无监督学习部分

PB16060130 顾健鑫

1. 算法描述
    a. KMeans

    KMeans 算法的基本思想为随机选取 K 个数据点，每一次迭代重复以下步骤直至 K 个点的坐标不改动或均小于某个阈值

    i. 计算出各数据点与 K 个聚类点之间的距离，此处用 Euclid 距离即可。具体实现类似于 KNN 中计算数据集中两点之间距离的实现，通过 Numpy 的广播与向量化去除循环以提高速度
    ii. 对每个数据点选出 K 个聚类点中距离最近的那个作为此数据点的类别
    iii. 对于每个聚类点，通过本轮迭代中归到该聚类点的数据点在数据的每一个维度上求平均值，得到一个新的点作为下一轮迭代该聚类点的取值

    具体实现为 KMeans.py 中 KMeans 类的 train 方法，其返回的为一个向量，表示每个数据点最终归到哪个聚类点。

    b. PCA 主元分析

    PCA 算法的计算步骤为

    i. 计算出数据集 X 的协方差矩阵 covX；
    ii. 计算出协方差矩阵的特征值和特征向量
    iii. 将特征值从大到小排列后计算前缀和，并与给定的阈值进行比较，得到应选前 M 个特征值与其对应的特征向量
    iv. 使用特征向量构成的矩阵与数据集 X 计算出降维后的矩阵 reducedX
    v. 选取 reducedX 中的前两维做散点图

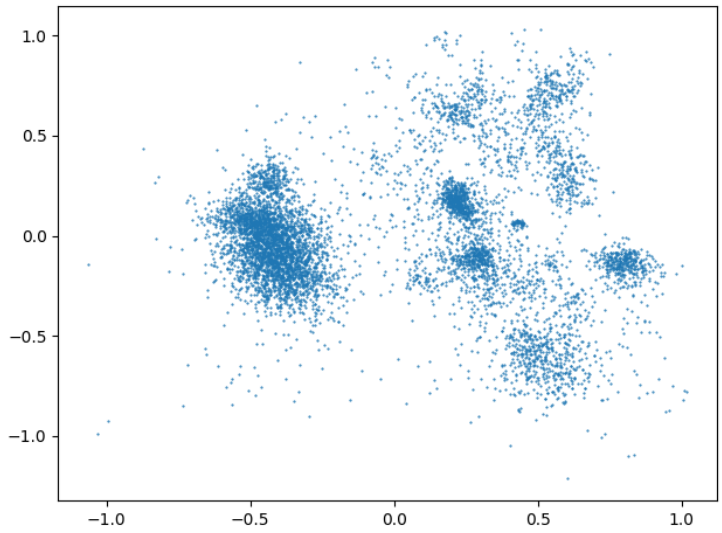
    c. 层次聚类

    具体描述与实验要求相同，在此不再赘述。

2. 实验结果

    KMeans 与 PCA

    当阈值选为 0.7 时，使用 PCA 降维后得到的散点图如下



KMeans 和 PCA 后再使用 KMeans 进行聚类得到的纯度与兰德系数如下图所示（图中结果均为在同一参数条件下运行 3 次后取平均得到的结果）

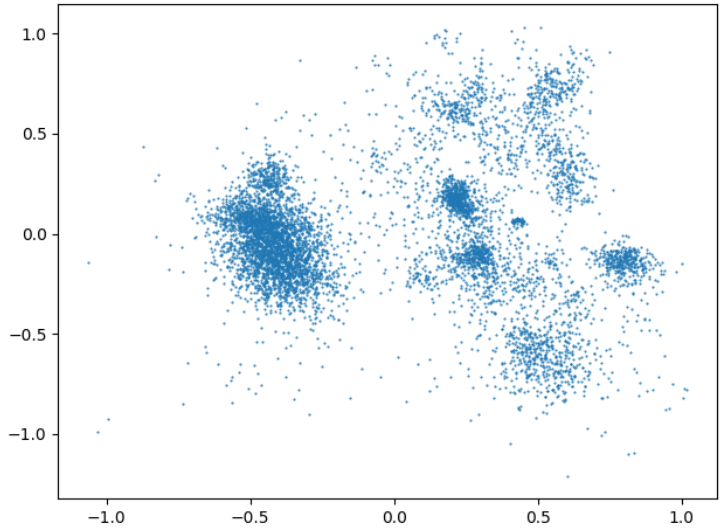
Purity

	4	8	10
KMeans	0.8157053509381514	0.8455872133425991	0.8434097753069261
PCA + KMeans	0.7834144081538105	0.869029418577716	0.8255733148019458

RI

	4	8	10
KMeans	0.7522013009966546	0.6746126301812909	0.6834277386973895
PCA + KMeans	0.6688824155254078	0.6789197481312929	0.5870635832797375

当阈值选为 0.8 时，使用 PCA 降维后得到的散点图如下



KMeans 和 PCA 后再使用 KMeans 进行聚类得到的纯度与兰德系数如下图所示（图中结果均为在同一参数条件下运行 3 次后取平均得到的结果）

Purity

	4	8	10
KMeans	0.8157053509381514	0.8455872133425991	0.8434097753069261
PCA + KMeans	0.7861477878156126	0.8793606671299513	0.8767199444058374

RI

	4	8	10
KMeans	0.7522013009966546	0.6746126301812909	0.6834277386973895
PCA + KMeans	0.6671121708571263	0.6377521187353449	0.6101444599452263

通过上述的对比结果可以得出，使用 PCA 主元分析法进行降维后，聚类的运算量会减少，运算速度相应也会有提升；但在本次实验中使用的青蛙数据集上，使用 PCA 进行主元分析降维后，并不能在所有的 K 值的选项中提升效果，在阈值选为 0.7 时，K 取 8 时有小量的提升，但当 K 取 4 与 10 时，聚类效果却有一定程度的下降；在阈值取为 0.8 时，兰德系数均呈现出下降，但纯度仅在 K 取 4 时发生下降。
综合实验结果，选择 K 取 8 进行 KMeans 聚类会是较好的选择。

层次聚类
由于算法的效率问题，层次聚类在对所有数据进行处理时，运行速度过于缓慢，因此在此仅使用 1000，1500，2000，2500 个数据点在 K 取 4 时的运行结果作实验结果展示

	Purity	RI
1000	0.627	0.477584
1500	0.604667	0.464927
2000	0.621	0.478678
2500	0.6016	0.461881

由此可以看出，层次聚类可以达到一定的分类效果，但由于数据集的使用规模的限制，其结果并不能给很好的反应与 KMeans 算法或是 PCA+KMeans 算法之间的优劣。

3. 实验总结

通过本次实验，我对三种聚类算法的具体实现有了更加深刻的认知，并且对 numpy 的中广播以及向量化的应用有了更深刻的理解，并清楚的认识到了代码去循环化带来的性能上的提升，收获颇多。