

---

## 《数据挖掘》实验报告（二）

专    业    \_\_\_\_\_信息管理与信息系统\_\_\_\_\_

年    级    \_\_\_\_\_2016\_\_\_\_\_

学    号    \_\_\_\_\_2016214288\_\_\_\_\_

学生姓名    \_\_\_\_\_曾德勤\_\_\_\_\_

指导老师    \_\_\_\_\_刘向\_\_\_\_\_

华中师范大学信息管理学院

# I 实验目的和意义

数据挖掘设计实验的目的是培养学生具有初步的 python 程序设计、编程、调试的能力。通过实验使学生进一步熟悉并掌握 python 程序的调试运行环境、程序设计过程、程序的基本结构以及程序设计的基本方法。通过实验，使学生将程序设计的理论知识与实践相结合，为学生学习其他计算机编程语言打下基础。

## II 实验内容

### 实验二 关联规则

#### 【实验意义】

关联分析是在一种在大规模数据集中寻找有趣关系的任务。这些关系可以有两种形式：频繁项目或者关联规则。频繁项集是经常出现在一起的物品的集合，关联规则暗示两种物品之间可能存在很强的关系。

#### 【实验要求及步骤】

发现毒蘑菇的相似特征，利用这些特征就能避免吃到那些有毒的蘑菇。

(1) 收集数据：提供文本文件 **【mushroom . dat】**

(2) 准备数据：解析 tab 键分隔的数据行。

(3) 训练算法：使用本章的 `apriori()` 来发现与毒蘑菇有关的特征。

**【数据说明】** mushroom . dat 的每一行有 23 个数，对应蘑菇的 23 种特征，第一个特征表示有毒或者可食用。如果样本有毒，则值为 2。如果可食用，则值为 1。下一个特征是蘑菇伞的形状，有 6 种可能的值，分别用 3—8 来表示。为了找出毒蘑菇中存在的公共特征，可以运用 Apriori 算法来寻找包含特征值为 2，最小支持度为 0.4 的频繁三项集。

#### 【代码参考】

```
from numpy import *

def loadDataSet(fileName):      #general function to parse tab -delimited floats
    dataMat = []                #assume last column is target value
    fr = open(fileName)
    for line in fr.readlines():
        curLine = line.split()
        dataMat.append(curLine)
    return dataMat

def createC1(dataSet):
```

```

C1 = []
for transaction in dataSet:
    for item in transaction:
        if not [item] in C1:
            C1.append([item])
C1.sort()
return map(frozenset, C1)#use frozen set so we
                        #can use it as a key in a dict
def scanD(D, Ck, minSupport):
    ssCnt = {}
    numItems = 1.0
    for tid in D:
        for can in Ck:
            if can.issubset(tid):
                if not can in ssCnt: ssCnt[can]=1
                else: ssCnt[can] += 1
        numItems += 1
    retList = []
    supportData = {}
    for key in ssCnt:
        support = ssCnt[key]/numItems
        if support >= minSupport:
            retList.insert(0, key)
            supportData[key] = support
    return retList, supportData

def aprioriGen(Lk, k): #creates Ck
    retList = []
    lenLk = len(Lk)
    for i in range(lenLk):
        for j in range(i+1, lenLk):
            L1 = list(Lk[i])[:k-2]; L2 = list(Lk[j])[:k-2]
            L1.sort(); L2.sort()
            if L1==L2: #if first k-2 elements are equal
                retList.append(Lk[i] | Lk[j]) #set union
    return retList

def apriori(dataSet, minSupport):
    C1 = createC1(dataSet)
    D = map(set, dataSet)
    L1, supportData = scanD(D, C1, minSupport)
    L = [L1]
    k = 2
    while (len(L[k-2]) > 0):
        Ck = aprioriGen(L[k-2], k)
        Lk, supK = scanD(D, Ck, minSupport)#scan DB to get Lk

```

```

    supportData.update(supK)
    L.append(Lk)
    k += 1
return L, supportData

```

## 【实验报告】

**实习时间:**

实习地点:

实习机号:

(1) 收集数据：提供文本文件【mushroom.dat】

(2) 准备数据：解析 tab 键分隔的数据行。

### 具体实验内容

```
coding=UTF-8
import apriori1
# 定义最小支持度
minSupport = 0.4
# 收集数据集
dataMap = apriori1.loadDataSet('mushroom.dat')
```

数据打印结果:

[illegible]

...

(3) 训练算法：使用本章的 `apriori()` 来发现与毒蘑菇有关的特征。

```

2 # 找出数据集中支持度>=0.4的所有的频繁项集
3 L, supportData = apriori1.apriori(dataMap, minSupport)
4
5 # 从所有的频繁项集中找出频繁3-项集L3
6 L3 = L[2]
7
8 # 遍历频繁3-项集L3, 筛选出包含特征值'2'的频繁3-项集放入结果集中
9 resultSet = []
10 for freqSet3 in L3:
11     if '2' in freqSet3:
12         resultSet.append(freqSet3)
13
14 print resultSet
15

```

打印结果:

```

[frozenset(['39', '2', '86']), frozenset(['39', '2', '85']), frozenset(['39', '2', '34']), frozenset(['90', '39', '2']), frozenset(['2', '59', '34']), frozenset(['2', '86', '85']), frozenset(['90', '2', '85']), frozenset(['39', '2', '59']), frozenset(['90', '2', '86']), frozenset(['2', '28', '85']), frozenset(['2', '86', '34']), frozenset(['59', '2', '86']), frozenset(['34', '2', '85']), frozenset(['2', '53', '85']), frozenset(['90', '2', '59']), frozenset(['90', '2', '34']), frozenset(['2', '59', '85'])]
python task.py 1.82s user 0.11s system 97% cpu 1.973 total

```

以上的三项集就是包括 '2' (样本有毒特征) 的频繁三项集。

通过本次实验, 我了解到了频繁项集产生的过程, 并利用收集到的数据集通过 Apriori 算法找出了包含毒蘑菇特征的频繁三项集。

利用 Apriori 算法, 可以找出所有的频繁-k 项集, 然后我们可以根据这些频繁项集找到我们想要的公共关系。

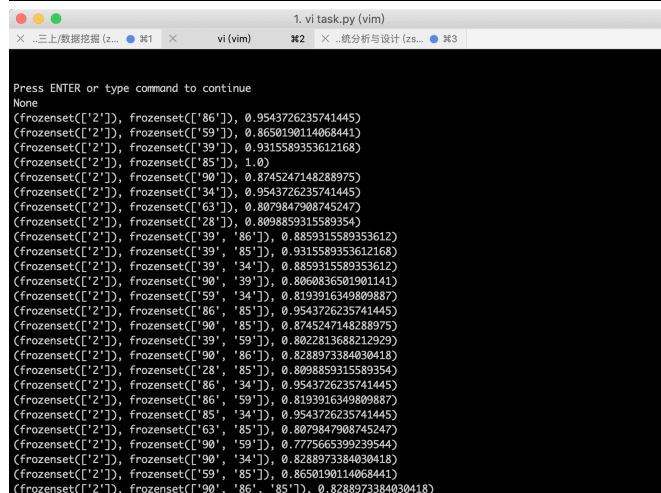
然而仅通过 Apriori 算法找出的频繁项集只是从支持度的角度找出的规则, 要使毒蘑菇的公共特征更具有可靠性, 还需要进行基于支持度的减支产生关联规则:

令最小置信度为 0.7:

```

26 # 置信度规则列表
27 bigRuleList = apriori1.generateRules(L, supportData)
28
29 # 遍历置信度规则列表, 找出毒蘑菇的强关联规则的特征集
30 for item in bigRuleList:
31     if (item[0] == frozenset(['2'])):
32         print item
33

```



```

1. vi task.py (vim)
Press ENTER or type command to continue
None
(frozenset(['2']), frozenset(['86']), 0.9543726235741445)
(frozenset(['2']), frozenset(['59']), 0.8650190114068441)
(frozenset(['2']), frozenset(['39']), 0.9315589353612168)
(frozenset(['2']), frozenset(['85']), 1.0)
(frozenset(['2']), frozenset(['90']), 0.8745247148288975)
(frozenset(['2']), frozenset(['34']), 0.9543726235741445)
(frozenset(['2']), frozenset(['63']), 0.8079847908745247)
(frozenset(['2']), frozenset(['28']), 0.8098859315589354)
(frozenset(['2']), frozenset(['39', '86']), 0.8859315589353612)
(frozenset(['2']), frozenset(['39', '85']), 0.9315589353612168)
(frozenset(['2']), frozenset(['39', '34']), 0.8859315589353612)
(frozenset(['2']), frozenset(['90', '39']), 0.8060836591901141)
(frozenset(['2']), frozenset(['59', '34']), 0.8193916349809887)
(frozenset(['2']), frozenset(['86', '85']), 0.9543726235741445)
(frozenset(['2']), frozenset(['90', '85']), 0.8745247148288975)
(frozenset(['2']), frozenset(['39', '59']), 0.8022813688212929)
(frozenset(['2']), frozenset(['90', '86']), 0.8288973384030418)
(frozenset(['2']), frozenset(['28', '85']), 0.8098859315589354)
(frozenset(['2']), frozenset(['86', '34']), 0.9543726235741445)
(frozenset(['2']), frozenset(['86', '59']), 0.8193916349809887)
(frozenset(['2']), frozenset(['85', '34']), 0.9543726235741445)
(frozenset(['2']), frozenset(['63', '85']), 0.8079847908745247)
(frozenset(['2']), frozenset(['90', '59']), 0.7775665399239544)
(frozenset(['2']), frozenset(['90', '34']), 0.8288973384030418)
(frozenset(['2']), frozenset(['59', '85']), 0.8650190114068441)
(frozenset(['2']), frozenset(['90', '86', '85']), 0.8288973384030418)

```