**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 唐健龙 学号 2018132100**  **实验时间 2020年5月18日至2020年5月25日**  **实验报告提交时间 2020年5月24日** |

**教务处制**

# 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

**k-means算法：**

**原理：**

在无监督学习中，有一个算法简单有效，且通常只需要一个参数，其鲁棒性也比较强，即k-means算法，它能够在无标签情况下，进行自发的聚类，并且可以很快收敛。

k-means算法通过给定k个初始中心点，随后通过某种相似度度量，将数据点分配给某个中心点，最后重新计算同类中心点并进行中心点的迭代更新，算作完成一次迭代。

而这里的相似度度量，常用的有两种，一种是欧氏距离，这是非常常用的相似度度量方法，适用于学习数据本身的结构；还有一种常用的相似度度量方法是余弦相似度，这种度量侧重于数据的高维方向的聚类学习，这种度量在出现文本词向量等场景能够用到。

值得一提的是，k-means算法收敛速度很快，并且，有研究证明了k-means是绝对能够收敛的，这无疑为机器学习提供了很大的便利。

对于初始点的选择，有许多方法，最初的算法是随机产生点的，也有一些方法，首先随机生成一个中心点，然后找到距离其最远的点作为第二个中心点，接着找到距离这两个点最远的点再作为第三个中心点，以此类推得到k个初始中心点，方法虽多，还是会有一个问题难以解决——离群点问题，最好的解决方法其实还是多尝试几次并取最好的结果。

还有一个问题是k的选取，在一些情况下，我们并不知道数据集会有几个类，只能手动尝试k的取值，通常可以在尝试连续的k值情况下，作出损失函数的下降曲线，找到接近变化平缓的那个k作为k值比较合适

接下来说明，对于欧氏距离，为什么通过取同类样本均值作为迭代值：

假设 ，为第 个样本， 为第 个中心点， 为被分为第 类样本点的集合， 定义损失函数为

令其对 求导并使之等于0，得到

即

其中， 为被分为第 类的样本数，也就是说，只需要取同类样本点的均值，就完成了一次迭代。

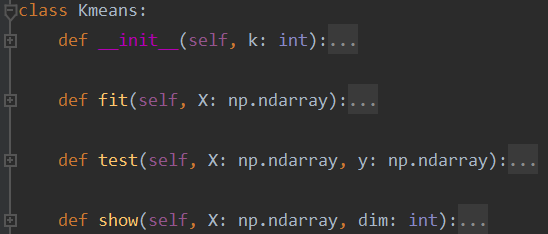
在经过不多的迭代之后，中心点的更新很快能够收敛。

**算法：**

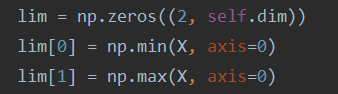
1. 初始化k个中心点，作为分类依据
2. 遍历每一个样本点，通过特定的相似度度量方法，确定归属中心点
3. 遍历每一个类，计算同类样本点的均值，得到其值以更新该类中心点
4. 回到（2）继续循环，直到中心点基本确定不再移动

# 三、实验步骤与过程

首先定义一个k-means类，需要k确定类别数，需要数据集进行模型拟合，加上模型训练函数，画出分类效果图的函数，和测试聚类精度的功能

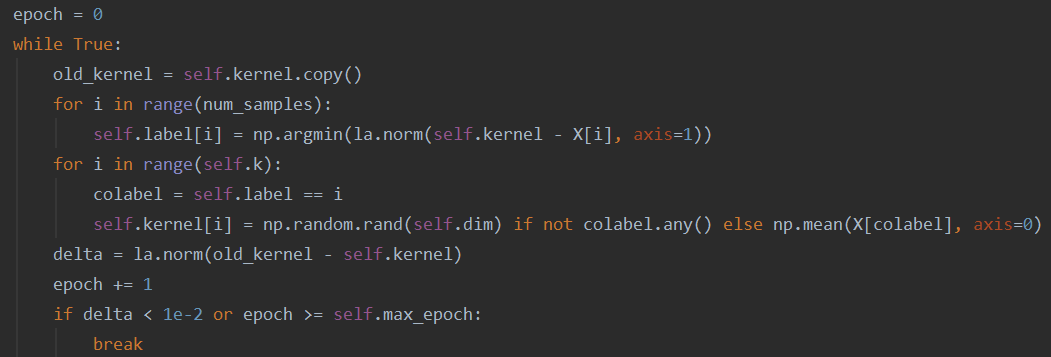


为了随机生成合适范围的初始中心点，需要计算每个维度的最小值最大值，并生成该范围内的值，作为聚类核

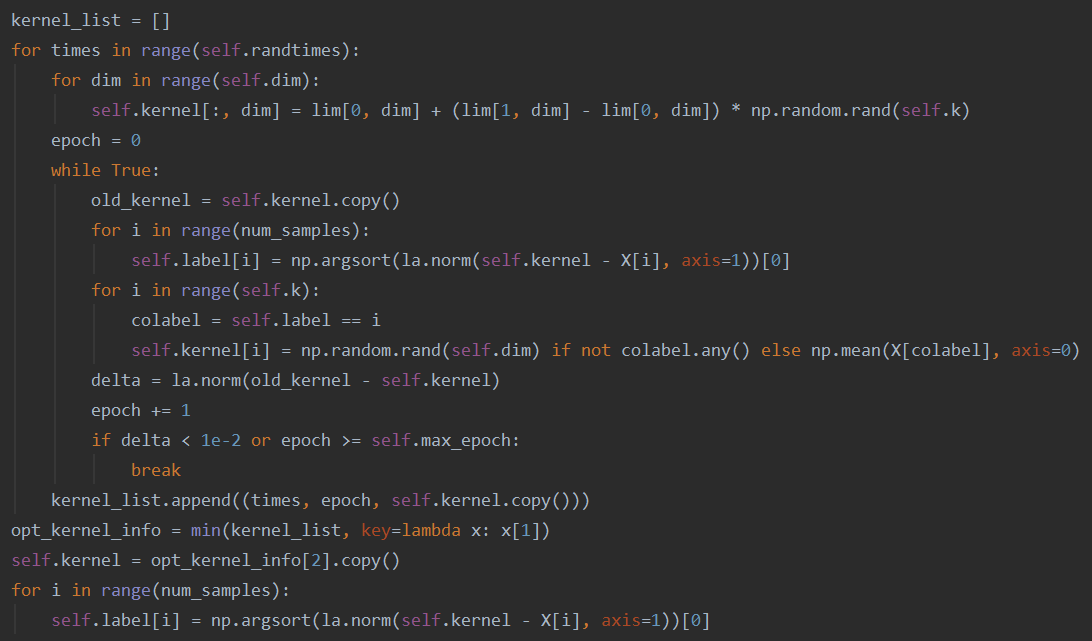




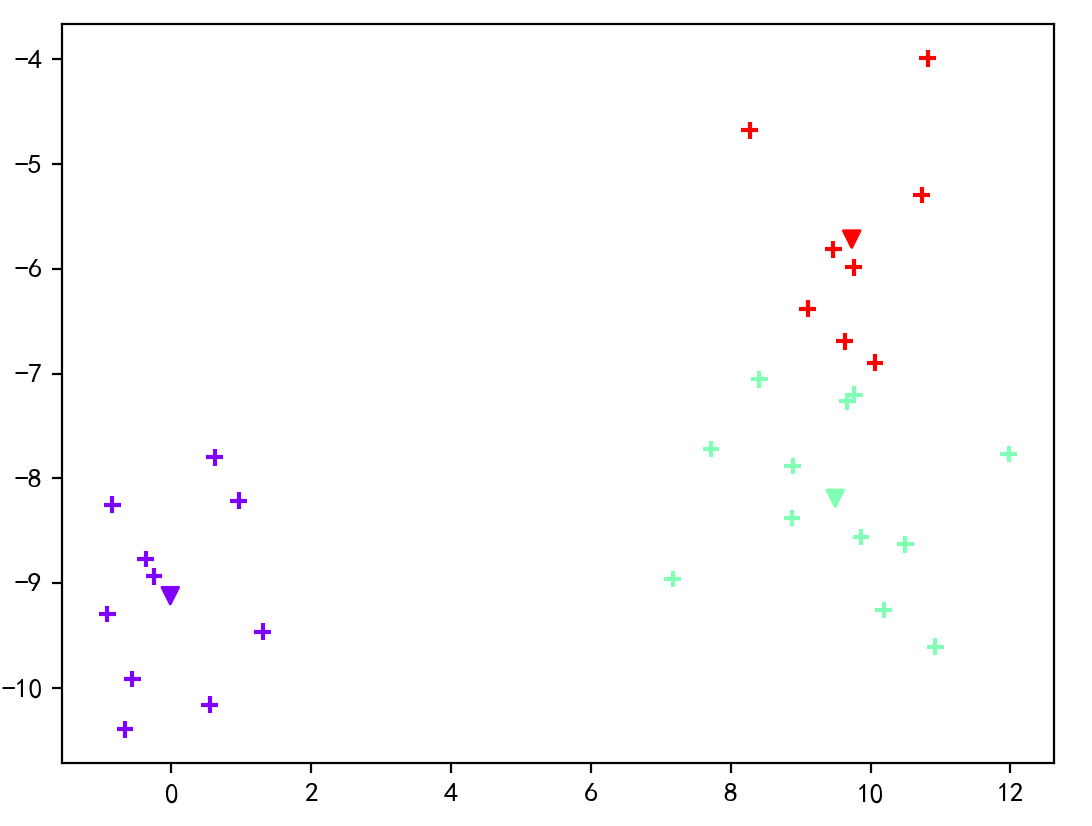
为了能够保证计算效率，设定最大迭代次数，以及记录中心点变动情况，若迭代次数大于最大迭代次数，或中心点基本不再变动，就退出迭代。由于在迭代过程中，可能会出现某个聚类核的周围无数据点作为其归属点的情况，我想到的方法是，在这种情况下为其重新生成新的聚类核。



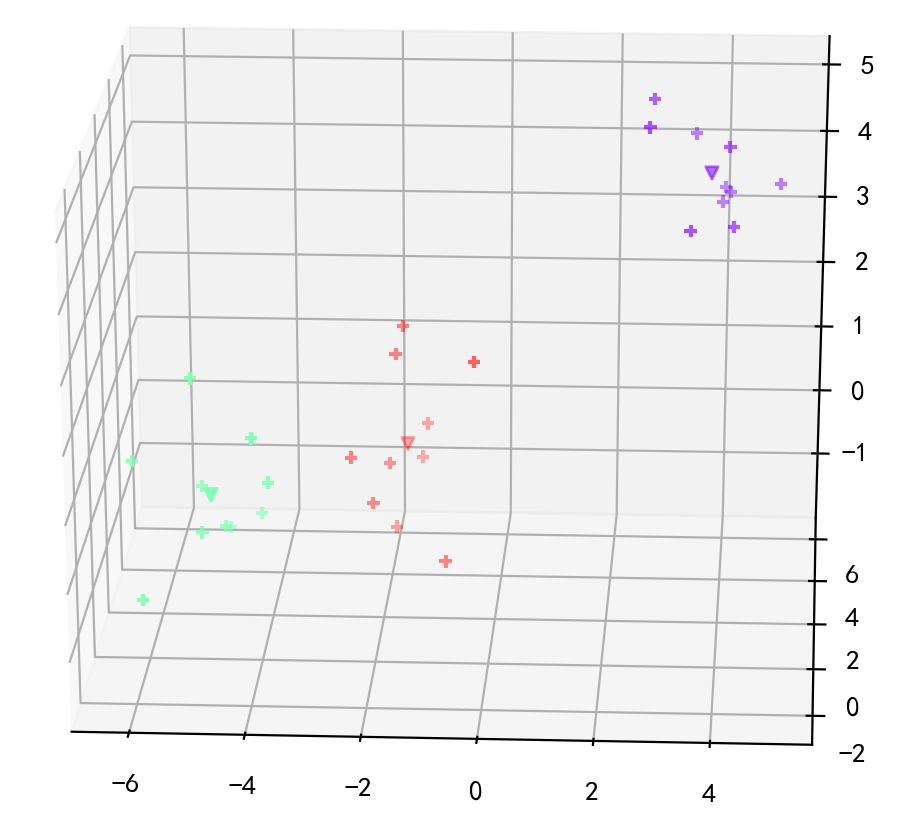
经过个人实验发现，几乎所有能够快速收敛的迭代结果都能很好聚类，因此，进行一定次数的尝试并记录结果，最后找到迭代步数最少的那次对应的迭代结果作为模型拟合结果，最后计算每一个点对应的标签并存储起来备用，如图所示



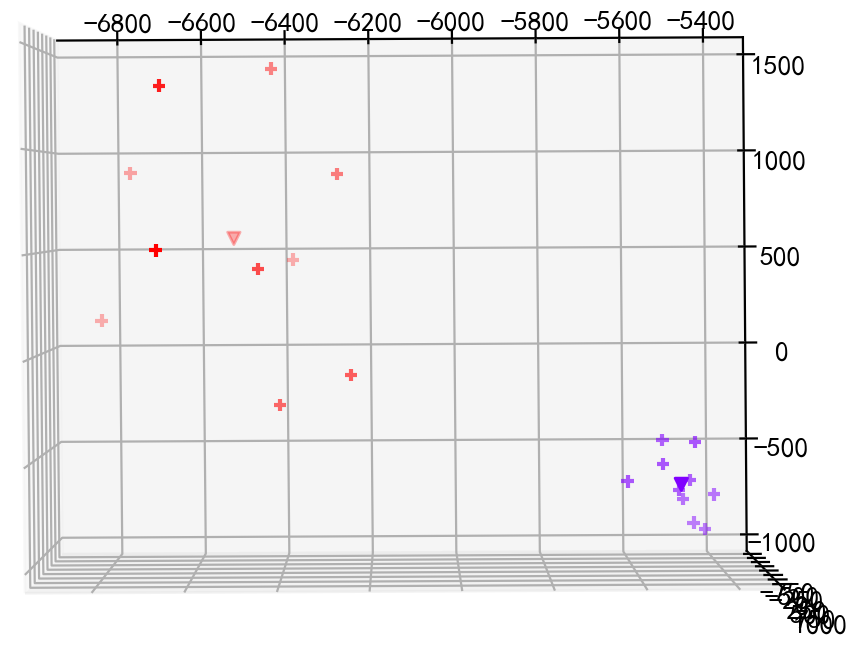
现在利用sklearn.datasets.make\_blobs函数随机生成二维数据，共3类点，每个类有10个点，接下来进行模型拟合，拟合的结果如下所示



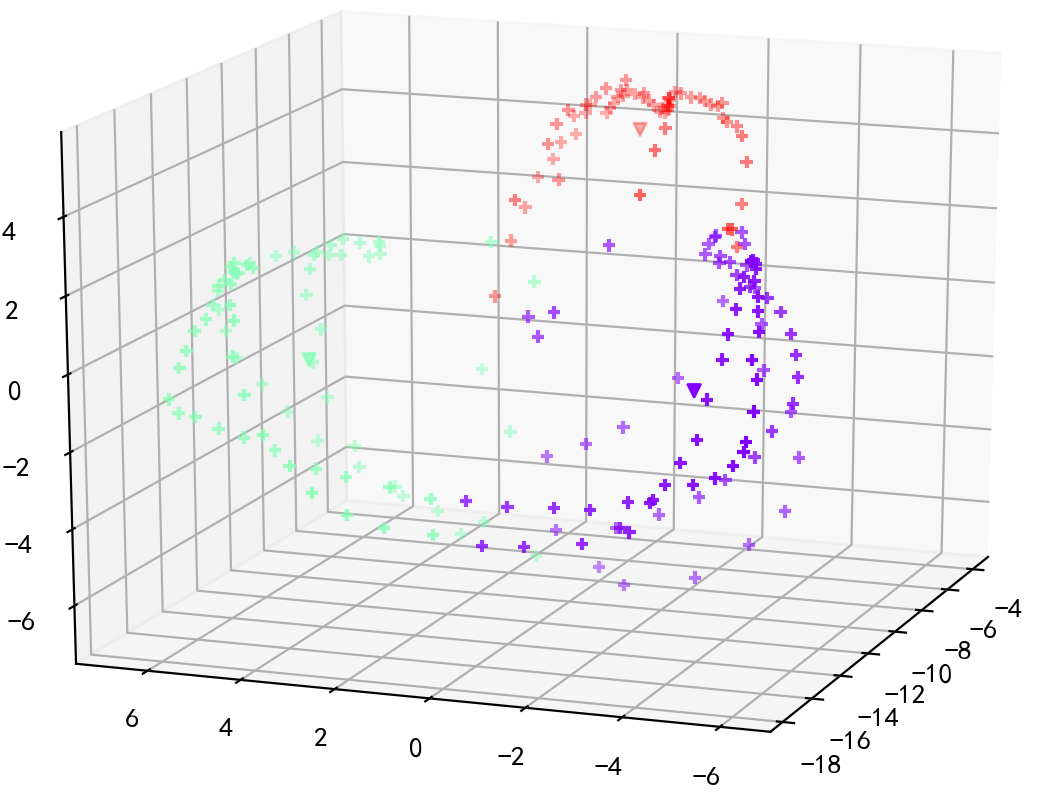
接下来再随机生成三维数据，共3类点，每个类有10个点，接下来进行模型拟合，拟合的结果如下所示



接下来测试ORL4646.mat数据集，该数据集的每张人脸图像会有少许的角度变化，正好适合k-means聚类，选择前2个人的每人10张照片拉成行向量，通过PCA将其降维到3维，随后通过k-means进行拟合，结果如图所示



接着测试旋转物体，取COIL20.mat数据集，并取前两个物体的图像拉成行向量，通过PCA降维到3维，随后进行k-means拟合，结果如图所示，容易发现此时k-means表现不佳



最后，给出聚类精度关于不同k值和不同数据库的表格，其中，聚类精度的判断是依据聚类为同类中的占比最大的原始类标签为聚类类别，剩余的原标签与该类别不同的数据点认为聚类错误，聚类精度表格如下所示

|  |  |  |  |
| --- | --- | --- | --- |
| 通过PCA降至3维 | | | |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| 通过PCA降至8维 | | | |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 四、实验结论或体会

**1. k-means算法原理简单，并且实现起来也简单，效果也不错**

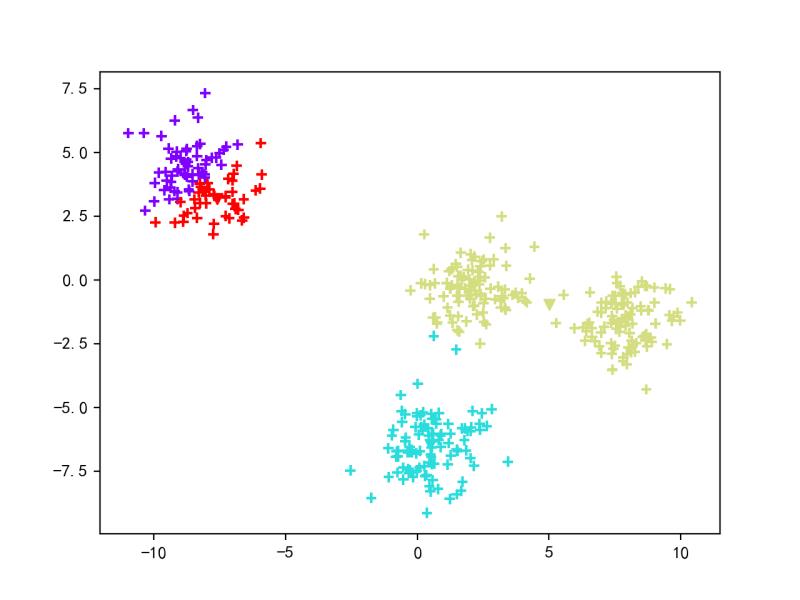
**2. 确定k值较为困难，初始中心点很关键但是不得不随机生成**

**3. 部分离群值容易影响聚类效果**

**4. 有时候一些数据分布可能并不太适合直接使用k-means算法，可以考虑配合其他算法，如旋转物体可以考虑预先通过流形学习进行降维处理**

# 五、思考题

**在进行实验时，我注意到经常出现一种情况，即在某一簇数据点附近恰好出现了两个中心点，而远处则存在两簇数据点和一个中心点，这种情况常常发展为，两个中心点聚集了同一个类，而另一个中心点在两个类中间，这显然不是好的聚类方式，如下所示**



注意到这种情况，对于左上角的两个中心点，两类点的分割平面附近常常聚集较大量的数据点，而右边的一个中心点聚两个类的情况，该中心点邻域点的数量相对其他中心点

较为稀疏，或许可以找到一些方法，能够检测出同时存在分割平面附近数据点密集且远处另一个中心点邻域点稀疏的情况，然后就能够讲冗余的中心点更新到缺省中心点的位置，可能可以起到一定的效果，在数据集较大的时候，这种方法能够凸显其优势，减少更多尝试次数，加快迭代速率。

|  |
| --- |
| 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。