MONASH University
Information Technology

FIT5201

Data Analysis Algorithms

Week 7 – Latent Variable Models and Expectation Maximization

Outline

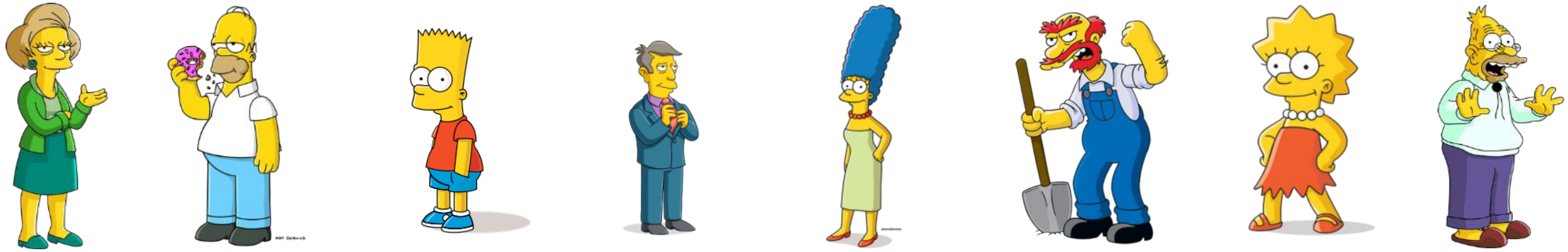
- Clustering
- KMeans
- Gaussian Mixture Models and Expectation-Maximization

Data Clustering

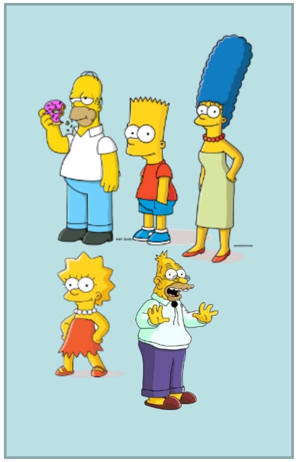
- Is a method of unsupervised learning
- Find a sensible structure from unlabelled data
- A clustering algorithm
  - Groups data into their natural categories
  - Based on the similarities between them
  - Without knowledge of their actual groups
  - Revealing the structure of the data
    - High intra-cluster similarity
    - Low inter-cluster similarity

Data Clustering…

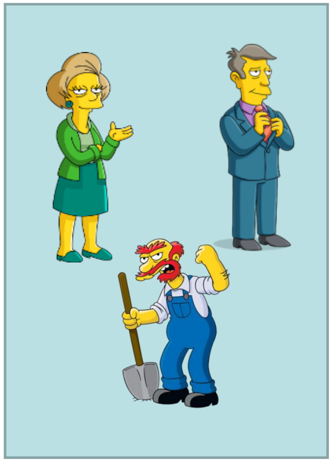
- What is a natural grouping among these objects?



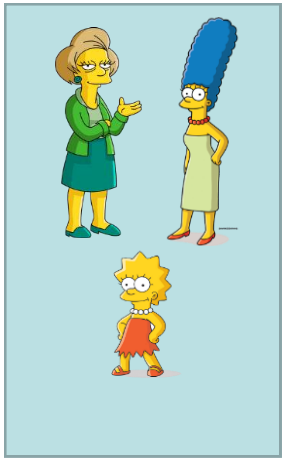
Clustering is subjective



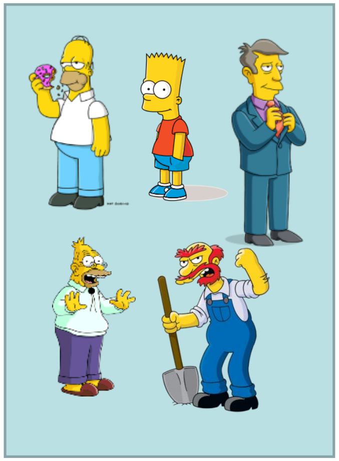
Simpsons Family



School Employees

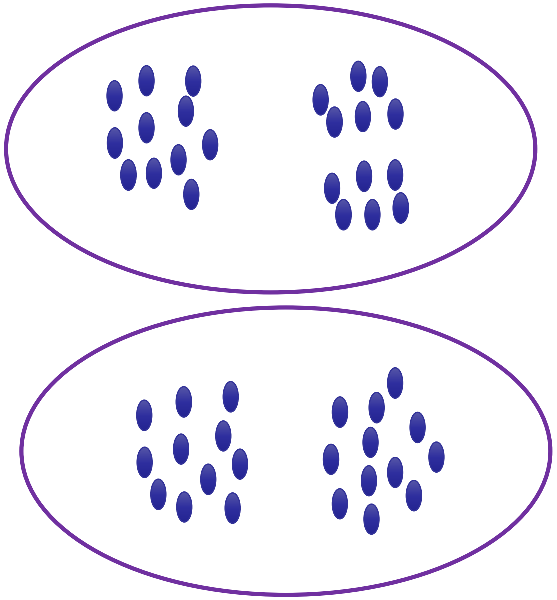


Females

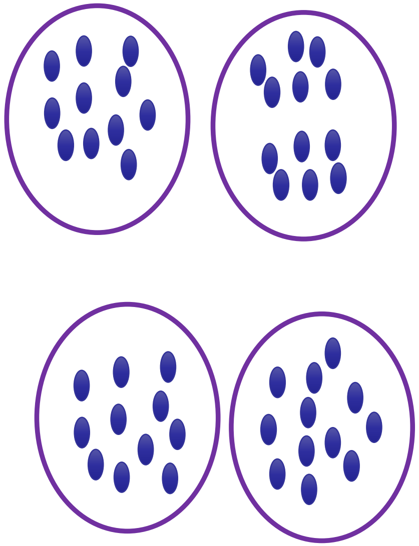


Males

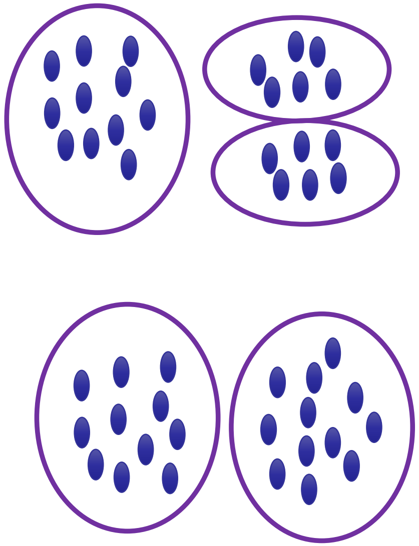
What is a good cluster?



What is a good cluster?



What is a good cluster?

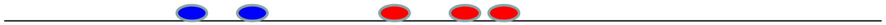


Clustering Algorithms

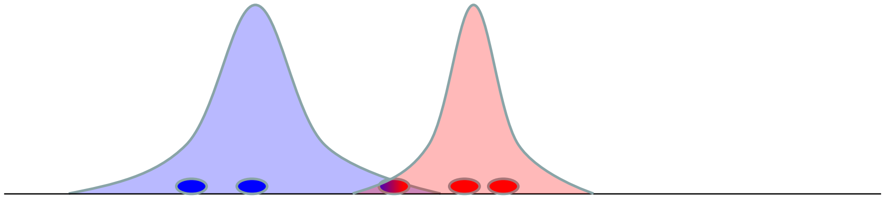
- Many algorithms exist
  - Centre-based (KMeans)
  - Density based (DBSCAN)
  - Hierarchical clustering
  - Graph based clustering

Soft vs Hard Clusters

- Hard Clusters
  - Data points belong to only one cluster



- Soft Clusters
  - Data points could belong to one or more clusters
  - Probability of belonging to each cluster is given



The KMeans Algorithm

- The simplest centre-based algorithm to solve clustering problems is KMeans

- N unlabelled data points x_n are given
- Goal: Partition the data points into K distinct groups (clusters)
  - Similar points are grouped together
  - Similarity is based on a distance measure $d(.)$

The KMeans Algorithm

- Is an iterative algorithm
- Starts with an initial random guess of K cluster centres $\left(\mu_1^{(0)}, \mu_2^{(0)}, \dots, \mu_K^{(0)}\right)$
- Iterate the following two steps until a stopping criterion is met:
  - Update assignment of data points to clusters
    - > Calculate the distance of each data point to all cluster centres
    - > Assign the data point to the cluster with the minimum distance
  - Update centers of the clusters
    - > For each cluster, calculate the new centre as the average of all data points assigned to it
    - > $\mu_K^{(\tau+1)} = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$
    - > $r_{nk} = \begin{cases} 1 & \text{if } x_n \text{ is assigned to cluster } k \\ 0 & \text{Otherwise} \end{cases}$

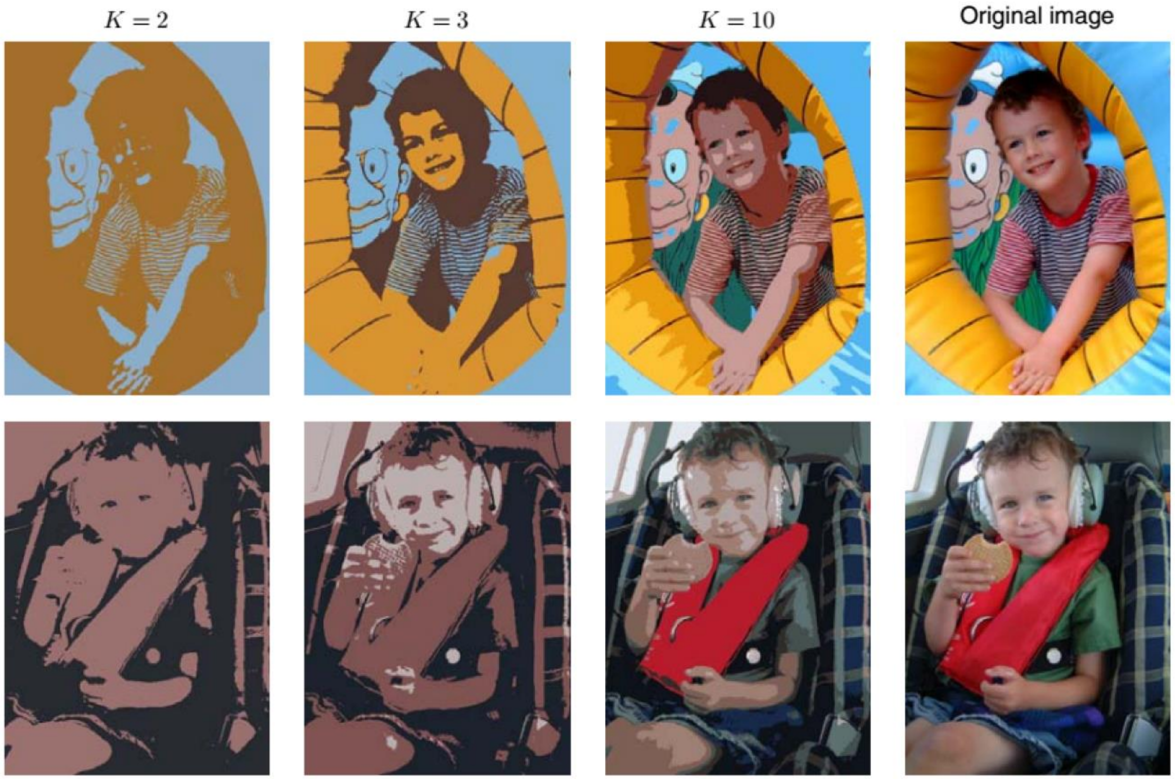
KMeans visualization

- A good visual simulation is available at http://tech.nitoyon.com/en/blog/2013/11/07/k-means/

KMeans Remarks

- KMeans is sensitive to initial values
  - which means the different execution of Kmeans with different initial cluster centers may result in different solutions
- KMeans is a non-probabilistic algorithm
  - which only supports *hard-assignment*
  - a data point can only be assigned to one and only one of the clusters

Applications of KMeans



- Data points: pixels colors
- Cluster: similar pixel colors
- Replace the colors in a cluster with the centroid
- Store the centroid only: reduced resolution and storage space

Latent Variables

- We wanted to partition a set of training data points into K groups of similar data points
- The label of the training data points are latent or hidden
- We call these latent variables

Gaussian Mixture Models (GMM)

- A Generative Story
  - Consider the following hypothetical generative story for generating a label-data point pair $(k, \boldsymbol{x})$
  - First
    - generate a cluster label k, by tossing a dice with K faces where each face of the dice corresponds to a cluster label
  - Second,
    - generate the data point $\boldsymbol{x}$, by sampling from the distribution $p_k(.)$ corresponding to the cluster label k
  - We are given data point $\boldsymbol{x}$ but not labels
  - We model it by $z \in \{1, \dots, K\}$
  - Now given the training data,
    - we would like to find the best value for the latent variables, *and*
    - the best estimates for the parameters of the above generative story.

The Probabilistic Generative Model

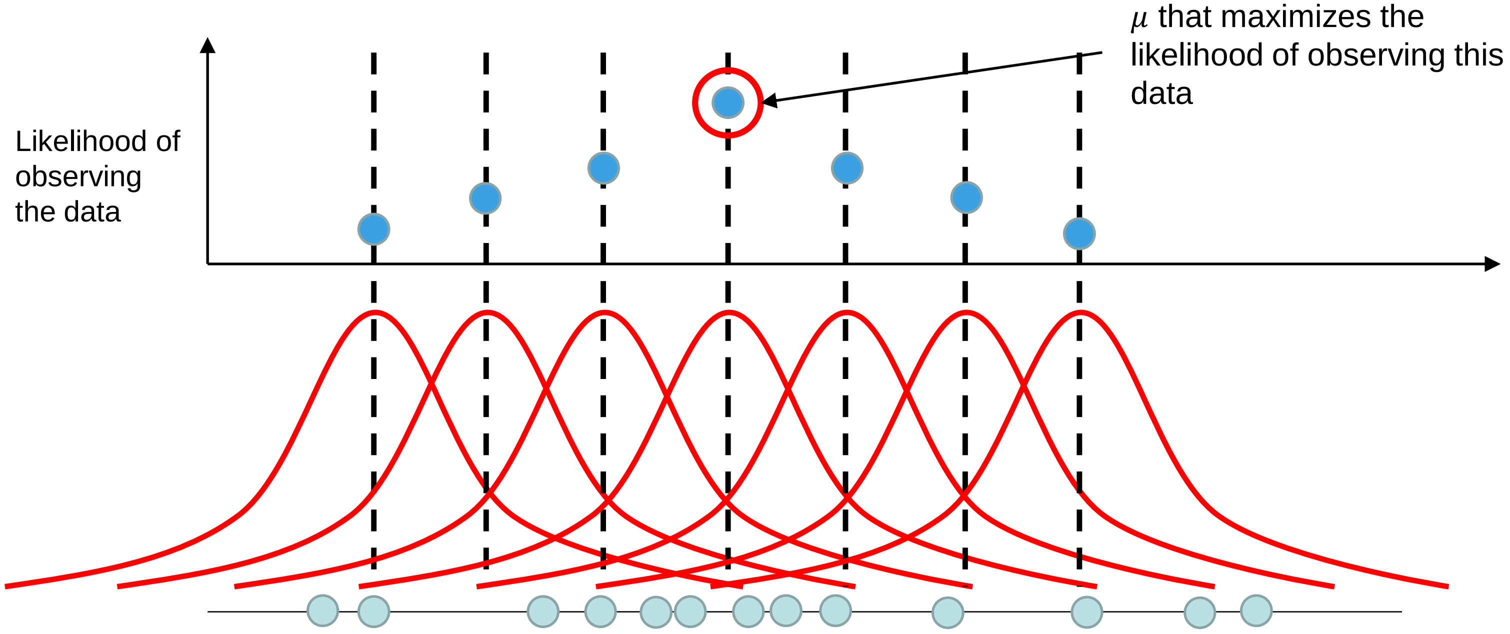
- Tossing a dice with K faces
  - is the same as sampling from a *multinomial distribution* on k elements
  - the parameters of the multinomial are

$$\phi_k \geq 0, \sum_{k=1}^{K} \phi_k = 1, p(z_n = k) = \phi_k$$

- For each k,
  - Assume data points are sampled from Gaussian distribution $N(\mu_k, \Sigma_k)$
  - Mean μ_k and covariance matrix Σ_k
  - Note that we have a collection of these Gaussian distributions,
  - each of which corresponds to one of K dice faces
- We don't know the labels and try to best guess the latent variables $(z_1, \dots, z_n)$ where $z_n \in \{1, \dots, K\}$ represents the latent label for a data point x_n
- $\theta := (\phi, \mu_1, \Sigma_1, \dots, \mu_k, \Sigma_k)$
- Use the maximum likelihood estimation

Maximum Likelihood Estimation

- maximum likelihood estimation (MLE) is a method of estimating the parameters of a statistical model given observations, by finding the parameter values that maximize the likelihood of making the observations



μ that maximizes the likelihood of observing this data
Likelihood of observing the data

- You can find variance similarly

Gaussian Mixture Model

- If we are given a complete data point (k, x)
  - Where the label was not hidden
  - The probability of the pair according to our generative story would be
  - $p(k, x_n) = p(face\ k)p(x_n|face\ k) = \varphi_k N(\mu_k, \Sigma_k)$
- In practice, we are given incomplete data (or observed data)
  - $p(x_n) = \sum_{z_{n\in\{1,\dots,K\}}} p(z_n, x_n) = \sum_{k=1}^{K} p(z_n = k)p(x|face\ k)$

$$= \sum_{k=1}^{K} \varphi_k N(\mu_k, \Sigma_k)$$

  - This model is called the Gaussian Mixture Model

Gaussian Mixture Model

- We are only given $\{x_1, x_2, \ldots, x_N\}$
- The labels are hidden (latent)
- We aim to best guess $(z_1, z_2, \ldots, z_N), z_n \epsilon \{1, \ldots, K\}$
- z_n is the latent label for a data point x_n
- The parameter of this model
  - $\theta = (\phi, \mu_1, \Sigma_1, \mu_2, \Sigma_2, \ldots, \mu_K, \Sigma_K)$
  - We like to best estimate these parameters

Latent variable models

- Use the maximum likelihood principle to do the parameter estimation
- Complete data likelihood function
  - We are given the class label
  - Gaussian classifier
  - $p(X,Z) = \Pi_{n=1}^{N}\Pi_{k=1}^{K}p(x_n, z_k)$
  - Easy to get the analytical global solutions

- Likelihood function (incomplete data likelihood function)
  - $p(X) = \Pi_{n=1}^{N}p(x_n) = \Pi_{n=1}^{N}\Sigma_{k=1}^{K}p(x_n, z_k)$
  - Hard to get the analytical global solutions (sum inside log)
  - Need a iterative optimization algorithm (EM method)
  - EM: iterative optimization algorithm for problems with latent variables

Problem to be solved

- Why is it hard to find the global solution of imcomplete data likelihood functions?
  - use Gaussian mixture model as an example

- What EM algorithm is and why?
  - Steps
  - Theoretical support

Gaussian Mixture Models

- $L(\Theta) = lnp(X) = ln\Pi_{n=1}^{N} p(x_n) = \Sigma_{n=1}^{N} lnp(x_n) = \Sigma_{n=1}^{N} ln\Sigma_{k=1}^{K} p(x_n, z_k)$
  $L(\Theta) = \Sigma_{n=1}^{N} ln\Sigma_{k=1}^{K} p(z_k)p(x_n|z_k) = \Sigma_{n=1}^{N} ln\Sigma_{k=1}^{K} \varphi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)$

- Prediction rule

$$\gamma(z_{nk}) := p(z_n = k|\boldsymbol{x}_n) = \frac{\varphi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \varphi_j \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

  – $\gamma(z_{nk})$: the posterior probability of the cluster k assigned to a given data x_n
  – For a given data, what’s the prior probability of the cluster k assigned to it?

Gaussian Mixture Models

- $L(\Theta) = lnp(X) = ln\Pi_{n=1}^{N}p(x_n) = \Sigma_{n=1}^{N}lnp(x_n) = \Sigma_{n=1}^{N}ln\Sigma_{k=1}^{K}p(x_n, z_k)$
  $L(\Theta) = \Sigma_{n=1}^{N}ln\Sigma_{k=1}^{K}p(z_k)p(x_n|z_k) = \Sigma_{n=1}^{N}ln\Sigma_{k=1}^{K}\varphi_k\mathcal{N}(x_n|\mu_k, \Sigma_k)$

- Prediction rule

$$\gamma(z_{nk}) := p(z_n = k|\boldsymbol{x}_n) = \frac{\varphi_k\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\sum_j \varphi_j\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)}$$

  - $\gamma(z_{nk})$: the posterior probability of the cluster k assigned to a given data x_n
  - For a given data, what's the prior probability of the cluster k assigned to it? (φ_k)

Gaussian Mixture Models

$$L(\Theta) = \Sigma_{n=1}^{N} ln\Sigma_{k=1}^{K} p(z_k)p(x_n|z_k) = \Sigma_{n=1}^{N} ln\Sigma_{k=1}^{K} \varphi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)$$

$$\gamma(z_{nk}) := p(z_n = k|\boldsymbol{x}_n) = \frac{\varphi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \varphi_j \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

- Let’s try to find the global optimal solutions first
  - Three types of parameters: the mean parameters (μ_k); the covariance Matrices (Σ_k); the mixing coefficients (φ_k)
  - Compute the global optimal solutions by setting the partial derivatives with respect to these parameters to 0 respectively.
  - Refer to the handwritten materials
  - Hard to get the global optimal solutions
    - > as all the solutions rely on $\gamma(z_{nk})$: the posterior probability of the cluster assignment; and $\gamma(z_{nk})$ itself relies on the three types of parameters in a complex way
  - Need an iterative algorithm!

Expectation Maximization (EM) for GMMs

Choose some initial values for the parameters;

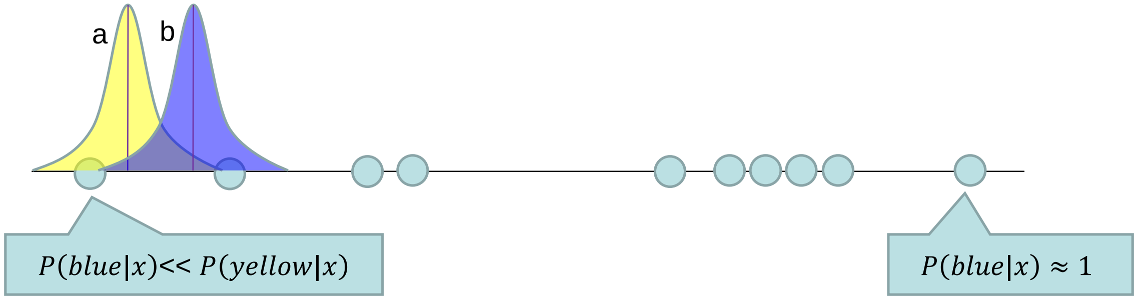
Alternate between the following two steps until a stopping condition (the change in the log likelihood function or parameters fall below some threshold) is met:

- *In the E (expectation) step, use the current values for the parameters to calculate the posterior probabilities* $\gamma(z_{nk})$
- *In the M (maximization) step, re-estimate the parameters (*μ_k*,* Σ_k*, and* φ_k*) based on the* $\gamma(z_{nk})$ *result from the above step.*
  - *Use the equations in the handwritten materials*

Example

$$P(x_i|b) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(x_i - \mu_b)^2}{2\sigma_b^2}\right)$$

$$P(b|x_i) = \frac{P(x_i|b)P(b)}{P(x_i|b)P(b) + P(x_i|a)P(a)}$$



$P(blue|x)$<< $P(yellow|x)$

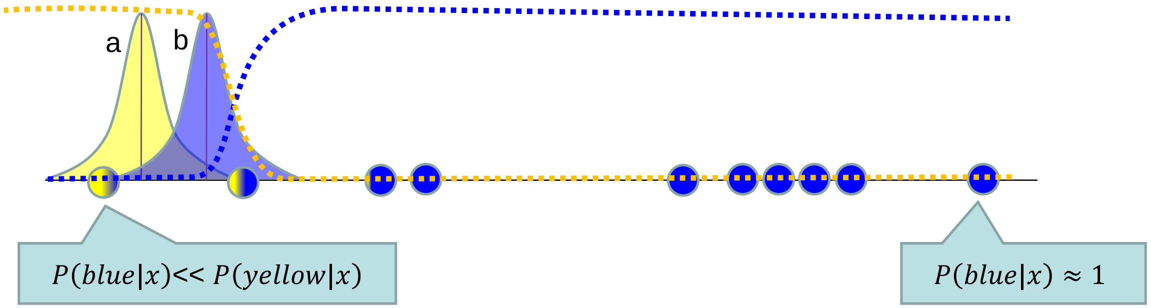
$P(blue|x) \approx 1$

For each point calculate $P(blue|x)$ and $P(yellow|x)$

Example

$$P(x_i|b) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(x_i - \mu_b)^2}{2\sigma_b^2}\right)$$

$$P(b|x_i) = \frac{P(x_i|b)P(b)}{P(x_i|b)P(b) + P(x_i|a)P(a)}$$

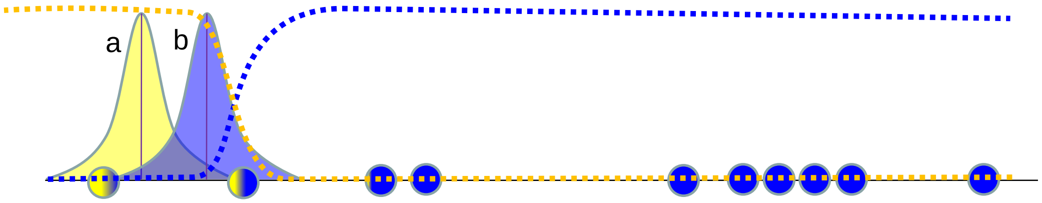


$P(blue|x) << P(yellow|x)$

$P(blue|x) \approx 1$

For each point calculate $P(blue|x)$ and $P(yellow|x)$

Example



Update means and variances

$$P(x_i|b) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(x_i-\mu_b)^2}{2\sigma_b^2}\right)$$

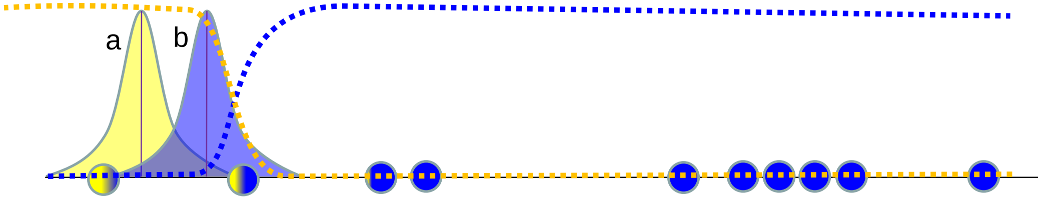
$$b_i = P(b|x_i) = \frac{P(x_i|b)P(b)}{P(x_i|b)P(b) + P(x_i|a)P(a)}$$

$$a_i = P(a|x_i) = 1 - b_i$$

$$\mu_b = \frac{b_1x_1 + b_2x_2 + \cdots + b_nx_n}{b_1 + b_2 + \cdots + b_n}$$

$$\sigma_b^2 = \frac{b_1(x_1-\mu_b)^2 + \cdots + b_n(x_n-\mu_b)^2}{b_1 + b_2 + \cdots + b_n}$$

Example



$$P(x_i|b) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(x_i - \mu_b)^2}{2\sigma_b^2}\right)$$

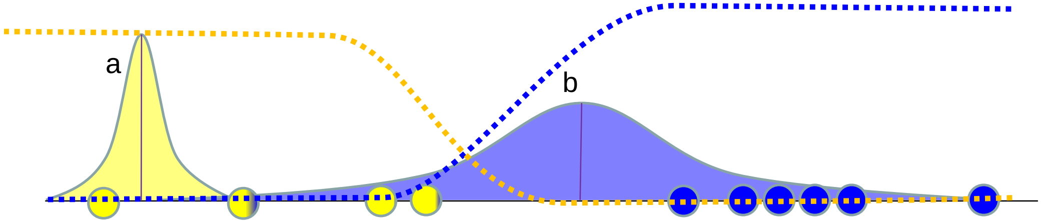
$$b_i = P(b|x_i) = \frac{P(x_i|b)P(b)}{P(x_i|b)P(b) + P(x_i|a)P(a)}$$

$$a_i = P(a|x_i) = 1 - b_i$$

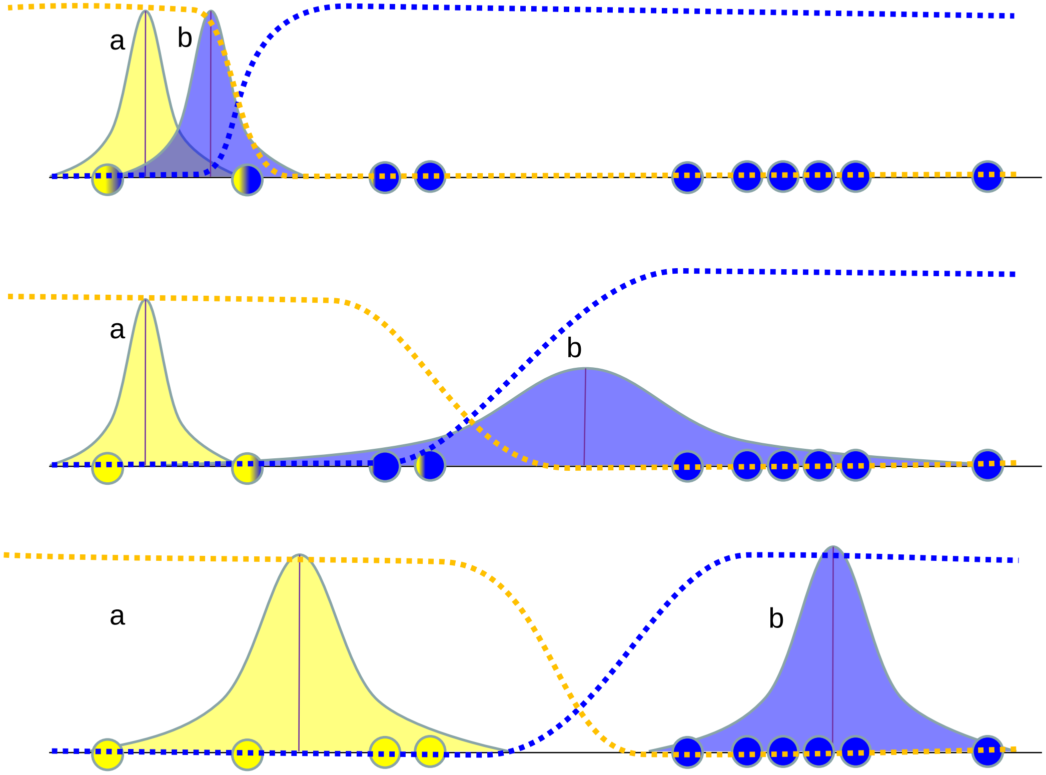
$$\mu_b = \frac{b_1x_1 + b_2x_2 + \cdots + b_nx_n}{b_1 + b_2 + \cdots + b_n}$$

$$\sigma_b^2 = \frac{b_1(x_1 - \mu_b)^2 + \cdots + b_n(x_n - \mu_b)^2}{b_1 + b_2 + \cdots + b_n}$$

Update means and variances



Example



$$P(x_i|b) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(x_i - \mu_b)^2}{2\sigma_b^2}\right)$$

$$b_i = P(b|x_i) = \frac{P(x_i|b)P(b)}{P(x_i|b)P(b) + P(x_i|a)P(a)}$$

$$a_i = P(a|x_i) = 1 - b_i$$

$$\mu_b = \frac{b_1x_1 + b_2x_2 + \cdots + b_nx_n}{b_1 + b_2 + \cdots + b_n}$$

$$\sigma_b^2 = \frac{b_1(x_1 - \mu_b)^2 + \cdots + b_n(x_n - \mu_b)^2}{b_1 + b_2 + \cdots + b_n}$$

The EM Algorithm: General Case

- Training objective: find maximum likelihood solution for models having latent variables.
  - Observed data X, Latent variable Z, set of model parameters θ
  - Log likelihood function

$$\ln p(X|\theta) = \ln \sum_Z p(X,Z|\theta)$$

- Algorithm:
  - Choose an initial setting for the parameters θ^{old}
  - While convergence is not met:
    - > **E Step**: Evaluate $p\left(Z|X,\theta^{old}\right)$
    - > **M Step**: Evaluate θ^{new} given by

$$\theta^{new} \leftarrow \arg\max_{\theta} \underbrace{\sum_Z p\left(Z|X,\theta^{old}\right) \ln p(X,Z|\theta)}_{Q\left(\theta,\theta^{old}\right)}$$

    - > $\theta^{old \leftarrow \theta^{new}}$

MONASH University
Information Technology

The EM Algorithm: General Case

- Questions:
  - Why do we use Q function instead of the log likelihood function as the objective function in M step?
  - Is each iteration guaranteed to increase the log likelihood function?
  - What’s the relationship between the Q function and log likelihood function?

The EM Algorithm: General Case

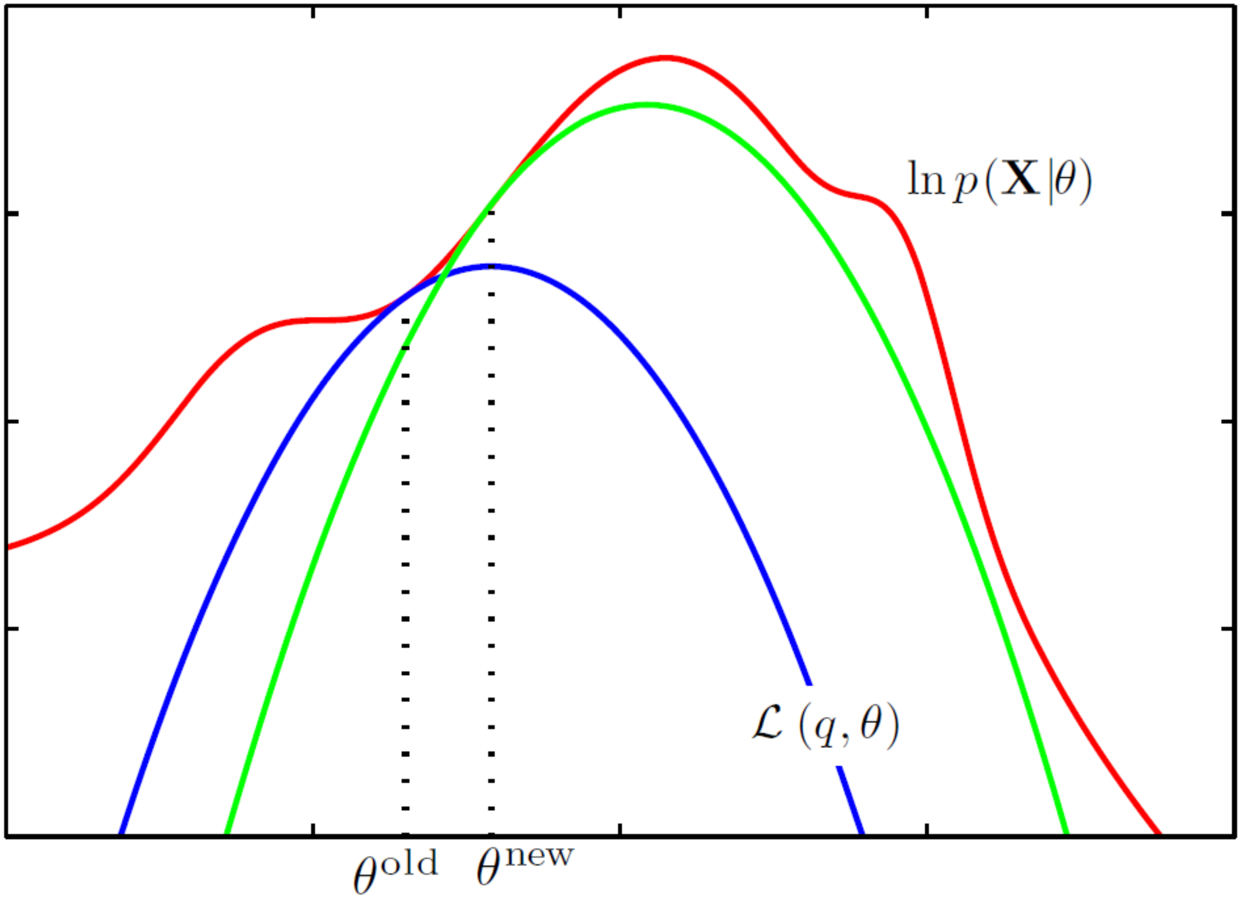
- Why Q function?

$$\sum_Z p\left(Z|X,\theta^{old}\right) \ln p(X,Z|\theta)$$

  - Solving $\ln p(X,Z|\theta)$ (complete data likelihood) is easy while solving $\ln p(X|\theta)$ (incomplete data likelihood) is hard
  - Focus on the complete data likelihood
  - Intuitive explanation: the expected value of complete data likelihood function under the posterior distribution of the latent variable ($p(Z|X,\theta^{old})$)

The EM Algorithm: General Case

- Is each iteration guaranteed to increase the log likelihood function?
- What’s the relationship between the Q function and log likelihood function?



The EM Algorithm: General Case

- Is each iteration guaranteed to increase the log likelihood function?
  - Yes, for the proof, refer to the text book (Bishop: Pattern Recognition and Machine Learning)
- What's the relationship between the Q function and log likelihood function?
  - Q function is a lower bound of the log likelihood function

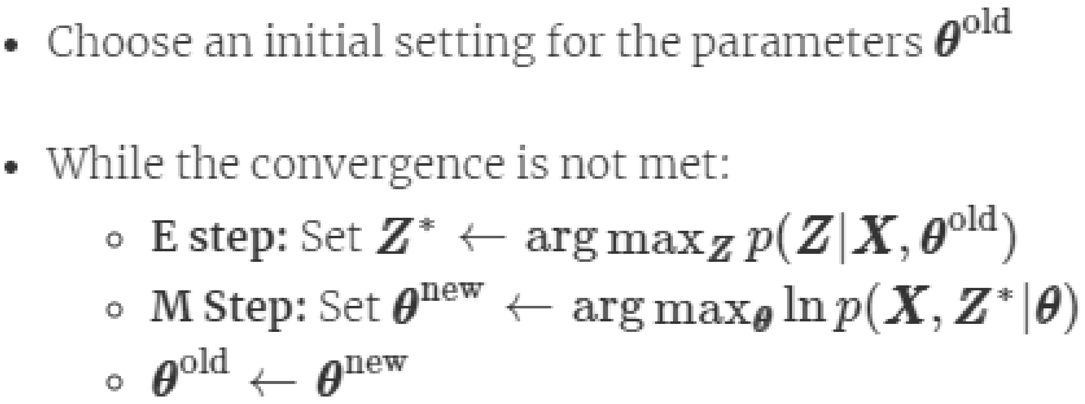
The hard-EM Algorithm

- Each data is assigned to one class with the largest posterior probability

$$Z^* = argmax_z p(Z|X, \theta^{old})$$

- There is no expectation over the latent variables Z in Q function

$$\ln p(X, Z^*|\theta)$$



- Choose an initial setting for the parameters $\boldsymbol{\theta}^{\text{old}}$
- While the convergence is not met:
  - **E step:** Set $\boldsymbol{Z}^* \leftarrow \arg\max_{\boldsymbol{Z}} p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{\text{old}})$
  - **M Step:** Set $\boldsymbol{\theta}^{\text{new}} \leftarrow \arg\max_{\boldsymbol{\theta}} \ln p(\boldsymbol{X}, \boldsymbol{Z}^*|\boldsymbol{\theta})$
  - $\boldsymbol{\theta}^{\text{old}} \leftarrow \boldsymbol{\theta}^{\text{new}}$

EM Algorithm for GMMs with Q function

$$Q\left(\theta^{new},\theta^{old}\right) \coloneqq \Sigma_Z p(Z|X,\theta^{old}) \ln p(X,Z|\theta^{new})$$
$$= \sum_{n=1}^{N} \sum_{k=1}^{K} (\gamma(z_{nk}) \ln \varphi_k + \gamma(z_{nk}) \ln \mathcal{N}(x_n|\mu_k,\Sigma_k))$$

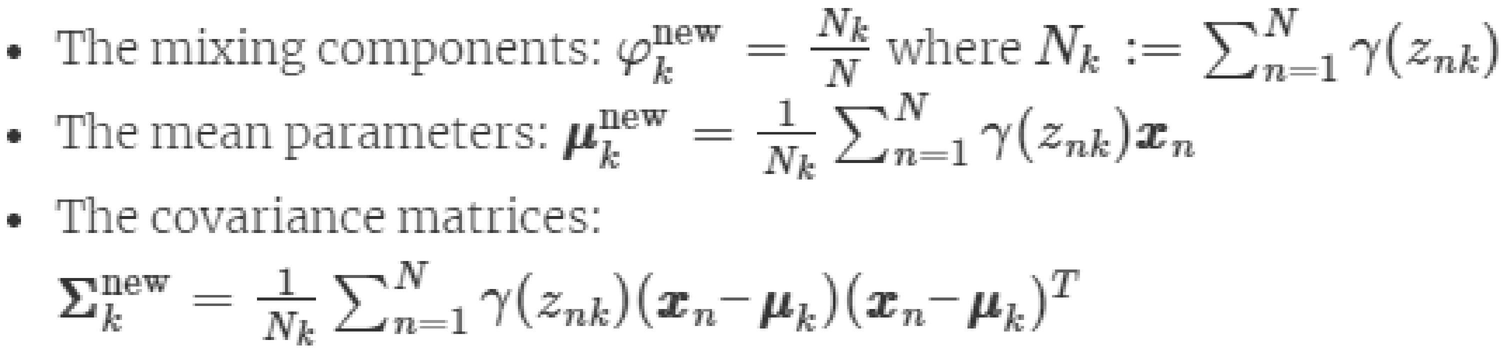
- $\gamma(z_{nk})$ is given in E step
- No sum inside log
- Easy to optimize

EM Algorithm for GMMs with Q function

$$Q\left(\theta^{new}, \theta^{old}\right) := \Sigma_Z p(Z|X, \theta^{old}) \ln p(X, Z|\theta^{new})$$

$$= \sum_{n=1}^{N} \sum_{k=1}^{K} (\gamma(z_{nk}) \ln \varphi_k + \gamma(z_{nk}) \ln \mathcal{N}(x_n|\mu_k, \Sigma_k))$$

- Maximizing the Q function, we get:



- The mixing components: $\varphi_k^{\text{new}} = \frac{N_k}{N}$ where $N_k := \sum_{n=1}^{N} \gamma(z_{nk})$
- The mean parameters: $\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \boldsymbol{x}_n$
- The covariance matrices:

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