Pattern Recognition
数据聚类

牛李金梁
201928014629008

2019 年 12 月 5 日

第一部分：问题、计算与证明

1

原理

对混合高斯密度函数参数估计进行适当简化。首先假设各类出现的先验概率相等。其次假设协方差矩阵是已知的。最后假设样本的后验概率是 0-1 近似的，即当 $\boldsymbol{x}_k \in w$，$P(\omega_i|\boldsymbol{x_k},\boldsymbol{\theta}) = 1$，否则 $P(\omega_i|\boldsymbol{x_k},\boldsymbol{\theta}) = 0$。又聚类的类别数 c 是已知的，所以，k-means 聚类简化为混合高斯密度函数中只有均值 $\boldsymbol{\mu}_i$ 未知的情况。

根据样本到聚类中心欧氏距离的平方聚出 k 个类别。用最大似然估计估计每类的均值 $\boldsymbol{\mu} = [\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_c]^T$。对每个 $\boldsymbol{\mu}_i$ 似然函数为 $\ln p(\boldsymbol{x}|\omega_i, \boldsymbol{\mu}_i) = -\ln\left((2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}\right) - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)$。

梯度为 $\nabla_{\boldsymbol{\mu}_i} \ln p(\boldsymbol{x}|\omega_i, \boldsymbol{\mu}_i) = {\boldsymbol{\Sigma}_i}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)$。

则均值 $\boldsymbol{\mu}_i$ 需要满足方程：

$$\sum_{k=1}^{n} p(\omega_i|x_k, \hat{\boldsymbol{\mu}}){\boldsymbol{\Sigma}_i}^{-1}(\boldsymbol{x}-\hat{\boldsymbol{\mu}}_i) = 0, i = 1, 2, \cdots, c$$

经过整理可得

$$\hat{\boldsymbol{\mu}}_i = \frac{\sum\limits_{k=1}^{n} P(\omega_i|\boldsymbol{x}_k, \hat{\boldsymbol{\mu}})\boldsymbol{x}_k}{\sum\limits_{k=1}^{n} P(\omega_i|\boldsymbol{x}_k, \hat{\boldsymbol{\mu}})} = \frac{1}{n_i}\sum_{\boldsymbol{x}_k \in \omega_i} \boldsymbol{x}_k, i = 1, 2 \cdots, c$$

不过，样本 $\boldsymbol{x}_k$ 属于的类别要计算到聚类中心欧氏距离的平方来判定，因此该过程需要迭代进行。

通过迭代得到 c 个高斯成分的均值之后，一这些均值作为 c 个簇的类中心，计算每个样本点到中心的欧式距离，将样本归入距离最近的类，从而完成一次迭代计算。

算法

1： 设定样本个数 n, 聚类类别数 c，随机初始化类中心 $\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_c$。

2： 对样本集进行分类。依据最近类中心原则将样本分类到某个类内。

3： 分类后计算新的类中心 $\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_c$。

4： 检查迭代前后类中心是否相等，若相等则算法结束，返回类中心；否则返回步骤 2 进行下一次迭代。

影响因素

聚类类别个数 c 对于聚类效果影响很大。

由于假设了每个类的先验概率是相等的，因此不均衡的样本、噪声较多的类和野点影响较大。

类的分布形式，如果类中心附近没有样本点，则无法聚出这一类。

2

经典算法

将样本集看成一个图结构，每个样本是图中的一个顶点，有度矩阵 $\boldsymbol{D}$。输入亲和度矩阵 $\boldsymbol{W}$ 和聚类的类别数 k

1： 计算图拉普拉斯矩阵 $\boldsymbol{L}=\boldsymbol{D}-\boldsymbol{W}$。

2： 计算 $\boldsymbol{L}$ 的特征向量，选出其中前 k 小的向量 $\boldsymbol{u}_1,\cdots,\boldsymbol{u}_k$ 组成矩阵 $\boldsymbol{U}=[\boldsymbol{u}_1,\cdots,\boldsymbol{u}_k]\in R^{n\times k}$

3： 把 $\boldsymbol{U}$ 中的每一行看作 1 个数据点 $\boldsymbol{y}_i\in R^k, i=1,2,\cdots,n$

4： 使用 k-means 聚类算法对 $\boldsymbol{y}_i$ 聚成 k 个类别，输出这个结果。

Shi 算法

将样本集看成一个图结构，每个样本是图中的一个顶点，有度矩阵 $\boldsymbol{D}$。输入亲和度矩阵 $\boldsymbol{W}$ 和聚类的类别数 k

1： 计算图拉普拉斯矩阵 $\boldsymbol{L}=\boldsymbol{D}-\boldsymbol{W}$。

2： 根据 $\boldsymbol{Lu}=\lambda\boldsymbol{Du}$ 计算随机游走型拉普拉斯矩阵 $\boldsymbol{L}_{rw}=\boldsymbol{D}^{-1}\boldsymbol{L}$ 的特征向量，选出其中前 k 小的向量 $\boldsymbol{u}_1,\cdots,\boldsymbol{u}_k$ 组成矩阵 $\boldsymbol{U}=[\boldsymbol{u}_1,\cdots,\boldsymbol{u}_k]\in R^{n\times k}$

3： 把 $\boldsymbol{U}$ 中的每一行看作 1 个数据点 $\boldsymbol{y}_i\in R^k, i=1,2,\cdots,n$

4： 使用 k-means 聚类算法对 $\boldsymbol{y}_i$ 聚成 k 个类别，输出这个结果。

Ng 算法

将样本集看成一个图结构，每个样本是图中的一个顶点，有度矩阵 $\boldsymbol{D}$。输入亲和度矩阵 $\boldsymbol{W}$ 和聚类的类别数 k

1： 根据 $\boldsymbol{L}=\boldsymbol{D}-\boldsymbol{W}$ 和 $\boldsymbol{L}_{sym}=\boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2}$ 计算对称型拉普拉斯矩阵 $\boldsymbol{L}_{sym}$。

2： 计算 $\boldsymbol{L}_{sym}$ 的特征向量，选出其中前 k 小的向量 $\boldsymbol{u}_1,\cdots,\boldsymbol{u}_k$ 组成矩阵 $\boldsymbol{U}=[\boldsymbol{u}_1,\cdots,\boldsymbol{u}_k]\in R^{n\times k}$

3： 对 $\boldsymbol{U}$ 进行线性变换得到 $\boldsymbol{T}\in R^{n\times k}$。其中，$t_{ij}=\frac{u_{ij}}{\sqrt{\sum_{m=1}^{n}u_{im}^2}}$，该变换使 $\boldsymbol{T}$ 的每行代数和为 1。

4： 把 $\boldsymbol{T}$ 中的每一行看作 1 个数据点 $\boldsymbol{y}_i\in R^k, i=1,2,\cdots,n$

5： 使用 k-means 聚类算法对 $\boldsymbol{y}_i$ 聚成 k 个类别，输出这个结果。

影响因素

构建亲和力矩阵 $\boldsymbol{W}$ 时，k 近邻和 ε 半径的选择会有影响。
大型矩阵的特征值分解不稳定。
进行 k-means 聚类的影响因素也会影响谱聚类性能。

3

（1）

$\omega_1 = \{\boldsymbol{x}_1, \boldsymbol{x}_2\}$，$\omega_2 = \{\boldsymbol{x}_3, \boldsymbol{x}_4\}$。
类内散度矩阵 $\boldsymbol{S}_1$、$\boldsymbol{S}_2$ 及总的类内散度矩阵 $\boldsymbol{S}_W$ 为：

$$\boldsymbol{S}_1 = \begin{bmatrix} 4.5 & 1.5 \\ 1.5 & 0.5 \end{bmatrix}, \boldsymbol{S}_2 = \begin{bmatrix} 12.5 & -2.5 \\ -2.5 & 0.5 \end{bmatrix}, \boldsymbol{S}_W = \begin{bmatrix} 17 & -1 \\ -1 & 1 \end{bmatrix}$$

平方误差和为 $J_e = tr\left(\boldsymbol{S}_W\right) = 18$，类内散度矩阵行列式为 $\det\left(\boldsymbol{S}_W\right) = 16$。

（2）

$\omega_1 = \{\boldsymbol{x}_1, \boldsymbol{x}_4\}$，$\omega_2 = \{\boldsymbol{x}_3, \boldsymbol{x}_3\}$。
类内散度矩阵 $\boldsymbol{S}_1$、$\boldsymbol{S}_2$ 及总的类内散度矩阵 $\boldsymbol{S}_W$ 为：

$$\boldsymbol{S}_1 = \begin{bmatrix} 0.5 & -2.5 \\ -2.5 & 12.5 \end{bmatrix}, \boldsymbol{S}_2 = \begin{bmatrix} 0.5 & 1.5 \\ 1.5 & 4.5 \end{bmatrix}, \boldsymbol{S}_W = \begin{bmatrix} 1 & -1 \\ -1 & 17 \end{bmatrix}$$

平方误差和为 $J_e = tr\left(\boldsymbol{S}_W\right) = 18$，类内散度矩阵行列式为 $\det\left(\boldsymbol{S}_W\right) = 16$。

（3）

$\omega_1 = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}$，$\omega_2 = \{\boldsymbol{x}_4\}$。
类内散度矩阵 $\boldsymbol{S}_1$、$\boldsymbol{S}_2$ 及总的类内散度矩阵 $\boldsymbol{S}_W$ 为：

$$\boldsymbol{S}_1 = \begin{bmatrix} 8.67 & 7.33 \\ 7.33 & 8.67 \end{bmatrix}, \boldsymbol{S}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \boldsymbol{S}_W = \begin{bmatrix} 8.67 & 7.33 \\ 7.33 & 8.67 \end{bmatrix}$$

平方误差和为 $J_e = tr\left(\boldsymbol{S}_W\right) = 17.33$，类内散度矩阵行列式为 $\det\left(\boldsymbol{S}_W\right) = 21.33$。
通过（1）（2）（3）中的计算，可以看出：
使用平方误差和最小准则，第三种划分方式最好。
使用类内散度矩阵行列式最小准则，前两种划分方式更好。

第二部分：计算机编程

本部分使用 matlab 2019b 进行编写，其他版本同样可以运行。

1

（1）

首先利用给定的代码生成了随机数组存到了文件 data1.mat 中，后续均采用这个数据。具体的 k-means 算法可以参照第一部分的第一题。运行结果如图。

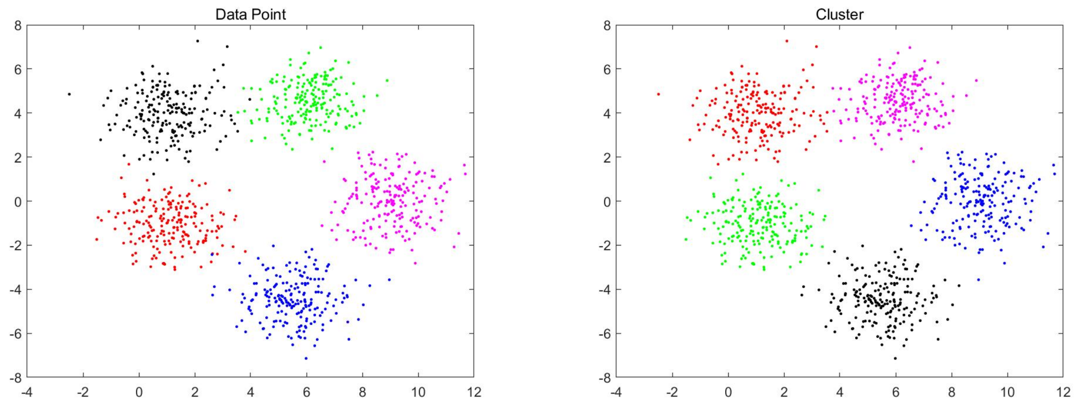

图 1: k-means 算法聚类结果

（2）

采用不同的初值对聚类结果差别很大，如果初始化中心周围样本很少，将有可能得不到聚类结果。现给出一个聚类成功的结果：

- 原始类中心 [1, -1]，聚类的类中心 [1.1088,-1.0682]，样本数量为 200。
- 原始类中心 [5.5, -4.5]，聚类的类中心 [5.5222,-4.4645]，样本数量为 199。
- 原始类中心 [1, 4]，聚类的类中心 [1.0383,3.9929]，样本数量为 200。
- 原始类中心 [6, 4.5]，聚类的类中心 [6.0861,4.5329]，样本数量为 201。
- 原始类中心 [9, 0.0]，聚类的类中心 [9.0563,-0.0286]，样本数量为 200。

均方误差为 [0.0049, 0.0016]。

2

（1）

算法参见第一部分第二题。聚类结果如图 2。

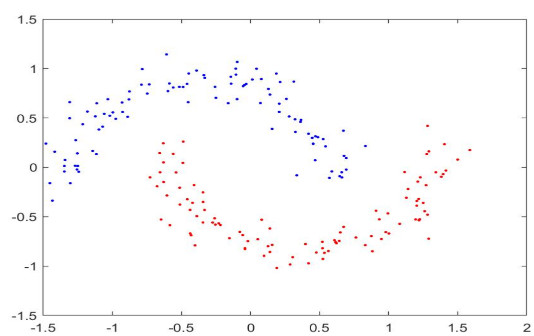

图 2: 聚类结果

（2）

为避免 k-means 算法中初始值的影响，将其固定为 [-0.5, 1] 和 [1, -0.5]。为测试 σ 对准确率的影响，将 k 固定为 3，让 σ 以 0.02 步长从 0.02 增长到 1，画出左图。为测试 k 对准确率的影响，将 σ 固定为 0.5，让 k 以 1 步长从 1 增长到 10，画出右图。

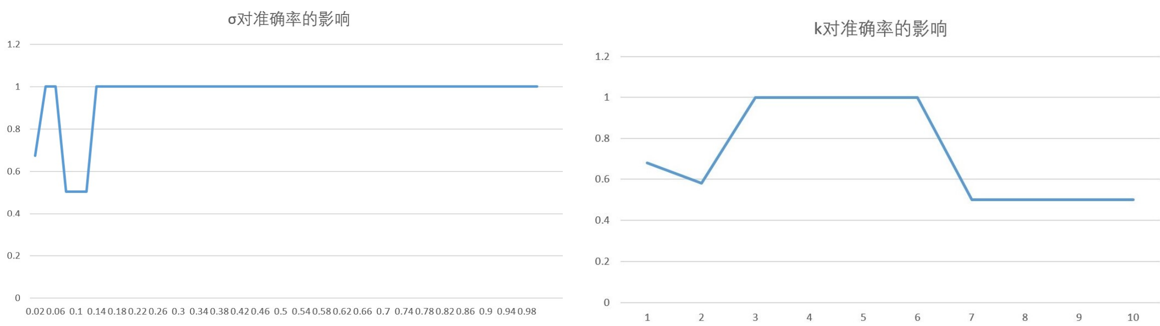

图 3: 谱聚类中参数对准确率的影响