第一次作业

李子龙
上海交通大学
计算机科学与工程系
logcreative@outlook.com

1 k-mean 算法

证明. 对两步分别证明。

(a) **E 步** 如果将每一个点 x_n 赋予类 k'_n 使得其相对于其他所有的类最近，即

$$\|x_n - \mu_{k'_n}\| = \min_k \|x_n - \mu_k\|$$

就意味着它将比上一次赋予的类 k_n 在现在的这种聚类分布下距离不会增加：

$$\|x_n - \mu_{k'_n}\| \le \|x_n - \mu_{k_n}\|$$

那么在对这个点求和的时候，根据指示函数 r_{nk} 的定义，该项也不会增加：

$$j_n = \sum_{k=1}^{K} r'_{nk}\|x_n - \mu_k\|^2 \le \sum_{k=1}^{K} r_{nk}\|x_n - \mu_k\|^2$$

这里

$$r_{nk} = \begin{cases} 1, & x_n \text{ 属于类 } k; \\ 0, & \text{其他情况.} \end{cases}$$

那么损失函数也不会增加：

$$J(\mu_1, \cdots, \mu_K) = \sum_{n=1}^{N} j_n \tag{1}$$

(b) **M 步** 记每个聚类 k 内的点损失函数贡献值为

$$j_k = \sum_{n=1}^{N} r_{nk}\|x_n - \mu_k\|^2 \tag{2}$$

求和可以交换，损失函数改写为

$$J(\mu_1, \cdots, \mu_K) = \sum_{k=1}^{K} j_k \tag{3}$$

将用引理 1 证明，使用聚类内点的平均点作为新的聚类点将不会增加该项。

引理 1 (距离平方和最小). 当 μ 是所有数据点的均值时，距离平方和

$$f = \sum_{t=1}^{N} \|\mu - x_t\|^2 \tag{4}$$

最小。

证明. 将公式 (4) 改写

$$
\begin{aligned}
f &= \sum_{t=1}^{N} \|\mu - x_t\|^2 \\
&= \sum_{t=1}^{N} (\mu - x_t)^T(\mu - x_t) \\
&= \sum_{t=1}^{N} \left(\mu^T\mu - 2x_t^T\mu + x_t^T x_t\right)
\end{aligned}
$$

对其求导，

$$
\frac{\partial f}{\partial \mu} = \sum_{t=1}^{N}(2\mu^T - 2x_t^T) = 0
$$

$$
\mu = \frac{1}{N}\sum_{t=1}^{N} x_t \tag{5}
$$

公式 (5) 表明当 μ 是所有数据点的均值时，导数为 0，距离平方和最小。 □

由于对于每一类而言，公式 (2) 都不会增加，而类别指示函数 r_{nk} 不会改变，所以损失函数 (3) 不会增加。

□

2 k-mean 与 GMM 之间

解. 为了将 GMM 退化为 k-mean，需要对 GMM 有三个方面的特殊化处理：

$$
\pi_k = \frac{1}{K} \tag{6}
$$

$$
\boldsymbol{\Sigma} = \boldsymbol{I} \tag{7}
$$

$$
p(k|x_n) = \begin{cases} 1, & \text{如果 } k = \arg\max_k \mathcal{N}(x_n|\mu_k, \Sigma_k) \\ 0, & \text{其他情况.} \end{cases} \tag{8}
$$

公式 (6) 是混合权重的归一，公式 (7) 是协方差一致使得其只计算欧氏距离，公式 (8) 是硬赋值只取可能性最大的那个聚类 k。

为了得到其中一个中间变种，这里我们只退化 (6)，使 GMM 中的 $\pi_k = \frac{1}{K}$，就可以得到一个更加一般的带方差项软赋值的 k-mean 算法 1。此时的对数似然值定义为

$$
\ln p(\boldsymbol{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N}\left[-\ln K + \ln\sum_{k=1}^{K}\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right] \tag{9}
$$

优点 这种算法可以很好地拓展 k-mean 算法，使其能够具有方差项（引入高斯分布），并且软赋值可以更好地考虑多个聚类。

缺点 这种算法无疑增加了一定的计算量。

3 k-mean 与 CL

Algorithm 1: 含方差软赋值的 k-mean 算法

Input: 数据点 $\boldsymbol{X} = \boldsymbol{x}_{n}{}_{n=1}^{N}$, 聚类数目 K

Output: 聚类的分类结果 $\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, \quad \forall j \in \mathbb{N} \cap [1, K]$

初始化均值矩阵 $\boldsymbol{\mu}_k$，协方差矩阵 $\boldsymbol{\Sigma}_k$，根据公式 (9) 初始化对数似然值;

repeat

  for $n \leftarrow 1$ *to* N **do**

    for $k \leftarrow 1$ *to* K **do**

      $\gamma_{nk}^{(t)} \leftarrow \frac{\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K}\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)}$; /* E 步 */

  for $k \leftarrow 1$ *to* K **do**

    $\boldsymbol{\mu}_k^{(t+1)} \leftarrow \frac{\sum_{n=1}^{N}\gamma_{nk}^{(t)}\boldsymbol{x}_n}{\sum_{n=1}^{N}\gamma_{nk}}$;

    $\boldsymbol{\Sigma}_k^{(t+1)} \leftarrow \frac{\sum_{n=1}^{N}\gamma_{nk}^{(t)}(\boldsymbol{x}_n-\boldsymbol{\mu}_k^{(t+1)})(\boldsymbol{x}_n-\boldsymbol{\mu}_k^{(t+1)})^T}{\sum_{n=1}^{N}\gamma_{nk}}$; /* M 步 */

until 公式 (9) 中的值没有明显变化;