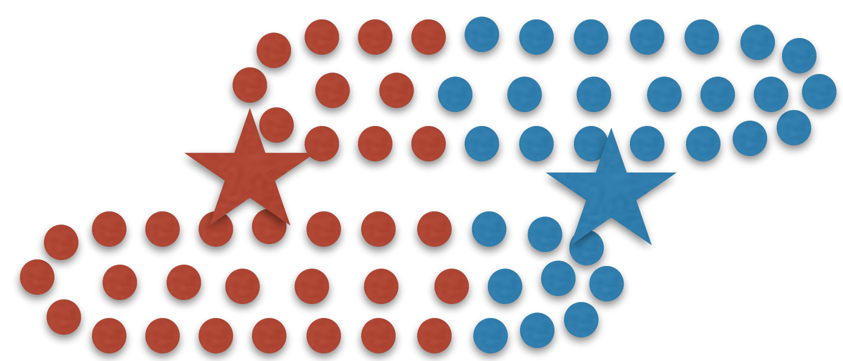
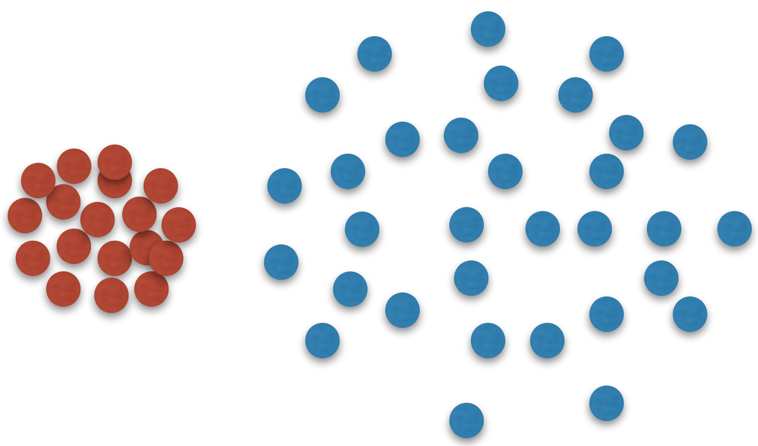
Machine Learning for Data Science (CS4786)
Lecture 14

Gaussian Mixture Model

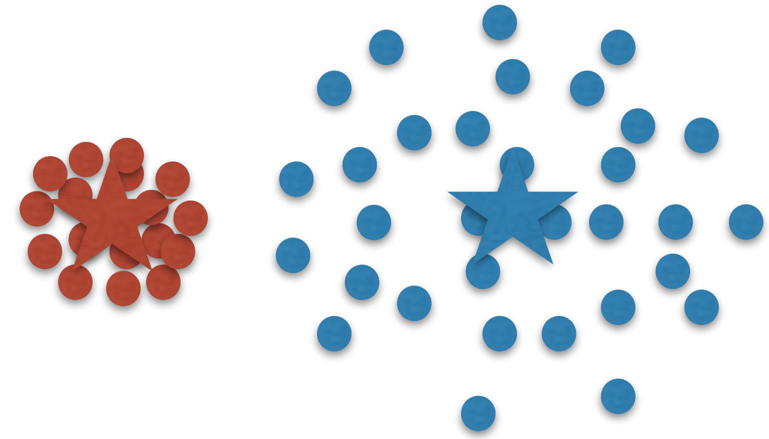
K-means: pitfalls



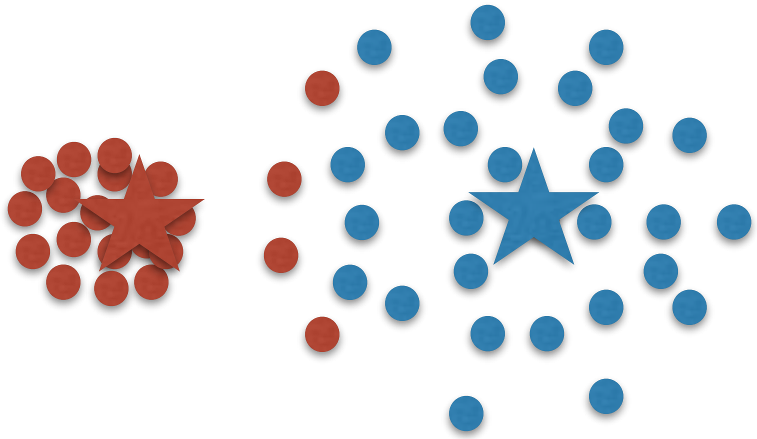
K-means: pitfalls



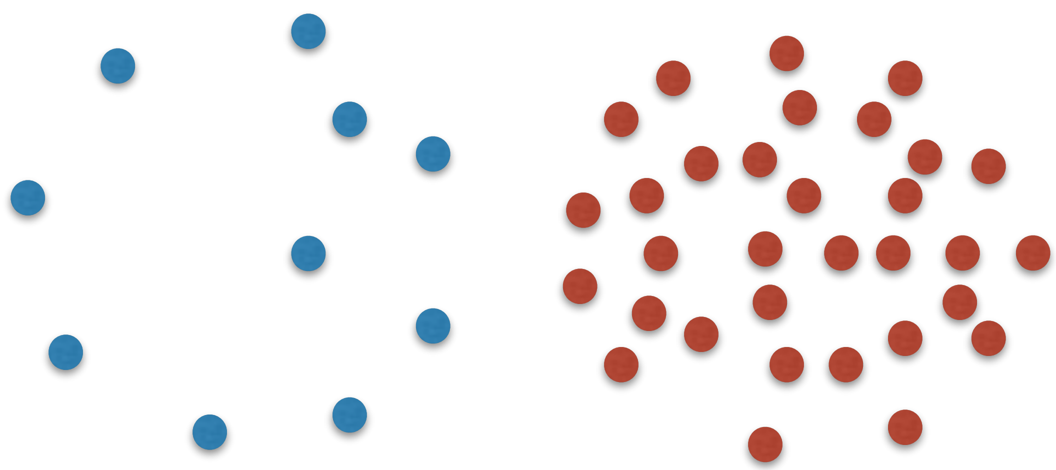
K-means: pitfalls



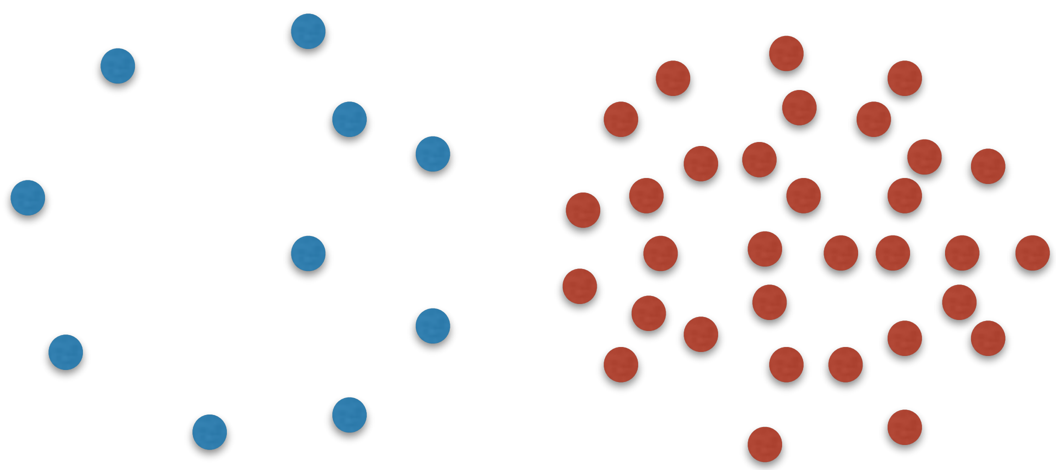
K-means: pitfalls



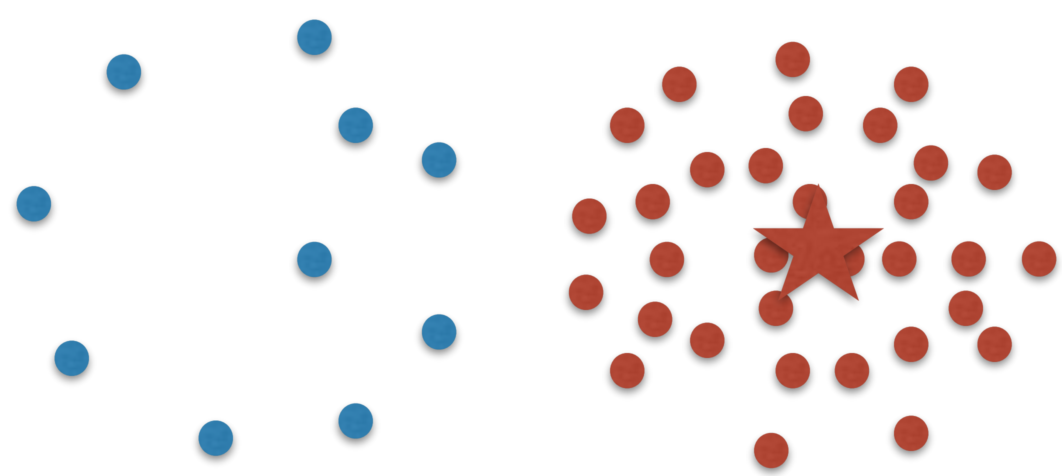
K-means: pitfalls



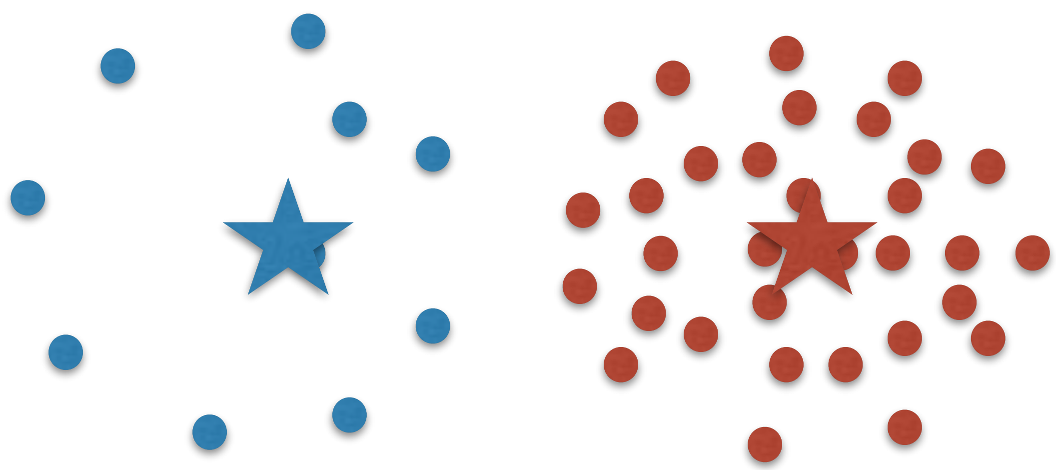
K-means: pitfalls



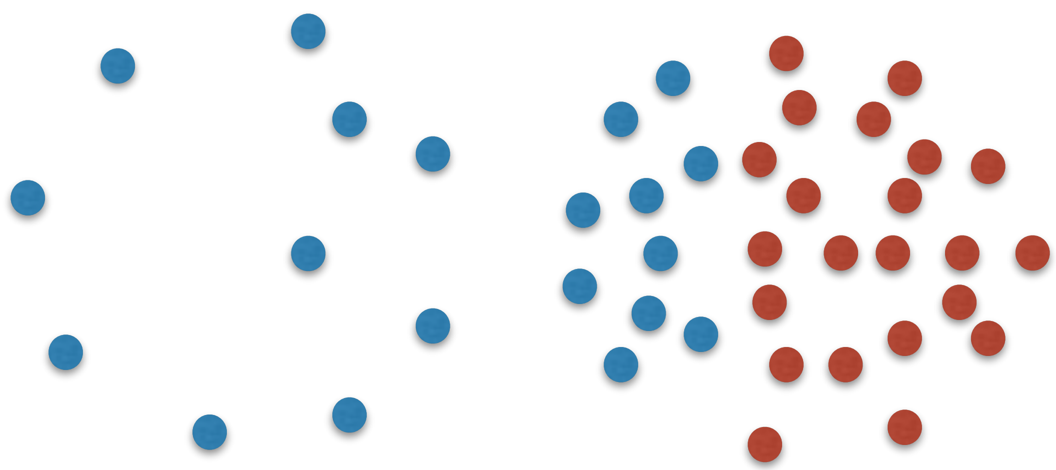
K-means: pitfalls



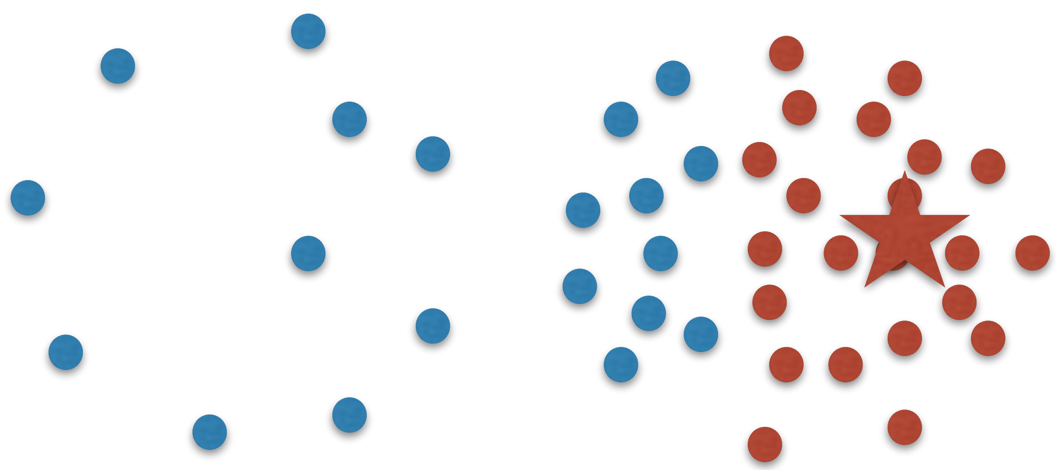
K-means: pitfalls



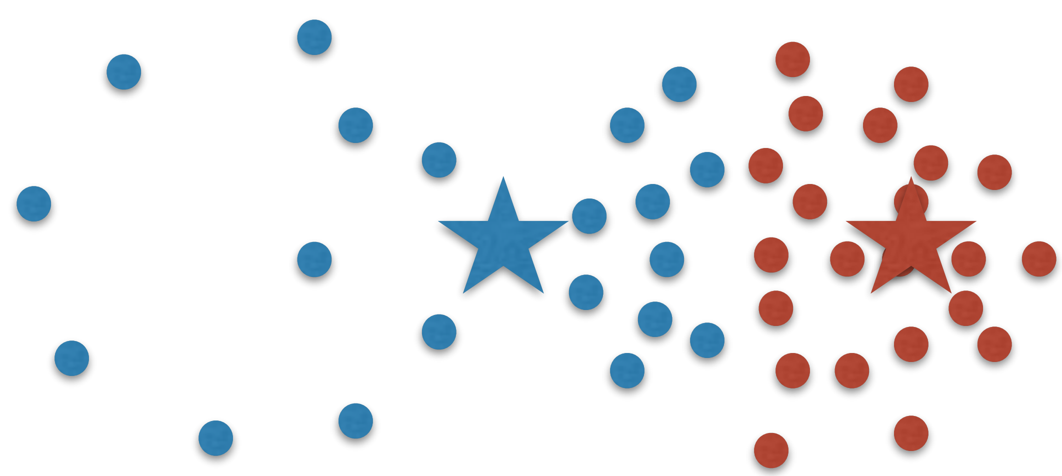
K-means: pitfalls



K-means: pitfalls



K-means: pitfalls



K-means: pitfalls

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

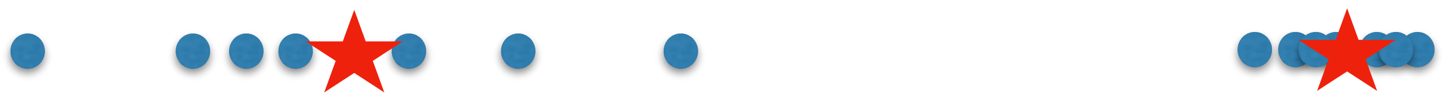
K-means: pitfalls

- Can we design algorithm that can address these shortcomings?

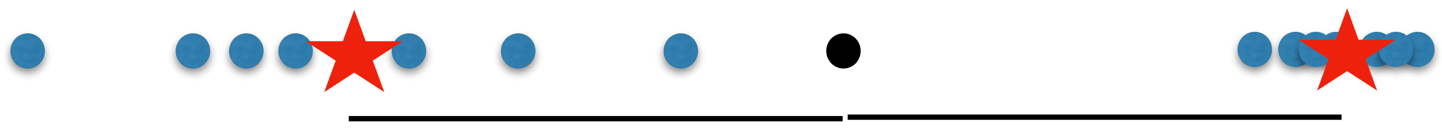
Variance and Radius

Variance and Radius

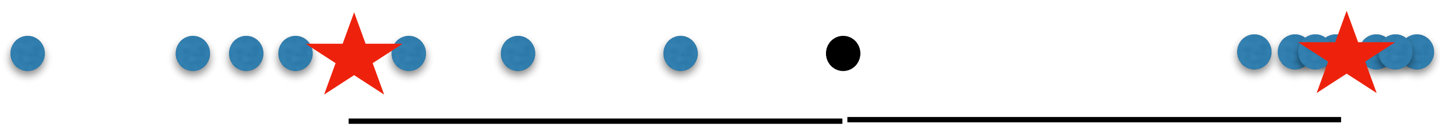
Variance and Radius



Variance and Radius

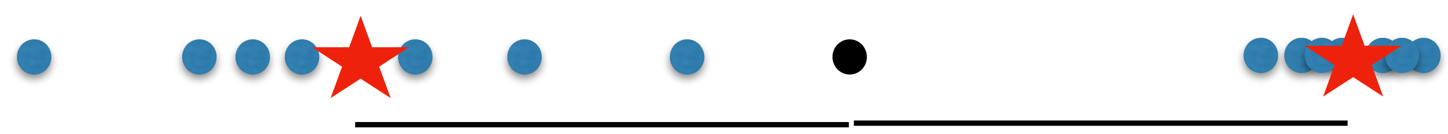


Variance and Radius



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

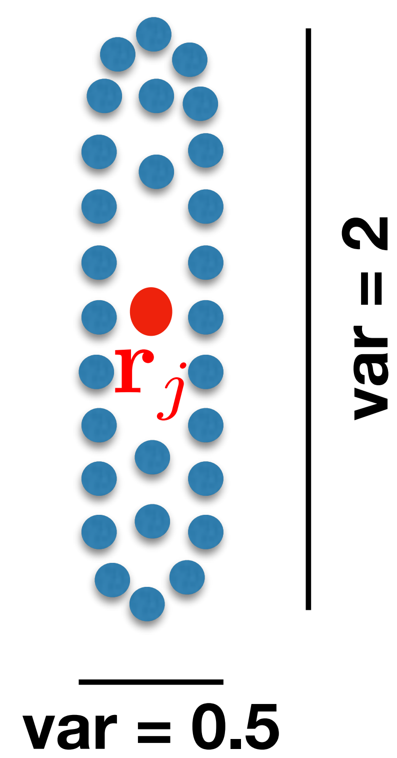
Variance and Radius



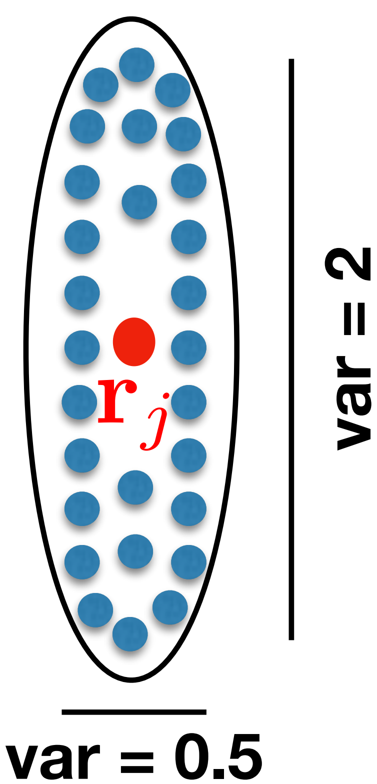
Distance to mean 1 should be smaller than distance to mean 2 as black dot is more likely in cluster 1 than 2

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

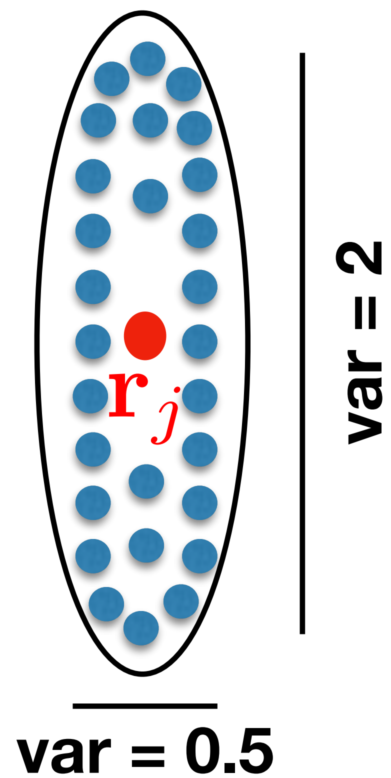
Axis Aligned Ellipsoid



Axis Aligned Ellipsoid

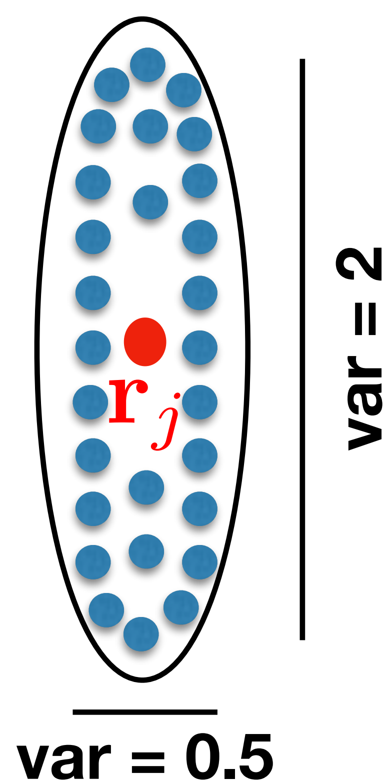


Axis Aligned Ellipsoid



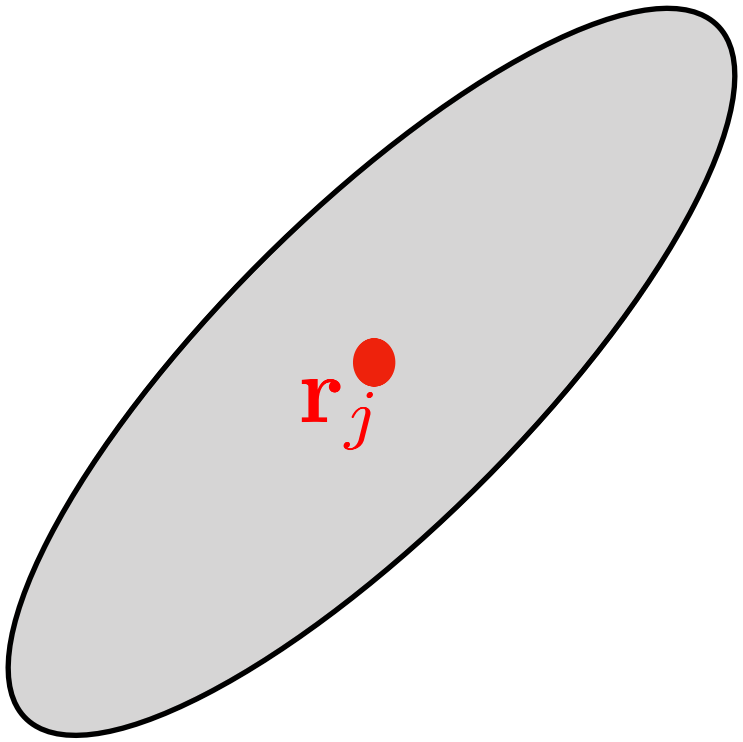
$$(\mathbf{x}-\mathbf{r}_j)^\top \left[\begin{array}{cc} 1/0.5 & 0 \\ 0 & 1/2 \end{array}\right](\mathbf{x}-\mathbf{r}_j)$$

Axis Aligned Ellipsoid



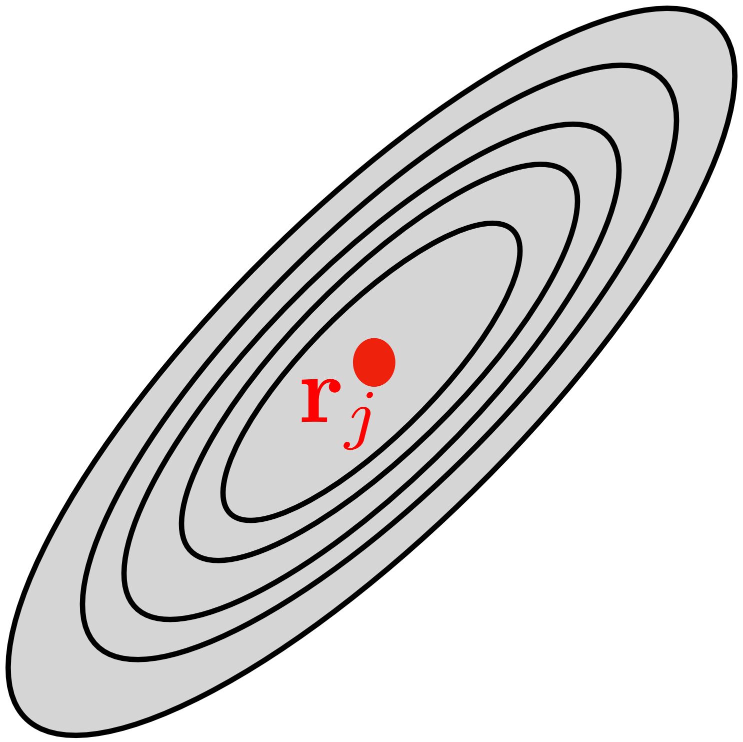
$$(\mathbf{x} - \mathbf{r}_j)^\top \left[\Sigma^{-1} \right] (\mathbf{x} - \mathbf{r}_j)$$

General Ellipsoid



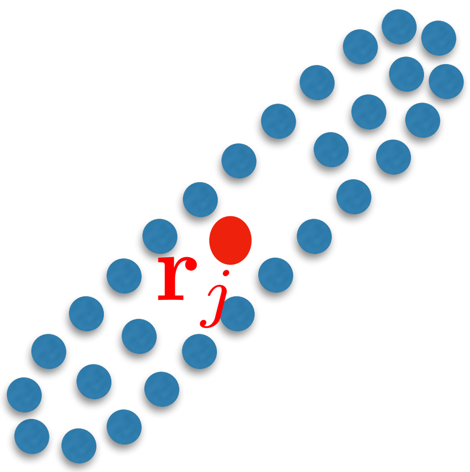
$$(\mathbf{x} - \mathbf{r}_j)^\top A(\mathbf{x} - \mathbf{r}_j) \leq 1$$

General Ellipsoid

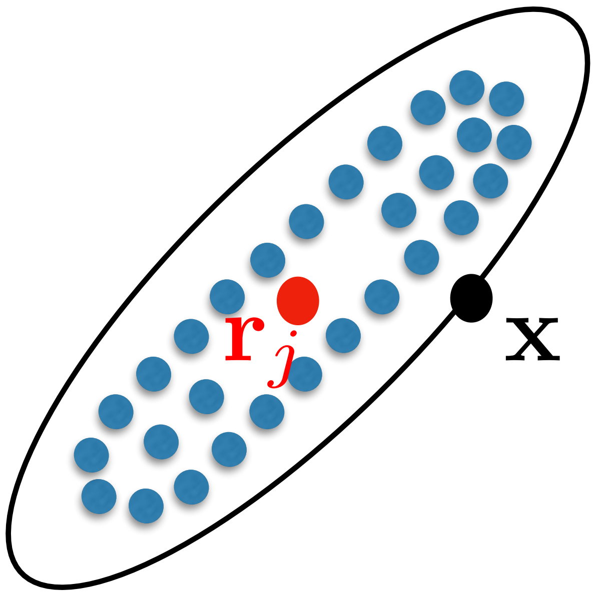


$$(\mathbf{x}-\mathbf{r}_j)^\top A(\mathbf{x}-\mathbf{r}_j) \leq 1$$

General Ellipsoid

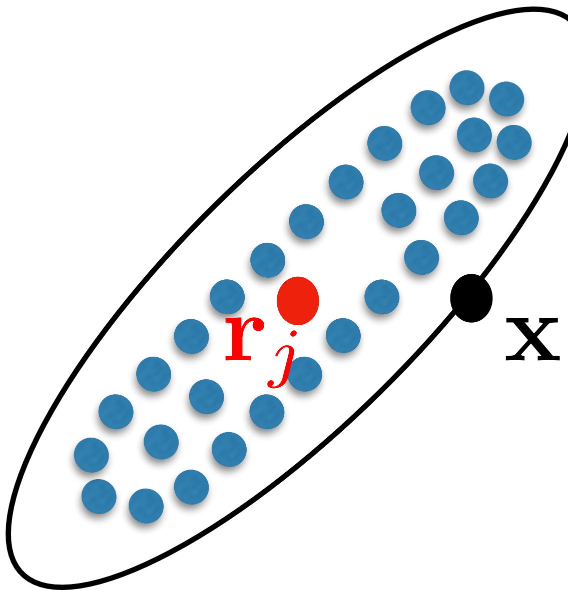


General Ellipsoid



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

General Ellipsoid



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

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

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

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

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

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

  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

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

  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

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

  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 $t \in \{1, \dots, n\}$, set cluster identity of the point

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

  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. $m \leftarrow m + 1$

Hard Gaussian Mixture Model

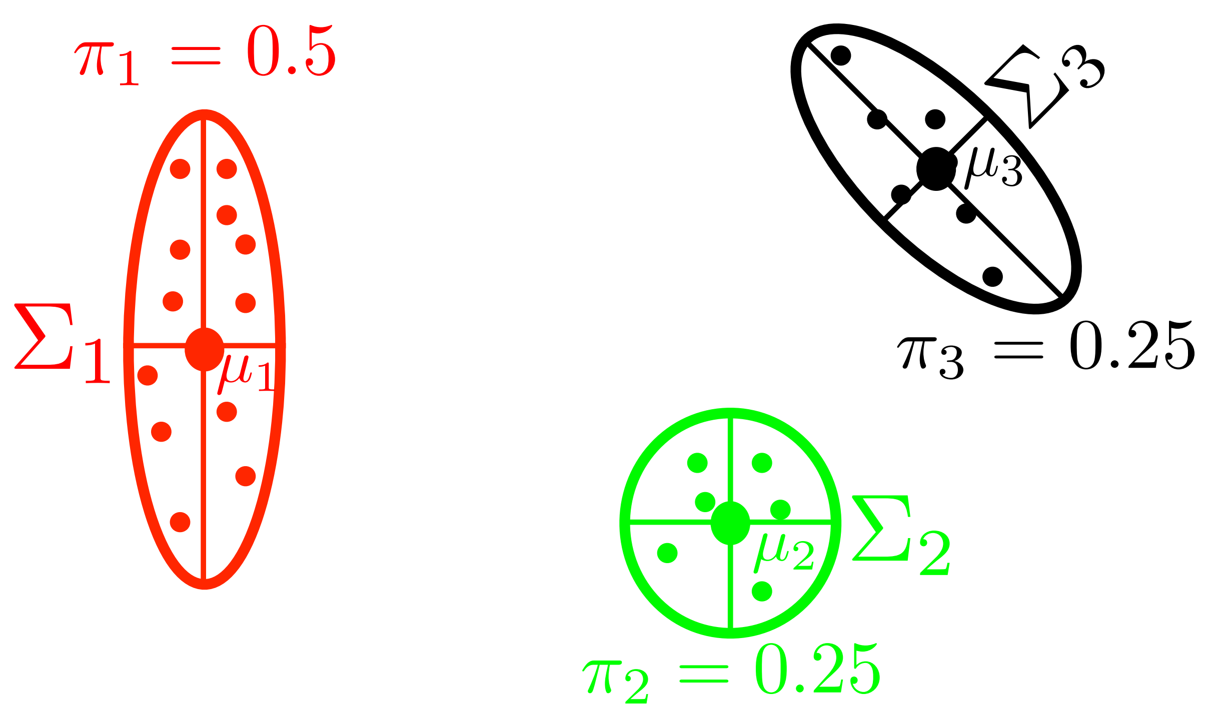
- For all $j \in [K]$, initialize cluster centroids $\hat{\mathbf{r}}_j^0$, ellipsoids $\hat{\Sigma}_j^0$ and initial proportions π^0 randomly and set $m = 1$
- Repeat until convergence (or until patience runs out)
  1. For each $t \in \{1, \ldots, n\}$, set cluster identity of the point
  $$\hat{c}^m(\mathbf{x}_t) = \operatorname*{argmin}_{j \in [K]} \ (\mathbf{x}_t - \hat{\mathbf{r}}_j^{m-1})^\top \left(\hat{\Sigma}^{m-1}\right)^{-1} (\mathbf{x}_t - \hat{\mathbf{r}}_j^{m-1}) - \log(\pi_j^{m-1})$$
  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. $m \leftarrow m + 1$

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:

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



Multivariate Gaussian

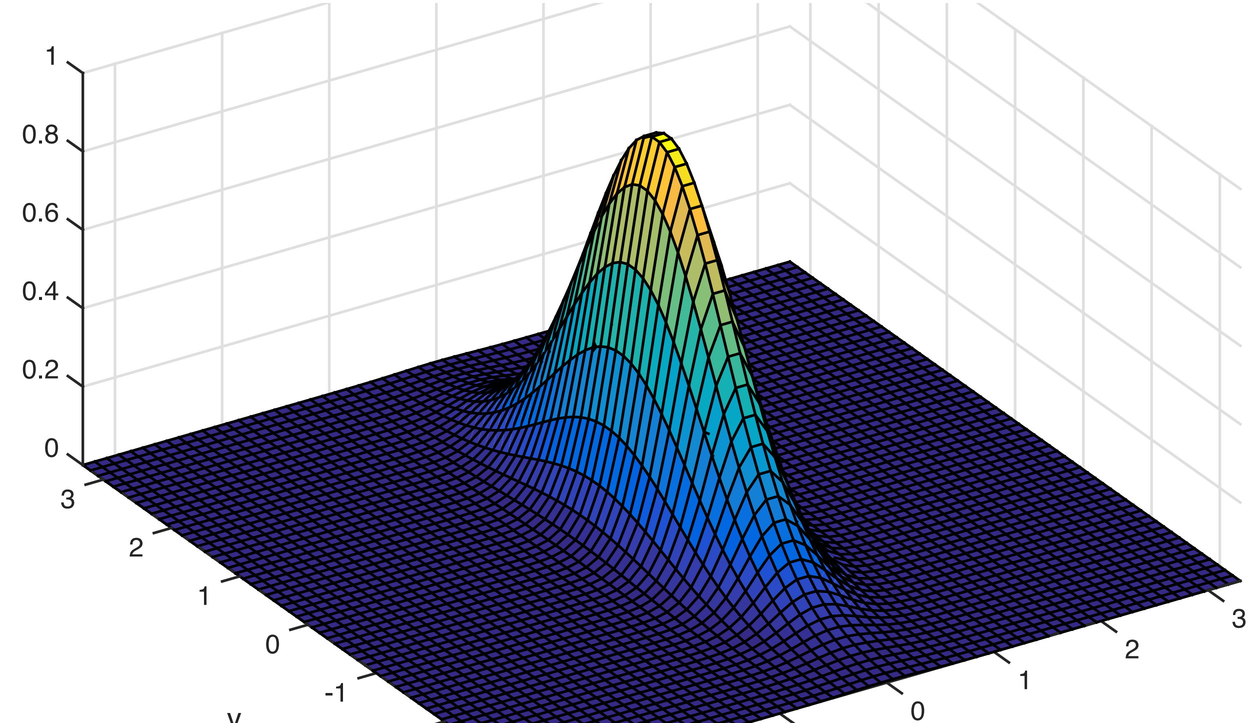
- Two parameters:
  - Mean $\mu \in \mathbb{R}^d$
  - Covariance matrix Σ of size dxd

$$p(x; \mu, \Sigma) = (2\pi)^{d/2} \det(\Sigma)^{-1/2} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1} (x-\mu)\right)$$

Multivariate Gaussian

- Two parameters:
  - Mean $\mu \in \mathbb{R}^d$
  - Covariance matrix Σ of size dxd

$$p(x; \mu, \Sigma) = (2\pi)^{d/2} \det(\Sigma)^{-1/2} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right)$$



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

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

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 $\log P_\theta(x_1, \ldots, x_n)$

MLE FOR GMM

Let us consider the one dimensional case, assume variances are 1 and π is uniform

$$\log P_\theta(x_{1,\ldots,n}) = \sum_{t=1}^{n} \log\left(\frac{1}{K}\sum_{i=1}^{K}\frac{1}{\sqrt{2 * 3.1415}}\exp\left(-(x_t - \mu_i)^2/2\right)\right)$$

Now consider the partial derivative w.r.t. μ_1, we have:

$$\frac{\partial \log P_\theta(x_{1,\ldots,n})}{\partial \mu_1} = \sum_{t=1}^{n} \frac{-(x_t - \mu_1)\exp\left(-\frac{(x_t-\mu_1)^2}{2}\right)}{\sum_{i=1}^{K}\exp\left(-\frac{(x_t-\mu_i)^2}{2}\right)}$$

Given all other parameters, optimizing w.r.t. even just μ_1 is hard!

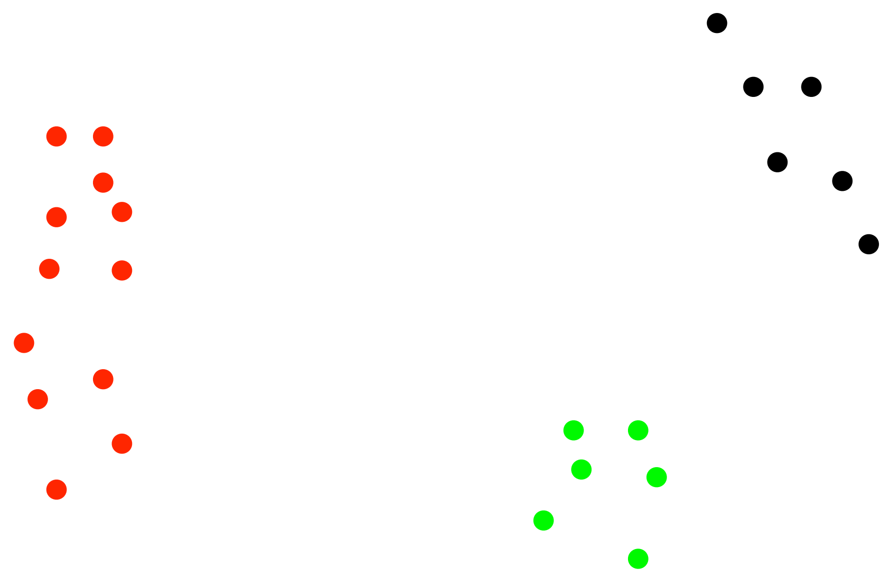
MLE FOR GMM

Say by some magic you knew cluster assignments, then

How would you compute parameters ?

MLE FOR GMM

Say by some magic you knew cluster assignments, then



How would you compute parameters ?

Latent Variables

- We only observe $x_1, \ldots, x_n$, cluster assignments $c_1, \ldots, c_n$ are not observed
- Finding $\theta \in \Theta$ (even for 1-d GMM) that directly maximizes Likelihood or A Posteriori given $x_1, \ldots, x_n$ is hard!
- Given latent variables $c_1, \ldots, c_n$, the problem of maximizing likelihood (or a posteriori) became easy

Can we use latent variables to device an algorithm?

TOWARDS EM ALGORITHM

- Latent variables can help, but we have a chicken and egg problem

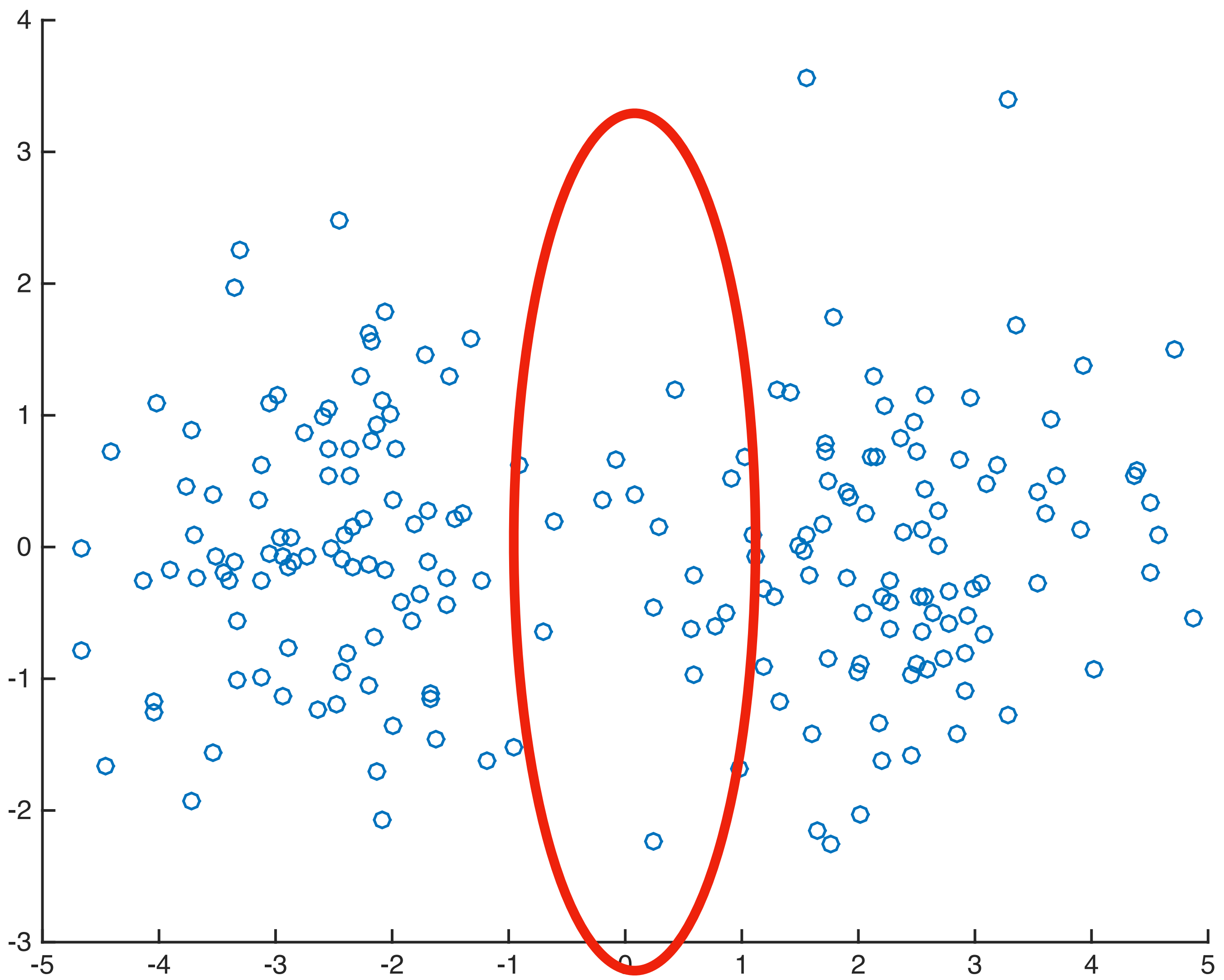
Given all variables including latent variables, finding optimal parameters is easy

Given model parameter, optimizing/finding distribution over the latent variables is easy

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



(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. $m \leftarrow m + 1$