A Review on EM algorithm

J.R

March 31, 2022

In this note, I want to present a review on the amazing Expectation Maximization(EM) algorithm, including the motivation, derivation and application in two concrete models and in general bayesian networks. EM algorithm is covered in CS229 by Andrew Ng and in the book *Probabilistic Graphical Models : Principles and Techniques* by Daphne Koller, however, in CS229, many steps in the derivation of mixture of Gaussian model and factor analysis model were skipped, and in that book, the theoretical foundation of EM algorithm seems to be a little obscure. Hopefully this note can be a good complementary material for Unsupervised Learning section of CS229 and chapter 19 of the book by Daphne Koller.

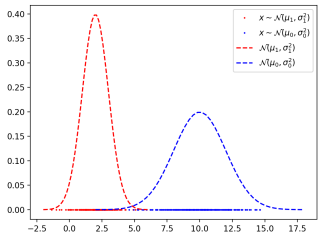
In this note, we will pose the problem of dealing with partially observed data, develop EM algorithm in a general context, apply EM algorithm in two specific models, and apply EM algorithm in general Bayesian Networks learning.

Contents

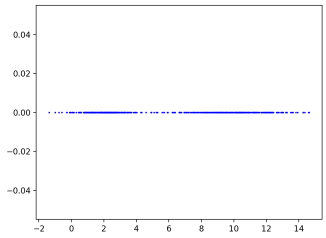
1 Motivation

The EM algorithm is an approach to perform Maximum Likelihood Estimation of partially observed models or models that includes latent variable. Let's illustrate this motivation by a simple example of mixture of Gaussian model.

Suppose we have a random variable $Z \sim multinomial(\phi)$, i.e. $p(Z = j) = \phi_j$. We first sample Z from the multinomial distribution. Having sampled a Z, if $Z = j$, we sample X from a gaussian $X \sim \mathcal{N}(\mu_j, \Sigma_j)$, as shown in Figure 1(a). This model can be represented with graphical model as shown in Figure 2, where $\theta_{cpd} = \{\mu_j, \Sigma_j \forall j\}$.

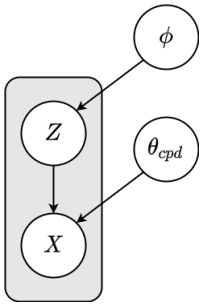


(a) Mixture of Gaussians

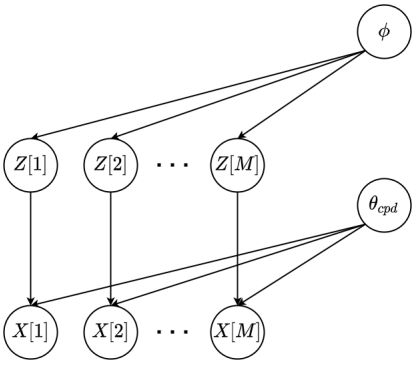


(b) Observed Data

Figure 1: Mixture of 2 Gaussians



(a) Template model



(b) Unrolled model

Figure 2

If we have observed the complete data $\mathcal{D} = \{(Z[1], X[1]), ...(Z[M], X[M])\}$, we can easily get the maximum likelihood estimation of $\mu_1, \mu_2, \Sigma_1, \Sigma_2$ based on the decomposability of likelihood function of bayesian network. However, if we have observed only X, as shown in Figure 1(b), how can we estimate the parameters, having known that the data comes from two Gaussians? The answer is, not surprisingly, the topic of this note, EM algorithm.

2 EM Algorithm

In the section, we will develop the general framework of EM algorithm. Suppose we have variable $\{Z, X\}$ in a model and dataset $\mathcal{D} = \{(Z[1], X[1]), ...(Z[M], X[M])\}$, where $X's$ are observed and

$Z's$ are hidden (not observed and take value "?"), we can still try maximizing its likelihood function, but what we try maximizing is the "marginal likelihood" $l(\theta; x[1], ..., x[M])$ in stead of the likelihood function $l(\theta; x[1], ..., x[M], z[1], ..., z[M])$ where $\theta = \{\phi, \mu_1, \mu_2, \Sigma_1, \Sigma_2\}$. Note that we use upper-case letter to indicate a random variable and lower-case letter to indicate a instantiation of a random variable.

$$\begin{aligned} l(\theta) &= \sum_{m=1}^{M} \log P(x[m]; \theta) \\ &= \sum_{m=1}^{M} \log \sum_{z[m] \in val(Z)} P(z[m], x[m]; \theta) \\ &= \sum_{m=1}^{M} \log \sum_{z[m] \in val(Z)} Q_m(z[m]) \frac{P(z[m], x[m]; \theta)}{Q_m(z[m])} \end{aligned} \tag{2.1}$$

In (2.1), for each $m = 1, ..., M$, we construct a distribution Q_m over possible values for $Z[m]$. Q_m seems to be coming out of nowhere. But let keep going on, we'll see where it lands in a few minutes. For now we just treat each Q_m as a known distribution Having that Q_m is a distribution, we can write (2.1) in the form of expectation

$$\begin{aligned} l(\theta) &= \sum_{m=1}^{M} \log \, E_{z[m] \sim Q_m}[\frac{P(z[m], x[m]; \theta)}{Q_m(z[m])}] \\ &\geq \sum_{m=1}^{M} E_{z[m] \sim Q_m} \left[log \, \frac{P(z[m], x[m]; \theta)}{Q_m(z[m])} \right] \qquad (2.2) \\ &= \sum_{m=1}^{M} \sum_{z[m] \in val(Z)} Q_m(z[m]) log \, \frac{P(z[m], x[m]; \theta)}{Q_m(z[m])} \qquad (2.3) \\ &= g(\theta) \end{aligned}$$

Having observed the value of $x[m]$, $log \frac{P(z[m], x[m]; \theta)}{Q_m(z[m])}$ is it self a random variable. In (2.2), we use the concavity of log function and the Jensen's inequality that for a concave function $f(x)$, $E[f(X)] \leq f(E[X])$, as illustrated in Figure 3. Then we expand the expectation and get (2.3).

Now we have that (2.3) is a lower bound of the log-likelihood function $l(\theta)$ for any valid θ. Just like what we do in Lagrange duality, a lower bound is not very interesting because the lower bound may take value $-\infty$. On the other hand, we want to have a good enough low bound. Wouldn't it be great if we can have a lower bound that's "tight" at θ? (2.2) takes equality.

In (2.2), since the log function is strictly concave, the equality holds only when $\frac{P(z[m], x[m]; \theta)}{Q_m(z[m])}$ is a constant for any value of $z[m]$. Thus we have $Q_m(z[m]) = \frac{1}{C} P(z[m], x[m]; \theta)$. Because Q_m is a probability distribution, it must sums to 1, so we have $\sum_{z[m] \in val(Z[m])} Q_m(z[m]) = 1$. On the other hand

$$\begin{aligned} \sum_{z[m] \in val(Z[m])} Q_m(z[m]) &= \sum_{z[m] \in val(Z[m])} \frac{1}{C} P(z[m], x[m]; \theta) \\ &= \frac{1}{C} P(x[m]; \theta) \end{aligned}$$

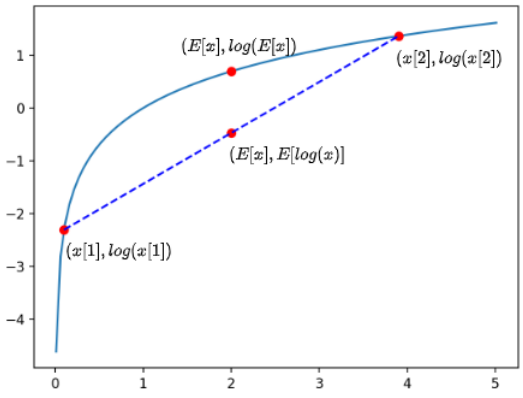


Figure 3: Jensen's inequality with $P(x = 0.1) = P(x = 3.9) = 0.5$

Then we have $C = P(x[m]; \theta)$, and $Q_m(z[m]) = \frac{P(z[m],x[m];\theta)}{P(x[m];\theta)} = P(z[m]|x[m]; \theta)$.

When we take $Q_m(z[m]) = P(z[m]|x[m]; \theta)$, $l(\theta) = g(\theta)$, and $l(\theta) \leq g(\theta)$ for all θ. We can first take $Q_m(z[m]) = P(z[m]|x[m]; \theta)$ to obtain a lower bound $g(\theta)$, then maximize that lower bound over θ to get θ', then $l(\theta') \geq g(\theta') \geq g(\theta) = l(\theta)$. We get an increase in the log-likelihood function. The increase in log-likelihood function is shown in Figure 4.

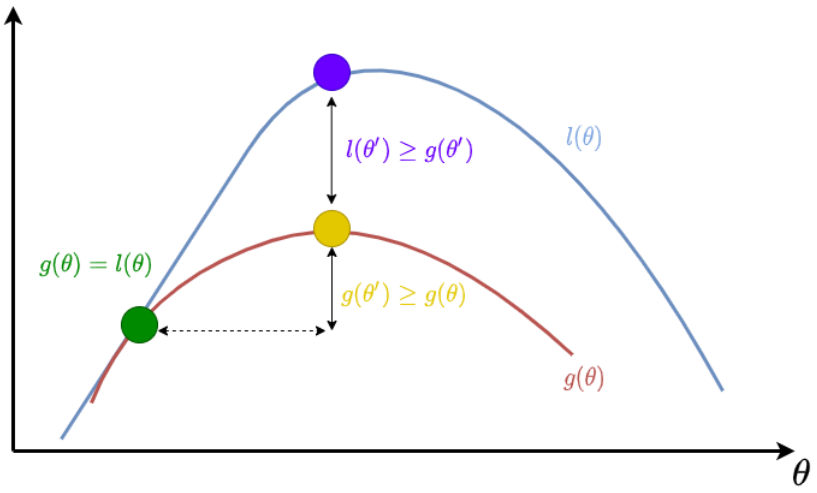


Figure 4: Parameter update in EM algorithm

Then we can conclude the algorithm as following

repeat until convergence

- E-step: $Q_m(z[m]) = P(z[m]|x[m]; \theta)$
- M-step: $\theta = \text{argmax}_\theta \sum_{m=1}^{M} \sum_{z[m] \in val(Z)} Q_m(z[m]) log \frac{P(z[m],x[m];\theta)}{Q_m(z[m])}$

Note that $logP(z[m], x[m]; \theta) = logP(z[m]; \theta) + logP(x[m]|z[m]; \theta)$, if $P(z[m]; \theta)$ and $P(x[m]|z[m]; \theta)$ are both log-concave in parameters (as a lot of distributions are), we can guarantee to obtain a global maximum with standard convex optimization techniques.

3 EM for Mixture of Gaussian

In this section, we will apply EM algorithm to mixture of Gaussian model, which is covered in CS229. Recall that in mixture of Gaussian model, we have $Z \sim Bernoulli(\phi)$, and $(X|Z = j) \sim \mathcal{N}(\mu_j, \Sigma_j)$

$$
\begin{aligned}
l(\theta) &= \sum_{m=1}^{M} log \sum_{j \in val(Z)} P(Z[m] = j, x[m]; \theta) \\
&= \sum_{m=1}^{M} log \sum_{j \in val(Z)} Q_m(Z[m] = j) \frac{P(Z[m] = j, x[m]; \theta)}{Q_m(Z[m] = j)} \\
&\geq \sum_{m=1}^{M} \sum_{j \in val(Z)} Q_m(Z[m] = j) log \frac{P(Z[m] = j, x[m]; \theta)}{Q_m(Z[m] = j)} \\
&\geq \sum_{m=1}^{M} \sum_{j \in val(Z)} Q_m(Z[m] = j) \big[\\
&\qquad log(\phi_j) - \frac{n}{2} log 2\pi - \frac{1}{2} logdet(\Sigma_j) - \frac{1}{2}(x[m] - \mu_j)^T \Sigma_j^{-1}(x[m] - \mu_j) - log(Q_m(Z[m] = j)) \\
&\quad \big] \\
&= g(\theta)
\end{aligned}
$$

3.1 E-step

In E-step, we compute

$$
\begin{aligned}
Q_m(Z[m] = j) &= P(Z[m] = j | x[m]; \theta) \\
&= \frac{P(Z[m] = j) P(x[m] | Z[m] = j; \theta)}{\sum_k P(Z[m] = k) P(x[m] | Z[m] = k; \theta)} \\
&= \frac{1}{Z} \frac{\phi_j}{(2\pi)^{n/2} |\Sigma_j|^{1/2}} exp(-\frac{1}{2}(x[m] - \mu_j)^T \Sigma_j^{-1}(x[m] - \mu_j))
\end{aligned}
$$

where Z is the normalizing constant.

3.2 M-step

In M-step, we maximize $g(\theta)$ over each variable in θ

- max over μ_j

$$
\begin{aligned}
max_{\mu_j} \sum_{m=1}^{M} -\frac{1}{2} Q_m(Z[m]=j)(x[m]-\mu_j)^T \Sigma_j^{-1}(x[m]-\mu_j) \\
\nabla \mu_j &= \sum_{m=1}^{M} Q_m(Z[m]=j)\Sigma_j^{-1}(x[m]-\mu_j) \\
&= \Sigma_j^{-1} \sum_{m=1}^{M} Q_m(Z[m]=j)(x[m]-\mu_j) \\
&= \Sigma_j^{-1}(\sum_{m=1}^{M} Q_m(Z[m]=j)x[m] - \sum_{m=1}^{M} Q_m(Z[m]=j)\mu_j) := 0 \\
\mu_j &= \frac{\sum_{m=1}^{M} Q_m(Z[m]=j)x[m]}{\sum_{m=1}^{M} Q_m(Z[m]=j)}
\end{aligned}
$$

- max over Σ_j

$$
\begin{aligned}
&\text{let } A_j = \Sigma_j^{-1} \\
&max_{A_j} \sum_{m=1}^{M} Q_m(Z[m]=j)\big[-\frac{1}{2} logdet(A_j^{-1}) - \frac{1}{2}(x[m]-\mu_j)^T A_j (x[m]-\mu_j)\big] \\
&= max_{A_j} \sum_{m=1}^{M} Q_m(Z[m]=j)\big[logdet(A_j) - (x[m]-\mu_j)^T A_j (x[m]-\mu_j)\big] \\
&= max_{A_j} \sum_{m=1}^{M} Q_m(Z[m]=j)\big[logdet(A_j) - tr(A_j, (x[m]-\mu_j)(x[m]-\mu_j)^T)\big] \\
&= max_{A_j} logdet(A_j) \sum_{m=1}^{M} Q_m(Z[m]=j) - tr(A_j, \sum_{m=1}^{M} Q_m(Z[m]=j)(x[m]-\mu_j)(x[m]-\mu_j)^T) \\
&\nabla A_j = A_j^{-1} \sum_{m=1}^{M} Q_m(Z[m]=j) - \sum_{m=1}^{M} Q_m(Z[m]=j)(x[m]-\mu_j)(x[m]-\mu_j)^T := 0 \\
&\Sigma_j = A_j^{-1} = \frac{\sum_{m=1}^{M} Q_m(Z[m]=j)(x[m]-\mu_j)(x[m]-\mu_j)^T}{\sum_{m=1}^{M} Q_m(Z[m]=j)}
\end{aligned}
$$

Here we use the fact that the $logdet(\cdot)$ function is concave and that $\frac{\partial}{\partial X} logdet(X) = X^{-1}$ for symmetric positive definite X. Refer to Appendix A for more detail.

- max over ϕ

$$
\begin{aligned}
& max_\phi \sum_{m=1}^{M} \sum_{j \in val(Z)} Q_m(Z[m]=j) log(\phi_j) \\
& s.t. \sum_j \phi_j = 1 \\
= & min_\phi - \sum_{m=1}^{M} \sum_{j \in val(Z)} Q_m(Z[m]=j) log(\phi_j) \\
& s.t. \sum_j \phi_j = 1
\end{aligned}
$$

We use Lagrangian to solve this optimization problem.

$$
\begin{aligned}
\mathcal{L}(\phi, v) &= -\sum_{m=1}^{M} \sum_{j \in val(Z)} Q_m(Z[m]=j) log(\phi_j) + v(\sum_j \phi_j - 1) \\
\nabla_{\phi_j} \mathcal{L} &= -\frac{1}{\phi_j} \sum_{m=1}^{M} Q_m(Z[m]=j) + v := 0 \\
\phi_j &= \frac{\sum_{m=1}^{M} Q_m(Z[m]=j)}{v} \\
g(v) &= \inf_\phi \mathcal{L}(\phi, v) \\
&= -\sum_{m=1}^{M} \sum_{j \in val(Z)} Q_m(Z[m]=j) log \frac{\sum_{m=1}^{M} Q_m(Z[m]=j)}{v} + \sum_{m=1}^{M} \sum_{j \in val(Z)} Q_m(Z[m]=j) - v \\
\nabla_v g &= \frac{1}{v} \sum_{m=1}^{M} \sum_{j \in val(Z)} Q_m(Z[m]=j) - 1 := 0 \\
v^* &= \sum_{m=1}^{M} \sum_{j \in val(Z)} Q_m(Z[m]=j) = M \\
\phi_j^* &= \frac{\sum_{m=1} Q_m(Z[m]=j)}{M}
\end{aligned}
$$

Thus we can conclude the EM algorithm for mixture of Gaussian model

Algorithm 1: Algorithms for solving mixture of Gaussian

while *not convergence* **do**

E-step:

$$Q_m(Z[m]=j) = \frac{\phi_j}{(2\pi)^{n/2}|\Sigma_j|^{1/2}} exp(-\frac{1}{2}(x[m]-\mu_j)^T\Sigma_j^{-1}(x[m]-\mu_j))$$

$$Q_m(Z[m]=j)/ = \sum_{j\in val(Z)} Q_m(Z[m]=j)$$

M-step:

$$\mu_j = \frac{\sum_{m=1}^{M} Q_m(Z[m]=j)x[m]}{\sum_{m=1}^{M} Q_m(Z[m]=j)}$$

$$\Sigma_j = \frac{\sum_{m=1}^{M} Q_m(Z[m]=j)(x[m]-\mu_j)(x[m]-\mu_j)^T}{\sum_{m=1}^{M} Q_m(Z[m]=j)}$$

$$\phi_j = \frac{\sum_{m=1} Q_m(Z[m]=j)}{M}$$

end

Figure 5 shows the result of Algorithm 1 on observed data of Figure 1(b).

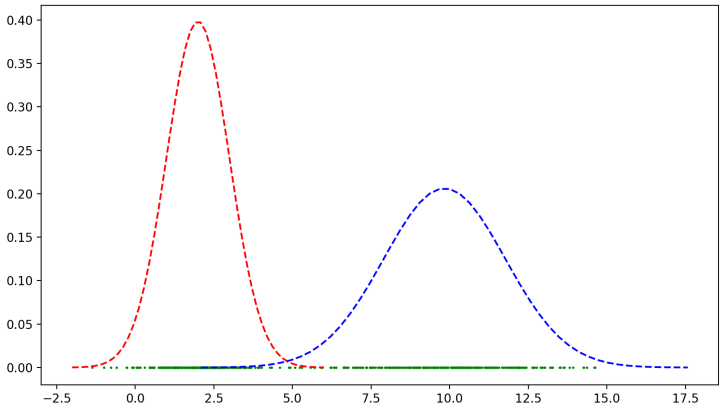


Figure 5

4 EM for Factor Analysis

In this section we apply EM algorithm to factor analysis model, which is also covered in CS229. In factor analysis, there is a random variable $Z \sim \mathcal{N}(0,1)$. Having observed Z, random variable X has a linear Gaussian distribution $X \sim \mathcal{N}(\mu + \lambda Z, \Phi)$. Figure 6 illustrates a factor analysis model with $\lambda = \binom{2}{1}$, $\mu = \binom{0}{0}$, $\Phi = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 2 \end{smallmatrix}\right)$

Factor analysis model can also be represented with a graphical model that's similar to Figure 2(a), except that the variable Z has no parameter and the CPD $P(X|Z)$ is a linear Gaussian

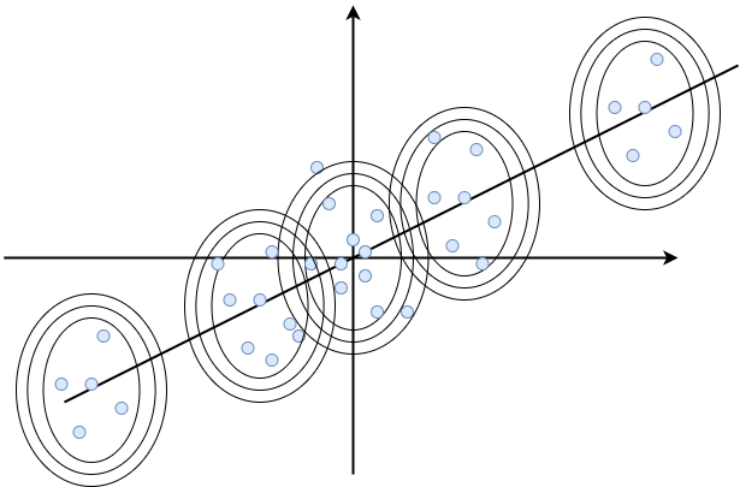


Figure 6: a factor analysis model with $\lambda = \binom{2}{1}$, $\mu = \binom{0}{0}$, $\Phi = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$

instead of a Gaussian.

In factor analysis, if we only observe X, we can apply EM algorithm to estimate λ, μ and Φ in a similar way as we did in mixture of Gaussian model. But in factor analysis, since Z is continuous, we need to replace a sum with integral. Let $\theta = \{\mu, \lambda, \Phi\}$ be the set of all parameters. The log-likelihood function can be written as

$$\begin{aligned}
l(\theta) &= \sum_{m=1}^{M} log(P(x[m];\theta)) \\
&= \sum_{m=1}^{M} log \int_{-\infty}^{+\infty} P(x[m], z[m];\theta) dz[m] \\
&= \sum_{m=1}^{M} log \int_{-\infty}^{+\infty} Q_m(z[m]) \frac{P(x[m], z[m];\theta)}{Q_m(z[m])} dz[m] \\
&= \sum_{m=1}^{M} log E_{Z[m]\sim Q_m} \left[\frac{P(x[m], Z[m];\theta)}{Q_m(Z[m])} \right] \\
&\geq \sum_{m=1}^{M} E_{Z[m]\sim Q_m} \left[log \frac{P(x[m], Z[m];\theta)}{Q_m(z[m])} \right] \\
&= \sum_{m=1}^{M} \int_{-\infty}^{+\infty} Q_m(z[m]) log \frac{P(x[m], z[m];\theta)}{Q_m(z[m])} dz[m] \\
&= g(\theta)
\end{aligned}$$

To derive the EM algorithm for factor analysis in a general form, we suppose $Z \in \mathbb{R}^{n_1}$, $X \sim \mathbb{R}^{n_2}$, $\lambda \in \mathbb{R}^{n_2 \times n_1}$, $\Phi \in \mathbb{R}^{n_2 \times n_2}$.

4.1 E-step

In E-step, we compute $Q_m(z[m]) = P(z[m]|x[m];\theta)$. We can infer this conditional distribution from the joint distribution. We can write the joint distribution as

$$\begin{aligned}
P(x[m], z[m];\theta) &= P(z[m];\theta)P(x[m]\big|z[m];\theta) \\
&= \frac{1}{(2\pi)^{n_1/2}} exp(-\frac{1}{2} z[m]^T z[m]) \\
&\quad \frac{1}{(2\pi)^{n_2/2}|\Phi|^{1/2}} exp(-\frac{1}{2}(x[m] - \lambda z[m] - \mu)^T \Phi^{-1}(x[m] - \lambda z[m] - \mu)) \\
&= \frac{1}{C} exp\Big[-\frac{1}{2} \begin{bmatrix} z[m] & x[m]-\mu \end{bmatrix} \begin{bmatrix} I+\lambda^T\Phi^{-1}\lambda & -\lambda^T\Phi^{-1} \\ -\Phi^{-1}\lambda & \Phi^{-1} \end{bmatrix} \begin{bmatrix} z[m] \\ x[m]-\mu \end{bmatrix} \Big]
\end{aligned}$$

Thus the joint distribution also has a Gaussian form. We can conclude that $\begin{bmatrix} z[m] \\ x[m] \end{bmatrix} \sim \mathcal{N}(\begin{bmatrix} 0 \\ \mu \end{bmatrix}, \Sigma)$, where $\Sigma = \begin{bmatrix} I+\lambda^T\Phi^{-1}\lambda & -\lambda^T\Phi^{-1} \\ -\Phi^{-1}\lambda & \Phi^{-1} \end{bmatrix}^{-1} = \begin{bmatrix} I & \lambda^T \\ \lambda & \Phi+\lambda\lambda^T \end{bmatrix}$. Thus the conditional distribution is a Gaussian, $(z[m]\big|x[m]) \sim \mathcal{N}(-\lambda^T(\Phi + \lambda\lambda^T)^{-1}(x[m] - \mu), I - \lambda^T(\Phi + \lambda\lambda^T)^{-1}\lambda)$.

Here we use the fact that

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^{-1} = \begin{bmatrix} (A_{11} - A_{12}A_{22}^{-1}A_{21})^{-1} & -(A_{11} - A_{12}A_{22}^{-1}A_{21})^{-1}A_{12}A_{22}^{-1} \\ -A_{22}^{-1}A21(A_{11} - A_{12}A_{22}^{-1}A_{21})^{-1} & A_{22}^{-1} + A_{22}^{-1}A_{21}(A_{11} - A_{12}A_{22}^{-1}A_{21})^{-1}A_{12}A_{22}^{-1} \end{bmatrix}$$

. and the conditional distribution of Gaussian. For more detail, please refer to my note on Gaussian distribution `http://lovinglavigne.com/PGM/gauss.pdf`

Now, in E-step, we compute $Q_m(z[m])$ as

$$Q_m(z[m]) = P(z[m]\big|x[m];\theta)$$

4.2 M-step

In M-step, we maximize $g(\theta)$ over the set of parameters. Thus we perform the optimization

$$\begin{aligned}
&max_\theta \sum_{m=1}^{M} \int_{-\infty}^{+\infty} Q_m(z[m]) log \frac{P(x[m], z[m];\theta)}{Q_m(z[m])} dz[m] \\
&= max_\theta \sum_{m=1}^{M} \int_{-\infty}^{+\infty} Q_m(z[m]) \big[logP(z[m];\theta) + logP(x[m]|z[m];\theta) - logQ_m(z[m]) \big] dz[m] \\
&= max_\theta \sum_{m=1}^{M} \int_{-\infty}^{+\infty} Q_m(z[m]) logP(x[m]|z[m];\theta) dz[m] \\
&= max_\theta \sum_{m=1}^{M} \int_{-\infty}^{+\infty} Q_m(z[m]) \big[log \frac{1}{(2\pi)^{n1/2}|\Phi|^{1/2}} - \frac{1}{2}(x[m] - \lambda z[m] - \mu)^T \Phi^{-1}(x[m] - \lambda z[m] - \mu) \big] dz[m]
\end{aligned}$$

- max over λ

$$\begin{aligned}
\nabla_\lambda &= \sum_{m=1}^{M} \int_{-\infty}^{\infty} Q_m(z[m]) \Phi^{-1}(x[m] - \lambda z[m] - \mu) z[m]^T dz[m] \\
&= \sum_{m=1}^{M} \int_{-\infty}^{\infty} Q_m(z[m]) \Phi^{-1}(x[m] - \mu) z[m]^T dz[m] - \sum_{m=1}^{M} \int_{-\infty}^{\infty} Q_m(z[m]) \Phi^{-1} \lambda z[m] z[m]^T dz[m] \\
&= \sum_{m=1}^{M} \Phi^{-1}(x[m] - \mu) \int_{-\infty}^{\infty} Q_m(z[m]) z[m]^T dz[m] - \sum_{m=1}^{M} \Phi^{-1} \lambda \int_{-\infty}^{\infty} Q_m(z[m]) z[m] z[m]^T dz[m] \\
&= \Phi^{-1} \sum_{m=1}^{M} (x[m] - \mu) E_{Z[m] \sim Q_m} \big[Z[m] \big]^T - \Phi^{-1} \lambda \sum_{m=1}^{M} E_{Z[m] \sim Q_m} \big[Z[m] Z[m]^T \big] := 0 \\
\lambda &= \Big(\sum_{m=1}^{M} (x[m] - \mu) E_{Z[m] \sim Q_m} \big[Z[m] \big]^T \Big) \Big(\sum_{m=1}^{M} E_{Z[m] \sim Q_m} \big[Z[m] Z[m]^T \big] \Big)^{-1} \qquad (4.2.1)
\end{aligned}$$

  Notice that $\sum_{m=1}^{M} E_{Z[m] \sim Q_m} \big[Z[m] Z[m]^T \big]$ is the sum of covariance matrices of gaussian distributions, thus is the sum of positive definite matrices and invertible.

- max over μ

$$\begin{aligned}
\nabla_\mu &= \sum_{m=1}^{M} \int_{-\infty}^{\infty} Q_m(z[m]) \Phi^{-1}(x[m] - \lambda z[m] - \mu) dz[m] \\
&= \Phi^{-1} \sum_{m=1}^{M} \int_{-\infty}^{\infty} Q_m(z[m]) (x[m] - \mu) dz[m] - \Phi^{-1} \lambda \sum_{m=1}^{M} \int_{-\infty}^{\infty} Q_m(z[m]) z[m] dz[m] \\
&= \Phi^{-1} \sum_{m=1}^{M} (x[m] - \mu) + \Phi^{-1} \lambda \sum_{m=1}^{M} \lambda^T (\Phi + \lambda \lambda^T)^{-1} (x[m] - \mu) \\
&= \Phi^{-1} (I + \lambda \lambda^T (\Phi + \lambda \lambda^T)^{-1}) \sum_{m-1}^{M} (x[m] - \mu) \\
&:= 0 \\
\mu &= \frac{x[m]}{M}
\end{aligned}$$

  Note that μ does not depend on the values of parameters.

- max over Φ

$$\begin{aligned}
& max_\Phi \sum_{m=1}^{M} \int_{-\infty}^{+\infty} Q_m(z[m]) \big[log \frac{1}{(2\pi)^{n1/2} |\Phi|^{1/2}} - \frac{1}{2} (x[m] - \lambda z[m] - \mu)^T \Phi^{-1} (x[m] - \lambda z[m] - \mu) \big] dz[m] \\
=& max_\Phi \sum_{m=1}^{M} \int_{-\infty}^{+\infty} Q_m(z[m]) \big[- \frac{1}{2} logdet(\Phi) - \frac{1}{2} (x[m] - \lambda z[m] - \mu)^T \Phi^{-1} (x[m] - \lambda z[m] - \mu) \big] dz[m] \\
=& min_\Phi \sum_{m=1}^{M} \int_{-\infty}^{+\infty} Q_m(z[m]) \big[logdet(\Phi) + tr\big(\Phi^{-1}, (x[m] - \lambda z[m] - \mu)(x[m] - \lambda z[m] - \mu)^T\big) \big] dz[m]
\end{aligned}$$

Let $J = \Phi^{-1}$, the optimization problem becomes

$$min_J \sum_{m=1}^{M} \int_{-\infty}^{+\infty} Q_m(z[m]) \big[- logdet(J) + tr\big(J, (x[m] - \lambda z[m] - \mu)(x[m] - \lambda z[m] - \mu)^T\big)\big] dz[m]$$

$$\begin{aligned}
\nabla_J &= \sum_{m=1}^{M} \int_{-\infty}^{+\infty} Q_m(z[m]) \big[- J^{-1} + (x[m] - \lambda z[m] - \mu)(x[m] - \lambda z[m] - \mu)^T\big] dz[m] \\
&= \sum_{m=1}^{M} -J^{-1} + \sum_{m=1}^{M} \int_{-\infty}^{+\infty} Q_m(z[m])(x[m] - \lambda z[m] - \mu)(x[m] - \lambda z[m] - \mu)^T dz[m] \\
&= -MJ^{-1} \\
&\quad + \sum_{m=1}^{M} \int_{-\infty}^{+\infty} Q_m(z[m]) \big[\lambda z[m] z[m]^T \lambda^T - 2\lambda z[m](x[m] - \mu)^T + (x[m] - \mu)(x[m] - \mu)^T\big] dz[m] \\
&= -MJ^{-1} \\
&\quad + \sum_{m=1}^{M} \lambda E_{Z[m]\sim Q_m}[Z[m]Z[m]^T] - 2\lambda E_{Z[m]\sim Q_m}[Z[m]](x[m] - \mu) + (x[m] - \mu)(x[m] - \mu)^T
\end{aligned}$$

$$\Phi = J^{-1} = \frac{1}{M} \sum_{m=1}^{M} \lambda E_{Z[m]\sim Q_m}[Z[m]Z[m]^T] - 2\lambda E_{Z[m]\sim Q_m}[Z[m]](x[m] - \mu) + (x[m] - \mu)(x[m] - \mu)^T$$

Note that in the derivation of M-step, $E_{Z[m]\sim Q_m}[Z[m]]$ and $E_{Z[m]\sim Q_m}[Z[m]Z[m]^T]$ are the expectation and covariance matrix of the conditional Gaussian distribution $(z[m]\big|x[m]) \sim \mathcal{N}(-\lambda^T(\Phi + \lambda\lambda^T)^{-1}(x[m] - \mu), I - \lambda^T(\Phi + \lambda\lambda^T)^{-1}\lambda)$. Thus $E_{Z[m]\sim Q_m}[Z[m]] = -\lambda^T(\Phi + \lambda\lambda^T)^{-1}(x[m] - \mu)$, $E_{Z[m]\sim Q_m}[Z[m]Z[m]^T] = I - \lambda^T(\Phi + \lambda\lambda^T)^{-1}\lambda$

Now we can conclude the EM algorithm for factor analysis model

Algorithm 2: Algorithms for solving mixture of Gaussian

$\mu^{(t+1)} = \frac{x[m]}{M}$

for *t = 0,1,...,until convergence* **do**

E-step:

$$E_{Z[m]\sim Q_m}[Z[m]] = -\lambda^{(t)T}(\Phi^{(t)} + \lambda^{(t)}\lambda^{(t)T})^{-1}(x[m] - \mu)$$
$$E_{Z[m]\sim Q_m}[Z[m]Z[m]^T] = I - \lambda^{(t)T}(\Phi^{(t)} + \lambda^{(t)}\lambda^{(t)T})^{-1}\lambda^{(t)}$$

M-step:

$$\lambda^{(t+1)} = \big(\sum_{m=1}^{M}(x[m] - \mu^t)E_{Z[m]\sim Q_m}\big[Z[m]\big]^T\big)\big(\sum_{m=1}^{M} E_{Z[m]\sim Q_m}\big[Z[m]Z[m]^T\big]\big)^{-1}$$

$$\begin{aligned}
\Phi^{(t+1)} = \frac{1}{M} \sum_{m=1}^{M} &\lambda E_{Z[m]\sim Q_m}[Z[m]Z[m]^T] - 2\lambda E_{Z[m]\sim Q_m}[Z[m]](x[m] - \mu) \\
&+ (x[m] - \mu)(x[m] - \mu)^T
\end{aligned}$$

if *convergence* **then**

return μ, $\lambda^{(t+1)}$, $\Phi^{(t+1)}$

end

end

5 EM for Bayesian Network

Having the theoretical foundation for EM algorithm and having derived the EM algorithm for two specific bayesian network, in this section we apply EM algorithm to general bayesian networks to learn from partially observed data.

Suppose that we have a set of observed data $\mathcal{D} = \{(o[m], Z[m])\forall m = 1, ..., M\}$ where $o[m]$ are instantiation of observed values in the mth data and $Z[m]$ are unobserved random variables in the mth data. For any $z[m] \in val(Z[m])$, $(o[m], z[m])$ is a full instantiation. And use the notation $(x[m], o[m]) < X_i >$ to indicate the value of random variable X_i in the full instantiation $(o[m], z[m])$. And let θ be the variables for CPDs in the bayesian network, and $\theta_{X|u}$ be the parameter for the CPD $P(X|U = u)$ where U are the parents of X in the Bayesian network.

We can write the log-likelihood of the bayesian network as

$$\begin{aligned}
l(\theta; \mathcal{D}) &= \sum_{m=1}^{M} logP(o[m]; \theta) \\
&= \sum_{m=1}^{M} log \sum_{z[m] \in val(Z[m])} P(o[m], z[m]; \theta) \\
&= \sum_{m=1}^{M} log \sum_{z[m] \in val(Z[m])} Q_m(z[m]) \frac{P(o[m], z[m]; \theta)}{Q_m(z[m])} \\
&= \sum_{m=1}^{M} \sum_{z[m] \in val(Z[m])} Q_m(z[m]) log \frac{P(o[m], z[m]; \theta)}{Q_m(z[m])} \\
&= \sum_{m=1}^{M} \sum_{z[m] \in val(Z[m])} Q_m(z[m]) \Bigg[\sum_{(U \to X) \in \mathcal{G}} logP((o[m], z[m]) < X > \big| (o[m], z[m]) < U >) - logQ_m(z[m]) \Bigg] \\
&= g(\theta)
\end{aligned}$$

In E-step, we compute $Q_m(z[m]) = P(z[m]|o[m]; \theta)$.

5.1 M-step

In M-step, we maximize the likelihood of the dataset on the bayesian network. $g(\theta)$ is the sum of log functions over parameters of different CPDs, Thus we can decompose $g(\theta)$ into functions into parameters for each CPD.

For each X in the scope of Bayesian network $\mathcal{G}$ and for each $u \in val(Parent(X))$, we solve the optimization problem.

$$\begin{aligned}
&Find \quad \theta_{x|u} \forall x \in val(X) \\
&min - \sum_{m=1}^{M} \sum_{z[m]} Q_m(z[m]) logP((o[m], z[m]) < X > \big| (o[m], z[m]) < U >) \\
&s.t. \sum_{x \in val(X)} \theta_{x|u} = 1 \quad \text{(dual variable } v\text{)}
\end{aligned}$$

We solve this optimization problem with Lagrangian multiplier.

$$\begin{aligned}
\mathcal{L}(\theta_{X|u}, v) &= -\sum_{m=1}^{M}\sum_{z[m]} Q_m(z[m]) log P((o[m], z[m]) < X > \big|(o[m], z[m]) < U >) + v(\sum_x \theta_{x|u} - 1) \\
&= -\sum_{x \in val(X)} log\theta_{x|u} \sum_{m=1}^{M}\sum_{z[m]} Q_m(z[m]) 1\{o[m], z[m]) < X, U >= (x, u)\} + v(\sum_x \theta_{x|u} - 1) \\
\nabla_{\theta_{x|u}}\mathcal{L} &= -\frac{1}{\theta_{x|u}} \sum_{m=1}^{M}\sum_{z[m]} Q_m(z[m]) 1\{o[m], z[m]) < X, U >= (x, u)\} + v := 0 \\
\theta_{x|u} &= \frac{1}{v} \sum_{m=1}^{M}\sum_{z[m]} Q_m(z[m]) 1\{o[m], z[m]) < X, U >= (x, u)\}
\end{aligned}$$

Let $h(x, v) = \sum_{m=1}^{M}\sum_{z[m]} Q_m(z[m]) 1\big\{o[m], z[m]) < X, U >= (x, u)\big\}$, then $\theta_{x|u} = \frac{1}{v} h(x, u)$. The dual function is

$$\begin{aligned}
g(v) &= \inf_{\theta_{X|u}} \mathcal{L}(\theta_{X|u}, v) \\
&= -\sum_x log(\frac{h(x,u)}{v}) h(x, u) + v(\sum_x \frac{h(x,u)}{v} - 1) \\
&= -\sum_x log(\frac{h(x,u)}{v}) h(x, u) + \sum_x h(x, u) - v \\
\nabla_v g &= \frac{1}{v}\sum_x h(x, u) - 1 := 0 \\
v^* &= \sum_x h(x, u)
\end{aligned}$$

Then the solution for the optimization problem is

$$\begin{aligned}
\theta_{x|u} &= \frac{h(x,u)}{\sum_x h(x,u)} \\
&= \frac{\sum_{m=1}^{M}\sum_{z[m]} Q_m(z[m]) 1\big\{(o[m], z[m]) < X, U >= (x, u)\big\}}{\sum_x \sum_{m=1}^{M}\sum_{z[m]} Q_m(z[m]) 1\big\{(o[m], z[m]) < X, U >= (x, u)\big\}} \\
&= \frac{\sum_{m=1}^{M}\sum_{z[m]} P(z[m]|o[m];\theta) 1\big\{(o[m], z[m]) < X, U >= (x, u)\big\}}{\sum_x \sum_{m=1}^{M}\sum_{z[m]} P(z[m]|o[m];\theta) 1\big\{(o[m], z[m]) < X, U >= (x, u)\big\}} \\
&= \frac{\sum_{m=1}^{M} P(x, u\big|o[m])}{\sum_x \sum_{m=1}^{M} P(x, u\big|o[m])}
\end{aligned}$$

We can see that we do not have to explicitly compute $Q_m(z[m])$, instead, we only have to compute $P(x, u\big|o[m]) \forall (U \rightarrow X) \in \mathcal{G}, \forall (x, u) \in val(X, U)$, which can be done with standard inference techniques for bayesian networks, for example, belief propagation and Gibbs sampling. Algorithm 3 and 4 describe the method to apply EM algorithm to Bayesian network.

Algorithm 3: Compute-Marginal-Q

Input: $\mathcal{G}$: Bayesian network, θ:current parameter, $\mathcal{D}$:partially observed data
for $(U \to X) \in \mathcal{G}$ **do**
  for $(x, u) \in val(X, U)$ **do**
    Q(x,u) = 0;
  end
end
for *m=1,...,M* **do**
  Run inference on $\mathcal{G}$ with evidence $o[m]$
  for $(U \to X) \in \mathcal{G}$ **do**
    for $(x, u) \in val(X, U)$ **do**
      Q(x,u) += P(x,u | o[m])
    end
  end
end
return Q

Algorithm 4: EM-Bayesian-Network

Input: $\mathcal{G}$: Bayesian network, $\theta^{(0)}$:initial parameter, $\mathcal{D}$:partially observed data
for *t = 0,1,... until convergence* **do**
  Q = Compute-Marginal-Q($\mathcal{G}$, $\theta^{(t)}$, $\mathcal{D}$)
  for $(U \to X) \in \mathcal{G}$ **do**
    for $(x, u) \in val(X, U)$ **do**
      $\theta_{x|u}^{(t+1)} = \frac{Q(x,u)}{\sum_x Q(x,u)}$
    end
  end
end
return $\theta^{(t)}$

Till now, hopefully I have made the derivation of EM algorithm clear to you. Thank you for reading this!

6 Appendix A: Log-determinant

6.1 Concavity of Logdet

We use the concavity of the line restriction to prove the concavity of $lodget(X)$ for positive definite X. The line restriction is

$$\begin{aligned} logdet(X + \alpha\Delta X) &= logdet\Big(X^{1/2}(I + \alpha X^{-1/2}\Delta X X^{-1/2})X^{1/2}\Big) \\ &= logdet(X) + logdet(I + \alpha X^{-1/2}\Delta X X^{-1/2}) \\ &= loget(X) + logdet(I + \alpha U\Lambda U^T) \\ &= logdet(X) + \sum_i log(1 + \alpha\lambda_i) \end{aligned}$$

Here we perform eigenvalue decomposition, so that $X^{-1/2}\Delta X X^{-1/2} = U\Lambda U^T$ and λ_i is the ith diagonal element of Λ (i.e the ith eigenvalue of $X^{-1/2}\Delta X X^{-1/2}$). Since the line restriction $logdet(X + \alpha\Delta X)$ is concave in α, $logdet$ is a concave function on positive definite space.

6.2 Derivative of Log-determinant

We continue the discussion from the previous section

$$\begin{aligned} logdet(X + \Delta X) &= logdet(X) + \sum_i log(1 + \lambda_i) \\ &\simeq logdet(X) + \sum_i \lambda_i \\ &= logdet(X) + tr(X^{-1/2}\Delta X X^{-1/2}) \\ &= logdet(X) + tr(X^{-1}, \Delta X) \end{aligned}$$

Thus $\nabla_X logdet(X) = X^{-1}$. At the "approximately equal sign" we used the fact that $\lim_{\lambda_i \to 0} log(1+\lambda_i) = \lambda_i$.