## **广州大学学生实验报告**

**开课学院及实验室：**计算机科学与工程实验室 **2021年12月2日**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **学院** | **计算机科学与网络工程学院** | **年级/专业/班** | **网络193** | **姓名** | 吴伟俊 | **学号** | 1806200107 |
| **实验课程名称** | **人工智能导论实验** | | | | | **成绩** |  |
| **实验项目名称** | **聚类算法** | | | | | **指导老师** | 张少宏 |

(\*\*\*报告只能为文字和图片,老师评语将添加到此处,学生请勿作答\*\*\*)

**一、实验内容**

**问题描述**：聚类分析是研究（样品或指标）分类问题的一种统计分析方法，同时也是人工智能中的一个重要算法。聚类（Cluster）分析是由若干模式（Pattern）组成的，通常，模式是一个度量（Measurement）的向量，或者是多维空间中的一个点。聚类分析以相似性为基础，在一个聚类中的模式之间比不在同一聚类中的模式之间具有更多的相似性。

**内容提要**：针对教师指定的两类公用数据集（纯数值型例如UCI Iris, 混杂型数据例如UCI Bank Marketing），学生至少实现两种聚类算法，并比较分析结果原因。本次实验主要内容包括数据处理、算法实现和评价方法。鼓励与其他方法尤其是业界领先算法进行比较分析，鼓励创新设计求解方法。

**二、实验设备**

1. 实验设备：计算机；

2. 平台：Windows操作系统，vscode

**三、实验步骤**

* 读取数据，并做预处理。
* 至少实现两种聚类算法，选择评价方法比较结果并分析原因
* 选择适当可视化方法显示结果。
* \*扩展选做题：分析考虑数据的特性并和具体聚类方法的匹配。

**四、分析说明（包括核心代码及解释）**

# 1. K-means算法

import matplotlib.pyplot as plt

from random import sample

import numpy as np

from sklearn.cluster import KMeans

from sklearn.datasets import load\_iris

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

    #导入数据集

    iris = load\_iris()

    X = iris.data[:,0:2] #取特征

    n = 3 #表示n个聚类

    u = sample(X.tolist(),n) #随机选择n个X中的向量作为聚类中心

    max\_iter = 0 #记录迭代次数

    u\_before = u

    #簇分配过程

    while max\_iter < 5: # 迭代次数

        c = []

        for i in range(len(X)):

            min = 1000000000000

            index = 0

            for j in range(n):

                dist = np.sqrt(np.sum(np.square(X[i] - u[j])))

                if dist < min:

                    min = dist

                    index = j

            c.append(index)

        for j in range(n):

            sum = np.zeros(2)

            cnt = 0

            for i in range(len(X)):

                if c[i] == j:

                    sum = sum+X[i]

                    cnt = cnt+1

            u[j] = sum / cnt

        max\_iter += 1

    label\_pred = np.array(c)

    #可视化k-means结果

    col = ["green", "blue", "red"]

    for i in range(3):

        res = X[label\_pred == i]

        plt.scatter(res[:, 0], res[:, 1], c = col[i], marker = 'x')

    plt.xlabel('x')

    plt.ylabel('y')

    plt.legend(loc=2)

    plt.show()

实验结果

图表, 散点图

描述已自动生成

2. **Mean Shift 聚类算法**

from sklearn.cluster import MeanShift

import matplotlib.pyplot as plt

from sklearn.manifold import TSNE

from matplotlib.pyplot import style

import numpy as np

def load\_data(path):

    f = open(path)  # 打开文件

    tmp1 = []

    tmp = []

    for line in f.readlines():

        lines = line.strip().split("\t")

        data\_tmp = []

        s = line.strip().split(',')

        tmp.append([float(s[0]), float(s[1]), float(s[2]), float(s[3])])

    tmp1.append(tmp)

    data = np.concatenate((tmp1))

    f.close()

    return data

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

    data = load\_data("E:\workworkwork\GitHub\WinnieLearning\python\聚类算法\iris.data")

    print(data)

    #搭建Mean-Shift聚类器

    clf=MeanShift()

    #对样本数据进行聚类

    predicted=clf.fit\_predict(data)

    colors = [['black','green','blue','yellow'][i] for i in predicted]

    #绘制聚类图

    plt.scatter(data[:,0],data[:,1],c=colors,marker='x')

    plt.title('Mean Shift')

    plt.show()

实验结果

图表, 散点图

描述已自动生成

**五、总结心得**

k-means算法：执行时间一般为0.4s左右,聚类效率还是挺不错的.原理简单，容易实现，可解释度强。缺点是K值很难确定，聚类的效果依赖于聚类中心的初始化。mean shift算法，不需要设置簇类的个数，算法结果稳定，不需要进行类似K均值的样本初始化。缺点是聚类的结果取决于带宽的设置，带宽设置的大小，收敛太慢等因素都可能导致簇类丢失。