**Stats 201C: Lecture Notes**

Melody Y. Huang

# 1. EM Algorithm

## 1.1. The Incomplete Data Problem

Let $Y$ be an $n \times p$ matrix of data. Denote $y_i$ as the $i$-th row of $Y$ (where $i = 1, ..., n$):

$$y_i = (y_{i1}, ..., y_{ip})$$

We can think of $y_i$ as a realization of a $p$-dimensional random variable. Each individual row of $Y$ has the pdf of $f(y_i|\theta)$. Assuming independence, this means that the pdf of $Y$ (or the likelihood) is:

$$P(Y|\theta) = \prod_{i=1}^{n} f(y_i|\theta)$$

In the case for which we have complete data, this is relatively straightforward to compute. However, when we cannot observe all of $Y$, this becomes more complicated. To compute the densities for rows with missing values, we must integrate over all the missing values to obtain the marginal distributions. More concretely, if in the $j$-th row, the second element is missing:

$$y_j = (y_{j1}, ?, y_{j3}, ...y_{jp})$$

Then the marginal distribution will be:

$$f(y_{j1}, y_{j3}, ...y_{jp}|\theta) = \int f(y_j|\theta) dy_{j2}$$

With different rows having different missing values, the pdf of $Y$ becomes a bunch of integrals.

To bypass this, we want to perform inference of $\theta$ over only the *observed* data. In other words, we want to compute $P(Y_{obs}|\theta)$ instead of $P(Y|\theta)$, where $Y$ is written as:

$$Y = \begin{pmatrix} Y_{obs} & Y_{mis} \end{pmatrix}$$

(As such, the likelihood function can be rewritten as: $L(\theta|Y) = L(\theta|Y_{obs}, Y_{mis})$.) In order for this to be valid, we need to establish **ignorability**.

## 1.2. Ignorability Assumptions

### Assumption 1: Missing at Random (MAR)

To begin, let $R_{ij}$ be an indicator variable that denotes whether or not $y_{ij}$ is missing:

$$R_{ij} = \begin{cases} 1 & \text{if } y_{ij} \text{ is observed} \\ 0 & \text{if } y_{ij} \text{ is missing} \end{cases}$$

The matrix containing all of these $R_{ij}$ values will be denoted as $R$. We can construct a probability model for $R$, which in essence represents the probability that some value in $Y$ is missing:

$$P(R|Y, \xi),$$

where $\xi$ are unknown parameters.

The MAR assumption is therefore:

$$P(R|Y, \xi) = P(R|Y_{obs}, \xi)$$

Intuitively, what this means is that the missing values of $Y$ do not tell us any information about the probability of being a missing value. For example, a violation of this would be if all the values of $Y$ below a certain threshold were missing.

**Assumption 2: Distinctness of Parameters** $(\theta, \xi)$

Recall that the data model $P(Y|\theta)$ has parameters represented by $\theta$, while the missing value model $P(R|Y, \xi)$ has (unknown) parameters $\xi$. $\theta$ and $\xi$ must be distinct in order for us to claim ignorability. Distinctness can be defined in two different ways:

1. *(Frequentist Approach).* The joint parameter space of $(\theta, \xi)$ is the Cartesian product of the individual parameter spaces for $\theta$ and $\xi$. In two dimensions, this means that if we plotted the parameter spaces, we would have a rectangle. Essentially, this implies that different values of $\theta$ does not correspond to varying values of $\xi$. (See plots from class.)

2. *(Bayesian Approach).* Any joint prior on $(\theta, \xi)$ must factor into independent marginal priors for $\theta$ and $\xi$. In other words, defining the joint prior as $\pi(\theta, \xi)$:

   $$\pi(\theta, \xi) = \pi_\theta(\theta) \cdot \pi_\xi(\xi)$$

   This is a slightly stronger assumption than the frequentist assumption (from (1)).

When both of these assumptions are met, then we can claim that the missing data mechanism is **ignorable**.

With ignorability, we can compute the likelihood and posterior function for the observed data:

1. **Likelihood of $Y_{obs}$:**
   We begin by looking at the joint density function between $R$ and $Y_{obs}$: $P(R, Y_{obs}|\theta, \xi)$. Recall that in order to obtain this, we simply integrate over all the missing values $Y$:

   $$P(R, Y_{obs}|\theta, \xi) = \int P(R, Y|\theta, \xi) dY_{mis}$$

   Now note that: $P(Y, R|\theta, \xi) = P(Y|\theta) \cdot P(R|Y, \xi)$. Therefore, we may rewrite the above as:

   $$\begin{aligned} P(R, Y_{obs}|\theta, \xi) &= \int P(R, Y|\theta, \xi) dY_{mis} \\ &= \int P(R|Y, \xi) \cdot P(Y|\theta) dY_{mis} \end{aligned}$$

   By MAR, $P(R|Y, \xi) = P(R|Y_{obs}, \xi)$, and $P(R|Y_{obs}) \perp\!\!\!\perp Y_{mis}$:

   $$\begin{aligned} &= P(R|Y_{obs}, \xi) \cdot \int P(Y|\theta) dY_{mis} \\ &= P(R|Y_{obs}, \xi) \cdot P(Y_{obs}|\theta) \end{aligned}$$

   Under distinctness, we can then claim that:

   $$L(\theta|Y_{obs}) \propto P(Y_{obs}|\theta)$$

2. **Posterior of observed data ($P(\theta|Y_{obs})$):**
   To begin, we first estimate $P(\theta, \xi|Y_{obs}, R)$:

   $$\begin{aligned} P(\theta, \xi|Y_{obs}, R) &\propto P(R, Y_{obs}|\theta, \xi) \cdot \pi(\theta, \xi) \\ &\text{By MAR:} \\ &= P(R|Y_{obs}, \xi) P(Y_{obs}|\theta) \cdot \pi(\theta, \xi) \\ &\text{By distinctness:} \\ &= P(R|Y_{obs}, \xi) P(Y_{obs}|\theta) \cdot \pi_\theta(\theta) \cdot \pi_\xi(\xi) \end{aligned}$$

   Therefore, it follows naturally that:

   $$\begin{aligned} P(\theta|Y_{obs}, R) &= \int P(\theta, \xi|Y_{obs}, R) d\xi \\ &\propto P(Y_{obs}|\theta) \cdot \pi_\theta(\theta) \cdot \int P(R|Y_{obs}, \xi) \cdot \pi_\xi(\xi) d\xi \\ &\propto L(\theta|Y_{obs}) \cdot \pi_\theta(\theta) \end{aligned}$$

## 1.3. EM Algorithm

So far we have talked about the issues that missing data presents, and the conditions that are necessary in order for us to estimate the likelihood of the data generating process, given only observed values. We assume now that the conditions are met. This means that we may now compute the maximum likelihood estimator:

$$\hat{\theta}_{MLE} = \underset{\theta}{\operatorname{argmax}} P(Y_{obs}|\theta) = \underset{\theta}{\operatorname{argmax}} \int P(Y_{obs}, Y_{mis}|\theta) dY_{mis}$$

In the usual setting in which we would have complete data $Y$, we could simply take the derivative of the likelihood (or log-likelihood) function and set it equal to zero to solve for the parameters. However, this can be impossible to do with missing data.

Therefore, we use the EM algorithm to iteratively estimate the value of $\hat{\theta}_{MLE}$ of the observed data.

**Definition 1.1 (Expectation Maximization Algorithm)**

*E-Step. Calculate Q (the expectation function).*

$$Q(\theta|\theta^{(t)}) = \mathbb{E}\left[\log P(Y_{obs}, Y_{mis}|\theta) \mid \theta^{(t)}, Y_{obs}\right]$$

*It should be noted that this is simply the expectation of the log-likelihood of the complete data (i.e., $\ell(\theta|Y)$), conditioned on our inputted value of $\theta^{(t)}$ and the observed data $Y_{obs}$. We simply*

*M-Step. Maximize Q in order to compute $\theta^{(t+1)}$.*

$$\theta^{(t+1)} = \underset{\theta}{\operatorname{argmax}} Q(\theta|\, \theta^{(t+1)})$$

Each iteration of the EM algorithm will lead to an increase in the likelihood. More formally:

$$\ell(\theta^{(t)}|Y_{obs}) \leq \ell(\theta^{(t+1)}|Y_{obs})$$

We can mathematically show this. To begin, we can express our $Q$ function as a combination of the likelihood function (i.e., the objective function) and an entropy-like term:

$$\begin{aligned}
Q(\theta|\theta^{(t)}) &= \int \log\left(P(Y_{obs}, Y_{mis}|\theta)\right) \cdot P(Y_{mis}|Y_{obs}, \theta^{(t)}) dY_{mis} \\
&= \int \log\left(P(Y_{obs}|\theta) \cdot P(Y_{mis}|Y_{obs}, \theta)\right) \cdot P(Y_{mis}|Y_{obs}, \theta^{(t)}) dY_{mis} \\
&= \int \left[\log\left(P(Y_{obs}|\theta)\right) + \log\left(P(Y_{mis}|Y_{obs}, \theta)\right)\right] \cdot P(Y_{mis}|Y_{obs}, \theta^{(t)}) dY_{mis} \\
&= \underbrace{\log\left(P(Y_{obs}|\theta)\right)}_{\ell(\theta|Y_{obs})} + \underbrace{\int \log\left(P(Y_{mis}|Y_{obs}, \theta)\right) P(Y_{mis}|Y_{obs}, \theta^{(t)}) dY_{mis}}_{H(\theta|\theta^{(t)})}
\end{aligned}$$

Therefore, we can rewrite the likelihood as:

$$\ell(\theta|Y_{obs}) = Q(\theta|\theta^{(t)}) - H(\theta|\theta^{(t)})$$

Taking the difference betweeen $\ell(\theta^{(t+1)}|Y_{obs})$ and $\ell(\theta^{(t)}|Y_{obs})$:

$$\ell(\theta^{(t+1)}|Y_{obs}) - \ell(\theta^{(t)}|Y_{obs}) = \underbrace{Q(\theta|\theta^{(t+1)}) - Q(\theta|\theta^{(t)})}_{(1)} + \underbrace{\Big(H(\theta^{(t)}|\theta^{(t)}) - H(\theta^{(t+1)}|\theta^{(t)})\Big)}_{(2)}$$

Term (1) should always be non-negative, by definition of how $Q$ is computed. Because at each iteration $t$, $\theta^{(t+1)}$ is picked in the $M$-step to maximize $Q$, $Q$ should be increasing between iterations. For the entropy term:[1]

$$\begin{aligned} H(\theta^{(t)}|\theta^{(t)}) - H(\theta^{(t+1)}|\theta^{(t)}) &= \int \log\left(P(Y_{mis}|Y_{obs}, \theta^{(t)})\right) P(Y_{mis}|Y_{obs}, \theta^{(t)}) dY_{mis} \\ &\quad - \int \log\left(P(Y_{mis}|Y_{obs}, \theta^{(t+1)})\right) P(Y_{mis}|Y_{obs}, \theta^{(t)}) dY_{mis} \\ &= \int -\log\left(\frac{P(Y_{mis}|Y_{obs}, \theta^{(t+1)})}{P(Y_{mis}|Y_{obs}, \theta^{(t)})}\right) \cdot P(Y_{mis}|Y_{obs}, \theta^{(t)}) dY_{mis} \\ &\equiv KL\left(P(Y_{mis}|Y_{obs}, \theta^{(t)}) || P(Y_{mis}|Y_{obs}, \theta^{(t+1)})\right) \geq 0 \end{aligned}$$

This is equivalent to the KL divergence, which is strictly non-negative.

As such:

$$\ell(\theta^{(t+1)}|Y_{obs}) - \ell(\theta^{(t)}|Y_{obs}) \geq 0$$

It should be noted that while we are iteratively increasing the likelihood using EM, this does not guarantee that we will converge to the maximum likelihood estimator.

## 2. Extensions of EM

We mostly discuss two different extensions of EM. One is a Bayesian treatment of expectation maximization, in which we opt to maximize the posterior distribution instead of the observed data likelihood. Another extension is Variational EM, in which we pose EM as an iterative maximization problem that can then be modified by imposing certain restrictions on how we opt to maximize. These two are discussed in more detail below.

### 2.1. Gibbs Sampling

A Bayesian approach to expectation maximization is to treat our parameter as a random variable, and sample from the distribution of potential $\theta$ values in order to estimate the most likely value $\theta$

[1] The entropy term is slightly confusing notationally, but in essence, we are comparing taking the expectation of $\log P(Y_{mis}|Y_{obs}, \theta)$ with respect to the density function specified with respect to some $\theta^{(t)}$ at time $t$.

would be. Therefore, in the Bayesian setting, we are trying to **maximize the posterior distribution**, over the observed data ($P(\theta|Y_{obs})$ (whereas within EM, we are simply trying to maximize the observed data likelihood $P(Y_{obs}|\theta)$). As it turns out, using our very profound knowledge and understanding of Bayes Rule, we will recall that the observed data likelihood is a component of the posterior distribution (see the blue term below), and that the main difference is that we are now including a prior term into our estimate of $\theta$.

$$P(\theta|Y_{obs}) \propto P(\theta)P(Y_{obs}|\theta)$$

We can factor our posterior distribution into a function between the posterior distribution of $\theta$, with respect to the complete data, and the conditional distribution of the missing data, given the observed data. More specifically:[2]

$$\begin{aligned} P(\theta|Y_{obs}) &= \int P(\theta, Y_{mis}|Y_{obs})dY_{mis} \\ &= \int P(\theta|Y_{mis}, Y_{obs}) \cdot P(Y_{mis}|Y_{obs})dY_{mis} \\ &= \int P(\theta|Y) \cdot P(Y_{mis}|Y_{obs})dY_{mis} \end{aligned}$$

**Definition 2.1 (Two Block Gibbs Sampler)**

*Step 1.* *Given $\theta^{(t)}$, sample $Y_{mis}^{(t+1)} \sim P(Y_{mis}|Y_{obs}, \theta^{(t)})$.*

*Step 2.* *Given $Y_{mis}^{(t+1)}$, sample $\theta^{(t+1)} \sim \underbrace{P(\theta|Y_{obs}, Y_{mis}^{(t+1)})}_{=P(\theta|Y^{(t+1)})}$*

*We can think about this as iteratively sampling from the marginal distributions of the joint distribution of $P(Y_{mis}, \theta|Y_{obs})$. Alternatively, we can think of this as iteratively estimating new values for the missing data.*

[2] This uses a trick that we often use in this class, which is write our observed probability density functions by summing over all the missing variables (s.t. the observed density is effectively cast as a marginal representation of the entire "complete" data).

## 2.2. Variational EM

### 2.2.1. Writing EM as a Maximization Problem

We begin by rewriting the (observed) log-likelihood:

$$\begin{aligned}\ell(\theta|Y) &= \log P(Y;\theta)\\ &= \log\left(\frac{P(Y,Z;\theta)}{P(Z|Y;\theta)}\right)\\ &= \log P(Y,Z;\theta) - \log P(Z|Y;\theta)\end{aligned}$$

We now introduce some arbitrary function $F(Z)$, defined with respect to the hidden/missing variable $Z$. We take the expectation of both sides with respect to $F$. This does nothing to change the left hand side, because there is no dependency on $F$ or $Z$: $\mathbb{E}_F(\ell(\theta|Y)) = \ell(\theta|Y)$. Therefore:

$$\begin{aligned}\ell(\theta|Y) &= \mathbb{E}_F(\log P(Y,Z;\theta)) - \mathbb{E}_F(\log P(Z|Y;\theta))\\ &= \mathbb{E}_F(\log P(Y,Z;\theta)) - \mathbb{E}_F(\log P(Z|Y;\theta)) + \mathbb{E}_F(\log F(Z)) - \mathbb{E}_F(\log F(Z))\\ &= \mathbb{E}_F(\log P(Y,Z;\theta)) - \mathbb{E}_F(\log F(Z)) - \mathbb{E}_F\left(\log\frac{P(Z|Y;\theta)}{F(Z)}\right)\\ &= E_F(\log P(Y,Z;\theta)) - \mathbb{E}_F(\log F(Z)) + KL(F||P(Z|Y;\theta))\\ &= E_F(\log P(Y,Z;\theta)) + \underbrace{\mathbb{E}_F(-\log F(Z))}_{\equiv H(F)} + KL(F||P(Z|Y;\theta))\\ &= \underbrace{E_F(\log P(Y,Z;\theta)) + H(F)}_{:=L(\theta,F)} + KL(F||P(Z|Y;\theta))\end{aligned}$$

As such, the likelihood can be written as a combination of some function that lower bounds it, and the KL divergence:

$$\ell(\theta|Y) = L(\theta,F) + KL(F||P(Z|Y;\theta))$$

Because the KL divergence is strictly non-negative, this implies:

$$\implies \ell(\theta|Y) \geq L(\theta,F)$$

This is cool, because this holds for all $F$'s!

Now, let's think about this in the context of the EM algorithm. In the $E$-step, we hold some $\theta^{(t)}$ value constant. Instead of imputing values into the expectation of the complete data likelihood, we now are trying to maximize the lower bound function. In other words:

$$\max_F L(\theta^{(t)},F)$$

Because $\theta^{(t)}$ is constant at this step, this means that the likelihood $\ell(\theta^{(t)}|Y)$ is fixed (since $Y$ is also fixed). Note that:

$$L(\theta^{(t)},F) = \underbrace{\ell(\theta^{(t)}|Y)}_{\text{Fixed!}} - KL(F||P(Z|Y;\theta^{(t)}))$$

In order to maximize the lower bound, we must *minimize* the KL divergence between $F$ and $P(Z|Y;\theta)$. The KL divergence has a lower bound of 0, which is achieved only when the two functions are equivalent to one another. As such, we must set $F$ equal to $P(Z|Y;\theta^{(t)})$ in order to minimize it:

$$F^{(t)} = P(Z|Y;\theta^{(t)})$$

In the $M$-step, we now hold $F^{(t)}$ fixed, and maximize $L(\theta, F^{(t)})$ with respect to $\theta$. If we actually do the math, supposedly, this works out to be equivalent to what we do in regular EM.

To summarize:

1. E-Step. Given $\theta^{(t)}$: $F^{(t)} = P(Z|Y;\theta^{(t)})$
2. M-Step. Given $F^{(t)}$: $\theta^{(t+1)} = \underset{\theta}{\operatorname{argmax}}\, L(\theta, F^{(t)})$

### 2.2.2. Restricting $F$

When there are complex dependence structures, there may not be a closed form representation of $P(Z|Y;\theta^{(t)})$, and as such, the problem of minimizing the KL divergence between $F$ and $P(Z|Y;\theta)$ can become computationally intractable. As such, Variational EM effectively *restricts the function class that $F$ may be*, such that the problem becomes easier to solve. Often times, this requires imposing some sort of assumptions into the structure of the hidden/missing variables (i.e., independence).

## 3. Mixture Models

Mixture models are a model-based approach to clustering. Assume that we want to model some $y$, where $y = (y_1, ..., y_k)$. In other words, $y$ is a mixture of $k$ different components. The density of $y$ would be written as:

$$P(y|\theta,\lambda) = \sum_{m=1}^{k} \lambda_m f(y_i|\theta_m),$$

where $\lambda_m$ is the weight of each mixture component. Each mixture can be parameterized differently (as given by the $\theta_m$).

From this alone, it can be difficult to maximize $f(y \mid \theta, \lambda)$. Instead, we can reframe this as a missing data problem, and apply EM to derive our parameter estimates.

## 3.1. Missing Data Problem

We introduce an indicator variable that denotes whether or not a particular $y_i$ is drawn from a particular mixture component:

$$Z_{im} = \begin{cases} 1 & \text{if } y_i \text{ is from the } m\text{-th mixture} \\ 0 & \text{else} \end{cases}$$

Therefore, for each $y_i$, we have a vector $Z_i = (Z_{i1}, ..., Z_{ik})$ associated with it that denotes the mixture membership. $Z_i$ is simply a Multinomial distribution:

$$Z_i \sim \text{Multinomial}(1, (\lambda_1, ..., \lambda_k))$$

Therefore, we now have a joint model $P(Z_i, y_i)$. The complete data likelihood is therefore:

$$\begin{aligned} P(Z, Y|\theta) &= P(Z)P(Y|Z) \\ &= \prod_{i=1}^{n} \prod_{m=1}^{k} (\lambda_m f(y_i|\theta_m))^{Z_im} \end{aligned}$$

As such, the complete data log-likelihood is:

$$\log P(Z, Y|\theta) = \sum_{i=1}^{n} \sum_{m=1}^{k} Z_{im} \left(\log \lambda_m + \log f(y_i|\theta_m)\right)$$

## 3.2. Applying EM

We can now solve for our parameter estimates using EM. Taking the expectation across the complete data log-likelihood:

$$\mathbb{E}(\log P(Z, Y|\theta)) = \sum_{i=1}^{n} \sum_{m=1}^{k} \mathbb{E}(Z_{im}) \left(\log \lambda_m + \log f(y_i|\theta_m)\right)$$

The expectation of $Z_{im}$ is equivalent to $P(Z_{im} = 1|y, \theta^{(t)}, \lambda^{(t)})$:

$$\begin{aligned} P(Z_{im} = 1|y, \theta^{(t)}, \lambda^{(t)}) &= \frac{P(y_i|Z_{im} = 1, \theta_m^{(t)}) \cdot P(Z_{im} = 1|\lambda^{(t)})}{\sum_{j=1}^{k} P(y_i|Z_{ij} = 1, \theta_j^{(t)}) \cdot P(Z_{ij} = 1|\lambda^{(t)})} \\ &= \frac{f(y_i|\theta_m^{(t)}) \cdot \lambda_m^{(t)}}{\sum_{j=1}^{k} f(y_i|\theta_j^{(t)})\lambda_j^{(t)}} \\ &\equiv w_{im}^{(t)} \end{aligned}$$

Therefore, we can think of this as the weight of the $m$-th mixture that makes up $y_i$. Using the weight notation, we can express the expectation as:

$$\mathbb{E}(\log P(Z, Y|\theta)) = \sum_{i=1}^{n} \sum_{m=1}^{k} w_{im} \left(\log \lambda_m + \log f(y_i|\theta_m)\right)$$

We can also write this as:

$$\begin{aligned}\mathbb{E}(\log P(Z, Y|\theta)) &= \sum_{i=1}^{n} \sum_{m=1}^{k} w_{im} \left(\log \lambda_m + \log f(y_i|\theta_m)\right) \\ &= \sum_{i=1}^{n} \sum_{m=1}^{k} w_{im} \log \lambda_m + \sum_{i=1}^{n} \sum_{m=1}^{k} w_{im} \log f(y_i|\theta_m) \\ &= \sum_{m=1}^{k} w_{\cdot m} \log \lambda_m + \sum_{i=1}^{n} \sum_{m=1}^{k} w_{im} \log f(y_i|\theta_m)\end{aligned}$$

This expression of the expectation is helpful when thinking about the $M$ step, because when optimizing for $\theta$, we can simply ignore the first term.

In the $M$-step, we have to update the parameter estimates for $\lambda$ and $\theta$. As such:

$$\hat{\lambda}_m^{(t+1)} = \frac{w_{\cdot m}^{(t)}}{\sum_{i=1}^{n} \sum_{m=1}^{k} w_{im}^{(t)}} = \frac{1}{n} w_{\cdot m}^{(t)}$$

$$\hat{\theta}_m^{(t+1)} = \underset{\theta}{\operatorname{argmax}} \left( \sum_{i=1}^{n} w_{im}^{(t)} \log f(y_i|\theta_m) \right)$$

## 3.3. Connection to $K$-Means Clustering

We can map mixture models to $k$-means clustering to show that $k$-means is really just a type of mixture model. In particular, we make the following assumptions:

- $y_i|Z_{im} = 1 \sim N_p(\mu_m, \sigma^2 I_p)$
- Clusters are well-separated and spherical in nature

Therefore, breaking this down into the $E$ and $M$ steps:

- $E$-Step:

$$w_{im} = \frac{\lambda_m f(y_i|\mu_m, \Sigma_m)}{\sum_j \lambda_j f(y_i|\mu_j, \Sigma_j)} = \frac{\lambda_m \exp(-\frac{1}{2\sigma^2}||y_i - \mu_m||^2)}{\sum_j \lambda_j \exp(-\frac{1}{2\sigma^2}||y_i - \mu_j||^2)}$$

As $\sigma^2 \to 0$, then $w_{im}$ will become close to a binary variable, where:

$$w_{im} = \begin{cases} 1 & \text{if } \mu_m \text{ is in } y_i \\ 0 & \text{else} \end{cases}$$

As such, we see that (1) there can no longer be fractional assignment to mixtures, and that $y_i$ is either in one mixture, or another (since the entire weight in a specific mixture will be 1), and (2) the assignment of $y_i$ into "mixtures" is equivalent to how $k$-means pushes points into clusters.

- $M$-Step:

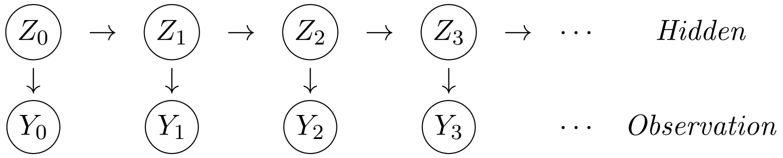
$$\mu_m^{(t+1)} = \frac{\sum_i w_{im}^{(t)} y_i}{|C_m|} = \frac{\sum_{i \in C_m} y_i}{|C_m|},$$

where $|C_m|$ denotes the size of cluster $m$. Note also that this step is equivalent to the step in $k$-means in which we re-compute the centroids of the clusters.

# 4. Hidden Markov Models

Up until now, we have dealt with models that have assumed an independence structure between the hidden variables. This makes the probability distribution easy to estimate, because ... Now, we will introduce a very simple form of local dependence between the hidden variables, in the form of a Markov chain.

**Definition 4.1 (Hidden Markov Models)**



*The different components of the model are:*

1. *Hidden States:* $\{1, ..., N\}$ *(State space of* $Z_t$*)*

2. *Observed States* $\{1, ..., M\}$ *(State space of* $Y_t$*)*

3. *State transition matrix:* $A = \Big(a_{ij}\Big)_{N \times N}$ *(where* $a_{ij} = P(Z_{t+1} = j | Z_t = i)$*)*

4. *Emission Probabilities:* $B = (b_j(k))$[3] *(where* $b_j(k) = P(Y_t = k | Z_t = j)$*)*

[3] For whatever reason, this is represented as a function of $k$. If we were to use the same notational practice as when we write our state transition matrix, then it would simply be $b_{jk}$, which is a nicer representation than $b_j(k)$, especially when we start using the sufficient statistics later on.

5. *Initial state distribution:* $\pi = (\pi_1, ..., \pi_N)$ *(where* $\pi_i = P(Z_t = i)$*)*

Usually, the different state spaces are assumed to be known, as well as the initial state distribution. As such, the parameters of interest are the **transition probabilities** ($a_{ij}$) and the **emission probabilities** ($b_j(k)$) for all $i, j, k$. In other words, we want to know the probabilities of $Z$ transitioning to a different state, as well as the probability of observing some $Y_t$, given the value $Z_t$ takes on.

Generally speaking, within a graph-like structure, graph separation implies conditional independence.[4] As such, looking at the visualization of a hidden Markov model, we see that if we are able to condition on a hidden state $Z_t$, the previous latent states are no longer relevant to the value that $Y_t$ (the observed state) takes on. If we do not condition on $Z_t$, the distribution of $Y_t$ becomes very complex, because it is then dependent (indirectly) on the other previous $Y_{t-1}, Y_{t-2}, ...$ values, which depend on the previous latent states.

Using this Markovian structure, we can factor the joint probability distribution of $Y$ and $Z$:

$$P(Y, Z) = P(Z_1)P(Y_1|Z_1) \cdot \prod_{t=2}^{n} P(Z_t|Z_{t-1})P(Y_t|Z_t)$$

## 4.1. Specification as Missing Data Problem

We can frame the estimation of an HMM model as a missing data problem and use the EM algorithm in order to estimate the transition and emission probabilities.

To begin, we introduce an indicator variable: $Z_{tj} = \mathbb{1}\{Z_t = j\}$. Now we can specify the complete data likelihood (where $\theta = (a_{ij}, b_j(k))$, and $T(k)$ represents the set of time indices that correspond to when $Y_t = k$:[5]

$$\begin{aligned} P(Y, Z, \theta) &= \prod_{j=1}^{N} \prod_{k=1}^{M} \prod_{t \in T(k)} b_j(k)^{Z_{tj}} \times \prod_{i=1}^{N} \prod_{j=1}^{N} \prod_{t=2}^{n} (a_{ij})^{Z_{(t-1)i} Z_{tj}} \\ &= \prod_{j=1}^{N} \prod_{k=1}^{M} b_j(k)^{\sum_{t \in T(k)} Z_{tj}} \times \prod_{i=1}^{N} \prod_{j=1}^{N} (a_{ij})^{\sum_{t=2}^{n} Z_{(t-1)i} Z_{tj}} \end{aligned}$$

[4] When causal inference people talk about "blocking" a particular path, this is what they're talking about!

[5] This was denoted as $t : Y_t = k$ in lecture, but I found that confusing. I think technically it should be the set of $t$ such that $Y_t = k$, in which case the formal notation would have been $t \in \{t \mid Y_t = k\}$, but then we have some abuse of notation because of all the $t$'s. Anyway, I digress.

The complete data log-likelihood function will be:

$$\begin{aligned}
\log P(Y, Z, \theta) &= \sum_{j,k} \sum_{t \in T(k)} Z_{tj} \log b_j(k) + \sum_{i,j} \sum_{t=2}^{n} Z_{(t-1)i} Z_{tj} \log a_{ij} \\
&= \sum_{i,j} \log b_j(k) \underbrace{\sum_{t \in T(k)} Z_{tj}}_{:= D_{jk}} + \sum_{i,j} \log a_{ij} \underbrace{\sum_{t=2}^{n} Z_{(t-1)i} Z_{tj}}_{:= C_{ij}} \\
&= \sum_{i,j} D_{jk} \log b_j(k) + \sum_{i,j} C_{ij} \log a_{ij}
\end{aligned}$$

By some stroke of miracle, $D_{jk}$ and $C_{ij}$ happen to be the sufficient statistics for the corresponding parameters of interest. (More specifically, $D_{jk}$ is the sufficient statistic for $b_j(k)$ and $C_{ij}$ is the sufficient statistic for $a_{ij}$.) This actually does make sense, because $D_{jk}$ is the count for the number of times we observe $Y_t$ emitting the state $k$ when the latent state $Z_t = j$. In a similar vein, $C_{ij}$ is the count for the number of times we observe the transition of $Z$ from state $j$ to $i$.

Now since we know the sufficient statistics it is easy to compute the MLE. We simply need to normalize in order to ensure that the probabilities do not sum to something over 1 and induce a mathematical felony. Therefore, the MLE estimates given the complete data would be:

$$\widehat{b}_j(k) = \frac{D_{jk}}{\sum_k D_{jk}} \quad \text{(Denominator sums over the } M \text{ observed states)}$$

$$\hat{a}_{ij} = \frac{C_{ij}}{\sum_j C_{ij}} \quad \text{(Denominator sums over the } N \text{ hidden states)}$$

However, since $Z$ is in fact unobserved, we have to use EM to estimate all of this. On the surface, it actually isn't too bad! To do the E-step, we simply take the expectation over the complete data log likelihood:[6]

$$\mathbb{E}(\log(P(Y, Z, \theta) | Y; \theta^{(m)})) = \sum_{j,k} \mathbb{E}(D_{jk} | Y, \theta^{(m)}) \log b_j(k)^{(m)} + \sum_{i,j} \mathbb{E}(C_{ij} | Y; \theta^{(m)}) \log a_{ij}^{(m)}$$

The expectation over $D_{jk}$ and $C_{ij}$ can be expanded:

$$\begin{aligned}
\mathbb{E}(D_{jk} | Y, \theta^{(m)}) &= \mathbb{E}\left( \sum_{t \in T(k)} Z_{tj} | Y, \theta^{(m)} \right) \\
&= \sum_{t \in T(k)} \mathbb{E}(Z_{tj} | Y, \theta^{(m)}) \\
&\equiv \sum_{t \in T(k)} P(Z_t = j | Y, \theta^{(m)}) \qquad (1)
\end{aligned}$$

[6] Note that because we have the actual time steps in the HMM (which are represented by the $t$ subscripts), we have to denote the iterations of the EM algorithm using $m$, because we don't have enough symbols already.

$$\begin{aligned}
\mathbb{E}(C_{ij}|Y,\theta^{(m)}) &= \mathbb{E}\left(\sum_{t\in T(k)} Z_{(t-1)i}Z_{tj}|Y,\theta^{(m)}\right) \\
&= \sum_{t\in T(k)} \mathbb{E}(Z_{(t-1)i}Z_{tj}|Y,\theta^{(m)}) \\
&\equiv \sum_{t\in T(k)} P(Z_t=j,Z_{t-1}=i|Y,\theta^{(m)}) \qquad (2)
\end{aligned}$$

This is where the problem becomes a Big Mess. As it turns out, estimating these probabilities is a non-trivial task. We can rewrite (1) as:

$$\begin{aligned}
P(Z_t=j|Y,\theta^{(m)}) &\propto P(Z_t=j,Y_{1:t},Y_{(t+1):n}) \\
&= P(Y_{1:t},Z_t=j)P(Y_{(t+1):n}|Z_t=j)
\end{aligned}$$

Additionally, we can rewrite (2) as:

$$\begin{aligned}
P(Z_{t-1}=i,Z_t=j|Y) &\propto P(Z_{t-1}=i,Z_t=j,Y) \\
&= P(Y_{1:(t-1)},Z_{t-1}=i)\cdot P(Y_t,Y_{(t+1):n},Z_t=j|Y_{1:(t-1)},Z_{t-1}=i) \\
&= P(Y_{1:(t-1)},Z_{t-1}=i)\cdot P(Y_t|Z_t=j)\cdot P(Y_{(t+1):n},Z_t=j|Y_{1:(t-1)},Z_{t-1}=i) \\
&= P(Y_{1:(t-1)},Z_{t-1}=i)\cdot \underbrace{P(Y_t|Z_t=j)}_{=b_j(Y_t)}\cdot P(Y_{(t+1):n}|Z_t=j)\,\underbrace{P(Z_t=j|Z_{t-1}=i)}_{=a_{ij}}
\end{aligned}$$

Notice that in the expanded form of (2), we actually do not have to estimate the quantities of $b_j(Y_t)$ and $a_{ij}$ because those are part of our parameter $\theta$. As such, we assume that they are given in this step. As such, the key to computing both of these include estimating the joint probability of $P(Y_{1:t},Z_t=j)$ and $P(Y_{(t+1):n}|Z_t=j)$. To add an extra layer of notation, because we're actually secretly trying to review our knowledge of the Greek and English alphabet and want to use every possible letter, we define:

$$\alpha_t(i)=P(Y_{1:t},Z_t=i)$$

$$\beta_t(j)=P(Y_{(t+1):n}|Z_t=j)$$

To estimate both of these quantities, we will need to use something known as forward and backwards summation. Let's unpack this!

1. Estimating $\alpha_t(j)$ (Forward Summation):
   Let's begin by factorizing this joint probability some more![7]

$$\begin{aligned}
\alpha_t(j) &= P(Y_{1:t}, Z_t = j) \\
&= \sum_{i=1}^{N} P(Y_{1:t}, Z_t = j, Z_{t-1} = i) \\
&= \sum_{i=1}^{N} P(Z_{t-1} = i | Z_t = j, Y_{1:t}) \cdot P(Y_t | Z_t = j) P(Y_{1:(t-1)}, Z_{t-1} = i) \\
&= P(Y_t | Z_t = j) \cdot \sum_{i=1}^{N} \underbrace{P(Z_{t-1} = i | Z_t = j, Y_{1:t})}_{=a_{ij}} \cdot \underbrace{P(Y_{1:(t-1)}, Z_{t-1} = i)}_{\alpha_{t-1}(i)} \\
&\equiv P(Y_t | Z_t = j) \cdot \sum_{i=1}^{N} a_{ij} \cdot \alpha_{t-1}(i)
\end{aligned}$$

   This means if we can calculate $\alpha_1(i)$, then we can iteratively plug in these different $\alpha$ values to solve for all $\alpha_t(\cdot)$ for $t = 1, ..., n$. The good thing is that $\alpha_1(i)$ is actually very simple[8] to compute:

$$\alpha_1(i) = P(Y_1, Z_1 = i) = \underbrace{P(Z_1 = i)}_{\equiv \pi_i} \underbrace{P(Y_1 | Z_1 = i)}_{b_1(Y_1)}$$

2. Estimating $\beta_t(i)$ (Backward Summation):
   In a similar vein, we can estimate $\beta_t(i)$ by working backwards. We begin by initializing $\beta_n(i) = 1$. Once again we may expand $\beta_t(i)$ to be of a more friendly form:

$$\begin{aligned}
\beta_t(i) &= P(Y_{(t+1):n} | Z_t = i) \\
&= \sum_{j=1}^{N} P(Y_{(t+1):n}, Z_{t+1} = j | Z_t = i) \\
&= \sum_{j=1}^{N} P(Y_{(t+1):n} | Z_{t+1} = j, Z_t = i) P(Z_{t+1} = j | Z_t = i) \\
&= \sum_{j=1}^{N} \underbrace{P(Y_{(t+2):n} | Z_{t+1} = j)}_{=\beta_{t+1}(j)} \underbrace{P(Y_{t+1}, Z_{t+1} = j)}_{=b_j(Y_{t+1})} \underbrace{P(Z_{t+1} = j | Z_t = i)}_{=a_{ij}} \\
&= \sum_{j=1}^{N} \beta_{t+1}(j) \cdot b_j(Y_{t+1}) \cdot a_{ij}
\end{aligned}$$

[7] We will do so by using that fun trick of summing over some variable that isn't in the initial joint distribution. Very cool, very legal.

[8] Not just simple from the standpoint of a professor who says that about all of Math because they have reached some sort of nirvana (Wu, 2018), but it literally is very simple.

As such, we may work backwards to solve for all $\beta_t(\cdot)$.

In summary:

Step 1: Fix $a_{ij}$ and $b_j(k)$. Perform forward summation to estimate $\alpha_t(\cdot)$ and then backward summation to estimate $\beta_t(\cdot)$ for all $t = 1, ..., n$.

Step 2: E-Step: Plug into the expectation of $\mathbb{E}(D_{jk}|Y, \theta^{(m)})$ and $\mathbb{E}(C_{ij}|Y, \theta^{(m)})$.

$$\mathbb{E}(D_{jk}|Y, \theta^{(m)}) = \alpha_t(j)\beta_t(j) := D_{jk}^{(m)}$$

$$\mathbb{E}(C_{ij}|Y, \theta^{(m)}) = a_{ij}b_j(Y_t)\alpha_{t-1}(i)\beta_t(j) := C_{ij}^{(m)}$$

Therefore, the full expected log-likelihood will be:

$$\mathbb{E}(\log P(Y, Z; \theta)) = \sum_{j,k} \alpha_t(j)\beta_t(j) \cdot \log b_j(k)^{(m)} + \sum_{i,j} a_{ij}b_j(Y_t)\alpha_{t-1}(i)\beta_t(j) \cdot \log a_{ij}^{(m)}$$

Step 3: M-Step: Update!

$$b_j(k)^{(m+1)} = \frac{D_{jk}^{(m)}}{\sum_k D_{jk}^{(m)}}$$

$$a_{ij}^{(m+1)} = \frac{C_{ij}^{(m)}}{\sum_j C_{ij}^{(m)}}$$

## 4.2. Viterbi Algorithm

The Viterbi Algorithm tries to detect the most likely sequence of hidden states (which is aptly referred to as the **Viterbi path**). In essence, we want to maximize the posterior distribution of $P(Z|Y)$ (where $Z$ here is technically a vector containing all the states through time: $Z = (Z_1, ..., Z_n)$, with the realized state vector $z = (z_1, ..., z_n)$. More specifically:

$$\begin{aligned} \hat{Z} &= \underset{z}{\operatorname{argmax}}\, P(Z = z \mid Y) \\ &\equiv \underset{z}{\operatorname{argmax}}\, P(Z = z, Y) \end{aligned}$$

Take a given time $t+1$:

$$\begin{aligned} P(Z_{1:(t+1)}, Y_{1:(t+1)}) &= P(Z_{1:t}, Y_{1:t}, Z_{t+1}, Y_{t+1}) \\ &= P(Z_{1:t}, Y_{1:t}) \cdot P(Y_{t+1}, Z_{t+1} | Z_{1:t}, Y_{1:t}) \\ &= P(Z_{1:t}, Y_{1:t}) \cdot \underbrace{P(Z_{t+1} | Z_{1:t}, Y_{1:t})}_{=P(Z_{t+1}|Z_t)} \underbrace{P(Y_{t+1} | Z_{t+1}, Z_{1:t}, Y_{1:t})}_{P(Y_{t+1}|Z_{t+1})} \\ &= P(Z_{1:t}, Y_{1:t}) \cdot P(Z_{t+1}|Z_t) P(Y_{t+1}|Z_{t+1}) \\ &= P(Z_{1:t}, Y_{1:t}) \cdot a_{ij} \cdot b_j(Y_{t+1}) \end{aligned}$$

The last line is given in the lecture slides. However, it is important to note that up until this point in time, we've been kind of loose about our notation with $i$'s and $j$'s. In this context though, it actually matters now which one we use. So let's rewrite the above more carefully:

$$P(Z_{1:(t+1)}, Y_{1:(t+1)}) = P(Z_{1:(t-1)}, Z_t = i, Y_{1:t}) \cdot \underbrace{P(Z_{t+1} = j \mid Z_t = i)}_{a_{ij}} \cdot P(Y_{t+1} | Z_{t+1} = j)$$

There are several takeaways from this. Firstly, the joint probability function of $Z_{1:t}$ and $Y_{1:t}$ can be written as a function of previous joint probabilities $Z_{1:(t-1)}, Y_{1:(t-1)}$. Therefore, by maximizing the joint probabilities iteratively for $t = 1, ..., n-1$, we can solve for the maximum sequence. Secondly, while iteratively maximizing, at a time point $t+1$, we want to compute, for all states $1, ..., N$ that $Z_{t+1}$ could take on, what is the most likely value that $Z_t$ takes on? In other words, for a fixed $j$, we want to find the state $i$ that is most likely.

To make this less (or potentially more?) notationally confusing, we introduce a vector $\delta_{t+1}$, where the $j$-th entry is:[9]

$$\delta_{t+1}(j) = \left( \max_i \delta_t(i) \cdot a_{ij} \right) \cdot b_j(Y_{t+1})$$

Notice that the $b_j(Y_{t+1})$ term is left out of the maximization problem, because there are no $i$'s to maximize over. Once we iteratively work through all $t = 1, ..., n-1$ maximizations, then we can backsolve for the maximal path.

Therefore, the Viterbi Algorithm can be formally written as as:

**Definition 4.2 (Viterbi Algorithm)**

***Step 1. Initialization:***

*For $i = 1, ..., N$:*

$$\delta_1(i) = \pi_i b_i(Y_1)$$

[9] In lecture, this was introduced as just a function, but I think it is somewhat important to note that there are $N$ of these at each time point for each $j$ value.

*Note that this is essentially solving for the posterior $P(Z_1 = i|Y_1)$ for all $i = 1, ..., N$. (More formally: $P(Z_1 = i) \cdot P(Y_1|Z_1 = i) = \pi_i \cdot b_i(Y_1) \equiv \delta_1(i)$.)*

***Step 2. Forward Maximization:***

*For $t = 1, ..., n - 1$:*

*For $j = 1, ..., N$:*

$$\delta_{t+1}(j) = \left(\max_i \delta_t(i) \cdot a_{ij}\right) \cdot b_j(Y_{t+1})$$

$$\gamma_{t+1}(j) = \operatorname*{argmax}_i \delta_t(i) \cdot a_{ij}$$

*$\gamma_t$ is a vector that tracks the indices that correspond to the maximum values at $t$.*

***Step 3. Backward Tracking:***

$$\hat{Z}_n = \operatorname*{argmax}_i \delta_n(i)$$

*For $t = n - 1, ..., 1$:*

$$\hat{Z}_t = \gamma_{t+1}(\hat{Z}_{t+1})$$

*Informally, at the $n$-th time step, we simply look at our vector $\delta_n$ and identify the most probable state for $Z_n$ by looking at the maximum entry. With this set, we can now work backwards. Plugging in the estimated state at a time $Z_{t+1}$, we set $j = Z_{t+1}$ and identify the maximum values given that particular $j$ value.*

## 5. Random Graphs

We now allow for complex dependencies between variables. This is done through random graphs. In this context, we observe various realizations of a random graph, and want to estimate the probability distribution of this random graph. More specifically, we observe (from data) an adjacency matrix $Y = (Y_{ij})_{n \times n}$, and want to estimate the population version of this matrix (sometimes referred to as $A$ in lecture). Therefore, $Y$ can be thought of a single sample. We want to estimate the population version of this ($A$). Additionally, we also care about include identifying communities within the graph (or, a more fancy way to say this is: identifying heterogeneity among nodes).

When estimating random graphs, we assume that each node $i \in V$ is associated with some latent variable $Z_i$. This hidden variable represents the location of the node in a **latent space**. The distribution of edges (edges are represented as $Y_{ij}$) in a particular graph depends on the distances between hidden points (i.e., $Y_{ij}$ depends on $||Z_i - Z_j||$). Therefore our goal is to estimate the probability distribution of edges, given these latent points: $P(Y_{ij}|Z_i, Z_j)$.

Intuitively, this should make sense. Edges between nodes represent some sort of association between the nodes. As such, the closer related the nodes are to one another, the higher the probability should be of there being an edge connecting the two nodes. Additionally, if the nodes are related to one another, then we expect the latent representation of the nodes in the latent space to reflect this.

We will talk about two specific types of random graphs: stochastic block models and graphons.

## 5.1. Stochastic Block Models

We assume that there are $K$ communities among $n$ nodes. There are a set of latent cluster labels for each node:

$$Z_i = (Z_{i1}, ...., Z_{ik})$$

This is effectively a one-hot vector that denotes cluster membership for the $i$-th node. We can think of $Z_i \sim \text{Multinomial}(1, \pi)$, where $\pi = (\pi_1, ..., \pi_k)$ is a vector containing the probability of being in a cluster. The key takeaway from this setup is that given $Z_i$ and $Z_j$ (i.e., the cluster memberships of nodes $i$ and $j$), the edge $Y_{ij}$ is drawn independently. The connection probabilities between the $k$ communities is given by a $k \times k$ matrix denoted $\gamma$. Technically speaking, $\gamma$ is just a matrix containing parameters that go into estimating the density associated with the probability of an edge; however, we still refer to it as the connection probability matrix.

More specifically:

$$[Y_{ij} | Z_{im} = 1, Z_{jl} = 1] \sim f(\gamma_{ml})$$

To decompose the math a little bit, this means that if the $i$-th node belongs to cluster $j$ and the $j$-th node belongs to cluster $l$, then the probability of there being an edge between $i$ and $j$ will depend on $\gamma_{ml}$ (where $\gamma_{ml}$ is an entry from the matrix $\gamma$).

### 5.1.1. Missing Data Problem

We can once again frame this as a missing data problem. The parameters we want to estimate are $\theta = (\pi, \gamma)$ (i.e., the probability of being in a cluster, and the set of parameters that determine if there are connections between nodes in different clusters). The hidden variable is $Z$, which denotes cluster membership, and the observed data will be the adjacency matrix $A = (Y_{ij})$, which represents the observed graph. We will assume for simplicity that the probability of an edge follows a Bernoulli distribution:

$$[Y_{ij} | Z_{im} = 1, Z_{jl} = 1] \sim \text{Bernoulli}(\gamma_{ml})$$

$$\implies P(Y_{ij}, \gamma_{ml}) = \gamma_{ml}^{Y_{ij}} \cdot (1 - \gamma_{ml})^{1 - Y_{ij}}$$

The observed data likelihood is effectively the probability of seeing the (sample) adjacency matrix

$Y$. The complete data likelihood includes the cluster membership variables $Z$ as well:[10]

$$\begin{aligned}P(Y,Z) &= P(Z)\cdot P(Y|Z)\\ &= \left(\prod_{m=1}^{k}\prod_{i=1}^{n}\pi_m^{Z_{im}}\right)\cdot\left(\prod_{i\neq j}\prod_{m,l} f(Y_{ij},\gamma_{ml})^{Z_{im}Z_{jl}}\right)^{\frac{1}{2}}\\ &= \left(\prod_{m=1}^{k}\pi_m^{\sum_{i=1}^{n} Z_{im}}\right)\cdot\left(\prod_{i\neq j}\prod_{m,l} f(Y_{ij},\gamma_{ml})^{Z_{im}Z_{jl}}\right)^{\frac{1}{2}}\end{aligned}$$

Therefore, the log-likelihood is:

$$\log P(Y,Z;\theta) = \sum_{m=1}^{k}\sum_{i=1}^{n} Z_{im}\log\pi_m + \frac{1}{2}\sum_{i\neq j}\sum_{m,l} Z_{im}Z_{jl} f(Y_{ij},\gamma_{ml})$$

Therefore, intuitively, a natural next step is to take the expectation over the complete data log-likelihood and then run the EM algorithm:

$$\mathbb{E}(\log P(Y,Z;\theta)|Y) = \sum_{m=1}^{k}\sum_{i=1}^{n}\mathbb{E}(Z_{im}|Y)\log\pi_m + \frac{1}{2}\sum_{i\neq j}\sum_{m,l}\mathbb{E}(Z_{im}Z_{jl}|Y) f(Y_{ij},\gamma_{ml})$$

Now the issue is that it is very difficult to compute $\mathbb{E}(Z_{im} \mid Y)$ and $\mathbb{E}(Z_{im}\cdot Z_{jl} \mid Y)$ because of all these complex dependencies that exist.

Therefore, we need to use the modified version of the EM algorithm: **Variational EM**. Recall that to apply variational EM, we must restrict the class of functions that we search over in order to maximize a lower bound that bounds the likelihood function. In this case, we will restrict the class of functions to the set of functions $F$ that impose an independence structure in the cluster labels:

$$F(Z) = \prod_{i=1}^{n} h(Z_i,\tau_i)$$

In essence, we assume that the rows of $Z$ (or rather, the cluster membership between the different nodes) are independent from one another, *conditional on $Y$*. In other words, we are imposing a conditional independence assumption on the latent space, when in reality, the conditional distribution of the latent space on the observed data is often times extremely complex. Therefore, this serves as an approximation of sorts to the actual relationship. Note that $Z_i|Y$ do *not* have to be

[10] In the first line, re-writing the joint probability distribution in this manner is particularly helpful because the distribution of the $Z$'s is simply multinomial, while the *conditional* distribution of $Y$, given the cluster membership is Bernoulli. When we do not condition on $Z$, the distribution of $Y$ becomes extremely complex, much like in the case of the Hidden Markov Model.

identically distributed, as each row/node membership has its own parameter $\tau_i$ that corresponds to it. This simplifies things massively, as now we may write:

$$\mathbb{E}_F(Z_{im}Z_{jl}) = \mathbb{E}_F(Z_{im}) \cdot \mathbb{E}_F(Z_{jl}) = \tau_{im}\tau_{jl}$$

Notice that all of these expectations are taken over the function $F$! Now, plugging back into the lower bound function:

$$\begin{aligned}
L(\theta, F) &= \mathbb{E}_F(\log P(Y, Z; \theta)) + H(F) \\
&= \sum_{m=1}^{k}\sum_{i=1}^{n} \mathbb{E}_F(Z_{im}) \log \pi_m + \frac{1}{2}\sum_{i \neq j}\sum_{m,l} \mathbb{E}_F(Z_{im}Z_{jl}) f(Y_{ij}, \gamma_{ml}) + \mathbb{E}_F(-\log F(Z)) \\
&= \sum_{m=1}^{k}\sum_{i=1}^{n} \tau_{im} \log \pi_m + \frac{1}{2}\sum_{i \neq j}\sum_{m,l} \tau_{im}\tau_{jl} f(Y_{ij}, \gamma_{ml}) - \sum_{i=1}^{n}\sum_{m} \tau_{im} \log \tau_{im}
\end{aligned}$$

Therefore, we just need to solve the following optimization problem in order to maximize the lower bound:

$$\begin{cases} \max\limits_{\tau} L(\theta, F) \\ \\ \sum\limits_{m} \tau_{im} = 1 \quad \forall\ i = 1, ..., n \end{cases}$$

We can use our profound multivariate calculus knowledge and note that this is simply a Lagrange Multiplier problem! The Lagrangian will take on the following form:

$$\mathcal{L} = L(\theta, F) - \sum_{i} \lambda_i \left( \sum_{m} \tau_{im} - 1 \right)$$

looking at just a single $\tau_{im}$:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \tau_{im}} &= \sum_{m=1}^{k} \log \pi_m + \frac{1}{2}\sum_{i \neq j}\sum_{m,l} \tau_{jl} \cdot f(Y_{ij}, \gamma_{ml}) - \sum_{m} \log \tau_{im} - (\lambda_i + 1) \\
&= \sum_{m=1}^{k} (\log \pi_m - \log \tau_{im}) + \frac{1}{2}\sum_{i \neq j}\sum_{m,l} \tau_{jl} \cdot f(Y_{ij}, \gamma_{ml}) - (\lambda_i + 1) = 0
\end{aligned}$$

$$\Longrightarrow \sum_{m=1}^{k} (\log \pi_m - \log \tau_{im}) + \frac{1}{2}\sum_{i \neq j}\sum_{m,l} \tau_{jl} \cdot f(Y_{ij}, \gamma_{ml}) = \lambda_i + 1$$

There isn't a closed form solution to this. However, we note that:

$$\tau_{im} \propto \pi_m^{(t)} \prod_{i \neq j}\prod_{l} f(Y_{ij}, \gamma_{ml}^{(t)})^{\tau_{jl}}$$

As such, we can use this to iteratively estimate $\tau^{(t)}$.

The $M$-step remains almost the same. Given $\tau^{(t)}$, $\max_\pi L(\theta, \tau^{(t)})$ s.t. $\sum_m \pi_m = 1$:

$$\pi_m^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} \tau_{im}^{(t)}$$

$$\gamma_{ml}^{(t+1)} = \frac{\sum_{i \neq j} \tau_{im}^{(t)} \cdot \tau_{jl}^{(t)} Y_{ij}}{\sum_{i \neq j} \tau_{im}^{(t)} \tau_{jl}^{(t)}}$$