Lecture 12: Gaussian Mixture Models

Shuai Li

John Hopcroft Center, Shanghai Jiao Tong University

https://shuaili8.github.io

https://shuaili8.github.io/Teaching/VE445/index.html

Outline

- Generative models
- GMM
- EM

Generative Models (review)

Discriminative / Generative Models

- Discriminative models
  - Modeling the dependence of unobserved variables on observed ones
  - also called conditional models
  - Deterministic: $y = f_\theta(x)$
  - Probabilistic: $p_\theta(y|x)$
- Generative models
  - Modeling the joint probabilistic distribution of data
  - Given some hidden parameters or variables

$$p_\theta(x, y)$$

  - Then do the conditional inference

$$p_\theta(y|x) = \frac{p_\theta(x, y)}{p_\theta(x)} = \frac{p_\theta(x, y)}{\sum_{y'} p_\theta(x, y')}$$

$$y = f_\theta(x)$$

$$p_\theta(y|x)$$

Generative Models

- Generative models
  - Modeling the joint probabilistic distribution of data
  - Given some hidden parameters or variables

$$p_\theta(x, y)$$

  - Then do the conditional inference

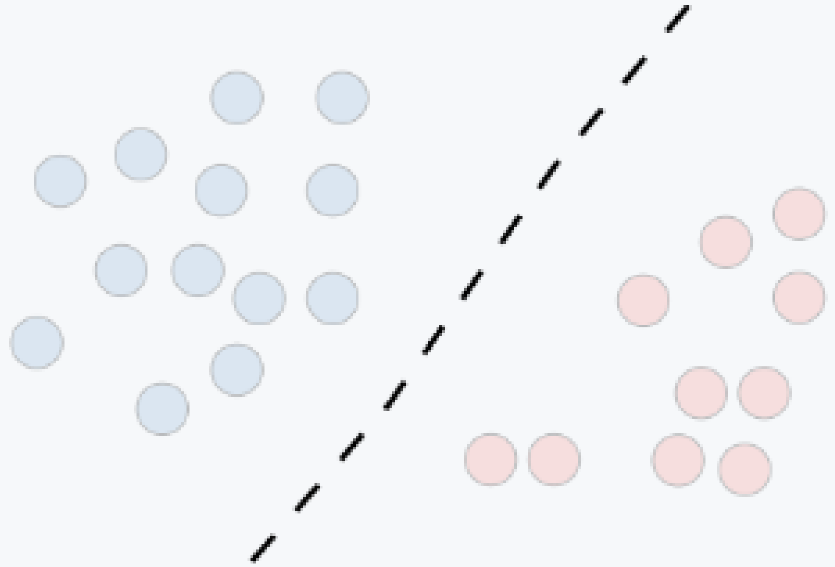
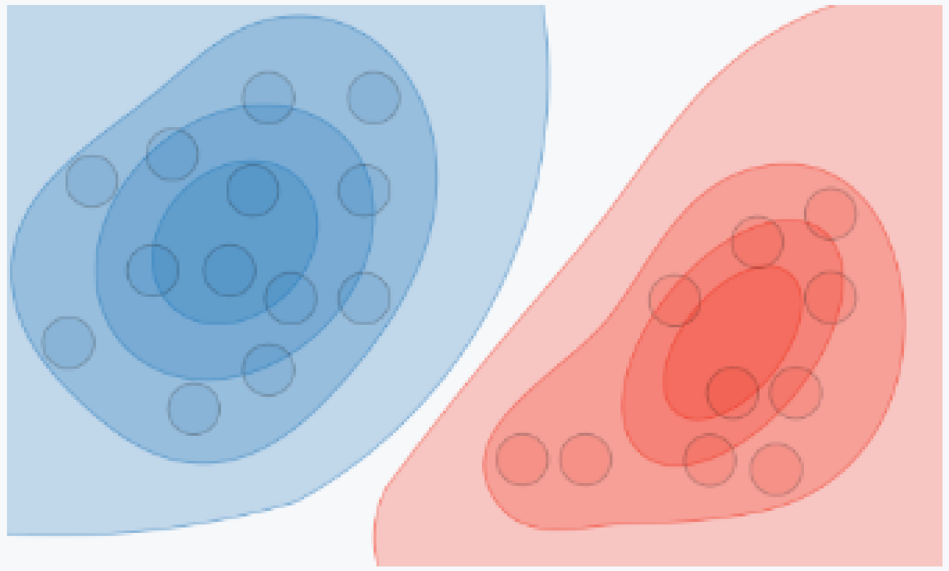
$$p_\theta(y|x) = \frac{p_\theta(x, y)}{p_\theta(x)} = \frac{p_\theta(x, y)}{\sum_{y'} p_\theta(x, y')}$$

- Recover the data distribution [essence of data science]
- Benefit from hidden variables modeling
- E.g. Naive Bayes, Hidden Markov Model, Mixture Gaussian, Markov Random Fields, Latent Dirichlet Allocation

Discriminative Models vs Generative Models

- In General
  - A Discriminative model models the **decision boundary between the classes**
  - A Generative Model explicitly models the **actual distribution of each class**

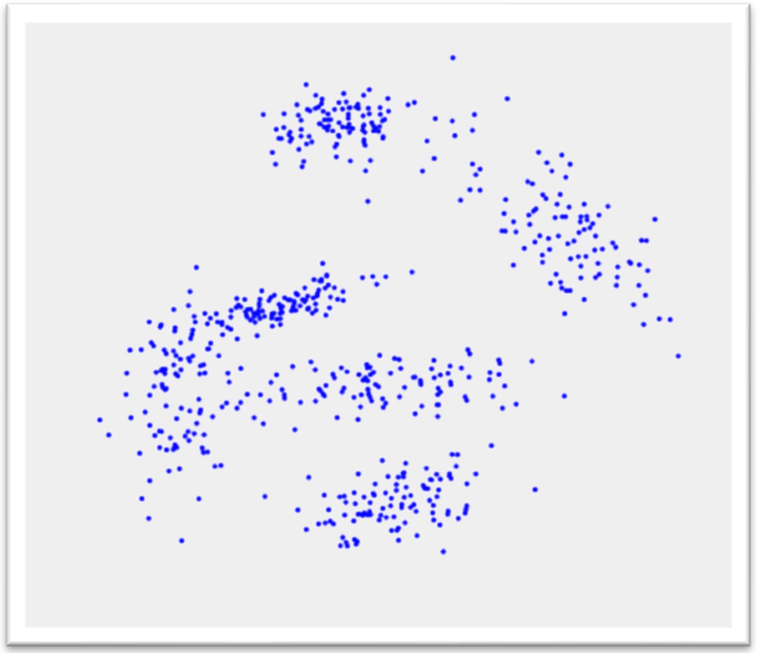
- Example: Our training set is a bag of fruits. Only apples and oranges Each labeled. Imagine a post-it note stuck to the fruit
  - A generative model will model various attributes of fruits such as color, weight, shape, etc
  - A discriminative model might model color alone, **should that suffice** to distinguish apples from oranges

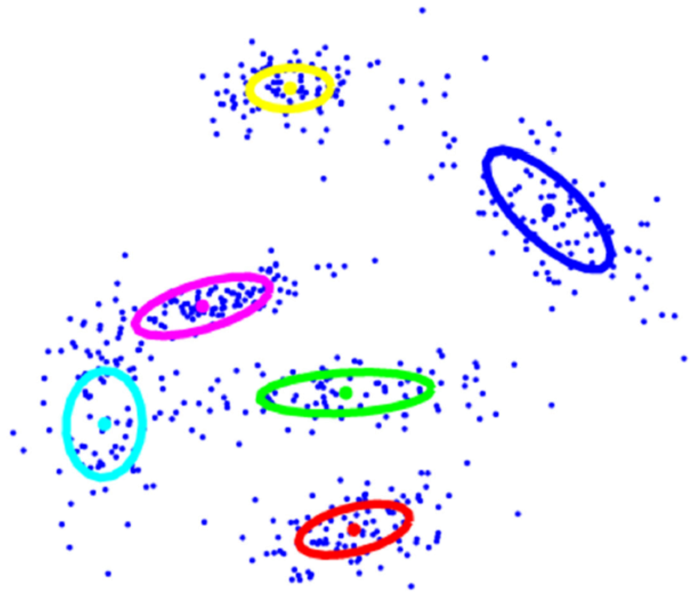
	Discriminative model	**Generative model**
Goal	Directly estimate $P(y\|x)$	Estimate $P(x\|y)$ to then deduce $P(y\|x)$
What's learned	Decision boundary	Probability distributions of the data
Illustration		
Examples	Regressions, SVMs	GDA, Naive Bayes

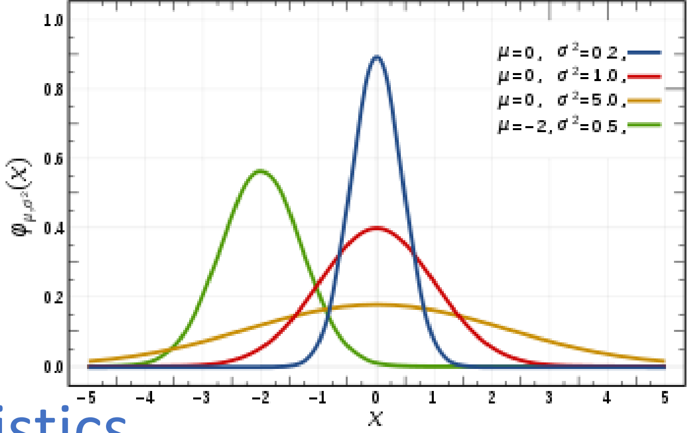
Gaussian Mixture Models

Gaussian Mixture Models

- Is a clustering algorithms
- Difference with K-means
  - K-means outputs the label of a sample
  - GMM outputs the probability that a sample belongs to a certain class
  - GMM can also be used to generate new samples!







Gaussian distribution

- Very common in probability theory and important in statistics
- often used in the natural and social sciences to represent real-valued random variables whose distributions are not known
- is useful because of the central limit theorem
  - averages of samples independently drawn from the same distribution converge in distribution to the normal with the true mean and variance, that is, they become normally distributed when the number of observations is sufficiently large
- Physical quantities that are expected to be the sum of many independent processes often have distributions that are nearly normal
- The probability density of the Gaussian distribution is

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

High-dimensional Gaussian distribution

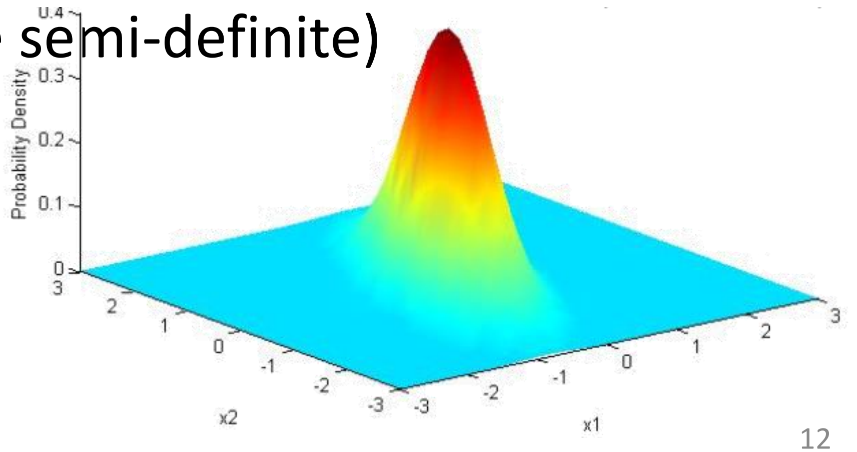
$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- The probability density of Gaussian distribution on $x = (x_1, \ldots, x_d)^\top$ is

$$\mathcal{N}(x|\mu,\Sigma) = \frac{\exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right)}{\sqrt{(2\pi)^d |\Sigma|}}$$

  - where μ is the mean vector
  - Σ is the symmetric covariance matrix (positive semi-definite)
- E.g. the Gaussian distribution with

$$\mu = (0,0)^T \qquad \Sigma = \begin{pmatrix} 0.25 & 0.30 \\ 0.30 & 1.00 \end{pmatrix}$$



Mixture of Gaussian

- The probability given in a mixture of K Gaussians is:

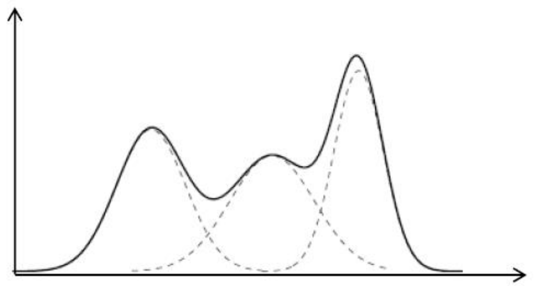
$$p(x) = \sum_{j=1}^{K} w_j \cdot \mathcal{N}(x|\mu_j, \Sigma_j)$$

where w_j is the prior probability of the j-th Gaussian

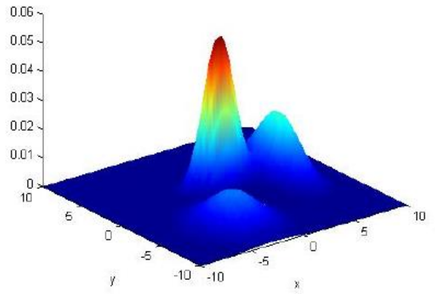
$$\sum_{j=1}^{K} w_j = 1 \quad \text{and} \quad 0 \le w_j \le 1$$

- Example

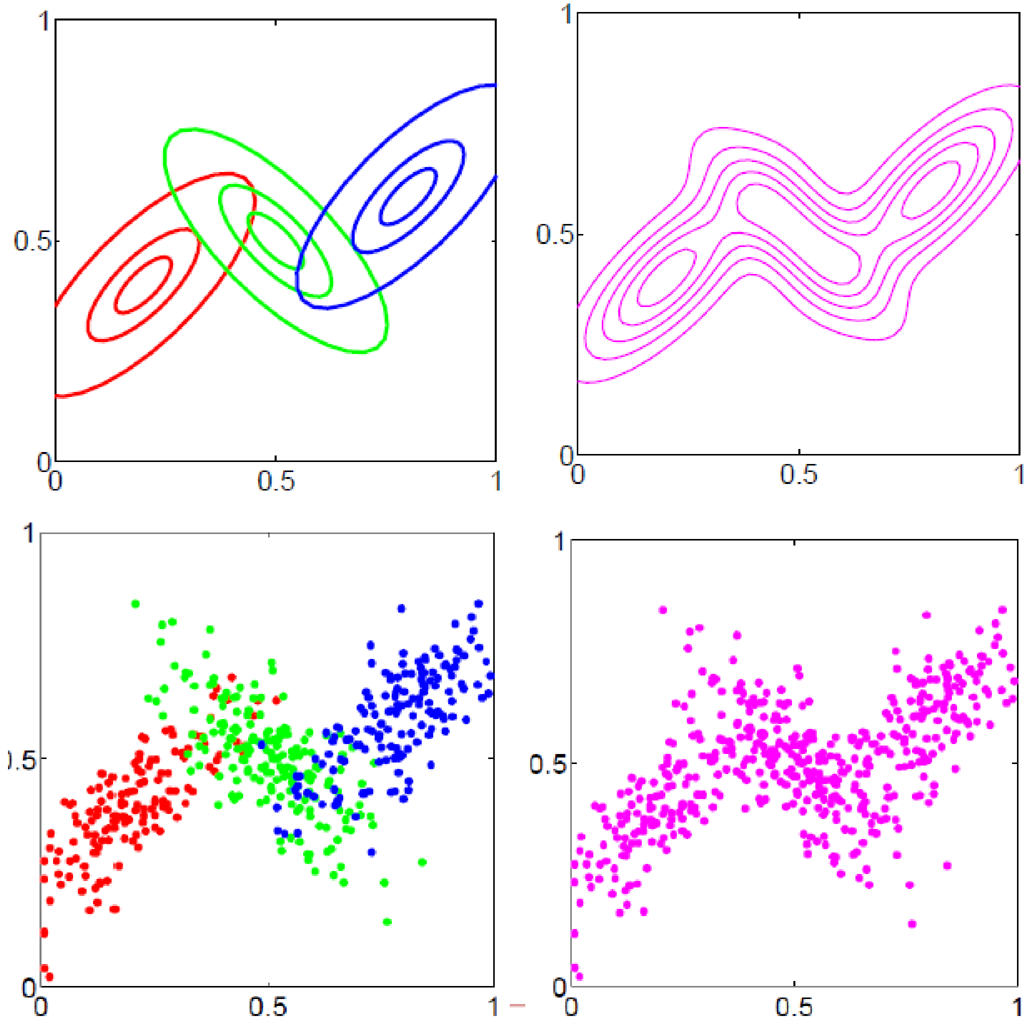
d=1:

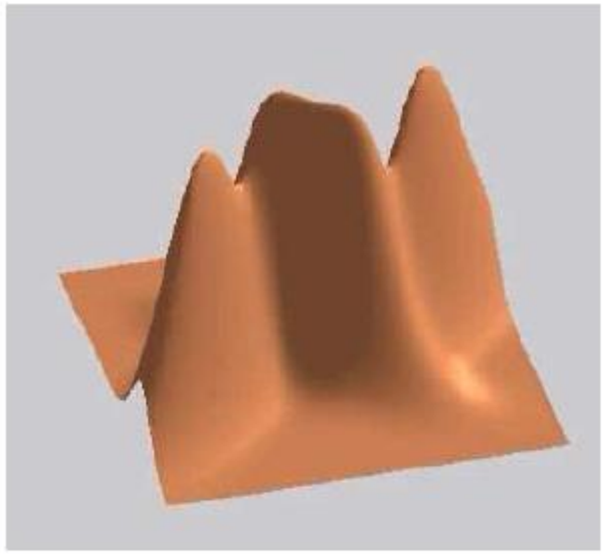


d=2:



Examples





Data generation

- Let the parameter set $\theta = \{w_j, \mu_j, \Sigma_j : j\}$, then the probability density of mixture Gaussian can be written as

$$p(x|\theta) = \sum_{j=1}^{K} w_j \cdot \mathcal{N}(x|\mu_j, \Sigma_j)$$

- Equivalent to generate data points in two steps
  - Select which component j the data point belongs to according to the categorical (or multinoulli) distribution of $(w_1, \dots, w_K)$
  - Generate the data point according to the PDF of j-th component

Learning task

- Given a dataset $X = \{x^{(1)}, x^{(2)}, \cdots, x^{(N)}\}$ to train the GMM model

- Find the best θ that maximizes the probability $p(X|\theta)$
- Maximal likelihood estimator (MLE)

$$\boxed{\theta^* = \arg\max_{\theta} p(X \mid \theta) = \arg\max_{\theta} \prod_{i=1}^{N} p(x_i \mid \theta)}$$

Maximal likelihood

- We want to solve

$$\operatorname{argmax} p(X|\theta) = \operatorname{argmax} \prod_{i=1}^{N} \mathbb{P}\left(x^{(i)} \middle| \theta\right)$$

$$= \operatorname{argmax} \sum_{i=1}^{N} \log \sum_{j=1}^{K} \mathbb{P}\left(z^{(i)} = j\right) \cdot \mathbb{P}\left(x^{(i)} \middle| \theta, j\right)$$

$$= \operatorname{argmax} \sum_{i=1}^{N} \log \sum_{j=1}^{K} w_j \cdot \mathcal{N}(x^{(i)} | \mu_j, \Sigma_j)$$

- No closed solution by solving

$$\frac{\partial p(X|\theta)}{\partial w} = 0, \qquad \frac{\partial p(X|\theta)}{\partial \mu} = 0, \qquad \frac{\partial p(X|\theta)}{\partial \Sigma} = 0$$

Introduce latent variable

- For data points $x^{(i)}, i = 1, \dots, N$, let's write the probability as
$$\mathbb{P}\left(x^{(i)} \middle| \theta\right) = \sum_{j=1}^{K} w_j \cdot \mathcal{N}(x^{(i)} | \mu_j, \Sigma_j)$$
where $\sum_{j=1}^{K} w_j = 1$
- Introduce latent variable
  - $z^{(i)}$ is the Gaussian cluster ID indicates which Gaussian $x^{(i)}$ comes from
  - $\mathbb{P}\left(z^{(i)} = j\right) = w_j$
  - $\mathbb{P}\left(x^{(i)} \middle| z^{(i)} = j; \theta\right) = \mathcal{N}(x^{(i)} | \mu_j, \Sigma_j)$
  - $\mathbb{P}\left(x^{(i)} \middle| \theta\right) = \sum_{j=1}^{K} \mathbb{P}\left(z^{(i)} = j\right) \cdot \mathbb{P}\left(x^{(i)} \middle| z^{(i)} = j; \theta\right)$

Likelihood maximization

- If we know $z^{(i)}$ for all i, the problem becomes

$$\operatorname{argmax} l(X|\theta) = \operatorname{argmax} \prod_{i=1}^{N} \mathbb{P}\left(x^{(i)}|\theta\right)$$

$$= \operatorname{argmax} \sum_{i=1}^{N} \log \mathbb{P}\left(x^{(i)}|\theta\right) = \operatorname{argmax} \sum_{i=1}^{N} \log \mathbb{P}\left(x^{(i)}, z^{(i)}|\theta\right)$$

$$= \operatorname{argmax} \sum_{i=1}^{N} \log \mathbb{P}\left(x^{(i)}|\theta, z^{(i)}\right) + \log \mathbb{P}\left(z^{(i)}|\theta\right)$$

$$= \operatorname{argmax} \sum_{i=1}^{N} \log \mathcal{N}\left(x^{(i)}|\mu_{z^{(i)}}, \Sigma_{z^{(i)}}\right) + \log w_{z^{(i)}}$$

The solution is

- $w_j = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{z^{(i)} = j\}$
- $\mu_j = \frac{\sum_{i=1}^{N} \mathbf{1}\{z^{(i)}=j\}x^{(i)}}{\sum_{i=1}^{N} \mathbf{1}\{z^{(i)}=j\}}$
- $\Sigma_j = \frac{\sum_{i=1}^{N} \mathbf{1}\{z^{(i)}=j\}\left(x^{(i)}-\mu_j\right)\left(x^{(i)}-\mu_j\right)^{\mathsf{T}}}{\sum_{i=1}^{N} \mathbf{1}\{z^{(i)}=j\}}$

Average over each cluster

Likelihood maximization (cont.)

- Given the parameter $\theta = \{w_j, \mu_j, \Sigma_j : j\}$, the posterior distribution of each latent variable $z^{(i)}$ can be inferred

- $\mathbb{P}\left(z^{(i)} = j \middle| x^{(i)}; \theta\right) = \dfrac{\mathbb{P}\left(x^{(i)}, z^{(i)} = j \middle| \theta\right)}{\mathbb{P}\left(x^{(i)} \middle| \theta\right)}$
- $= \dfrac{\mathbb{P}\left(x^{(i)} \middle| z^{(i)} = j; \mu_j, \Sigma_j\right)\mathbb{P}\left(z^{(i)} = j \middle| w\right)}{\sum_{j'=1}^{K} \mathbb{P}\left(x^{(i)} \middle| z^{(i)} = j'; \mu_{j'}, \Sigma_{j'}\right)\mathbb{P}\left(z^{(i)} = j' \middle| w\right)}$

Expectation maximization methods

- E-step:
  - Infer the posterior distribution of the latent variables given the model parameters
- M-step:
  - Tune parameters to maximize the data likelihood given the latent variable distribution
- EM methods
  - Iteratively execute E-step and M-step until convergence

EM for GMM

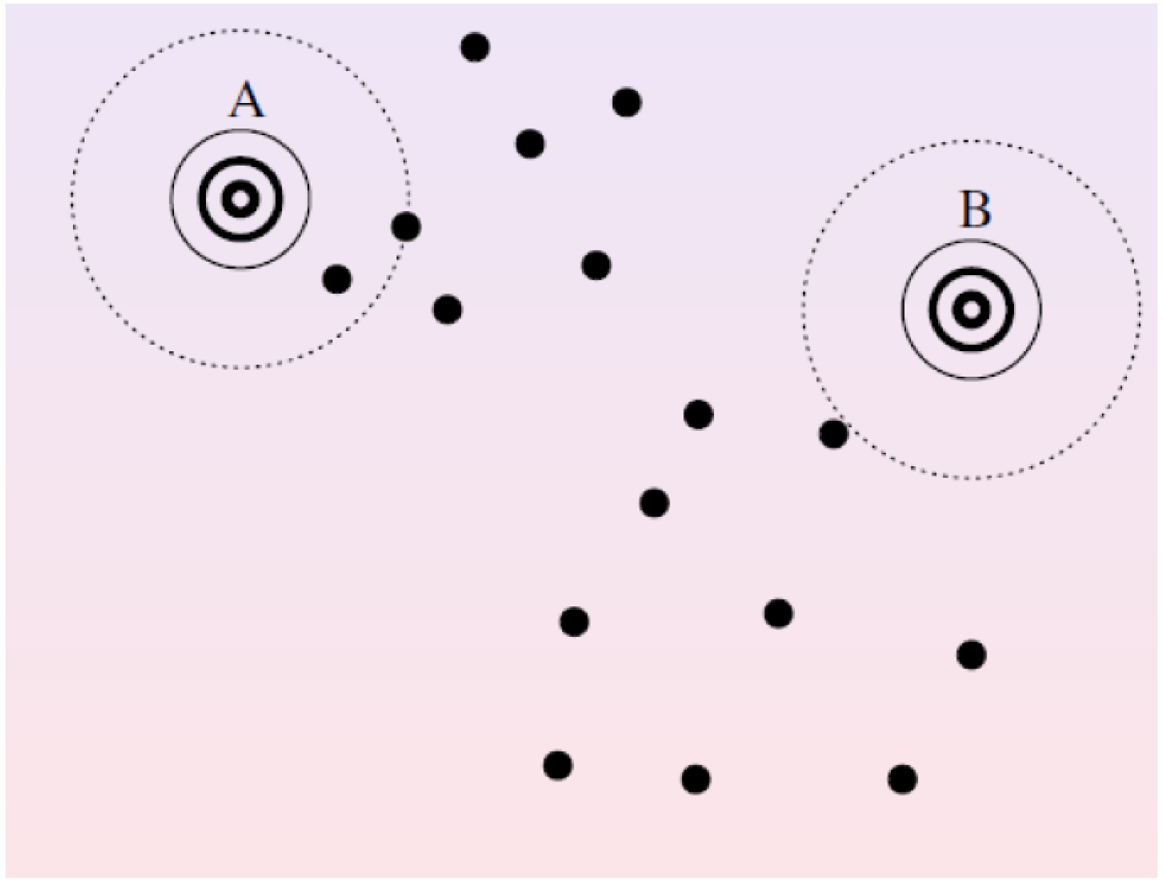
- Repeat until convergence:{
  (E-step) For each i, j, set
  $$w_j^{(i)} = \mathbb{P}\left(z^{(i)} = j \middle| x^{(i)}, w, \mu, \Sigma\right)$$
  (M-step) Update the parameters
  $$w_j = \frac{1}{N}\sum_{i=1}^{N} w_j^{(i)}, \qquad \mu_j = \frac{\sum_{i=1}^{N} w_j^{(i)} x^{(i)}}{\sum_{i=1}^{N} w_j^{(i)}}$$
  $$\Sigma_j = \frac{\sum_{i=1}^{N} w_j^{(i)} \left(x^{(i)} - \mu_j\right)\left(x^{(i)} - \mu_j\right)^\top}{\sum_{i=1}^{N} w_j^{(i)}}$$

  }

Example

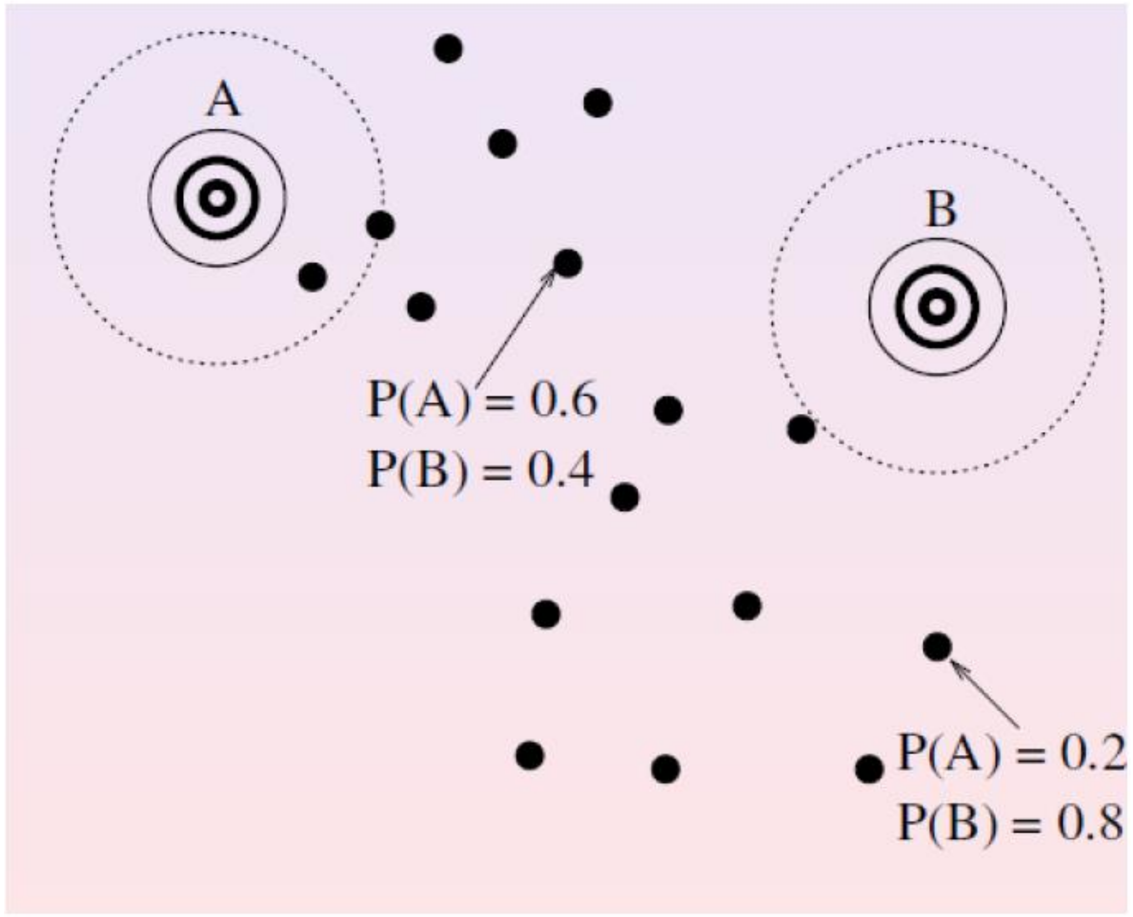
$$p(x|\theta) = \sum_{j=1}^{K} w_j \cdot \mathcal{N}(x|\mu_j, \Sigma_j)$$

- Hidden variable: for each point, which Gaussian generates it?



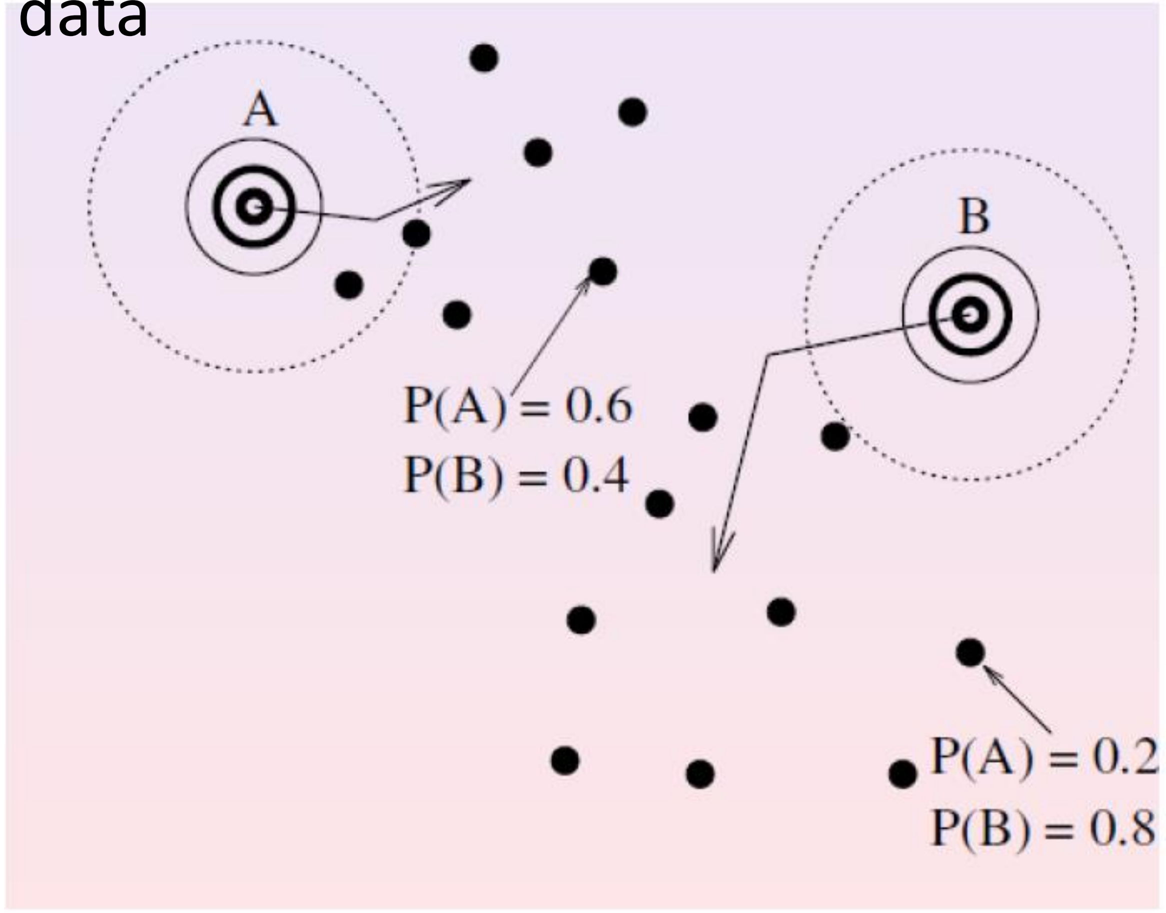
Example (cont.)

- E-step: for each point, estimate the probability that each Gaussian component generated it



GMM example

- M-Step: modify the parameters according to the hidden variable to maximize the likelihood of the data



Remaining issues

- Initialization:
  - GMM is a kind of EM algorithm which is very sensitive to initial conditions

- Number of Gaussians:
  - Use some information-theoretic criteria to obtain the optima *K*
  - Minimal description length (MDL)

Ex: $MDL = -\log(L(X, Z, \theta)) + \left\{(K-1) + K\left[D + \frac{1}{2}D(D+1)\right]\right\}\log(N)$

Other criteria : $AIC, BIC, MML, etc.$

Minimal description length (MDL)

- All statistical learning is about finding regularities in data, and the best hypothesis to describe the regularities in data is also the one that is able to compress the data most
- In an MDL framework, a good model is one that allows an efficient (i.e. short) encoding of the data, but whose *own* description is *also* efficient
- For the GMM

(negative) log-likelihood
The cost of coding data

The cost of coding GMM itself

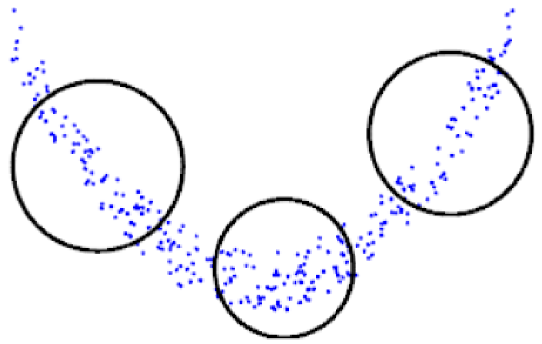
$$L(X) = -\underbrace{\sum_{x \in X} \log p(x)}_{\text{(negative) log-likelihood / The cost of coding data}} + \underbrace{\frac{1}{2} P \log |X|}_{\text{The cost of coding GMM itself}}$$

  - X is the vector of data elements
  - $p(x)$ is the probability by GMM
  - P is the number of free parameters needed to describe the GMM
    - $P = K[d + d(d+1)/2] + (K - 1)$

Remaining issues (cont.)

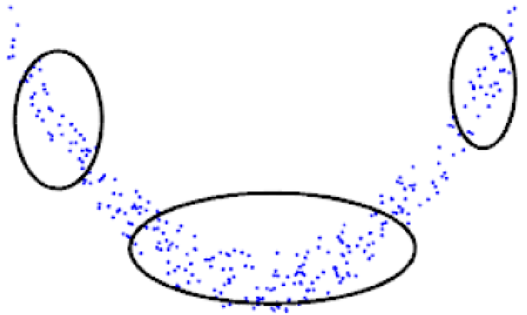
- Simplification of the covariance matrices

Case 1: Spherical covariance matrix $\Sigma_j = diag(\sigma_j^2, \sigma_j^2, ..., \sigma_j^2) = \sigma_j^2 I$



-Less precise.
-Very efficient to compute.

Case 2: Diagonal covariance matrix $\Sigma_j = diag(\sigma_{j1}^2, \sigma_{j2}^2, ..., \sigma_{jd}^2)$

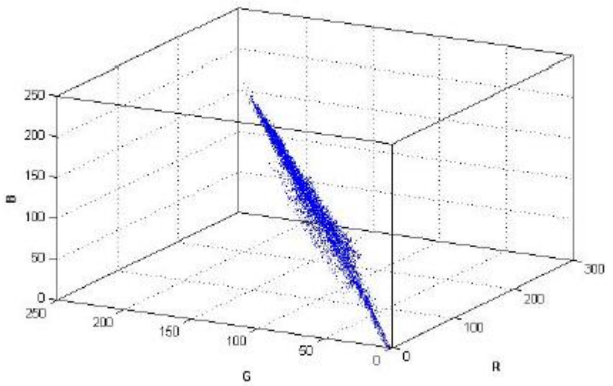


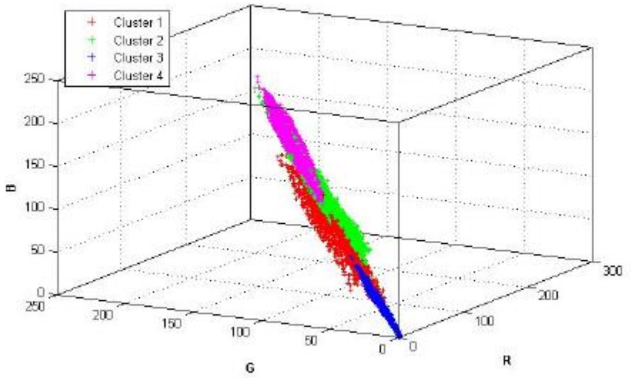
-More precise.
-Efficient to compute.

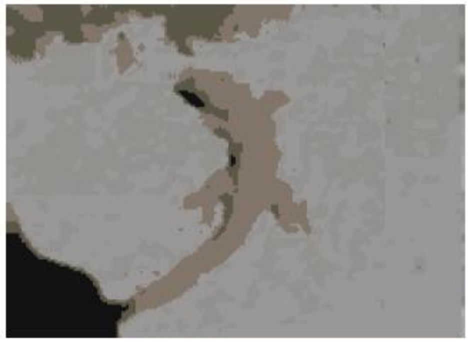
Application in computer vision

- Image segmentation

$$X = (R, G, B)^T$$

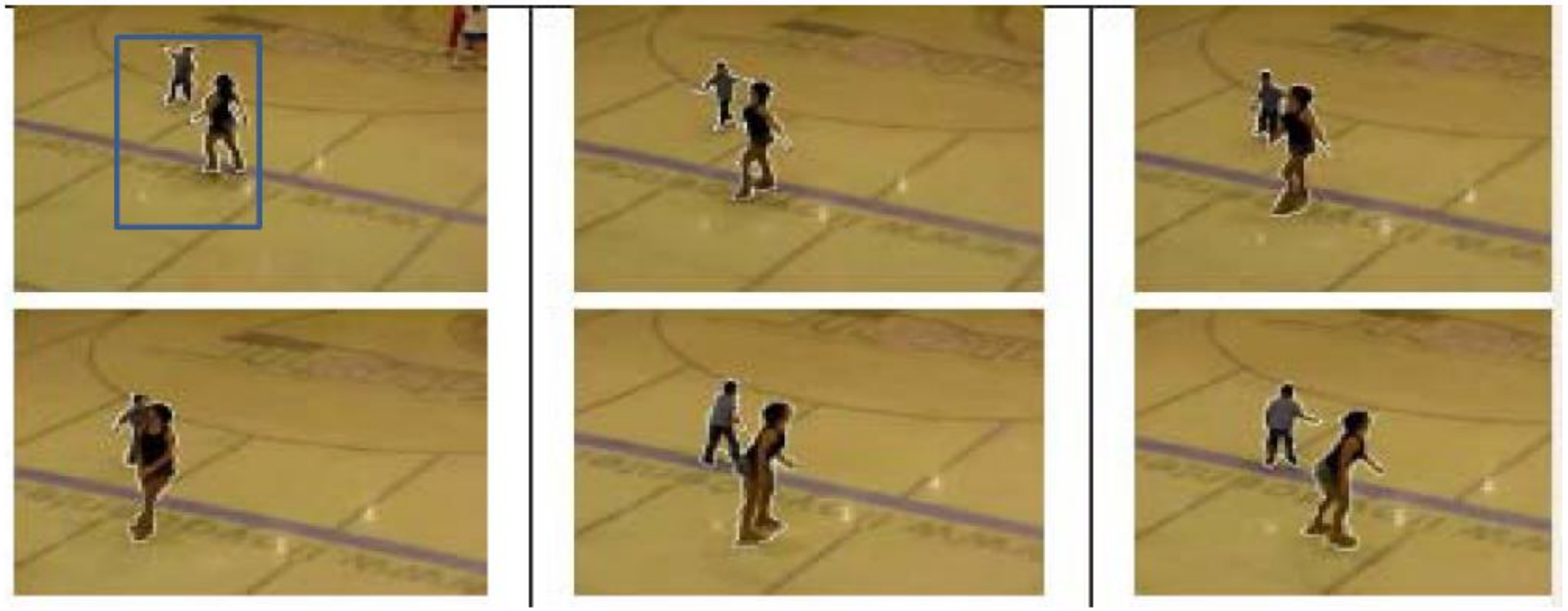






Application in computer vision (cont.)

- Object tracking: Knowing the moving object distribution in the first frame, we can localize the objet in the next frames by tracking its distribution

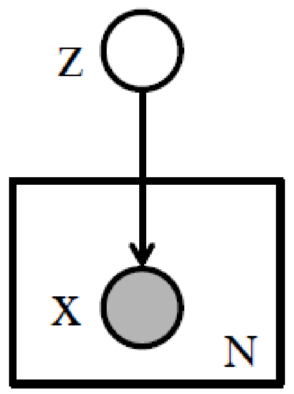


Expectation and Maximization

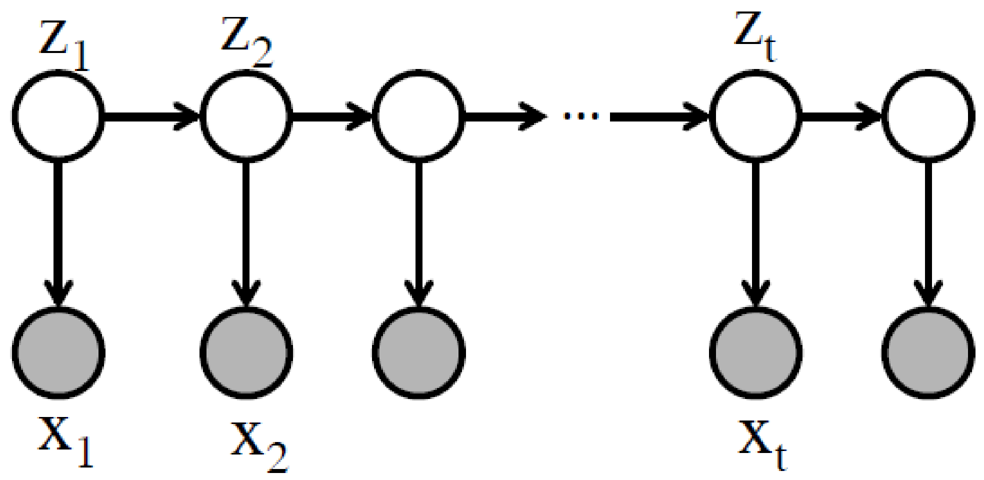
Background: Latent variable models

- Some of the variables in the model are not observed
- Examples: mixture model, Hidden Markov Models, LDA, etc

Mixture Model



Hidden Markov Model



Background: Marginal likelihood

- Joint model $\mathbb{P}(x, z|\theta)$, θ is the model parameter
- With z unobserved, we marginalize out z and use the marginal log-likelihood for learning

$$\log \mathbb{P}(x|\theta) = \log \sum_{z} \mathbb{P}(x, z|\theta)$$

- Example: mixture model

$$\log \mathbb{P}(x|\theta) = \log \sum_{k} \mathbb{P}(x|z = k, \theta_k)\mathbb{P}(z = k|\theta_k)$$

$$= \log \sum_{k} \pi_k \mathbb{P}(x|z = k, \theta_k)$$

where π_k is the mixing proportions

Example of marginal likelihood

- Mixture of Bernoulli

$$p(\mathbf{x}|\mathbf{z}=k,\theta_k) = p(\mathbf{x}|\mu_k) = \prod_i \mu_k^{x_i}(1-\mu_k)^{1-x_i}$$

- Mixture of Gaussians

$$\begin{aligned} p(\mathbf{x}|\mathbf{z}=k,\theta_k) &= p(\mathbf{x}|\mu_k,\Sigma_k) \\ &= \frac{1}{|2\pi\Sigma_k|^{\frac{D}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_k)^\top \Sigma_k^{-1}(\mathbf{x}-\mu_k)\right) \end{aligned}$$

- Hidden Markov Model

$$p(\mathbf{Z}) = p(\mathbf{z}_1)\prod_t p(\mathbf{z}_t|\mathbf{z}_{t-1})$$

$$p(\mathbf{X}|\mathbf{Z}) = \prod_t p(\mathbf{x}_t|\mathbf{z}_t)$$

Learning the hidden variable Z

- If all z observed, the likelihood factorizes, and learning is relatively easy

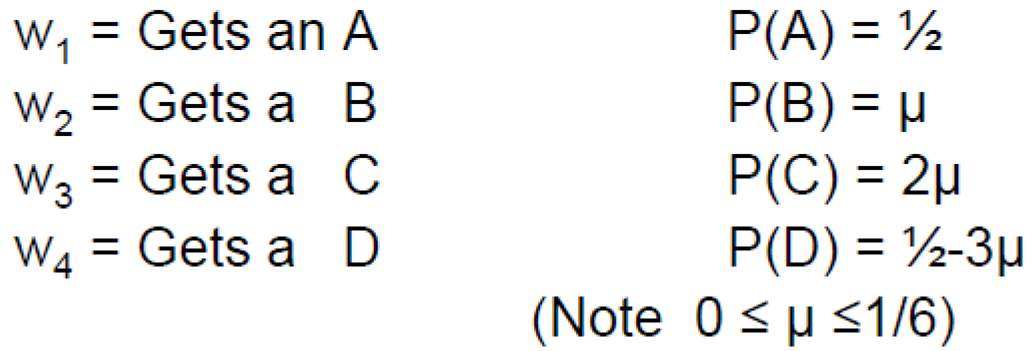
$$\ell(\mathbf{x}, \mathbf{z}|\theta) = \log p(\mathbf{x}, \mathbf{z}|\theta) = \log p(\mathbf{z}|\theta) + \log p(\mathbf{x}|\mathbf{z}, \theta)$$

- If z not observed, we have to handle a sum inside log
  - Idea 1: ignore this problem and simply take derivative and follow the gradient
  - Idea 2: use the current θ to estimate z, fill them in and do fully-observed learning

- Is there a better way?

Example

- Let events be "grades in a class"



w_1 = Gets an A	P(A) = ½
w_2 = Gets a B	P(B) = μ
w_3 = Gets a C	P(C) = 2μ
w_4 = Gets a D	P(D) = ½-3μ

(Note $0 \leq \mu \leq 1/6$)

- Assume we want to estimate μ from data. In a class, suppose there are
  - a students get A, b students get B, c students get C and d students get D.

- What's the maximum likelihood estimate of μ given a, b, c, d ?

Example (cont.)

P(A) = ½ P(B) = μ P(C) = 2μ P(D) = ½-3μ

$P(a,b,c,d \mid \mu) = K(\tfrac{1}{2})^a(\mu)^b(2\mu)^c(\tfrac{1}{2}-3\mu)^d$

$\log P(a,b,c,d \mid \mu) = \log K + a\log \tfrac{1}{2} + b\log \mu + c\log 2\mu + d\log (\tfrac{1}{2}-3\mu)$

- for MLE μ, set $\frac{\partial \mathrm{LogP}}{\partial \mu} = 0$

$$\frac{\partial \mathrm{LogP}}{\partial \mu} = \frac{b}{\mu} + \frac{2c}{2\mu} - \frac{3d}{1/2 - 3\mu} = 0$$

- Solve $\mu = \frac{b+c}{6(b+c+d)}$

- So if class got

A	B	C	D
14	6	9	10

max like $\mu = \frac{1}{10}$

Example (cont.)

- What if the observed information is

Number of High grades (A's + B's)	= h
Number of C's	= c
Number of D's	= d

- What is the maximal likelihood estimator of μ now?

REMEMBER

P(A) = ½

P(B) = μ

P(C) = 2μ

P(D) = $½-3\mu$

Example (cont.)

- What is the MLE of μ now?
- We can answer this question circularly:

Number of High grades (A's + B's)	$= h$
Number of C's	$= c$
Number of D's	$= d$

Expectation

If we know the value of μ we could compute the expected value of a and b

$$a = \frac{\frac{1}{2}}{\frac{1}{2} + \mu} h \qquad b = \frac{\mu}{\frac{1}{2} + \mu} h$$

Since the ratio $a : b$ should be the same as the ratio ½ : μ

Maximization

If we know the expected values of a and b we could compute the maximum likelihood value of μ

$$\mu = \frac{b + c}{6(b + c + d)}$$

REMEMBER

P(A) = ½

P(B) = μ

P(C) = 2μ

P(D) = ½-3μ

Example – Algorithm

- We begin with a guess for μ, and then iterate between EXPECTATION and MAXIMALIZATION to improve our estimates of μ and a, b
- Define
  - $\mu^{(t)}$ the estimate of μ on the t-th iteration
  - $b^{(t)}$ the estimate of *b* on t-th iteration

$$\mu^{(0)} = \text{ initial guess}$$

$$b^{(t)} = \frac{\mu^{(t)} h}{1/2 + \mu^{(t)}} = \mathrm{E}\left[b \mid \mu^{(t)}\right]$$

E-step

$$\mu^{(t+1)} = \frac{b^{(t)} + c}{6\left(b^{(t)} + c + d\right)} = \text{ max like est. of } \mu \text{ given } b^{(t)}$$

M-step

Example – Algorithm (cont.)

- Iteration will converge
- But only assured to converge to local optimum

$$\mu^{(0)} = \text{ initial guess}$$

$$b^{(t)} = \frac{\mu^{(t)}h}{\frac{1}{2} + \mu^{(t)}} = \mathrm{E}\left[b \mid \mu^{(t)}\right]$$

$$\mu^{(t+1)} = \frac{b^{(t)} + c}{6\left(b^{(t)} + c + d\right)} = \text{ max like est. of } \mu \text{ given } b^{(t)}$$

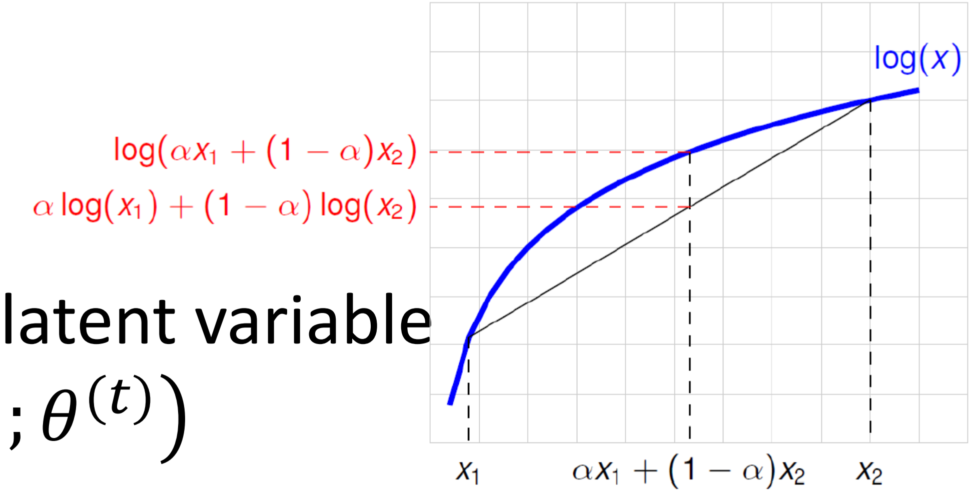
General EM methods

- Repeat until convergence{
  (E-step) For each data point i, set
  $$q_i(j) = p\left(z^{(i)} = j | x^{(i)}; \theta\right)$$
  (M-step) Update the parameters
  $$\theta = \mathrm{argmax}_\theta \sum_{i=1}^{N} \sum_{j} q_i(j) \log \frac{p(x^{(i)}, z^{(i)} = j; \theta)}{q_i(j)}$$
  }

Convergence of EM

- Denote $\theta^{(t)}$ and $\theta^{(t+1)}$ as the parameters of two successive iterations of EM, we want to prove that
$$l\left(\theta^{(t)}\right) \leq l\left(\theta^{(t+1)}\right)$$
which shows EM always monotonically improves the log-likelihood, thus ensures EM will at least converge to a local optimum

Proof



- Start from $\theta^{(t)}$, we choose the posterior of latent variable
$$q_i^{(t)}(j) = p\left(z^{(i)} = j | x^{(i)}; \theta^{(t)}\right)$$
- By Jensen's inequality on the log function
- $l\left(q; \theta^{(t)}\right) = \sum_{i=1}^{N} \sum_j q_i(j) \log \frac{p(x^{(i)}, z^{(i)} = j; \theta^{(t)})}{q_i(j)}$
- $\leq \sum_{i=1}^{j} \log \sum_j q_i(j) \cdot \frac{p(x^{(i)}, z^{(i)} = j; \theta^{(t)})}{q_i(j)}$
- $= \sum_{i=1}^{j} \log \sum_j p\left(x^{(i)}, z^{(i)} = j; \theta^{(t)}\right)$
- When $q_i(j) = p\left(z^{(i)} = j | x^{(i)}; \theta^{(t)}\right)$, the above inequality is equality

Proof (cont.)

- Then the parameter $\theta^{(t+1)}$ is obtained by maximizing

$$\sum_{i=1}^{N}\sum_{j} q_i^{(t)} \log \frac{p(x^{(i)}, z^{(i)} = j; \theta)}{q_i^{(t)}}$$

- Thus

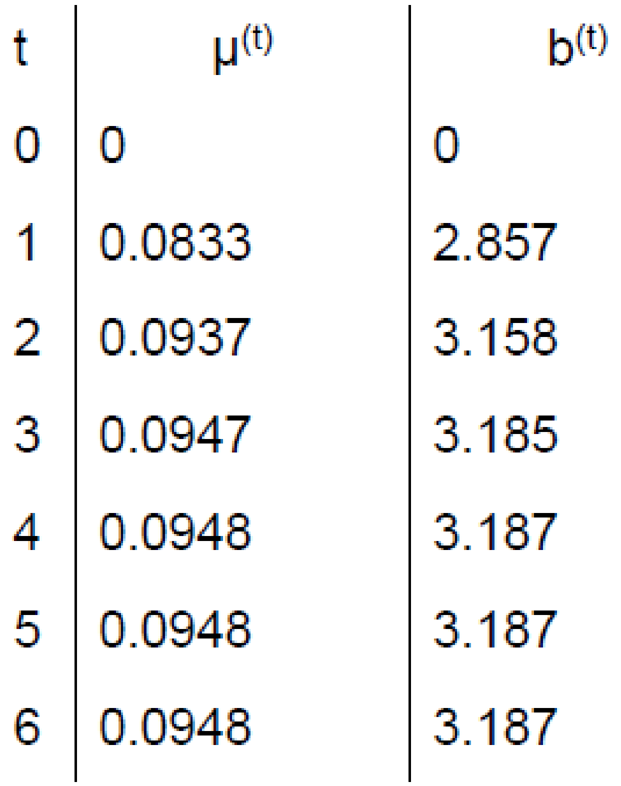
$$l\left(\theta^{(t+1)}\right) = \sum_{i=1}^{N}\sum_{j} q_i^{(t)} \log \frac{p\left(x^{(i)}, z^{(i)} = j; \theta^{(t+1)}\right)}{q_i^{(t)}}$$

$$\geq \sum_{i=1}^{N}\sum_{j} q_i^{(t)} \log \frac{p\left(x^{(i)}, z^{(i)} = j; \theta^{(t)}\right)}{q_i^{(t)}} = l\left(\theta^{(t)}\right)$$

- Since the likelihood is at most 1, EM will converge (to a local optimum)

Example

- Follow previous example, suppose $h = 20, c = 10, d = 10, \mu^{(0)} = 0$
- Then
- Convergence is generally linear: error decreases by a constant factor each time step



t	$\mu^{(t)}$	$b^{(t)}$
0	0	0
1	0.0833	2.857
2	0.0937	3.158
3	0.0947	3.185
4	0.0948	3.187
5	0.0948	3.187
6	0.0948	3.187

Advantages

- EM is often applied to NP-hard problems
- Widely used
- Often effective

K-Means and EM

- After initialize the position of the centers, K means interactively execute the following two operations:
  - Assign the label of the points based on their distances to the centers
    - Specific posterior distribution of latent variable
    - E-step
  - Update the positions of the centers based on the labeling results
    - Given the labels, optimize the model parameters
    - M-step