**数据挖掘**

**--作业三实验报告**

学 院：自动化学院

专 业：系统工程

年 级：2017 级

姓 名：胡笛

学 号：2620170029

提交日期：2018/4/28

1. 实验背景

 数据挖掘是从大量的、不完全的、有噪声的、模糊的、随机的数据中提取隐含在其中的、人们事先不知道的、但又潜在于有用的信息和知识的获取过程。随着信息技术的高速发展，积累的数据量急剧增长，动辄以TB计，如何从海量的数据中提取有用的知识成为时代当务之急问题。数据挖掘就是顺应这种需要去应运而生发展起来的数据处理技术是知识发现的关键步骤。

数据挖掘和知识发现是一个涉及多学科的研究领域。近年来，与数据库的知识发现研究领域已经成为热点，其中关联规则数据挖掘算法是数据挖掘中的一个很重要的课题，它是从背后发现数据中的关联或联系。

分类是找出一个类别的概念描述，代表了数据的整体信息，分类的内涵描述，并用描述来构造模型，一般用作于规则或决策树模式表示出来。分类是利用训练数据集中通过一定的算法而求得分类规则。分类可被用于规则描述和数据预测。

聚类是把数据按照相似性归纳成若干类别分类出来，同一类中的数据彼此相似，不同类中的数据则相异。聚类分析可以建立宏观的概念，发布数据的分布模式，以及可能性的数据属性之间的相互关系。

2.实验目的

掌握数据挖掘的基础知识，能够深刻理解并熟练运用决策树算法和最近邻算法对数据进行分类，能够熟练Matlab软件进行算法的实现。

3.操作环境

系统环境：Windows 10

软件环境：matlab2017a

软件简介：MATLAB是美国[MathWorks](https://baike.baidu.com/item/MathWorks)公司出品的商业[数学软件](https://baike.baidu.com/item/%E6%95%B0%E5%AD%A6%E8%BD%AF%E4%BB%B6)，用于算法开发、数据可视化、数据分析以及[数值计算](https://baike.baidu.com/item/%E6%95%B0%E5%80%BC%E8%AE%A1%E7%AE%97)的高级技术计算语言和交互式环境，MATLAB可以进行[矩阵](https://baike.baidu.com/item/%E7%9F%A9%E9%98%B5)运算、绘制函数和数据、实现算法、创建用户界面、连接其他编程语言的程序等，主要应用于工程计算、控制设计、[信号处理](https://baike.baidu.com/item/%E4%BF%A1%E5%8F%B7%E5%A4%84%E7%90%86)与通讯、[图像处理](https://baike.baidu.com/item/%E5%9B%BE%E5%83%8F%E5%A4%84%E7%90%86)、[信号检测](https://baike.baidu.com/item/%E4%BF%A1%E5%8F%B7%E6%A3%80%E6%B5%8B)、[金融建模](https://baike.baidu.com/item/%E9%87%91%E8%9E%8D%E5%BB%BA%E6%A8%A1)设计与分析等领域。

4.实验数据

本实验选用的的是实验一中的数据集1，对其数据的标称属性进行关联规则挖掘。

5. 分类分析

5.1决策树算法简介

决策树算法是一种逼近离散函数值的方法。它是一种典型的分类方法，首先对数据进行处理，利用归纳算法生成可读的规则和决策树，然后使用决策对新数据进行分析。本质上决策树是通过一系列规则对数据进行分类的过程。其主要优点是模型具有可读性，分类速度快。学习时，利用训练数据，根据损失函数最小化的原则建立决策树模型。预测时，对新的数据，利用决策树模型进行分类。

5.2利用决策树算法进行分类

利用决策树算法对数据进行分类，最终结果以混淆矩阵和分布图的形式表现，决策树算法的代码如下：

%% 决策树

% 训练分类器

t = ClassificationTree.fit(Xtrain,Ytrain,'CategoricalPredictors',catPred);

% 进行预测

Y\_t = t.predict(Xtest);

% 计算混淆矩阵

disp('决策树方法分类结果：')

C\_t = confusionmat(Ytest,Y\_t)

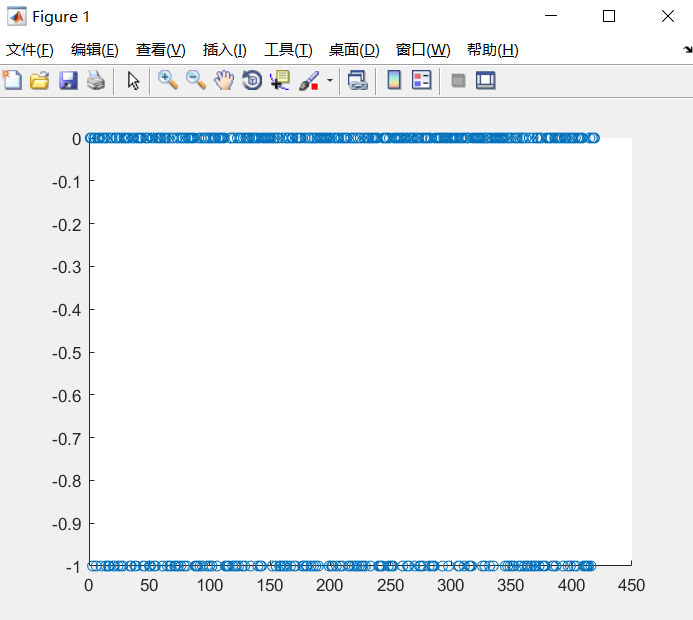
error=Y\_t-Ytest;

scatter(1:419,error)

执行代码，得到混淆矩阵如下图所示，可以看出，在一共419个样本中，最终结果共有267个应该属于0，152个应该属于1，但预测结果将绝大多数的1分类数据都划分在了0分类，故准确率很低。



由于代码是将结果归纳为0和1，将预测结果与真实结果进行相减，结果是0则预测正确，结果是-1则预测错误，最终得到分布图如下所示：



5.3最近邻算法简介

KNN的工作原理是：存在一个样本数据集合，也称为训练样本集，而且样本集中每个数据都存在标签，也就是我们知道样本集中每一个数据与所属分类对应关系。输入没有标签的新数据后，将新数据的每个特征与样本集中数据对应的特征进行比较，然后算法提取样本集中特征最相似数据（最近邻）的分类标签。

5.4利用最近邻算法进行分类

利用最近邻算法对数据进行分类，最终结果以混淆矩阵和分布图的形式表现，最近邻算法的代码如下：

%% 最近邻

% 训练分类器

knn = ClassificationKNN.fit(Xtrain,Ytrain,'Distance','seuclidean',...

'NumNeighbors',5);

% 进行预测

[Y\_knn, Yscore\_knn] = knn.predict(Xtest);

Yscore\_knn = Yscore\_knn(:,2);

% 计算混淆矩阵

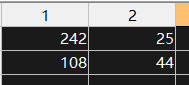
disp('最近邻方法分类结果：')

C\_knn = confusionmat(Ytest,Y\_knn)

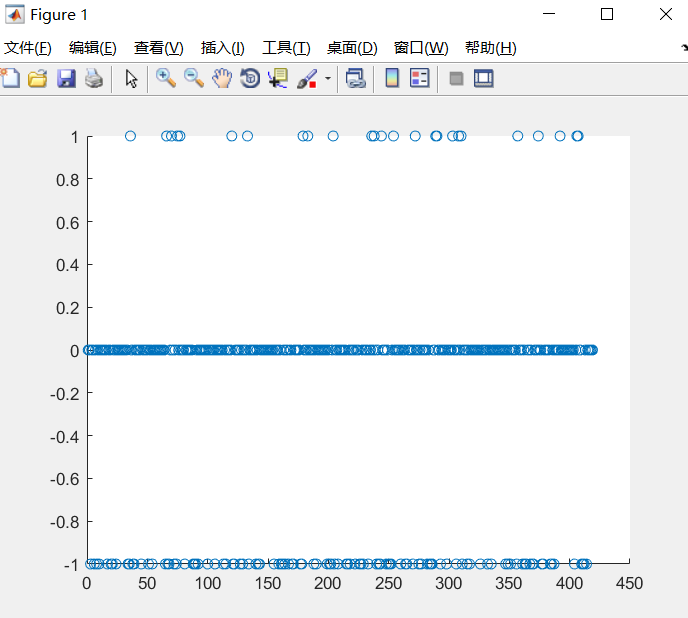
error=Y\_knn-Ytest;

scatter(1:419,error)

执行代码，得到混淆矩阵如下图所示，可以看出，在一共419个样本中，最终结果共有267个应该属于0，152个应该属于1，算法成功将242个0分类数据，44个1分类数据预测正确，另把108个1分类数据预测为0分类，25条0分类数据预测为1，正确率较高。



其分布图如下图所示，大多数的情况预测值和真实值相同。



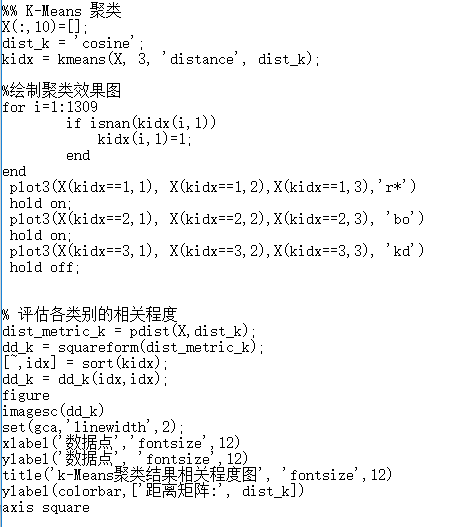
6. 聚类分析

6.1 K-MEANS算法简介

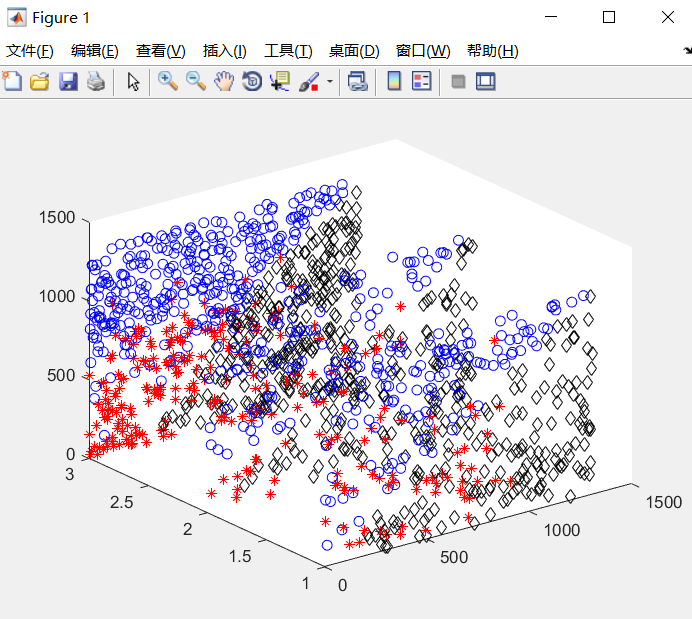
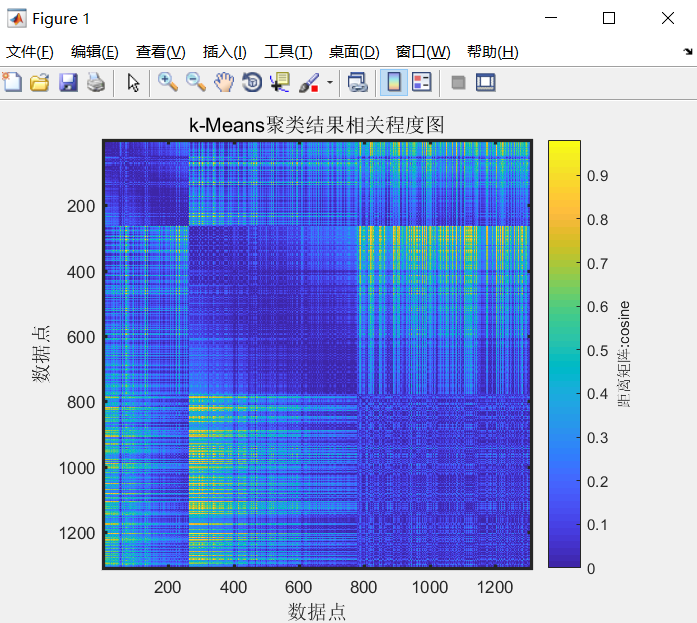
K-MEANS算法是输入聚类个数k，以及包含 n个数据对象的数据库，输出满足方差最小标准k个聚类的一种算法。k-means 算法接受输入量 k ；然后将n个数据对象划分为 k个聚类以便使得所获得的聚类满足：同一聚类中的对象相似度较高；而不同聚类中的对象相似度较小。

6.2 K-MEANS算法结果

算法代码如下所示



最终可以得到聚类效果图（左）和评价图（右）。



6.3 层次聚类算法简介

层次聚类方法的基本思想是：通过某种[相似性](https://baike.baidu.com/item/%E7%9B%B8%E4%BC%BC%E6%80%A7)测度计算节点之间的相似性，并按相似度由高到低排序，逐步重新连接个节点。该方法的优点是可随时停止划分，主要步骤如下：

（1）移除网络中的所有边，得到有n个孤立节点的初始状态；

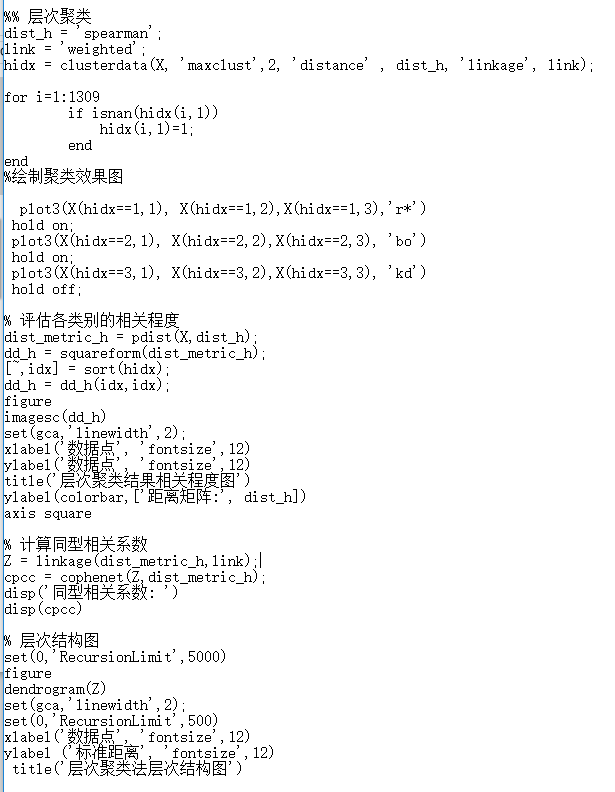
（2)计算网络中每对节点的相似度；

（3）根据相似度从强到弱连接相应节点对，形成[树状图](https://baike.baidu.com/item/%E6%A0%91%E7%8A%B6%E5%9B%BE)；

(4）根据实际需求横切树状图，获得[社区结构](https://baike.baidu.com/item/%E7%A4%BE%E5%8C%BA%E7%BB%93%E6%9E%84)

6.4层次聚类算法结果

算法代码如下所示



执行代码得到聚类图（上），评价图（中）和结构图（下）。

