Machine Learning for Data Science (CS4786)
Lecture 12

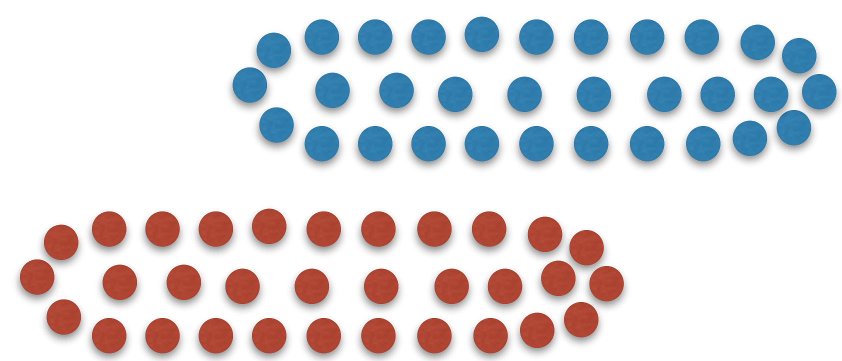
Gaussian Mixture Models

Course Webpage :
http://www.cs.cornell.edu/Courses/cs4786/2016fa/

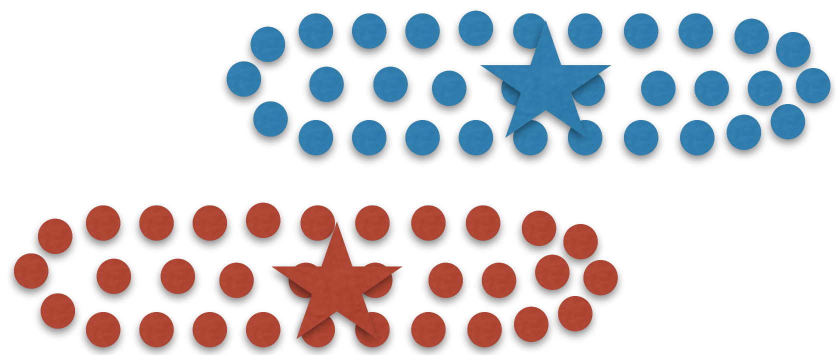
Back to K-means

- Single link is sensitive to outliners
- We need a good clustering algorithm after spectral embedding: K-means?

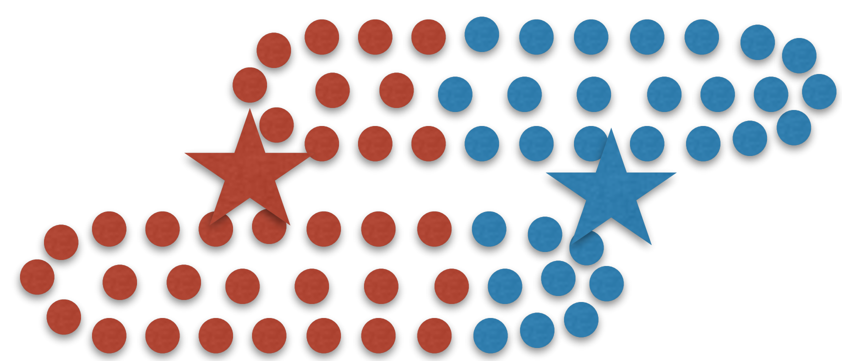
Back to K-means



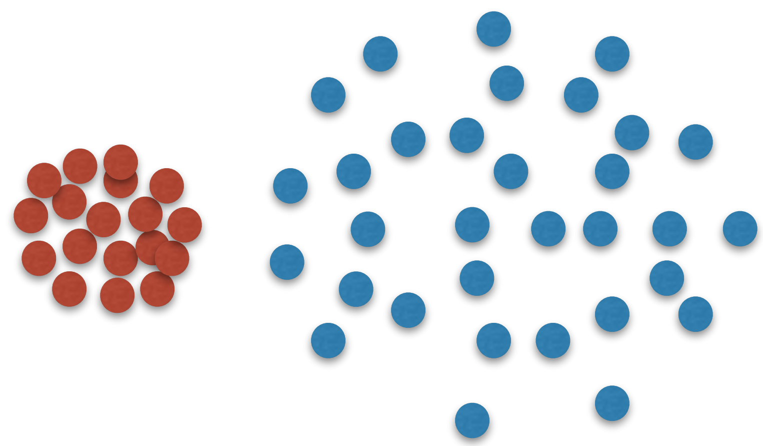
Back to K-means



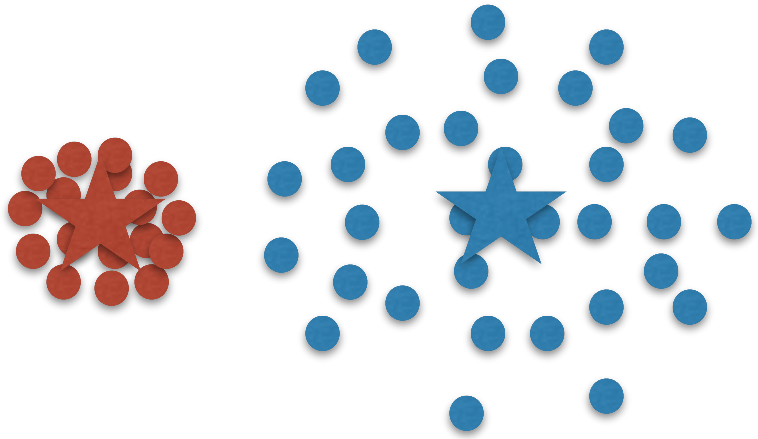
Back to K-means



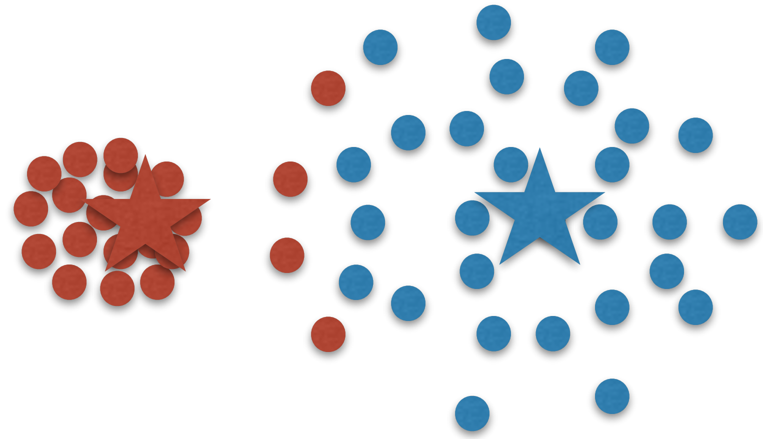
Back to K-means



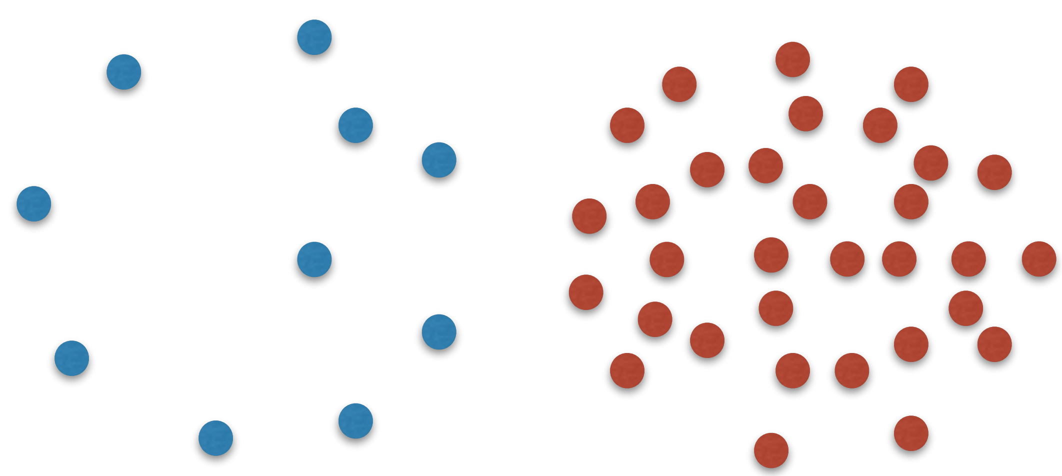
Back to K-means



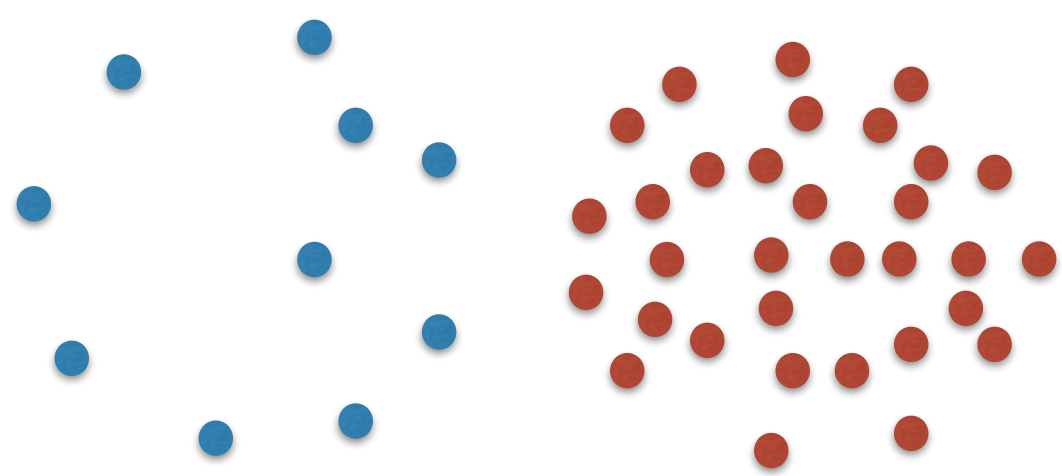
Back to K-means



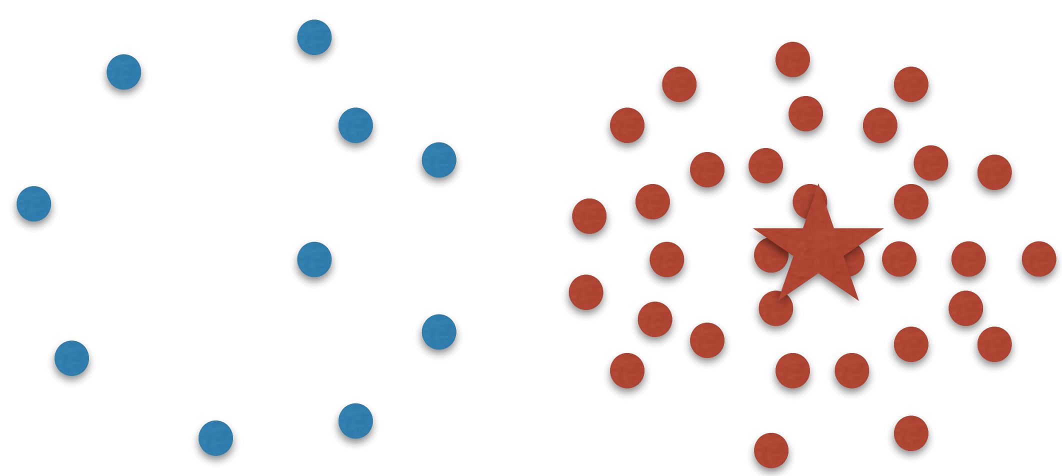
Back to K-means



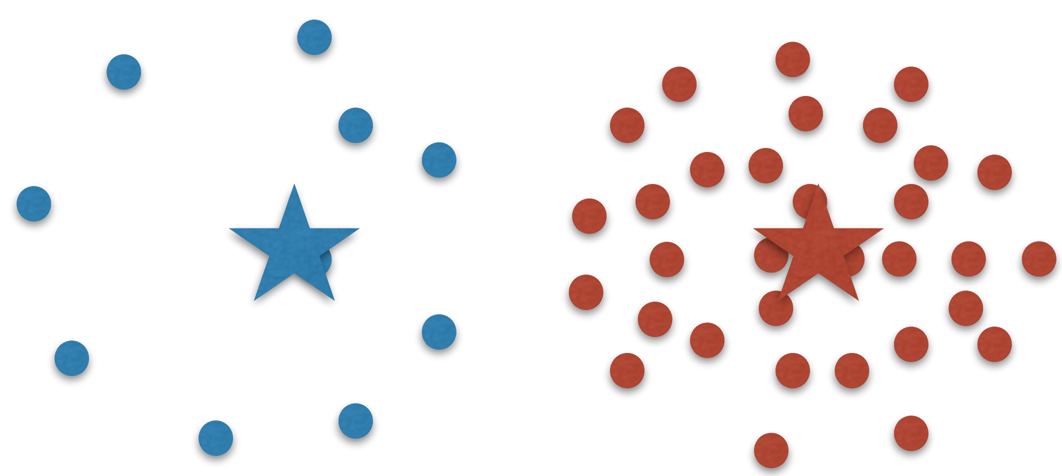
Back to K-means



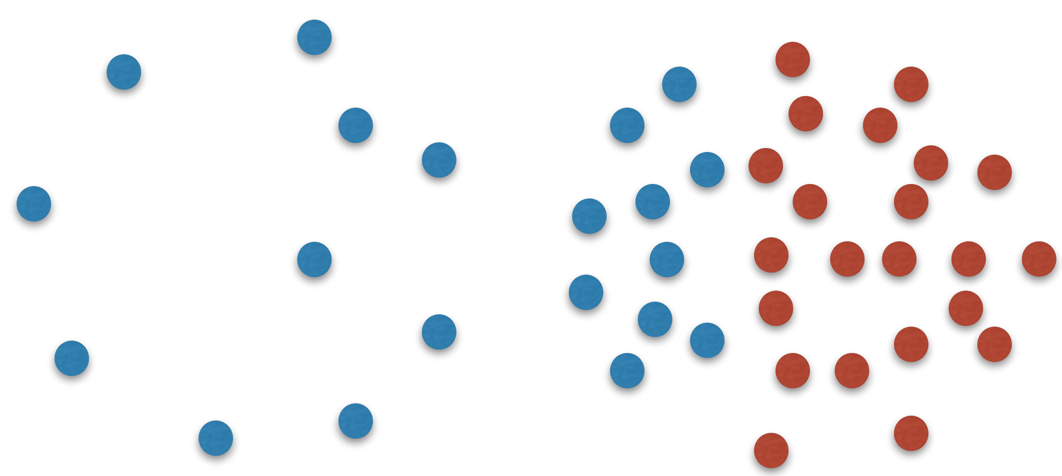
Back to K-means



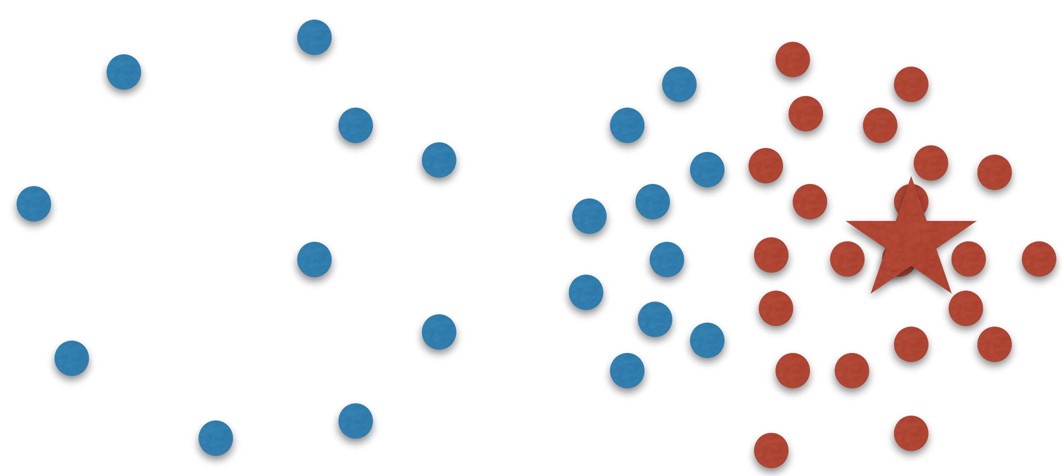
Back to K-means



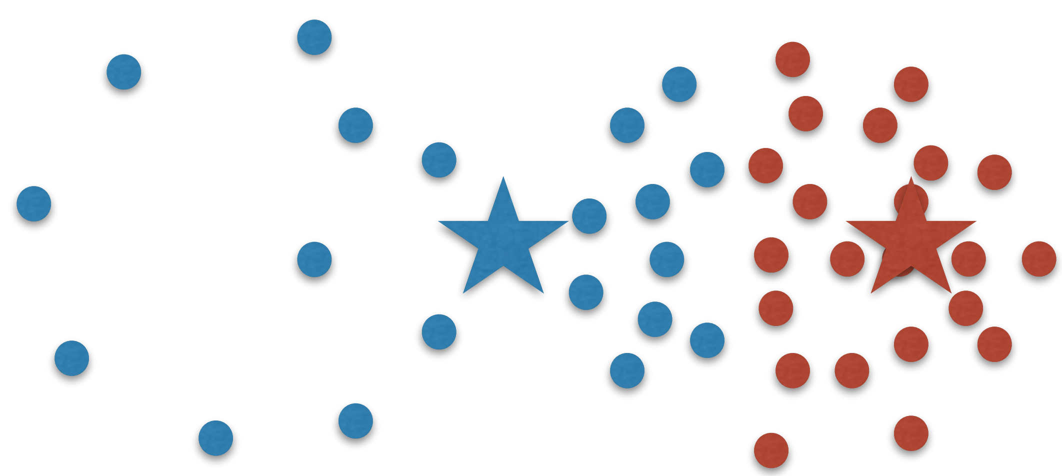
Back to K-means



Back to K-means



Back to K-means



Back to K-means

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

No Free Lunch

No Free Lunch

- When averaged across all possible situations, all algorithms perform equally well/badly

No Free Lunch

- When averaged across all possible situations, all algorithms perform equally well/badly

No Assumptions => No method

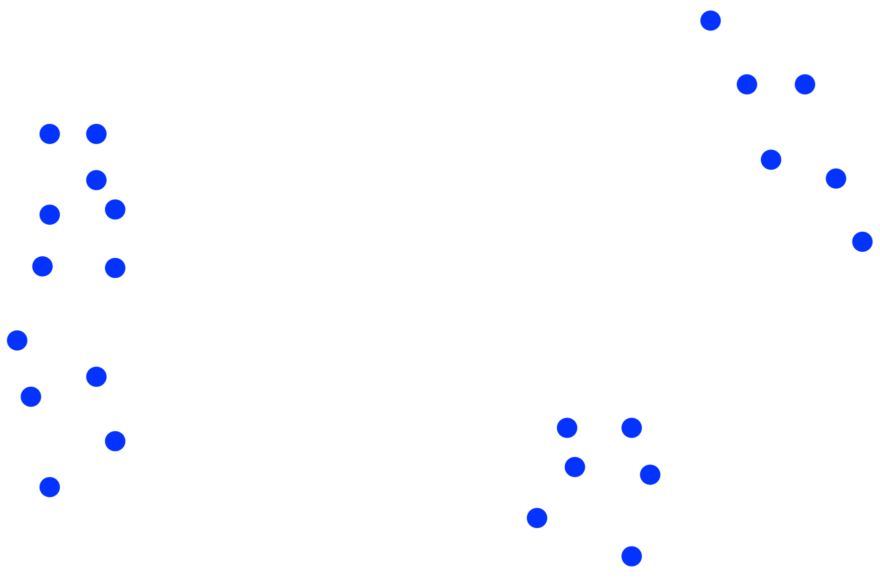
No Free Lunch

- When averaged across all possible situations, all algorithms perform equally well/badly

No Assumptions => No method

Lets model our assumptions in a more principled way

How do we model the following?



Multivariate Gaussian

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

Multivariate Gaussian

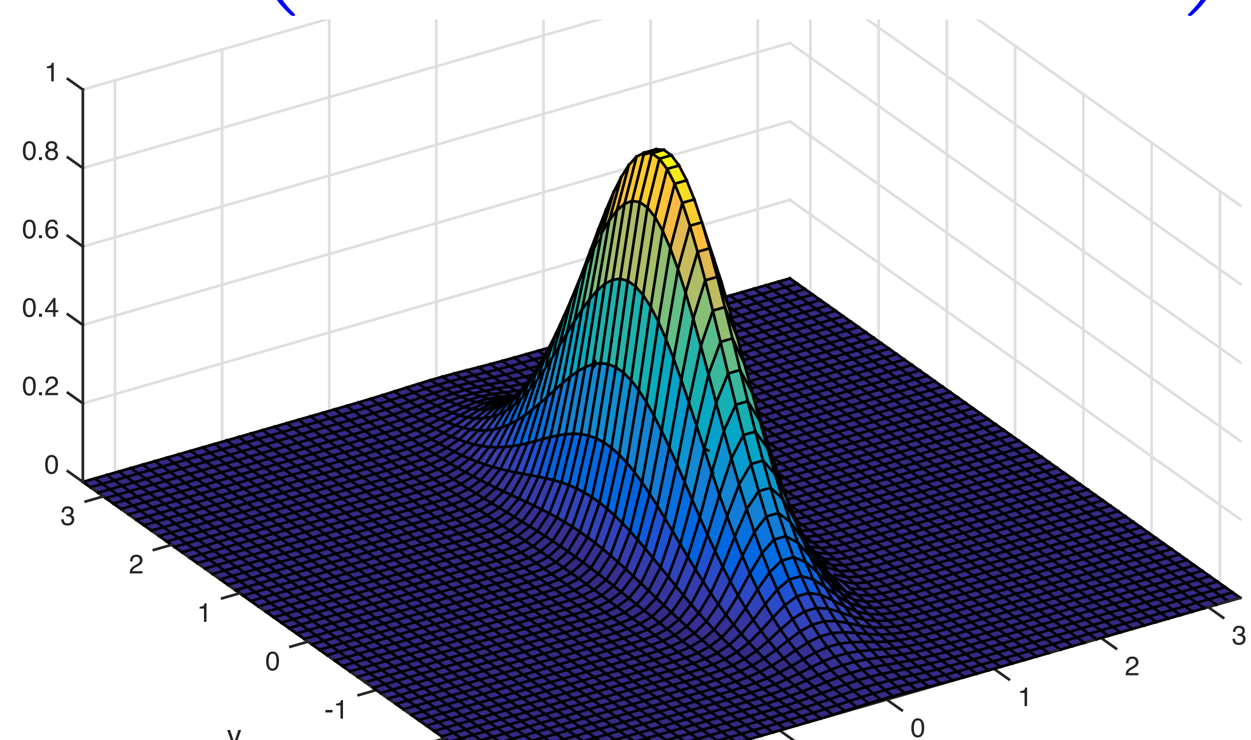
- Two parameters:
  - Mean $\mu \in \mathbb{R}^d$
  - Covariance matrix Σ of size dxd

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

Multivariate Gaussian

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

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

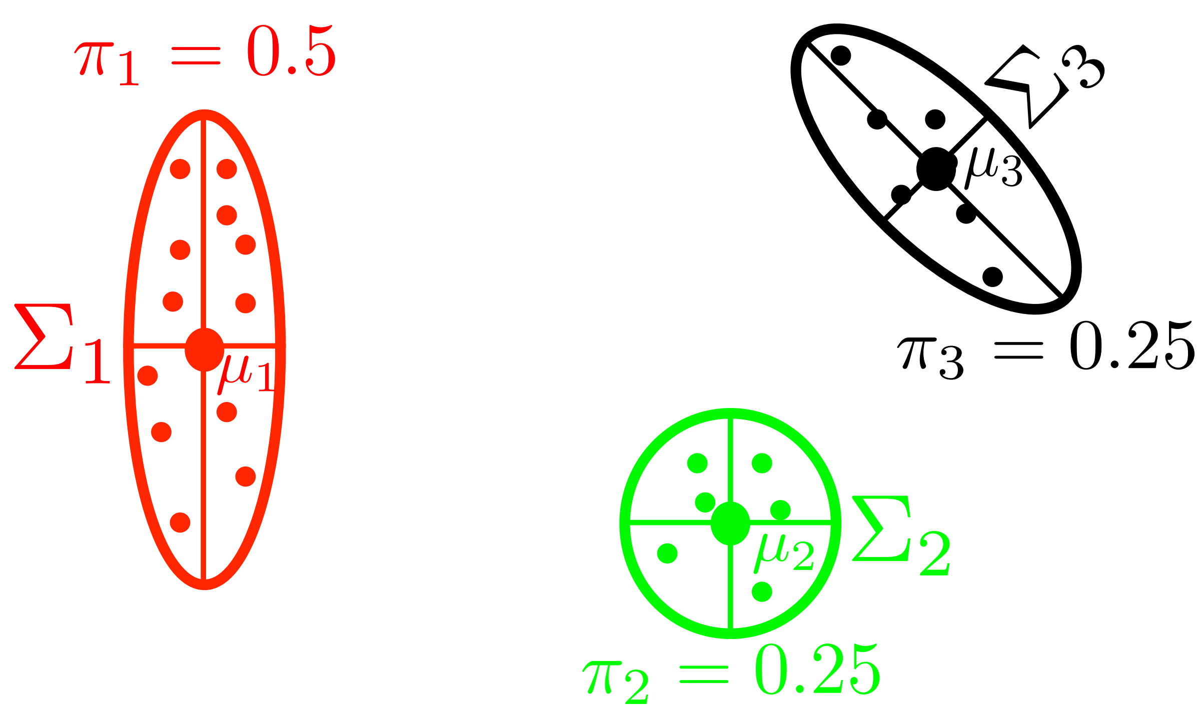


Gaussian Mixture Models

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:

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



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

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

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

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

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

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

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

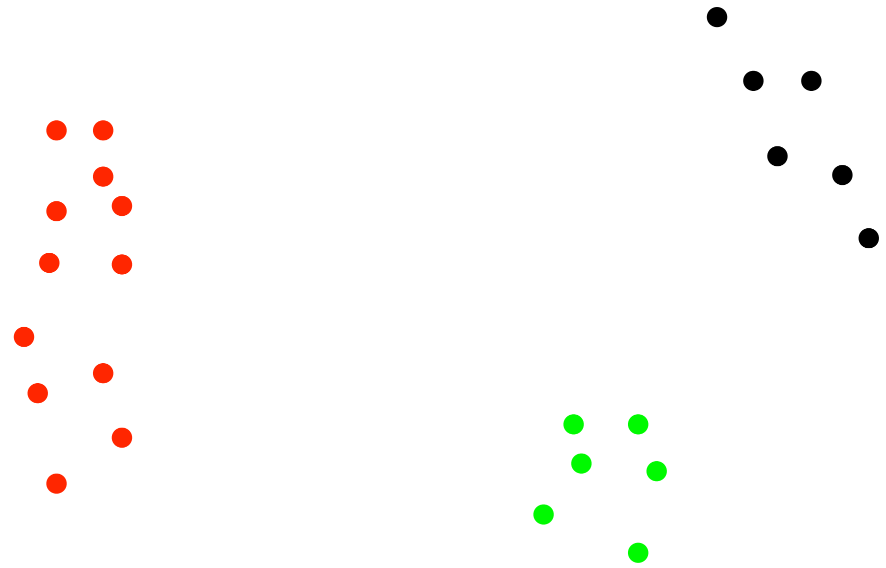
MLE FOR GMM

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

GMM: Power of Wishful Thinking

1. Initialize model parameters $\pi^{(0)}, \mu_1^{(0)}, \ldots, \mu_K^{(0)}$ and $\Sigma_1^{(0)}, \ldots, \Sigma_K^{(0)}$
2. For i = 1 until convergence or bored
   1. Under current model parameters $\theta^{(i-1)}$, compute probability $Q_t^{(i)}(k)$ of each point $\mathbf{x}_t$ belonging to cluster k
   2. Given probabilities of each point belonging to the various clusters, compute optimal parameters $\theta^{(i)}$
3. End For

EM Algorithm for GMM

1. Initialize model parameters $\pi^{(0)}, \mu_1^{(0)}, \ldots, \mu_K^{(0)}$ and $\Sigma_1^{(0)}, \ldots, \Sigma_K^{(0)}$
2. For i = 1 until convergence or bored
    1. $Q_t^{(i)}(k) \propto p(\mathbf{x}_t; \mu_k^{(i-1)}, \Sigma_k^{(i-1)}) \cdot \pi_k^{(i-1)}$
    2. For every $k \in [K]$,

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

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

3. End For

Demo