模式识别作业六

December 2022

1 题目一

K-Means 聚类算法是一种基于距离的聚类算法，它的基本思想是：选择 K 个初始为簇中心，然后将每个数据点分配到最近的中心所在的簇。然后，对于每个簇，重新计算簇中心（即求取簇中所有数据点的平均值）。最后，重新分配数据点到最近的中心所在的簇。这个过程会不断进行，直到簇中心不再变化为止。簇中心可以看作是混合高斯密度函数的均值，而每个簇中的数据点就可以看作是由这个高斯分布生成的样本。因此，K-Means 聚类算法可以被看作是一种混合高斯密度函数估计的方法。但是 K-Means 聚类算法并不是真正意义上的混合高斯密度函数估计算法，因为它并不考虑高斯分布的协方差。实际上，K-Means 聚类算法假设所有簇的协方差都是相等的，并且假设所有簇的权重（即簇中数据点的数量）也是相等的。这使得 K-Means 聚类算法在处理形状不均匀的数据集时效果不佳，因为它不能很好地拟合不同形状的簇。

K-Means 算法的步骤：

1. 选择 k 个初始聚类中心。通常是随机选择 k 个数据点作为初始聚类中心。
2. 将每个数据点分配给最近的聚类中心。对于每个数据点，计算它到每个聚类中心的距离，并将数据点分配给距离最近的聚类中心。
3. 更新聚类中心。计算每个聚类的平均值，将其作为新的聚类中心。
4. 重复步骤 2 和 3，直到聚类中心不再发生变化。
5. 输出最终的聚类结果。

影响 K-Means 算法的聚类性能的因素：

1. 初始聚类中心的选择：选择的初始聚类中心越合理，聚类性能就越好。
2. 聚类数量 k 的选择：如果 k 过小，可能会漏掉一些重要的簇，导致聚类结果不理想；如果 k 过大，可能会将一些相似的数据点分到不同的簇中，也会导致聚类结果不理想。
3. 数据的分布：如果数据点呈现出明显的聚类结构，那么 K-Means 算法的聚类性能就会更好；如果数据点分布较为分散，那么 K-Means 算法的聚类性能就会较差。
4. 距离度量方法的选择：K-Means 算法使用的距离度量方法也会影响聚类性能。常用的距离度量方法有欧几里得距离和曼哈顿距离。

5. 迭代次数的选择：如果迭代次数较少，那么聚类中心可能不会收敛到最优解，导致聚类结果不理想；如果迭代次数过多，那么会增加计算代价，但并不一定能得到更优的聚类结果

6. 聚类形状：K-Means 算法假设聚类具有圆形的形状，因此对于具有非圆形形状的聚类，K-Means 算法的聚类性能可能较差。

2 题目二

谱聚类（Spectral Clustering）是一种常用的聚类算法，它可以将数据点划分到若干个不同的簇中。谱聚类算法的基本思路是：

1. 对数据点之间的相似度进行建模，得到相似矩阵 $\mathbf{W}$。相似矩阵 $\mathbf{W}$ 的元素 w_{ij} 表示数据点 i 和数据点 j 之间的相似度。

2. 对相似矩阵 $\mathbf{W}$ 进行归一化，得到归一化的相似矩阵 $\mathbf{L}$。归一化的相似矩阵 $\mathbf{L}$ 的元素 l_{ij} 表示数据点 i 和数据点 j 之间的归一化相似度。

3. 对归一化的相似矩阵 $\mathbf{L}$ 进行特征分解，得到特征矩阵 $\mathbf{U}$。特征矩阵 $\mathbf{U}$ 的每一列都是一个特征向量，其中特征向量 $\mathbf{u}_i$ 表示数据点 i 在特征空间中的映射。

4. 在特征空间中聚类，将特征向量 $\mathbf{u}_i$ 划分到最近的聚类中心。

5. 输出最终的聚类结果。

谱聚类算法主要用于解决具有非圆形聚类结构的数据。由于 K-Means 算法假设聚类具有圆形形状，因此对于具有非圆形形状的聚类，K-Means 算法的聚类性能可能较差。而谱聚类算法通过将数据映射到特征空间中来解决这个问题，因此它在处理具有非圆形聚类结构的数据时表现较好。

影响聚类性能的因素：

1. 聚类数量：聚类数量的选择会影响聚类结果的质量。如果聚类数量过少，则可能会导致某些数据点无法被正确划分到聚类中；如果聚类数量过多，则可能会导致某些数据点被不必要地划分到多个聚类中。

2. 相似度模型：聚类算法依赖相似度模型来建模数据点之间的相似度。如果相似度模型不能准确地反映数据点之间的相似关系，则可能会导致聚类性能较差。

3. 数据分布：聚类算法的性能受到数据分布的影响。如果数据具有较复杂的分布结构，则可能会导致聚类算法的性能较差。

4. 噪声：如果数据中存在大量噪声，则可能会导致聚类算法的性能较差。

5. 聚类形状：聚类的形状也会影响聚类算法的性能。如果聚类具有非圆形形状，则 K-Means 算法的性能可能较差，而谱聚类算法的性能可能较好。

3 题目三

Hard-Margin SVM 的优化目标是在保证正确分类的同时使得分隔超平面与最近的样本之间的间隔最大。

设训练数据集为 $(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)$，其中 x_i 为 i 号样本的特征向量，y_i 为 i 号样本的标签（y_i 为 1 或 -1）。SVM 的目标是找到一个分隔超平面（也称为决策边界）$w \cdot x + b = 0$，使得超平面与最近的样本之间的间隔最大。超平面与样本 x_i 之间的间隔为：

$$\frac{|w \cdot x_i + b|}{||w||}$$

在 Hard-Margin SVM 中，我们希望所有样本都被正确分类，即对于所有 i，有：

$$y_i(w \cdot x_i + b) \geq 1$$

同时，我们希望超平面与最近的样本之间的间隔最大。因此，Hard-Margin SVM 的优化目标为：

$$\min_{w,b} \frac{1}{2}||w||^2$$

$$s.t.\ y_i(w \cdot x_i + b) \geq 1, i = 1, 2, ..., m$$

这是一个凸二次规划问题，要求出最优解，我们可以使用拉格朗日乘数法。

4 题目四

Hinge Loss 在 SVM 中的意义是作为损失函数用于衡量超平面分类的准确性。SVM 中的 Hinge Loss 定义为：

$$L = \frac{1}{n} \sum_{i=1}^{n} \max(0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b))$$

其中 n 是训练样本数，$\mathbf{w}$ 和 b 分别为超平面的法向量和截距。

Hinge Loss 的意义在于，对于每一个样本 $(\mathbf{x}_i, y_i)$，它会计算超平面与样本之间的间隔，即 $y_i(\mathbf{w}^\top \mathbf{x}_i + b)$。如果这个值大于等于 1，说明样本已经被正确分类，Hinge Loss 的值为 0。如果这个值小于 1，说明样本被错误分类，Hinge Loss 的值就是 $1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)$。

SVM 的目标是在保证所有训练样本都被正确分类的情况下，使得分类超平面与最近的训练样本间的距离最大，也就是间隔最大。因此 Hinge Loss 可以用来衡量超平面的分类准确性。