《数据挖掘技术》

⭐ CH09 EM算法及其推广

➡️ Created by *Wang JingHui*

➡️ Version: *4.0*

主要内容

1. EM算法的引入
   i. EM算法
   ii. EM算法的导出
   iii. EM算法在非监督学习中的应用
2. EM算法的收敛性
3. EM算法在高斯混合模型学习中的应用
   i. 高斯混合模型
   ii. 高斯混合模型参数估计的EM算法
4. EM算法的推广
   i. F函数的极大极大算法

EM算法

- 是一种迭代算法，1977年由Dempster等人总结提出，用于含有隐变量的概率模型参数的极大似然估计，或极大后验概率，E求期望；M求极大似然估计，简称EM算法

EM 算法的引入

三硬币模型

$$
\begin{aligned}
P(y|\theta) &= \sum_z P(y,z|\theta) \\
&= \sum_z P(z|\theta)P(y|z,\theta) \\
&= \pi p^y(1-p)^{1-y} + (1-\pi)q^y(1-q)^{1-y}
\end{aligned}
$$

1. 随机变量y是观测变量，表示一次试验观测的结果是**1或0**
2. 随机变量z是隐变量，表示未观测到的掷硬币A的结果；
3. $\theta=(\pi,p,q)$是模型参数；
4. 这个模型是**以上数据**(1,1,0,1,0,0,1,0,1,1)的生成模型。

观测数据表示为$Y=(Y_1,Y_2,Y_3,\ldots,Y_n)^T$,

未观测数据表示为$Z=(Z_1,Z_2,Z_3,\ldots,Z_n)^T$,

则观测数据的**似然函数**为

$$P(Y|\theta)=\sum_Z P(Z|\theta)P(Y|Z,\theta)$$

$$P(Y|\theta)=\prod_{j=1}^{n}[\pi p^{y_j}(1-p)^{1-y_j}+(1-\pi)q^{y_j}(1-q)^{1-y_j}]$$

考虑求模型参数$\theta=(\pi,p,q)$的极大似然估计, 即

$$\hat{\theta}=\arg\max_{\theta}\log P(Y|\theta)$$

三硬币模型的EM算法

1.初值

EM算法首选参数初值，记作$\theta^{(0)}=(\pi^{(0)},p^{(0)},q^{(0)})$, 然后迭代计算参数的估计值。

如果第i次迭代的模型参数估计值为$\theta^{(i)}=(\pi^{(i)},p^{(i)},q^{(i)})$

2.E步

那么第$i+1$次迭代的模型参数估计值表示为，计算在模型参数$\pi^{(i)},p^{(i)},q^{(i)}$下观测数据$y_j$来自与$B$的概率，实际估计的是$\pi$

$$\mu_j^{i+1}=\frac{\pi^{(i)}(p^{(i)})^{y_j}(1-p^{(i)})^{1-y_j}}{\pi^{(i)}(p^{(i)})^{y_j}(1-p^{(i)})^{1-y_j}+(1-\pi^{(i)})(q^{(i)})^{y_j}(1-q^{(i)})^{1-y_j}}$$

因为是硬币，只有0，1两种可能，所以有上面的表达。

3.M步

$$\pi^{(i+1)} = \frac{1}{n}\sum_{j=1}^{n}\mu_j^{(i+1)}$$

$$p^{(i+1)} = \frac{\sum_{j=1}^{n}\mu_j^{(i+1)}y_j}{\sum_{j=1}^{n}\mu_j^{(i+1)}}$$

$$q^{(i+1)} = \frac{\sum_{j=1}^{n}(1-\mu_j^{(i+1)})y_j}{\sum_{j=1}^{n}(1-\mu_j^{(i+1)})}$$

初值影响

这个结果说明，如果A是均匀的，那么一个合理的解就是B，C是同质的。他们的分布情况和观测的分布一致。

算法推导

条件:

- 对于 m 个相互独立的样本 $x = (x^{(1)}, x^{(1)}, ..., x^{(m)})$,
- 对应的隐含数据 $z = (z^{(1)}, z^{(1)}, ..., z^{(m)})$,
- 此时 (x, z) 即为完全数据，样本的模型参数为 θ ,
- 观察数据 $x^{(i)}$ 的概率为 $P(x^{(i)}|\theta)$,
- 完全数据 $(x^{(i)}, z^{(i)})$ 的似然函数为 $P(x^{(i)}, z^{(i)}|\theta)$ 。

假如没有隐含变量 z，我们仅需要找到合适的 θ, z 极大化对数似然函数即可：

$$\hat{\theta} = \arg\max_{\theta} L(\theta) = \arg\max_{\theta} \sum_{i=1}^{m} \log P(x^{(i)}|\theta)$$

增加隐含变量 z 之后，目标变成了找到合适的 θ 和 z 让对数似然函数极大：

$$\hat{\theta} = \arg\max_{\theta, z} L(\theta) = \arg\max_{\theta, z} \sum_{i=1}^{m} \log \sum_{z^{(i)}} P(x^{(i)}, z^{(i)}|\theta)$$

相当于$\log(f_1(x) + f_2(x) + ...)$求复合函数的导数。

引入一个未知的新分布$Q_i(z^{(i)})$满足:

$$\sum_z Q_i(z) = 1, 0 \leq Q_i(z) \leq 1$$

引用Jensen不等式(对数函数是凹函数)

$$\log(E(y)) \geq E(\log(y))$$

得到

$$\sum_{i=1}^m \log \sum_{z^{(i)}} P(x^{(i)}, z^{(i)}|\theta) = \sum_{i=1}^m \log \sum_{z^{(i)}} Q_i(z^{(i)}) \frac{P(x^{(i)}, z^{(i)}|\theta)}{Q_i(z^{(i)})} \geq \sum_{i=1}^m \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{P(x^{(i)}, z^{(i)}|\theta)}{Q_i(z^{(i)})}$$

权重$Q_i(z^{(1)})$累积和为1，所以上式是$\log \frac{P(x^{(i)},z^{(i)}|\theta)}{Q_i(z^{(i)})}$的加权平均，也就是**期望**，下一步寻找合适的$Q_i(z)$最优化这个下界。

EM算法

输入: 观测变量数据Y，隐变量数据Z，联合分布$P(Y,Z|\theta)$，条件分布$P(Z|Y,\theta)$

输出: 模型参数θ

> 1. 选择参数的初值$\theta^{(0)}$，开始迭代
> 2. E步：记$\theta^{(i)}$为第 i 次迭代参数θ的估计值，在第$i+1$次迭代的E步，计算
>
> $$\begin{aligned} Q(\theta,\theta^{(i)}) &= E_Z[\log P(Y,Z|\theta)|Y,\theta^{(i)}] \\ &= \sum_Z \log P(Y,Z|\theta)P(Z|Y,\theta^{(i)}) \end{aligned}$$
>
> 3. M步：求使$Q(\theta,\theta^{(i)})$最大化的θ，确定第$i+1$次迭代的参数估计值
>
> $$\theta^{(i+1)} = \arg\max_\theta Q(\theta,\theta^{(i)})$$
>
> 4. 重复步骤（2）和（3）步，直到收敛。

Q 函数

注意Q函数的定义，可以帮助理解上面E步中的求和表达式

完全数据的**对数似然函数**$\log P(Y,Z|\theta)$**关于**给定观测数据Y的当前参数$\theta^{(i)}$下对为观测数据Z的**条件概率分布**$P(Z|Y,\theta^{(i)})$**的期望**称为Q函数。

目标函数

$$L(\theta) = \log P(Y|\theta) = \log \sum_Z P(Y, Z|\theta) = \log(\sum_Z P(Y|Z, \theta)P(Z|\theta))$$

目标函数是不完全数据的对数似然

EM算法导出

$$
\begin{aligned}
L(\theta)-L(\theta^{(i)}) &= \log\left(\sum_Z P(Y|Z,\theta^{(i)})\frac{P(Y|Z,\theta)P(Z|\theta)}{P(Y|Z,\theta^{(i)})}\right)-\log P(Y|\theta^{(i)}) \\
&\geq \sum_Z P(Z|Y,\theta^{(i)})\log\frac{P(Y|Z,\theta)P(Z|\theta)}{P(Z|Y,\theta^{(i)})}-\log P(Y|\theta^{(i)}) \\
&= \sum_Z P(Z|Y,\theta^{(i)})\log\frac{P(Y|Z,\theta)P(Z|\theta)}{P(Z|Y,\theta^{(i)})}-\sum_Z P(Z|Y,\theta^{(i)})\log P(Y|\theta^{(i)}) \\
&= \sum_Z P(Z|Y,\theta^{(i)})\log\frac{P(Y|Z,\theta)P(Z|\theta)}{P(Z|Y,\theta^{(i)})P(Y|\theta^{(i)})}
\end{aligned}
$$

琴声不等式

$$
\log\sum_j \lambda_j y_j \geq \sum_j \lambda_j \log y_j,\ s.t.,\ \lambda_j\geq 0,\ \sum_j\lambda_j=1
$$

令

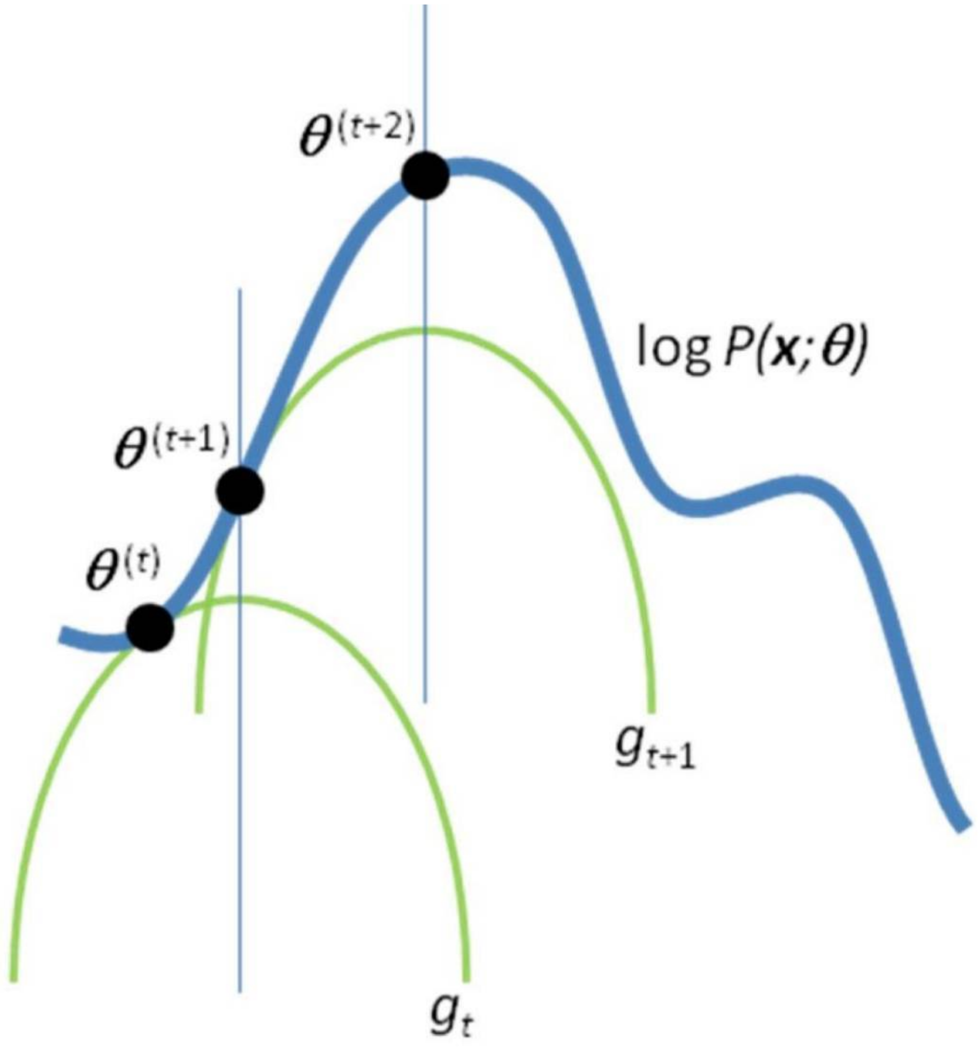
$$B\left(\theta,\theta^{(i)}\right)=L\left(\theta^{(i)}\right)+\sum_{Z}P\left(Z|Y,\theta^{(i)}\right)\log\frac{P\left(Y|Z,\theta\right)P\left(Z|\theta\right)}{P\left(Z|Y,\theta^{(i)}\right)P\left(Y|\theta^{(i)}\right)}$$

则

$$L\left(\theta\right)\geq B\left(\theta,\theta^{(i)}\right)$$

即函$B\left(\theta,\theta^{(i)}\right)$ 是$L(\theta)$ 的一个下界；选择$\theta^{(i)}$使 $B\left(\theta,\theta^{(i)}\right)$达到极大，即

$$
\begin{aligned}
\theta^{(i+1)} &= \arg\max B\left(\theta,\theta^{(i)}\right) \\
&= \arg\max\left(L\left(\theta^{(i)}\right)+\sum_{Z}P\left(Z|Y,\theta^{(i)}\right)\log\frac{P(Y|Z,\theta)P(Z|\theta)}{P\left(Z|Y,\theta^{(i)}\right)P\left(Y|\theta^{(i)}\right)}\right) \\
&= \arg\max\left(\sum_{Z}P\left(Z|Y,\theta^{(i)}\right)\log\left(P(Y|Z,\theta)\right)P(Z|\theta)\right) \\
&= \arg\max\left(\sum_{Z}P\left(Z|Y,\theta^{(i)}\right)\log P(Y,Z|\theta)\right) \\
&= \arg\max Q\left(\theta,\theta^{(i)}\right)
\end{aligned}
$$



总结：极大化观测数据（不完全数据）Y 关于参数θ的对数似然函数

$$L(\theta) \Rightarrow B(\theta, \theta^{(i)}) \Rightarrow \theta^*$$

EM 算法的收敛性

定理保证参数估计序列收敛到对数似然函数序列的稳定性，不能保证收敛到极大值点。

高斯混合模型

混合模型, 有多种, 高斯混合模型是最常用的.

高斯混合模型(Gaussian Mixture Model)是具有如下**概率分布**的模型:

$$P(y|\theta) = \sum_{k=1}^{K} \alpha_k \phi(y|\theta_k)$$

其中, α_k是系数, $\alpha_k \geq 0, \sum_{k=1}^{K} \alpha_k = 1, \phi(y|\theta_k)$ 是**高斯分布密度**, $\theta_k = (\mu, \sigma^2)$

$$\phi(y|\theta_k) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{(y-\mu_k)^2}{2\sigma_k^2}\right)$$

上式表示第k个**分**模型.

注意:

1. GMM的描述是概率分布，形式上可以看成是加权求和
2. 加权求和的权重α满足$\sum_{k=1}^{K}\alpha_k = 1$的约束
3. 求和符号中除去权重的部分，是高斯分布密度(PDF)。高斯混合模型是一种$\sum(\text{权重} \times \text{分布密度}) = \text{分布}$。
4. 高斯混合模型的参数估计是EM算法的一个重要应用，隐马尔科夫模型的非监督学习也是EM算法的一个重要应用。
5. d维的高斯混合模型形式如下，被称作多元正态分布，也叫多元高斯分布，其中，协方差矩阵$\Sigma \in \mathbb{R}^{n\times n}$

$$\phi(y|\theta_k) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}}\exp\left(-\frac{(y-\mu_k)^T\Sigma^{-1}(y-\mu_k)}{2}\right)$$

The probability given in a mixture of *K* Gaussians is:

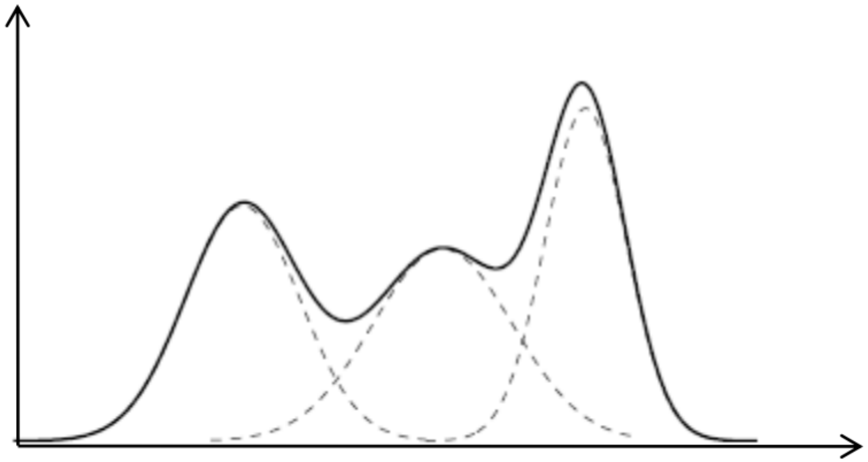
$$p(x) = \sum_{j=1}^{K} w_j \cdot N(x \mid \mu_j, \Sigma_j)$$

where w_j is the prior probability (weight) of the jth Gaussian.

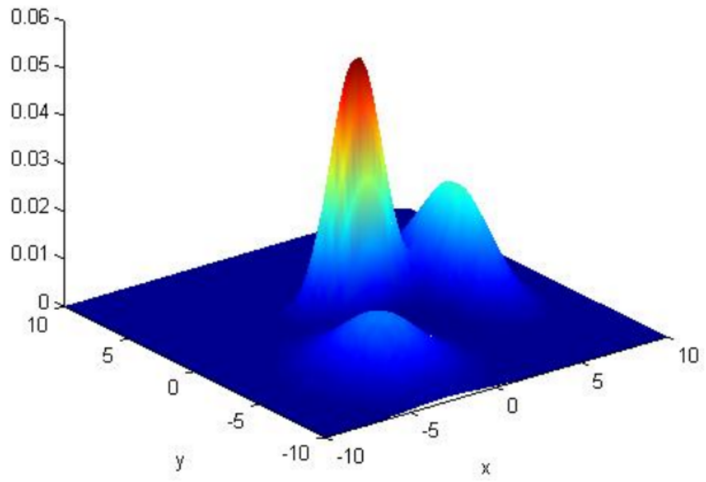
$$\sum_{j=1}^{K} w_j = 1 \qquad \text{and} \qquad 0 \le w_j \le 1$$

Examples:

d=1:



d=2:



Variance-Covariance Matrix

Variance and covariance are often displayed together in a variance-covariance matrix, (aka, a covariance matrix). The variances appear along the diagonal and covariances appear in the off-diagonal elements, as shown below.

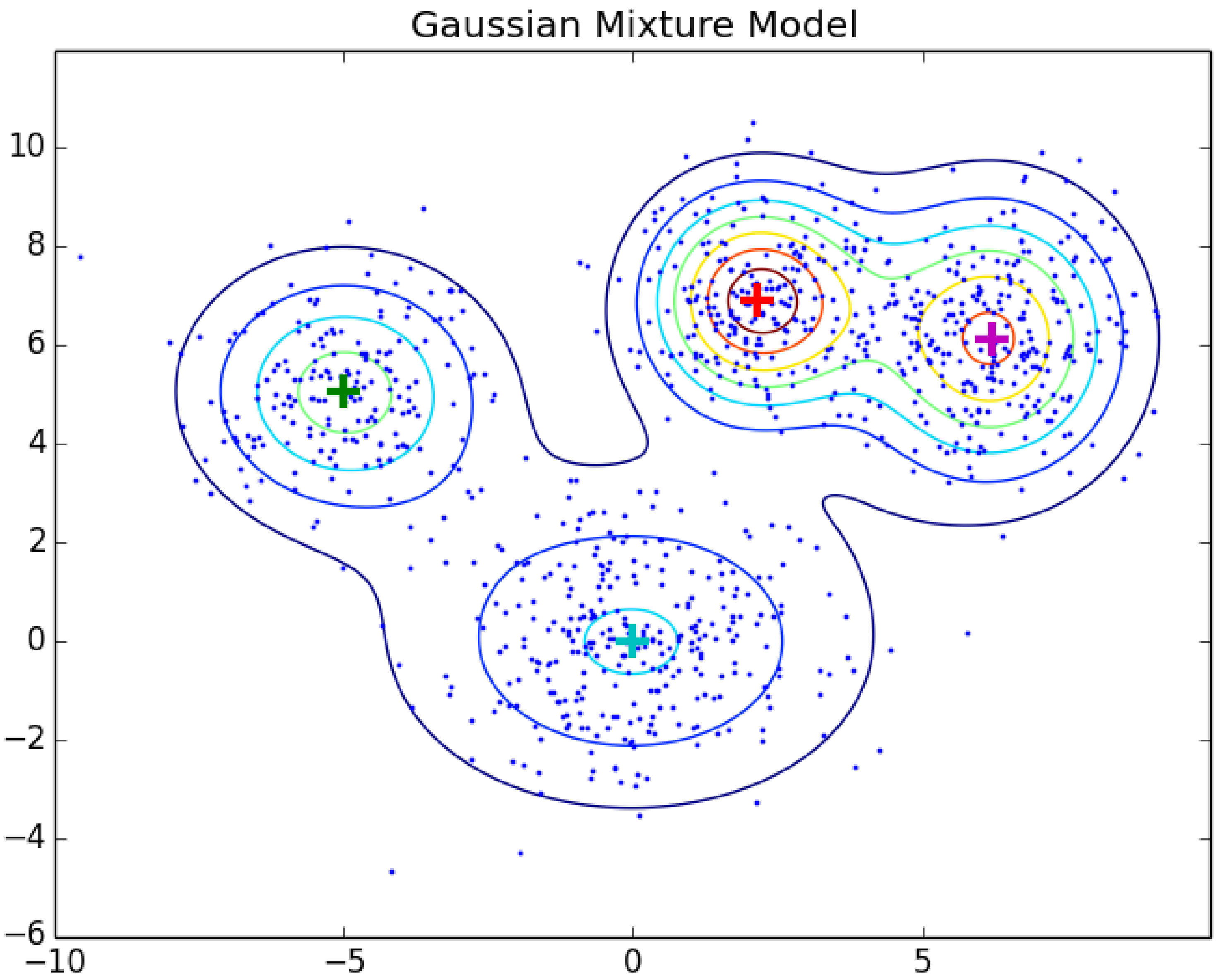
$$\mathbf{V} = \begin{bmatrix} \Sigma x_1^2 / N & \Sigma x_1 x_2 / N & \ldots & \Sigma x_1 x_c / N \\ \Sigma x_2 x_1 / N & \Sigma x_2^2 / N & \ldots & \Sigma x_2 x_c / N \\ \ldots & \ldots & \ldots & \ldots \\ \Sigma x_c x_1 / N & \Sigma x_c x_2 / N & \ldots & \Sigma x_c^2 / N \end{bmatrix}$$

where

V is a c x c variance-covariance matrix
N is the number of scores in each of the c data sets
x_i is a deviation score from the *ith* data set

$\Sigma x_i^2 / N$ is the variance of elements from the *i*th data set

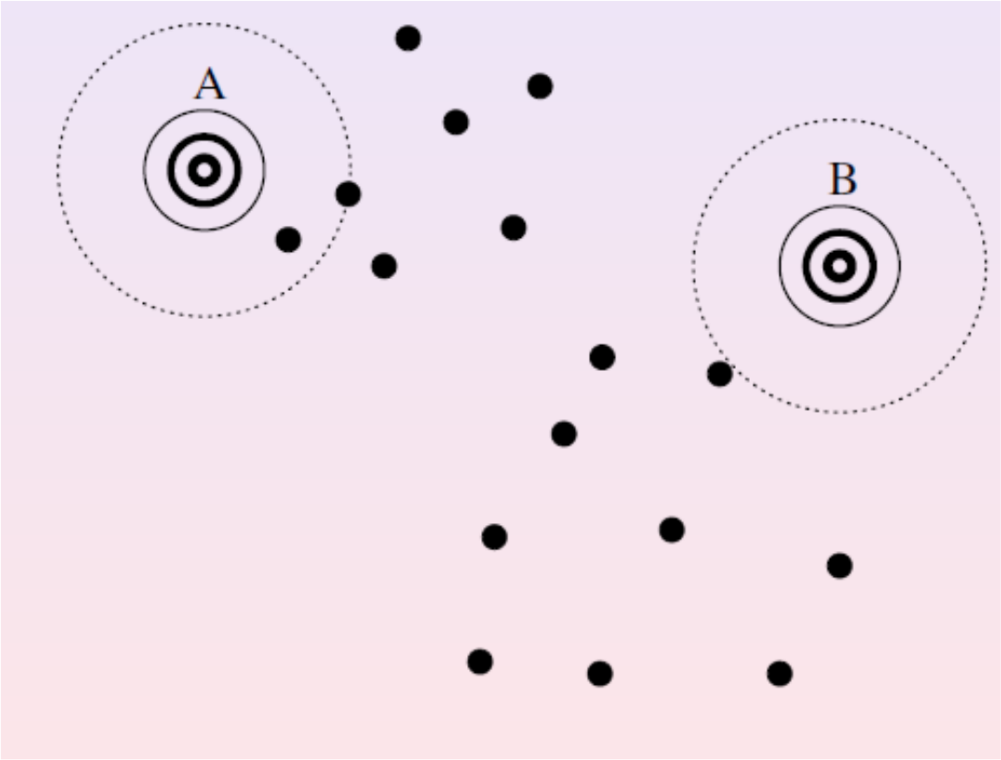
$\Sigma x_i x_j / N$ is the covariance for elements from the *i*th and *j*th data sets



Gaussian Mixture Model
10
8
6
4
2
0
−2
−4
−6
−10
−5
0
5

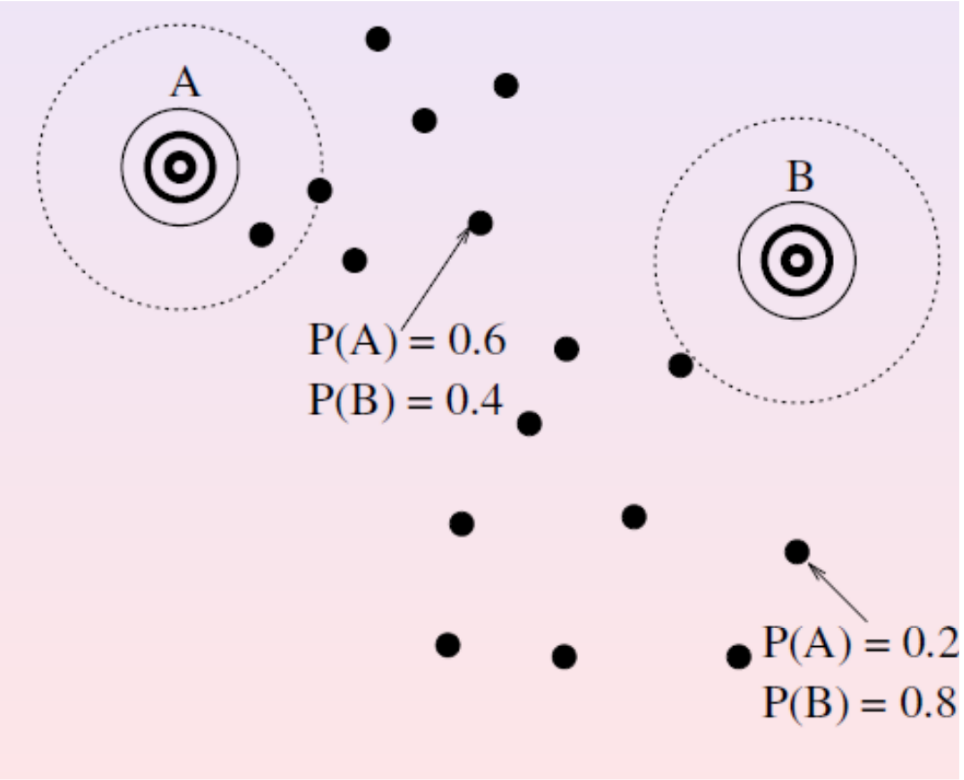
The EM for the GGM (graphical view 1)

Hidden variable: for each point, which Gaussian generated it?



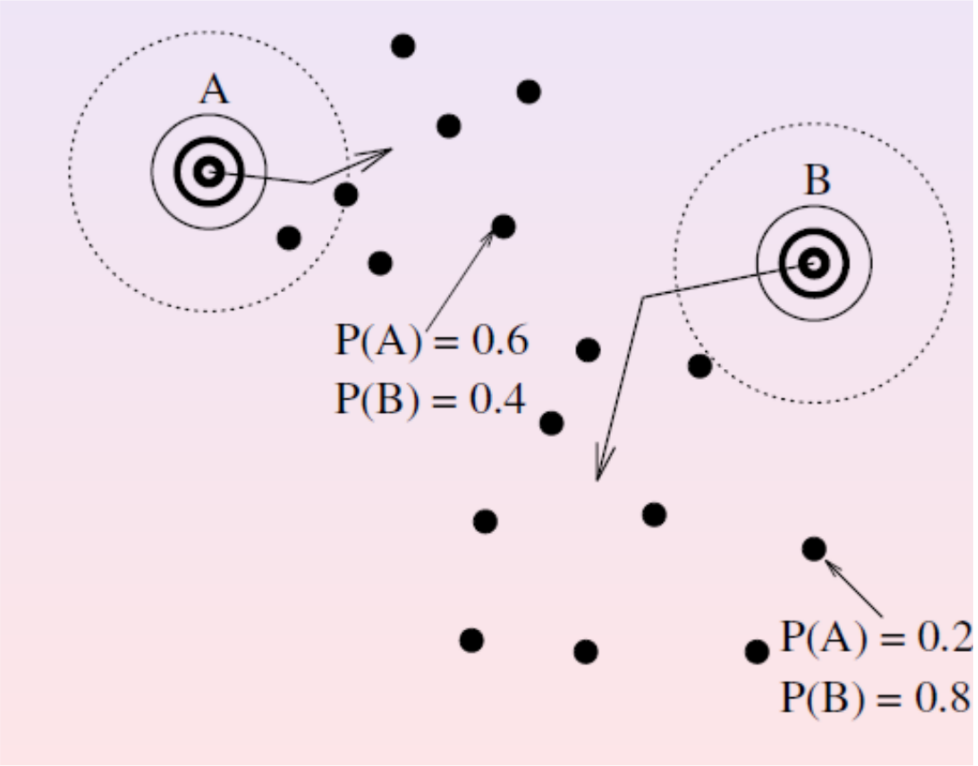
The EM for the GGM (graphical view 2)

E-Step: for each point, estimate the probability that each Gaussian generated it.



The EM for the GGM (graphical view 3)

M-Step: modify the parameters according to the hidden variable to maximize the likelihood of the data (and the hidden variable).



高斯混合模型参数估计的EM算法：

输入：观测数据$y_1, y_2, \cdots, y_N$，高斯混合模型；

输出：高斯混合模型参数

1. 取参数的初始值开始迭代
2. E步：计算分模型k对观测数据y_i的响应度，依据当前参数，计算每个数据 j 来自子模型 k 的可能性。

$$\hat{\gamma}_{jk} = \frac{\alpha_k \phi\left(y|\theta_k\right)}{\sum_{k=1}^{K} \alpha_k \phi\left(y|\theta_k\right)} \qquad j = 1, 2, \cdots, N; k = 1, 2, \cdots, K$$

3. M步：计算新迭代的模型参数

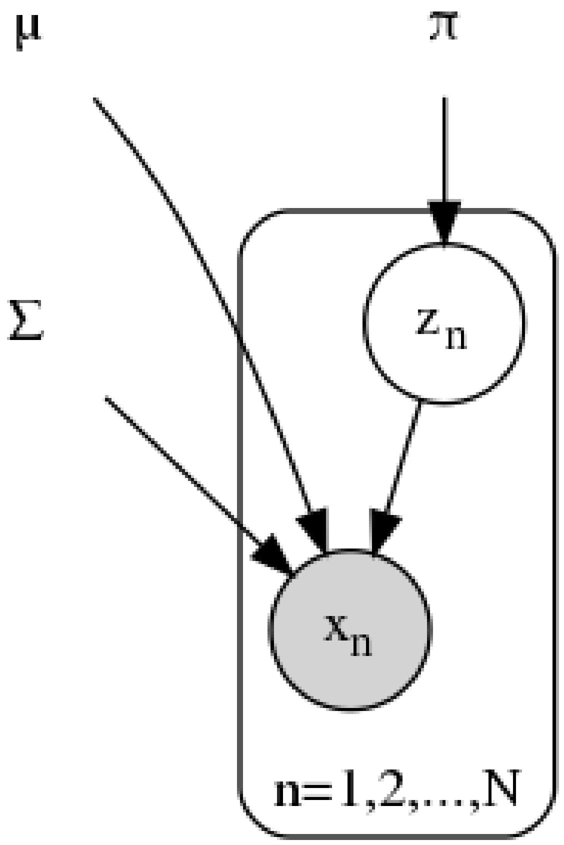
$$\hat{\mu}_k = \frac{\sum_{j=1}^{N} \hat{\gamma}_{jk} y_j}{\sum_{j=1}^{N} \hat{\gamma}_{jk}}, \quad k = 1, 2, \cdots, K$$

$$\hat{\sigma}_k^2 = \frac{\sum_{j=1}^{N} \hat{\gamma}_{jk} \left(y_j - \mu_k\right)^2}{\sum_{j=1}^{N} \hat{\gamma}_{jk}}, \quad k = 1, 2, \cdots, K$$

$$\hat{\alpha}_k = \frac{\sum_{j=1}^{N} \hat{\gamma}_{jk}}{N}, \quad k = 1, 2, \cdots, K$$

4. 重复2.步和3.步，直到收敛。

GMM的图模型



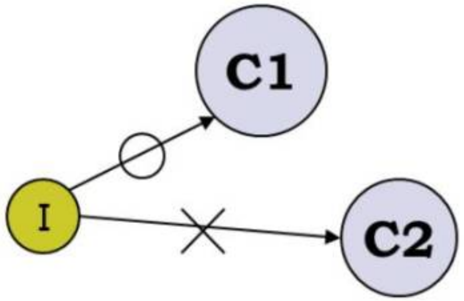
K-means

算法

1. 在Kmeans常见的描述中都有距离的概念，对应了方差，二范数平方等。
2. 每轮刷过距离之后，重新划分样本的分类

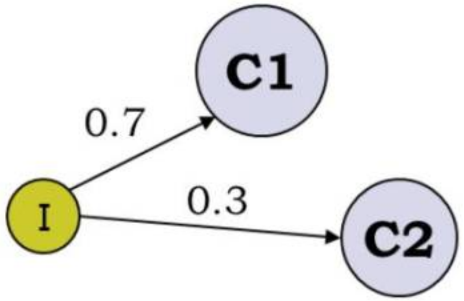
Hard clustering/Soft clustering

k-means algorithm



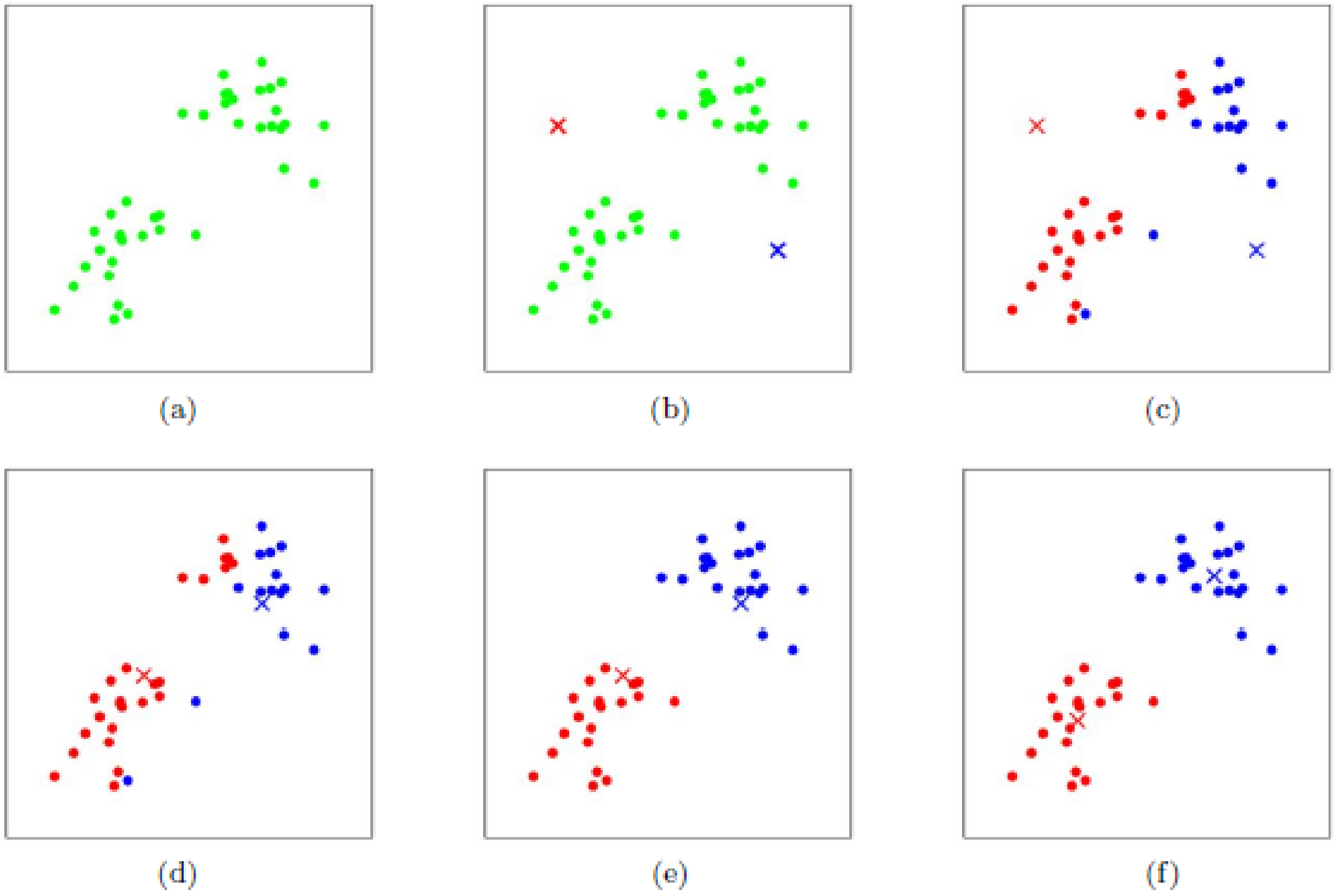
- Hard clustering
  - A instance belong to only one cluster
- Based on Euclidean distance
- Not robust on outlier, value range

EM algorithm

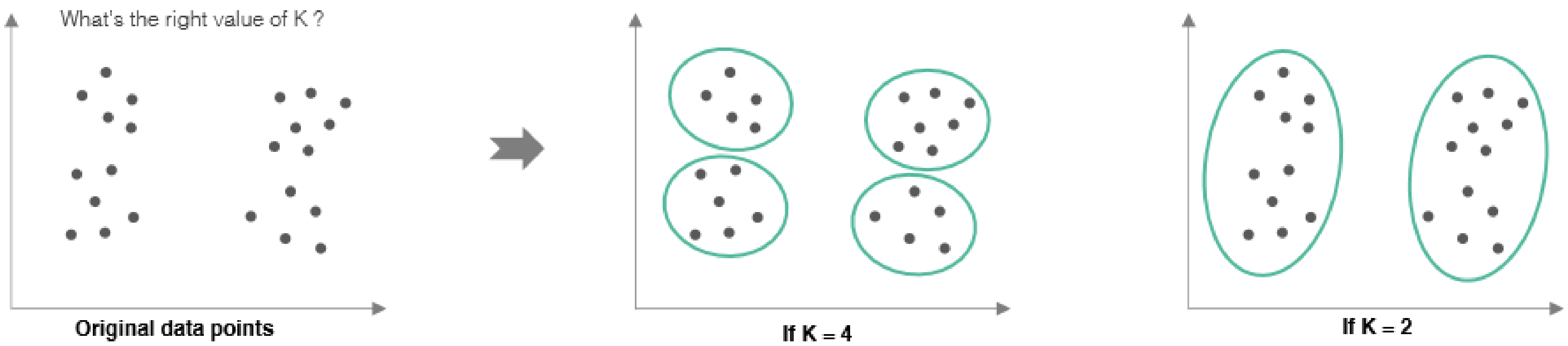


- Soft clustering
  - A instance belong to several clusters with membership pr obability
- Based on density probability
- Can handle both numeric and nominal attributes

K-Means 算法示例



K怎么定



参考

1. [^3]: Maximum-likelihood from incomplete data via the EM algorithm
2. EM Algorithm
3. Sklearn Gaussian Mixed Model
4. [^1]: Gap Statistics
5.
6. mml
7. [^3]: probability and likelihood
8. [^4]: Convex Combination
9. 图片解释(https://zhuanlan.zhihu.com/p/36331115)

↘ Enjoy your machine learning!

https://github.com/wjssx/Statistical-Learning-Slides-Code

E-mail: csr_dsp@sina.com