In The Name of Allah

Digital Media Laboratory
Sharif University of Technology

Statistical Pattern Recognition

Expectation Maximization (EM) and Mixture Models

Hamid R. Rabiee
Jafar Muhammadi, Mohammad J. Hosseini

Spring 2012
http://ce.sharif.edu/courses/90-91/2/ce725-1/

Template designed by Jafar Muhammadi

Agenda

- Expectation-maximization (EM) Overview
- EM Applications
- EM Algorithm
- EM Examples
- Mixture Models
- Gaussian Mixtures

Template designed by Jafar Muhammadi

Expectation-Maximization (EM)

- EM algorithm is a general technique for finding maximum likelihood estimators under missing (unobserved) data.
- EM is perhaps most often used and mostly half understood algorithm for unsupervised learning.
  - It is very intuitive
  - Many people rely on their intuition to apply the algorithm in different problem domains
- The EM algorithm estimates the parameters of a model iteratively.
  - Starting from some initial guess, each iteration consists of an Expectation step, and an Maximization step

Template designed by Jafar Muhammadi

Missing Data Problem

- **Occurs whenever part of the data is unknown**
  - **intrinsically inaccessible**
    - Example: which model does a data point belong to in mixture models?
  - **data is lost / erroneous**
    - Example: Some faulty / noisy process has generated the data.
- **If the missing data is correlated in any way with the observed, we can hope to extract information about the missing data from the observed.**
- **If the missing data is independent from the observed, everything is lost.**

Template designed by Jafar Muhammadi

EM Applications

- **Application Examples**
  - **PoS (Part of Speech) Tagging**
    - Complete data: A sentence (a sequence of words) and a corresponding sequence of PoS tags.
    - Observed data: the sentence
    - Unobserved data: the sequence of tags
    - Model: an HMM with transition/emission probability tables
  - **Model Building with Partial Observations ← We'll discuss this example today.**
    - Our goal is to build a probabilistic model
    - The model parameters can be estimated from a set of training examples: $x_1, x_2, \ldots, x_n$
      - **x_i's are i.i.d (identically and independently distributed)**
    - Unfortunately, we only get to observe part of each training example:
      - **$x_i=(x_{io}, x_{iu})$ and we can only observe x_{io}.**
    - How do we build the model?

Template designed by Jafar Muhammadi

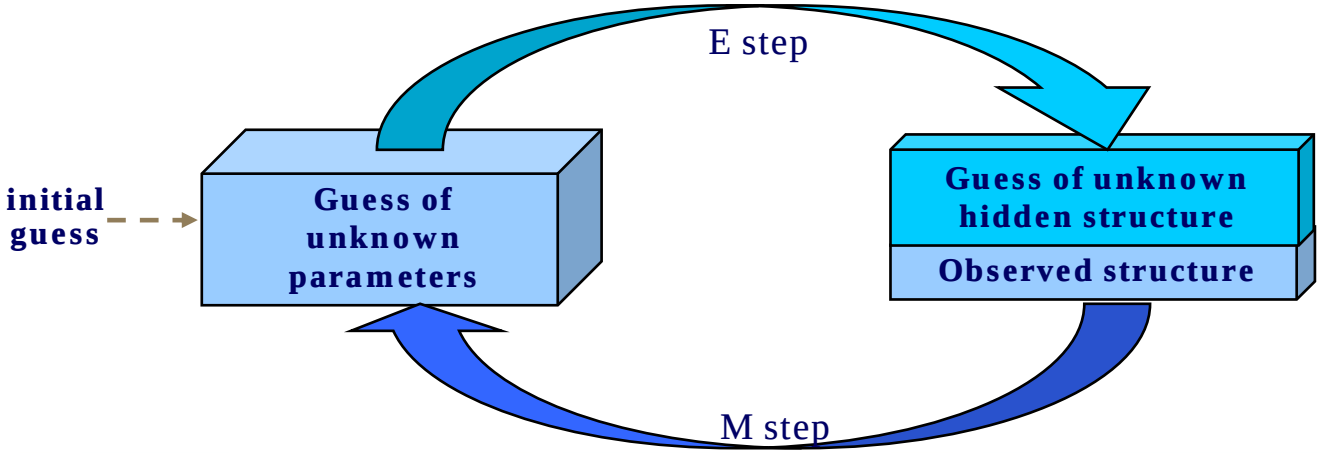
EM Applications

- **More applications**
  - Filling in missing data in samples
  - Discovering the value of latent variables
  - Estimating the parameters of HMMs
  - Estimating parameters of finite mixtures
  - Unsupervised learning of clusters
  - ...

Template designed by Jafar Muhammadi

EM Algorithm

- **General Idea**
  - **Given a set of incomplete (observed) data**
  - **Assume observed data come from a specific model**
  - **Iterate following steps until convergence**
    - Expectation step: formulate some parameters for that model, use this to guess the missing (latent / unobserved) data
    - Maximization step: from the missing data and observed data, find the most likely parameters



Template designed by Jafar Muhammadi

EM Algorithm

- **Assumptions:**
  - Suppose that observations are X_os. Latent data are X_us., and the unknown parameters are θ.
- **Initialization:**
  - Initialize the parameters of θ to some random value
- **Iterate following steps, until Convergence:**
  - E Step: Compute the best value for X_u given current parameters values.
  - M Step: Use the just-computed values of X_u to compute a better estimate for the parameters.

Template designed by Jafar Muhammadi

EM Algorithm

✧ General algorithm

1. **Choose an initial setting for the parameters $\theta^{(0)}$, and set *t=0*.**
2. **E step: Evaluate** $P(X_u \mid X_o, \theta^{(t)})$
3. **M step: Evaluate $\theta^{(t+1)}$ given by** $\theta^{(t+1)} = \arg\max_{\theta} \Phi(\theta, \theta^{(t)})$

   Where

$$\Phi(\theta, \theta^{(t)}) = E_{X_u \mid \theta^{(t)}, X_o}[\ln(P(X_o, X_u \mid \theta))] = \sum_{X_u} P(X_u \mid X_o, \theta^{(t)}) \ln P(X_o, X_u \mid \theta)$$

4. **Check for convergence of either the log likelihood or the parameter values . If the convergence criterion is not satisfied, then return to step 2.**

Template designed by Jafar Muhammadi

EM Algorithm

- **A simple example – Maximum likelihood**
  - **Assume after an exam in the class we have these grades:**

Grade	A	B	C	D
# of students	a	b	c	d

  - **And suppose that we know: P(A)=½, P(B)=μ, P(C)=2μ, P(D)=½-3μ**
  - **What's the maximum likelihood estimate of μ ?**

$$P(a,b,c,d\,|\,\mu) = K\left(\frac{1}{2}\right)^a (\mu)^b (2\mu)^c \left(\frac{1}{2}-3\mu\right)^d$$

$$\ln P(a,b,c,d\,|\,\mu) = \ln K + a\ln\frac{1}{2} + b\ln\mu + c\ln 2\mu + d\ln\left(\frac{1}{2}-3\mu\right)$$

$$\frac{\partial \ln P}{\partial \mu} = \frac{b}{\mu} + \frac{2c}{2\mu} - \frac{3d}{\frac{1}{2}-3\mu} = 0 \Rightarrow \mu = \frac{b+c}{6(b+c+d)}$$

Template designed by Jafar Muhammadi

EM Algorithm

Grade	A	B	C	D
# of students	a	b	c	d

P(A)=½, P(B)=μ, P(C)=2μ, P(D)=½-3μ

- **A simple example – Hidden Information**
  - **Suppose that we know that:**
    - number of high grades (A's + B's) = h
    - Number of C's = c
    - Number of D's = d
  - **What is the Maximum Likelihood estimate for μ now?**
  - **Expectation:**
    - If we knew the value of μ we could compute the expected value for a and b

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

Template designed by Jafar Muhammadi

EM Algorithm

Grade	A	B	C	D
# of students	a	b	c	d

P(A)=½, P(B)=μ, P(C)=2μ, P(D)=½-3μ

- **A simple example – Hidden Information (cont.)**
  - **Maximization:**
    - If we knew the expected values of a and b we could compute the maximum likelihood value of μ like before
    - **Then, we begin with a first estimate for μ and iterate between expectation and maximization to improve our estimates for μ, a and b.**

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

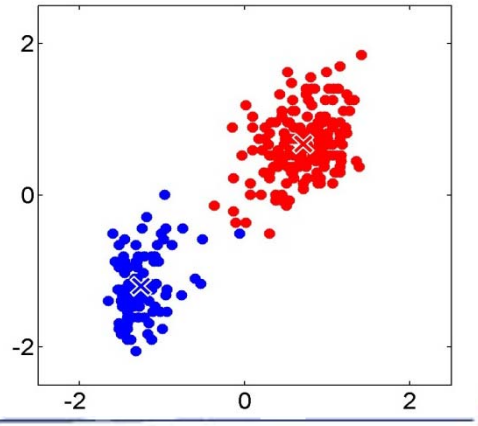
$$b(t) = \frac{\mu(t)h}{\frac{1}{2}+\mu(t)} = E\left[b \mid \mu(t)\right]$$

$$\mu(t+1) = \frac{b(t)+c}{6\left(b(t)+c+d\right)} = \text{ML estimates of } \mu \text{ given } b(t)$$

Template designed by Jafar Muhammadi

EM Algorithm

- **Another example: K-means clustering**
  - **Goal: represent a data set $\{x_1, \ldots, x_N\}$ in terms of K clusters each of which is summarized by a prototype μ_k**
  - **Initialize prototypes, then iterate between two phases:**
    - E-step: assign each data point to nearest prototype
    - M-step: update prototypes to be the cluster means
  - **Simplest version is based on Euclidean distance**
  - **HW: Derive the EM equations for k-means algorithm ($P(X_u \mid X_o, \theta^{(t)})$, $\Phi(\theta, \theta^{(t)})$).**



Template designed by Jafar Muhammadi

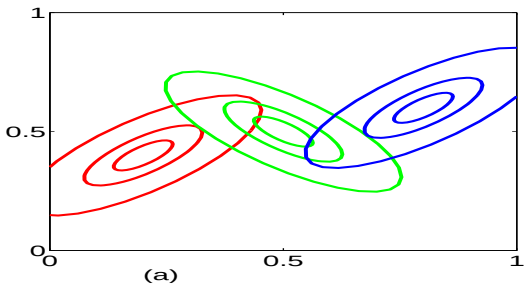
Mixture Models

- **Mixture density model estimation**
  - **Models data with mixture density** $P(x \mid \theta) = \sum_{j=1}^{m} p(x \mid c_j, \theta_j) P(c_j)$
  - **Where** $\theta = \{\theta_1, ..., \theta_m\}$, $P(c_1) + ... + P(c_m) = 1$
  - **To generate a sample from distribution *P(X|θ)***
    - first select class *j* with probability $P(c_j)$
    - then generate *x* according to probability $P(x|c_j, \theta_j)$
  - **Provides a framework for building more complex probability distributions**
  - **Can be used to cluster data (How?)**

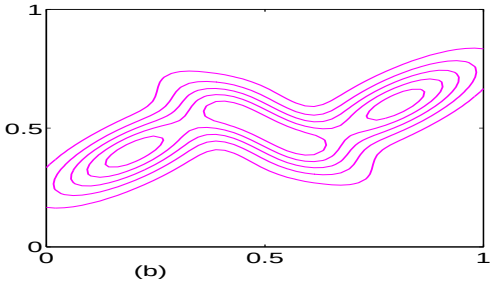
Template designed by Jafar Muhammadi

Gaussian Mixtures

- **Linear super-position of Gaussians** $P(x) = \sum_{k=1}^{K} P(c_k) N(x \mid \mu_k, \Sigma_k)$
- **Normalization and positivity require** $\sum_{K=1}^{k} P(c_K) = 1, \; 0 \le P(c_K) \le 1$
- **Example: Mixture of 3 Gaussians**



Separated

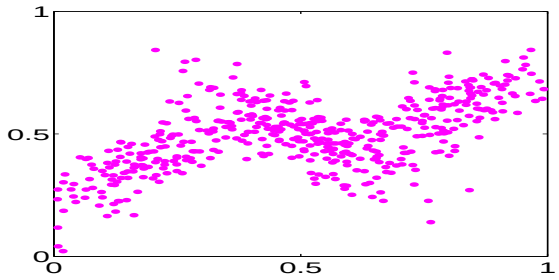


Mixed

Template designed by Jafar Muhammadi

Gaussian Mixtures

- **Fitting the Gaussian mixture model**
  - **The goal: given the data set, find the corresponding parameters:**
    - mixing coefficients (or prior probabilities), means, and covariances
  - **If we knew which component generated each data point, the maximum likelihood solution would involve fitting each component to the corresponding cluster**
    - Problem: the data set is unlabelled
    - We'll refer to the labels as *latent (= hidden)* variables
  - **Synthetic data set without labels**



Template designed by Jafar Muhammadi

Gaussian Mixtures

- **Maximum likelihood for the GMM**
  - **The log likelihood function takes the form**

$$\ln p(X \mid \pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k N(x_n \mid \mu_k, \Sigma_k) \right\}$$

Note: The sum over components, appears *inside* the log.

  - **There is no closed form solution for maximum likelihood.**
  - **Then, how to maximize the log likelihood?**
    - Using EM algorithm.

Template designed by Jafar Muhammadi

Gaussian Mixtures

- **EM Algorithm**
  - **Initialize the means μ_k, covariances Σ_k and mixing coefficients π_k.**
  - **Repeat the following steps until convergence:**
    - E step: Evaluate z_{ij}s (latent variables) using the current parameter values
      - **z_{ij} : a binary variable which is *1* if x_i is drawn from the j^{th} distribution**

$$z_{ij} \equiv p(c_j \mid x_i) = \frac{p(c_j)p(x_i \mid c_j)}{p(x_i)} = \frac{\pi_j N(x_i \mid \mu_j, \Sigma_j)}{\sum_{k=1}^{K} \pi_k N(x_i \mid \mu_k, \Sigma_k)}$$

    - M step: Re-estimate the parameters using the current z_{ij}s
      - Equations in the next slides

Template designed by Jafar Muhammadi

Gaussian Mixtures

- **EM algorithm – M step**
  - **Let us proceed by simply differentiating the log likelihood**
    - Setting derivative with respect to μ_k equal to zero, gives

$$-\sum_{i=1}^{N}\frac{\pi_k N(x_i \mid \mu_k,\Sigma_k)}{\sum_j \pi_j N(x_i \mid \mu_j,\Sigma_j)}\Sigma_k(x_i-\mu_k) = -\sum_{i=1}^{N} z_{ik}\Sigma_k(x_i-\mu_k) = 0$$

we suppose that z_{ik} values are known, in M step.

multiplying both sides by Σ_k^{-1} gives $\mu_k = \frac{1}{N_k}\sum_{n=1}^{N} z_{nk}x_n$, $N_k = \sum_{n=1}^{N} z_{nk}$, which is simply the weighted mean of the data

- **Similarly for the covariances, we obtain $\Sigma_k = \frac{1}{N_k}\sum_{n=1}^{N} z_{nk}(x_n-\mu_k)(x_n-\mu_k)^T$**
- **Note that the condition which requires the mixing coefficients to sum to 1, must be satisfied, when maximizing log-likelihood with respect to the π_k.**
  - Then, we use the Lagrange multiplier method, as shown in the next slide

Template designed by Jafar Muhammadi

Gaussian Mixtures

- **EM algorithm – M step**
  - **Estimating π_ks:**
    - Using Lagrange multiplier method, we must maximize the following quantity

$$\ln P(X \mid \pi, \mu, \Sigma) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right)$$

which gives

$$\sum_{n=1}^{N} \frac{N(x_n \mid \mu_k, \Sigma_k)}{\sum_j \pi_j N(x_n \mid \mu_j, \Sigma_j)} + \lambda = 0$$

multiplying both sides by π_k and sum over k, we find $\lambda = -N$. So $\pi_k = \frac{N_k}{N}$.

Template designed by Jafar Muhammadi

Gaussian Mixtures

- **EM algorithm – Latent variable view to obtain M step estimations:**
  - **We have:**

$$P(z_{nk}=1)=\pi_k \Rightarrow P(z_n)=\prod_{k=1}^{K}\pi_k^{z_{nk}}$$

$$P(x_n \mid z_k=1)=N(x_n \mid \mu_k,\Sigma_k) \Rightarrow P(x_n \mid z_n)=\prod_{k=1}^{K}N(x_n \mid \mu_k,\Sigma_k)^{z_{nk}}$$

  - **Then,**

$$p(X,Z \mid \mu,\Sigma,\pi)=\prod_{n=1}^{N}\prod_{k=1}^{K}\pi_k^{z_{nk}}N(x_n \mid \mu_k,\Sigma_k)^{z_{nk}}$$

$$\Rightarrow \ln p(X,Z \mid \mu,\Sigma,\pi)=\sum_{n=1}^{N}\sum_{k=1}^{K}z_{nk}\left\{\ln \pi_k+\ln N(x_n \mid \mu_k,\Sigma_k)\right\}$$

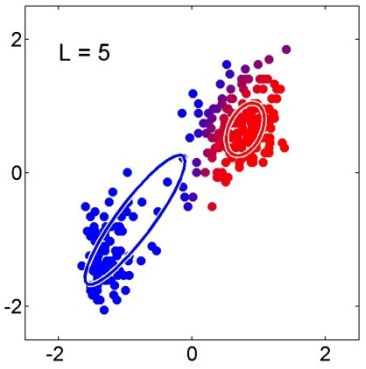
  - **Keeping the z_{ij}s fixed and maximizing with respect to the parameters give the previous results:**

$$\mu_k=\frac{1}{N_k}\sum_{n=1}^{N}z_{nk}x_n \qquad \Sigma_k=\frac{1}{N_k}\sum_{n=1}^{N}z_{nk}(x_n-\mu_k)(x_n-\mu_k)^T \qquad \pi_k=\frac{N_k}{N}$$

Template designed by Jafar Muhammadi

Gaussian Mixtures

- **Example:**
  - **Mixture of two Gaussians**
    - After *20* cycles the algorithm is close to convergence.



Template designed by Jafar Muhammadi

Any Question?

End of Lecture 14

Thank you!

Spring 2012

http://ce.sharif.edu/courses/90-91/2/ce725-1/

Template designed by Jafar Muhammadi