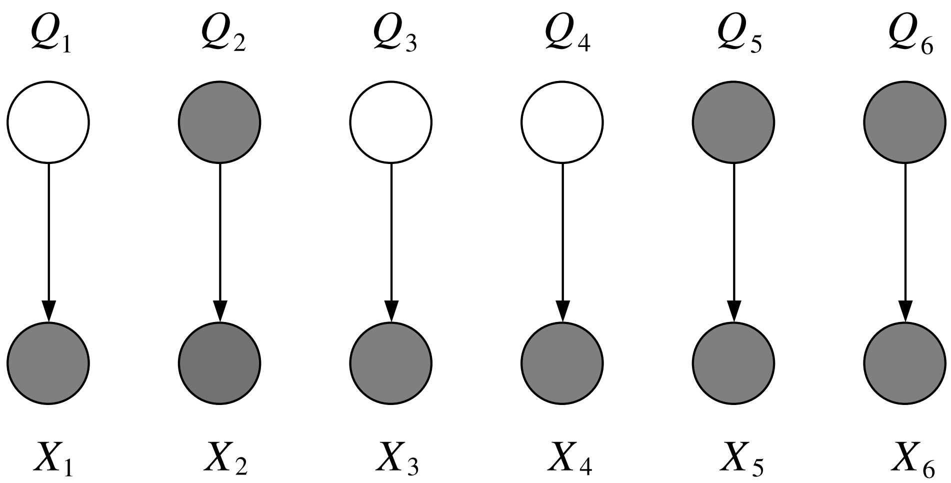
Uncertainty Modelling in AI

Mixture Models and the EM Algorithm

Lee Wee Sun

Semester 2, 2005

Unobserved Variables



- Some variables in a graphical model are sometimes or possibly all the time unobserved. We shade the nodes in the model to indicate observed variables.
- If variables are occasionally unobserved, they are missing data e.g. missing input values or class labels.
- If variables are always unobserved, they are usually called latent or hidden variables.

- In probability theory, observations with some variables missing means that the variables have been marginalized i.e. summed or integrated away.
- For example, likelihood function in the presence of some missing class labels:

$$\begin{aligned} l(\theta; D) &= \sum_{\text{complete}} \log p(x^c, y^c|\theta) + \sum_{\text{missing}} \log p(x^m|\theta) \\ &= \sum_{\text{complete}} \log p(x^c, y^c|\theta) + \sum_{\text{missing}} \log \sum_y p(x^m, y|\theta) \end{aligned}$$

Latent Variables

- Random variables whose values are usually not specified in the observed data (as opposed to sometimes being missing) but contains useful information are usually called *latent* (or *hidden* or *mediating*) variables.
- Latent variables may be variables that cannot be measured because of occlusion, privacy, ... etc.
- Often we introduce latent variables into the models because they create simpler models with fewer parameters.

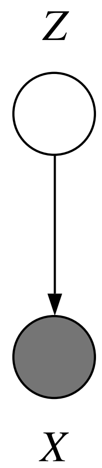
- We will look at models based on discrete latent variables: unconditional mixture models (we will skip conditional mixture models)
- Unconditional models solve the density estimation problem whereas conditional models solve the regression and classification problems.
- Latent variable methodology in these problems allows the problems to be broken down into subproblems allowing a "divide and conquer" approach to statistical modelling.

Unconditional Mixture Models

- Useful for density estimation which allows us to answer queries on how "typical" or "untypical" a particular observation is.
- Most of the distributions we described earlier in the course are very simple. By using mixtures, we can approximate arbitrarily complicated distributions.
- Direct application for density estimation include detection of outliers and algorithms for data compression. For these applications, we are only interested in obtaining a good estimate of the probability density.

- In other applications, we may actually be interested in the structure of the mixture model. For example, we may think that there are “subpopulations” underlying the data and would like to identify them.
- This is closely related to classification with generative models. We normally refer to settings where the labels of the latent variables are observed as *classification* while settings where the labels are absent are refered as *clustering*.

- Graphical model for a mixture model is shown below:



- Let the latent variable be a multinomial random variable denoted Z with components Z^i. Then

$$p(x|\theta) = \sum_i p(Z^i = 1|\pi_i)p(x|Z^i = 1, \theta_i) = \sum_i \pi_i p(x|Z^i = 1, \theta_i),$$

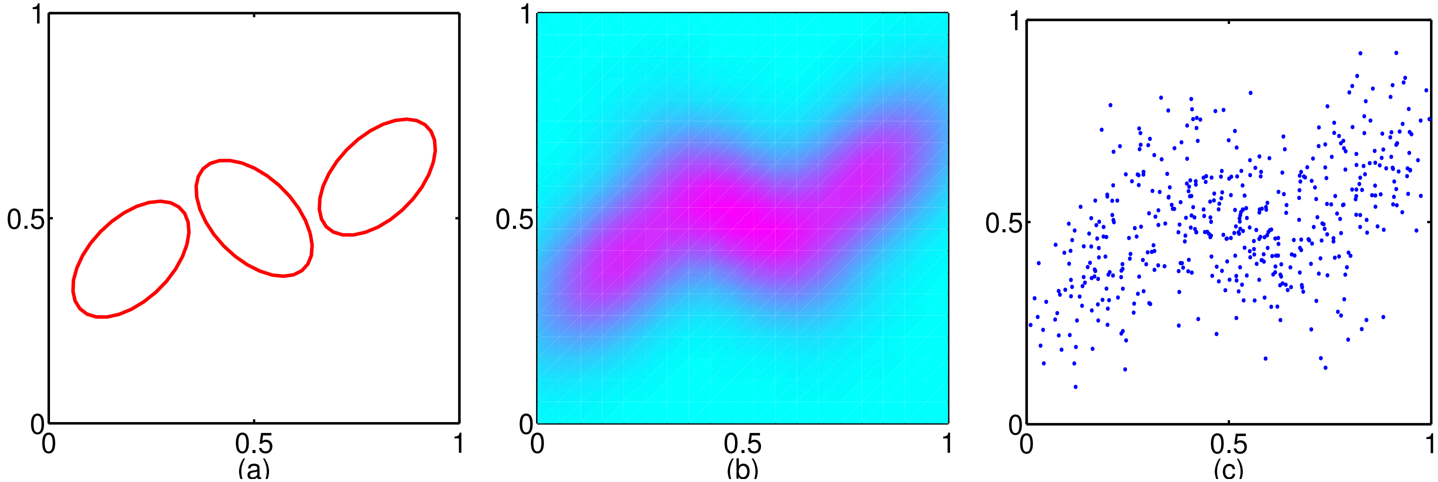
where $\theta = (\pi_1, \ldots, \pi_K, \theta_1, \ldots, \theta_K)$ and the π_i sum to one.

- The mixture components are usually simple distributions such as Gaussians or multinomials.

- Using Bayes rule, we can calculate also the probability that x was generated from the k-th component

$$p(Z^k = 1|x, \theta) = \frac{\pi_k p(x|Z^k = 1, \theta_k)}{\sum_j \pi_k p(x|Z^j = 1, \theta_j)}.$$

- Gaussian example with three components shown below.



Expectation Maximization (EM) Algorithm

- When latent variables are present, there is a sum inside the log in the log likelihood function, so optimization is not as simple as in the fully observed model

$$l(\theta; D) = \log \sum_z p(x, z|\theta) = \log \sum_z p(z|\theta_z) p(x|z, \theta_x).$$

- If the latent variable is observed as in the classification models, then the parameters decouple and estimation is easy

$$l(\theta; D) = \log p(x, z|\theta) = \log p(z|\theta_z) + \log p(x|z, \theta_x).$$

- The EM algorithm iteratively uses $p(z|x, \theta^t)$, where θ^t is the current model to guess the value of the latent variable and then do estimation with the fully observed model.

E-step: The latent variables $\hat{z}^t$ are given (fractional) observations $p(z|x, \theta^t)$.

M-step: The parameters are updated using

$$\theta^{t+1} \leftarrow \arg\max l(\theta, x, \hat{z}^t).$$

The E-step can often be efficiently calculated using graphical models. The M-step is usually no harder than in the fully observed case.

- The procedure monotonically improves l hence it always converges to a local maximum of the likelihood function.

Example: EM with Naive Bayes

- Let the latent variable $\mathbf{Z}$ be a multinomial random variable with components $\mathbf{Z}^i$, defining the prior probability vector π, where $\pi_i = p(\mathbf{Z}^i = 1|\pi)$.

- Within each feature j, we have a multinomial random variable $\mathbf{X}_j$ with components $\mathbf{X}_j^k$, where $\mathbf{X}_j^k = 1$ for one and only one value of k.

- Let $\eta_{ijk} = p(\mathbf{x}_j^k = 1|\mathbf{Z}^i = 1, \eta)$, the probability that the j-th feature $\mathbf{X}_j$ takes on its k-th value when $\mathbf{Z}^i = 1$. The class conditional density is then

$$p(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m|\mathbf{Z}^i = 1, \eta) = \prod_j^m \prod_k \eta_{ijk}^{\mathbf{x}_j^k}.$$

- Using Bayes rule, we obtain:

$$p(\mathbf{Z}^i = 1|\mathbf{x}, \eta) = \frac{\pi_i \prod_j \prod_k \eta_{ijk}^{\mathbf{x}_j^k}}{\sum_l \pi_l \prod_j \prod_k \eta_{ljk}^{\mathbf{x}_j^k}}.$$

- Given data $(\mathbf{z}_1, \mathbf{x}_1), \ldots, (\mathbf{z}_N, \mathbf{x}_N)$, the complete log likelihood is

$$\sum_n^N \sum_i \mathbf{z}_n^i \log \pi_i + \sum_n^N \sum_i \sum_j \sum_k \mathbf{z}_n^i \mathbf{x}_{n,j}^k \log \eta_{ijk}.$$

- The MLE estimate of π_i is $\sum_n \mathbf{z}_n^i / N$.
- The MLE estimate of η_{ijk} is $\frac{\sum_n \mathbf{x}_{j,n}^k \mathbf{z}_n^i}{\sum_n \mathbf{z}_n^i}$.

- When the $\mathbf{z}_n$ are not observed, we can use the EM algorithm.
- Denote the parameters (π_i, η_{ijk} etc.) after the tth iteration as $\theta^{(t)}$.
- In EM, we replace $\mathbf{z}_n$ with $E(\mathbf{z}_n|\theta^{(t)})$ in the log likelihood to get

$$\sum_n^N \sum_i E(\mathbf{z}_n^i|\theta^{(t)}) \log \pi_i + \sum_n^N \sum_i \sum_j \sum_k E(\mathbf{z}_n^i|\theta^{(t)}) \mathbf{x}_{n,j}^k \log \eta_{ijk}.$$

- Maximizing, we get $\pi_i^{(t+1)} = \sum_n E(\mathbf{z}_n^i|\theta^{(t)})/N$ and $\eta_{ijk}^{(t+1)} = \frac{\sum_n \mathbf{x}_{j,n}^k E(\mathbf{z}_n^i|\theta^{(t)})}{\sum_n E(\mathbf{z}_n^i|\theta^{(t)})}$.
- Using the new values of the parameters, the process is repeated until convergence.

- The expected value of $\mathbf{z}$ for any of the instance is calculated:

$$E(\mathbf{z}^i|\theta^{(t)}) = p(\mathbf{Z}^i = 1|\mathbf{x}, \theta^{(t)}) = \frac{\pi_i^{(t)} \prod_j \prod_k (\eta_{ijk}^{(t)})^{\mathbf{x}_j^k}}{\sum_l \pi_l^{(t)} \prod_j \prod_k (\eta_{ljk}^{(t)})^{\mathbf{x}_j^k}}.$$

EM increases Log Likelihood

- With the latent variables z observed, the *complete log likelihood* is
$$l_c(\theta; x, z) = \log p(x, z|\theta).$$
- With z unobserved, the function we want to optimize is the log of the *marginal* probability which we call the *incomplete log likelihood*
$$l(\theta; x) = \log p(x|\theta) = \log \sum_z p(x, z|\theta).$$
- Define the *expected log likelihood* using "averaging distribution" $q(z|x)$
$$\langle l_c(\theta; x, z)\rangle_q = \sum_z q(z|x, \theta) \log p(x, z|\theta).$$
- The expected complete log likelihood is linear in the complete log likelihood and inherit its favorable computational

properties.

- We first show the following relationship between the log likelihood and the expected log likelihood, which holds because of the concavity of the log function and Jensen's inequality (described later)

$$l(\theta; x) \geq l_q(\theta; x) + H(q)$$

where $H(q) = -\sum_z q(z|x,\theta) \log q(z|x,\theta)$ is the entropy function. The right hand side is a *lower bound* for the left hand side.

$$
\begin{aligned}
l(\theta; x) &= \log p(x|\theta) \\
&= \log \sum_z p(x, z|\theta) \\
&= \log \sum_z q(z|x) \frac{p(x, z|\theta)}{q(z|x)} \\
&\geq \sum_z q(z|x) \log \frac{p(x, z|\theta)}{q(z|x)} \quad \text{using Jensen's inequality} \\
&= \sum_z q(z|x) \log p(x, z|\theta) - \sum_z q(z|x) \log q(z|x).
\end{aligned}
$$

- Jensen's inequality states that for a concave function f

$$f(\sum_{k=1}^{n} \lambda_k x_k) \geq \sum_{k=1}^{n} \lambda_k f(x_k),$$

where $\lambda_k > 0$ and $\sum_k = 1^n \lambda_k = 1$. A function is concave if

$$f(\lambda a + (1-\lambda)b) \leq \lambda f(a) + (1-\lambda) f(b)$$

for $\lambda \in (0,1)$.

- Define an *auxiliary function*

$$\begin{aligned} L(q,\theta) &= \sum_z q(z|x) \log \frac{p(x,z|\theta)}{q(z|x)} \\ &= l_q(\theta;x) + H(q). \end{aligned}$$

The auxiliary function is a lower bound for the log likelihood $L(q,\theta) \leq l(\theta;x)$

- The EM algorithm is a coordinate ascent algorithm on $L(q, \theta)$

  E-step: $q^{(t+1)} = \arg\max_q L(q, \theta^{(t)})$

  M-step: $\theta^{(t+1)} = \arg\max_\theta L(q^{(t+1)}, \theta)$.

- Consider first the M-step. In $L(q, \theta) = l_q(\theta; x) + H(q)$, only the first term, which is the expected complete log likelihood depend on θ. Hence, the M-step is equivalent to maximizing the expected complete log likelihood.

- Now consider the E-step, maximization of $L(q, \theta^(t))$ with respect to q. We verify that $q^{(t+1)}(z|x) = p(z|x, \theta^{(t)})$ maximizes $L(q, \theta^(t))$.

$$\begin{aligned} L(p(z|x, \theta^{(t)}), \theta^{(t)}) &= \sum_z p(z|x, \theta^{(t)}) \log \frac{p(x, z|\theta^{(t)})}{p(z|x, \theta^{(t)})} \\ &= \sum_z p(z|x, \theta^{(t)}) \log p(x|\theta^{(t)}) \\ &= \log p(x|\theta^{(t)}) \\ &= l(\theta^{(t)}; x). \end{aligned}$$

As $l(\theta^{(t)}; x) \geq L(p(z|x, \theta^{(t)}), \theta^{(t)})$, this shows that $q^{(t+1)}(z|x) = p(z|x, \theta^{(t)})$ maximizes $L(q, \theta^(t))$.

- At each E-step and each M-step, the log likelihood either increases or remain the same, hence the algorithm converges to a local maximum of the likelihood function.

- Other than maximizing the likelihood for mixture models, the EM algorithm can also be used to learn when there is missing variables. In this case the cost function is

$$l(\theta; D) = \sum_{\text{complete}} \log p(x^c, y^c|\theta) + \sum_{\text{missing}} \log \sum_y p(x^m, y|\theta)$$

- At the E-step, we estimate the hidden variables in the incomplete case only. The M-step then optimizes the combination of the log likelihood on the complete data and the expected log likelihood on the incomplete data.

Properties of EM

- Each iteration is efficient if the E-step and M-step are efficient. With graphical models, the E-step is often efficient. The maximization step is very efficient for simple distributions such as the multinomial distribution.
- Each step will increase the likelihood function.
- Converge to a local maximum which may not be a global maximum. The quality of the solution can be sensitive to the initialization of the parameters. If some labeled data is available, this can be used to obtain reasonable initial parameters. Otherwise random parameters are often used for initialization.

- Commonly used for training mixture models, hidden markov models and models with missing variables.
- The M-step is sometimes computationally expensive to compute exactly. In this case, a gradient step that improves the likelihood slightly is often used. This is called Generalized EM.