## 第2小组的数据挖掘作业1

小组成员：高鹏昺 蒋世豪 李进雄 周亮 苏金涛 刘昊轩

# 方法1 ：采用Apriori算法进行数据挖掘

1.对于采用Apriori算法的初步构想

考虑到给出的双色球数据是关于红球的数据序列。并且其数据序列频繁地出现在以时间为单位的历史数据库中，所以可以将关于红球的数据看作是一个频繁序列模式。那么对于双色球之中红球的序列预测，我们可以通过挖掘频繁6项集来做到一定程度的预测。

2.基于Apriori的预测方法建立

（1）相关原理介绍

Apriori（先验）算法是一种发现频繁项集的基本算法。其通过限制候选产生发现频繁项集。

使用一种称为逐层搜索的迭代方法，其中*k*项集用于搜索（*k+*1）项集。首先，通过扫描给出的数据，累积每个项的计数，并收集满足最小支持度的项，找出频繁1项集的集合。该集合即为。然后，使用找出频繁2项集的集合，使用找出，如此下去，直到不能再找到频繁*k*项集。

为了提高频繁项集逐层产生的效率，引入一种称为先验性质的重要性质进行压缩搜索空间。

根据定义，如果项集*I*不满足最小支持度阈值*min\_sup*，则*I*不是频繁的，即*P(I)<min\_sup*。

接着，考量如何通过先验性质，使用找出

**(Ⅰ)连接步：**为找出，通过将与自身连接产生候选*k*项集的集合。该候选项集的集合记为。设和是中的项集。记号表示的第*j*项（例如，表示的倒数第2项）。为了有效地实现，Apriori算法假定事务或项集中的项按字典序排序。对于(*k-*1)项集，这意味着把项排序，使得。执行连接；其中的元素是可连接的，如果它们前(*k-2*)个项相同。即，的元素和是可连接的，如。条件是简单地确保不产生重复。连接和产生的结果项集是。

**(Ⅱ)剪枝步：**是的超集，也就是说，的成员可以是也可以不是频繁的，但所有的频繁*k*项集都包含在中。扫描数据时，确定中每个候选的计数，从而确定（即根据定义，计数值不小于最小支持度计数的所有候选都是频繁的，从而属于）。然而，可能很大，因此所涉及的计数量就很大。为了压缩，可以用以下办法使用先验性质。任何非频繁的(*k-*1)项集都不是频繁*k*项集的子集。因此，如果一个候选*k*项集的(*k*-1)项子集不在中，则该候选集也不可能是频繁的，从而可以从中删除。这种子集测试可以使用素有频繁项集的散列树快速完成。

（2）具体的操作实现

①导入数据

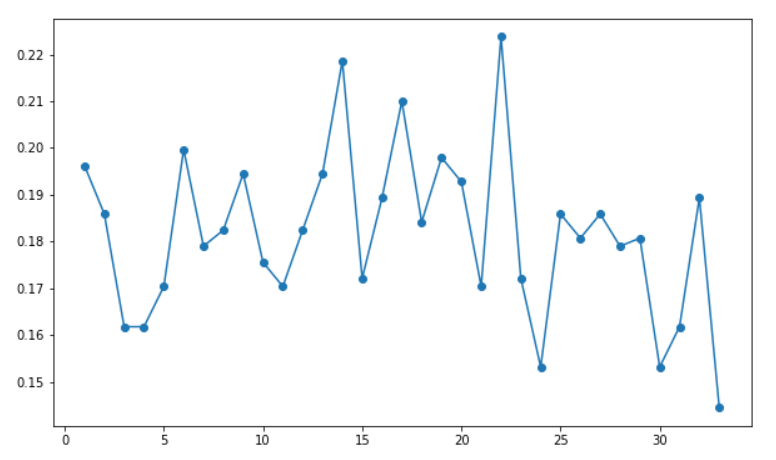
②通过概率计算函数进行概率计算，并对频繁1项集概率分布进行绘图，从而确定较为优秀的*min\_sup*取值。

③根据上面的理论论述，实现集合计数、集合剪枝等一系列功能，构造Apriori算法代码。

④调用Apriori算法，输入数据，求出符合规则的频繁6项集

3.输出结果

在数据探索阶段我们可以通过计算红球各号码的概率，并以此为依据来设定最小支持度，通过python可视化作图得到下图。



运行程序之后得到对应的红球预测序列，对符合要求的频繁6项集进行输出，得到如下表1所示的结果。

表1 双色球2016122期Apriori方法预测结果表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 红球1 | 红球2 | 红球3 | 红球4 | 红球5 | 红球6 |
| 1 | 9 | 13 | 14 | 17 | 19 |
| 1 | 9 | 13 | 17 | 19 | 22 |
| 1 | 6 | 9 | 13 | 14 | 22 |
| 1 | 6 | 9 | 13 | 14 | 17 |
| 1 | 6 | 9 | 13 | 14 | 19 |
| 1 | 6 | 9 | 14 | 19 | 22 |
| 1 | 9 | 14 | 17 | 19 | 22 |
| 1 | 6 | 9 | 17 | 19 | 22 |
| 1 | 6 | 13 | 14 | 17 | 19 |
| 1 | 6 | 13 | 14 | 19 | 22 |
| 1 | 13 | 14 | 17 | 19 | 22 |
| 6 | 9 | 13 | 17 | 19 | 22 |
| 6 | 9 | 13 | 14 | 17 | 22 |
| 1 | 6 | 9 | 13 | 17 | 19 |
| 1 | 9 | 13 | 17 | 19 | 22 |
| 1 | 9 | 13 | 14 | 19 | 22 |
| 1 | 6 | 9 | 14 | 17 | 19 |
| 1 | 9 | 14 | 17 | 19 | 22 |
| 1 | 6 | 9 | 17 | 19 | 22 |
| 1 | 6 | 9 | 13 | 14 | 17 |
| 1 | 9 | 13 | 14 | 17 | 22 |
| 1 | 6 | 13 | 14 | 17 | 19 |
| 1 | 13 | 14 | 17 | 19 | 22 |
| 1 | 6 | 13 | 14 | 19 | 22 |
| 1 | 6 | 14 | 17 | 19 | 22 |
| 1 | 6 | 13 | 17 | 19 | 22 |
| 6 | 9 | 13 | 14 | 17 | 22 |
| 6 | 9 | 13 | 14 | 17 | 19 |
| 6 | 13 | 14 | 17 | 19 | 22 |
| 1 | 9 | 13 | 17 | 19 | 22 |
| 1 | 9 | 14 | 17 | 19 | 22 |
| 1 | 6 | 9 | 13 | 17 | 22 |
| 1 | 13 | 14 | 17 | 19 | 22 |
| 1 | 6 | 13 | 17 | 19 | 22 |
| 6 | 9 | 13 | 14 | 17 | 22 |
| 6 | 9 | 13 | 14 | 19 | 22 |
| 9 | 13 | 14 | 17 | 19 | 22 |

4.模型程序

程序位置：Apriori文件夹

环境：Python3.6 Jupyter notebook

可执行文件：Apriori.ipynb Apriori.html

结果输出文件：Output\_Apriori.txt

# 方法2 ：朴素贝叶斯

1.对于采用朴素贝叶斯算法的初步构想

本方法假设的是双色球的摇取不是随机的，而是符合球号从小到大的一定规律的，也就是说如果把双色球的号码从小到大排列，6个红球是一个中奖号码的特征（暂不考虑蓝球） 基本的想法是从朴素贝叶斯中得出的，朴素贝叶斯是基于特征条件独立假设学习输入/输出的联合概率分布，我们也假设双色球的抽取满足从小到大的号码之间的抽取互不干扰。 具体是用极大似然估计法估计后验概率，并对后验概率取对数防止乘数过小，加入了贝叶斯估计，其中取lamda = 1的拉普拉斯平滑，然后取最大化后验概率的解当作我们的预测。这里面并没有用到朴素贝叶斯的分类方法，只是借鉴了朴素贝叶斯的前提假设条件。

2.朴素贝叶斯介绍

朴素贝叶斯法是基于贝叶斯定理与特征条件独立假设的分类方法。对于给定的训练数据集，首先基于特征条件独立假设学习输入\输出的联合概率分布；然后基于此模型，对给定的输入x，利用贝叶斯定理求出后验概率最大的输出y。

朴素贝叶斯分类的正式定义如下：

1、设http://latex.codecogs.com/gif.latex?x=\%7ba_1,a_2,...,a_m\%7d为一个待分类项，而每个a为x的一个特征属性。

2、有类别集合http://latex.codecogs.com/gif.latex?C=\%7by_1,y_2,...,y_n\%7d。

3、计算http://latex.codecogs.com/gif.latex?P(y_1|x),P(y_2|x),...,P(y_n|x)。

4、如果http://latex.codecogs.com/gif.latex?P(y_k|x)=max\%7bP(y_1|x),P(y_2|x),...,P(y_n|x)\%7d，则http://latex.codecogs.com/gif.latex?x%20\in%20y_k。

那么现在的关键就是如何计算第3步中的各个条件概率。我们可以这么做：

1、找到一个已知分类的待分类项集合，这个集合叫做训练样本集。

2、统计得到在各类别下各个特征属性的条件概率估计。即http://latex.codecogs.com/gif.latex?P(a_1|y_1),P(a_2|y_1),...,P(a_m|y_1);P(a_1|y_2),P(a_2|y_2),...,P(a_m|y_2);...;P(a_1|y_n),P(a_2|y_n),...,P(a_m|y_n)。

3、如果各个特征属性是条件独立的，则根据贝叶斯定理有如下推导：

http://latex.codecogs.com/gif.latex?P(y_i|x)=\frac%7bP(x|y_i)P(y_i)%7d%7bP(x)%7d

因为分母对于所有类别为常数，因为我们只要将分子最大化皆可。又因为各特征属性是条件独立的，所以有： http://latex.codecogs.com/gif.latex?P(x|y_i)P(y_i)=P(a_1|y_i)P(a_2|y_i)...P(a_m|y_i)P(y_i)=P(y_i)\prod%5em_%7bj=1%7dP(a_j|y_i)

3.基本步骤

①导入数据，只保存6个红球的信息

②利用极大似然估计估计后验概率，使用拉普拉斯平滑，处理值为0的数据

③对后验概率取log，防止乘数过小

④基于特征条件独立假设，枚举红球的状态，求出概率最大的红球组合作为最后的预测解

4.输出结果

运行程序之后得到一组概率最大化的红球预测序列，如下表2所示。

表2 双色球2016122期Naive Bayes方法预测结果表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 红球1 | 红球2 | 红球3 | 红球4 | 红球5 | 红球6 |
| 1 | 6 | 14 | 17 | 26 | 32 |

5.模型程序

程序位置：NaiveBayes文件夹

环境：Python2.7/3.6 Jupyter notebook

可执行文件：NaiveBayes.ipynb Naivehtml

结果输出文件：result.txt

# 方法3: 基于马尔可夫链的状态转移概率预测方法

1.随机过程与马尔可夫链介绍

从19世纪开始，概率论的研究已经从单一的随机变量延伸到对随机变量的时间序列，即**随机过程**的研究。设是一无限整数集，对于任意的，均为随机变量，所有可能取值的集合称为随机过程的**状态空间**。

随即过程的研究要比随机变量复杂很多，因为对于任意一个时刻的状态，它的取值可能与其他各个状态都有关。俄罗斯著名的数学家马尔可夫为了简化问题，引入了马尔可夫性(或称为无后效性)，即当前的状态仅与它前面的一个状态有关，而与前面的状态无关，即有：

满足上式假设的随机过程称为**马尔可夫过程**，当时间和状态空间都离散时，也把马尔可夫过程称为**马尔可夫链(Markov Chain)**，简称马氏链。

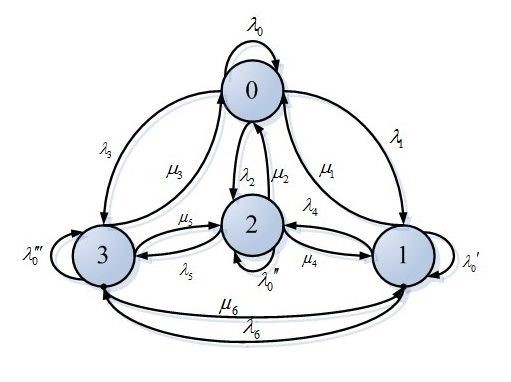
从时刻的状态之间的转移，用一个矩阵来表示，也被成为**状态转移矩阵**，满足：

设随机过程的状态空间大小为，那么显然转移矩阵是一个大小为的矩阵。转移矩阵满足下面的两个性质。

·

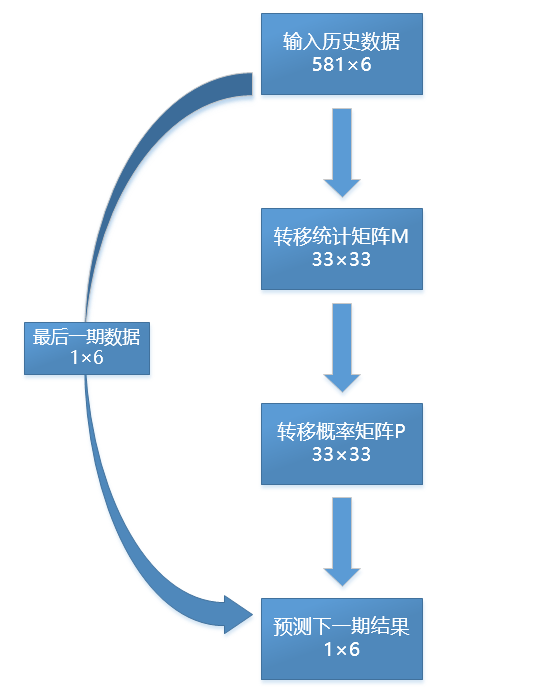
·，即矩阵中每一行转移概率之和等于1

假设状态空间有个可能的取值，给定初始概率分布，其中表示在初始时刻状态取值为的概率，那么当给定初始概率分布和状态转移矩阵，可以得到下一时刻的状态取值分布，依次类推，可以得到第时刻的概率分布计算公式为：



2.问题解决思路

针对本次数据挖掘作业，经过基本的数据预处理，我们的这一模型主要利用了马尔科夫链的方法，核心是利用了马尔科夫概率转移矩阵。主要的问题解决流程如下图所示。



3.具体解决方法

1. 根据所给出的581期福彩双色球数据(我们只关注6个红球的数值)，我们可以对580次转移结果进行统计。统计的对象是红球的33个号码，每一个数字跳转到所有33个号码的总次数。统计的结果是一个的整数矩阵。其中表示：从当前期红球号码为，跳转到下一期红球号码为的总次数。
2. 根据得到的**转移统计矩阵**，我们可以将其转变为**马尔科夫转移概率矩阵**，其中表示：从当前期红球号码，跳转到下一期红球号码的概率。这里具有的性质，即当前期红球号码跳转到下一期红球中所有33个号码的概率之和为1。此时的矩阵是包含了所有历史数据状态转移的统计信息，在统计意义上具有较为可靠的预测性、合理性。
3. 最后的预测步骤，只需要将根据最后一期每个红球的号码，可以从马尔科夫转移概率矩阵中找到下一期跳转概率最大的红球号码。如果跳转到的号码已经在下一期的结果当中，则选取跳转概率次大的红球号码，依次类推。由此便可得到一组(6个)跳转概率最大化的红球号码。

4.输出结果

运行程序之后得到一组概率最大化的红球预测序列，如下表3所示。

表3 双色球2016122期Markov Chain方法预测结果表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 红球1 | 红球2 | 红球3 | 红球4 | 红球5 | 红球6 |
| 1 | 2 | 10 | 14 | 19 | 22 |

5.模型程序

程序位置：MarkovChain文件夹

环境：Python3.6 Jupyter notebook

可执行文件：MarkovChain.ipynb MarkovChain.html

结果输出文件：Output\_MarkovChain.txt

# 方法4: 基于先验概率的轮盘赌模拟预测方法

1.模型使用理由

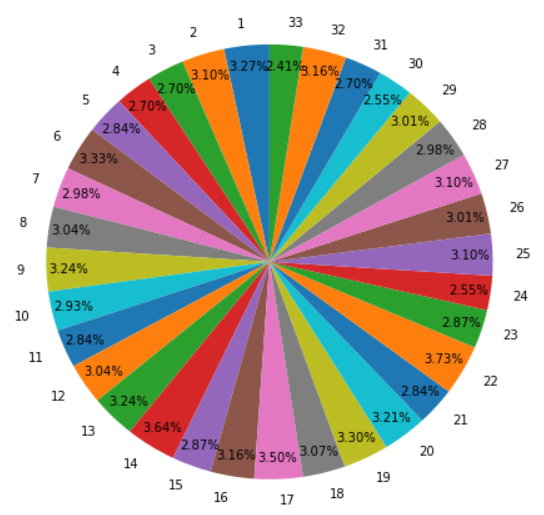
通过理性分析，我们可以认为对于公平的彩票开奖操作，每一期的开奖都应当是一次独立事件，开奖的号码完全是一组随机序列。从科学的角度来看，从历史开奖数据进行任何时间序列上的分析预测其实都是无济于事的，所以根据**奥卡姆剃刀原则(Occam’s Razor)**，预测准确性基本相同的情况下，应当选择最为简单的模型。

2.模型原理

这里我们提出一种直接基于先验概率进行预测的简单方法。该方法基于历史统计信息，得出33个红球号码的每个号码出现的概率，然后计算其累计概率，然后通过在范围内产生一个随机数，通过该随机数落在的区间来确定新生成的红球号码。重复该过程6次，当新生成的号码已经是序列当中时，重新产生一个随机数，以保证生成的预测的结果不重复。

上面的累计概率可以形成一个转盘，每一种号码都对应转盘上的一个扇形块，其面积越大即代表该号码被选中的概率越大。然后，就可以利用轮盘赌的随机选择方式模拟预测，选取6个数字作为新的预测结果。

根据本题，我们可以利用python的可视化工具绘制对应的大转盘(每个红球号码的先验概率饼图)，如下图所示。



4.输出结果

运行程序之后得到20组概率最大化的红球预测序列，如下表4所示。

表4 双色球2016122期Random Predict方法预测结果表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 红球1 | 红球2 | 红球3 | 红球4 | 红球5 | 红球6 |
| 2 | 4 | 5 | 15 | 26 | 31 |
| 5 | 10 | 11 | 16 | 17 | 20 |
| 8 | 16 | 18 | 28 | 31 | 32 |
| 14 | 19 | 23 | 24 | 27 | 31 |
| 6 | 11 | 13 | 27 | 28 | 31 |
| 3 | 12 | 13 | 19 | 20 | 21 |
| 10 | 15 | 16 | 19 | 24 | 28 |
| 6 | 8 | 18 | 21 | 23 | 29 |
| 2 | 3 | 15 | 25 | 27 | 28 |
| 3 | 4 | 7 | 10 | 19 | 31 |
| 1 | 12 | 16 | 17 | 24 | 30 |
| 1 | 6 | 10 | 20 | 28 | 33 |
| 17 | 18 | 20 | 21 | 25 | 33 |
| 2 | 5 | 10 | 16 | 18 | 21 |
| 2 | 5 | 9 | 14 | 17 | 18 |
| 2 | 3 | 16 | 19 | 29 | 33 |
| 3 | 9 | 16 | 22 | 28 | 31 |
| 2 | 5 | 18 | 19 | 20 | 31 |
| 7 | 14 | 15 | 17 | 19 | 31 |
| 3 | 15 | 16 | 21 | 32 | 33 |

4.模型程序

程序位置：RandomPredict文件夹

环境：Python3.6 Jupyter notebook

可执行文件：RandomPreidict.ipynb RandomPredict.html

结果输出文件：Output\_RandomPredict.txt