第一章　聚类-K-Means算法

1.1　实验内容与任务

现在有一批鸢尾花的数据，共包含150个样本，每个样本有四个属性，Sepal Length （花萼长度），Sepal Width（花萼宽度），Petal Length （花瓣长度），Petal （花瓣宽度） Width。同时，每个样本所属类别也已经标出，一共有3个类别：Iris Setosa（山鸢尾）、Iris Versicolour（杂色鸢尾），以及Iris Virginica（维吉尼亚鸢尾）。

要求学生根据样本的属性数据将鸢尾花用K-Means 算法进行聚类，获得3个类别，并将每个样本分到一个类别中。然后将聚类所得的样本类别分布情况与原始数据中的样本类别分布情况进行对比，分析K-Means算法的性能。

1.2　实验过程及要求

1. 实验环境要求：Windows/Linux操作系统，Python编译环境，numpy，matplotlib等程序库。
2. 加载鸢尾花数据集，观察数据集特征。
3. 实现K-Means算法, 运行并观察聚类结果
4. 研究初始聚类中心的设置对K-Means算法收敛性的影响
5. 研究参数K对聚类结果的影响

1.3　教学目标

1. 能理解聚类算法的理论基础。
2. 能够应用K-Means算法完成聚类。
3. 能够分析聚类算法的优缺点。
4. 提高对复杂工程问题建模和分析的能力。

1.4　相关知识及背景

聚类是人类挖掘知识的重要手段，例如对自然界的生物进行类别和群体的划分。在商业活动中，对客户群体进行划分，能对客户特点进行分析并对不同群体进行针对性营销。

在机器学习中，聚类属于无监督学习，直接在数据中挖掘类别关系。聚类跟有监督学习中的分类的区别是缺乏有标记的训练数据。聚类有两个任务，首先要确定将数据划分多少类，其次要将每个样本分到一个类别中。例如图1.1中的数据点，我们仅根据数据点的分布情况，可以考虑将其划分4个类，坐标比较相似的点处于同一个类中。完成聚类要基于两个原则：

1. 不同类别的样本之间相似性很小。
2. 同一类别的样本之间相似性很大。

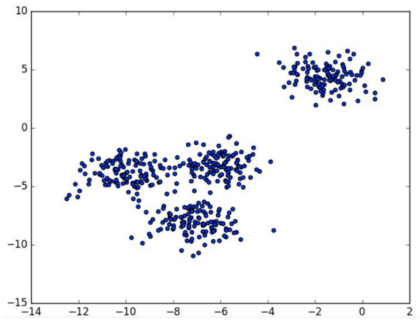

图 1.1: 无标签数据

1.5　实验教学与指导

1.5.1　数据加载

```
from sklearn import datasets
import matplotlib.pyplot as plt
import numpy as np

# 获取数据集并进行探索
iris = datasets.load_iris()
irisFeatures = iris["data"]
irisFeaturesName = iris["feature_names"]
irisLabels = iris["target"]
```

1.5.2　K-Means算法原理

K-Means算法常用于对欧氏空间的样本点进行聚类。两个样本点x_i和x_j的相似度可用距离$\|x_i - x_j\|$来定义。假设聚类一共有K个类别，每个类别C_k定义一个聚类中心点u_k(注意，聚类中心点并不是样本点)，K-Means 算法规定，一个样本点根据其离每个聚类中心点的距离，划分到最近的类别中去。因此完成聚类的两个任务，只要能确定K个中心点即可。为了求出最好的中心点，定义一个损失函数，对聚类效果符合前述聚类原则的程度进行评价：

$$J(u_1, u_2, \ldots, u_K) = \sum_{k=1}^{K} \sum_{i \in C_k} \|x_i - u_k\|^2 \tag{1.1}$$

从J的定义上看，各样本划分到距离最近聚类中心，能够达到更小的J值。另外，由于J是局部光滑的，可以通过对解求$\nabla J = 0$，计算出J最小时的聚类中心位置，

$$u_k^* = \frac{1}{|C_k|} \sum_{i \in C_k} x_i \tag{1.2}$$

可见最好的聚类中心u_k^*实际上是该类别的样本均值点。而将u_k移动到u_k^*后，如果所有样本点的划分不发生变化，则J到达一个局部最优值，K-Means算法达到稳定。否则，可以继续求解在新的划分情况下的最优的中心点。

1.5.3　K-Means算法实现

可以设想，如果某个聚类中心k远离所有的样本点，则样本点都会划分到其他类别中。这个聚类中心k将会获得一个空的聚类，影响了聚类效果。因此初始聚类点落到样本点中间较好。初始化时，限制聚类中心的坐标可以达到这个目的。K-Means算法实现如下：

```
def norm2(x):
    #求2范数的平方值
    return np.sum(x*x)

class KMeans(object):
    def __init__(self, k: int, n: int):
    #k: 聚类的数目，n: 数据维度
        self.K = k
        self.N = n
        self.u = np.zeros((k,n))
        self.C=[[]for i in range(k)]
        #u[i]：第i个聚类中心，C[i]：第i个类别所包含的点

    def fit(self, data: np.ndarray):
    #data: 每一行是一个样本
        self.select_u0(data)
        #聚类中心初始化
        J=0
        oldJ=100
        while abs(J-oldJ) >0.001:
            oldJ=J
            J=0
            self.C=[[]for i in range(self.K)]
            for x in data:
                nor=[ norm2(self.u[i]-x) \
                      for i in range(self.K)]
                J += np.min(nor)
                self.C[ np.argmin(nor) ].append(x)
            self.u=[np.mean(np.array(self.C[i]),axis=0) \
                    for i in range(self.K)]

    def select_u0(self, data: np.ndarray):
```

```
        for j in range(self.N):
            # 得到该列数据的最小值,最大值
            minJ = np.min(data[:, j])
            maxJ = np.max(data[:, j])
            rangeJ = float(maxJ - minJ)
            #聚类中心的第j维数据值随机为位于(最小值，最大值)内
            self.u[:,j]=minJ +rangeJ * np.random.rand(self.K)
```

1.5.4　训练并显示聚类结果

设置$K = 3$, 运行K-Means算法聚类后，用样本特征数据的前两个维度显示聚类效果。

```
model = KMeans(3,4)
#k=3，n=4
model.fit(irisFeatures)

x=np.array(model.C[0])
plt.scatter(x[:,0], x[:,1], c = "red", marker='o', label='
    cluster1')
x=np.array(model.C[1])
plt.scatter(x[:,0], x[:,1], c = "green", marker='*', label='
    cluster2')
x=np.array(model.C[2])
plt.scatter(x[:,0], x[:,1], c = "blue", marker='+', label='
    cluster3')
u=np.array(model.u)
plt.scatter(u[:,0], u[:,1], c = "black", marker='X', label='
    center')
plt.xlabel('petal length')
plt.ylabel('petal width')
plt.legend(loc=2)
plt.show()
```

1.6　实验报告要求

实验报告需包含实验任务、实验平台、实验原理、实验步骤、实验数据记录、实验结果分析和实验结论等部分，特别是以下重点内容：

1. Kmeans算法的设计与实现。
2. 聚类结果的可视化。
3. 分析参数K及聚类中心初始值对算法结果以及收敛性的影响。

1.7　考核要求与方法

实验总分100分，通过实验报告进行考核，标准如下：

1. 报告的规范性10分。报告中的术语、格式、图表、数据、公式、标注及参考文献是否符合规范要求。
2. 报告的严谨性40分。结构是否严谨，论述的层次是否清晰，逻辑是否合理，语言是否准确。
3. 实验的充分性50分。实验是否包含“实验报告要求”部分的3个重点内容，数据是否合理，是否有创新性成果或独立见解。

1.8　案例特色或创新

本实验的特色在于：培养学生应用K-Means算法实现数据的聚类，学生能够理解K-Means算法的原理、实现K-Means算法并能研究分析算法参数对聚类结果的影响。能够对实验结果进行有效的可视化展示，培养学生对复杂工程问题建模和分析的能力。