Machine Learning for Data Science (CS4786)
Lecture 16

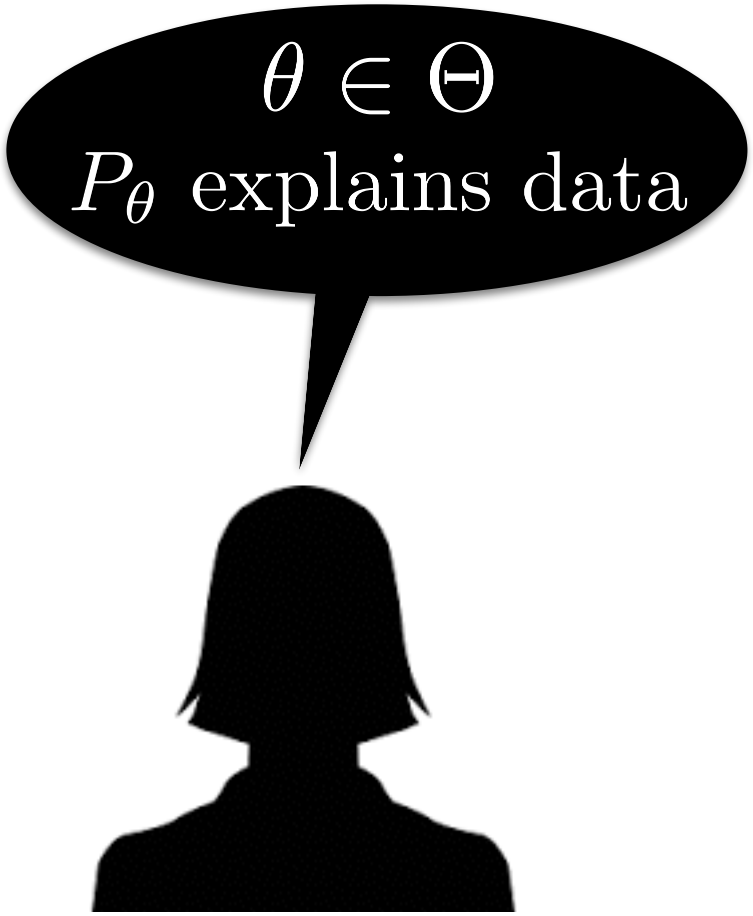
Probabilistic Modeling and EM Algorithm

Announcements

Changes to Grading

- Until May 12th, new last day of classes to drop or change to S/U
- Instead of Prelims 1, I will be giving out short quizzes every Thursday via cms that will be due on Tuesday
- Take home finals
- Only 3 assignments and a shorter competition

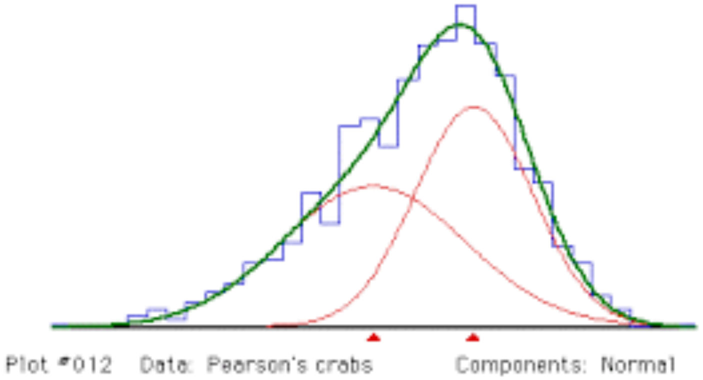
Probabilistic Model



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

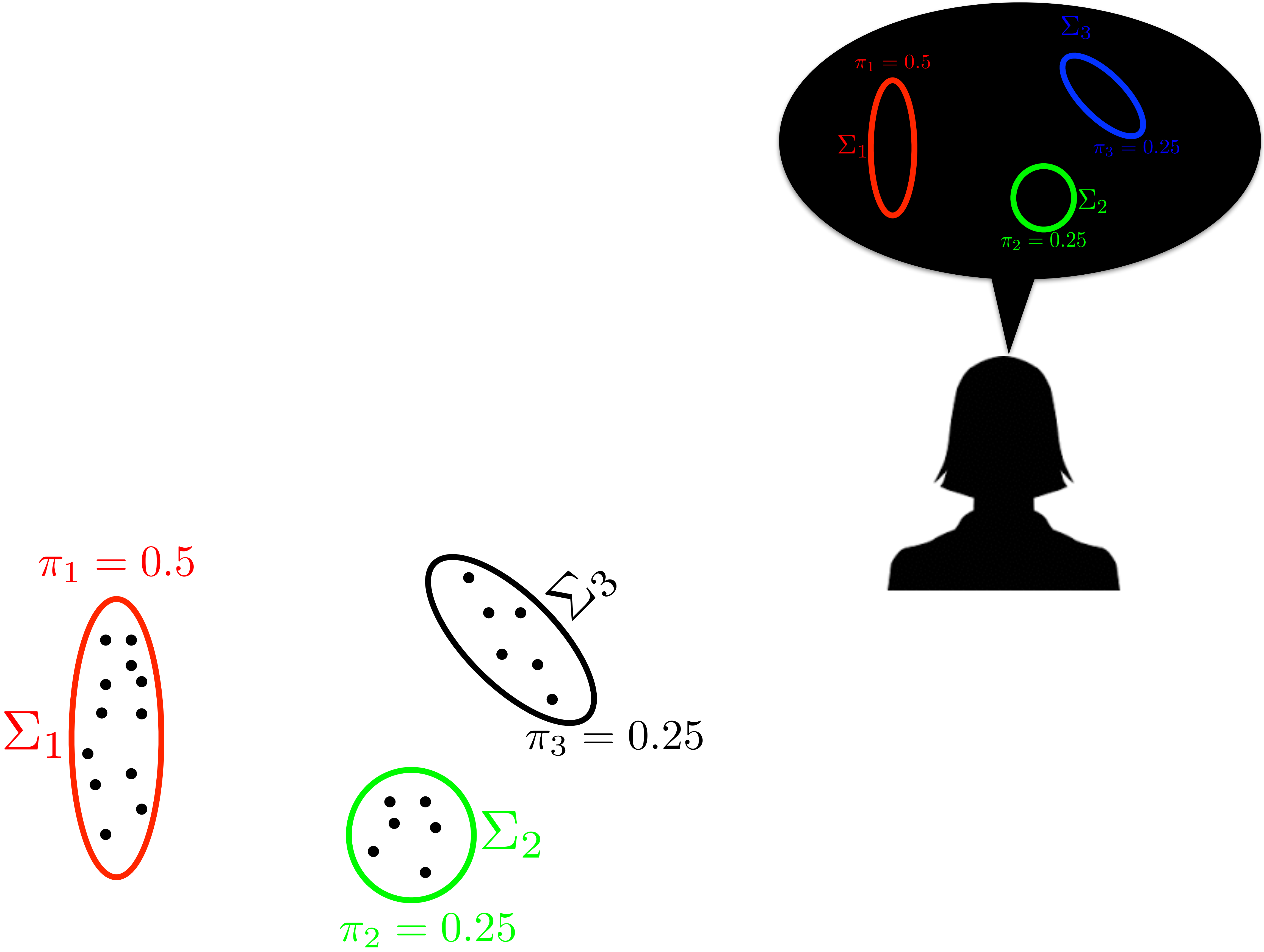
Weldon’s Crab dataset

- 23 attributes, 1000 measurements
- All but one attribute were fit well by normal distribution
- One of them looked like…



Discovered that there were two species of crabs

Probabilistic Model



PROBABILISTIC MODELS

- Set of models Θ consists of parameters s.t. P_θ for each $\theta \in \Theta$ is a distribution over data.
- Learning: Estimate $\theta^* \in \Theta$ that best models given data

Maximum Likelihood Principal

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

Maximum Likelihood Principal

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

Reasoning:

- One of the models in Θ is the correct one

Maximum Likelihood Principal

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

Reasoning:

- One of the models in Θ is the correct one
- Given data we pick the one that best explains the observed data

Maximum Likelihood Principal

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

Reasoning:

- One of the models in Θ is the correct one
- Given data we pick the one that best explains the observed data
- Equivalently pick the maximum likelihood estimator,

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

Maximum Likelihood Principal

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

Reasoning:

- One of the models in Θ is the correct one
- Given data we pick the one that best explains the observed data
- Equivalently pick the maximum likelihood estimator,

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

Often referred to as frequentist view

Maximum Likelihood Principal

Pick θ ∈ Θ that maximizes probability of observation

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

- A priori all models are equally good, data could have been generated by any one of them

Maximum A Posteriori

Say you had a prior belief about models provided by $P(\theta)$

Pick $\theta \in \Theta$ that is most likely given data

Maximum A Posteriori

Say you had a prior belief about models provided by $P(\theta)$

Pick θ ∊ Θ that is most likely given data

Reasoning:

- Models are abstractions that capture our belief

Maximum A Posteriori

Say you had a prior belief about models provided by $P(\theta)$

Pick θ ∊ Θ that is most likely given data

Reasoning:

- Models are abstractions that capture our belief
- We update our belief based on observed data

Maximum A Posteriori

Say you had a prior belief about models provided by $P(\theta)$

Pick θ ∊ Θ that is most likely given data

Reasoning:

- Models are abstractions that capture our belief
- We update our belief based on observed data
- Given data we pick the model that we believe the most

Maximum A Posteriori

Say you had a prior belief about models provided by $P(\theta)$

Pick $\theta \in \Theta$ that is most likely given data

Reasoning:

- Models are abstractions that capture our belief
- We update our belief based on observed data
- Given data we pick the model that we believe the most
- Pick θ that maximizes $\log P(\theta|x_1, \ldots, x_n)$

Maximum A Posteriori

Say you had a prior belief about models provided by $P(\theta)$

Pick θ ∊ Θ that is most likely given data

Reasoning:

- Models are abstractions that capture our belief
- We update our belief based on observed data
- Given data we pick the model that we believe the most
- Pick θ that maximizes $\log P(\theta|x_1, \ldots, x_n)$

I want to say : Often referred to as Bayesian view

MAXIMUM A POSTERIORI

Say you had a prior belief about models provided by $P(\theta)$

Pick $\theta \in \Theta$ that is most likely given data

Reasoning:

- Models are abstractions that capture our belief
- We update our belief based on observed data
- Given data we pick the model that we believe the most
- Pick θ that maximizes $\log P(\theta|x_1, \ldots, x_n)$

I want to say : Often referred to as Bayesian view

There are Bayesian and there Bayesians

Maximum A Posteriori

Pick θ ∈ Θ that is most likely given data

Maximize a posteriori probability of model given data

$$\theta_{MAP} = \text{argmax}_{\theta \in \Theta} P(\theta | x_1, \ldots, x_n)$$

The Bayesian Choice

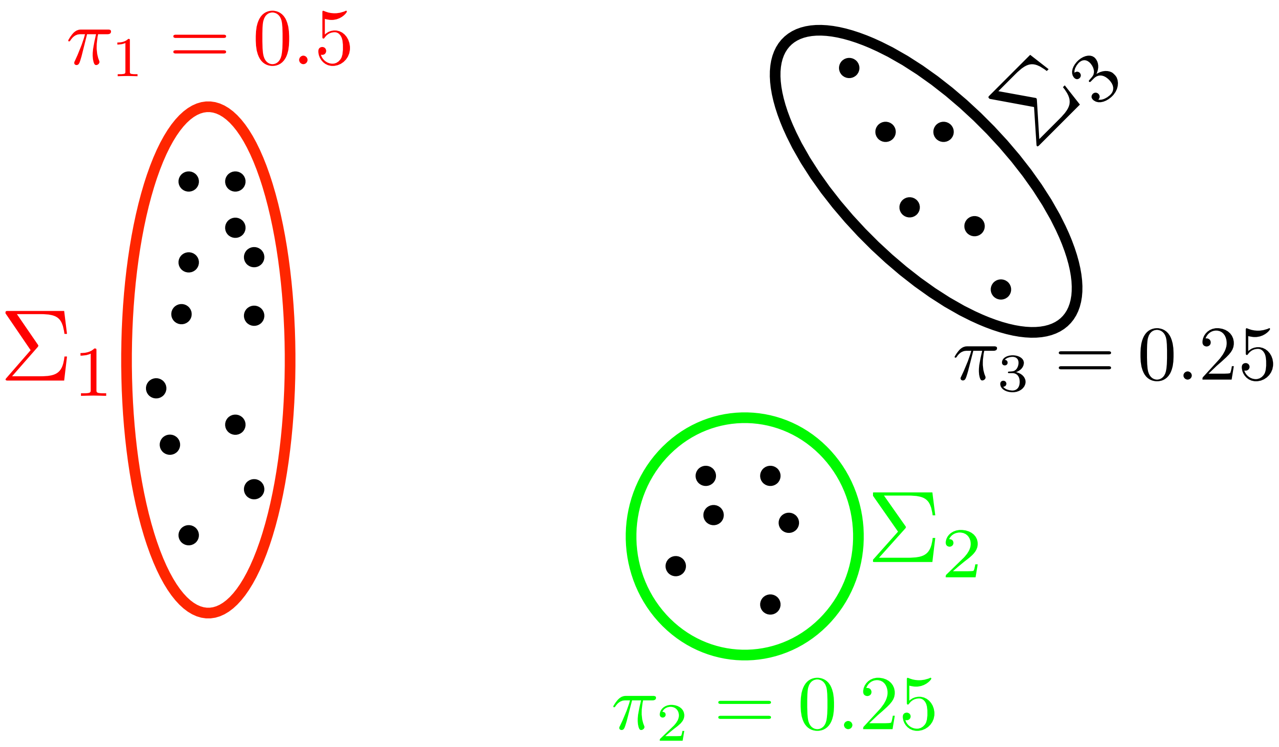
Don't pick any $\theta^* \in \Theta$

- Model is simply an abstraction
- We have a prosteriori distribution over models, why pick one θ?

$$P(X|\text{data}) = \sum_{\theta \in \Theta} P(X, \theta|\text{data}) = \sum_{\theta \in \Theta} P(X|\theta) P(\theta|\text{data})$$

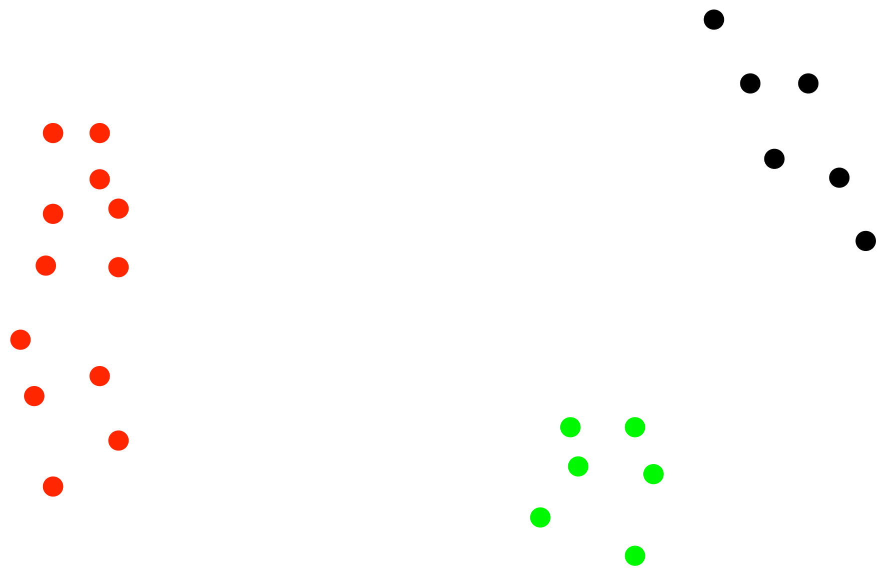
Lets get back to GMM

Probabilistic Model



MLE FOR GMM

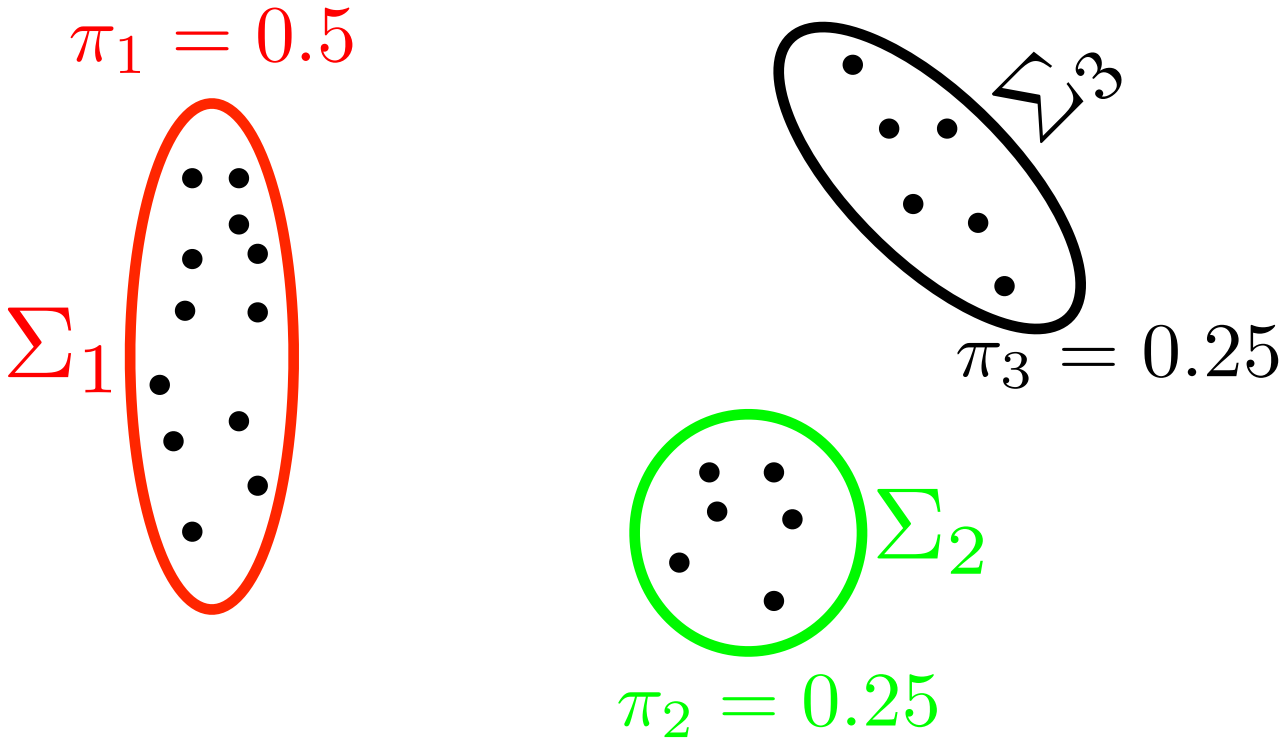
Say by some magic you knew cluster assignments, then



How would you compute parameters ?

MLE for GMM

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



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

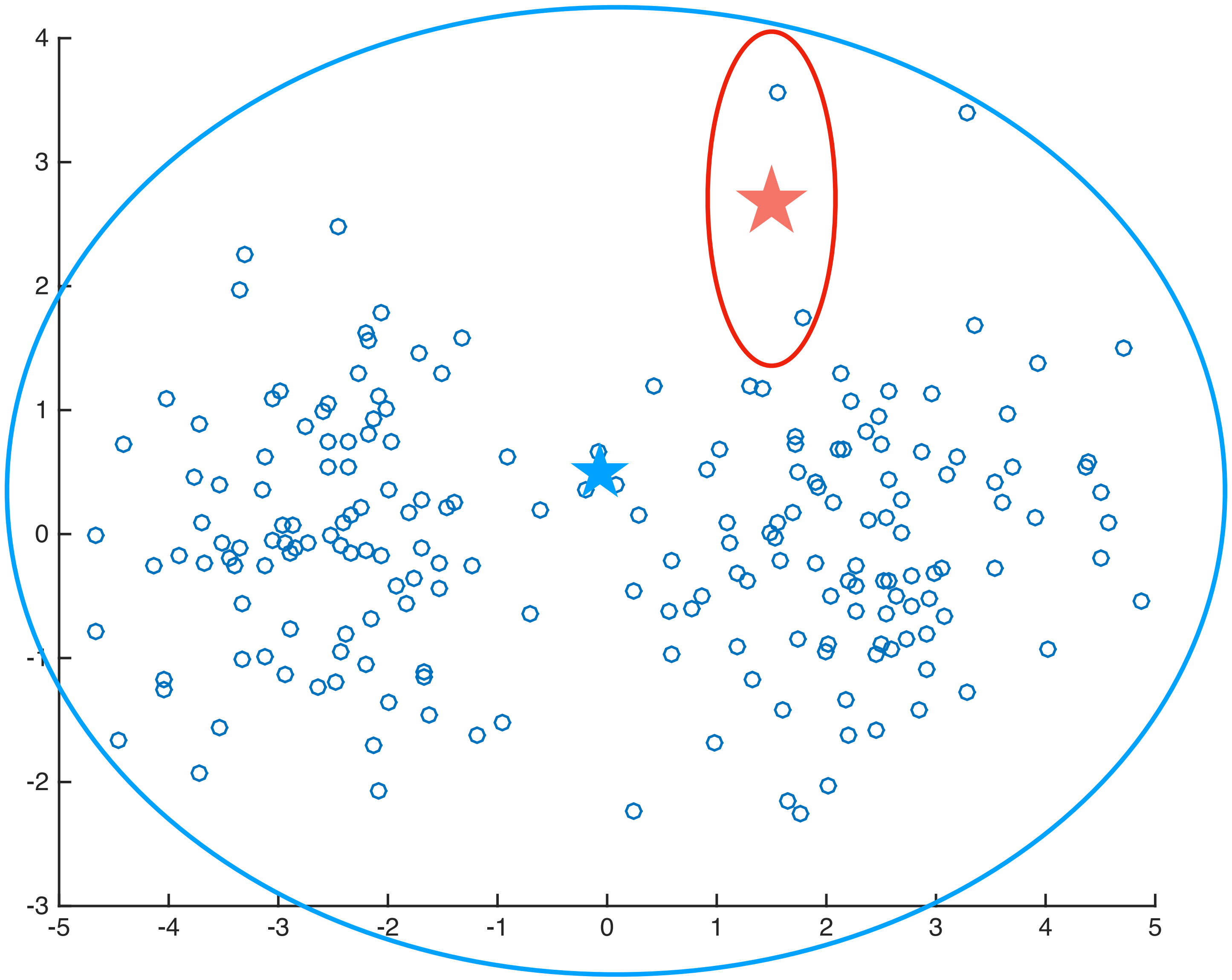
$$\hat{c}^m(\mathbf{x}_t) = \underset{j \in [K]}{\arg\max}\ p(\mathbf{x}_t, \hat{\mathbf{r}}_j^{m-1}, \hat{\Sigma}^{m-1}) \times \pi^m(j)$$

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

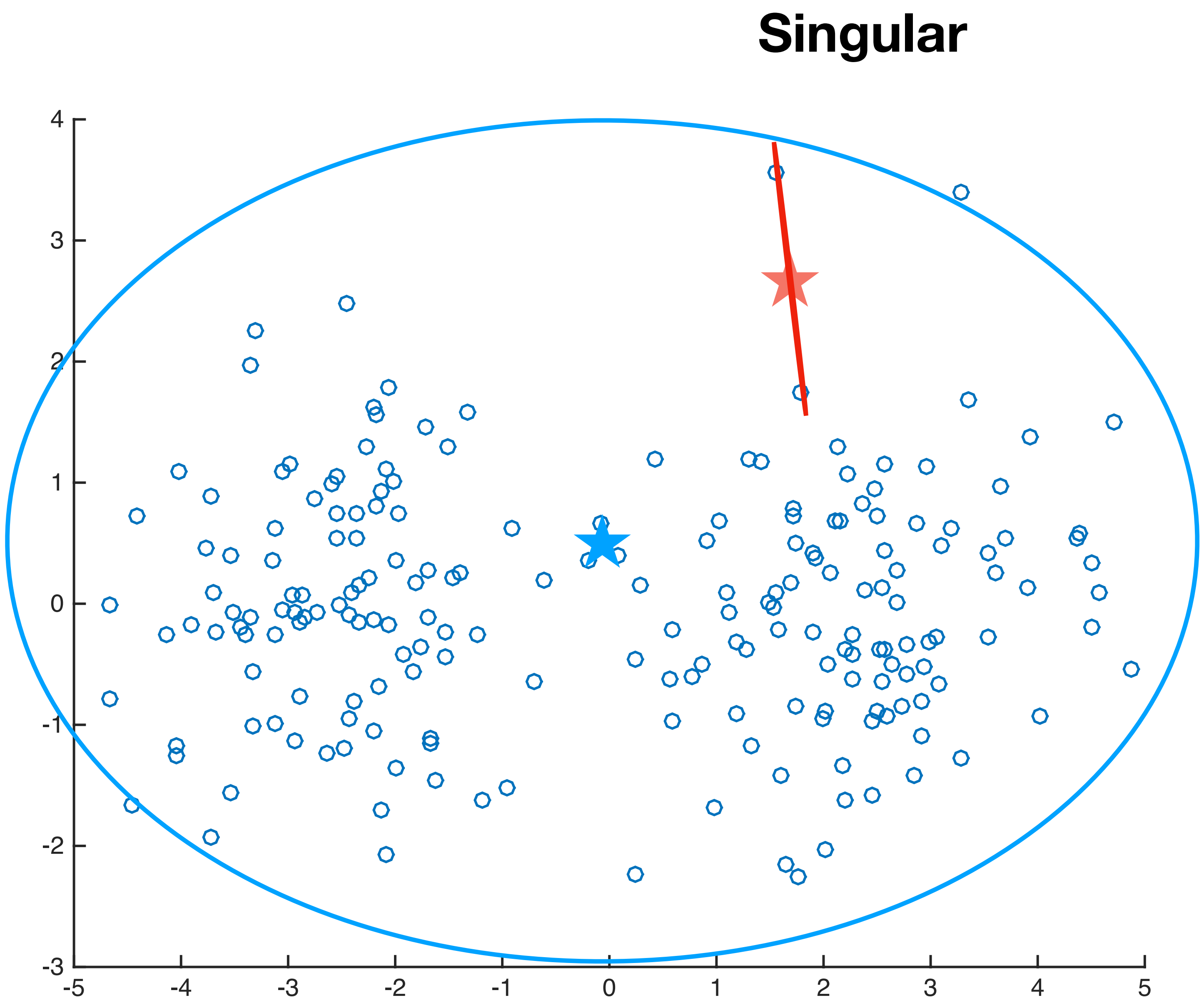
$$\hat{\mathbf{r}}_j^m = \frac{1}{|\hat{C}_j^m|} \sum_{\mathbf{x}_t \in \hat{C}_j^m} \mathbf{x}_t \qquad \hat{\Sigma}^m = \frac{1}{|C_j|} \sum_{t \in C_j} (\mathbf{x}_t - \hat{\mathbf{r}}_j^m)(\mathbf{x}_t - \hat{\mathbf{r}}_j^m)^\top \qquad \pi_j^m = \frac{|C_j^m|}{n}$$

3. $m \leftarrow m + 1$

Pitfall of Hard Assignment



Pitfall of Hard Assignment

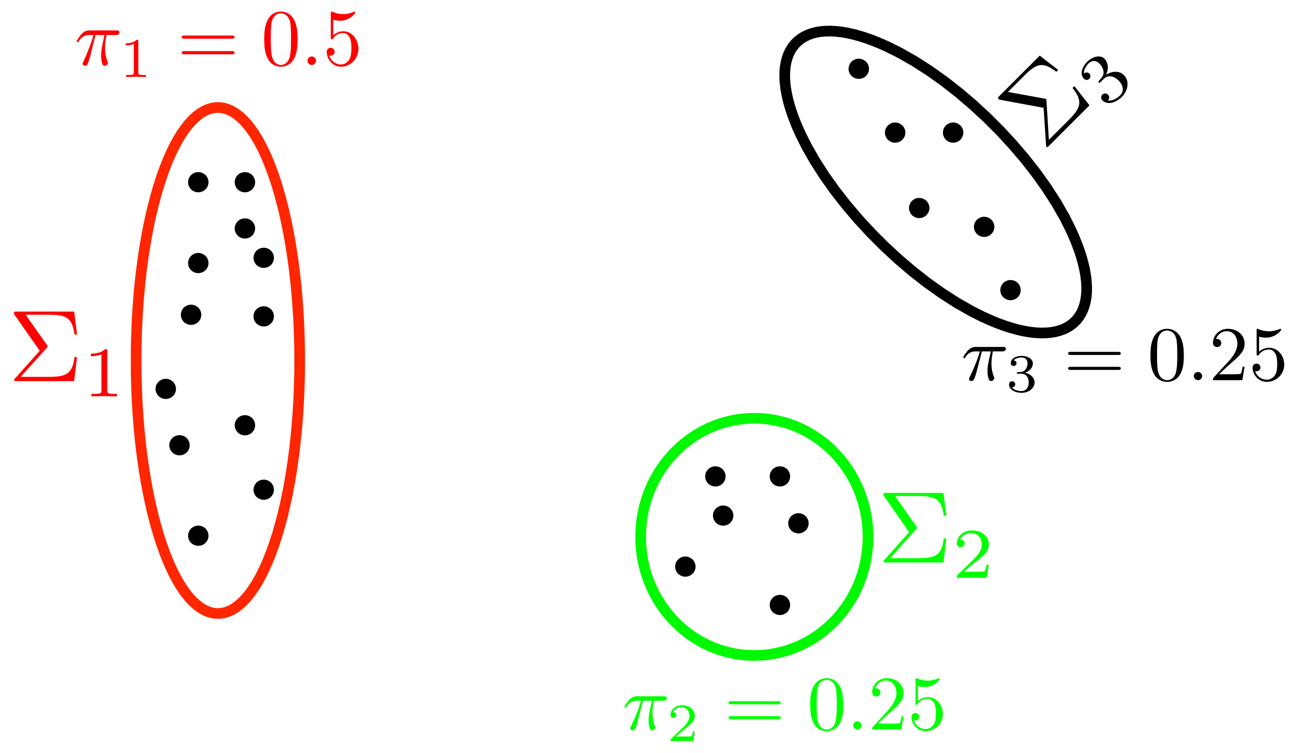


MLE FOR GMM

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 $k \in [K]$,

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

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,

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

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

Let us derive this!

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

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

Log(average) > average of Log

Why should EM work?

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

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

M-step

E-step

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 (no. of clusters)

- 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)
- We can leave the burden of choosing K to the probabilistic model