k-Means and EM Algorithm

K-means

聚类，一种非监督的学习方法。

假设数据 $\{x^{(1)}, \ldots x^{(N)}\}$ 在欧拉空间中，$\mathbf{x}^{(n)} \in \mathbb{R}^D$。每一个数据点都属于 K 个聚类中的一个，相同类中的点最相似，而不同类之间的点不相似。

K-means 的目标：找到聚类的中心 $\{m_k\}_{k=1}^K$，以及 assignment $\{r^{(n)}\}_{n=1}^N$，从而使得每一类所以的数据点 $\{x^{(n)}\}$ 到其属于的聚类中心的距离之和最小。

- Mathematically:

$$\min_{\{\mathbf{m}_k\},\{\mathbf{r}^{(n)}\}} J(\{\mathbf{m}_k\},\{\mathbf{r}^{(n)}\}) = \min_{\{\mathbf{m}_k\},\{\mathbf{r}^{(n)}\}} \sum_{n=1}^{N}\sum_{k=1}^{K} r_k^{(n)} ||\mathbf{m}_k - \mathbf{x}^{(n)}||^2$$

注：assignment $r_k^{(n)}$ 是一个 one-hot (或 1-of-k) encoding 值。$r_k^{(n)} = \mathbb{I}\left[\mathbf{x}^{(n)} \text{ is assigned to cluster } k\right]$，即 $\mathbf{r}^{(n)} = [0,..,1,..,0]^\top$。

- Optimization problem:

$$\min_{\{\mathbf{m}_k\},\{\mathbf{r}^{(n)}\}} \sum_{n=1}^{N} \underbrace{\sum_{k=1}^{K} r_k^{(n)} ||\mathbf{m}_k - \mathbf{x}^{(n)}||^2}_{\substack{\text{distance between } x^{(n)} \\ \text{and its assigned cluster center}}}$$

上面括起来的式子中，虽然进行了 K 次，但实际上只有一个结果是非零的 (一个数据点只能属于一类)：

- E.g. say sample $\mathbf{x}^{(n)}$ is assigned to cluster $k = 3$, then

$$\mathbf{r}^n = [0, 0, 1, 0, \ldots]$$

$$\sum_{k=1}^{K} r_k^{(n)} ||\mathbf{m}_k - \mathbf{x}^{(n)}||^2 = ||\mathbf{m}_3 - \mathbf{x}^{(n)}||^2$$

Alternating Minimization

现在，我们要对 k-means 进行优化：

Optimization problem:

$$\min_{\{\mathbf{m}_k\},\{\mathbf{r}^{(n)}\}} \sum_{n=1}^{N}\sum_{k=1}^{K} r_k^{(n)} ||\mathbf{m}_k - \mathbf{x}^{(n)}||^2$$

如果我们能确定聚类的中心 $\{m_k\}$，那么很容易能为每个点找到最好的 assignment $\{r^{(n)}\}$
。

$$\min_{\mathbf{r}^{(n)}} \sum_{k=1}^{K} r_k^{(n)} ||\mathbf{m}_k - \mathbf{x}^{(n)}||^2$$

- Assign each point to the cluster with the nearest center

$$r_k^{(n)} = \begin{cases} 1 & \text{if } k = \arg\min_j ||\mathbf{x}^{(n)} - \mathbf{m}_j||^2 \\ 0 & \text{otherwise} \end{cases}$$

之后，如果我们确定了 assignment $\{r^{(n)}\}$，那么我们可以根据每个聚类的数据确定最好的聚类中心 $\{m_k\}$。我们可以通过所有属于该聚类的点的坐标，来确定最好的聚类中心。

Set each cluster's center to the average of its assigned data points:
For $l = 1, 2, ..., K$

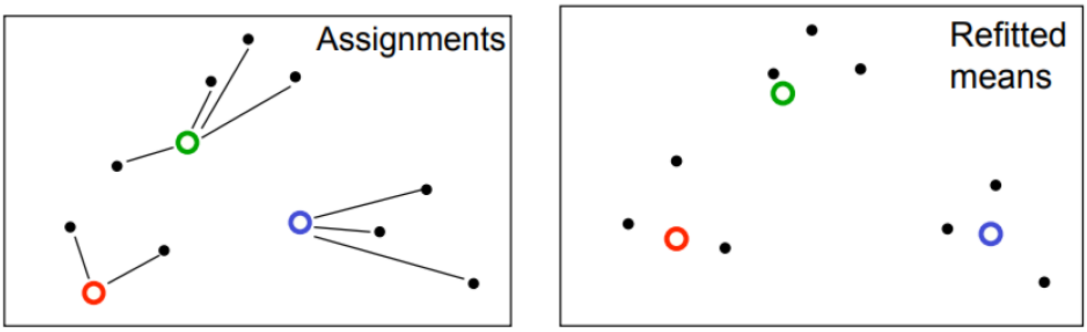
$$\begin{aligned} 0 =& \frac{\partial}{\partial \mathbf{m}_l} \sum_{n=1}^{N}\sum_{k=1}^{K} r_k^{(n)} ||\mathbf{m}_k - \mathbf{x}^{(n)}||^2 \\ =& 2\sum_{n=1}^{N} r_l^{(n)} (\mathbf{m}_l - \mathbf{x}^{(n)}) \quad \Longrightarrow \quad \mathbf{m}_l = \frac{\sum_n r_l^{(n)} \mathbf{x}^{(n)}}{\sum_n r_l^{(n)}} \end{aligned}$$

我们重复这样来确定聚类中心和 assignment，这就叫 alternating minimization。

K-means Algorithm

High level overview of algorithm:

- Initialization: randomly initialize cluster centers
- The algorithm iteratively alternates between two steps:
  - Assignment step: Assign each data point to the closest cluster
  - Refitting step: Move each cluster center to the mean of the data assigned to it



- Initialization: Set K cluster means $\mathbf{m}_1, \ldots, \mathbf{m}_K$ to random values
- Repeat until convergence (until assignments do not change):
  - Assignment: Optimize J w.r.t. $\{\mathbf{r}\}$: Each data point $\mathbf{x}^{(n)}$ assigned to nearest center

$$\hat{k}^{(n)} = \arg\min_k ||\mathbf{m}_k - \mathbf{x}^{(n)}||^2$$

and Responsibilities (1-hot or 1-of-K encoding)

$$r_k^{(n)} = \mathbb{I}[\hat{k}^{(n)} = k] \quad \text{for} \quad k = 1, .., K$$

  - Refitting: Optimize J w.r.t. $\{\mathbf{m}\}$: Each center is set to mean of data assigned to it

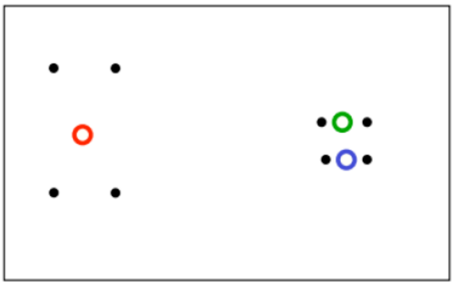
$$\mathbf{m}_k = \frac{\sum_n r_k^{(n)} \mathbf{x}^{(n)}}{\sum_n r_k^{(n)}}.$$

注：assignment step 就是确定点是哪个类；refitting step 就是确定新的聚类中心在哪。

k-means 的每一次迭代，都会使类内点到中心的总距离 J 变小。当中心不再变化时，k-means 便收敛了。

Local Minima

由于 J 是非凸函数，因此我们不能保证一定有最好的结果。k-means 可能被困在局部最小值中。



Soft K-means

相对于 hard assignment，我们可以使用 soft assignment，即让一个点可能属于多个聚类 (比如有 0.7 属于某类，有 0.3 属于另一类)。这样我们在 refitting step 时可以使用更多的点的信息。

- Initialization: Set K means $\{\mathbf{m}_k\}$ to random values
- Repeat until convergence (measured by how much J changes):
  - Assignment: Each data point n given soft "degree of assignment" to each cluster mean k, based on responsibilities

$$r_k^{(n)} = \frac{\exp[-\beta\|\mathbf{m}_k - \mathbf{x}^{(n)}\|^2]}{\sum_j \exp[-\beta\|\mathbf{m}_j - \mathbf{x}^{(n)}\|^2]}$$

$$\Longrightarrow \mathbf{r}^{(n)} = \text{softmax}(-\beta\{\|\mathbf{m}_k - \mathbf{x}^{(n)}\|^2\}_{k=1}^K)$$

注：开始和 k-means 一样，随机选择 K 个聚类中心。然后对于每一个点，求它和每个聚类中心的距离，然后用 softmax 给出该点属于每个聚类的概率 (即权重)。现在的 $r^{(n)}$ 由 K 个权重组成。

- Refitting: Model parameters, means, are adjusted to match sample means of datapoints they are responsible for:

$$\mathbf{m}_k = \frac{\sum_n r_k^{(n)}\mathbf{x}^{(n)}}{\sum_n r_k^{(n)}}$$

注：现在的 refitting 是每个点都参与，根据权重来计算点的贡献。

The Generative Model

现在我们面论两个问题：

- 怎么设置 β
- 权重和宽度不相等的簇？

我们可以使用 generative model 来解决这些问题。

- We'll be working with the following generative model for data $\mathcal{D}$
- Assume a datapoint $\mathbf{x}$ is generated as follows:
  - Choose a cluster z from $\{1, \ldots, K\}$ such that $p(z = k) = \pi_k$
  - Given z, sample $\mathbf{x}$ from a Gaussian distribution $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_z, \mathbf{I})$
- Can also be written:

$$p(z = k) = \pi_k$$

$$p(\mathbf{x}|z = k) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \mathbf{I})$$

注：上面干了两件事：确定从 K 个聚类中选择一个聚类 z 为 k 的概率为 π_k，以及确定在给定聚类 z 后生成 z 中的数据 x 的概率。

- This defines joint distribution $p(z, \mathbf{x}) = p(z)p(\mathbf{x}|z)$ with parameters $\{\pi_k, \boldsymbol{\mu}_k\}_{k=1}^K$

- The marginal of $\mathbf{x}$ is given by $p(\mathbf{x}) = \sum_z p(z, \mathbf{x})$

- $p(z = k|\mathbf{x})$ can be computed using Bayes rule

$$p(z = k|\mathbf{x}) = \frac{p(\mathbf{x} \mid z = k)p(z = k)}{p(\mathbf{x})}$$

  and tells us the probability $\mathbf{x}$ came from the k^{th} cluster

注：上面可以得到 x 属于 k 类的概率。marginal 是说，假如给了头疼的概率 A 和感冒的概率 B，marginal P(A) 就是我不管我感冒不感冒，我其他什么因素都不考虑，我头疼的概率是多少。

- How should we choose the parameters $\{\pi_k, \boldsymbol{\mu}_k\}_{k=1}^{K}$?
- Maximum likelihood principle: choose parameters to maximize likelihood of observed data
- We don't observe the cluster assignments z, we only see the data $\mathbf{x}$
- Given data $\mathcal{D} = \{\mathbf{x}^{(n)}\}_{n=1}^{N}$, choose parameters to maximize:

$$\log p(\mathcal{D}) = \sum_{n=1}^{N} \log p(\mathbf{x}^{(n)})$$

- We can find $p(\mathbf{x})$ by marginalizing out z:

$$p(\mathbf{x}) = \sum_{k=1}^{K} p(z=k, \mathbf{x}) = \sum_{k=1}^{K} p(z=k) p(\mathbf{x}|z=k)$$

What is $p(\mathbf{x})$?

$$p(\mathbf{x}) = \sum_{k=1}^{K} p(z=k) p(\mathbf{x}|z=k) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \mathbf{I})$$

- This distribution is an example of a Gaussian Mixture Model (GMM), and π_k are known as the mixing coefficients
- In general, we would have different covariance for each cluster, i.e., $p(\mathbf{x} \mid z=k) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. For this lecture, we assume $\boldsymbol{\Sigma}_k = \mathbf{I}$ for simplicity.

接下来我们要算出数据 D 的最大似然:

Maximum likelihood objective:

$$\log p(\mathcal{D}) = \sum_{n=1}^{N} \log p(\mathbf{x}^{(n)}) = \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}^{(n)}|\boldsymbol{\mu}_k, \mathbf{I}) \right)$$

和 k-means 一样，如果我们知道每个点 $x^{(n)}$ 对应的聚类 $z^{(n)}$，那么我们很容易得到最大似然。

- **Observation: if we knew** $z^{(n)}$ for every $\mathbf{x}^{(n)}$, (i.e. our dataset was $\mathcal{D}_{\text{complete}} = \{(z^{(n)}, \mathbf{x}^{(n)})\}_{n=1}^{N}$) the maximum likelihood problem is easy:

$$\begin{aligned}
\log p(\mathcal{D}_{\text{complete}}) &= \sum_{n=1}^{N} \log p(z^{(n)}, \mathbf{x}^{(n)}) \\
&= \sum_{n=1}^{N} \log p(\mathbf{x}^{(n)}|z^{(n)}) + \log p(z^{(n)}) \\
&= \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{I}[z^{(n)} = k] \left(\log \mathcal{N}(\mathbf{x}^{(n)}|\boldsymbol{\mu}_k, \mathbf{I}) + \log \pi_k\right)
\end{aligned}$$

接下来我们可以做和朴素贝叶斯中类似的操作:

- By maximizing $\log p(\mathcal{D}_{\text{complete}})$, we would get this:

$$\begin{aligned}
\hat{\boldsymbol{\mu}}_k &= \frac{\sum_{n=1}^{N} \mathbb{I}[z^{(n)} = k]\, \mathbf{x}^{(n)}}{\sum_{n=1}^{N} \mathbb{I}[z^{(n)} = k]} = \text{class means} \\
\hat{\pi}_k &= \frac{1}{N} \sum_{n=1}^{N} \mathbb{I}[z^{(n)} = k] = \text{class proportions}
\end{aligned}$$

注: $\hat{\boldsymbol{\mu}}_k$ 得到的是聚类中所有点的均值; $\hat{\pi}_k$ 得到的是聚类中点的数量(除以总数据量)。

接下来，我们可以计算出 x 属于哪个聚类:

- Conditional probability (using Bayes rule) of z given $\mathbf{x}$

$$\begin{aligned}
p(z = k|\mathbf{x}) &= \frac{p(z = k)p(\mathbf{x}|z = k)}{p(\mathbf{x})} \\
&= \frac{p(z = k)p(\mathbf{x}|z = k)}{\sum_{j=1}^{K} p(z = j)p(\mathbf{x}|z = j)} \\
&= \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \mathbf{I})}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \mathbf{I})}
\end{aligned}$$

然后是另一种求 $\hat{\boldsymbol{\mu}}_k$ 和 $\hat{\pi}_k$ 的方法:

$$\log p(\mathcal{D}_{\text{complete}}) = \sum_{n=1}^{N}\sum_{k=1}^{K} \mathbb{I}[z^{(n)} = k](\log \mathcal{N}(\mathbf{x}^{(n)}|\boldsymbol{\mu}_k, \mathbf{I}) + \log \pi_k)$$

- We don't know the cluster assignments $\mathbb{I}[z^{(n)}=k]$, but we know their expectation $\mathbb{E}[\mathbb{I}[z^{(n)}=k] \mid \mathbf{x}^{(n)}]=p(z^{(n)}=k|\mathbf{x}^{(n)})$.
- If we plug in $r_k^{(n)} = p(z^{(n)} = k|\mathbf{x}^{(n)})$ for $\mathbb{I}[z^{(n)} = k]$, we get:

$$\sum_{n=1}^{N}\sum_{k=1}^{K} r_k^{(n)}(\log \mathcal{N}(\mathbf{x}^{(n)}|\boldsymbol{\mu}_k, \mathbf{I}) + \log \pi_k)$$

- This is still easy to optimize! Solution is similar to what we have seen:

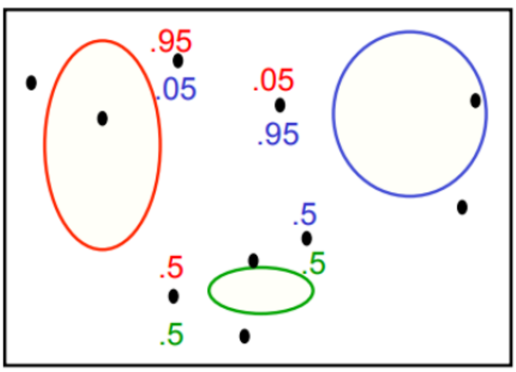
$$\hat{\boldsymbol{\mu}}_k = \frac{\sum_{n=1}^{N} r_k^{(n)}\mathbf{x}^{(n)}}{\sum_{n=1}^{N} r_k^{(n)}} \qquad \hat{\pi}_k = \frac{\sum_{n=1}^{N} r_k^{(n)}}{N}$$

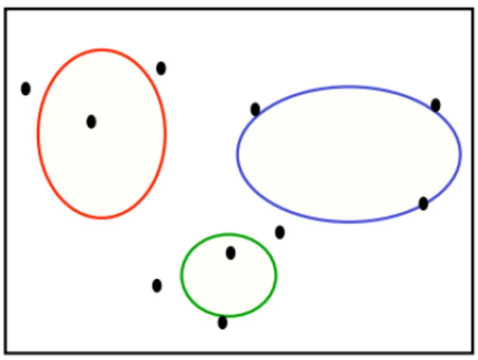
- Note: this only works if we treat $r_k^{(n)} = \frac{\pi_k \mathcal{N}(\mathbf{x}^{(n)}|\boldsymbol{\mu}_k,\mathbf{I})}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}^{(n)}|\boldsymbol{\mu}_j,\mathbf{I})}$ as fixed.

EM Algorithm for GMM

GMM 就是上面那些，现在我们提供一个整体的思路：

- This motivates the Expectation-Maximization algorithm, which alternates between two steps:
  1. E-step: Compute the posterior probabilities $r_k^{(n)} = p(z^{(n)} = k|\mathbf{x}^{(n)})$ given our current model - i.e. how much do we think a cluster is responsible for generating a datapoint.
  2. M-step: Use the equations on the last slide to update the parameters, assuming $r_k^{(n)}$ are held fixed- change the parameters of each Gaussian to maximize the probability that it would generate the data it is currently responsible for.





我们先求出一个后验概率 $r_k^{(n)}$(或 $p(z=k|x)$)，即 k-means 中的 assignment step，得到每个点属于哪一类。之后我们得到参数 $\hat{\boldsymbol{\mu}}_k$ 和 $\hat{\pi}_k$ 来更新 $r_k^{(n)}$，即得到现在每个点属于哪一类。

因为由 E-step 和 M-step 组成，这个方法叫 EM algorithm。

- Initialize the means $\hat{\boldsymbol{\mu}}_k$ and mixing coefficients $\hat{\pi}_k$
- Iterate until convergence:
  - E-step: Evaluate the responsibilities $r_k^{(n)}$ given current parameters

$$r_k^{(n)} = p(z^{(n)}=k|\mathbf{x}^{(n)}) = \frac{\hat{\pi}_k \mathcal{N}(\mathbf{x}^{(n)}|\hat{\boldsymbol{\mu}}_k, \mathbf{I})}{\sum_{j=1}^K \hat{\pi}_j \mathcal{N}(\mathbf{x}^{(n)}|\hat{\boldsymbol{\mu}}_j, \mathbf{I})} = \frac{\hat{\pi}_k \exp\{-\frac{1}{2}\|\mathbf{x}^{(n)} - \hat{\boldsymbol{\mu}}_k\|^2\}}{\sum_{j=1}^K \hat{\pi}_j \exp\{-\frac{1}{2}\|\mathbf{x}^{(n)} - \hat{\boldsymbol{\mu}}_j\|^2\}}$$

  - M-step: Re-estimate the parameters given current responsibilities

$$\begin{aligned} \hat{\boldsymbol{\mu}}_k &= \frac{1}{N_k} \sum_{n=1}^N r_k^{(n)} \mathbf{x}^{(n)} \\ \hat{\pi}_k &= \frac{N_k}{N} \quad \text{with} \quad N_k = \sum_{n=1}^N r_k^{(n)} \end{aligned}$$

  - Evaluate log likelihood and check for convergence

$$\log p(\mathcal{D}) = \sum_{n=1}^N \log \left(\sum_{k=1}^K \hat{\pi}_k \mathcal{N}(\mathbf{x}^{(n)}|\hat{\boldsymbol{\mu}}_k, \mathbf{I}) \right)$$

Review

现在回顾一下我们之前干了什么：

- The maximum likelihood objective $\sum_{n=1}^N \log p(\mathbf{x}^{(n)})$ was hard to optimize
- The complete data likelihood objective was easy to optimize:

$$\sum_{n=1}^N \log p(z^{(n)}, \mathbf{x}^{(n)}) = \sum_{n=1}^N \sum_{k=1}^K \mathbb{I}[z^{(n)} = k](\log \mathcal{N}(\mathbf{x}^{(n)}|\boldsymbol{\mu}_k, \mathbf{I}) + \log \pi_k)$$

由于不知道 $z^{(n)}$，即 $x^{(n)}$ 属于哪一类：

- We don't know $z^{(n)}$'s (they are latent), so we replaced $\mathbb{I}[z^{(n)} = k]$ with responsibilities $r_k^{(n)} = p(z^{(n)} = k|\mathbf{x}^{(n)})$.
- That is: we replaced $\mathbb{I}[z^{(n)} = k]$ with its expectation under $p(z^{(n)}|\mathbf{x}^{(n)})$ (E-step).

- We ended up with the expected complete data log-likelihood:

$$\sum_{n=1}^{N} \mathbb{E}_{p(z^{(n)}|\mathbf{x}^{(n)})}[\log p(z^{(n)}, \mathbf{x}^{(n)})] = \sum_{n=1}^{N} \sum_{k=1}^{K} r_k^{(n)} \left(\log \mathcal{N}(\mathbf{x}^{(n)}|\boldsymbol{\mu}_k, \mathbf{I}) + \log \pi_k \right)$$

  which we maximized over parameters $\{\pi_k, \boldsymbol{\mu}_k\}_k$ (M-step)
- The EM algorithm alternates between:
  - The E-step: computing the $r_k^{(n)} = p(z^{(n)} = k|\mathbf{x}^{(n)})$ (i.e. expectations $\mathbb{E}[\mathbb{I}[z^{(n)} = k]|\mathbf{x}^{(n)}]$) given the current model parameters $\pi_k, \boldsymbol{\mu}_k$
  - The M-step: update the model parameters $\pi_k, \boldsymbol{\mu}_k$ to optimize the expected complete data log-likelihood