Machine Learning for Data Science (CS4786)
Lecture 15

Gaussian Mixture Model and EM Algorithm

COVID 19 Announcement

COVID 19 Announcement

- All classes at Cornell (and most universities) to go to go virtual

COVID 19 Announcement

- All classes at Cornell (and most universities) to go to go virtual
- We will continue with classes as usual, only online

COVID 19 Announcement

- All classes at Cornell (and most universities) to go to go virtual
- We will continue with classes as usual, only online
- Break up of grades remain the same

COVID 19 Announcement

COVID 19 Announcement

- From next lecture, Mar 17th, till end of semester, we will have virtual classes due to COVID 19

COVID 19 Announcement

- From next lecture, Mar 17th, till end of semester, we will have virtual classes due to COVID 19
  - I am thinking zoom for now (lec. Can be recorded as well)

COVID 19 Announcement

- From next lecture, Mar 17th, till end of semester, we will have virtual classes due to COVID 19
    - I am thinking zoom for now (lec. Can be recorded as well)
- Prelims postponed to 1 week after spring break (will be a virtual/online one as well)

COVID 19 Announcement

- From next lecture, Mar 17th, till end of semester, we will have virtual classes due to COVID 19
    - I am thinking zoom for now (lec. Can be recorded as well)
- Prelims postponed to 1 week after spring break (will be a virtual/online one as well)
- Finals will be a virtual/online one as well

COVID 19 Announcement

- From next lecture, Mar 17th, till end of semester, we will have virtual classes due to COVID 19
    - I am thinking zoom for now (lec. Can be recorded as well)
- Prelims postponed to 1 week after spring break (will be a virtual/online one as well)
- Finals will be a virtual/online one as well
- HWs and competition as planned

K-means Clustering

- For all $j \in [K]$, initialize cluster centroids $\hat{\mathbf{r}}_j^0$ randomly and set $m = 1$
- Repeat until convergence (or until patience runs out)
  1. For each $t \in \{1, \ldots, n\}$, set cluster identity of the point

$$\hat{c}^m(\mathbf{x}_t) = \underset{j \in [K]}{\operatorname{argmin}} \left\|\mathbf{x}_t - \hat{\mathbf{r}}_j^{m-1}\right\|$$

  2. For each $j \in [K]$, set new representative as

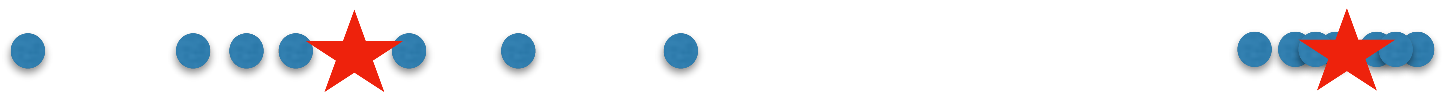
$$\hat{\mathbf{r}}_j^m = \frac{1}{|\hat{C}_j^m|} \sum_{\mathbf{x}_t \in \hat{C}_j^m} \mathbf{x}_t$$

  3. $m \leftarrow m + 1$

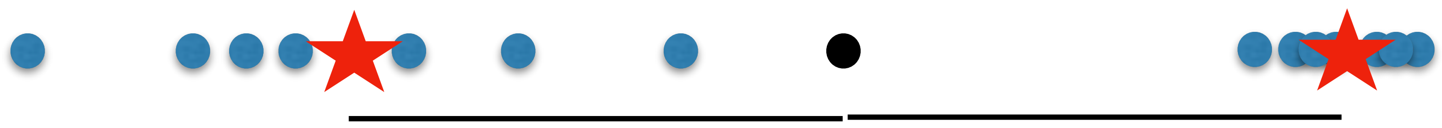
Variance and Radius

Variance and Radius

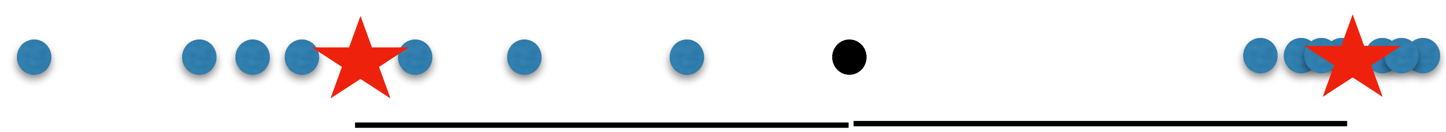
Variance and Radius



Variance and Radius

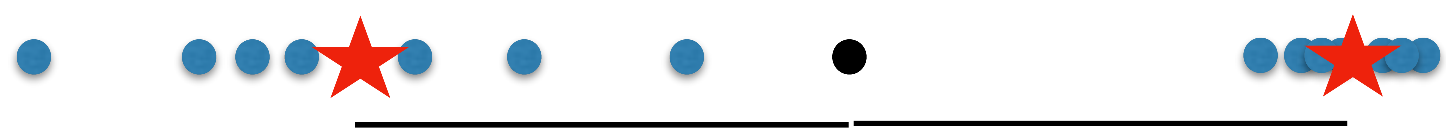


Variance and Radius



Distance to mean 1 should be smaller than distance to mean 2 as black dot is more likely in cluster 1 than 2

Variance and Radius

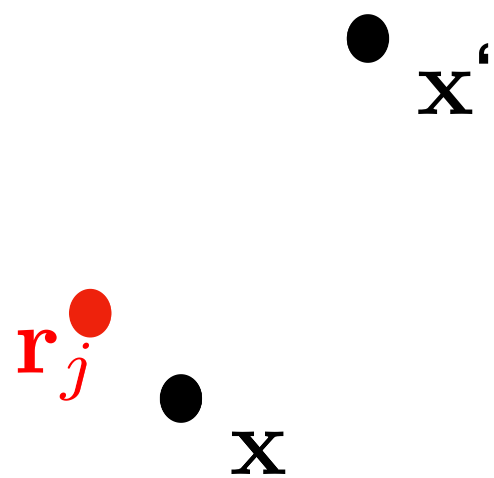


Distance to mean 1 should be smaller than distance to mean 2 as black dot is more likely in cluster 1 than 2

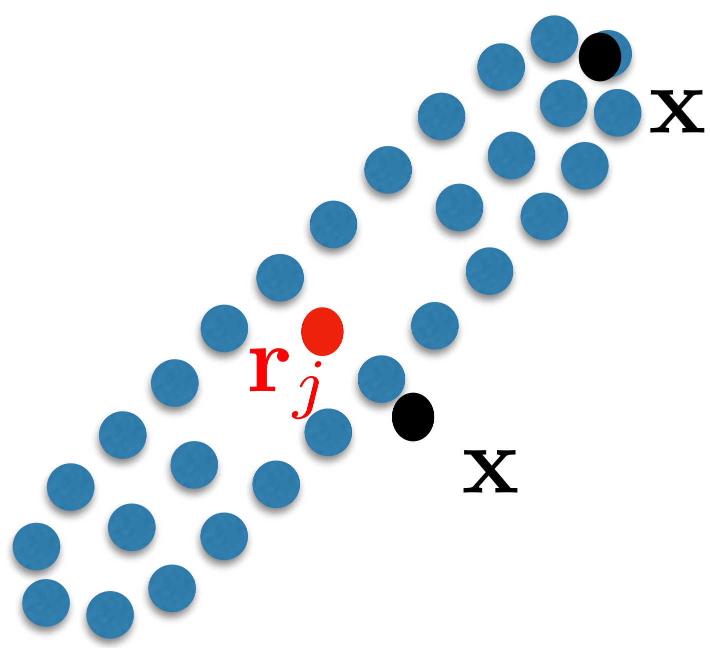
$$d^2(x, C_j) = \frac{(x-\mu_j)^2}{\sigma_j^2}$$

General Ellipsoid

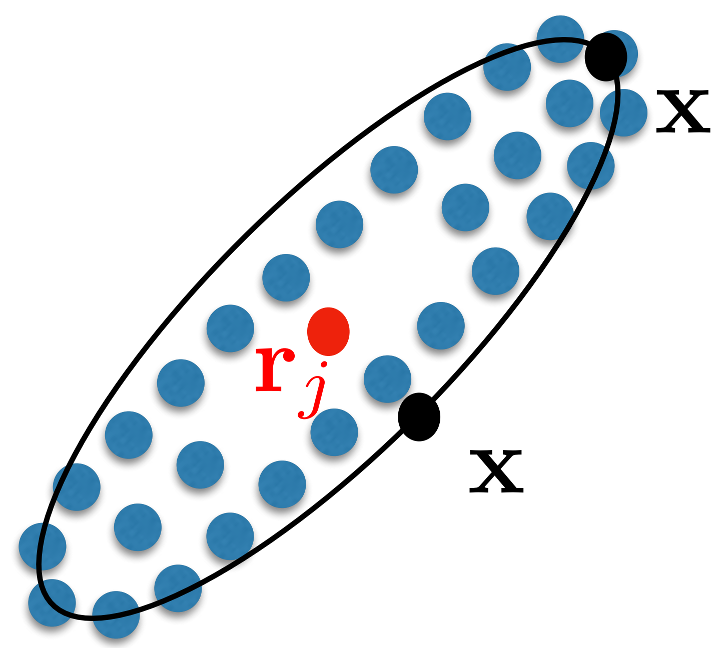
General Ellipsoid



General Ellipsoid



General Ellipsoid



$$(\mathbf{x} - \mathbf{r}_j)^\top \Sigma^{-1} (\mathbf{x} - \mathbf{r}_j)$$

$$\Sigma = \frac{1}{|C_j|} \sum_{t \in C_j} (\mathbf{x}_t - \mathbf{r}_j)(\mathbf{x}_t - \mathbf{r}_j)^\top$$

Ellipsoidal Clustering

- For all $j \in [K]$, initialize cluster centroids $\hat{\mathbf{r}}_j^0$ and ellipsoids $\hat{\Sigma}_j^0$ randomly and set $m = 1$
- Repeat until convergence (or until patience runs out)
  1. For each $t \in \{1, \ldots, n\}$, set cluster identity of the point

$$\hat{c}^m(\mathbf{x}_t) = \underset{j \in [K]}{\operatorname{argmin}} \; (\mathbf{x}_t - \hat{\mathbf{r}}_j^{m-1})^\top \left(\hat{\Sigma}^{m-1}\right)^{-1} (\mathbf{x}_t - \hat{\mathbf{r}}_j^{m-1})$$

  2. For each $j \in [K]$, set new representative as

$$\hat{\mathbf{r}}_j^m = \frac{1}{|\hat{C}_j^m|} \sum_{\mathbf{x}_t \in \hat{C}_j^m} \mathbf{x}_t \qquad \hat{\Sigma}^m = \frac{1}{|C_j|} \sum_{t \in C_j} (\mathbf{x}_t - \hat{\mathbf{r}}_j^m)(\mathbf{x}_t - \hat{\mathbf{r}}_j^m)^\top$$

  3. $m \leftarrow m + 1$

Ellipsoidal Clustering

- For all $j \in [K]$, initialize cluster centroids $\hat{\mathbf{r}}_j^0$ and ellipsoids $\hat{\Sigma}_j^0$ randomly and set $m = 1$
- Repeat until convergence (or until patience runs out)
  1. For each $t \in \{1, \ldots, n\}$, set cluster identity of the point

$$\hat{c}^m(\mathbf{x}_t) = \underset{j \in [K]}{\operatorname{argmin}} \ (\mathbf{x}_t - \hat{\mathbf{r}}_j^{m-1})^\top \left(\hat{\Sigma}^{m-1}\right)^{-1} (\mathbf{x}_t - \hat{\mathbf{r}}_j^{m-1})$$

$$d(\mathbf{x}_t, C_j)$$

  2. For each $j \in [K]$, set new representative as

$$\hat{\mathbf{r}}_j^m = \frac{1}{|\hat{C}_j^m|} \sum_{\mathbf{x}_t \in \hat{C}_j^m} \mathbf{x}_t \qquad \hat{\Sigma}^m = \frac{1}{|C_j|} \sum_{t \in C_j} (\mathbf{x}_t - \hat{\mathbf{r}}_j^m)(\mathbf{x}_t - \hat{\mathbf{r}}_j^m)^\top$$

  3. $m \leftarrow m + 1$

K-means: pitfalls

- Looks for spherical clusters
- Of same radius
- And with roughly equal number of points

K-means: pitfalls

- Looks for spherical clusters ✓
- Of same radius
- And with roughly equal number of points

K-means: pitfalls

- Looks for spherical clusters
- Of same radius
- And with roughly equal number of points

K-means: pitfalls

- Looks for spherical clusters ✓
- Of same radius ✓
- And with roughly equal number of points ✗

Hard Gaussian Mixture Model

- For all $j \in [K]$, initialize cluster centroids $\hat{\mathbf{r}}_j^0$, ellipsoids $\hat{\Sigma}_j^0$ and initial proportions π^0 randomly and set $m = 1$
- Repeat until convergence (or until patience runs out)
  1. For each $t \in \{1, \ldots, n\}$, set cluster identity of the point

$$\hat{c}^m(\mathbf{x}_t) = \underset{j \in [K]}{\operatorname{argmin}} \; (\mathbf{x}_t - \hat{\mathbf{r}}_j^{m-1})^\top \left(\hat{\Sigma}^{m-1}\right)^{-1} (\mathbf{x}_t - \hat{\mathbf{r}}_j^{m-1}) - \log(\pi_j^{m-1})$$

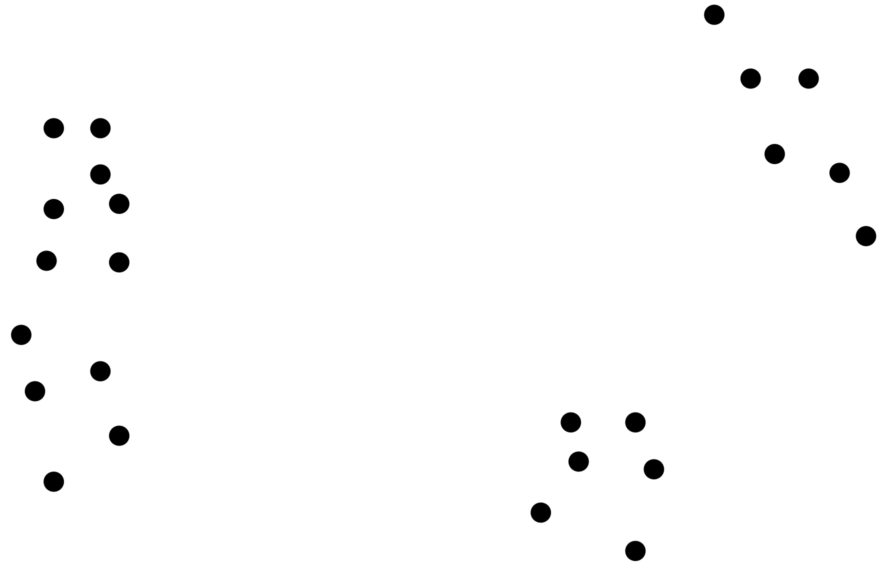
Penalty for smaller clusters

  2. For each $j \in [K]$, set new representative as

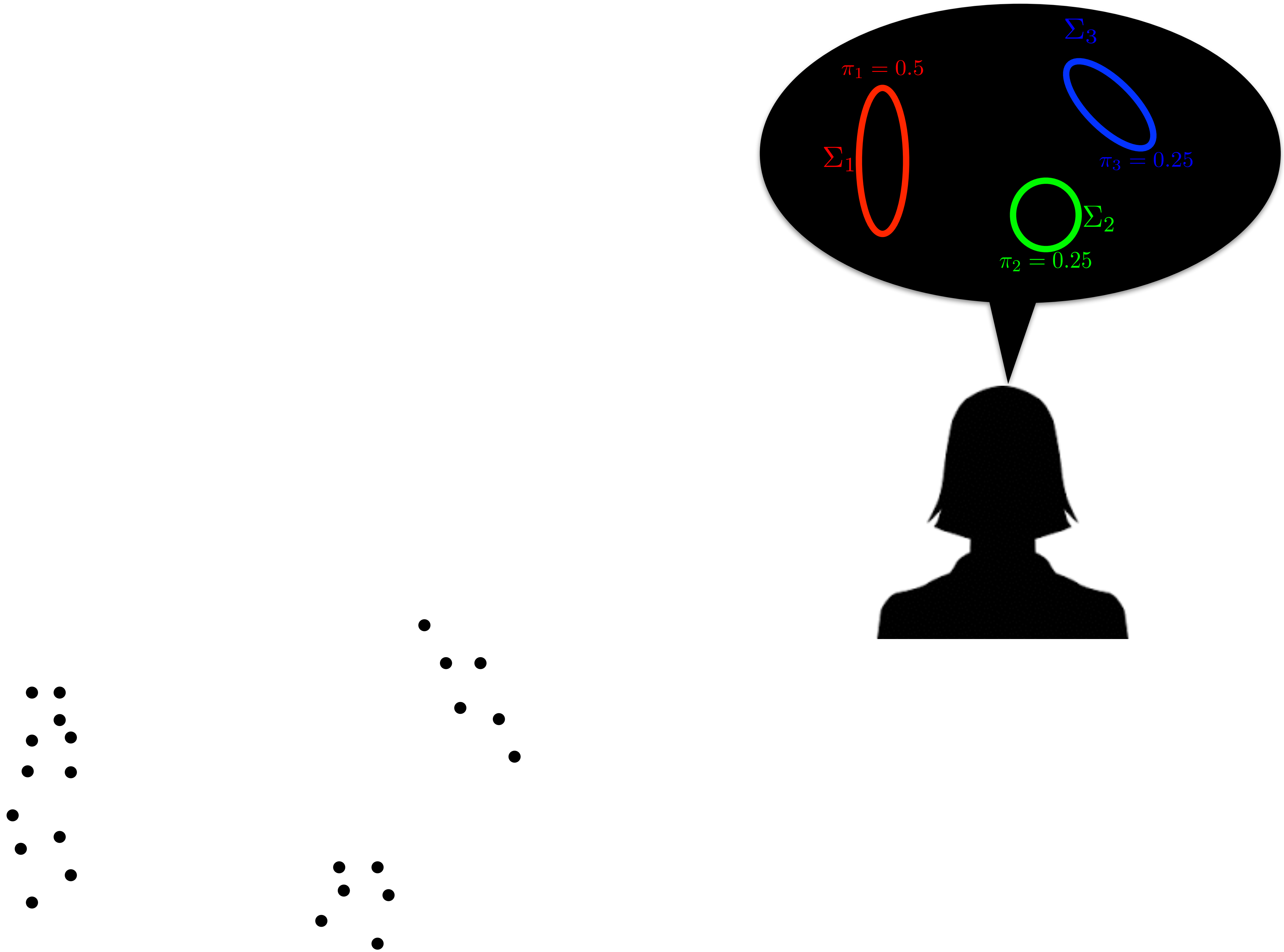
$$\hat{\mathbf{r}}_j^m = \frac{1}{|\hat{C}_j^m|} \sum_{\mathbf{x}_t \in \hat{C}_j^m} \mathbf{x}_t \qquad \hat{\Sigma}^m = \frac{1}{|C_j|} \sum_{t \in C_j} (\mathbf{x}_t - \hat{\mathbf{r}}_j^m)(\mathbf{x}_t - \hat{\mathbf{r}}_j^m)^\top \qquad \pi_j^m = \frac{|C_j^m|}{n}$$

  3. $m \leftarrow m + 1$

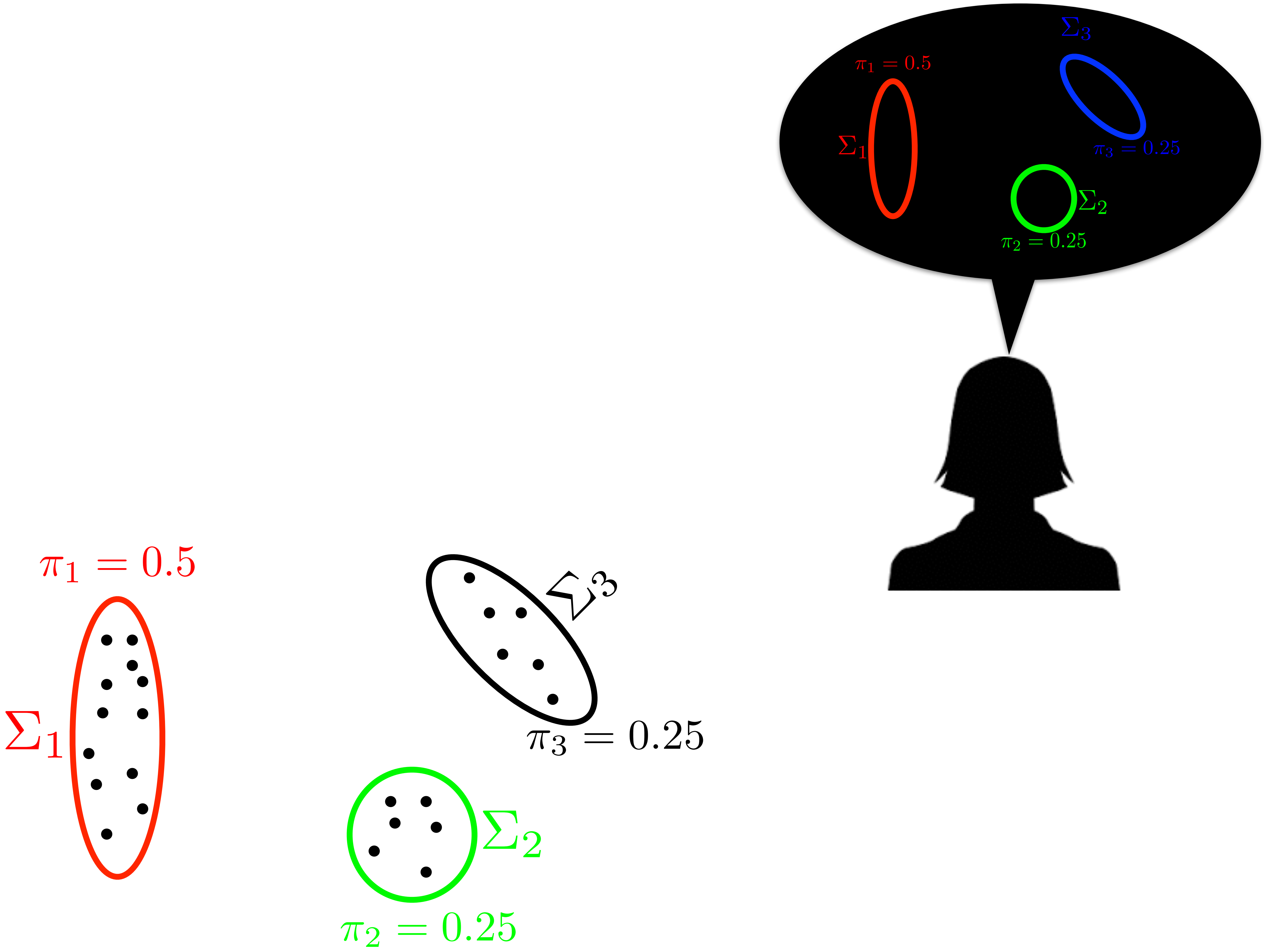
Probabilistic Model



Probabilistic Model



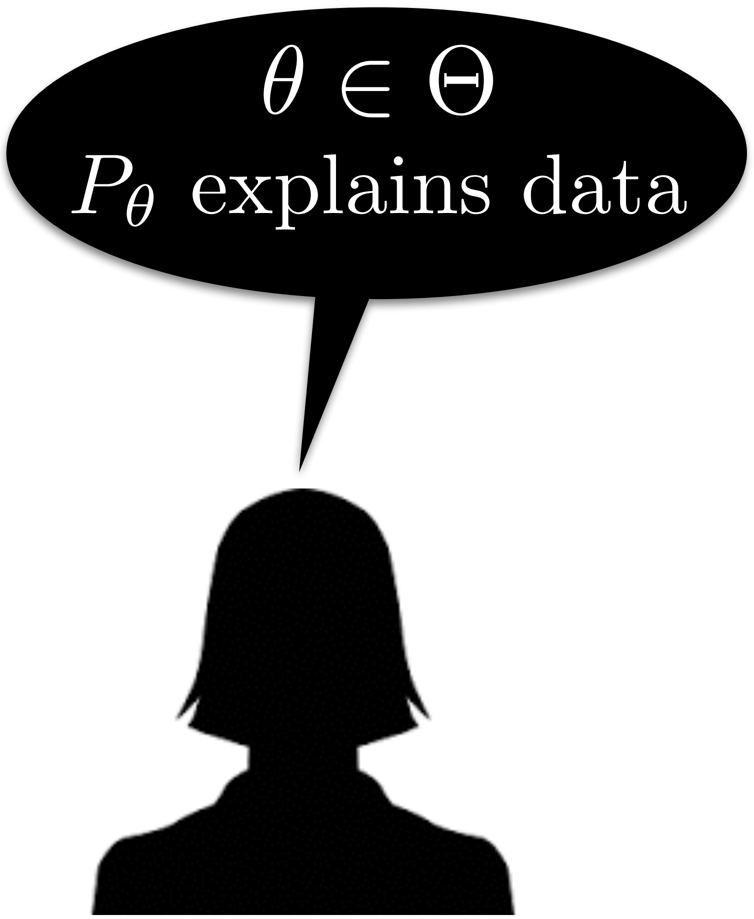
Probabilistic Model



Probabilistic Model

Data: $\mathbf{x}_1 \ldots, \mathbf{x}_n$

Probabilistic Model



Data: $\mathbf{x}_1, \ldots, \mathbf{x}_n$

Probabilistic Models

- Θ consists of set of possible parameters
- We have a distribution P_θ over the data induced by each $\theta \in \Theta$
- Data is generated by one of the $\theta \in \Theta$
- Learning: Estimate value or distribution for $\theta^* \in \Theta$ given data

Maximum Likelihood Principal

Pick $\theta \in \Theta$ that maximizes probability of observation

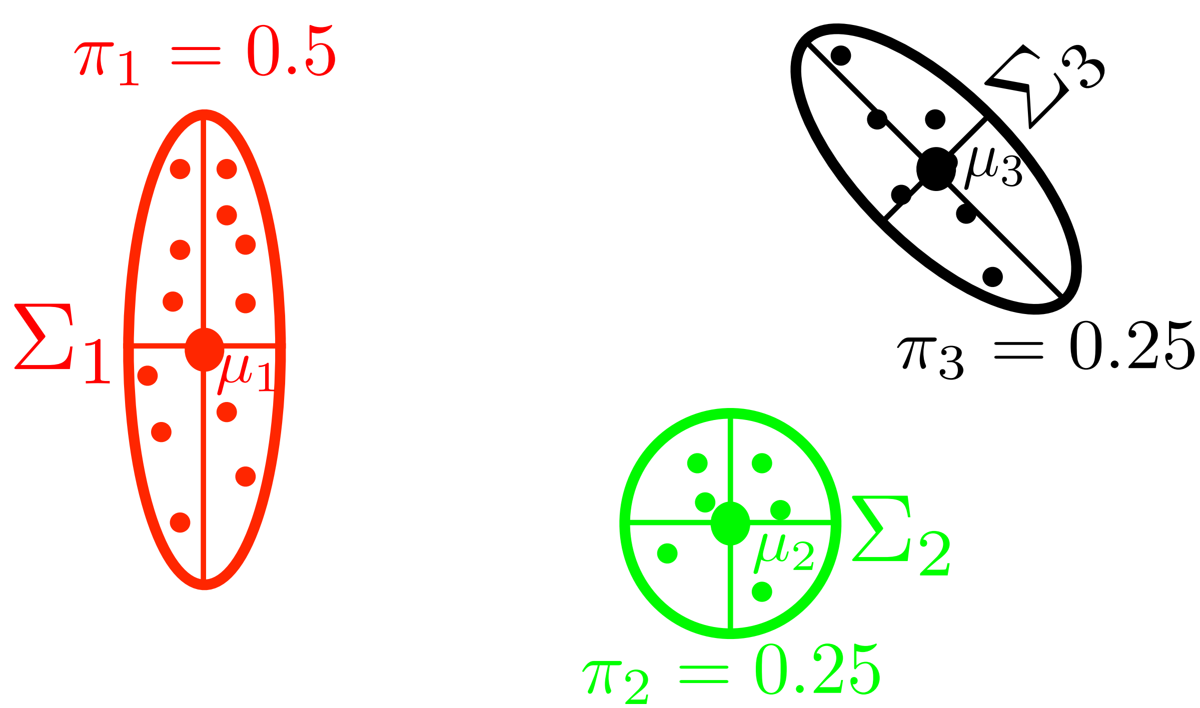
$$\theta_{MLE} = \operatorname{argmax}_{\theta \in \Theta} \log \underbrace{P_\theta(x_1, \ldots, x_n)}_{\text{Likelihood}}$$

Gaussian Mixture Models

Each $\theta \in \Theta$ is a model.

- Gaussian Mixture Model
  - Each θ consists of mixture distribution $\pi = (\pi_1, \ldots, \pi_K)$, means $\mu_1, \ldots, \mu_K \in \mathbb{R}^d$ and covariance matrices $\Sigma_1, \ldots, \Sigma_K$
  - For each t, independently:

$$c_t \sim \pi, \qquad x_t \sim N(\mu_{c_t}, \Sigma_{c_t})$$



Example: Gaussian Mixture Model

What is the likelihood for Gaussian Mixture Models?

What is the likelihood for one point x under model?

Example: Gaussian Mixture Model

MLE: $\theta = (\mu_1, \ldots, \mu_K), \pi, \Sigma$

$$P_\theta(x_1, \ldots, x_n) = \prod_{t=1}^{n} \left(\sum_{i=1}^{K} \pi_i \frac{1}{\sqrt{(2 * 3.1415)^2 |\Sigma_i|}} \exp\left(-(x_t - \mu_i)^\top \Sigma_i (x_t - \mu_i)\right) \right)$$

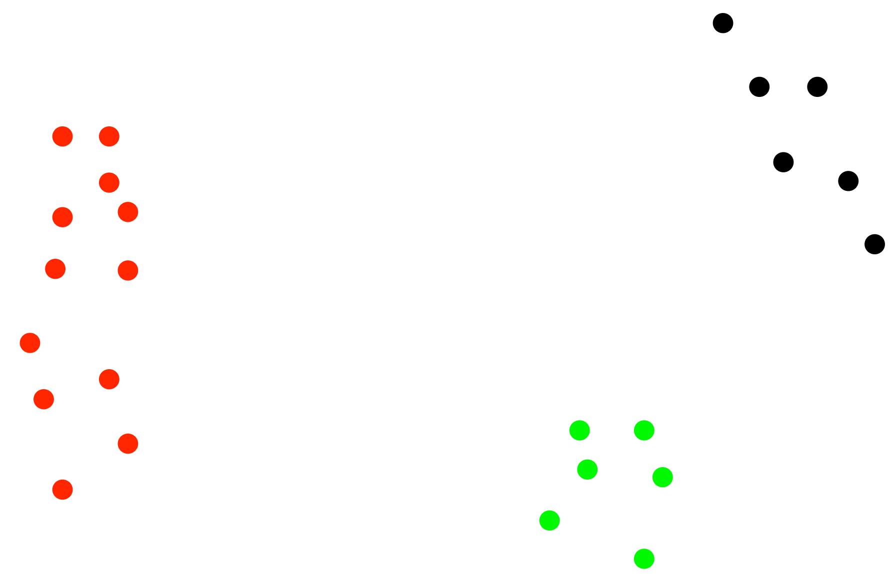
Find θ that maximizes $\log P_\theta(x_1, \ldots, x_n)$

Example: Gaussian Mixture Model

Directly optimizing is hard!

MLE for GMM

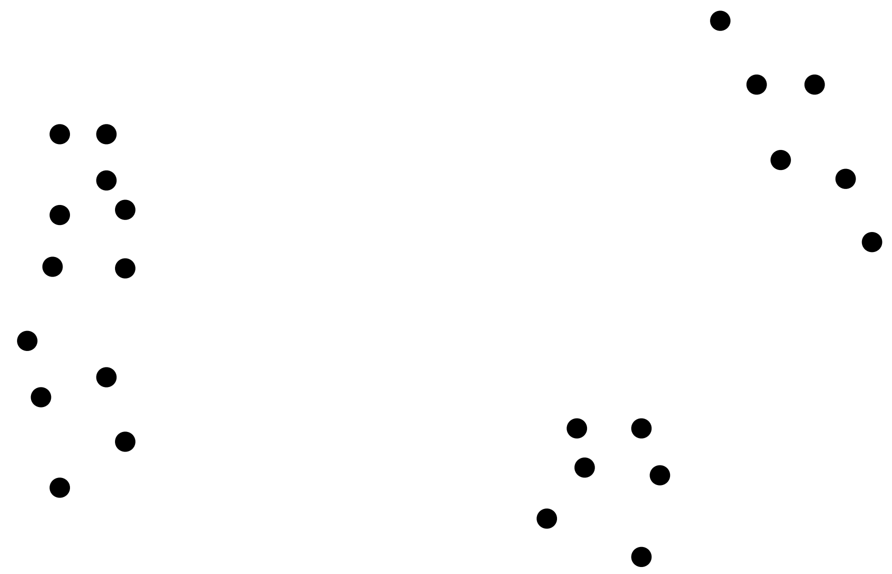
Say by some magic you knew cluster assignments, then



How would you compute parameters ?

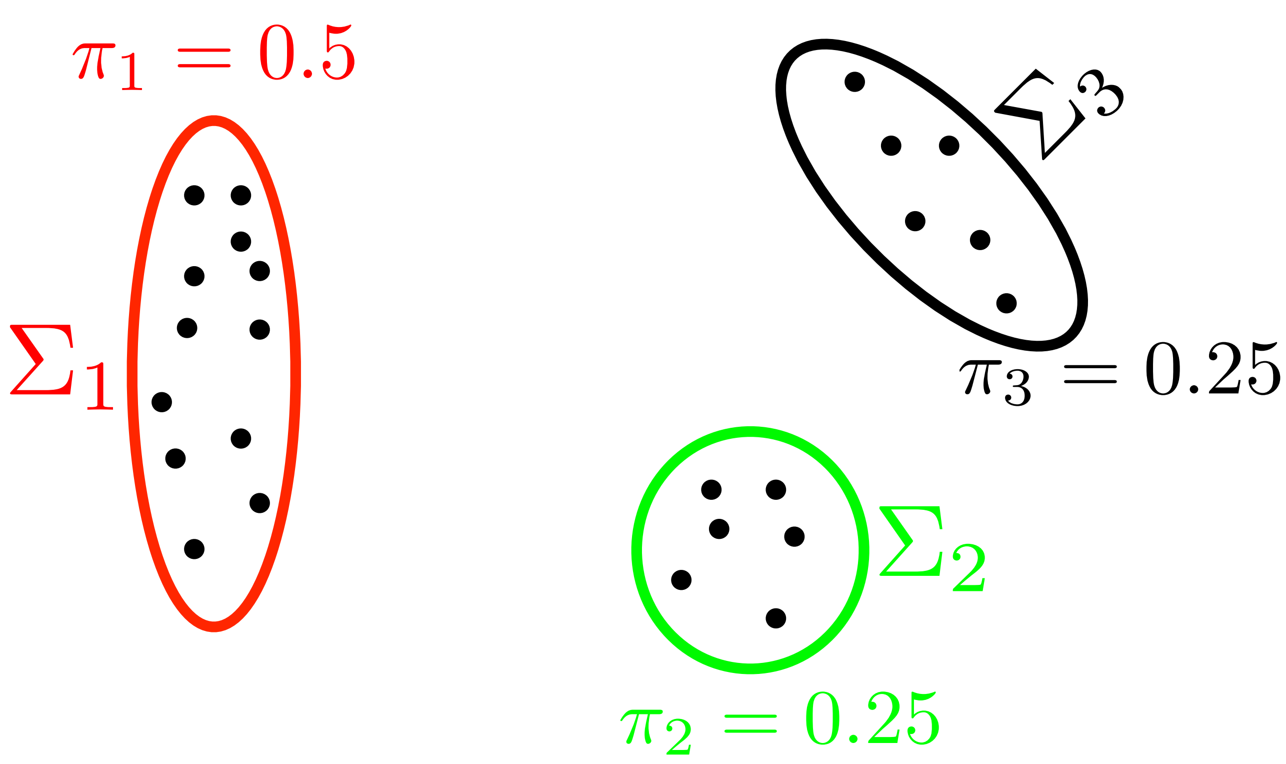
MLE for GMM

Say we knew model parameters, how do we assign clusters?



MLE for GMM

Say we knew model parameters, how do we assign clusters?



Hard Gaussian Mixture Model

- For all $j \in [K]$, initialize cluster centroids $\hat{\mathbf{r}}_j^0$, ellipsoids $\hat{\Sigma}_j^0$ and initial proportions π^0 randomly and set $m = 1$
- Repeat until convergence (or until patience runs out)
  1. For each $t \in \{1, \ldots, n\}$, set cluster identity of the point

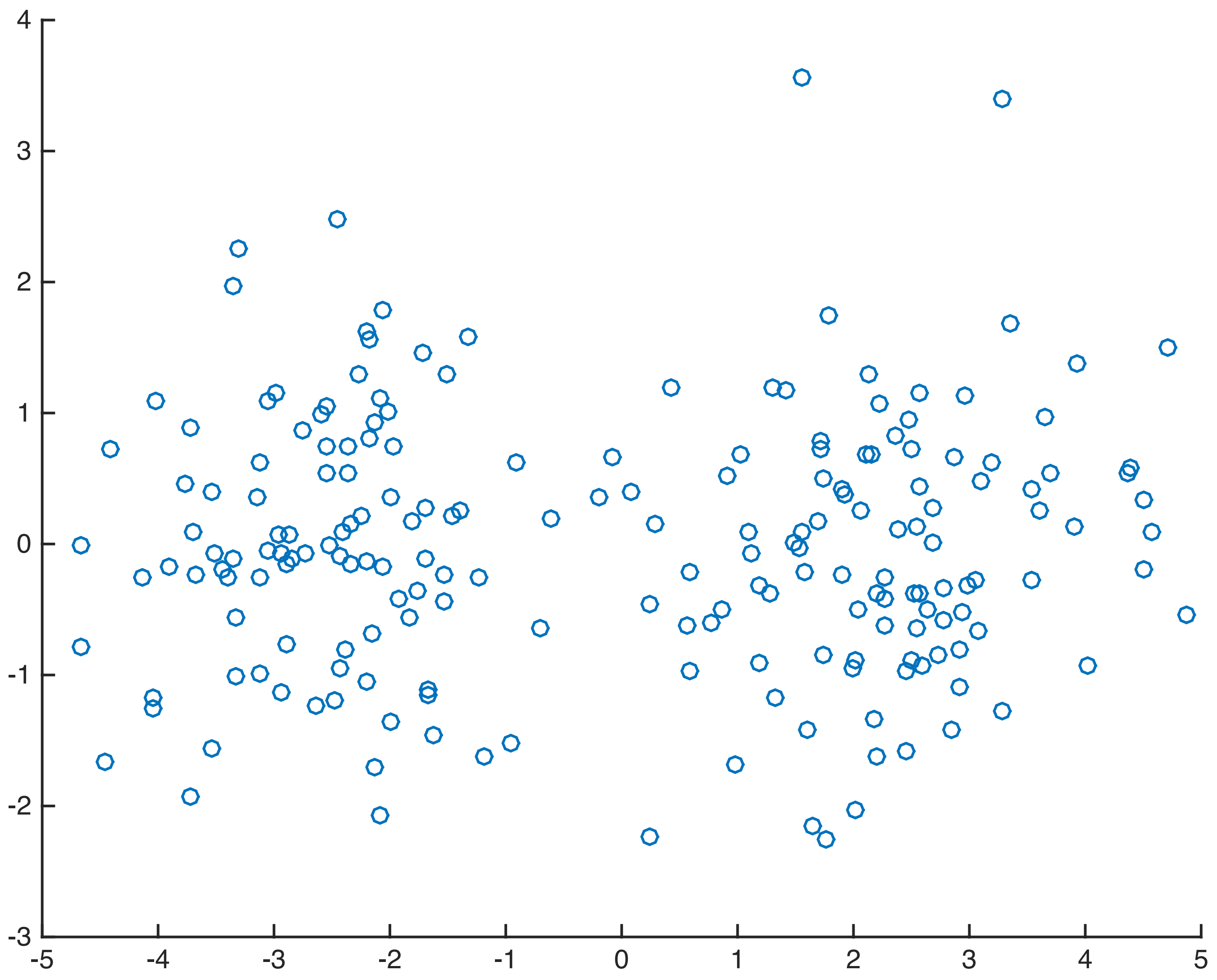
     $$\hat{c}^m(\mathbf{x}_t) = \arg\max_{j \in [K]} \; p(\mathbf{x}_t, \hat{\mathbf{r}}_j^{m-1}, \hat{\Sigma}^{m-1}) \times \pi^m(j)$$

  2. For each $j \in [K]$, set new representative as

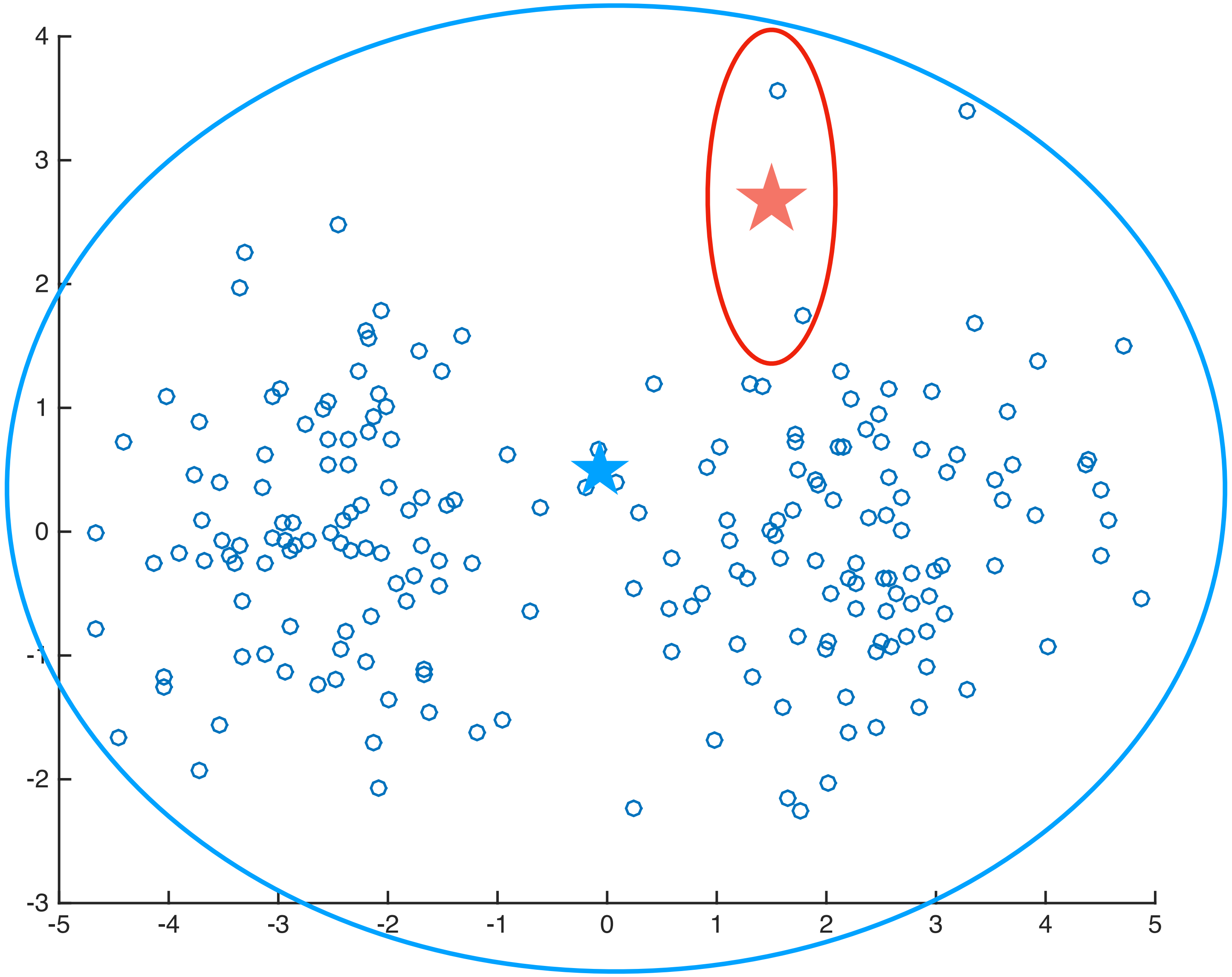
     $$\hat{\mathbf{r}}_j^m = \frac{1}{|\hat{C}_j^m|} \sum_{\mathbf{x}_t \in \hat{C}_j^m} \mathbf{x}_t \qquad \hat{\Sigma}^m = \frac{1}{|C_j|} \sum_{t \in C_j} (\mathbf{x}_t - \hat{\mathbf{r}}_j^m)(\mathbf{x}_t - \hat{\mathbf{r}}_j^m)^\top \qquad \pi_j^m = \frac{|C_j^m|}{n}$$

  3. $m \leftarrow m + 1$

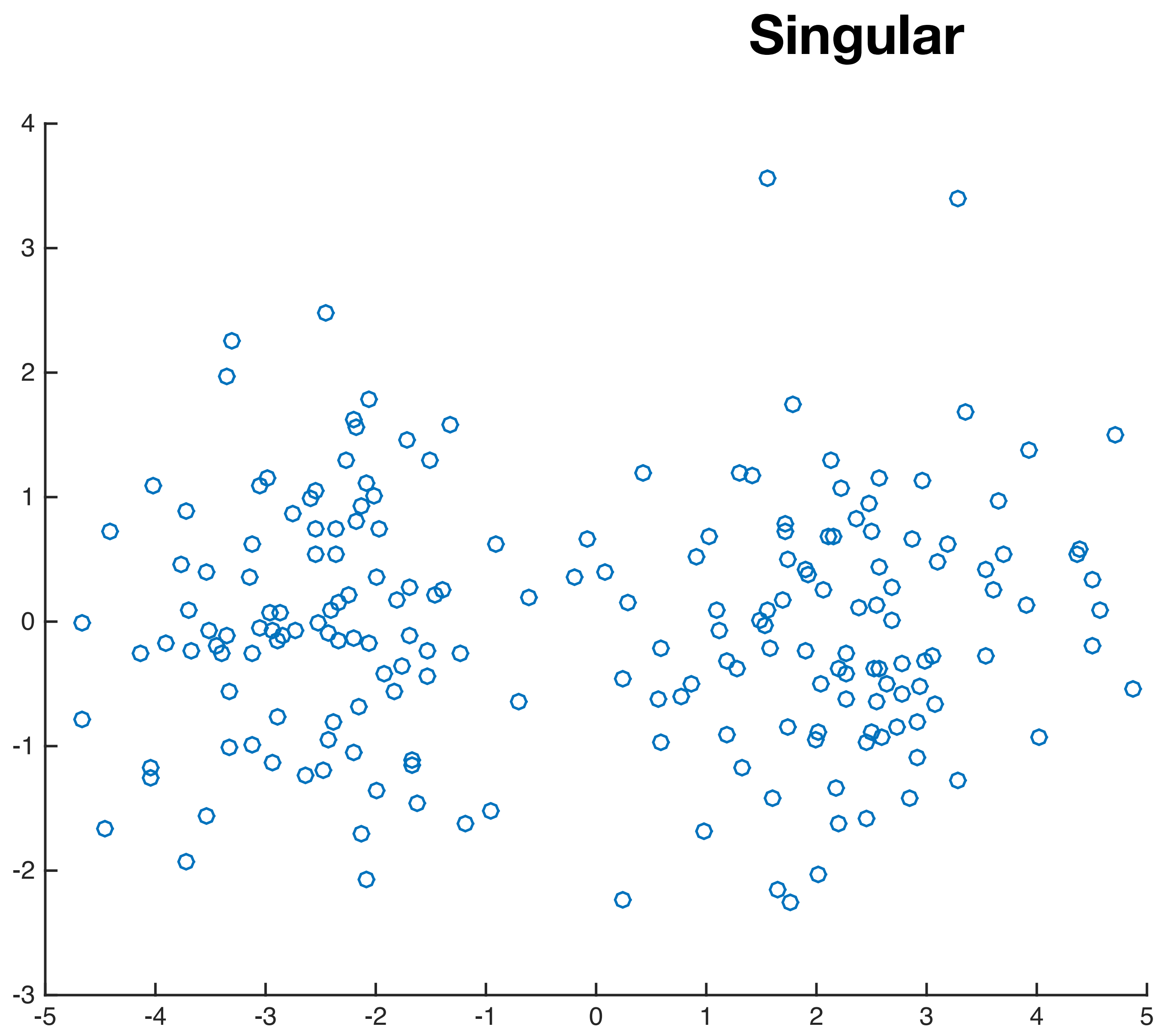
Pitfall of Hard Assignment



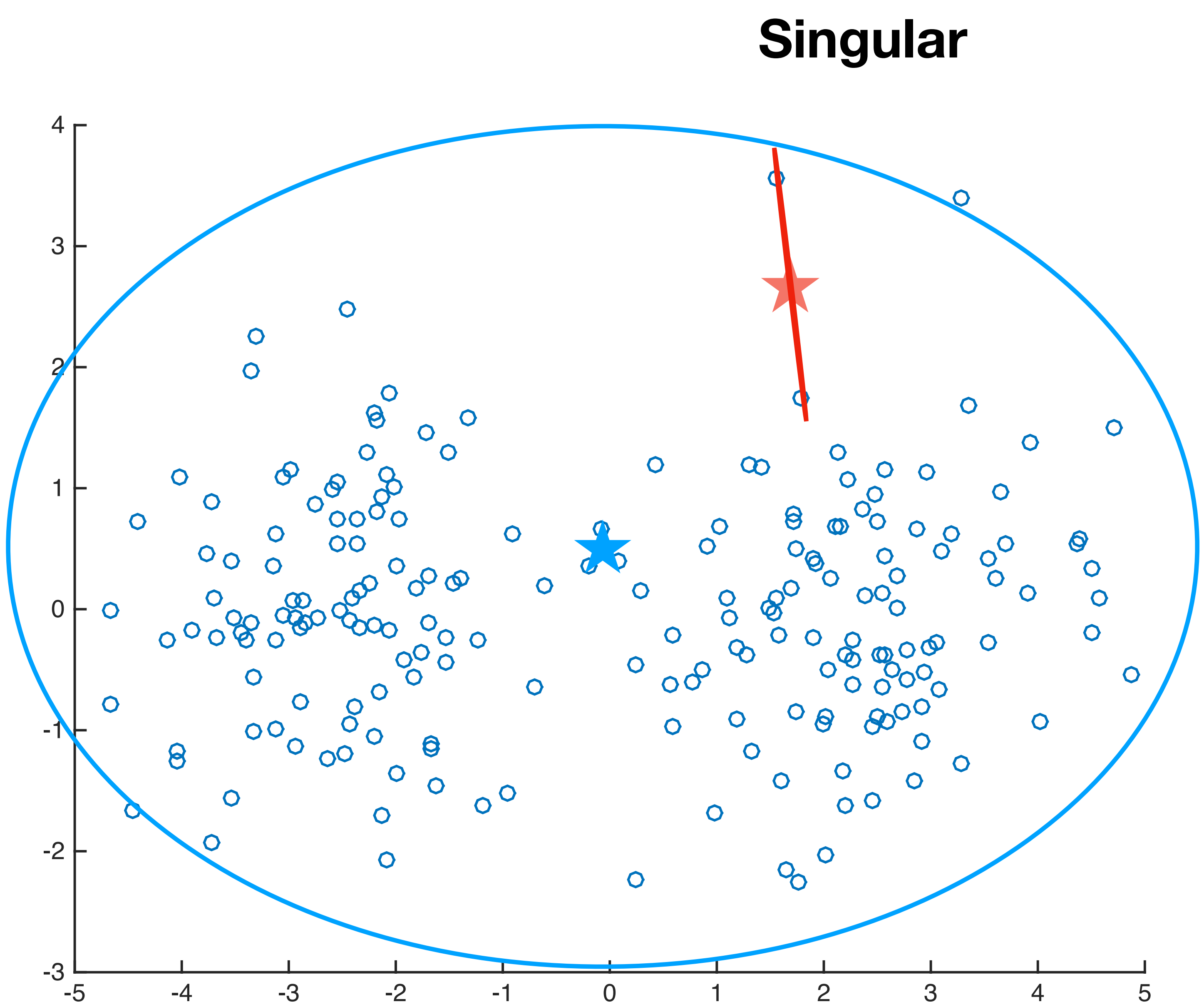
Pitfall of Hard Assignment



Pitfall of Hard Assignment



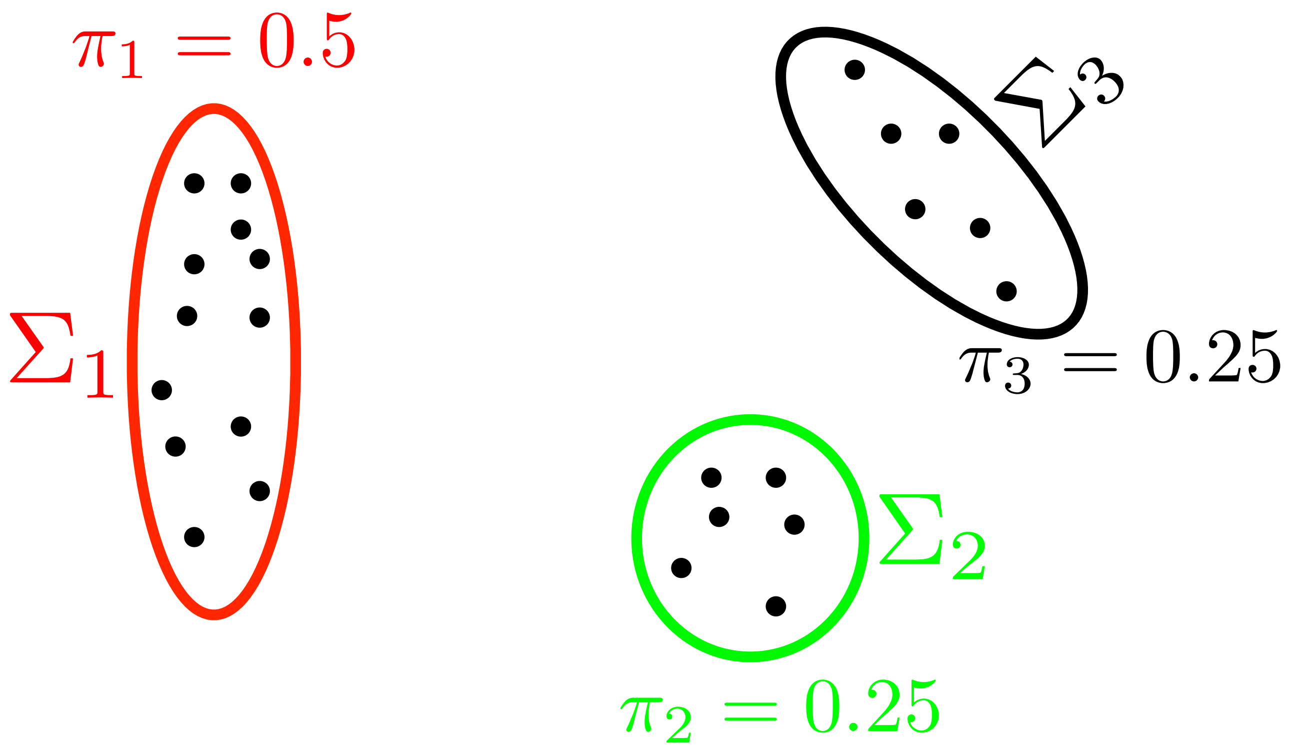
Pitfall of Hard Assignment



MLE for GMM

Say we knew model parameters, how do we assign clusters?

Given probability of each point belonging to each of the clusters, how do we compute model parameters?

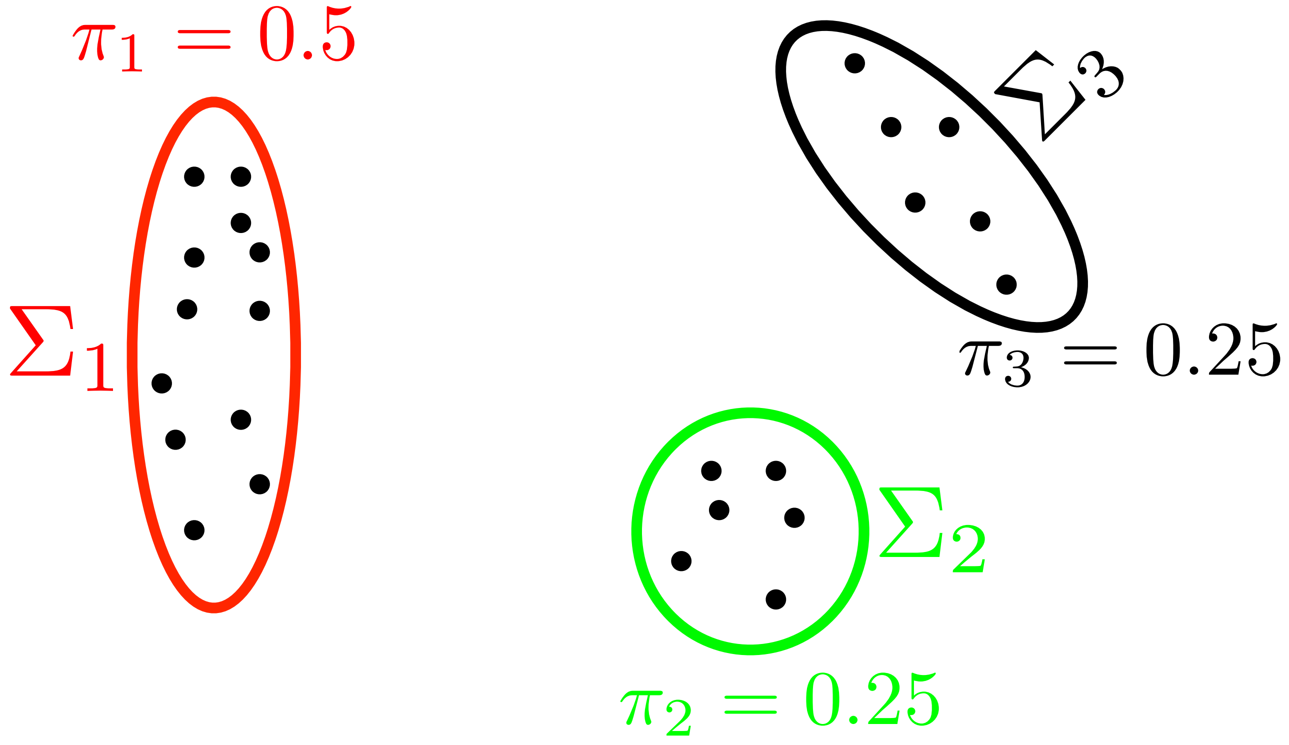


MLE FOR GMM

Say we knew model parameters, ~~how do we assign clusters?~~

what are the probabilities of points falling in each of the clusters?

Given probability of each point belonging to each of the clusters, how do we compute model parameters?



(Soft) Gaussian Mixture Model

- For all $j \in [K]$, initialize cluster centroids $\hat{\mathbf{r}}_j^0$ and ellipsoids $\hat{\Sigma}_j^0$ randomly and set $m = 1$
- Repeat until convergence (or until patience runs out)
  1. For each $t \in \{1, \ldots, n\}$, set cluster identity of the point

$$Q_t^m(j) = p(\mathbf{x}_t, \hat{\mathbf{r}}_j^{m-1}, \hat{\Sigma}^{m-1}) \times \pi^m(j)$$

  2. For each $j \in [K]$, set new representative as

$$\hat{\mathbf{r}}_j^m = \frac{\sum_{t=1}^n Q_t(j)\mathbf{x}_t}{\sum_{t=1}^n Q_t(j)} \qquad \hat{\Sigma}^m = \frac{\sum_{t=1}^n Q_t(j)(\mathbf{x}_t - \hat{\mathbf{r}}_j^m)(\mathbf{x}_t - \hat{\mathbf{r}}_j^m)^\top}{\sum_{t=1}^n Q_t(j)}$$

$$\pi_j^m = \frac{\sum_{t=1}^n Q_t(j)}{n}$$

  3. $m \leftarrow m + 1$

Expectation Maximization Algorithm

- For demonstration we shall consider the problem of finding MLE (MAP version is very similar)
- Initialize $\theta^{(0)}$ arbitrarily, repeat unit convergence:

(E step) For every t, define distribution Q_t over the latent variable c_t as:

$$Q_t^{(i)}(c_t) = P(c_t|x_t, \theta^{(i-1)})$$

(M step)

$$\theta^{(i)} = \text{argmax}_{\theta \in \Theta} \sum_{t=1}^{n} \sum_{c_t} Q_t^{(i)}(c_t) \log P(x_t, c_t|\theta)$$

Example: EM for GMM

- E step: For every $k \in [K]$,

$$Q_t^{(i)}(c_t = k) = P\left(c_t = k | x_t, \theta^{(i-1)}\right) = P\left(x_t | c_t = k, \theta^{(i-1)}\right) \times P\left(c_t = k | \theta^{(i-1)}\right)$$

$$\propto \underbrace{\phi\left(x_t; \mu_k^{(i-1)}, \Sigma_k^{(i-1)}\right)}_{\text{gaussian p.d.f.}} \times \pi_k^{(i-1)}$$

Example: EM for GMM

- E step: For every $k \in [K]$,

$$Q_t^{(i)}(c_t = k) = P\left(c_t = k | x_t, \theta^{(i-1)}\right) = P\left(x_t | c_t = k, \theta^{(i-1)}\right) \times P\left(c_t = k | \theta^{(i-1)}\right)$$
$$\propto \underbrace{\phi\left(x_t; \mu_k^{(i-1)}, \Sigma_k^{(i-1)}\right)}_{\text{gaussian p.d.f.}} \times \pi_k^{(i-1)}$$

- M step: Given $Q_1, \ldots, Q_n$, we need to find

$$\theta^{(i)} = \underset{\theta \in \Theta}{\operatorname{argmax}} \sum_{t=1}^{n} \sum_{k=1}^{K} Q_t^{(i)}(k) \log P(x_t, c_t = k | \theta)$$
$$= \underset{\theta}{\operatorname{argmax}} \sum_{t=1}^{n} \sum_{k=1}^{K} Q_t^{(i)}(k) \left(\log P(x_t | c_t = k, \theta) + \log P(c_t = k | \theta)\right)$$
$$= \underset{\pi, \mu_{1,\ldots,K}, \Sigma_{1,\ldots,K}}{\operatorname{argmax}} \sum_{t=1}^{n} \sum_{c_t=1}^{K} Q_t^{(i)}(k) \left(\log \phi\left(x_t; \mu_k, \Sigma_k\right) + \log \pi_k\right)$$

Example: EM for GMM

For every $k \in [K]$, the maximization step yields,

$$\mu_k^{(i)} = \frac{\sum_{t=1}^n Q_t^{(i)}(k) x_t}{\sum_{t=1}^n Q_t(k)}, \quad \Sigma_k^{(i)} = \frac{\sum_{t=1}^n Q_t^{(i)}(k) \left(x_t - \mu_k^{(i)}\right)\left(x_t - \mu_k^{(i)}\right)^\top}{\sum_{t=1}^n Q_t(k)}$$

$$\pi_k^{(i)} = \frac{\sum_{t=1}^n Q_t^{(i)}(k)}{n}$$

Why should EM work?

A very high level view:

- Performing E-step will never decrease log-likelihood (or log a posteriori)

Why should EM work?

A very high level view:

- Performing E-step will never decrease log-likelihood (or log a posteriori)
- Performing M-step will never decrease log-likelihood (or log a posteriori)

Why should EM work?

Steps to show that $\log \text{Lik}(\theta^{(i)}) \geq \log \text{Lik}(\theta^{(i-1)})$:

$$\log P_{\theta^{(i)}}(x_1, \ldots, x_n)$$

WHY SHOULD EM WORK?

Steps to show that $\log \mathrm{Lik}(\theta^{(i)}) \geq \log \mathrm{Lik}(\theta^{(i-1)})$:

$$\log P_{\theta^{(i)}}(x_1, \ldots, x_n) = \sum_{t=1}^{n} \log P_{\theta^{(i)}}(x_t)$$

Why should EM work?

Steps to show that $\log \text{Lik}(\theta^{(i)}) \geq \log \text{Lik}(\theta^{(i-1)})$:

$$
\begin{aligned}
\log P_{\theta^{(i)}}(x_1, \ldots, x_n) &= \sum_{t=1}^{n} \log P_{\theta^{(i)}}(x_t) \\
&= \sum_{t=1}^{n} \log \left(\sum_{c_t=1}^{K} P_{\theta^{(i)}}(x_t, c_t) \right)
\end{aligned}
$$

WHY SHOULD EM WORK?

Steps to show that $\log \mathrm{Lik}(\theta^{(i)}) \geq \log \mathrm{Lik}(\theta^{(i-1)})$:

$$
\begin{aligned}
\log P_{\theta^{(i)}}(x_1, \ldots, x_n) &= \sum_{t=1}^{n} \log P_{\theta^{(i)}}(x_t) \\
&= \sum_{t=1}^{n} \log \left(\sum_{c_t=1}^{K} P_{\theta^{(i)}}(x_t, c_t) \right) \\
&= \sum_{t=1}^{n} \log \left(\sum_{c_t=1}^{K} Q^{(i)}(c_t) \left(\frac{P_{\theta^{(i)}}(x_t, c_t)}{Q^{(i)}(c_t)} \right) \right)
\end{aligned}
$$

Why should EM work?

Steps to show that $\log \mathrm{Lik}(\theta^{(i)}) \geq \log \mathrm{Lik}(\theta^{(i-1)})$:

$$
\begin{aligned}
\log P_{\theta^{(i)}}(x_1, \ldots, x_n) &= \sum_{t=1}^{n} \log P_{\theta^{(i)}}(x_t) \\
&= \sum_{t=1}^{n} \log \left(\sum_{c_t=1}^{K} P_{\theta^{(i)}}(x_t, c_t) \right) \\
&= \sum_{t=1}^{n} \log \left(\sum_{c_t=1}^{K} Q^{(i)}(c_t) \left(\frac{P_{\theta^{(i)}}(x_t, c_t)}{Q^{(i)}(c_t)} \right) \right) \\
&\geq \sum_{t=1}^{n} \sum_{c_t=1}^{K} Q^{(i)}(c_t) \log \left(\frac{P_{\theta^{(i)}}(x_t, c_t)}{Q^{(i)}(c_t)} \right)
\end{aligned}
$$

WHY SHOULD EM WORK?

Steps to show that $\log \text{Lik}(\theta^{(i)}) \geq \log \text{Lik}(\theta^{(i-1)})$:

$$
\begin{aligned}
\log P_{\theta^{(i)}}(x_1, \ldots, x_n) &= \sum_{t=1}^{n} \log P_{\theta^{(i)}}(x_t) \\
&= \sum_{t=1}^{n} \log \left(\sum_{c_t=1}^{K} P_{\theta^{(i)}}(x_t, c_t) \right) \\
&= \sum_{t=1}^{n} \log \left(\sum_{c_t=1}^{K} Q^{(i)}(c_t) \left(\frac{P_{\theta^{(i)}}(x_t, c_t)}{Q^{(i)}(c_t)} \right) \right) \\
&\geq \sum_{t=1}^{n} \sum_{c_t=1}^{K} Q^{(i)}(c_t) \log \left(\frac{P_{\theta^{(i)}}(x_t, c_t)}{Q^{(i)}(c_t)} \right)
\end{aligned}
$$

Log(average) > average of Log

Why should EM work?

Steps to show that $\log \text{Lik}(\theta^{(i)}) \geq \log \text{Lik}(\theta^{(i-1)})$:

$$\log P_{\theta^{(i)}}(x_1, \ldots, x_n) \geq \sum_{t=1}^{n} \sum_{c_t=1}^{K} Q^{(i)}(c_t) \log\left(\frac{P_{\theta^{(i)}}(x_t, c_t)}{Q^{(i)}(c_t)}\right)$$

Why should EM work?

Steps to show that $\log \mathrm{Lik}(\theta^{(i)}) \geq \log \mathrm{Lik}(\theta^{(i-1)})$:

$$\log P_{\theta^{(i)}}(x_1, \ldots, x_n) \geq \sum_{t=1}^{n} \sum_{c_t=1}^{K} Q^{(i)}(c_t) \log \left(\frac{P_{\theta^{(i)}}(x_t, c_t)}{Q^{(i)}(c_t)} \right)$$

$$\geq \sum_{t=1}^{n} \sum_{c_t=1}^{K} Q^{(i)}(c_t) \log \left(\frac{P_{\theta^{(i-1)}}(x_t, c_t)}{Q^{(i)}(c_t)} \right)$$

M-step

WHY SHOULD EM WORK?

Steps to show that $\log \mathrm{Lik}(\theta^{(i)}) \geq \log \mathrm{Lik}(\theta^{(i-1)})$:

$$\log P_{\theta^{(i)}}(x_1, \ldots, x_n) \geq \sum_{t=1}^{n} \sum_{c_t=1}^{K} Q^{(i)}(c_t) \log \left(\frac{P_{\theta^{(i)}}(x_t, c_t)}{Q^{(i)}(c_t)} \right)$$

$$\geq \sum_{t=1}^{n} \sum_{c_t=1}^{K} Q^{(i)}(c_t) \log \left(\frac{P_{\theta^{(i-1)}}(x_t, c_t)}{Q^{(i)}(c_t)} \right)$$

M-step

$$= \sum_{t=1}^{n} \sum_{c_t=1}^{K} Q^{(i)}(c_t) \log \left(\frac{P_{\theta^{(i-1)}}(x_t, c_t)}{P_{\theta^{(i-1)}}(c_t | x_t)} \right)$$

E-step

WHY SHOULD EM WORK?

Steps to show that $\log \mathrm{Lik}(\theta^{(i)}) \geq \log \mathrm{Lik}(\theta^{(i-1)})$:

$$\log P_{\theta^{(i)}}(x_1, \ldots, x_n) \geq \sum_{t=1}^{n} \sum_{c_t=1}^{K} Q^{(i)}(c_t) \log \left(\frac{P_{\theta^{(i)}}(x_t, c_t)}{Q^{(i)}(c_t)} \right)$$

$$\geq \sum_{t=1}^{n} \sum_{c_t=1}^{K} Q^{(i)}(c_t) \log \left(\frac{P_{\theta^{(i-1)}}(x_t, c_t)}{Q^{(i)}(c_t)} \right)$$ **M-step**

$$= \sum_{t=1}^{n} \sum_{c_t=1}^{K} Q^{(i)}(c_t) \log \left(\frac{P_{\theta^{(i-1)}}(x_t, c_t)}{P_{\theta^{(i-1)}}(c_t | x_t)} \right)$$ **E-step**

$$= \sum_{t=1}^{n} \sum_{c_t=1}^{K} Q^{(i)}(c_t) \log P_{\theta^{(i)}}(x_t)$$

WHY SHOULD EM WORK?

Steps to show that $\log \mathrm{Lik}(\theta^{(i)}) \geq \log \mathrm{Lik}(\theta^{(i-1)})$:

$$\log P_{\theta^{(i)}}(x_1, \ldots, x_n) \geq \sum_{t=1}^{n} \sum_{c_t=1}^{K} Q^{(i)}(c_t) \log\left(\frac{P_{\theta^{(i)}}(x_t, c_t)}{Q^{(i)}(c_t)}\right)$$

$$\geq \sum_{t=1}^{n} \sum_{c_t=1}^{K} Q^{(i)}(c_t) \log\left(\frac{P_{\theta^{(i-1)}}(x_t, c_t)}{Q^{(i)}(c_t)}\right)$$

M-step

$$= \sum_{t=1}^{n} \sum_{c_t=1}^{K} Q^{(i)}(c_t) \log\left(\frac{P_{\theta^{(i-1)}}(x_t, c_t)}{P_{\theta^{(i-1)}}(c_t | x_t)}\right)$$

E-step

$$= \sum_{t=1}^{n} \sum_{c_t=1}^{K} Q^{(i)}(c_t) \log P_{\theta^{(i)}}(x_t)$$

$$= \sum_{t=1}^{n} \log P_{\theta^{(i)}}(x_t)$$

Why should EM work?

- Likelihood never decreases
- So whenever we converge we converge to a local optima
- However problem is non-convex and can have many local optimal
- In general no guarantee on rate of convergence
- In practice, do multiple random initializations and pick the best one!

EM Algorithm Generally

- More generally, EM can be used to learn any probabilistic model with some Latent (unseen) variables and some observed variables whenever
  - Its is easy to find parameters given distribution/ observation for all variables
  - Given all parameters finding distribution for latent variables is easy

How to choose K

- Elbow method:
  - plot Objective versus K, typically it monotonically decreases.
  - Pick point where there is a kink
  - Intuition: look at rate of change
- Add to objective penalty (+ pen(K)) and minimize, pen increases with K
  - intuition we prefer smaller number of clusters
  - Use prior knowledge to pick p
  - (AIC, BIC etc can been seen to be specific cases)