Clustering: Models and Algorithms

Shikui Tu

2019-03-07

Outline

- Gaussian Mixture Models (GMM)
- Expectation-Maximization (EM) for maximum likelihood
- Model selection, Bayesian learning

From distance to probability

distance

likely

$$\| x - \mu \|^2 \longrightarrow \exp\{-\lambda \| x - \mu \|^2\}$$

"The closer, the more likely."

Sum or integral to be one

Probability

It is more powerful to consider everything in probability framework!

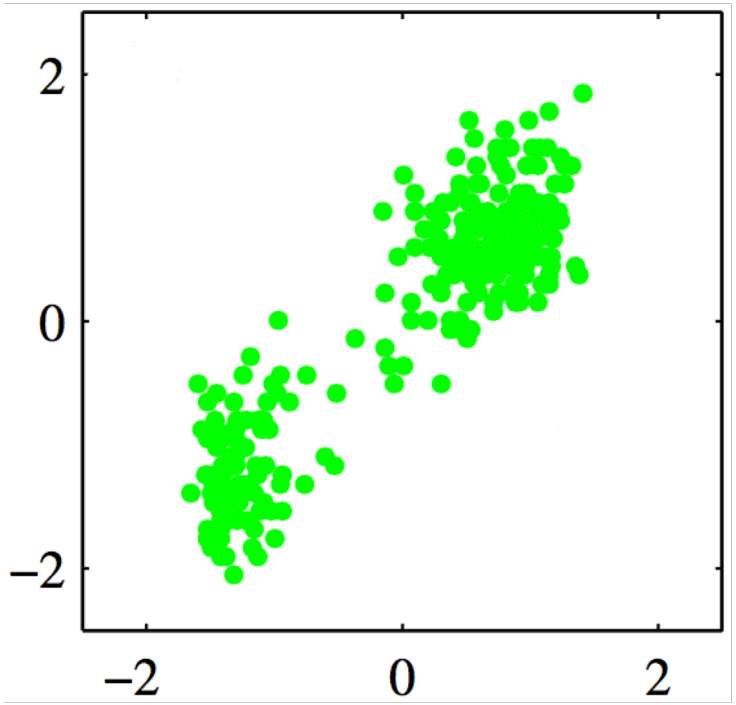
$$\frac{1}{\sqrt{(2\pi)^k|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

Gaussian distribution with the Mahalanobis distance

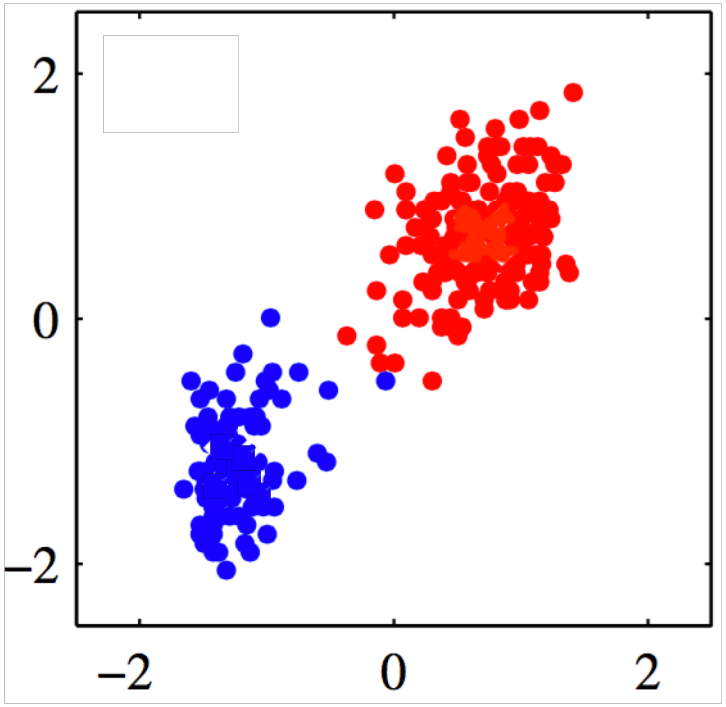
$$D_M(x) = \sqrt{(x-\mu)^T \Sigma^{-1} (x-\mu)}.$$

Review the clustering problem again

We have the following data:

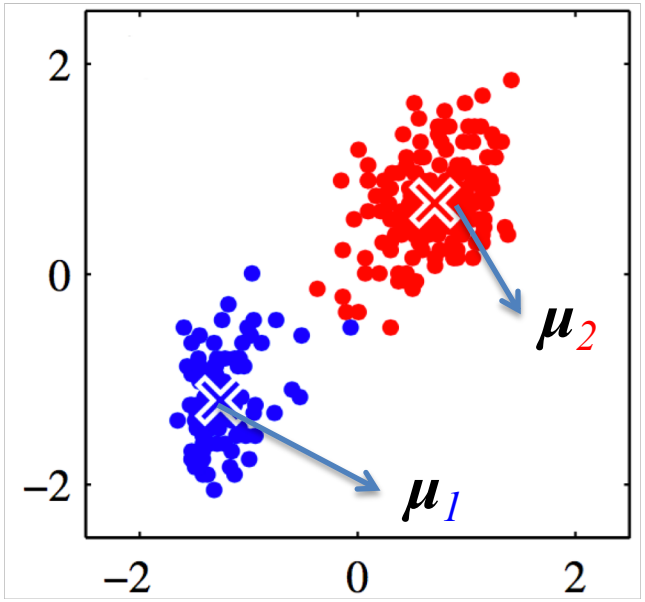


We want to cluster the data into two clusters (red and blue)

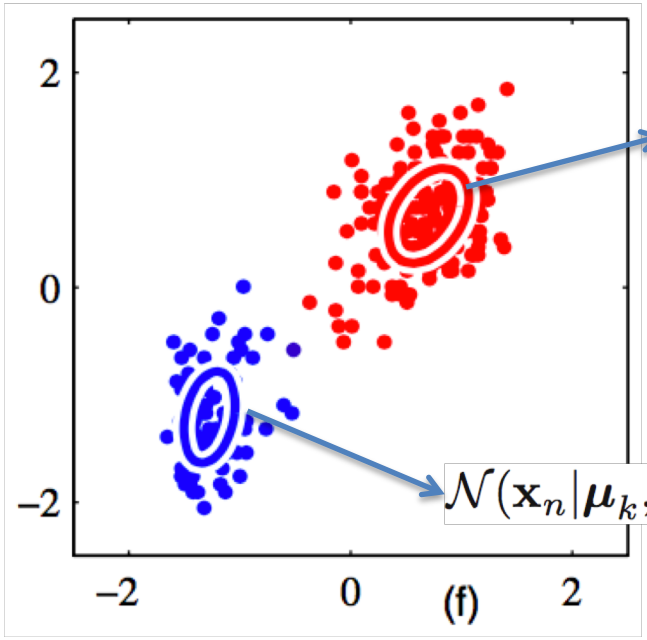


Instead if using $\{\mu_1, \mu_2\}$, each cluster is represented as a Gaussian distribution

K-means



Gaussian Mixture Model (GMM)



k=2

k=1

$$\mathcal{N}(\mathbf{x}\ |\boldsymbol{\mu}\ ,\boldsymbol{\Sigma}\ ) = \frac{1}{\sqrt{(2\pi)^k|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

Maximum likelihood

Given a data set $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^{\mathrm{T}}$ in which the observations $\{\mathbf{x}_n\}$ are assumed to be drawn independently from a multivariate Gaussian distribution, we can estimate the parameters of the distribution by maximum likelihood. The log likelihood function is given by

$$\ln p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{ND}{2}\ln(2\pi) - \frac{N}{2}\ln|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{n=1}^{N}(\mathbf{x}_n - \boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}_n - \boldsymbol{\mu})$$

Maximizing the log-likelihood function:

$$\frac{\partial}{\partial \boldsymbol{\mu}}\ln p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N}\boldsymbol{\Sigma}^{-1}(\mathbf{x}_n - \boldsymbol{\mu}) = 0 \quad \longrightarrow \quad \boldsymbol{\mu}_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n$$

Similarly we get $$\boldsymbol{\Sigma}_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \boldsymbol{\mu}_{\mathrm{ML}})(\mathbf{x}_n - \boldsymbol{\mu}_{\mathrm{ML}})^{\mathrm{T}}$$

$\boldsymbol{\mu}_{\mathrm{ML}}$ and $\boldsymbol{\Sigma}_{\mathrm{ML}}$ are the maximum likelihood estimates of the mean and the co-variance matrix.

Matrix derivatives

$$\left[\frac{\partial \mathbf{x}}{\partial y}\right]_i = \frac{\partial x_i}{\partial y} \qquad \left[\frac{\partial x}{\partial \mathbf{y}}\right]_i = \frac{\partial x}{\partial y_i} \qquad \left[\frac{\partial \mathbf{x}}{\partial \mathbf{y}}\right]_{ij} = \frac{\partial x_i}{\partial y_j}$$

$$\frac{\partial \mathbf{x}^T\mathbf{a}}{\partial \mathbf{x}} = \frac{\partial \mathbf{a}^T\mathbf{x}}{\partial \mathbf{x}} = \mathbf{a} \tag{69}$$

$$\frac{\partial \mathbf{a}^T\mathbf{X}\mathbf{b}}{\partial \mathbf{X}} = \mathbf{a}\mathbf{b}^T \tag{70}$$

$$\frac{\partial \mathbf{a}^T\mathbf{X}^T\mathbf{b}}{\partial \mathbf{X}} = \mathbf{b}\mathbf{a}^T \tag{71}$$

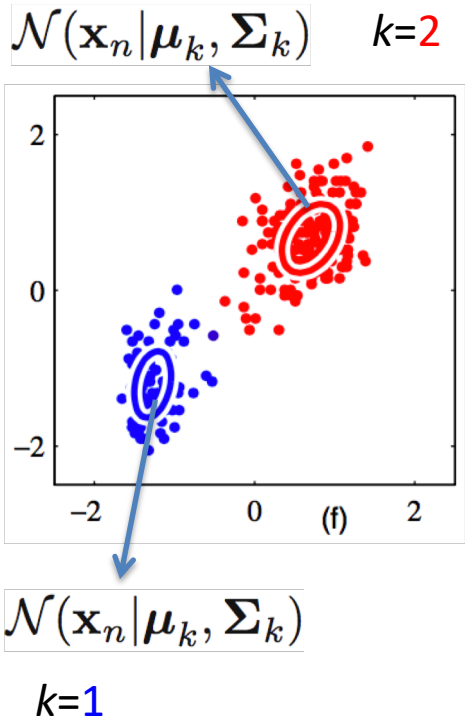
$$\frac{\partial \mathbf{a}^T\mathbf{X}\mathbf{a}}{\partial \mathbf{X}} = \frac{\partial \mathbf{a}^T\mathbf{X}^T\mathbf{a}}{\partial \mathbf{X}} = \mathbf{a}\mathbf{a}^T \tag{72}$$

$$\frac{\partial \det(\mathbf{X})}{\partial \mathbf{X}} = \det(\mathbf{X})(\mathbf{X}^{-1})^T \tag{49}$$

$$\frac{\partial \mathbf{Y}^{-1}}{\partial x} = -\mathbf{Y}^{-1}\frac{\partial \mathbf{Y}}{\partial x}\mathbf{Y}^{-1} \tag{59}$$

http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/3274/pdf/imm3274.pdf

Gaussian Mixture Model (GMM)



We use $z_k = 1$ to indicate a point $\mathbf{x}$ belongs to cluster k

$$\mathbf{z} = (z_1, \ldots, z_K) \qquad z_k \in \{0, 1\} \qquad \sum_k z_k = 1$$

Assume the points in the same cluster follow a **Gaussian distribution**

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

A mixing weight for each cluster:

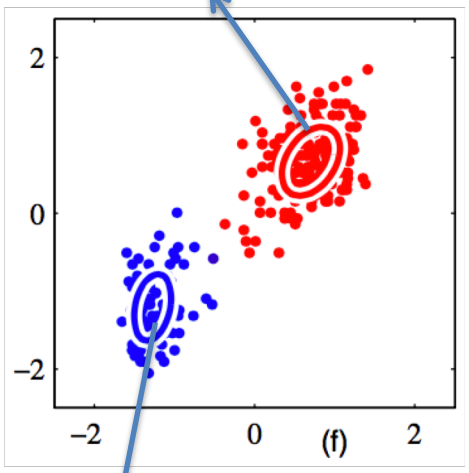
$$p(z_k = 1) = \pi_k \qquad 0 \leqslant \pi_k \leqslant 1 \qquad \sum_{k=1}^{K} \pi_k = 1$$

prior probability of point belonging to a cluster

So, we get a distribution for the data point $\mathbf{x}$:

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z})p(\mathbf{x}|\mathbf{z}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Introduce a latent variable



We use $z_k = 1$ to indicate a point $\mathbf{x}$ belongs to cluster k

$\mathbf{z} = (z_1, \ldots, z_K)$ $\quad z_k \in \{0, 1\}$ $\quad \sum_k z_k = 1$

A mixing weight for each cluster:

$p(z_k = 1) = \pi_k$ $\quad 0 \leqslant \pi_k \leqslant 1$ $\quad \sum_{k=1}^{K} \pi_k = 1$

prior probability of point belonging to a cluster

$$p(\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k}$$

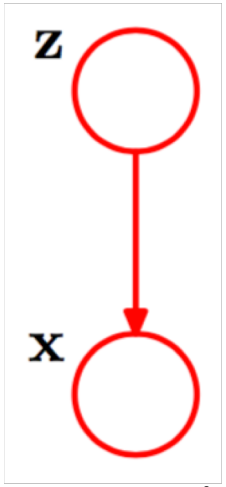
Assume the points in the same cluster follow a **Gaussian distribution**

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Gaussian Mixture Model (GMM)

Generative process

- Randomly sample a $\mathbf{z}$ from a categorical distribution $[\pi_1, \ldots, \pi_K]$;
- Generate $\boldsymbol{x}$ according to Gaussian distribution $p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$



Graphical representation of $\mathbf{p}(\mathbf{x}, \mathbf{z}) = \mathbf{p}(\boldsymbol{z})\mathbf{p}(\mathbf{x}|\mathbf{z})$

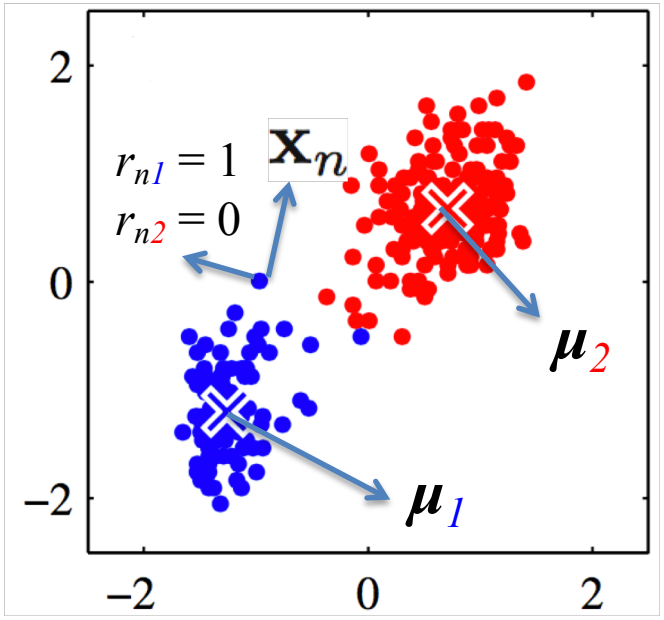
So, we get a distribution for the data point $\mathbf{x}$:

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z})p(\mathbf{x}|\mathbf{z}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

From minimizing sum of square distances to finding maximum likelihood

minimize

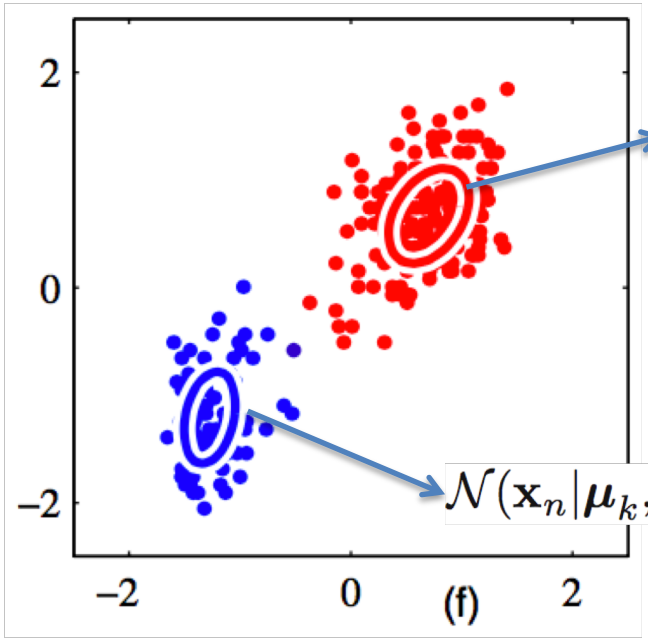
$$J = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$



maximize likelihood

$$p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})$$

$$X = \{x_1, \ldots, x_N\}$$
$$\pi = \{\pi_1, \ldots, \pi_K\}$$
$$\mu = \{\mu_1, \ldots, \mu_K\}$$
$$\Sigma = \{\Sigma_1, \ldots, \Sigma_K\}$$



Remember: **The closer the distance, the more likely the probability.**

Outline

- Gaussian Mixture Models (GMM)

- Expectation-Maximization (EM) for maximum likelihood

- Model selection, Bayesian learning

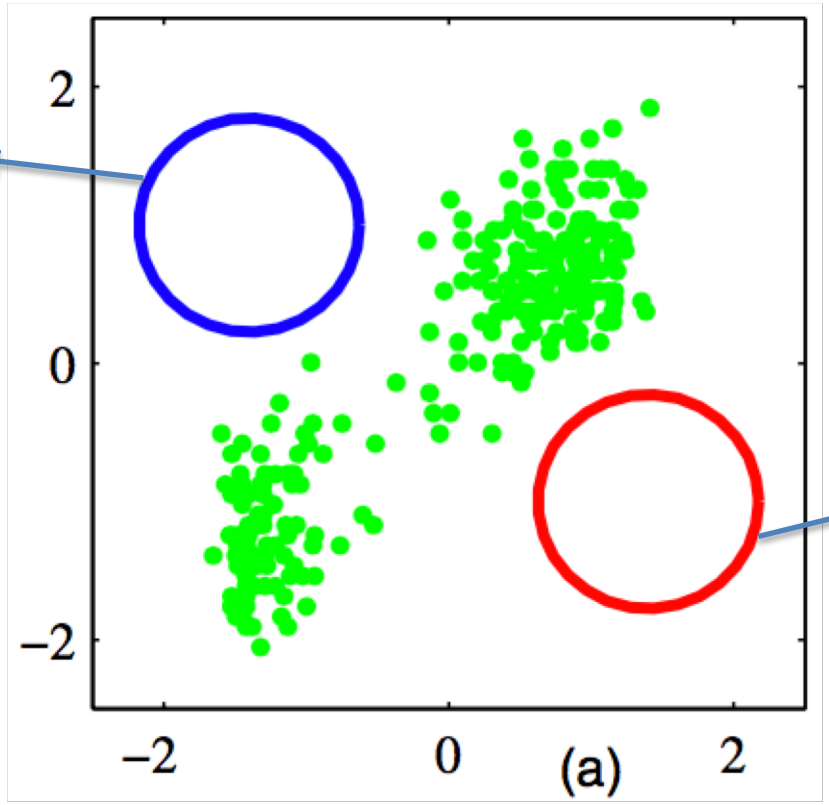
Expectation-Maximization (EM) algorithm for maximum likelihood

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

Initialization

$\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

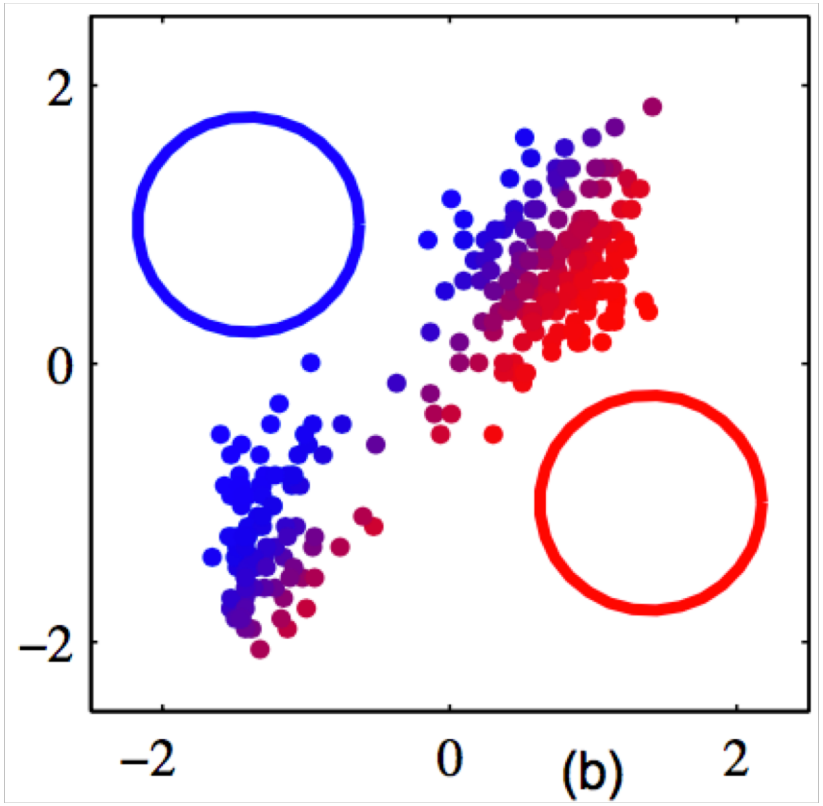
k=1



$\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

k=2

E Step



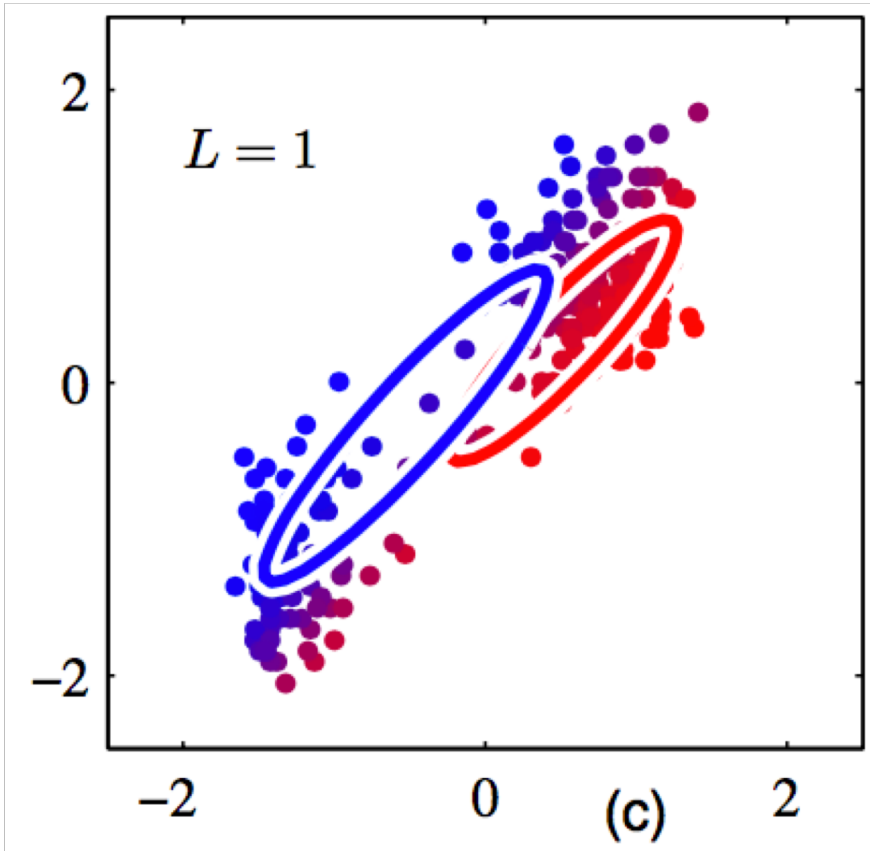
When the parameters are given, the assignments of the points can be calculated by the posterior probability, i.e., the probability of a data point belonging to a cluster once we have observed the data point.

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

Soft assignment:
A point fractionally belongs to two clusters.

For example,
0.2 belong to cluster 1
0.8 belong to cluster 2

M Step



When the assignments $\gamma(z_{nk})$ of the points to the clusters are known, parameters could be calculated for each cluster (Gaussian) separately.

Mixing weight π_k: the proportion of number of points in cluster k within all data points

$$\pi_k = \frac{N_k}{N} \quad ; \quad N_k = \sum_{n=1}^{N} \gamma(z_{nk}).$$

$\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$: the mean and the covariance matrix are calculated for each cluster

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n$$

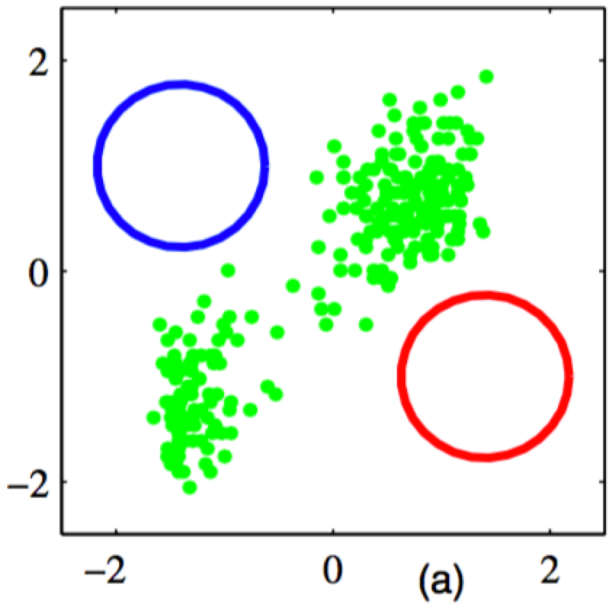
$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}}$$

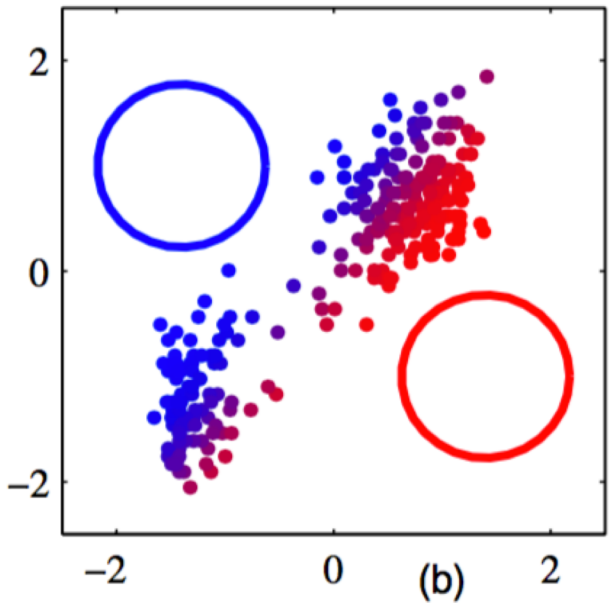
L denotes the number of cycles of the EM algorithm.

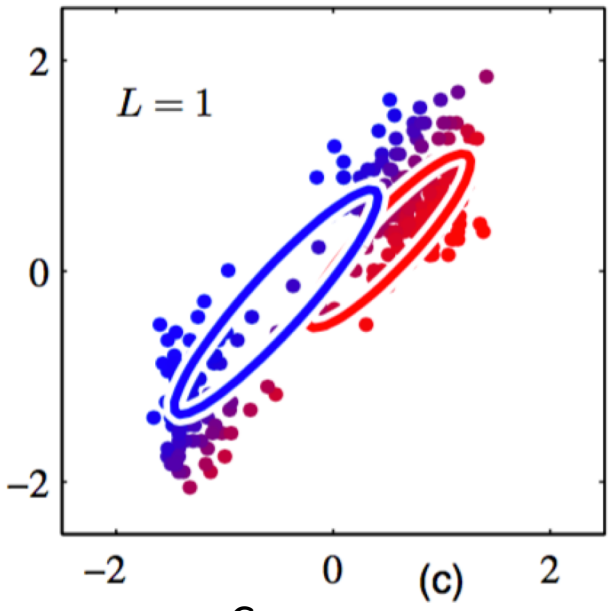
initialization

E-Step

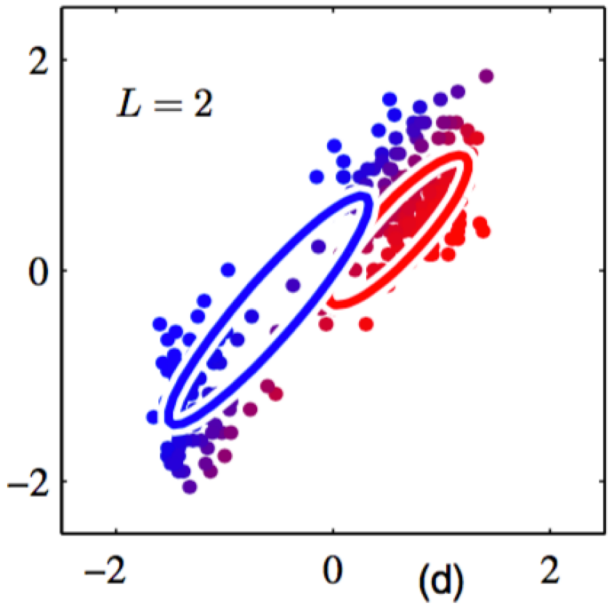
M-Step

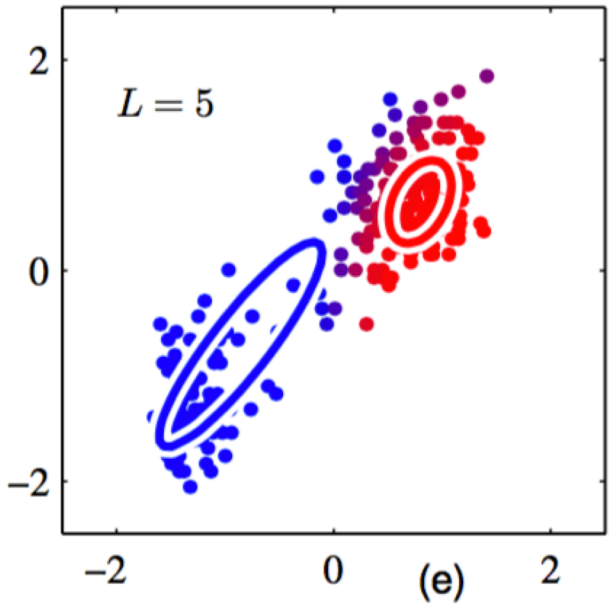


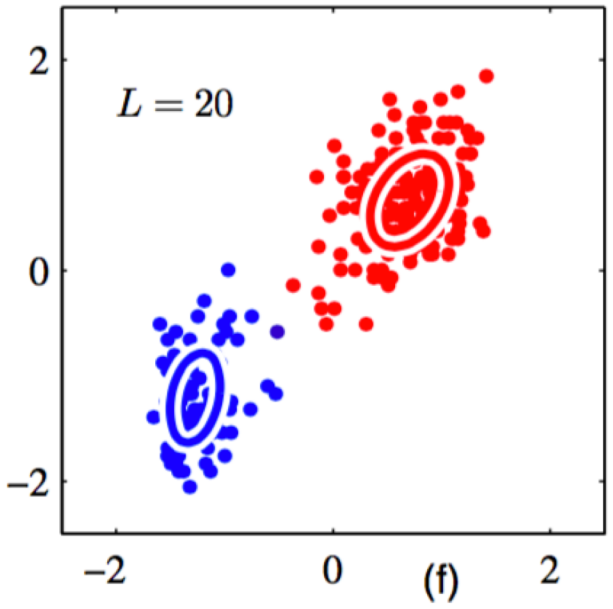




Convergence







L denotes the number of cycles of E-Step and M-Step.

Details of the EM Algorithm

1. Initialize the means $\boldsymbol{\mu}_k$, covariances $\boldsymbol{\Sigma}_k$ and mixing coefficients π_k, and evaluate the initial value of the log likelihood.
2. **E step**. Evaluate the responsibilities using the current parameter values

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.$$

3. **M step**. Re-estimate the parameters using the current responsibilities

$$\begin{aligned}
\boldsymbol{\mu}_k^{\text{new}} &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n \\
\boldsymbol{\Sigma}_k^{\text{new}} &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right) \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right)^{\mathrm{T}} \\
\pi_k^{\text{new}} &= \frac{N_k}{N}
\end{aligned}$$

where

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk}).$$

4. Evaluate the log likelihood

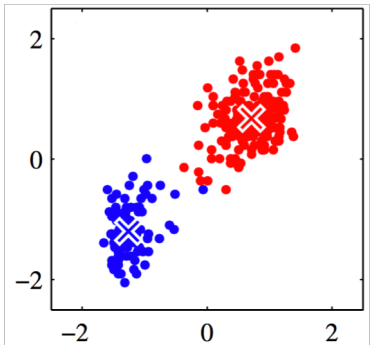
$$\ln p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

and check for convergence of either the parameters or the log likelihood. If the convergence criterion is not satisfied return to step 2.

K-means is a hard-cut EM

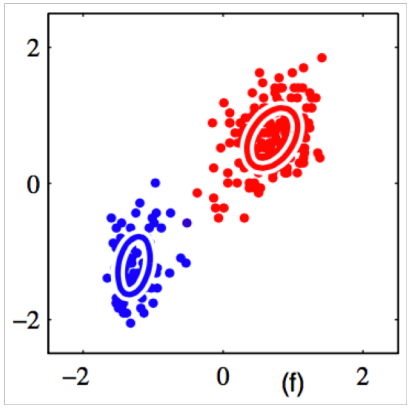
Fixed equal mixing weights

$\|\mathbf{x} - \boldsymbol{\mu}_k\|^2$



$\{\boldsymbol{\mu}_k\}$

$\boldsymbol{\Sigma}_k = \varepsilon \mathbf{I}$



GMM considers covariance and mixing weights.

$$p(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi\epsilon)^{1/2}} \exp\left\{-\frac{1}{2\epsilon}\|\mathbf{x} - \boldsymbol{\mu}_k\|^2\right\}$$

One-in-K assignment

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{otherwise.} \end{cases}$$

Soft assignment

$$\gamma(z_{nk}) = \frac{\pi_k \exp\left\{-\|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2/2\epsilon\right\}}{\sum_j \pi_j \exp\left\{-\|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2/2\epsilon\right\}}$$

The General EM Algorithm

Given a joint distribution $p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$ over observed variables $\mathbf{X}$ and latent variables $\mathbf{Z}$, governed by parameters $\boldsymbol{\theta}$, the goal is to maximize the likelihood function $p(\mathbf{X}|\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$.

1. Choose an initial setting for the parameters $\boldsymbol{\theta}^{\text{old}}$.
2. **E step** Evaluate $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}})$.
3. **M step** Evaluate $\boldsymbol{\theta}^{\text{new}}$ given by

$$\boldsymbol{\theta}^{\text{new}} = \arg\max_{\boldsymbol{\theta}} \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}})$$

where

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}).$$

4. Check for convergence of either the log likelihood or the parameter values. If the convergence criterion is not satisfied, then let

$$\boldsymbol{\theta}^{\text{old}} \leftarrow \boldsymbol{\theta}^{\text{new}}$$

and return to step 2.

Summary for the EM algorithm for GMM

- Does it find the global optimum?
  - No, like K-means, EM only finds the nearest local optimum and the optimum depends on the initialization
- GMM is more general then K-means by considering mixing weights, covariance matrices, and soft assignments.
- Like K-means, it does not tell you the best K.

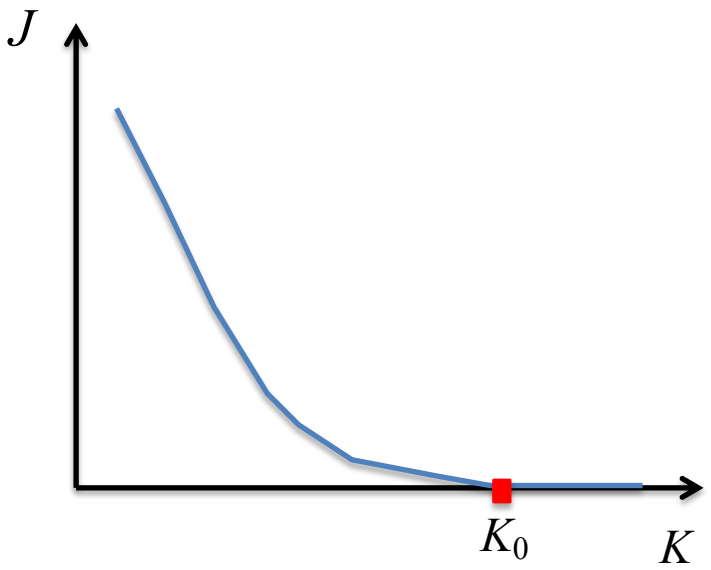
Outline

- Gaussian Mixture Models (GMM)
- Expectation-Maximization (EM) for maximum likelihood
- Model selection, Bayesian learning

How to determine the cluster number K?

K-mean

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

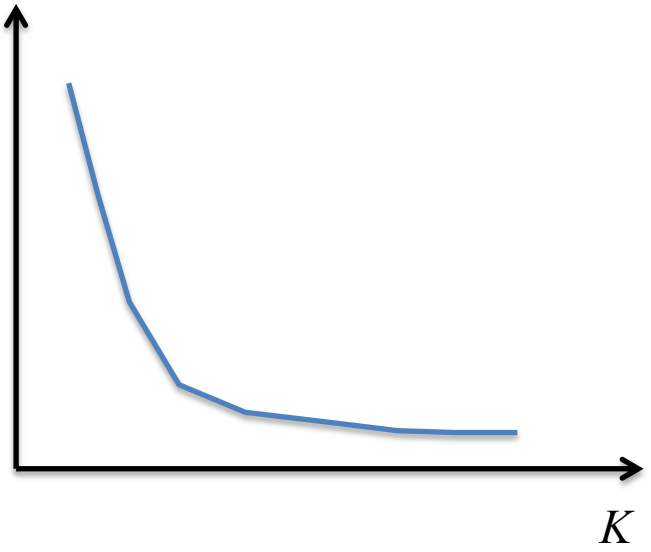


J does not tell which K is better.

GMM

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

- Log-likelihood



Negative log-likelihood also decreases as K increases.

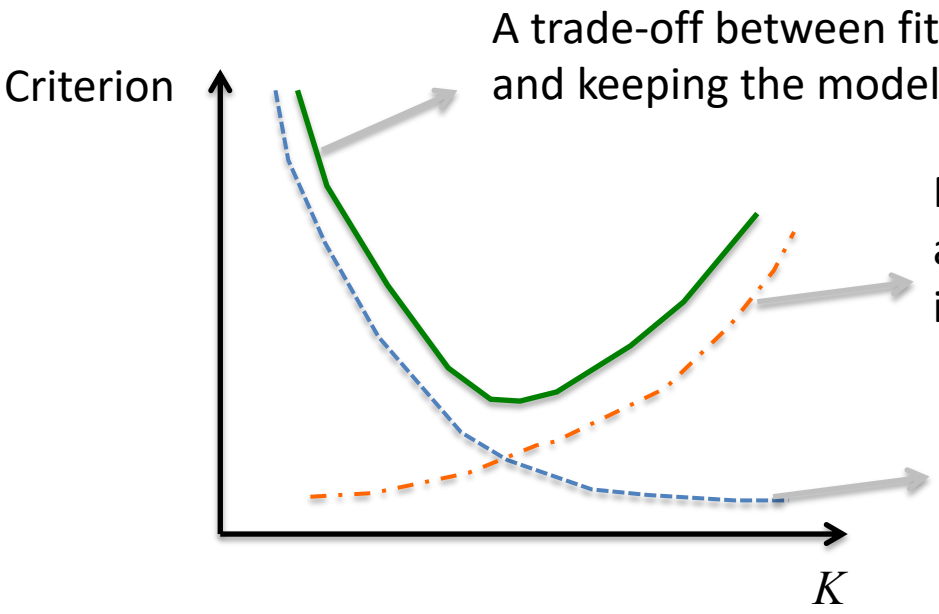
Model selection in general

Probabilistic model

$$p(X_N \mid \Theta_K)$$

Candidate models:

$$\Theta_1 \subseteq \Theta_2 \subseteq \cdots \subseteq \Theta_K \subseteq \cdots$$



A trade-off between fitting the data well and keeping the model simple

Models become more and more complex as K increases.

Negative log-likelihood (or fitting error) to make sure the model fit the data well.

Akaike's Information Criterion (AIC) $\ln p(X_N \mid \hat{\Theta}_K) - d_k$

Bayesian Information Criterion (BIC) $\ln p(X_N \mid \hat{\Theta}_K) - \frac{1}{2} d_k \ln N$

d_k: number of free parameters

N: sample size

Bayesian learning

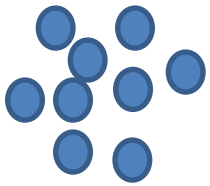
- Maximum A Posteriori (MAP)

$$\max_{\Theta} p(\Theta|X)$$

Equivalent to:

$$\log p(X, \Theta) = \log p(X|\,\Theta) + \log p(\,\Theta)$$

Consider a simple example:



$$p(x|\Theta) = G(x|\mu, \Sigma)$$

$$p(\mu) = G(\mu|\mu_0, \sigma_0^2)$$

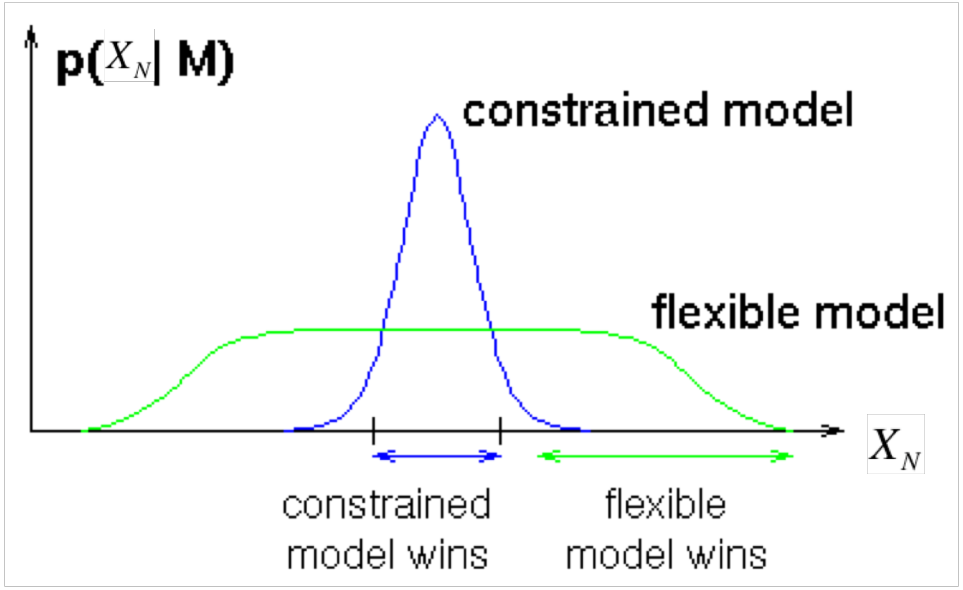
Model selection

Probabilistic model

$$p(X_N \mid \Theta_K)$$

Candidate models:

$$\Theta_1 \subseteq \Theta_2 \subseteq \cdots \subseteq \Theta_K \subseteq \cdots$$



Using Occam's Razor to Learn Model Structure

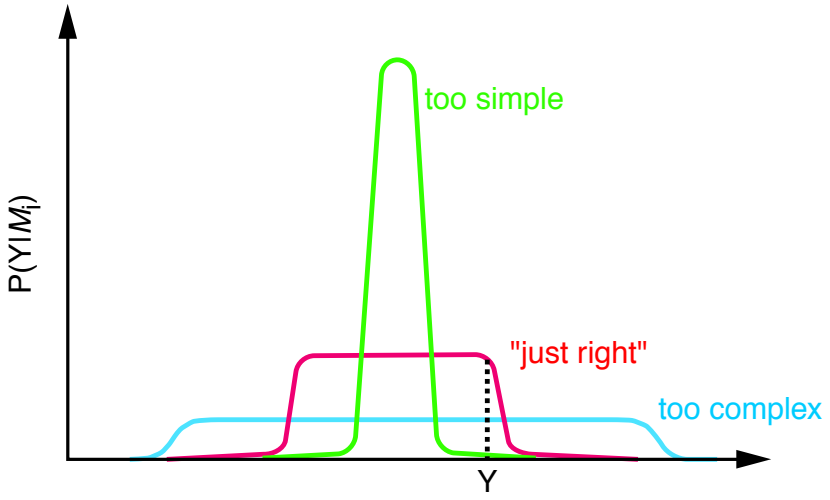
Compare model classes m using their posterior probability given the data:

$$P(m|\mathbf{y}) = \frac{P(\mathbf{y}|m)P(m)}{P(\mathbf{y})}, \qquad P(\mathbf{y}|m) = \int_{\Theta_m} P(\mathbf{y}|\boldsymbol{\theta}_m, m)P(\boldsymbol{\theta}_m|m)\, d\boldsymbol{\theta}_m$$

Interpretation of $P(\mathbf{y}|m)$: The probability that *randomly selected* parameter values from the model class would generate data set $\mathbf{y}$.

Model classes that are too simple are unlikely to generate the data set.

Model classes that are too complex can generate many possible data sets, so again, they are unlikely to generate that particular data set at random.



Bayesian model selection

- A model class m is a set of models parameterised by $\boldsymbol{\theta}_m$, e.g. the set of all possible mixtures of m Gaussians.

- The marginal likelihood of model class m:

$$P(\mathbf{y}|m) = \int_{\Theta_m} P(\mathbf{y}|\boldsymbol{\theta}_m, m) P(\boldsymbol{\theta}_m|m) \, d\boldsymbol{\theta}_m$$

  is also known as the Bayesian evidence for model m.

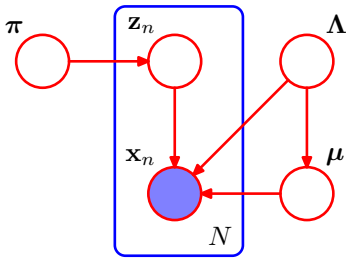
- The ratio of two marginal likelihoods is known as the Bayes factor:

$$\frac{P(\mathbf{y}|m)}{P(\mathbf{y}|m')}$$

- The Occam's Razor principle is, roughly speaking, that one should prefer simpler explanations than more complex explanations.

- Bayesian inference formalises and *automatically* implements the Occam's Razor principle.

VBEM for GMM



- Model descriptions:

$$p(\mathbf{Z}|\boldsymbol{\pi}) = \prod_{n=1}^{N}\prod_{k=1}^{K}\pi_k^{z_{nk}} \qquad p(\mathbf{X}|\mathbf{Z},\boldsymbol{\mu},\boldsymbol{\Lambda}) = \prod_{n=1}^{N}\prod_{k=1}^{K}\mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k^{-1}\right)^{z_{nk}}$$

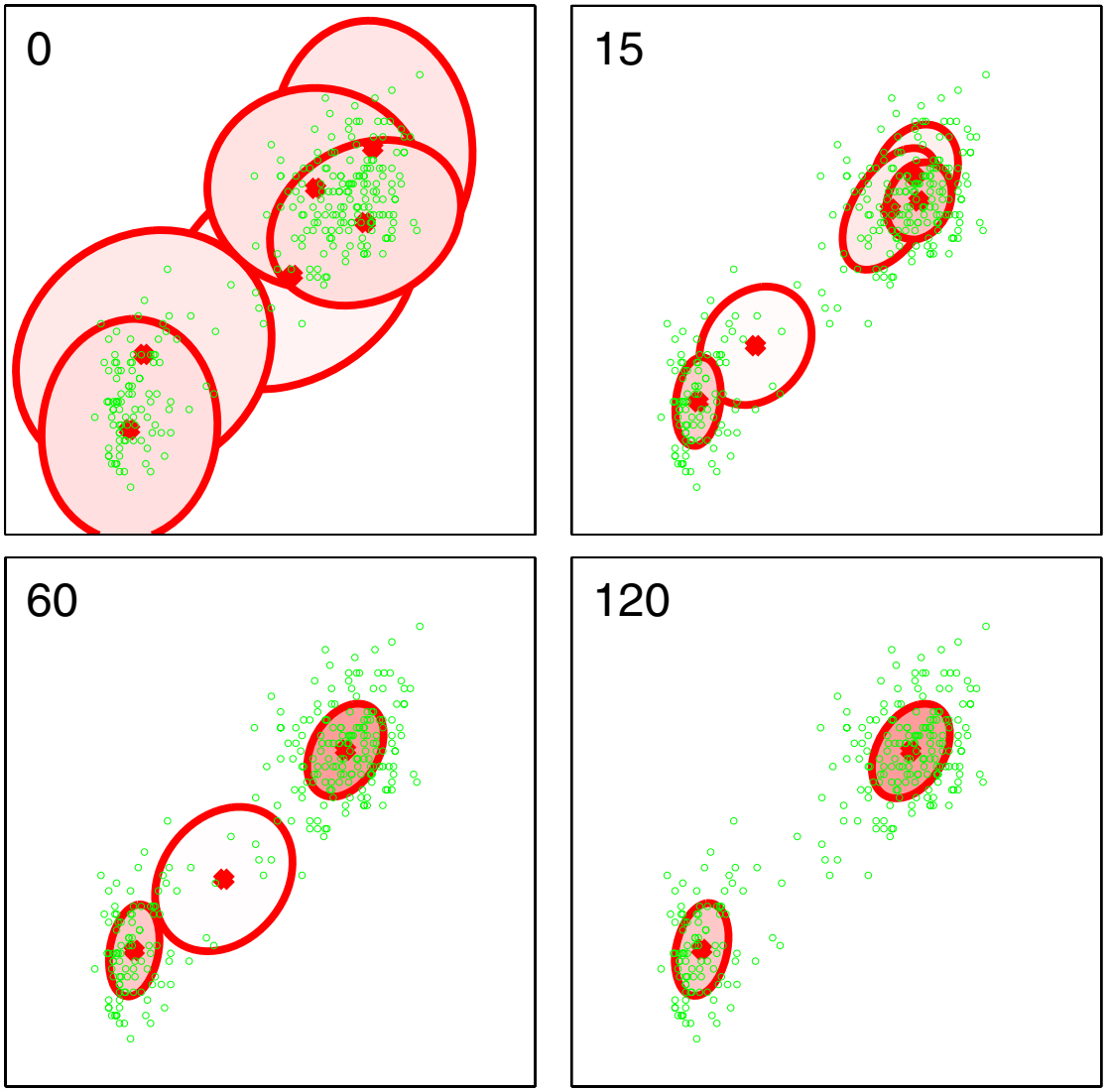
- Prior distributions over parameters:

$$p(\boldsymbol{\pi}) = \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}_0) = C(\boldsymbol{\alpha}_0)\prod_{k=1}^{K}\pi_k^{\alpha_0-1}$$

$$\begin{aligned} p(\boldsymbol{\mu},\boldsymbol{\Lambda}) &= p(\boldsymbol{\mu}|\boldsymbol{\Lambda})p(\boldsymbol{\Lambda}) \\ &= \prod_{k=1}^{K}\mathcal{N}\left(\boldsymbol{\mu}_k|\mathbf{m}_0,(\beta_0\boldsymbol{\Lambda}_k)^{-1}\right)\mathcal{W}(\boldsymbol{\Lambda}_k|\mathbf{W}_0,\nu_0) \end{aligned}$$

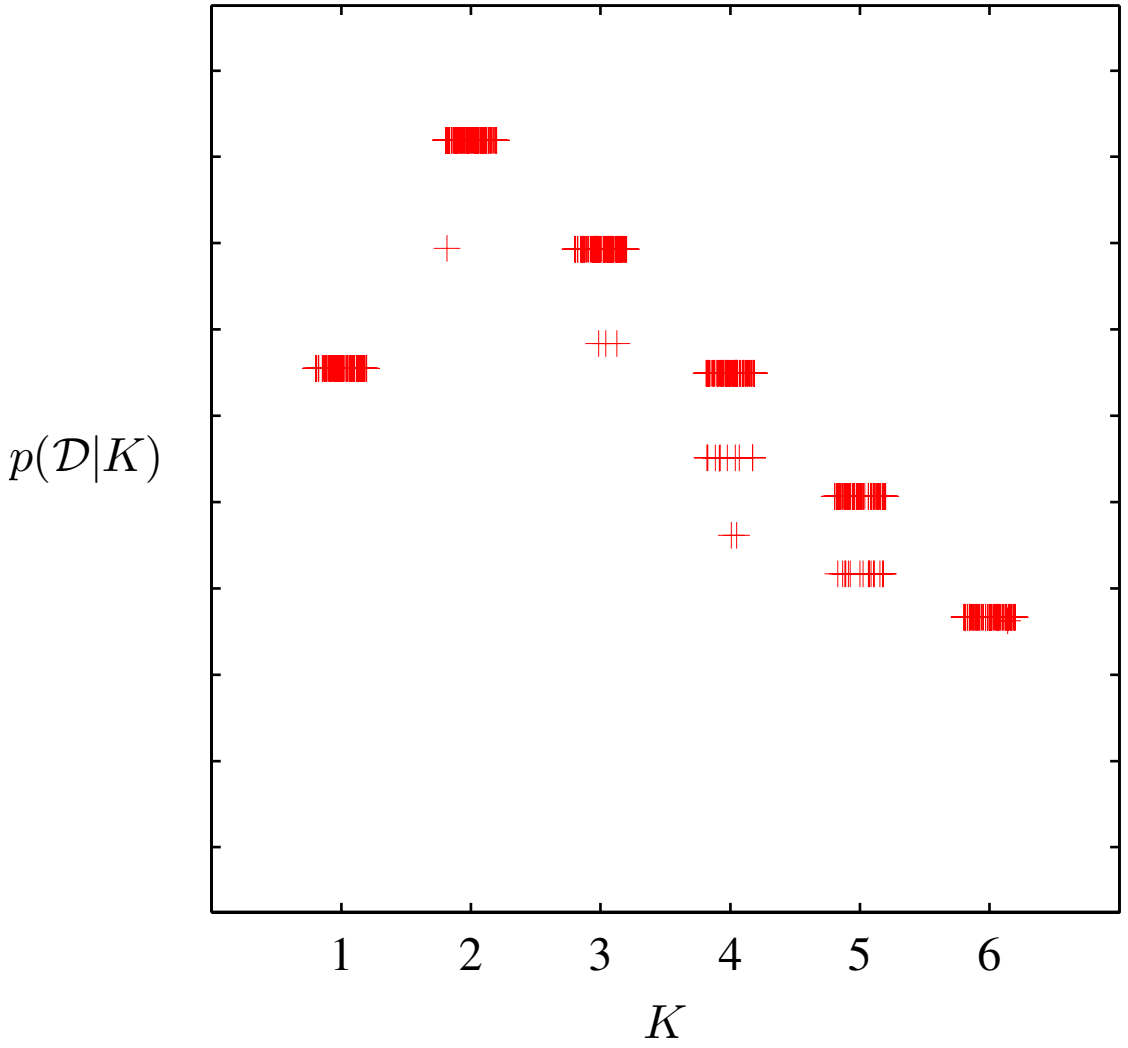
$$p(\mathbf{X},\mathbf{Z},\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda}) = p(\mathbf{X}|\mathbf{Z},\boldsymbol{\mu},\boldsymbol{\Lambda})p(\mathbf{Z}|\boldsymbol{\pi})p(\boldsymbol{\pi})p(\boldsymbol{\mu}|\boldsymbol{\Lambda})p(\boldsymbol{\Lambda})$$

How VBEM for GMM works



http://www.cs.ubc.ca/~murphyk/Software/VBEMGMM/index.html
http://scikit-learn.org/stable/modules/mixture.html

Determine K by the variational lower bound (free energy)



Thank you!

EM的九层理解

1. EM 就是 E + M
2. EM 是一种局部下限构造
3. K-Means是一种Hard EM算法
4. 从EM 到 广义EM
5. 广义EM的一个特例是VBEM
6. 广义EM的另一个特例是WS算法
7. 广义EM的再一个特例是Gibbs抽样算法
8. WS算法是VAE和GAN组合的简化版
9. KL距离的统一

http://www.elecfans.com/d/604076.html