = p(4, 3) + 1;

else

p(4, 4) = p(4, 4) + 1;

end

if strcmp(RAW(i, 5), '提高就业竞争力，提升毕业薪资')

p(5, 1) = p(5, 1) + 1;

elseif strcmp(RAW(i, 5), '满足家人期望')

p(5, 2) = p(5, 2) + 1;

elseif strcmp(RAW(i, 5), '名校情结')

p(5, 3) = p(5, 3) + 1;

elseif strcmp(RAW(i, 5), '科研深造')

p(5, 4) = p(5, 4) + 1;

elseif strcmp(RAW(i, 5), '为了男/女朋友')

p(5, 5) = p(5, 5) + 1;

else

p(6, 5) = p(6, 5) + 1;

end

if strcmp(RAW(i, 6), '本校本专业')

p(6, 1) = p(6, 1) + 1;

elseif strcmp(RAW(i, 6), '本校跨专业')

p(6, 2) = p(6, 2) + 1;

elseif strcmp(RAW(i, 6), '跨校本专业')

p(6, 3) = p(6, 3) + 1;

else

p(6, 4) = p(6, 4) + 1;

end

if strcmp(RAW(i, 7), '线下渠道')

p(7, 1) = p(7, 1) + 1;

else

p(7, 2) = p(7, 2) + 1;

end

if strcmp(RAW(i, 8), '通过学长学姐介绍')

p(8, 1) = p(8, 1) + 1;

elseif strcmp(RAW(i, 8), '图书馆查阅资料')

p(8, 2) = p(8, 2) + 1;

else

p(8, 3) = p(8, 3) + 1;

end

if strcmp(RAW(i, 9), '考研网站及贴吧社区')

p(9, 1) = p(9, 1) + 1;

elseif strcmp(RAW(i, 9), '考研资讯App')

p(9, 2) = p(9, 2) + 1;

elseif strcmp(RAW(i, 9), '报考院校官网')

p(9, 3) = p(9, 3) + 1;

else

p(9, 4) = p(9, 4) + 1;

end

str = regexp(RAW(i, 10), '，', 'split');

[line, len] = size(str{1});

for t = 1:len

if strcmp(str{1}{t}, '权威可信')

p(10, 1) = p(10, 1) + 1;

elseif strcmp(str{1}{t}, '全面及时')

p(10, 2) = p(10, 2) + 1;

elseif strcmp(str{1}{t}, '方便快捷')

p(10, 3) = p(10, 3) + 1;

else

p(10, 4) = p(10, 4) + 1;

end

end

if strcmp(RAW(i, 11), '线上班')

p(11, 1) = p(11, 1) + 1;

elseif strcmp(RAW(i, 11), '线下班')

p(11, 2) = p(11, 2) + 1;

elseif strcmp(RAW(i, 11), '都报')

p(11, 3) = p(11, 3) + 1;

else

p(11, 4) = p(11, 4) + 1;

end

str = regexp(RAW(i, 12), '，', 'split');

[line, len] = size(str{1});

for t = 1:len

if strcmp(str{1}{t}, '时间安排灵活')

p(12, 1) = p(12, 1) + 1;

elseif strcmp(str{1}{t}, '无需往返课堂')

p(12, 2) = p(12, 2) + 1;

elseif strcmp(str{1}{t}, '节省交通成本')

elseif strcmp(str{1}{t}, '可重复性学习')

p(12, 3) = p(12, 3) + 1;

elseif strcmp(str{1}{t}, '价格便宜')

p(12, 4) = p(12, 4) + 1;

else

p(12, 5) = p(12, 5) + 1;

end

end

if strcmp(RAW(i, 13), '0—3个月')

p(13, 1) = p(13, 1) + 1;

elseif strcmp(RAW(i, 13), '3—6个月')

p(13, 2) = p(13, 2) + 1;

elseif strcmp(RAW(i, 13), '6—12个月')

p(13, 3) = p(13, 3) + 1;

else

p(13, 4) = p(13, 4) + 1;

end

if strcmp(RAW(i, 14), '自习室')

p(14, 1) = p(14, 1) + 1;

elseif strcmp(RAW(i, 14), '图书馆')

p(14, 2) = p(14, 2) + 1;

elseif strcmp(RAW(i, 14), '学校宿舍')

p(14, 3) = p(14, 3) + 1;

elseif strcmp(RAW(i, 14), '家里')

p(14, 4) = p(14, 4) + 1;

else

p(14, 5) = p(14, 5) + 1;

end

if strcmp(RAW(i, 15), '3-6小时')

p(15, 1) = p(15, 1) + 1;

elseif strcmp(RAW(i, 15), '6-9小时')

p(15, 2) = p(15, 2) + 1;

elseif strcmp(RAW(i, 15), '9-12小时')

p(15, 3) = p(15, 3) + 1;

else

p(15, 4) = p(15, 4) + 1;

end

if strcmp(RAW(i, 16), '线下渠道（含辅导班、讲座、自学）')

p(16, 1) = p(16, 1) + 1;

elseif strcmp(RAW(i, 16), '线上PC端')

p(16, 2) = p(16, 2) + 1;

else

p(16, 3) = p(16, 3) + 1;

end

if strcmp(RAW(i, 17), '0-1小时')

p(17, 1) = p(17, 1) + 1;

elseif strcmp(RAW(i, 17), '1-2小时')

p(17, 2) = p(17, 2) + 1;

elseif strcmp(RAW(i, 17), '2—3小时')

p(17, 3) = p(17, 3) + 1;

elseif strcmp(RAW(i, 17), '3—4小时')

p(17, 4) = p(17, 4) + 1;

else

p(17, 5) = p(17, 5) + 1;

end

if strcmp(RAW(i, 18), '考研课程辅导')

p(18, 1) = p(18, 1) + 1;

elseif strcmp(RAW(i, 18), '试题练习')

p(18, 2) = p(18, 2) + 1;

else

p(18, 3) = p(18, 3) + 1;

end

if strcmp(RAW(i, 19), '0-1小时')

p(19, 1) = p(19, 1) + 1;

elseif strcmp(RAW(i, 19), '1-2小时')

p(19, 2) = p(19, 2) + 1;

elseif strcmp(RAW(i, 19), '2-3小时')

p(19, 3) = p(19, 3) + 1;

elseif strcmp(RAW(i, 19), '3-4小时')

p(19, 4) = p(19, 4) + 1;

else

p(19, 5) = p(19, 5) + 1;

end

if strcmp(RAW(i, 20), '录播课程类产品 ')

p(20, 1) = p(20, 1) + 1;

elseif strcmp(RAW(i, 20), '直播课程类产品')

p(20, 2) = p(20, 2) + 1;

elseif strcmp(RAW(i, 20), '题库/练习类产品')

p(20, 3) = p(20, 3) + 1;

elseif strcmp(RAW(i, 20), '单词/词典类产品')

p(20, 4) = p(20, 4) + 1;

elseif strcmp(RAW(i, 20), '考点类产品')

p(20, 5) = p(20, 5) + 1;

else

p(20, 6) = p(20, 6) + 1;

end

if strcmp(RAW(i, 21), '不认同')

p(21, 2) = p(21, 2) + 1;

else

p(21, 1) = p(21, 1) + 1;

end

str = regexp(RAW(i, 22), '，', 'split');

[line, len] = size(str{1});

for t = 1:len

if strcmp(str{1}{t}, '随时随地')

p(22, 1) = p(22, 1) + 1;

elseif strcmp(str{1}{t}, '时间碎片化')

p(22, 2) = p(22, 2) + 1;

elseif strcmp(str{1}{t}, '学习利用率高')

elseif strcmp(str{1}{t}, '个性化推送功能更强')

p(22, 3) = p(22, 3) + 1;

elseif strcmp(str{1}{t}, '及时提醒功能更强')

p(22, 4) = p(22, 4) + 1;

else

p(22, 5) = p(22, 5) + 1;

end

end

if strcmp(RAW(i, 23), '更方便找到报考院校学长学姐获取相关经验')

p(23, 1) = p(23, 1) + 1;

elseif strcmp(RAW(i, 23), '搭建研友交流互助平台')

p(23, 2) = p(23, 2) + 1;

elseif strcmp(RAW(i, 23), '增加微讲座、微课程方便碎片化学习')

p(23, 3) = p(23, 3) + 1;

elseif strcmp(RAW(i, 23), '集资讯、答疑、课程学习于一体')

p(23, 4) = p(23, 4) + 1;

else

p(23, 5) = p(23, 5) + 1;

end

is\_record = 1;

for y = 1:5

st = R1(ct, y);

if isnan(st{1}) == 0

str = regexp(R1(ct, y), '\_', 'split');

[line, len] = size(str{1});

str{1}{1}(1) = [];

if p(str2double(str{1}{1}), str2double(str{1}{2})) == 0

is\_record = 0;

end

elseif is\_record == 1

support = support + 1;

end

end

end

ct

mat(ct, 1) = ct;

mat(ct, 2) = support;

mat

end

xlswrite('p1\_final.xls', mat);

**4、**

p2.m：输出马尔科夫链预测市场占有率的初始状态和状态矩阵元素

clc, clear

%输出马尔科夫链预测市场占有率的初始状态和状态矩阵元素

[NUM, TEXT, RAW] = xlsread('data', 1, 'B2:X10001');

p = zeros(23, 6); %最多只有6个可选项

topc = 0;

tomo = 0;

offline = 0;

s = zeros(3);

for i = 1:10000

if strcmp(RAW(i, 17), '0-1小时')

pc = 0;

elseif strcmp(RAW(i, 17), '1-2小时')

pc = 1;

elseif strcmp(RAW(i, 17), '2—3小时')

pc = 2;

elseif strcmp(RAW(i, 17), '3—4小时')

pc = 3;

else

pc = 4;

end

if strcmp(RAW(i, 19), '0-1小时')

mobile = 0;

elseif strcmp(RAW(i, 19), '1-2小时')

mobile = 1;

elseif strcmp(RAW(i, 19), '2—3小时')

mobile = 2;

elseif strcmp(RAW(i, 19), '3—4小时')

mobile = 3;

else

mobile = 4;

end

if strcmp(RAW(i, 16), '线下渠道（含辅导班、讲座、自学）')

now\_s = 3;

elseif strcmp(RAW(i, 16), '线上PC端')

now\_s = 1;

else

now\_s = 2;

end

if strcmp(RAW(i, 11), '线上班')

if pc >= mobile

topc = topc + 1;

next\_s = 1;

else

tomo = tomo + 1;

next\_s = 2;

end

elseif strcmp(RAW(i, 11), '线下班')

offline = offline + 1;

next\_s = 3;

else

next\_s = 0;

end

if next\_s ~= 0

s(next\_s, now\_s) = s(next\_s, now\_s) + 1;

end

end

topc

tomo

offline

s

sum(s(1,:))

s(1,:)./sum(s(1,:))

sum(s(2,:))

s(2,:)./sum(s(2,:))

sum(s(3,:))

s(3,:)./sum(s(3,:))

**5、**

fp\_growth.py：本文所使用的FP-growth开源程序，来源 https://github.com/enaeseth/python-fp-growth# encoding: utf-8

"""

A Python implementation of the FP-growth algorithm.

Basic usage of the module is very simple:

> from fp\_growth import find\_frequent\_itemsets

> find\_frequent\_itemsets(transactions, minimum\_support)

"""

from collections import defaultdict, namedtuple

from itertools import imap

\_\_author\_\_ = 'Eric Naeseth <eric@naeseth.com>'

\_\_copyright\_\_ = 'Copyright © 2009 Eric Naeseth'

\_\_license\_\_ = 'MIT License'

def find\_frequent\_itemsets(transactions, minimum\_support, include\_support=False):

"""

Find frequent itemsets in the given transactions using FP-growth. This

function returns a generator instead of an eagerly-populated list of items.

The `transactions` parameter can be any iterable of iterables of items.

`minimum\_support` should be an integer specifying the minimum number of

occurrences of an itemset for it to be accepted.

Each item must be hashable (i.e., it must be valid as a member of a

dictionary or a set).

If `include\_support` is true, yield (itemset, support) pairs instead of

just the itemsets.

"""

items = defaultdict(lambda: 0) # mapping from items to their supports

# Load the passed-in transactions and count the support that individual

# items have.

for transaction in transactions:

for item in transaction:

items[item] += 1

# Remove infrequent items from the item support dictionary.

items = dict((item, support) for item, support in items.iteritems()

if support >= minimum\_support)

# Build our FP-tree. Before any transactions can be added to the tree, they

# must be stripped of infrequent items and their surviving items must be

# sorted in decreasing order of frequency.

def clean\_transaction(transaction):

transaction = filter(lambda v: v in items, transaction)

transaction.sort(key=lambda v: items[v], reverse=True)

return transaction

master = FPTree()

for transaction in imap(clean\_transaction, transactions):

master.add(transaction)

def find\_with\_suffix(tree, suffix):

for item, nodes in tree.items():

support = sum(n.count for n in nodes)

if support >= minimum\_support and item not in suffix:

# New winner!

found\_set = [item] + suffix

yield (found\_set, support) if include\_support else found\_set

# Build a conditional tree and recursively search for frequent

# itemsets within it.

cond\_tree = conditional\_tree\_from\_paths(tree.prefix\_paths(item))

for s in find\_with\_suffix(cond\_tree, found\_set):

yield s # pass along the good news to our caller

# Search for frequent itemsets, and yield the results we find.

for itemset in find\_with\_suffix(master, []):

yield itemset

class FPTree(object):

"""

An FP tree.

This object may only store transaction items that are hashable

(i.e., all items must be valid as dictionary keys or set members).

"""

Route = namedtuple('Route', 'head tail')

def \_\_init\_\_(self):

# The root node of the tree.

self.\_root = FPNode(self, None, None)

# A dictionary mapping items to the head and tail of a path of

# "neighbors" that will hit every node containing that item.

self.\_routes = {}

@property

def root(self):

"""The root node of the tree."""

return self.\_root

def add(self, transaction):

"""Add a transaction to the tree."""

point = self.\_root

for item in transaction:

next\_point = point.search(item)

if next\_point:

# There is already a node in this tree for the current

# transaction item; reuse it.

next\_point.increment()

else:

# Create a new point and add it as a child of the point we're

# currently looking at.

next\_point = FPNode(self, item)

point.add(next\_point)

# Update the route of nodes that contain this item to include

# our new node.

self.\_update\_route(next\_point)

point = next\_point

def \_update\_route(self, point):

"""Add the given node to the route through all nodes for its item."""

assert self is point.tree

try:

route = self.\_routes[point.item]

route[1].neighbor = point # route[1] is the tail

self.\_routes[point.item] = self.Route(route[0], point)

except KeyError:

# First node for this item; start a new route.

self.\_routes[point.item] = self.Route(point, point)

def items(self):

"""

Generate one 2-tuples for each item represented in the tree. The first

element of the tuple is the item itself, and the second element is a

generator that will yield the nodes in the tree that belong to the item.

"""

for item in self.\_routes.iterkeys():

yield (item, self.nodes(item))

def nodes(self, item):

"""

Generate the sequence of nodes that contain the given item.

"""

try:

node = self.\_routes[item][0]

except KeyError:

return

while node:

yield node

node = node.neighbor

def prefix\_paths(self, item):

"""Generate the prefix paths that end with the given item."""

def collect\_path(node):

path = []

while node and not node.root:

path.append(node)

node = node.parent

path.reverse()

return path

return (collect\_path(node) for node in self.nodes(item))

def inspect(self):

print 'Tree:'

self.root.inspect(1)

print

print 'Routes:'

for item, nodes in self.items():

print ' %r' % item

for node in nodes:

print ' %r' % node

def conditional\_tree\_from\_paths(paths):

"""Build a conditional FP-tree from the given prefix paths."""

tree = FPTree()

condition\_item = None

items = set()

# Import the nodes in the paths into the new tree. Only the counts of the

# leaf notes matter; the remaining counts will be reconstructed from the

# leaf counts.

for path in paths:

if condition\_item is None:

condition\_item = path[-1].item

point = tree.root

for node in path:

next\_point = point.search(node.item)

if not next\_point:

# Add a new node to the tree.

items.add(node.item)

count = node.count if node.item == condition\_item else 0

next\_point = FPNode(tree, node.item, count)

point.add(next\_point)

tree.\_update\_route(next\_point)

point = next\_point

assert condition\_item is not None

# Calculate the counts of the non-leaf nodes.

for path in tree.prefix\_paths(condition\_item):

count = path[-1].count

for node in reversed(path[:-1]):

node.\_count += count

return tree

class FPNode(object):

"""A node in an FP tree."""

def \_\_init\_\_(self, tree, item, count=1):

self.\_tree = tree

self.\_item = item

self.\_count = count

self.\_parent = None

self.\_children = {}

self.\_neighbor = None

def add(self, child):

"""Add the given FPNode `child` as a child of this node."""

if not isinstance(child, FPNode):

raise TypeError("Can only add other FPNodes as children")

if not child.item in self.\_children:

self.\_children[child.item] = child

child.parent = self

def search(self, item):

"""

Check whether this node contains a child node for the given item.

If so, that node is returned; otherwise, `None` is returned.

"""

try:

return self.\_children[item]

except KeyError:

return None

def \_\_contains\_\_(self, item):

return item in self.\_children

@property

def tree(self):

"""The tree in which this node appears."""

return self.\_tree

@property

def item(self):

"""The item contained in this node."""

return self.\_item

@property

def count(self):

"""The count associated with this node's item."""

return self.\_count

def increment(self):

"""Increment the count associated with this node's item."""

if self.\_count is None:

raise ValueError("Root nodes have no associated count.")

self.\_count += 1

@property

def root(self):

"""True if this node is the root of a tree; false if otherwise."""

return self.\_item is None and self.\_count is None

@property

def leaf(self):

"""True if this node is a leaf in the tree; false if otherwise."""

return len(self.\_children) == 0

@property

def parent(self):

"""The node's parent"""

return self.\_parent

@parent.setter

def parent(self, value):

if value is not None and not isinstance(value, FPNode):

raise TypeError("A node must have an FPNode as a parent.")

if value and value.tree is not self.tree:

raise ValueError("Cannot have a parent from another tree.")

self.\_parent = value

@property

def neighbor(self):

"""

The node's neighbor; the one with the same value that is "to the right"

of it in the tree.

"""

return self.\_neighbor

@neighbor.setter

def neighbor(self, value):

if value is not None and not isinstance(value, FPNode):

raise TypeError("A node must have an FPNode as a neighbor.")

if value and value.tree is not self.tree:

raise ValueError("Cannot have a neighbor from another tree.")

self.\_neighbor = value

@property

def children(self):

"""The nodes that are children of this node."""

return tuple(self.\_children.itervalues())

def inspect(self, depth=0):

print (' ' \* depth) + repr(self)

for child in self.children:

child.inspect(depth + 1)

def \_\_repr\_\_(self):

if self.root:

return "<%s (root)>" % type(self).\_\_name\_\_

return "<%s %r (%r)>" % (type(self).\_\_name\_\_, self.item, self.count)

if \_\_name\_\_ == '\_\_main\_\_':

from optparse import OptionParser

import csv

p = OptionParser(usage='%prog data\_file')

p.add\_option('-s', '--minimum-support', dest='minsup', type='int',

help='Minimum itemset support (default: 2)')

p.add\_option('-n', '--numeric', dest='numeric', action='store\_true',

help='Convert the values in datasets to numerals (default: false)')

p.set\_defaults(minsup=2)

p.set\_defaults(numeric=False)

options, args = p.parse\_args()

if len(args) < 1:

p.error('must provide the path to a CSV file to read')

transactions = []

with open(args[0]) as database:

for row in csv.reader(database):

if options.numeric:

transaction = []

for item in row:

transaction.append(long(item))

transactions.append(transaction)

else:

transactions.append(row)

result = []

for itemset, support in find\_frequent\_itemsets(transactions, options.minsup, True):

result.append((itemset,support))

result = sorted(result, key=lambda i: i[0])

for itemset, support in result:

print str(itemset) + ' ' + str(support)