

模式识别 Kmeans / FCM

22009200601 汤栋文

2024 年 12 月 06 日

目录

摘要	2
1. 方法	2
1.1 数据集	2
1.2 验证方式	2
2. 实验	3
2.1 轮廓系数评价	3
2.2 可视化评价	3
3. 结论	4

摘要

本研究通过 Iris 数据集和 MNIST 数据集对 KMeans 和 FCM 两种聚类算法进行了对比分析。通过对数据集进行标准化处理，使用轮廓系数和 PCA 可视化方法来评估聚类效果。结果显示，KMeans 和 FCM 在 Iris 数据集上表现较为相似，均能较好地识别出 setosa 类别，但在区分 versicolor 和 virginica 类别时存在一定混淆。在 MNIST 数据集上，KMeans 聚类结果显示各个簇之间分离较明显，而 FCM 按隶属度最高划分未能充分体现其软聚类优势。

1. 方法

1.1 数据集

本研究使用了 Iris 数据集和 MNIST 数据集。Iris 数据集包含了 150 个样本，每个样本有 4 个特征，分别是萼片长度、萼片宽度、花瓣长度和花瓣宽度，这些特征都是连续数值型数据，标记为 3 种鸢尾花的类。MNIST 数据集包含了 70,000 个手写数字的灰度图像（其中训练集 60,000 个，测试集 10,000 个），每个图像的大小为 28x28 像素，标记为 0 到 9 的数字类别。

为了消除特征间的量纲影响，我们对原始数据进行了标准化处理。对于 Iris 数据集，我们对每个特征值减去该特征的平均值并除以标准差，使得每个特征具有零均值和单位方差。对于 MNIST 数据集，同样地，我们对每个像素值进行了相同的操作，即减去所有训练图像中对应像素位置的平均值，并除以标准差，确保每个像素特征也具有零均值和单位方差。

1.2 验证方式

轮廓系数 (Silhouette Coefficient) 是一个用来评估聚类效果的指标，它结合了凝聚度 (Cohesion，即一个样本与其所在簇中其他样本的平均距离) 和分离度 (Separation，即一个样本与最近的另一个簇中所有样本的平均距离)。轮廓系数可以衡量单个样本、某个特定簇或整个数据集的聚类效果。具体步骤如下：

1. 计算凝聚度 a : 对于某个样本 i , 计算它与所在簇中所有其他样本平均距离, 记为 $a(i)$ 。
2. 计算分离度 b : 对于某个样本 i , 计算它与最近的另一簇中所有样本平均距离, 记为 $b(i)$ 。
3. 计算轮廓系数 s : 对于某个样本 i , 轮廓系数的计算公式为:
$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$
4. 聚类的整体轮廓系数: 整个数据集的轮廓系数可以通过对所有样本的轮廓系数取平均值得到。这一值可以帮助我们评估聚类算法在整个数据集上的表现。

另一方面通过 PCA，我们可以将高维度的数据降维到 2 维空间，并对其进行可视化处理。将聚类后的数据点映射到主成分空间中，不同颜色或形状的点代表不同的簇，这样我们就可以通过观察这些点在图中的分布情况来评估聚类的效果。如果同一簇的数据点在降维后的图中形成了明显的簇状结构，且不同簇之间有较为明显的分离，则说明聚类效果较好。

2. 实验

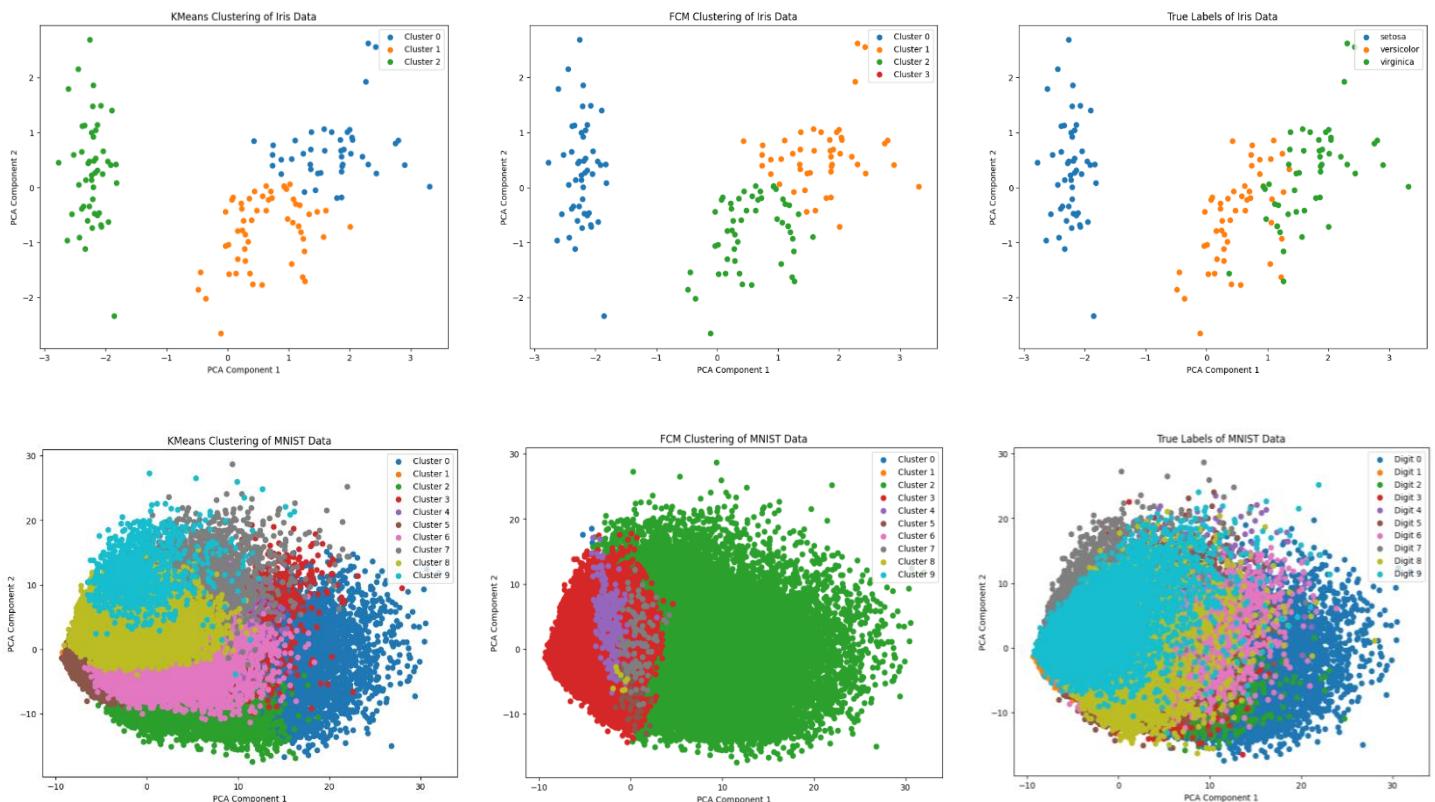
2.1 轮廓系数评价

Method	Dataset	Silhouette Coefficient
KMeans	Iris	0.46
FCM	Iris	0.46
KMeans	MNIST	0.01
FCM	MNIST	-0.03

这个表格展示了两种聚类方法（KMeans 和 FCM）在两个数据集（Iris 和 MNIST）上的轮廓系数（Silhouette Coefficient）评价结果。轮廓系数是衡量聚类效果的指标，数值越高表示聚类效果越好。表格中的数据表明：

在 Iris 数据集上，KMeans 和 FCM 的轮廓系数均为 0.46，说明这两种方法在该数据集上的聚类效果相当，这与 2.2 部分的可视化评价一致。在 MNIST 数据集上，KMeans 的轮廓系数为 0.01，而 FCM 的轮廓系数为 -0.03，表明这两种方法在该数据集上的聚类效果都不理想，且 FCM 的效果更差。结合可视化结果，这也从另一个方面揭示了轮廓系数在评价聚类结果的合理性。在不同的数据集上，聚类方法的效果可能会有显著差异，需要根据具体情况选择合适的方法。在 Iris 数据集中，两种方法表现相似，但在更复杂的 MNIST 数据集中，KMeans 略胜一筹，这也是因为 FCM 未能充分发挥其软聚类的优势。

2.2 可视化评价



这张图展示了 Iris 数据集和 MNIST 数据集的聚类结果和真实标签的对比。左边的图是 KMeans 聚类结果，中间是 FCM 聚类的结果，右侧是真实标签。上边是 Iris 数据集，下边是 MNIST 数据集。

在 Iris 数据集上，通过对聚类结果和真实标签可以观察到，KMeans 和 FCM 聚类方法都较好地将 setosa 类别分离出来，但在区分 versicolor 和 virginica 时存在一些混淆。在 MNIST 数据集上，KMeans 将数据分成十个聚类。结果显示数据点分布较为均匀，各个簇之间有明显的分离，但某些簇内样本的边界比较生硬，可能导致边界点的错误聚类。FCM 算法允许数据点属于多个簇，此处图中选择隶属度最高的一类簇作为真实类别，得到上图结果，我们观察到绝大多数点都划分到 cluster 2，但这并不一定合理，FCM 算法的优势就是提供了软聚类的隶属度，这种划分方法其实并未体现出 FCM 算法的优势。

3. 结论

通过实验，我们发现 KMeans 和 FCM 算法各有优缺点。KMeans 在数据点分离方面表现较好，但可能在簇边界处理上存在误差。FCM 算法则提供了软聚类的隶属度，聚类划分更加灵活，但还需要有另一套划分方法辅助才能充分发挥其优势。总体而言，选择合适的聚类算法需要结合具体数据集和应用场景进行综合考虑，以达到最佳聚类效果。