K-均值算法、GMM算法

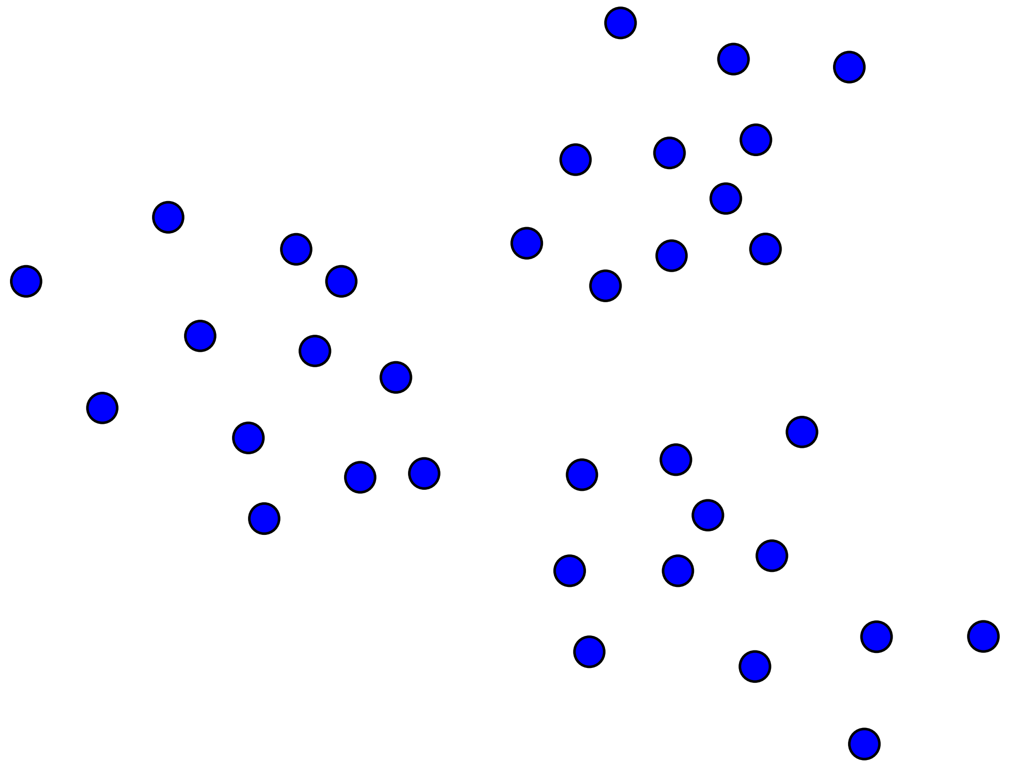
李 波

k-means算法与GMM算法

- k-means算法与GMM算法都属于无监督学习算法.
- 无监督学习算法也被称为聚类(clustering)算法，将无标签数据按照某些标准分成几个簇（cluster）.
- 同一个簇内的样本相似，不同簇的样本不相似.

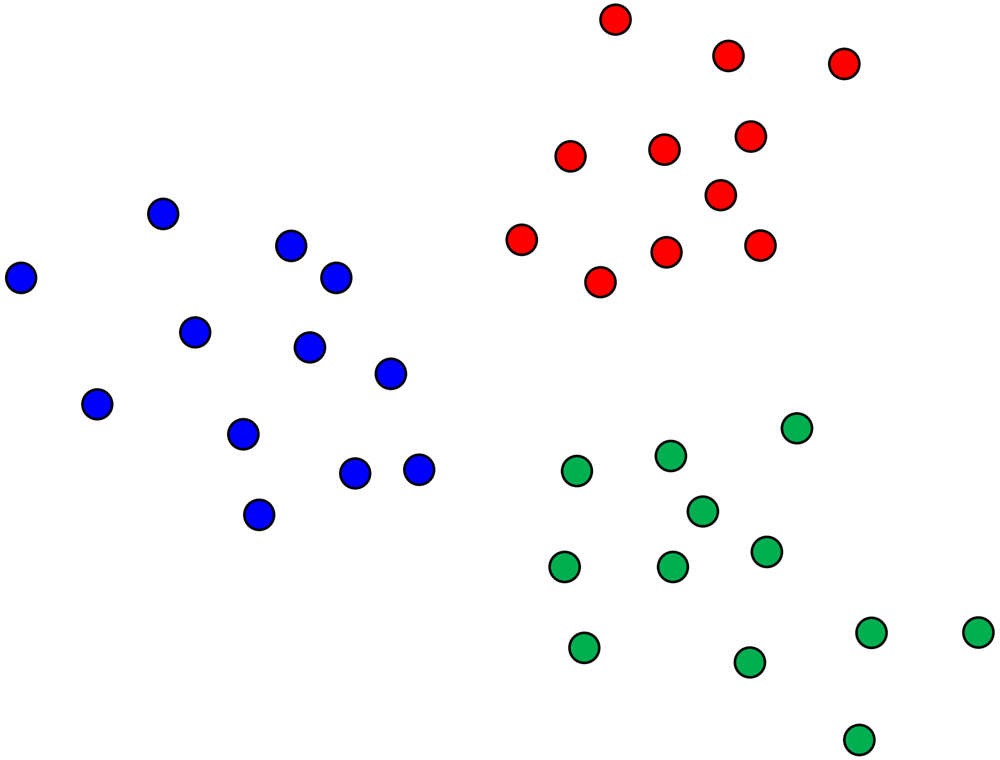
k-means算法与GMM算法

如果想把下面的数据聚类成三个簇，结果是什么?



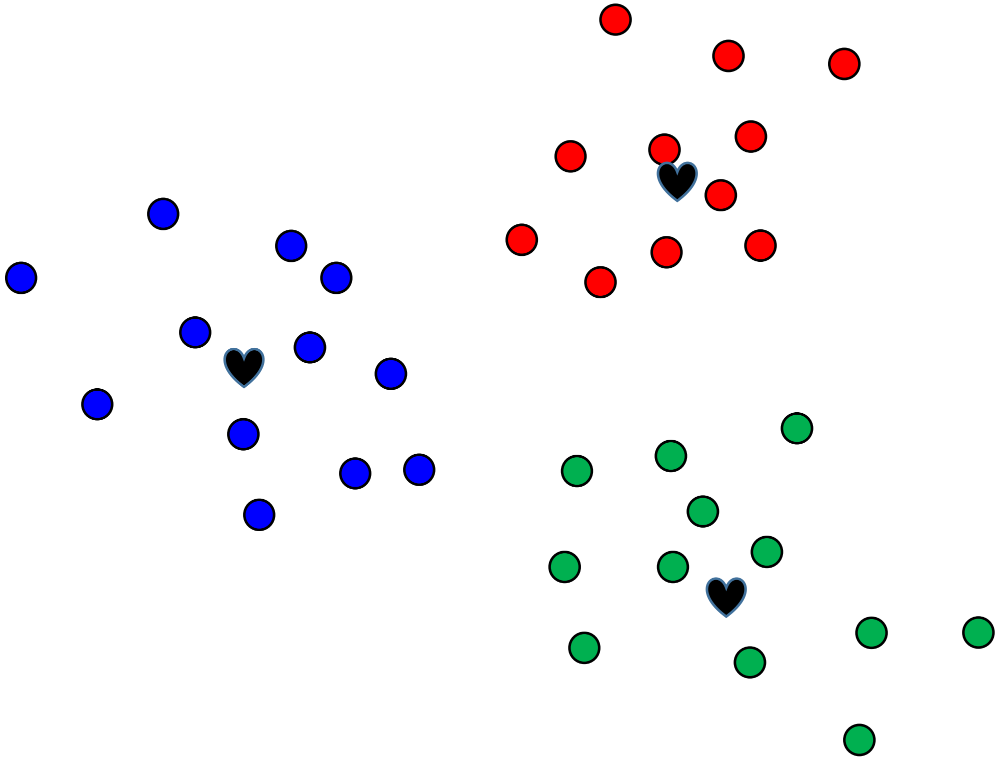
k-means算法与GMM算法

如果想把下面的数据聚类成三个簇，结果是什么?



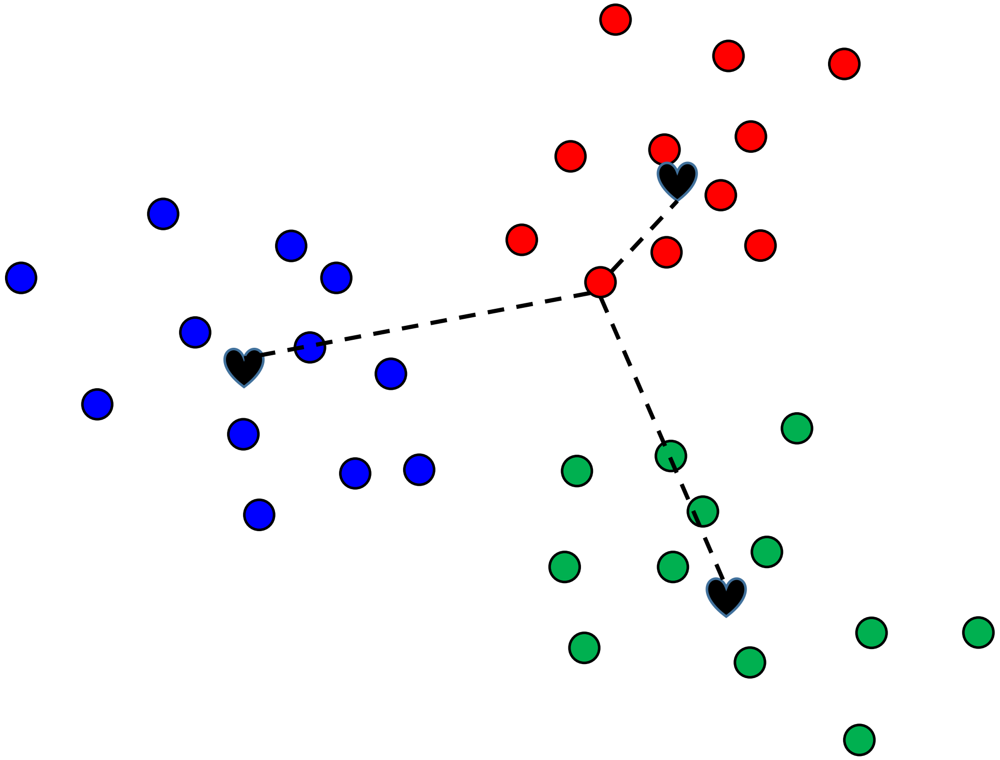
k-means算法与GMM算法

- 每个样本都有一个归属值(membership)，即归属于哪个簇.
- 每个簇都有一个中心.



k-means算法与GMM算法

- 每个样本都有一个归属值(membership)，即归属于哪个簇.
- 每个簇都有一个中心.

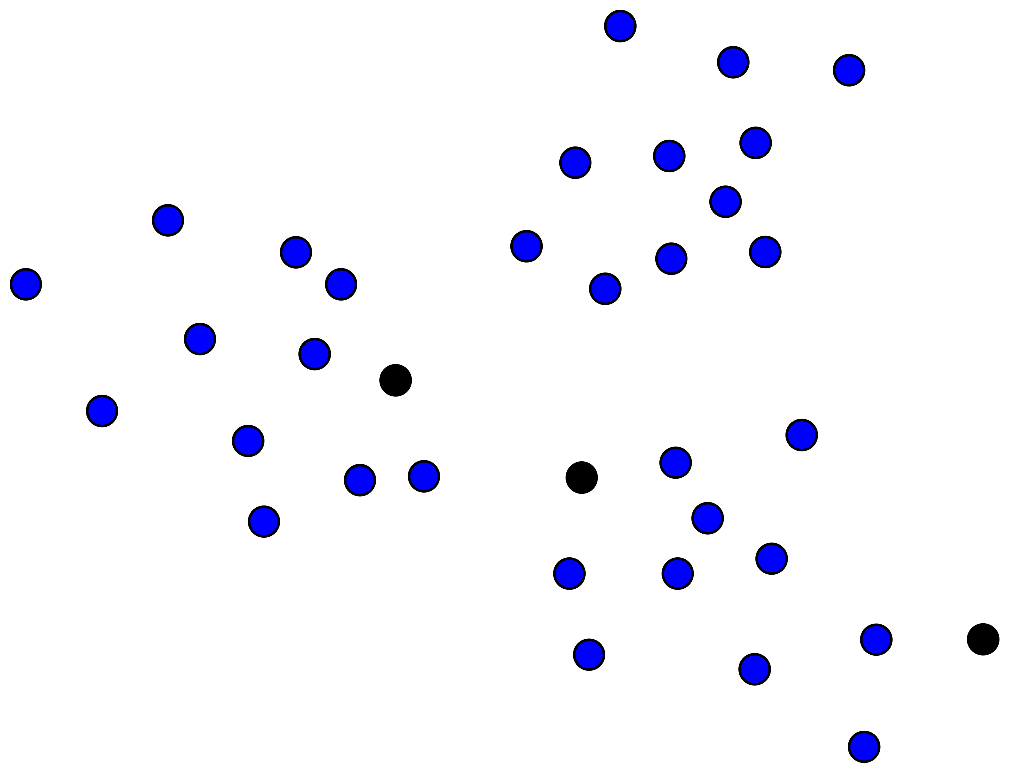


k-means算法与GMM算法

- 在初始化过程，k-means算法随机选择k个样本点作为k个簇的中心点.
- k-means算法是一个迭代算法，每次迭代完成以下两个步骤：
  - 计算每个样本点的归属，即每个样本点归属于哪个簇.
  - 更新每个簇的中心点.

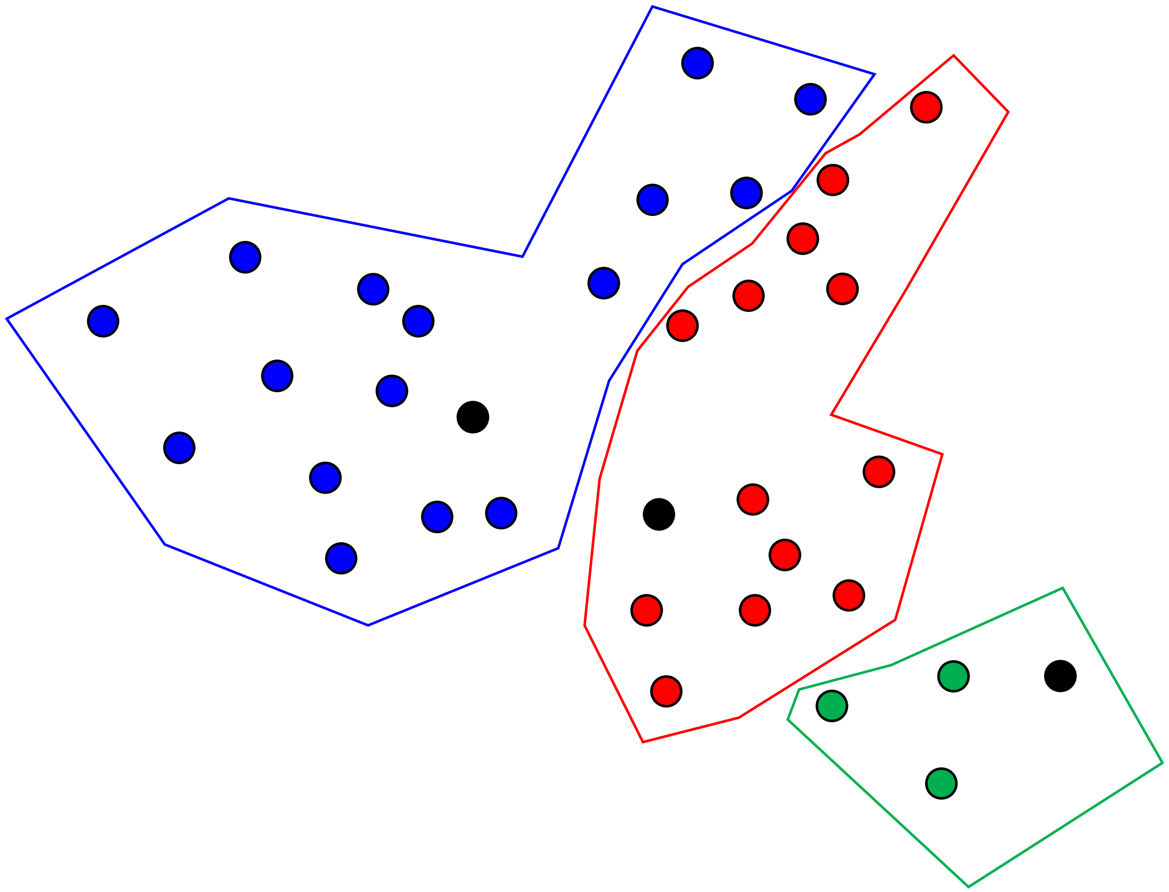
k-means算法与GMM算法

在初始化过程，k-means算法随机选择k个样本点作为k个簇的中心点.



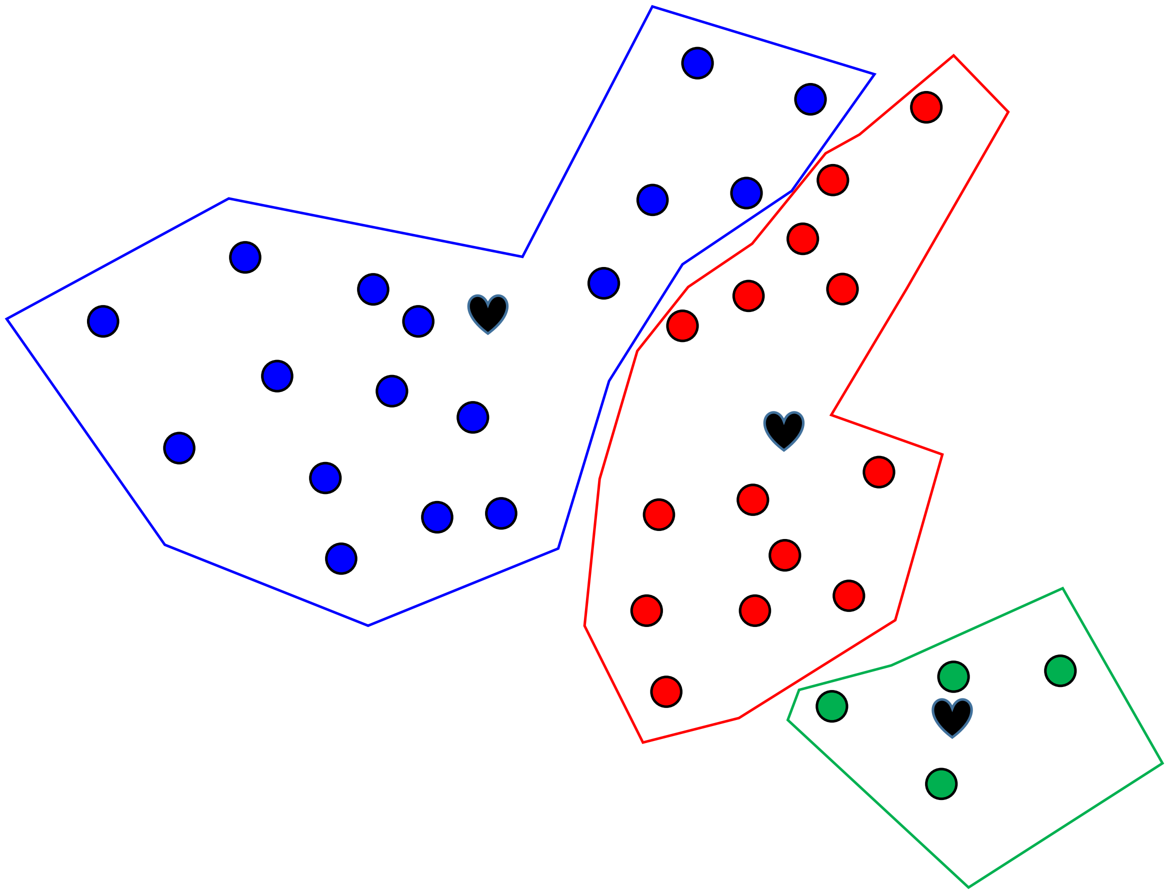
k-means算法与GMM算法

第一次迭代：计算每个样本点的归属值.



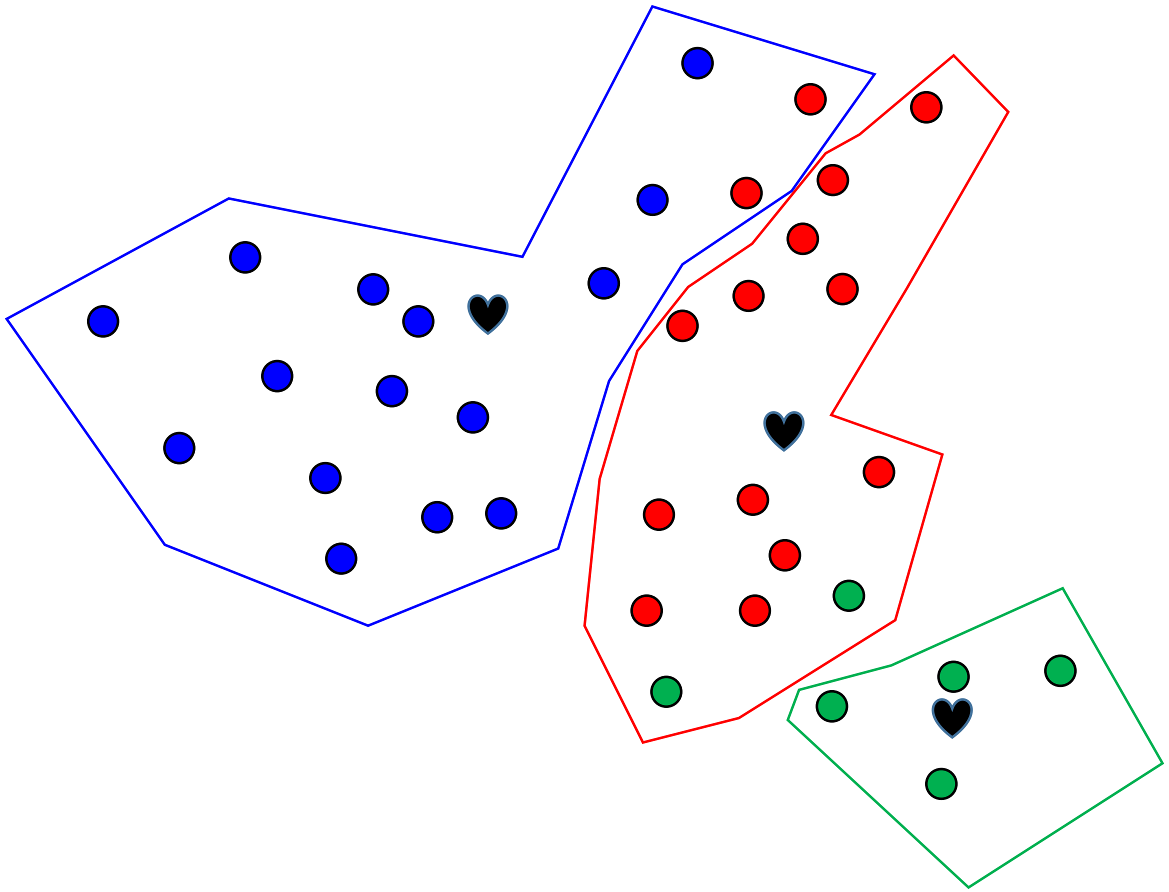
k-means算法与GMM算法

第一次迭代：更新每个簇的中心点.



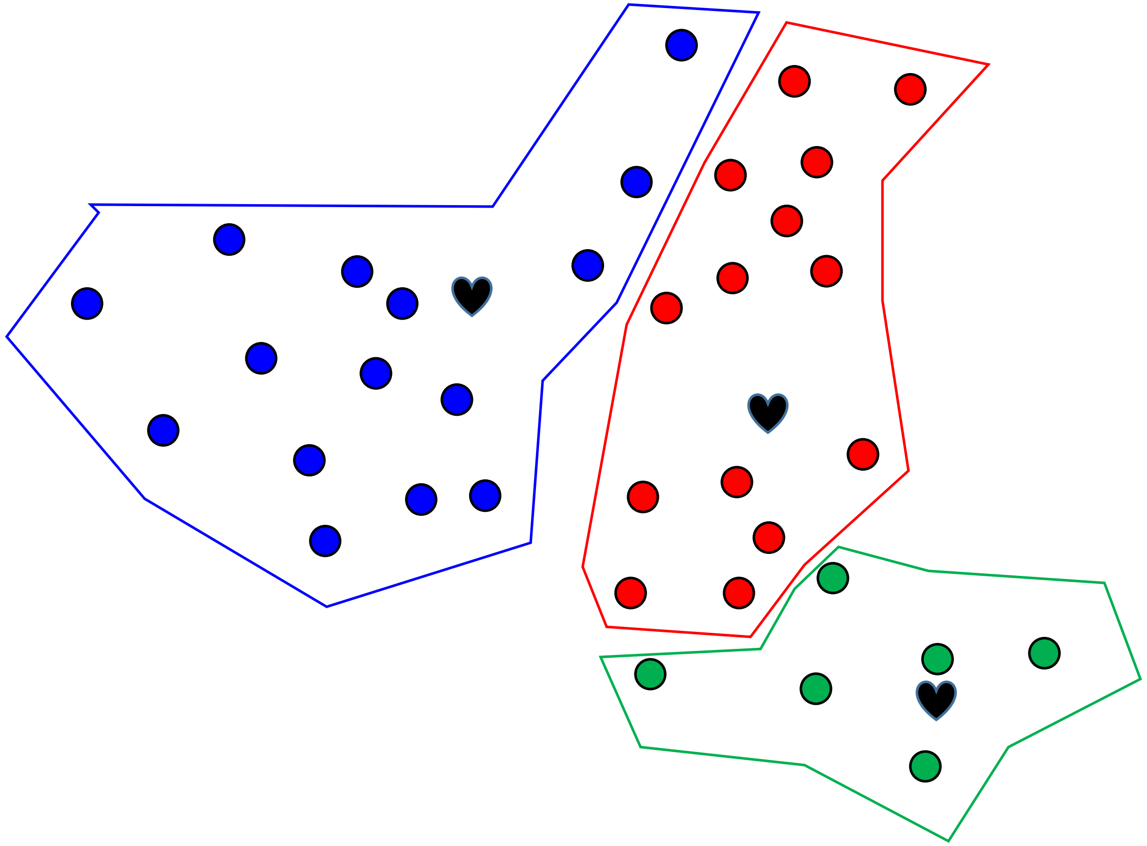
k-means算法与GMM算法

第二次迭代：计算每个样本点的归属值.



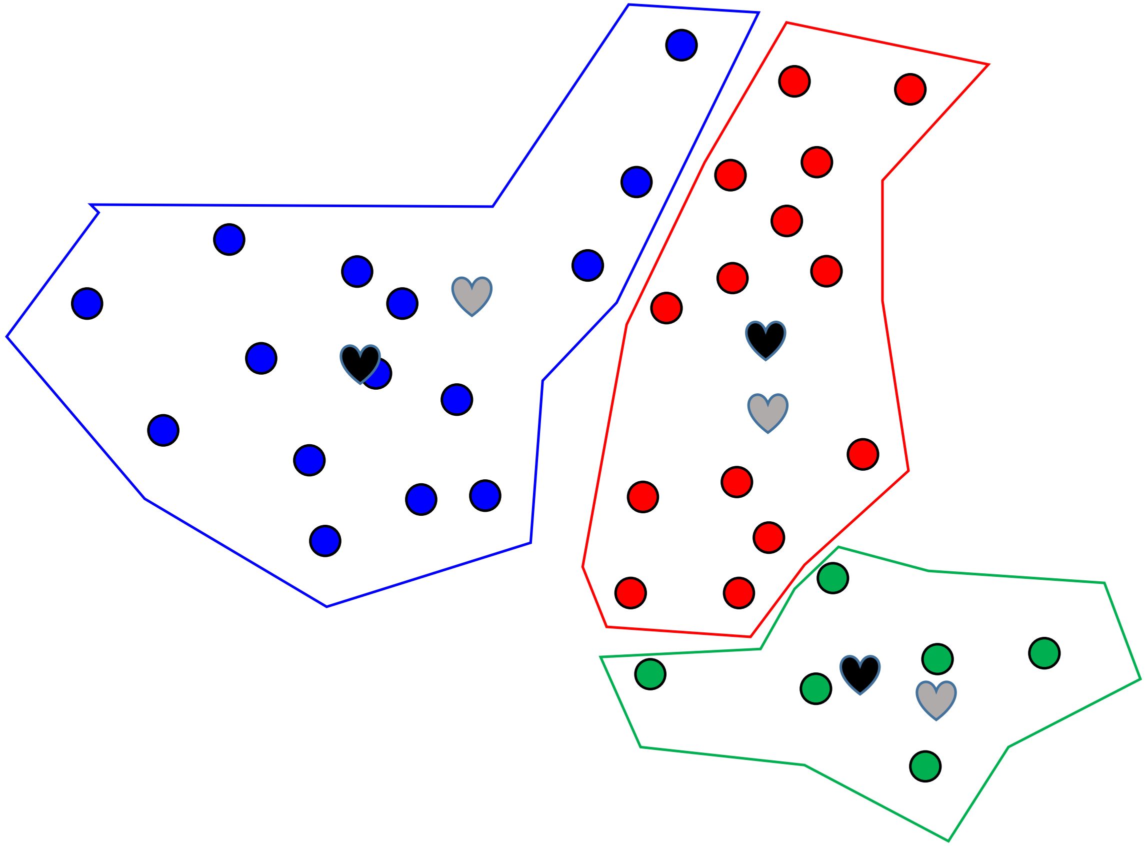
k-means算法与GMM算法

第二次迭代：计算每个样本点的归属值.



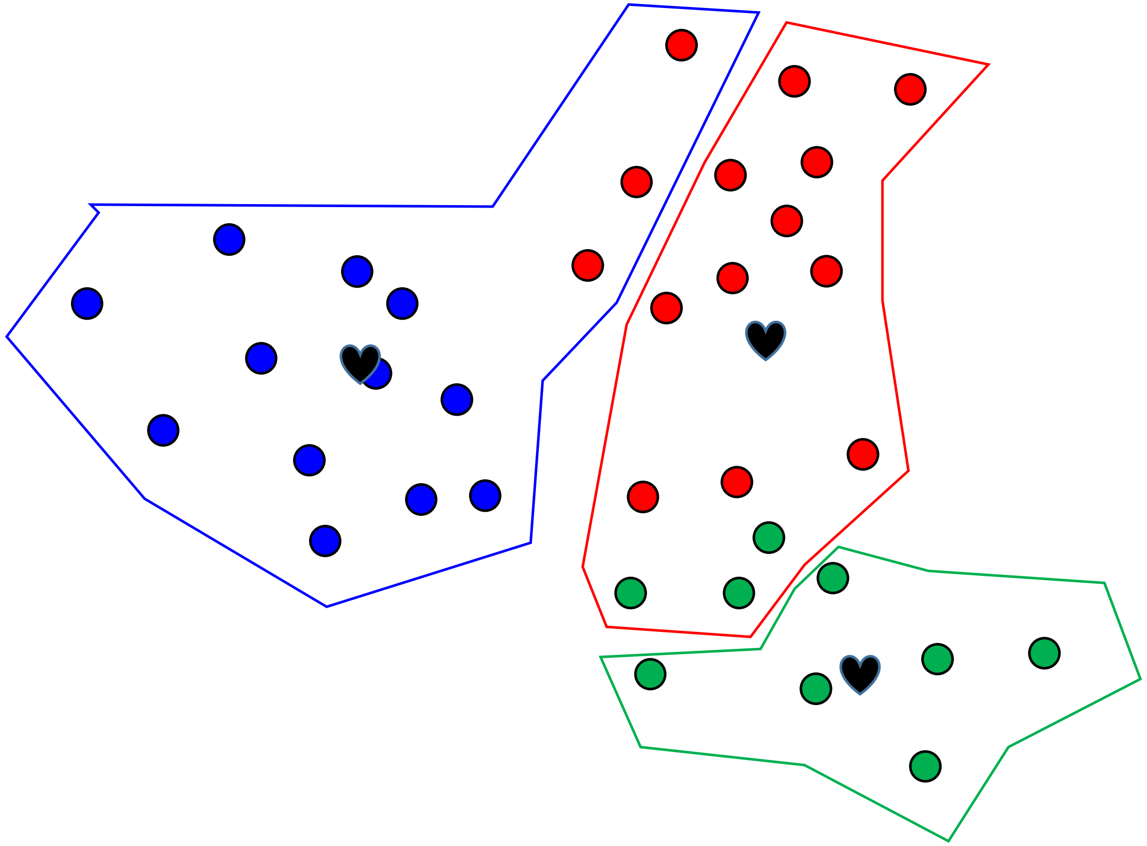
k-means算法与GMM算法

第二次迭代：更新每个簇的中心点.



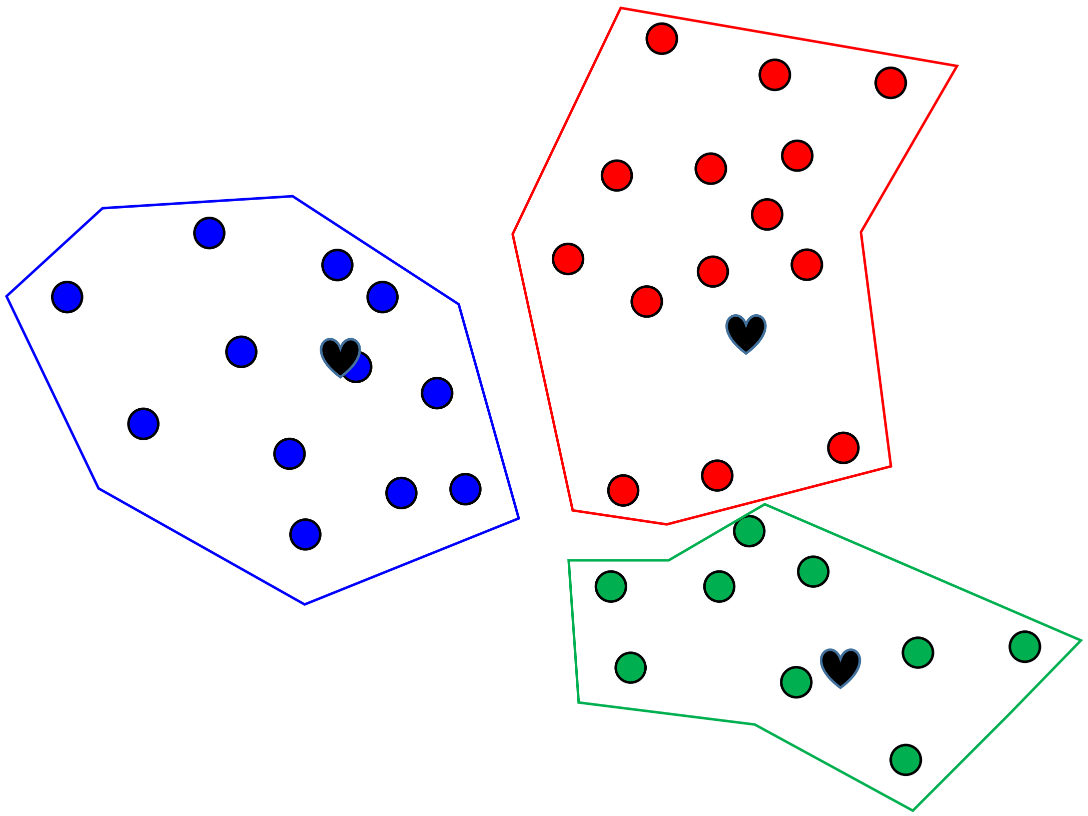
k-means算法与GMM算法

第三次迭代：计算每个样本点的归属值.



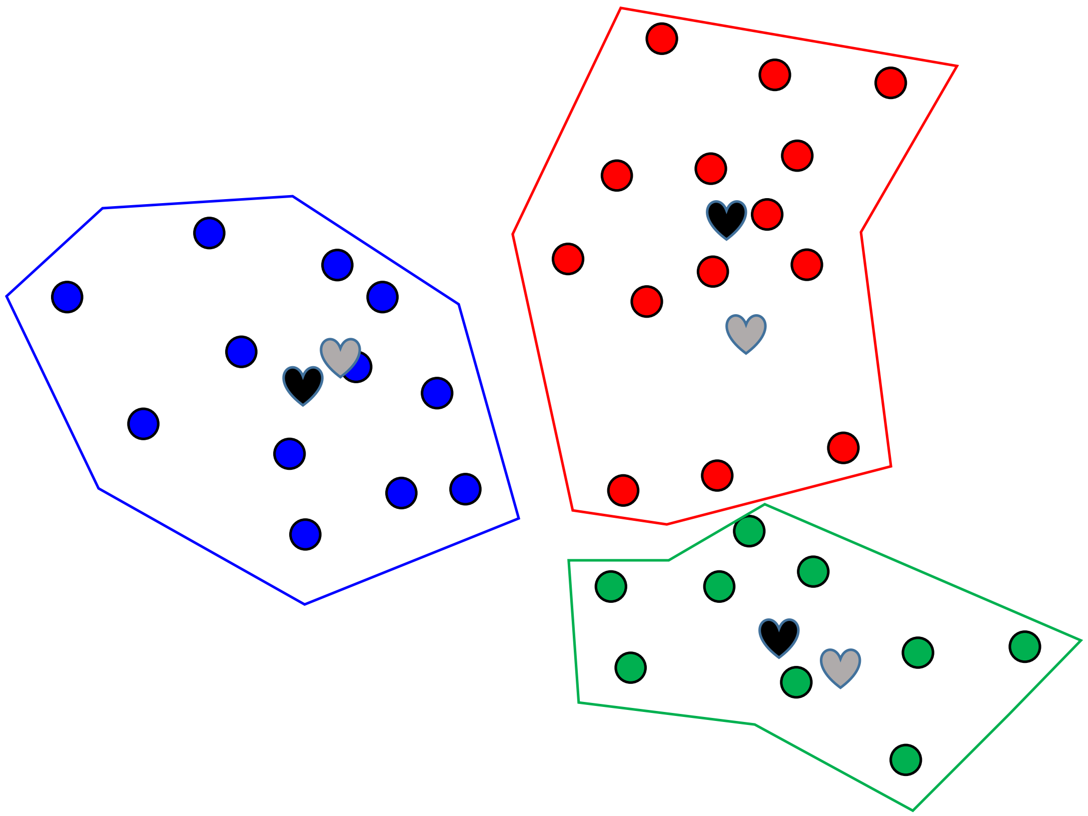
k-means算法与GMM算法

第三次迭代：计算每个样本点的归属值.



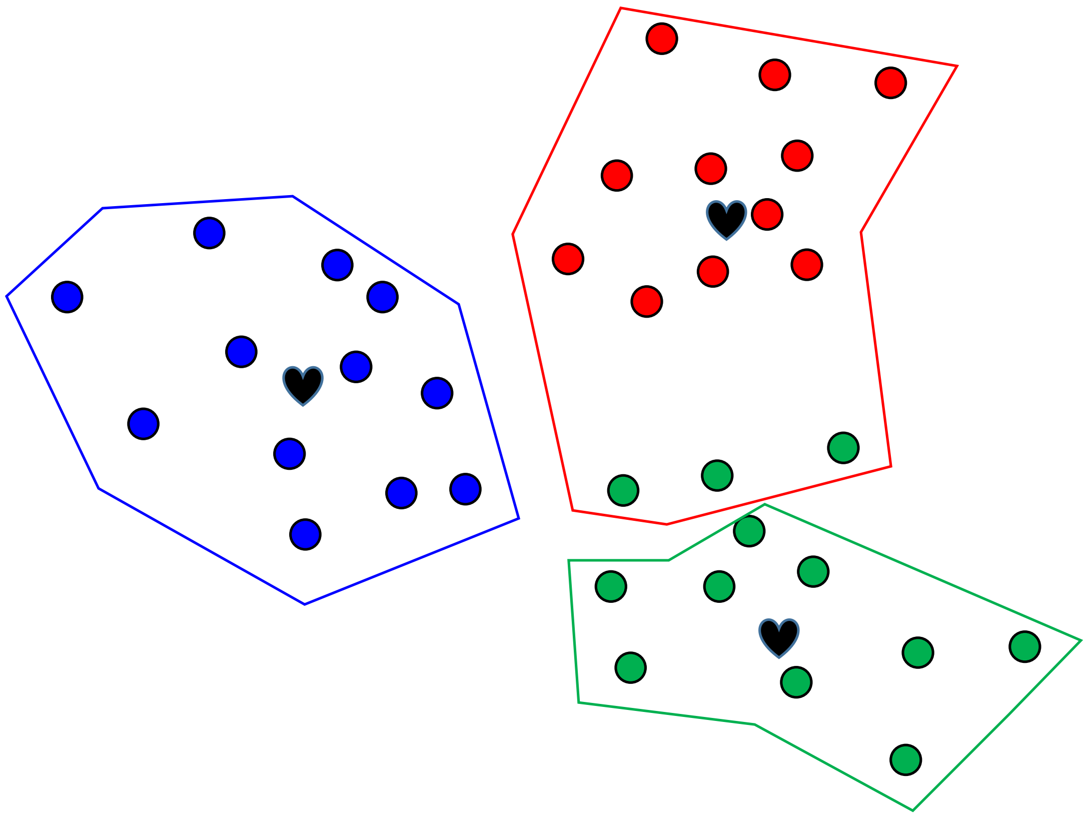
k-means算法与GMM算法

第四次迭代：更新每个簇的中心点.



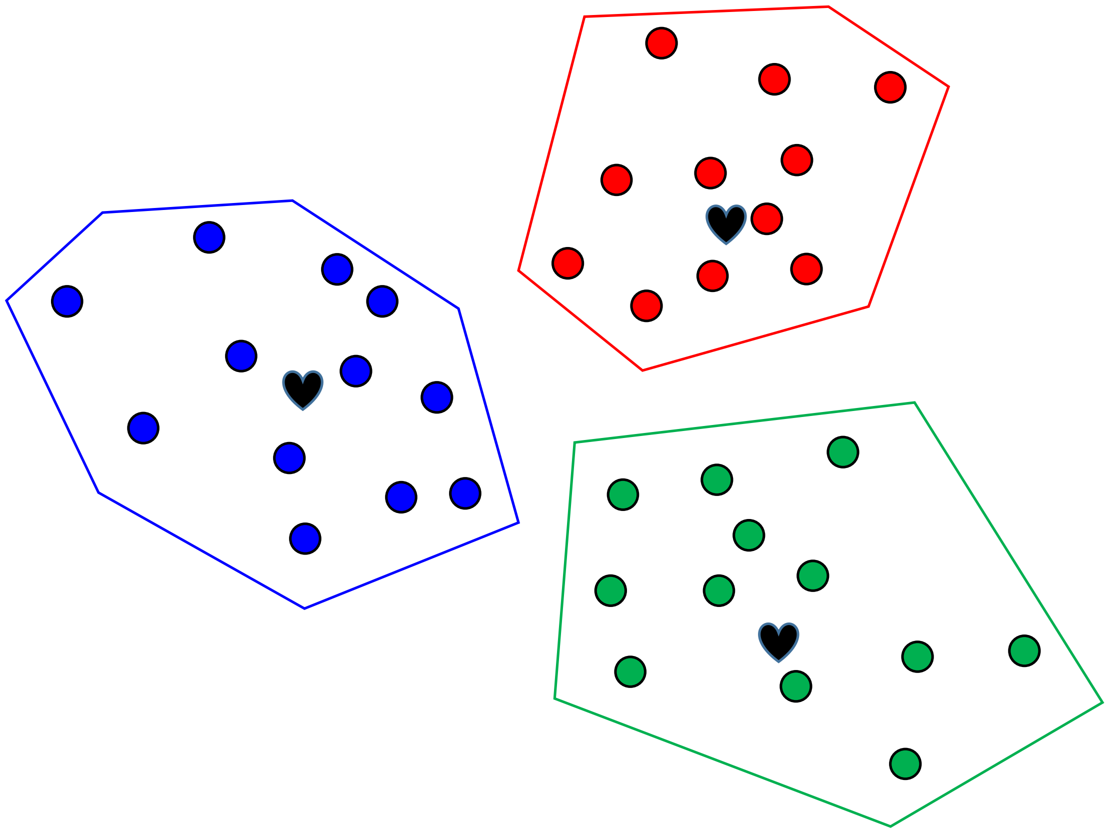
k-means算法与GMM算法

第四次迭代：计算每个样本点的归属值.



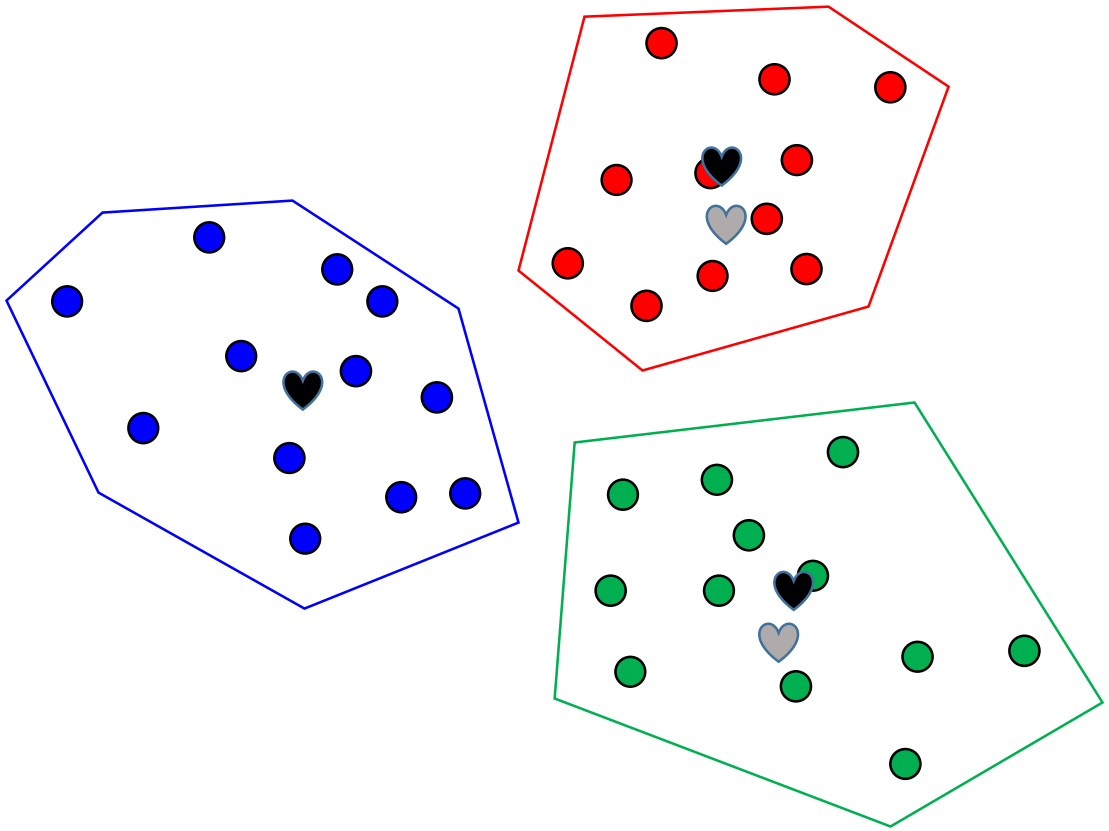
k-means算法与GMM算法

第四次迭代：计算每个样本点的归属值.



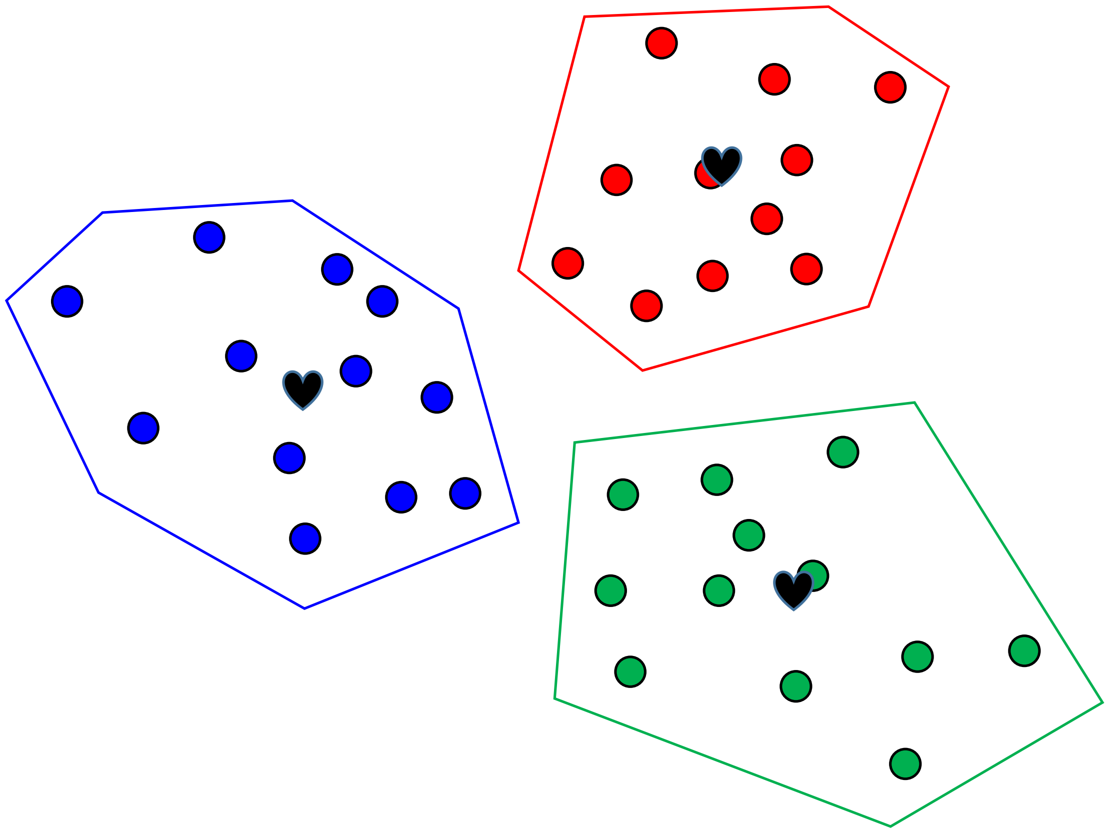
k-means算法与GMM算法

第四次迭代：更新每个簇的中心点.



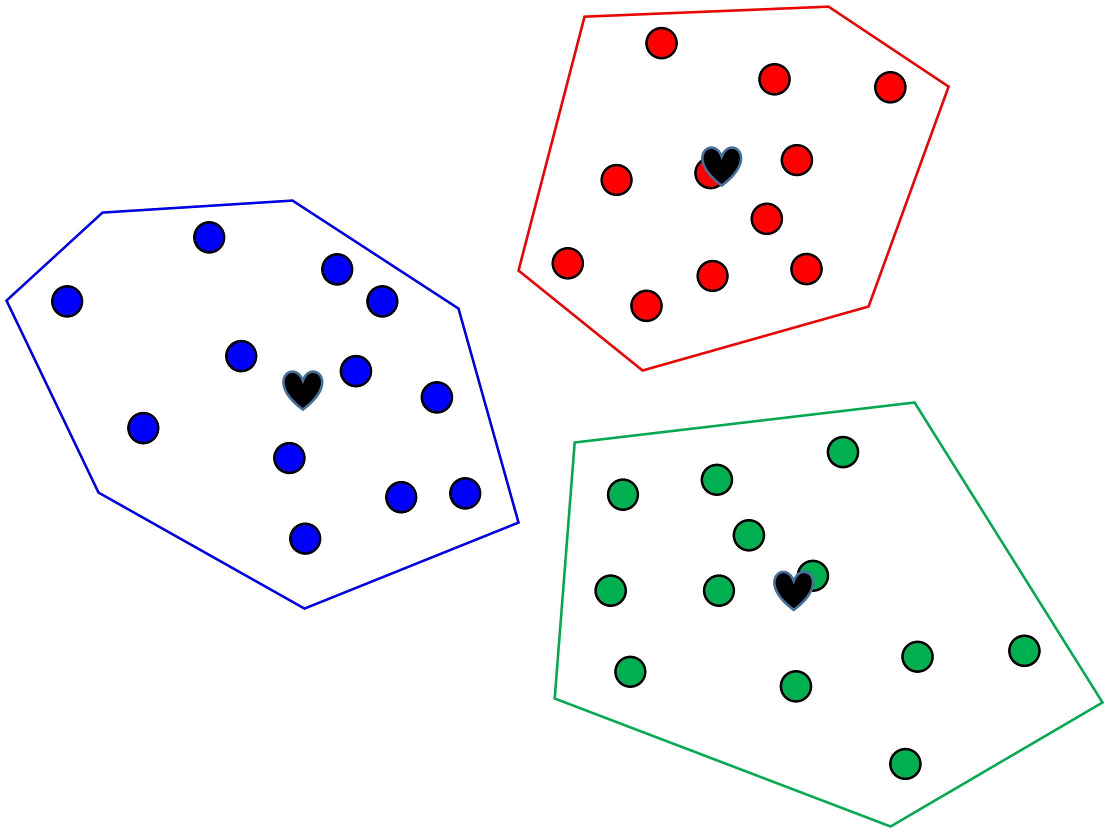
k-means算法与GMM算法

第五次迭代：计算每个样本点的归属值.



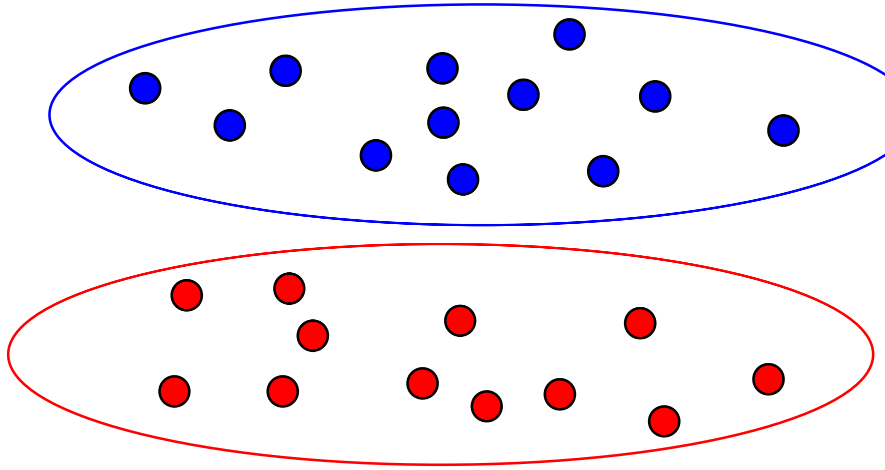
k-means算法与GMM算法

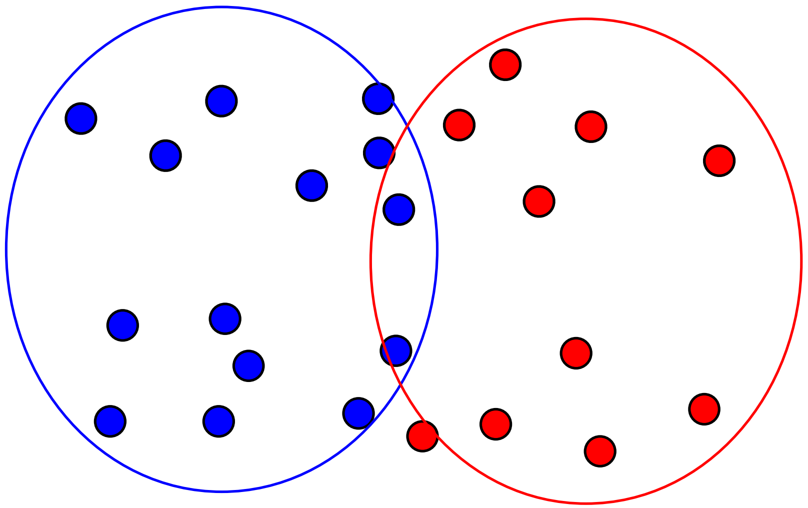
第五次迭代：更新每个簇的中心点.



k-means算法与GMM算法

k-means算法对中心点的初始值敏感.





k-means算法与GMM算法

- 高斯混合模型（GMM）是一个生成模型，即可以生成数据。
- 数据按照如下方式生成：
  - 按照如下概率生成归属值

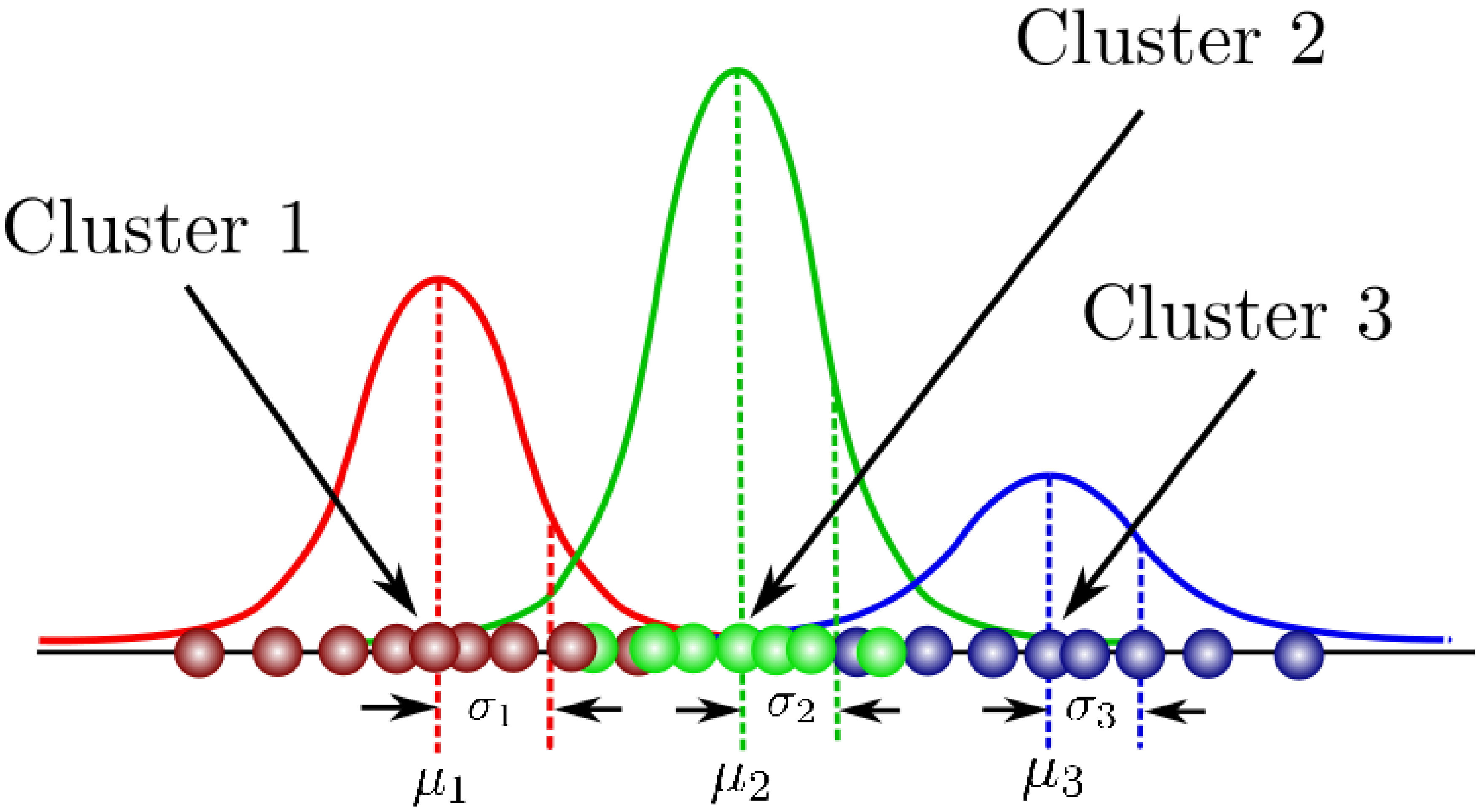
$$p(I = i) = p_i$$

  - 假设生成的归属值为i，利用第i个簇的均值和协方差矩阵产生高斯分布的样本

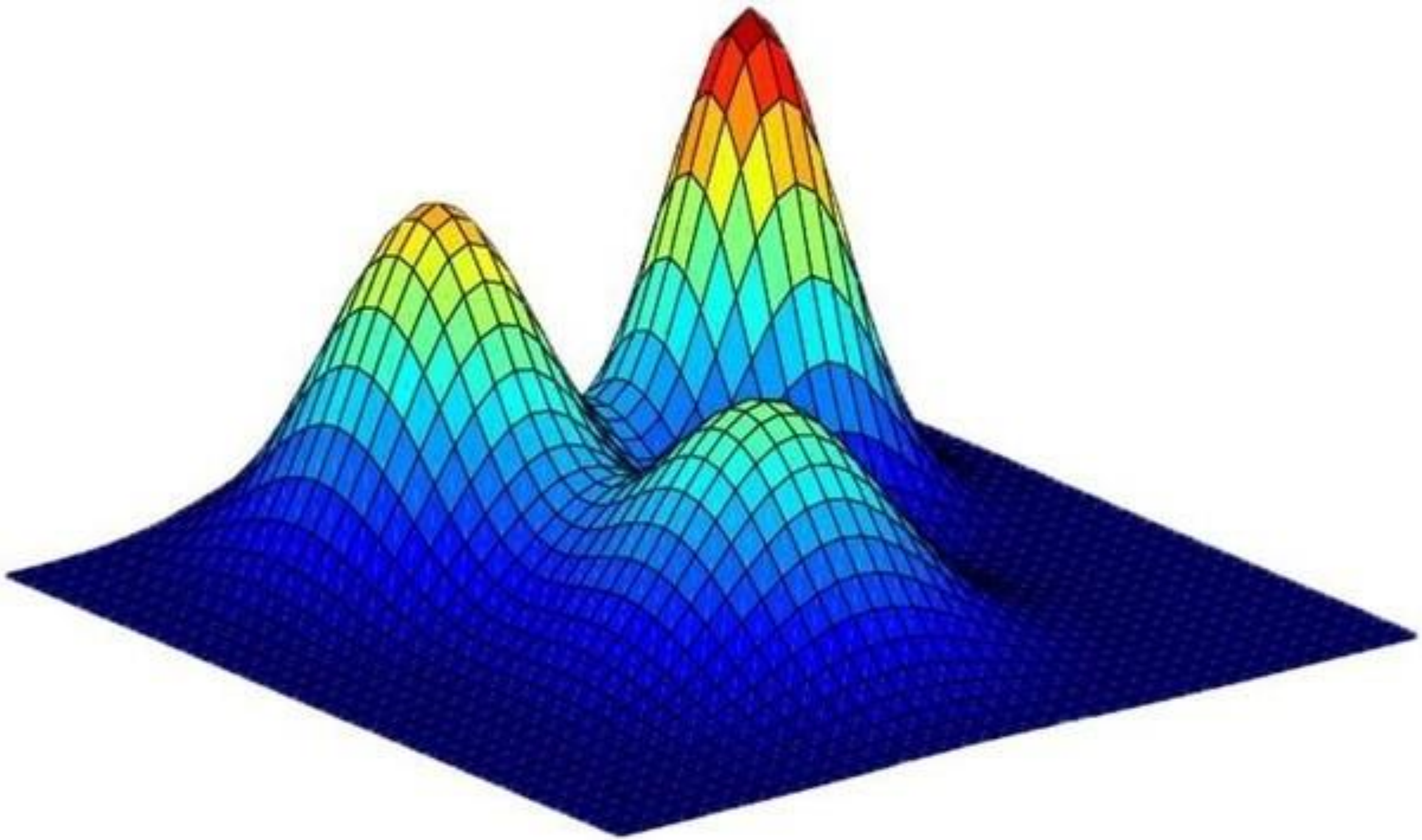
$$\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{C}_i)$$

- 这种数据生成方式被称为层次分布(hierarchical distribution).
- GMM模型需要估计的参数为$\{p_1, \boldsymbol{\mu}_1, \boldsymbol{C}_1, p_2, \boldsymbol{\mu}_2, \boldsymbol{C}_2, \cdots p_k, \boldsymbol{\mu}_k, \boldsymbol{C}_k\}$

k-means算法与GMM算法



k-means算法与GMM算法



k-means算法与GMM算法

- GMM算法假设每个样本的归属值为随机变量.
- GMM也通过迭代方法估计参数.
- 初始化参数$\{p_1, \boldsymbol{\mu}_1, \boldsymbol{C}_1, p_2, \boldsymbol{\mu}_2, \boldsymbol{C}_2, \cdots p_k, \boldsymbol{\mu}_k, \boldsymbol{C}_k\}$，初始化归属值矩阵为$n$行$k$列的零值矩阵$\boldsymbol{M}$.
- 每次迭代，完成以下两步
  - 计算每个样本归属值的概率

$$\boldsymbol{M}_{it} = p(I(\boldsymbol{x}_i) = t) = \frac{\hat{p}_k \mathcal{N}(\boldsymbol{x}_i; \widehat{\boldsymbol{\mu}}_t, \widehat{\boldsymbol{C}}_t)}{\sum_{j=1}^{k} \hat{p}_j \mathcal{N}(\boldsymbol{x}_i; \widehat{\boldsymbol{\mu}}_j, \widehat{\boldsymbol{C}}_j)}$$

  - 计算每个簇的参数，$\widehat{\boldsymbol{\mu}}_t = \frac{\sum_{i=1}^{n} M_{it}\boldsymbol{x}_i}{\sum_{i=1}^{n} M_{it}}$, $\widehat{\boldsymbol{C}}_t = \frac{\sum_{i=1}^{n} M_{it}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_t)(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_t)^T}{\sum_{i=1}^{n} M_{it}}$.

k-means算法与GMM算法

初始化：共3个簇，初始化参数$\{p_1, \boldsymbol{\mu}_1, \boldsymbol{C}_1, p_2, \boldsymbol{\mu}_2, \boldsymbol{C}_2, \cdots p_k, \boldsymbol{\mu}_k, \boldsymbol{C}_k\}$，可以利用k-means结果初始化这些参数；初始化归属值矩阵.

	Cluster 1	Cluster 2	Cluster 3
Sample 1	0	0	0
Sample 2	0	0	0
Sample 3	0	0	0
Sample 4	0	0	0
Sample 5	0	0	0
Sample 6	0	0	0
Sample 7	0	0	0

k-means算法与GMM算法

第一次迭代：更新归属值矩阵，$\boldsymbol{M}_{it} = p(I(\boldsymbol{x}_i) = t) = \frac{\hat{p}_t \mathcal{N}(\boldsymbol{x}_i;\hat{\boldsymbol{\mu}}_t,\widehat{\boldsymbol{C}}_t)}{\sum_{j=1}^{k} \hat{p}_j \mathcal{N}(\boldsymbol{x}_i;\hat{\boldsymbol{\mu}}_j,\widehat{\boldsymbol{C}}_j)}$.

	Cluster 1	Cluster 2	Cluster 3
Sample 1	1/4	1/2	1/4
Sample 2	1	0	0
Sample 3	3/5	1/5	1/5
Sample 4	1/2	1/2	0
Sample 5	1/3	1/6	1/2
Sample 6	4/7	1/7	2/7
Sample 7	2/5	1/5	2/5

k-means算法与GMM算法

第一次迭代：更新参数， $\widehat{\boldsymbol{\mu}}_t = \frac{\sum_{i=1}^{n} M_{it}\boldsymbol{x}_i}{\sum_{i=1}^{n} M_{it}}$, $\widehat{\boldsymbol{C}}_t = \frac{\sum_{i=1}^{n} M_{it}(\boldsymbol{x}_i-\widehat{\boldsymbol{\mu}}_t)(\boldsymbol{x}_i-\widehat{\boldsymbol{\mu}}_t)^T}{\sum_{i=1}^{n} M_{it}}$.

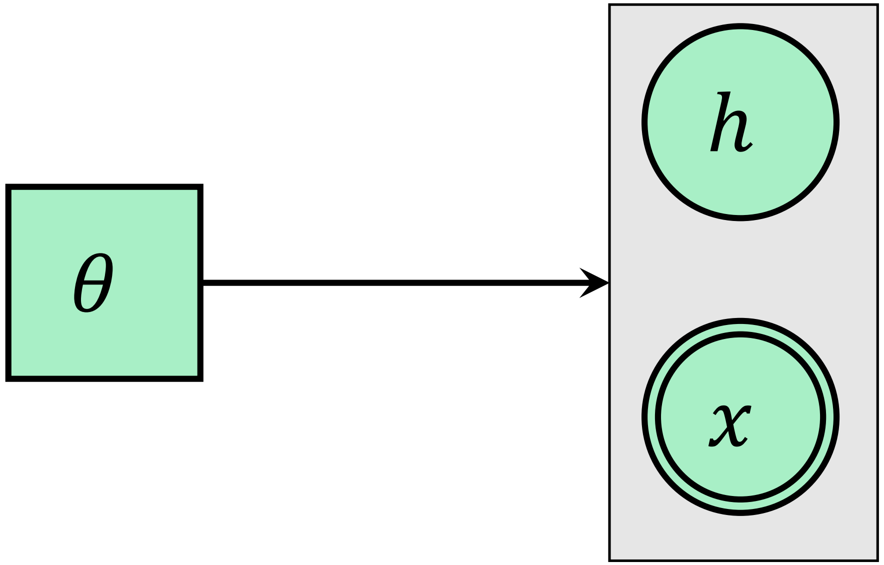
	Cluster 1	Cluster 2	Cluster 3
Sample 1	1/4	1/2	1/4
Sample 2	1	0	0
Sample 3	3/5	1/5	1/5
Sample 4	1/2	1/2	0
Sample 5	1/3	1/6	1/2
Sample 6	4/7	1/7	2/7
Sample 7	2/5	1/5	2/5

k-means算法与GMM算法

- 在k-means算法中，每个样本的归属值确定，即每个样本只能归属于某一个簇，这种归属为硬归属(hard membership).
- 在GMM算法中，每个样本的归属值为随机变量，每个样本可以同时归属于所有的簇，归属于每一个簇的值为一个概率，这种归属为软归属(soft membership).
- GMM算法属于Expectation maximization (EM)算法。在EM算法中，目标函数有变量和随机变量。EM算法是一种迭代算法，在每次迭代中，进行如下两步：
    - E步：处理随机变量，对目标函数取期望.
    - M步：优化变量，使目标函数最大.

EM算法

- EM算法是一个参数估计算法，适用于量测或观测量不完整情况。
- 在下图中，θ是一个参数，x是关于θ的观测值(observation)或量测值(measurement)。
- 但仅有x，量测是不完整的， x和h组成一个完整量测值。但h未知。



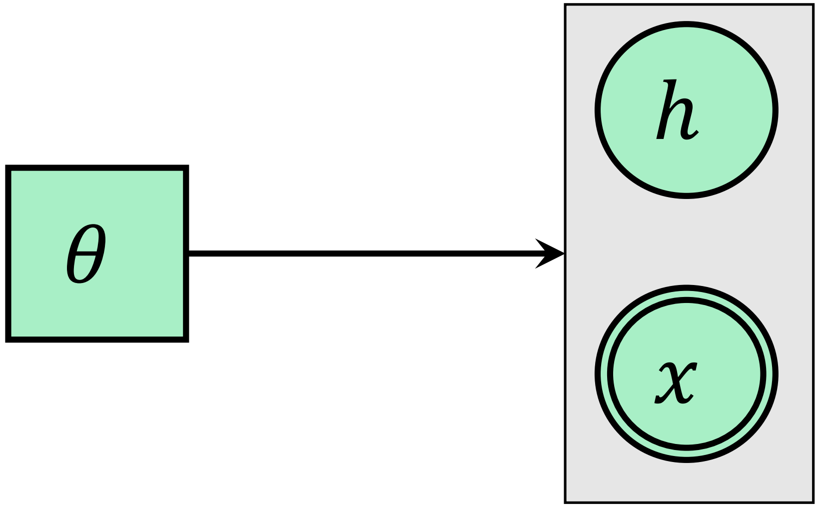
EM算法

- 但仅有x，量测是不完整的， x和h组成一个完整量测值。但h未知。
- 之所以称“量测不完整”，是因为根据已有的量测，难以通过最大似然估计参数，

$$\max_\theta p(x;\theta)$$

- 如果有完整量测x和h，通过最大似然估计参数变得简单，即

$$\max_\theta p(x,h;\theta) \text{ 或者 } \max_\theta \log(p(x,h;\theta))$$



EM算法

$$\log(p(x;\theta)) = \log\left(\sum_h q(h)\frac{p(x,h;\theta)}{q(h)}\right) = \log\left(\mathrm{E}_{q(h)}\left[\frac{p(x,h;\theta)}{q(h)}\right]\right)$$

$$\geq \mathrm{E}_{q(h)}\left[\log\left(\frac{p(x,h;\theta)}{q(h)}\right)\right]$$

$$= \mathrm{E}_{q(h)}[\log(p(x,h;\theta))] - \mathrm{E}_{q(h)}[\log(q(h))]$$

- 如果$q(h)$已知，寻找θ最大化 $\log(p(x;\theta))$ 等价于最大化 $\mathrm{E}_{q(h)}[\log(p(x,h;\theta))]$。
- 最优的$q(h)$应该是什么样?

EM算法

$$\log(p(x;\theta)) \geq \mathrm{E}_{q(h)}\left[\log\left(\frac{p(x,h;\theta)}{q(h)}\right)\right] = \mathrm{E}_{q(h)}\left[\log\left(\frac{p(x;\theta)p(h|x;\theta)}{q(h)}\right)\right]$$

$$= \mathrm{E}_{q(h)}[\log(p(x;\theta))] - \mathrm{E}_{q(h)}\left[\log\left(\frac{p(h|x;\theta)}{q(h)}\right)\right]$$

$$= \log(p(x;\theta)) - D_{KL}(q(h)||p(h|x;\theta))$$

- 当$q(h) = p(h|x;\theta)$时， $D_{KL}(q(h)||p(h|x;\theta))$取最小值，为0；
- 此时大于等于号变成等号。

EM算法

$$\log(p(x;\theta)) \geq \mathrm{E}_{q(h)}[\log(p(x,h;\theta))] - \mathrm{E}_{q(h)}[\log(q(h))] \Rightarrow \hat{\theta} = argmax_{\theta}\mathrm{E}_{q(h)}[\log(p(x,h;\theta))]$$

$$\log(p(x;\theta)) \geq \log(p(x;\theta)) - D_{KL}(q(h)||p(h|x;\theta)) \Rightarrow q(h) = p(h|x;\theta)$$

- 在估计θ时候需要$q(h)$;
- 在估计$q(h)$时候需要θ。

EM算法

$$\log(p(x;\theta)) \geq \mathrm{E}_{q(h)}\left[\log\left(p(x,h;\theta)\right)\right] - \mathrm{E}_{q(h)}\left[\log\left(q(h)\right)\right] \Rightarrow \hat{\theta} = \mathrm{argmax}_{\theta}\mathrm{E}_{q(h)}\left[\log\left(p(x,h;\theta)\right)\right]$$

$$\log(p(x;\theta)) \geq \log(p(x;\theta)) - D_{KL}(q(h)||p(h|x;\theta)) \Rightarrow q(h) = p(h|x;\theta)$$

- 在估计θ时候需要$q(h)$；
- 在估计$q(h)$时候需要θ。

交互迭代

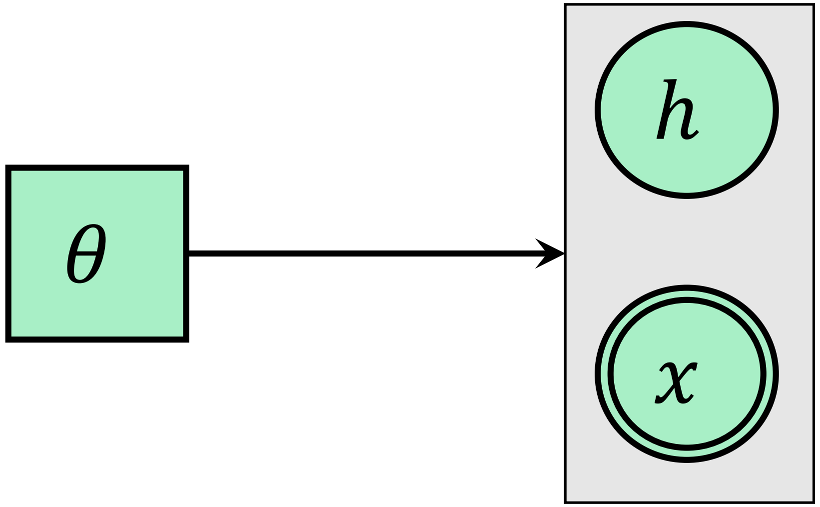
EM算法

输入： 量测x。
输出： $\hat{\theta}$。

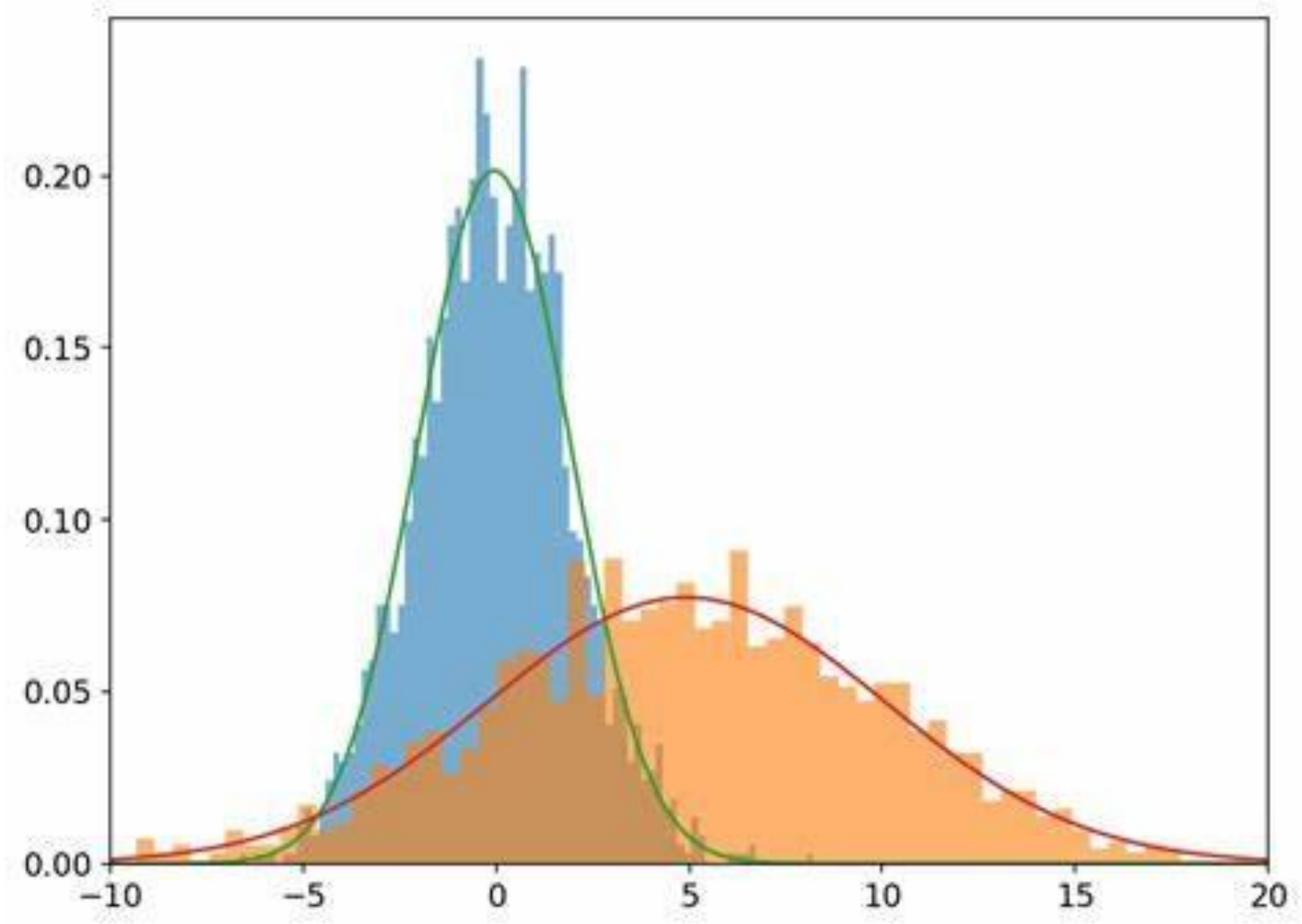
(1) 初始化：随机初始化参数$\hat{\theta}$，$q(h) = p(h|x;\hat{\theta})$
(2) 循环步骤(3)(4)直至终止条件满足
(3) 计算$\hat{\theta} = \mathrm{argmax}_\theta \mathrm{E}_{q(h)}[\log(p(x,h;\theta))]$。
(4) 计算$q(h) = p(h|x;\hat{\theta})$。
(5) 返回$\hat{\theta}$。

EM算法

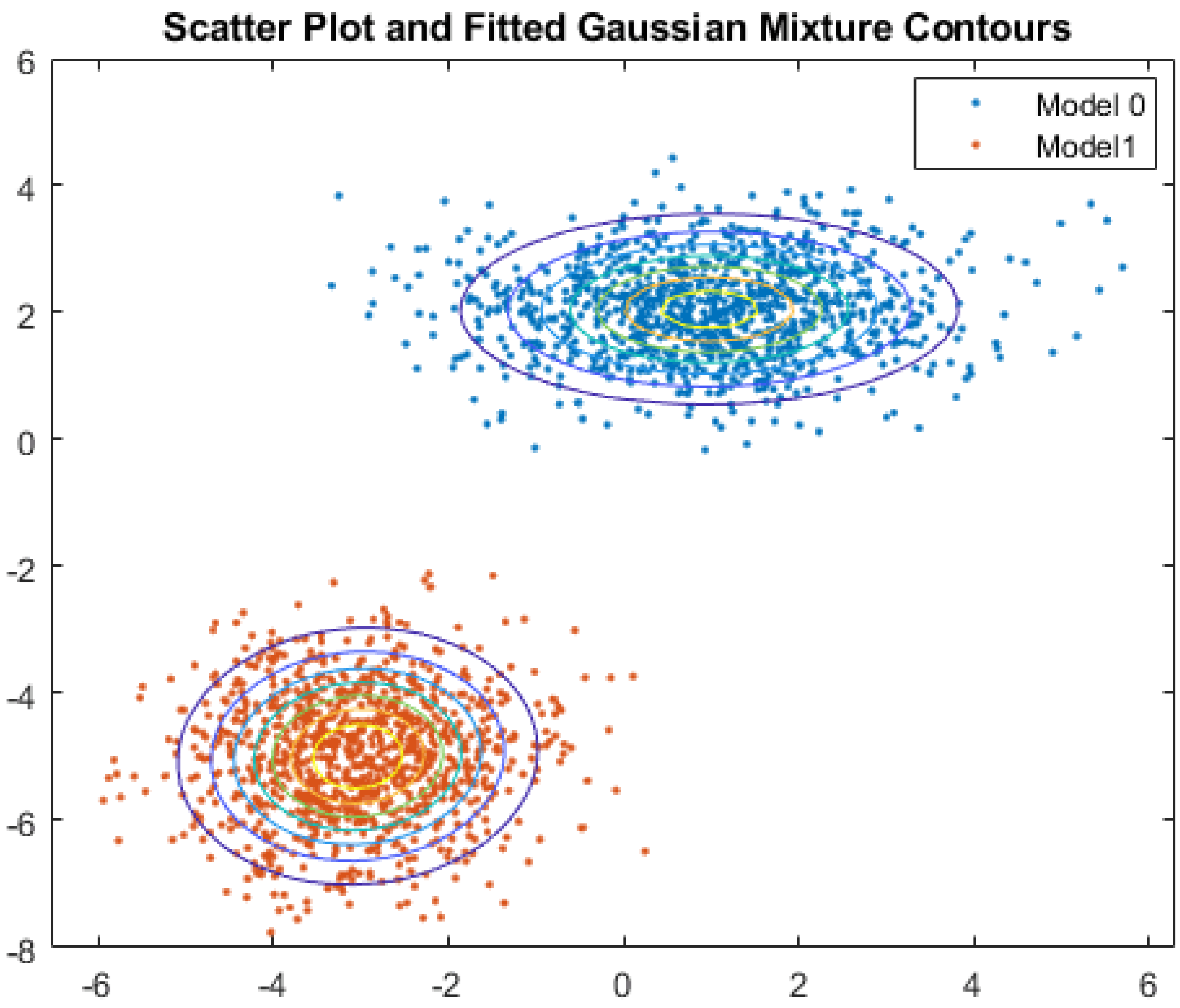
$$\max_\theta p(x, H; \theta) \Rightarrow \max_\theta \mathrm{E}_{p(h|x;\theta)}\big[\log\big(p(x, h; \theta)\big)\big] \Rightarrow \begin{cases} \operatorname{argmax}_\theta \mathrm{E}_{q(h)}\big[\log\big(p(x, h; \theta)\big)\big] \\ q(h) = p(h|x; \hat{\theta}) \end{cases}$$



EM算法



EM算法


Scatter Plot and Fitted Gaussian Mixture Contours
Model 0
Model1
6
4
2
0
-2
-4
-6
-8
-6
-4
-2
0
2
4
6

EM算法

- 训练数据为，$x_1, x_2 \cdots, x_n$，
- 簇的个数为2，簇的标签本别为0和1，令变量H表示簇的类别，

$$p(H=1)=q, \quad p(H=0)=1-q$$

- 每个簇内的样本都服从高斯分布，有

$$p(x|H=1)=\mathcal{N}(\mu_1,\sigma_1^2)=\frac{1}{\sqrt{2\pi}}e^{-\frac{(x-\mu_1)^2}{\sigma_1^2}}, \quad p(x|H=0)=\mathcal{N}(\mu_0,\sigma_0^2)=\frac{1}{\sqrt{2\pi}}e^{-\frac{(x-\mu_0)^2}{\sigma_0^2}}$$

- 参数为 $\theta=\{q,\mu_0,\sigma_0^2,\mu_1,\sigma_1^2\}$

EM算法

$$p(x;\theta) = p(x, H=0) + p(x, H=1) = p(H=0)p(x, H=0) + p(x, H=1)p(x, H=1)$$

$$= (1-q)\frac{1}{\sqrt{2\pi}}e^{-\frac{(x-\mu_0)^2}{\sigma_0^2}} + q\frac{1}{\sqrt{2\pi}}e^{-\frac{(x-\mu_1)^2}{\sigma_1^2}}$$

$$p(x_1, \cdots, x_n;\theta) = \prod_{i=1}^{n} p(x_i;\theta) = \prod_{i=1}^{n}\left((1-q)\frac{1}{\sqrt{2\pi}}e^{-\frac{(x_i-\mu_0)^2}{\sigma_0^2}} + q\frac{1}{\sqrt{2\pi}}e^{-\frac{(x_i-\mu_1)^2}{\sigma_1^2}}\right)$$

$$\log\big(p(x_1, \cdots, x_n;\theta)\big) = \sum_{i=1}^{n}\log\left((1-q)\frac{1}{\sqrt{2\pi}}e^{-\frac{(x_i-\mu_0)^2}{\sigma_0^2}} + q\frac{1}{\sqrt{2\pi}}e^{-\frac{(x_i-\mu_1)^2}{\sigma_1^2}}\right)$$

EM算法

$$\max_\theta \log\big(p(x_1,\cdots,x_n;\theta)\big) \Rightarrow \max_\theta \sum_{i=1}^{n} \log\left((1-q)\frac{1}{\sqrt{2\pi}} e^{-\frac{(x_i-\mu_0)^2}{\sigma_0^2}} + q\frac{1}{\sqrt{2\pi}} e^{-\frac{(x_i-\mu_1)^2}{\sigma_1^2}} \right)$$

EM算法

输入： 量测x。
输出： $\hat{\theta}$。

(1) 初始化：随机初始化参数$\hat{\theta}$，$q(h) = p\left(h \middle| x; \hat{\theta}\right)$

(2) 循环步骤(3)(4)直至终止条件满足

(3) 　计算$\hat{\theta} = \mathrm{argmax}_{\theta} \mathrm{E}_{q(h)}\left[\log\left(p(x, h; \theta)\right)\right]$。

(4) 　计算$q(h) = \boxed{p\left(h \middle| x; \hat{\theta}\right)。}$

(5) 返回$\hat{\theta}$。

EM算法

$$p\left(H_i=0|x_i;\hat{\theta}\right)=\frac{p\left(H_i=0,x_i;\hat{\theta}\right)}{p\left(x_i;\hat{\theta}\right)}=\frac{p\left(H_i=0\right)p\left(x|H_i=0;\hat{\theta}\right)}{p\left(x_i;\hat{\theta}\right)}=\frac{(1-\hat{q})\mathcal{N}(\hat{\mu}_0,\hat{\sigma}_0^2)}{p\left(x_i;\hat{\theta}\right)}$$

$$p\left(H_i=1|x_i;\hat{\theta}\right)=\frac{p\left(H_i=1,x_i;\hat{\theta}\right)}{p\left(x_i;\hat{\theta}\right)}=\frac{p\left(H_i=1\right)p\left(x_i|H_i=1;\hat{\theta}\right)}{p\left(x_i;\hat{\theta}\right)}=\frac{\hat{q}\mathcal{N}(\hat{\mu}_1,\hat{\sigma}_1^2)}{p\left(x_i;\hat{\theta}\right)}$$

$$\left.\begin{array}{l} p\left(H_i=0|x_i;\hat{\theta}\right)=\dfrac{(1-\hat{q})\mathcal{N}(\hat{\mu}_0,\hat{\sigma}_0^2)}{q\mathcal{N}(\hat{\mu}_1,\hat{\sigma}_1^2)+(1-\hat{q})\mathcal{N}(\hat{\mu}_0,\hat{\sigma}_0^2)}=\pi_{i0} \\[3em] p\left(H_i=1|x_i;\hat{\theta}\right)=\dfrac{\hat{q}\mathcal{N}(\hat{\mu}_1,\sigma_1^2)}{\hat{q}\mathcal{N}(\hat{\mu}_1,\hat{\sigma}_1^2)+(1-\hat{q})\mathcal{N}(\hat{\mu}_0,\sigma_0^2)}=\pi_{i1} \end{array}\right\} \quad E[H_i]=\pi_{i1}$$

EM算法

输入： 量测x。
输出： $\hat{\theta}$。

(1) 初始化：随机初始化参数$\hat{\theta}$，$q(h) = p\left(h \middle| x; \hat{\theta}\right)$

(2) 循环步骤(3)(4)直至终止条件满足

(3) 计算$\hat{\theta} = \operatorname{argmax}_{\theta} \boxed{\mathrm{E}_{q(h)}\left[\log\left(p(x, h; \theta)\right)\right]}$。

(4) 计算$q(h) = p\left(h \middle| x; \hat{\theta}\right)$。

(5) 返回$\hat{\theta}$。

EM算法

$$\mathrm{E}_{q(h)}\left[\log\left(p(x,h;\theta)\right)\right]$$

$$p(H_i=0,x_i;\theta)=p(H=0)p(x_i|H=0;\theta)=(1-q)\mathcal{N}(\mu_0,\sigma_0^2)$$

$$p(H_i=1,x_i;\theta)=p(H=1)p(x_i|H=1;\theta)=q\mathcal{N}(\mu_1,\sigma_1^2)$$

$$p(H_i,x_i;\theta)=[q\mathcal{N}(\mu_1,\sigma_1^2)]^{H_i}[(1-q)\mathcal{N}(\mu_0,\sigma_0^2)]^{1-H_i}$$

$$p(\{H_i,x_i\}_{i=1}^n;\theta)=\prod_{i=1}^{n}p(H_i,x_i;\theta)=[q\mathcal{N}(\mu_1,\sigma_1^2)]^{H_i}[(1-q)\mathcal{N}(\mu_0,\sigma_0^2)]^{1-H_i}$$

$$\log\left(p(\{H_i,x_i\}_{i=1}^n;\theta)\right)$$

$$=\sum_{i=1}^{n}\left\{H_i\log(q)+H_i\log\left(\mathcal{N}(\mu_1,\sigma_1^2)\right)+(1-H_i)\log(1-q)+(1-H_i)\log(\mathcal{N}(\mu_0,\sigma_0^2))\right\}$$

EM算法

$$\log\left(p(\{H_i, x_i\}_{i=1}^{n}; \theta)\right)$$

$$\mathrm{E}_{q(h)}\left[\log\left(p(x, h; \theta)\right)\right]$$

$$= \sum_{i=1}^{n} \left\{ H_i \log(q) + H_i \log\left(\mathcal{N}(\mu_1, \sigma_1^2)\right) + (1 - H_i)\log(1 - q) + (1 - H_i)\log(\mathcal{N}(\mu_0, \sigma_0^2)) \right\}$$

$$E\left[\log\left(p(\{H_i, x_i\}_{i=1}^{n}; \theta)\right)\right]$$

$$= \sum_{i=1}^{n} \left\{ E[H_i] \log(q) + E[H_i] \log\left(\mathcal{N}(\mu_1, \sigma_1^2)\right) + (1 - E[H_i])\log(1 - q) + (1 - E[H_i])\log(\mathcal{N}(\mu_0, \sigma_0^2)) \right\}$$

$$= \sum_{i=1}^{n} \left\{ \pi_{i1} \log(q) + \pi_{i1} \log\left(\mathcal{N}(\mu_1, \sigma_1^2)\right) + (1 - \pi_{i1})\log(1 - q) + (1 - \pi_{i1})\log(\mathcal{N}(\mu_0, \sigma_0^2)) \right\}$$

EM算法

$$\log\left(\mathcal{N}(\mu_0,\sigma_0^2)\right) = -\log(\sqrt{2\pi}) - \log(\sigma_0) - \frac{(x-\mu_0)^2}{2\sigma_0^2}$$

$$E_{q(h)}\left[\log\left(p(x,h;\theta)\right)\right]$$

$$\log\left(\mathcal{N}(\mu_1,\sigma_1^2)\right) = -\log(\sqrt{2\pi}) - \log(\sigma_1) - \frac{(x-\mu_1)^2}{2\sigma_1^2}$$

$$E\left[\log\left(p(\{H_i,x_i\}_{i=1}^n;\theta)\right)\right]$$

$$= \sum_{i=1}^n \left\{\pi_{i1}\log(q) + \pi_{i1}\log\left(\mathcal{N}(\mu_1,\sigma_1^2)\right) + (1-\pi_{i1})\log(1-q) + (1-\pi_{i1})\log(\mathcal{N}(\mu_0,\sigma_0^2))\right\}$$

$$\propto \sum_{i=1}^n \left\{\pi_{i1}\log(q) - \pi_{i1}\log(\sigma_1) - \pi_{i1}\frac{(x_i-\mu_1)^2}{2\sigma_1^2} + \pi_{i0}\log(1-q) - \pi_{i0}\log(\sigma_0) - \pi_{i0}\frac{(x_i-\mu_0)^2}{2\sigma_0^2}\right\}$$

EM算法

$$E\left[\log\left(p(\{H_i, x_i\}_{i=1}^{n}; \theta)\right)\right]$$

$$\mathrm{E}_{q(h)}\left[\log\left(p(x, h; \theta)\right)\right]$$

$$\propto \sum_{i=1}^{n}\left\{\pi_{i1}\log(q) - \pi_{i1}\log(\sigma_1) - \pi_{i1}\frac{(x_i-\mu_1)^2}{2\sigma_1^2} + (1-\pi_{i1})\log(1-q) - \pi_{i0}\log(\sigma_0) - \pi_{i0}\frac{(x_i-\mu_0)^2}{2\sigma_0^2}\right\}$$

$$\frac{\partial E\left[\log\left(p(\{H_i, x_i\}_{i=1}^{n}; \theta)\right)\right]}{\partial q} = \sum_{i=1}^{n}\left(\frac{\pi_{i1}}{q} - \frac{\pi_{i0}}{1-q}\right) = 0 \Rightarrow q = \frac{\sum_{i=1}^{n}\pi_{i1}}{\sum_{i=1}^{n}(\pi_{i1}+\pi_{i10})} = \frac{\sum_{i=1}^{n}\pi_{i1}}{n}$$

$$\frac{\partial E\left[\log\left(p(\{H_i, x_i\}_{i=1}^{n}; \theta)\right)\right]}{\partial \mu_1} = \pi_{i1}\frac{(x_i-\mu_1)}{\sigma_1^2} = 0 \Rightarrow \mu_1 = \frac{\sum_{i=1}^{n}\pi_{i1}x_i}{\sum_{i=1}^{n}\pi_{i1}}$$

$$\frac{\partial E\left[\log\left(p(\{H_i, x_i\}_{i=1}^{n}; \theta)\right)\right]}{\partial \mu_0} = \pi_{i0}\frac{(x_i-\mu_0)}{\sigma_0^2} = 0 \Rightarrow \mu_0 = \frac{\sum_{i=1}^{n}\pi_{i0}x_i}{\sum_{i=1}^{n}\pi_{i0}}$$

EM算法

$$E\left[\log\left(p(\{H_i, x_i\}_{i=1}^{n}; \theta)\right)\right]$$

$$\mathrm{E}_{q(h)}\left[\log\left(p(x, h; \theta)\right)\right]$$

$$\propto \sum_{i=1}^{n} \left\{ \pi_{i1} \log(q) - \pi_{i1} \log(\sigma_1) - \pi_{i1} \frac{(x_i - \mu_1)^2}{2\sigma_1^2} + (1 - \pi_{i1}) \log(1 - q) - \pi_{i0} \log(\sigma_0) - \pi_{i0} \frac{(x_i - \mu_0)^2}{2\sigma_0^2} \right\}$$

$$\frac{\partial E\left[\log\left(p(\{H_i, x_i\}_{i=1}^{n}; \theta)\right)\right]}{\partial \sigma_1} = -\frac{\pi_{i1}}{\sigma_1} + \pi_{i1} \frac{(x_i - \mu_1)^2}{\sigma_1^3} = 0 \Rightarrow \sigma_1^2 = \frac{\sum_{i=1}^{n} \pi_{i1} (x_i - \mu_1)^2}{\sum_{i=1}^{n} \pi_{i1}}$$

$$\frac{\partial E\left[\log\left(p(\{H_i, x_i\}_{i=1}^{n}; \theta)\right)\right]}{\partial \sigma_0} = -\frac{\pi_{i0}}{\sigma_0} + \pi_{i0} \frac{(x_i - \mu_0)^2}{\sigma_0^3} = 0 \Rightarrow \sigma_0^2 = \frac{\sum_{i=1}^{n} \pi_{i0} (x_i - \mu_0)^2}{\sum_{i=1}^{n} \pi_{i0}}$$

EM算法

GMM算法

输入： 量测x。
输出： $\hat{\theta}$。

(1) 初始化：随机初始化参数$\hat{\theta}$，$q(h)=p\left(h\middle|x;\hat{\theta}\right)$
(2) 循环步骤(3)(4)直至终止条件满足
(3) 计算$\hat{\theta}=\mathrm{argmax}_{\theta}\mathrm{E}_{q(h)}\left[\log\left(p(x,h;\theta)\right)\right]$。
(4) 计算$q(h)=p\left(h\middle|x;\hat{\theta}\right)$。
(5) 返回$\hat{\theta}$。

$$q=\frac{\sum_{i=1}^{n}\pi_{i1}}{\sum_{i=1}^{n}(\pi_{i1}+\pi_{i10})}=\frac{\sum_{i=1}^{n}\pi_{i1}}{n}$$

$$\mu_1=\frac{\sum_{i=1}^{n}\pi_{i1}x_i}{\sum_{i=1}^{n}\pi_{i1}}$$

$$\mu_0=\frac{\sum_{i=1}^{n}\pi_{i0}x_i}{\sum_{i=1}^{n}\pi_{i0}}$$

$$\sigma_1^2=\frac{\sum_{i=1}^{n}\pi_{i1}(x_i-\mu_1)^2}{\sum_{i=1}^{n}\pi_{i1}}$$

$$\sigma_0^2=\frac{\sum_{i=1}^{n}\pi_{i0}(x_i-\mu_0)^2}{\sum_{i=1}^{n}\pi_{i0}}$$

$$p\left(H_i=0|x_i;\hat{\theta}\right)=\frac{(1-\hat{q})\mathcal{N}(\hat{\mu}_0,\hat{\sigma}_0^2)}{q\mathcal{N}(\hat{\mu}_1,\hat{\sigma}_1^2)+(1-\hat{q})\mathcal{N}(\hat{\mu}_0,\hat{\sigma}_0^2)}=\pi_{i0}$$

$$p\left(H_i=1|x_i;\hat{\theta}\right)=\frac{\hat{q}\mathcal{N}(\mu_1,\sigma_1^2)}{\hat{q}\mathcal{N}(\hat{\mu}_1,\hat{\sigma}_1^2)+(1-\hat{q})\mathcal{N}(\mu_0,\sigma_0^2)}=\pi_{i1}$$

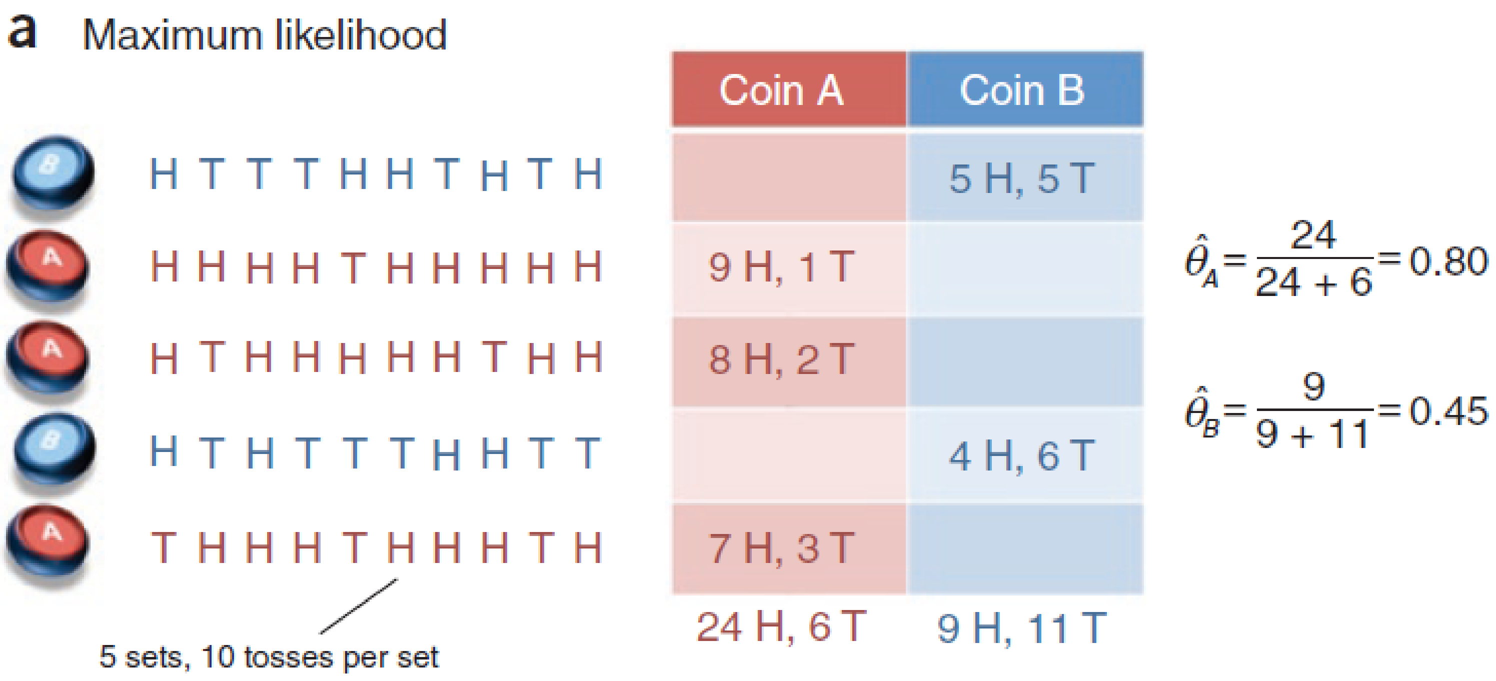
EM算法

- 有两枚硬币，抛一次硬币，正面朝上的概率分别为θ_A、θ_B；
- 重复以下步骤五次：每次首先随机（等概率）从两枚硬币中选择一枚硬币，然后抛硬币十次，得到十个结果

H	T	T	T	H	H	T	H	T	H
H	H	H	H	T	H	H	H	H	H
H	T	H	H	H	H	H	T	H	H
H	T	H	T	T	T	H	H	T	T
T	H	H	H	T	H	H	H	T	H

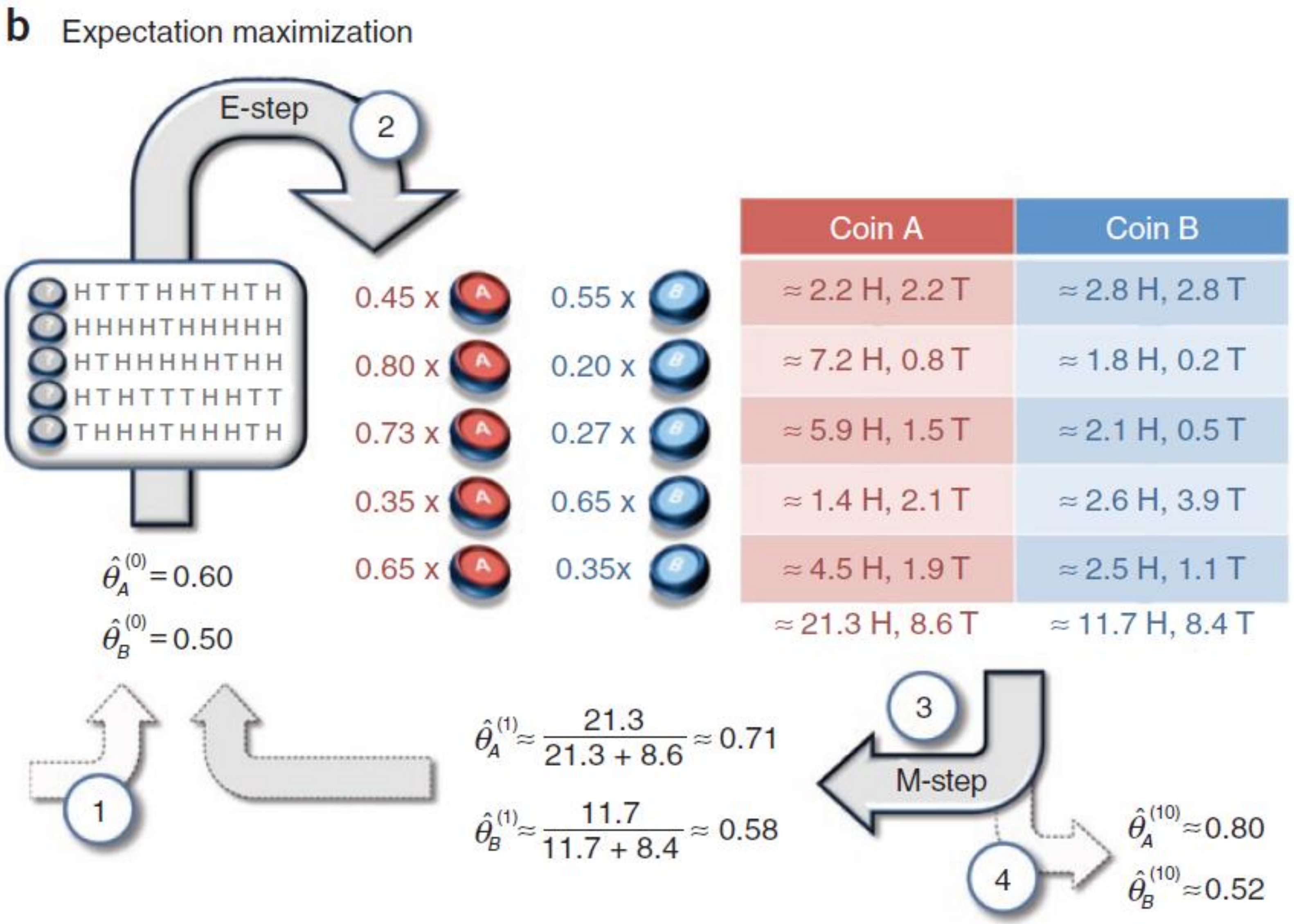
Do, C., Batzoglou, S. What is the expectation maximization algorithm?. *Nat Biotechnol* **26**, 897–899 (2008).

EM算法



Do, C., Batzoglou, S. What is the expectation maximization algorithm?. *Nat Biotechnol* **26**, 897–899 (2008).

EM算法



Do, C., Batzoglou, S. What is the expectation maximization algorithm?. *Nat Biotechnol* **26**, 897–899 (2008).