深 圳 大 学 实 验 报 告

课程名称 机器学习

项目名称 实验五：聚类算法

学　　院 计算机与软件学院

专　　业 软件工程

指导教师 赖志辉

报 告 人 郑杨 学号 2020151002

实验时间 2022 年 6 月 1 日至 2022 年 6 月 2 日

实验报告提交时间 2022 年 6 月 2 日

教务处制

目录

一、实验目的与要求

1. 简述 K-means 聚类算法的原理与算法过程。
2. 熟练掌握 K-means 聚类算法与结果的展示，并代码实现，做一个 2 维或三维空间中的 2~3 类点（每个类有 10 个点）聚类实验，把聚类结果用不同的颜色与符号表示。
3. 实现人脸图像（取前 2~3 个人的人脸图像）聚类实验与旋转物体（在 COIL20 数据集中取前 2~3 个类的图像），把聚类结果用不同的颜色与符号表示，并把对应的图像放在相应点的旁边，让人一眼看出结果对不对；同时列表给出其在不同数据库在不同 K 时的聚类精度。

二、实验内容与方法

1、参考西瓜书，简述聚类思想、形式化表述与度量标准。
2、简述 K-means 聚类算法的原理与过程。
3、使用 Matlab 实现了 K-means 算法，在二维空间做了小数据上的聚类实验，对数据迭代过程进行可视化，理解算法运行过程。
4、实现不同数据集中人脸图像与旋转物体图像聚类效果并进行可视化，列表给出不同数据集在不同 K 时的聚类精度。

三、实验步骤与过程

1 聚类

1.1 基本概念

聚类是一种经典的无监督学习。在无监督学习中，样本的标签往往是未知的，学习的目标是通过对无标记样本的学习来揭示数据的内在结构与联系，为进一步的数据分析提供基础。聚类的思想来源于人类的思考方式，在没有人特意教给我们不同种群的称谓之前，我们就已具备了通过一些特征把不同事物区分开的能力。我们自然具备的这种凭借主观认知把特征近似或形态相同的事物归结为一个整体的思维，本身就是一种“聚类”。

聚类的结果是把一个数据集划分为若干个不相交的子集，我们把这些子集称为“簇”。“簇”的内部会存在着一定内在结构，比如同类人的人脸照片可能会划分为一个“簇”。当然，聚类算法事先不知道同个簇内的人脸照片属于哪类人，是我们通过结果去定义的。

聚类既能作为一个单独过程，用于寻找数据的内在结构，也可以作为分类等其他任务的前驱。

1.2 形式化表示

假设样本集合为 $D=\{x^{(1)},x^{(2)},...,x^{(m)}\}$，$m$ 为样本个数，某一个样本 $x^{(i)}$ 是一个 n 维列向量，聚类算法的目标是把集合 D 划分为 k 个不相交的簇 $C=\{C_1,C_2,...,C_k\}$，即满足以下

两式：

$$C_i \bigcap_{i \neq j} C_j = \varnothing (1 \leq i, j \leq k)$$

$$\bigcup_{i=1}^{k} C_i = D$$

对于样本而言，我们使用簇标记 $\lambda_i \in \{1,2,...,k\}$ 来记录样本 $x^{(i)}$ 被分到了哪个簇，即 $x^{(i)} \in C_{\lambda_i}$。

2 K-means 算法

2.1 算法原理

K-means 聚类算法的基本思想是使得簇内的样本尽可能聚集，也就是让簇内的每个样本与簇中心尽可能接近。假设无标签样本集合为 $D = \{x^{(1)}, x^{(2)}, ..., x^{(m)}\}$，要求的聚类类别数为 k，第 j 个簇为 C_j，第 j 个簇的均值向量为 μ_j，第 i 个样本类别向量为 $r^{(i)}$。其中，

$$\mu_j = \frac{1}{\sum_{i=1}^{m} [x^{(i)} \in C_j]} \sum_{x^{(i)} \in C_j} x^{(i)}$$

$$r_j^{(i)} = \begin{cases} 1, if\ (x^{(i)} \in C_j) \\ 0, else \end{cases}$$

那么我们可以用簇内每个样本与簇中心的距离之和来表示算法的优化目标，用数学语言形式化之后可以把目标函数表示为：

$$J(\{r^{(i)}\}_{i=1}^{m}, \{\mu_j\}_{j=1}^{k}) = \sum_{i=1}^{m} \sum_{j=1}^{k} r_j^{(i)} dist(x^{(i)}, \mu_j)$$

其中，$dist(x^{(i)}, \mu_j)$ 表示向量 $x^{(i)}$ 与 μ_j 之间的距离，可以使用曼哈顿距离、欧氏距离等距离计算方法进行衡量。为了方便，这里使用欧氏距离的平方进行距离衡量，即：

$$dist(x^{(i)}, \mu_j) = \| x^{(i)} - \mu_j \|_2^2$$

故 K-means 算法的目标函数为：

$$J(\{r^{(i)}\}_{i=1}^{m}, \{\mu_j\}_{j=1}^{k}) = \sum_{i=1}^{m} \sum_{j=1}^{k} r_j^{(i)} \| x^{(i)} - \mu_j \|_2^2$$

故优化目标为：

$$\min_{\{r^{(i)}\}_{i=1}^{m}, \{\mu_j\}_{j=1}^{k}} J(\{r^{(i)}\}_{i=1}^{m}, \{\mu_j\}_{j=1}^{k}) = \min_{\{r^{(i)}\}_{i=1}^{m}, \{\mu_j\}_{j=1}^{k}} \sum_{i=1}^{m} \sum_{j=1}^{k} r_j^{(i)} \| x^{(i)} - \mu_j \|_2^2$$

可以看到，我们的优化目标中包含 $m+k$ 个优化变量，直接对其求解比较困难，这里采用坐标下降法对该函数进行优化推导。包括 1)求出每一个 $r^{(i)}$ 2)求出每一个 μ_j

- 在求 $r^{(i)}$ 时我们假设其他所有变量已经求出，对于每一个 $r^{(i)}$，由 $r^{(i)}$ 的定义我们知道，它对目标函数 $J(\{r^{(i)}\}_{i=1}^{m},\{\mu_j\}_{j=1}^{k})$ 的贡献是 k 个簇的均值向量与样本 $x^{(i)}$ 产生的 k 个距离中的一个。于是我们只需要最小化这个贡献就可以了，即：

$$r_j^{(i)}=\begin{cases}1, if\ (j=\arg\min_{j'} \| x^{(i)}-\mu \|_2^2)\\ 0, else\end{cases}$$

这个优化结果直观看来就是，对于每一个样本归类到离它距离最近的簇中心对应的簇中。

- 在求 μ_j 时我们依然假设其他所有变量已经求出，对于每一个 μ_j，只有样本 $r^{(i)}$ 属于第 i 个簇时，它对目标函数才会产生贡献。于是 μ_j 对目标函数 $J(\{r^{(i)}\}_{i=1}^{m},\{\mu_j\}_{j=1}^{k})$ 产生的贡献如下：

$$J(\mu_j)=\sum_{i=1}^{m} r_j^{(i)} \| x^{(i)}-\mu_j \|_2^2=\sum_{i=1}^{m} r_j^{(i)}(x^{(i)}-\mu_j)^T(x^{(i)}-\mu_j)$$

$$=\sum_{i=1}^{m} r_j^{(i)}(x^{(i)^T}x^{(i)}-x^{(i)^T}\mu_j-{\mu_j}^T x^{(i)}+{\mu_j}^T\mu_j)$$

令

$$\frac{\partial J(\mu_j)}{\partial \mu_j}=\frac{\partial}{\partial \mu_j}(\sum_{i=1}^{m} r_j^{(i)}(x^{(i)^T}x^{(i)}-x^{(i)^T}\mu_j-{\mu_j}^T x^{(i)}+{\mu_j}^T\mu_j))$$

$$=\sum_{i=1}^{m} r_j^{(i)}\frac{\partial}{\partial \mu_j}(x^{(i)^T}x^{(i)}-x^{(i)^T}\mu_j-{\mu_j}^T x^{(i)}+{\mu_j}^T\mu_j)$$

$$=\sum_{i=1}^{m} r_j^{(i)}(-2x^{(i)}+2\mu_j)=\mu_j\sum_{i=1}^{m}2r_j^{(i)}-\sum_{i=1}^{m}2r_j^{(i)}x^{(i)}=0$$

得：

$$\mu_j\sum_{i=1}^{m}2r_j^{(i)}=\sum_{i=1}^{m}2r_j^{(i)}x^{(i)}$$

即：

$$\mu_j=\frac{\sum_{i=1}^{m} r_j^{(i)}x^{(i)}}{\sum_{i=1}^{m} r_j^{(i)}}$$

故求出了 μ_j 的最优解，这个结果直观理解上就是把某一个簇的均值设置为簇中所

有样本点的均值向量。

2.2 算法流程

由上节优化推导可得，K-means 算法可以使用迭代实现，具体流程如下：

输入为样本集合 $D=\{x^{(1)},x^{(2)},...,x^{(m)}\}$，聚类的簇数为 k。

输出为簇划分 $C=\{C_1,C_2,...,C_k\}$。

- 首先从 D 中随机选择 k 个样本点作为初始的 k 个质心向量 $\{\mu_1,...,\mu_k\}$
- 进行循环
  - 将所有的簇划分 C 初始化为空集，即 $\forall j, C_j=\varnothing$
  - 循环每一个样本点，对于样本点 $x^{(i)}$，初始化 $r^{(i)}$ 为零向量计算 $x^{(i)}$ 与各个质心向量 μ_j 的距离 $d_{ij}=\| x^{(i)}-\mu_j \|_2^2$，将距离 $x^{(i)}$ 最近的质心向量对应的类记为 j'，则令 $r_{j'}^{(i)}$。
  - 更新每一个质心向量 μ_j 为：

$$\mu_j=\frac{\sum_{i=1}^{m} r_j^{(i)} x^{(i)}}{\sum_{i=1}^{m} r_j^{(i)}}$$

  - 若所有的质心向量坐标没有改变，则跳出循环；否则继续循环
- 输出此时的簇划分 $C=\{C_1,C_2,...,C_k\}$

2.3 伪代码

Algorithm 1 K-means

Input: Data vectors $D = \{x^{(1)}, ..., x^{(m)}\}$, number of clusters k

Output: $\{r^{(1)}, ..., r^{(k)}\}$

for $j \leftarrow 1...k$ **do**
  $k^{'} \leftarrow \text{RandomInteger}(1, m)$
  $\mu_j \leftarrow x^{(k^{'})}$
end for
repeat
  for $i \leftarrow 1...m$ **do**
    $r^{(i)} \leftarrow [0, 0, ..., 0]$
    $k^{'} \leftarrow arg\ min_j ||x^{(i)} - \mu_j||^2$
    $r_j^{(i)} \leftarrow 1$
  end for
  for $j \leftarrow 1...k$ **do**
    $N_j \leftarrow \sum_{i=1}^{m} r_j^{(i)}$
    $\mu_j \leftarrow \frac{1}{N_j} \sum_{i=1}^{m} r_j^{(i)} x^{(i)}$
  end for
until none of the μ_j change
return $\{r^{(1)}, ..., r^{(k)}\}$

2.4 优缺点

对 K-means 算法的优缺点进行总结。

优点：

- 原理简单，实现简单，收敛速度较快
- 聚类效果较好
- 算法可解释性强
- 需要调参的参数只有一个，簇数 k

缺点：

- k 值需要人为设定，不同 k 值得到的结果不一样
- 对初始的簇中心敏感，不同选取方式会得到不同结果
- 对异常值敏感
- 不适合太离散的分类、样本类别不平衡的分类、非凸形状的分类

3 K-means 算法相关实验

3.1 K-means 算法在二维平面上的小样本聚类实验与聚类过程可视化

随机了三个数据区域的数据点，每个区域 100 个数据点，可视化算法迭代过程与聚类结果。原始数据点如图 1 所示，

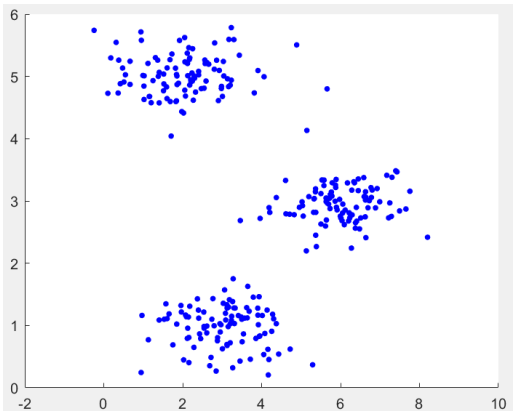

图 1：二维小数据

第一次运行算法迭代过程如图 2 所示，第二次运行算法迭代过程如图 3 所示，可以看出，算法迭代次数与随机的起点有关，起点不同，最终分成的簇效果也不同。这使得 K-means 算法整体的运行速度与性能不稳定。

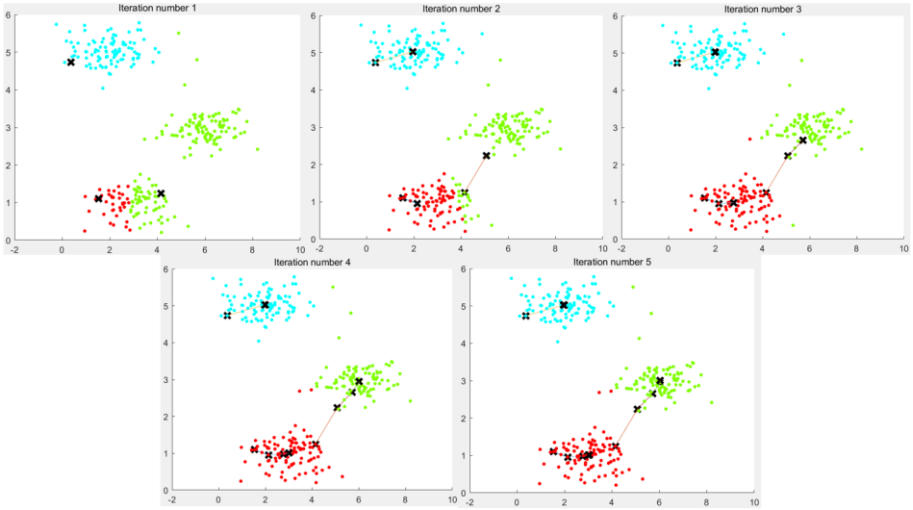

图 2：算法在二维小数据上的迭代过程（1）

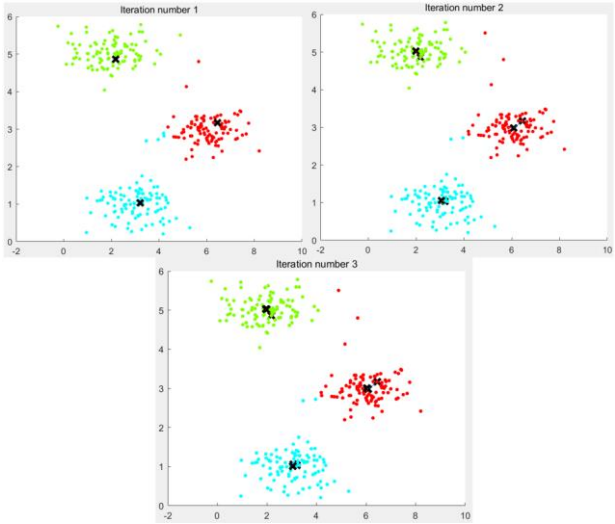

图 3：算法在二维小数据上的迭代过程（2）

3.2 K-means 算法在人脸数据集上的聚类实验

在 ORL 数据集上进行实验。取出前三类人的所有人脸图像（每类 10 张图像）并使用 PCA 降维至二维平面，再使用 K-means 算法进行聚类，实验结果如下。

PCA 降维之后的三类人物图像分布如图 4 所示，基本上把三类人物图像划分开了，其中的颜色只是为了与聚类结果比对，没有打标签的意思。聚类结果如图 5 所示，可以看出，聚类效果较好，K-means 算法在很少的迭代次数下就已经收敛。右下角的簇分布较散，因为该类人脸图像受光照与姿势的影响较大，而中间的簇分布紧密，因为该类人物图像非常相似，没有太大的光照与姿势影响。

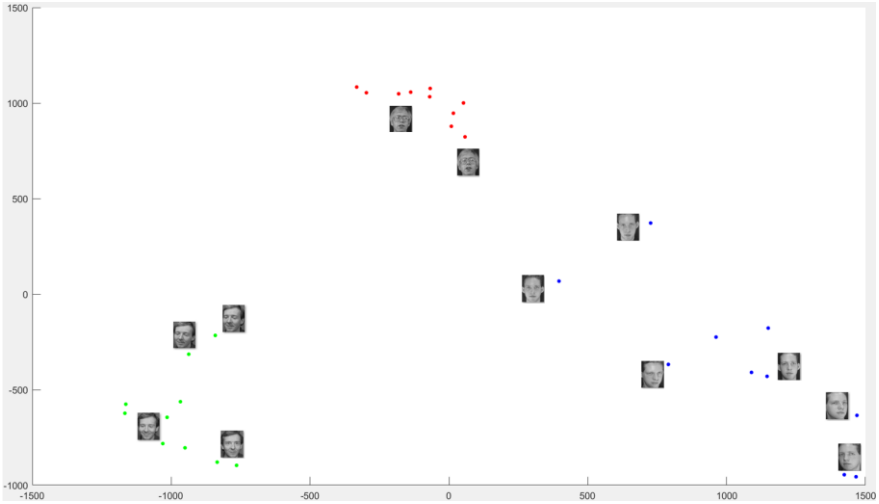

图 4：ORL 数据集降维之后的数据分布

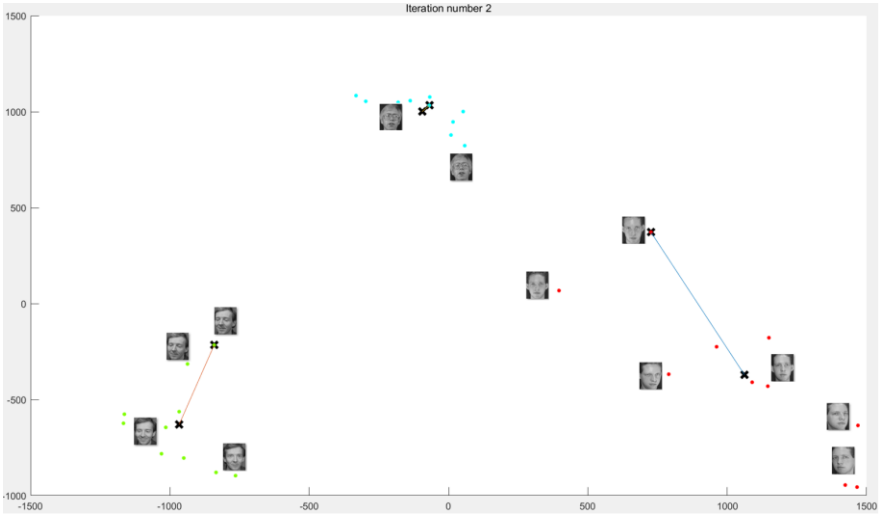

图 5：ORL 数据集的聚类结果

3.3 K-means 算法在旋转物体数据集上的聚类实验

在 COIL20 旋转物体数据集上进行实验。取出前三类物体的所有图像（每类 71 张图像）并使用 PCA 降维至三维空间，再使用 K-means 算法进行聚类，实验结果如下。

PCA 降维之后的物品图像如图 6 所示，可以看到，PCA 基本保持了物品的表面流形。聚类结果如图 7 所示，可以看到，聚类效果并不好，K-means 没有注意到数据的流形结构，

或者说数据的密度关系，只是在欧式空间中最小化簇内的距离，这会使得这种具有流形结构的数据使用 K-means 进行聚类的效果不好，破坏了数据的流形结构。

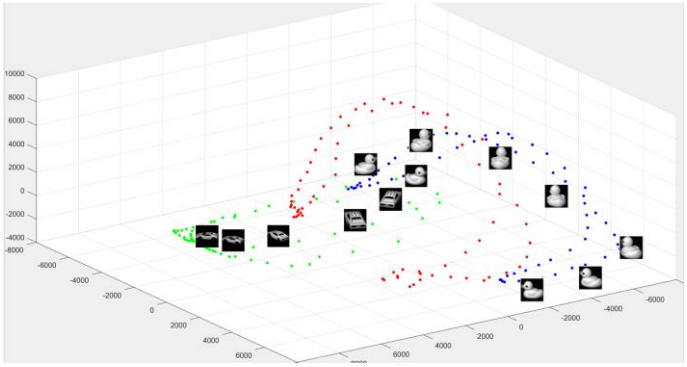

图 6：PCA 对三类物品图像降维至三维空间后的数据分布

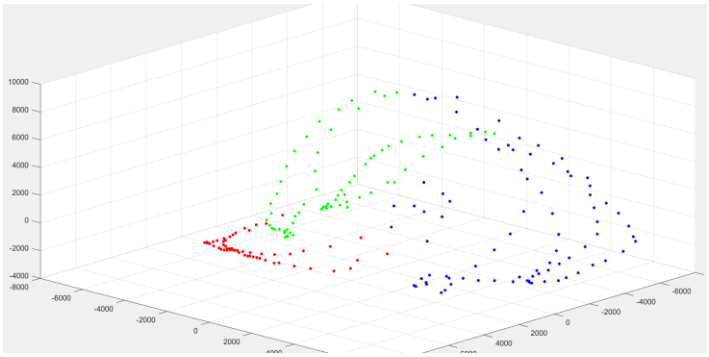

图 7：K-means 对物品数据集的聚类结果

3.4 K-means 算法在不同人脸数据集上取不同 K 时的聚类精度

聚类精度按照每个簇中原始标签比重最大的那一类作为簇的类别，将原始标签与簇类别相同的样本判断为聚类正确，原始标签与簇类别不同的样本判断为聚类错误。

在 ORL、Yale 与 AR 数据集下，分别取出前四类人物图像的所有样本进行聚类（不使用 PCA 进行降维），取 K=2，3，4，5，6 进行测试，测试结果如表 1 所示。

表 1：不同数据集上 K 取不同值时算法的聚类精度

	2	3	4	5	6
ORL	0.500	0.725	0.750	**0.925**	**0.925**
AR	0.477	0.477	0.705	0.682	**0.841**
Yale	0.500	0.442	0.673	0.673	**0.692**

四、实验结论或体会

(1) 理解了 K-means 算法思想并详细地优化推导了 K-means 算法。

(2) 在小规模二维数据集上对算法进行测试并进行可视化，比较了算法在不同起点时的性能。

(3) 在人脸数据集与旋转物体数据集上使用 PCA 进行降维再使用 K-means 聚类并进行可视化。

(4) 计算了在不同数据集上算法取不同 k 时的聚类精度并进行比较。

指导教师批阅意见： 成绩评定： 指导教师签字： 年　　月　　日
备注：

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。