Introduction to ML (CS771), Autumn 2020
Indian Institute of Technology Kanpur
Bonus Assignment 1

Student Name: Deeksha Arora
Roll Number: 20111017
Date: December 19, 2020

QUESTION 1

Let $\boldsymbol{v} \in \mathbb{R}^N$ is the eigen vector of the covariance matrix $\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^T$, where $\boldsymbol{X}$ is a $N \times D$ matrix ($D > N$).

We know that, $\boldsymbol{A}\boldsymbol{v} = \lambda\boldsymbol{v}$, where $\boldsymbol{v}$ is the eigen vector of matrix $\boldsymbol{A}$. Therefore,

$$\left(\frac{1}{N}\mathbf{X}\mathbf{X}^\top\right)\boldsymbol{v} = \lambda\boldsymbol{v} \tag{1}$$

Multiplying by $\boldsymbol{X}^\top$ we get:

$$\mathbf{X}^\top\left(\frac{1}{N}\mathbf{X}\mathbf{X}^\top\right)\boldsymbol{v} = \lambda\mathbf{X}^\top\boldsymbol{v} \tag{2}$$

$$\left(\frac{1}{N}\mathbf{X}^\top\mathbf{X}\right)\left(\mathbf{X}^\top\boldsymbol{v}\right) = \lambda\left(\mathbf{X}^\top\boldsymbol{v}\right) \tag{3}$$

Let $\boldsymbol{u} = \mathbf{X}^\top\boldsymbol{v}$. Therefore,

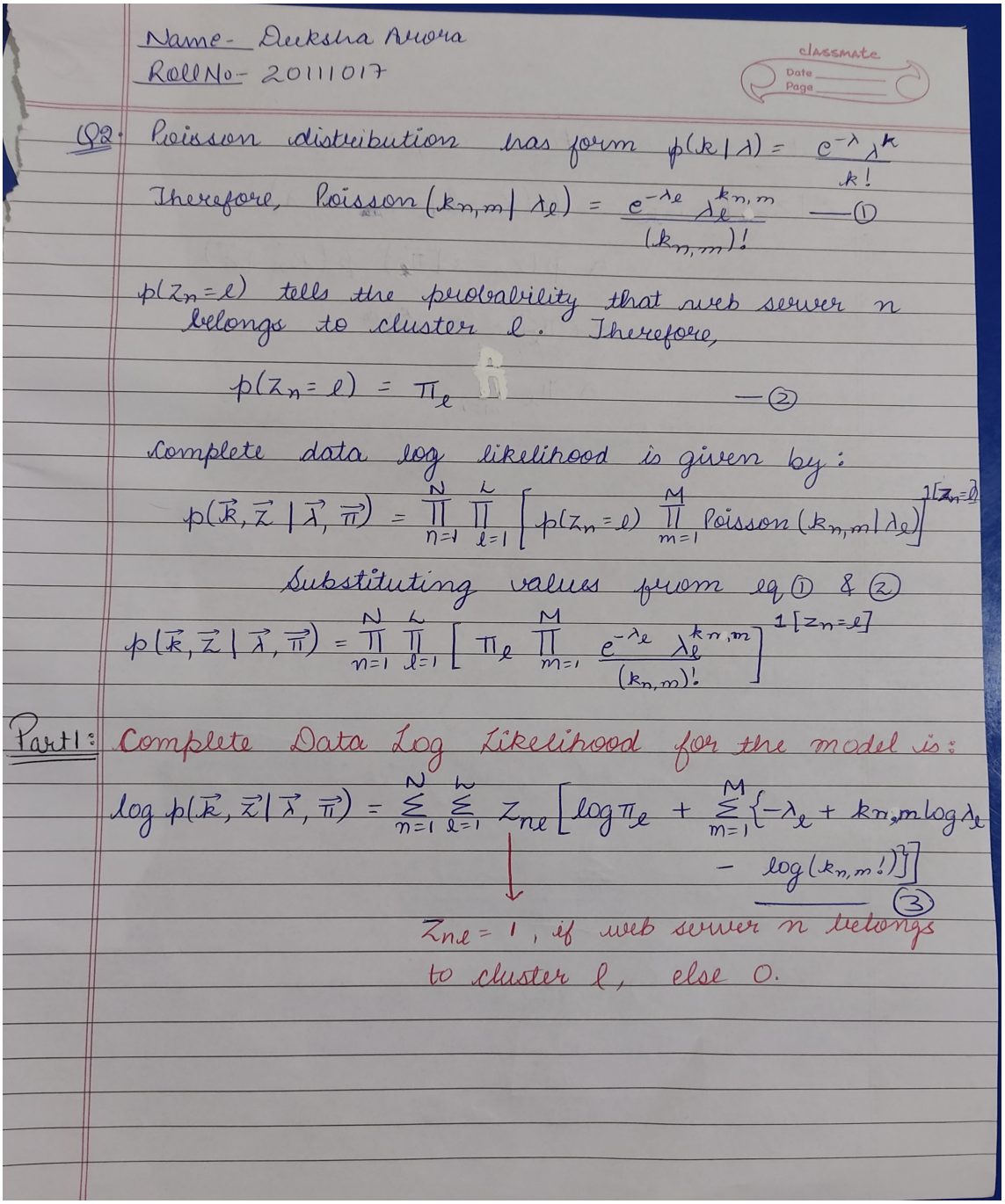
$$\left(\frac{1}{N}\mathbf{X}^\top\mathbf{X}\right)\boldsymbol{u} = \lambda\boldsymbol{u} \tag{4}$$

So, $\boldsymbol{u} = \mathbf{X}^\top\boldsymbol{v}$ is an eigen vector of the matrix $\frac{1}{N}\mathbf{X}^\top\mathbf{X}$. So for every eigen vector $\boldsymbol{v}$ of $\frac{1}{N}\mathbf{X}\mathbf{X}^\top$ there exists an eigen vector $\boldsymbol{u} = \mathbf{X}^\top\boldsymbol{v}$ for the matrix $\frac{1}{N}\mathbf{X}^\top\mathbf{X}$.
Therefore, instead of doing eigen decomposition of $\frac{1}{N}\mathbf{X}^\top\mathbf{X}$, $D \times D$ matrix, we can find eigen vectors by doing eigen decomposition of $\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^T$ which is a $N \times N$ matrix. This way of obtaining eigen vectors is advantageous because $D > N$ and hence it is computationally more efficient to find eigen vectors of $N \times N$ matrix instead of $D \times D$ matrix.

Introduction to ML (CS771), Autumn 2020
Indian Institute of Technology Kanpur
Bonus Assignment 1

Student Name: Deeksha Arora
Roll Number: 20111017
Date: December 19, 2020

QUESTION 2



Name- Deeksha Arora
Roll No- 20111017

Q2. Poisson distribution has form $p(k|\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$

Therefore, $\text{Poisson}(k_{n,m}|\lambda_\ell) = \frac{e^{-\lambda_\ell}\lambda_\ell^{k_{n,m}}}{(k_{n,m})!}$ —①

$p(z_n=\ell)$ tells the probability that web server n belongs to cluster ℓ. Therefore,

$$p(z_n=\ell) = \pi_\ell \quad \text{—②}$$

Complete data log likelihood is given by:

$$p(\vec{k},\vec{z}\,|\,\vec{\lambda},\vec{\pi}) = \prod_{n=1}^{N}\prod_{\ell=1}^{L}\left[p(z_n=\ell)\prod_{m=1}^{M}\text{Poisson}(k_{n,m}|\lambda_\ell)\right]^{\mathbb{1}[z_n=\ell]}$$

Substituting values from eq ① & ②

$$p(\vec{k},\vec{z}\,|\,\vec{\lambda},\vec{\pi}) = \prod_{n=1}^{N}\prod_{\ell=1}^{L}\left[\pi_\ell\prod_{m=1}^{M}\frac{e^{-\lambda_\ell}\lambda_\ell^{k_{n,m}}}{(k_{n,m})!}\right]^{\mathbb{1}[z_n=\ell]}$$

Part 1: Complete Data Log Likelihood for the model is:

$$\log p(\vec{k},\vec{z}\,|\,\vec{\lambda},\vec{\pi}) = \sum_{n=1}^{N}\sum_{\ell=1}^{L} z_{n\ell}\left[\log\pi_\ell + \sum_{m=1}^{M}\{-\lambda_\ell + k_{n,m}\log\lambda_\ell - \log(k_{n,m}!)\}\right] \quad \text{—③}$$

$z_{n\ell} = 1$, if web server n belongs to cluster ℓ, else 0.

Name- Deeksha Arora
Roll No - 20111017

Part2: Estimation Step

$$E[z_{ne}] = r_{ne} = p(z_n = l \mid \vec{k}_n, \vec{\pi}, \vec{\lambda})$$

$$\propto p(z_n = l \mid \vec{\pi})\, p(\vec{k}_n \mid \vec{\lambda})$$

$$\propto \pi_l \prod_{m=1}^{M} \frac{e^{-\lambda_l} \lambda_l^{k_{n,m}}}{(k_{n,m})!}$$

Part3: Maximization Step : Update θ by maximizing the expected complete data log-likelihood

$$\hat{\theta} = \text{argmax}_\theta \; E_{p(z \mid \vec{k}, \vec{\pi}, \vec{\lambda})} \log\left(p(\vec{k}, \vec{z} \mid \vec{\lambda}, \vec{\pi})\right)$$

$$= \text{argmax}_\theta \sum_{n=1}^{N} E_{p(z_n \mid \vec{k}_n, \vec{\pi}, \vec{\lambda})} \log\left(p(\vec{k}_n, \vec{z}_n \mid \vec{\lambda}, \vec{\pi})\right) \quad \text{---- (4)}$$

Differentiating the above equation w.r.t. π_l and equating it to 0 gives:

$$\pi_l = \frac{\sum_{n=1}^{N} E[z_{nl}]}{N}$$

Let $r_{nl} = E[z_{nl}]$ and $N_l = \sum_{n=1}^{N} r_{nl}$

$$\therefore \quad \boxed{\pi_l = \frac{N_l}{N}}$$

Therefore, the MLE estimate for the mixing

Name- Deeksha Arora
Roll No- 20111017

proportions are given by the ratio of sample sizes from each cluster

⇒ Differentiating eq (4) w.r.t. λ_l and equating to 0 gives :-

$$\frac{\partial \log p(\vec{k}, \vec{z} \mid \vec{\lambda}, \vec{\pi})}{\partial \lambda_l} = \sum_{n=1}^{N} \left[E[z_{nl}] \sum_{m=1}^{M} (-1) + \sum_{m=1}^{M} \frac{k_{n,m}}{\lambda_l} \right]$$

$$\therefore \lambda_l = \frac{\sum_{n=1}^{N} \sum_{m=1}^{M} k_{n,m}}{M \sum_{n=1}^{N} E[z_{nl}]}$$

Since $\gamma_{nl} = E[z_{nl}]$ and $N_l = \sum_{n=1}^{N} \gamma_{nl}$

$$\therefore \lambda_l = \frac{\sum_{n=1}^{N} \sum_{m=1}^{M} k_{n,m}}{M \cdot N_l}$$

$$\boxed{\lambda_l = \frac{\sum_{n=1}^{N} k_n}{M \cdot N_l}}, \text{ where } k_n = \{k_{n1}, \cdots, k_{nM}\}$$

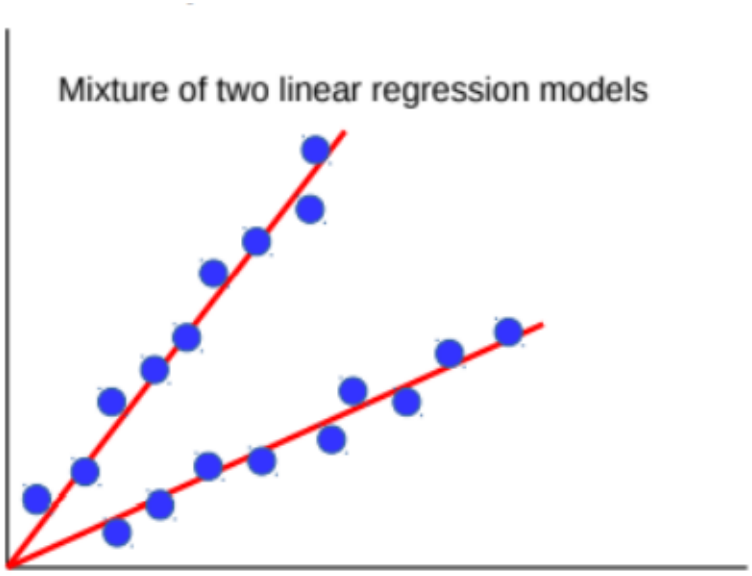
Therefore, λ_l for cluster 'l' is the sample average for cluster l.

Introduction to ML (CS771), Autumn 2020
Indian Institute of Technology Kanpur
Bonus Assignment 1

Student Name: Deeksha Arora
Roll Number: 20111017
Date: December 19, 2020

QUESTION 3

This model will learn more than one weight vector to model the given data. In many cases a single linear curve may not fit the data efficiently and we need more than one linear curve to model the data. In such cases this model will group the data into different clusters and will learn a different linear model for data present within each cluster. At test time, first the cluster of the test input is identified and then prediction is made using the linear model of that class. This method also reduces the impact of outliers on the linear curve because outliers may get separate out due to clustering. Hence this model is expected to work better than a standard probabilistic model.
The following image shows a possible scenario where latent variable model for regression is useful:



EM Algorithm:

Given : $z_n \sim \text{multinoulli}\,(\pi_1, \ldots, \pi_K)$, $\boldsymbol{x}_n \sim \mathcal{N}\left(\mu_{z_n}, \boldsymbol{\Sigma}_{z_n}\right)$ and $y_n \sim \mathcal{N}\left(\boldsymbol{w}_{z_n}^\top \boldsymbol{x}_n, \beta^{-1}\right)$.
Therefore for the latent variable model:
Prior Probability, $p\left(z_n = k\right) = \pi_k$

Likelihood, $p\left(y_n \mid z_n, \theta\right) = \mathcal{N}\left(\mathbf{w}_{z_n}^\top \mathbf{x}_n, \beta^{-1}\right)$ and $p\left(\mathbf{x}_n \mid z_n, \theta\right) = \mathcal{N}\left(\mu_{z_n}, \boldsymbol{\Sigma}_{z_n}\right)$
Here, we have two likelihood terms unlike GMM.

Conditional Posterior Probability,

$$p\left(z_n = k \mid \boldsymbol{x_n}, y_n, \theta\right) = p\left(z_n = k\right) p\left(\boldsymbol{x_n} \mid z_n = k\right) p\left(y_n \mid z_n = k, \boldsymbol{x_n}\right) \tag{5}$$

$$= \pi_k \mathcal{N}\left(\boldsymbol{x_n} \mid \boldsymbol{\mu_k}, \Sigma_k\right) \mathcal{N}\left(y_n \mid w_n^\top \boldsymbol{x_n}, \beta^{-1}\right) \tag{6}$$

Computing $\mathbb{E}[z_{nk}]$:

$$\mathbb{E}[z_{nk}] \propto \pi_k \, p(\boldsymbol{x_n} \mid z_n = k) \, p(y_n \mid z_n = k, \boldsymbol{x_n}) \tag{7}$$

$$\propto \pi_k \mathcal{N}(\boldsymbol{x_n} \mid \boldsymbol{\mu_k}, \Sigma_k) \, \mathcal{N}\left(y_n \mid w_n^\top \boldsymbol{x_n}, \beta^{-1}\right) \tag{8}$$

$$= \frac{\pi_k \mathcal{N}(\boldsymbol{\mu_k}, \Sigma_k) \, \mathcal{N}\left(\mathbf{w}_k^T \mathbf{x}_n, \beta^{-1}\right)}{\sum_{l=1}^{K} \pi_l \mathcal{N}(\boldsymbol{\mu_k}, \Sigma_k) \, \mathcal{N}\left(\mathbf{w}_l^T \mathbf{x}_n, \beta^{-1}\right)} \tag{9}$$

Therefore,

$$\mathbb{E}[z_{nk}] = \frac{\pi_k \exp\left(\frac{-1}{2}(\mathbf{x}_n - \mu_k)^T \Sigma_k (\mathbf{x}_n - \mu_k)\right) \exp\left(\frac{-1}{2}\left(y_n - \mathbf{w}_k^T \mathbf{x}_n\right)^T \beta \left(y_n - \mathbf{w}_k^T \mathbf{x}_n\right)\right)}{\sum_{l=1}^{K} \pi_l \exp\left(\frac{-1}{2}(\mathbf{x}_n - \mu_k)^T \Sigma_l (\mathbf{x}_n - \mu_l)\right) \exp\left(\frac{-1}{2}\left(y_n - \mathbf{w}_l^T \mathbf{x}_n\right)^T \beta \left(y_n - \mathbf{w}_l^T \mathbf{x}_n\right)\right)} \tag{10}$$

Complete data log likelihood (CLL) :

$$\log p(X, Y, Z \mid \Theta) = \log \prod_{n=1}^{N} p(\boldsymbol{x_n}, y_n, z_n \mid \theta) \tag{11}$$

$$= \log \prod_{n=1}^{N} \prod_{k=1}^{k} \left[p(\boldsymbol{x_n} \mid z_n = k) \, p(y_n \mid z_n = k, \boldsymbol{x_n}) \, p(z_n = k)\right]^{z_{nk}} \tag{12}$$

$$= \sum_{n=1}^{N} \sum_{k=1}^{k} z_{nk} \left[-\frac{1}{2} \log |\boldsymbol{\Sigma_k}| - \frac{1}{2} (\boldsymbol{x_n} - \boldsymbol{\mu_k})^\top \Sigma_k^{-1} (\boldsymbol{x_n} - \boldsymbol{\mu_k}) - \frac{\beta}{2} \left(y_n - \boldsymbol{w_k}^\top \boldsymbol{x_n}\right)^2 + \log \pi_k\right] \tag{13}$$

Therefore, the expression for CLL (after ignoring constants) is given by equation 12.

The EM Algorithm

1. Initialize Θ as $\Theta^{(0)}$, set t=1.

2. **The Expectation Step :** Compute the expectation of each z_n using equation 8.

$$\mathbb{E}\left[z_{nk}^{(t)}\right] = \frac{\pi_k^{(t-1)} \mathcal{N}\left(\boldsymbol{\mu_k}^{(t-1)}, \Sigma_k^{(t-1)}\right) \mathcal{N}\left(\mathbf{x}_n^T \mathbf{w}_k^{(t-1)}, \beta^{-1}\right)}{\sum_{l=1}^{K} \pi_l^{(t-1)} \mathcal{N}\left(\boldsymbol{\mu_k}^{(t-1)}, \Sigma_k^{(t-1)}\right) \mathcal{N}\left(\mathbf{x}_n^T \mathbf{w}_l^{(t-1)}, \beta^{-1}\right)}$$

$$= \frac{\pi_k^{(t-1)} \exp\left(\frac{-1}{2}\left(\mathbf{x}_n - \mu_k^{(t-1)}\right)^T \Sigma_k^{(t-1)} \left(\mathbf{x}_n - \mu_k^{(t-1)}\right)\right) \exp\left(\frac{-1}{2}\left(y_n - \mathbf{w}_k^{(t-1)T} \mathbf{x}_n\right)^T \beta \left(y_n - \mathbf{w}_k^{(t-1)T} \mathbf{x}_n\right)\right))}{\sum_{l=1}^{K} \pi_l^{(t-1)} \exp\left(\frac{-1}{2}\left(\mathbf{x}_n - \mu_k^{(t-1)}\right)^T \Sigma_l^{(t-1)} \left(\mathbf{x}_n - \mu_l^{(t-1)}\right)\right) \exp\left(\frac{-1}{2}\left(y_n - \mathbf{w}_l^{(t-1)T} \mathbf{x}_n\right)^T \beta \left(y_n - \mathbf{w}_l^{(t-1)T} \mathbf{x}_n\right)\right.}$$

3. **The Maximization Step :** Update Θ by maximizing the Complete data log likelihood (CLL)

$$\widehat{\Theta} = \text{argmax}_\Theta \, \mathbb{E}_{p(\mathbf{Z} \mid \mathbf{X}, \mathbf{Y}, \widehat{\Theta})}[\log p(\mathbf{X}, \mathbf{Y}, \mathbf{Z} \mid \Theta)] \tag{14}$$

$$= \text{argmax}_\Theta \sum_{n=1}^{N} \mathbb{E}_{p(z_n \mid \mathbf{x}_n y_n, \widehat{\Theta})} [\log p(\mathbf{x}_n, y_n, z_n \mid \Theta)] \tag{15}$$

Differentiating the above equation gives the following MLE estimate of Θ:
Suppose $\gamma_{nk} = \mathbb{E}[z_{nk}]$, and $N_k = \sum_{n=1}^{N} \gamma_{nk}$

$$\pi_k = N_k/N \tag{16}$$

$$\boldsymbol{\mu_k} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} \boldsymbol{x_n} \tag{17}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_K} \sum_{n=1}^{N} \gamma_{nk} (\boldsymbol{x_n} - \boldsymbol{\mu_k}) (\boldsymbol{x_n} - \boldsymbol{\mu_k})^T \tag{18}$$

$$\hat{\mathbf{w}_k} = \left(\sum_{n=1}^{N} \gamma_{nk} \boldsymbol{x_n} \boldsymbol{x_n}^\top \right)^{-1} \left(\sum_{n=1}^{N} \gamma_{nk}\, y_n\, \boldsymbol{x_n} \right) \tag{19}$$

4. Set t=t+1 and go to step 2 if not yet converged.

Intuitive sense of update equation of each weight vector:
In the update expression of $\mathbf{w}_k$, each training input is multiplied by γ_{nk} which denotes the expectation of training input n belonging to cluster k. Therefore, the contribution of each input in the updation of weight vector of class k depends on the probability of input n belonging to class k and logically this makes sense because the linear model of a class will depend more on the values of points belonging to the class.

The ALT-OPT Algorithm :
In Alt-Opt algorithm, we assume a "hard" (most probable) guess of z_n.

1. Initialize Θ as $\hat{\Theta}$.
2. For $n = 1, \ldots . N$ compute the most probable value of z_n. Since $\pi_k = \frac{1}{K}$ for all $k \in \{1, \ldots K\}$, therefore, we don't need to compute π_k. Thus, the update equation for z_n can be written as :

$$\hat{z_n} = \text{argmin}_{k=1,\ldots . K} \left[-\log p(\boldsymbol{x_n} \mid z_n = k) - \log p(y_n \mid z_n = k, \boldsymbol{x_n}) \right] \tag{20}$$

$$= \text{argmin}_{k=1,\ldots . K} \left[\frac{1}{2} \log |\boldsymbol{\Sigma}| + \frac{1}{2} (\mathbf{x}_n - \mu_k)^T \Sigma_k^{-1} (\mathbf{x}_n - \mu_k) + \frac{1}{2} \left(y_n - \mathbf{w}_k^T \mathbf{x}_n \right)^2 \right] \tag{21}$$

   The optimal values of z_n can be used to create one-hot vector $\gamma_n = [\gamma_{n1}, \ldots, \gamma_{nk}]$

3. Re-estimate Θ using MLE, given $\mathbf{Z} = \{\hat{z}_1, \ldots . \hat{z}_n\}$,

$$\hat{\Theta} = \text{argmax}_{\Theta} \log p(\mathbf{X}, \mathbf{Y}, \hat{\mathbf{Z}} \mid \Theta) \tag{22}$$

$$= \text{argmax}_{\Theta} \sum_{n=1}^{N} \sum_{k=1}^{K} \hat{z}_{nk} \left[\log p(\boldsymbol{x_n} \mid z_n = k) + \log p(y_n \mid z_n = k, \boldsymbol{x_n}) \right] \tag{23}$$

   The updates will be same as EM except that here γ_n is a one-hot vector and only the points with $\gamma_{nk} = 1$ will contribute to the updates. Therefore,

$$\boldsymbol{\mu_k} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} \boldsymbol{x_n} \tag{24}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_K} \sum_{n=1}^{N} \gamma_{nk} \left(\boldsymbol{x_n} - \boldsymbol{\mu_k}\right) \left(\boldsymbol{x_n} - \boldsymbol{\mu_k}\right)^T \tag{25}$$

$$\hat{\mathbf{w}}_k = \left(\sum_{n=1}^{N} \gamma_{nk} \boldsymbol{x_n} \boldsymbol{x_n}^\top \right)^{-1} \left(\sum_{n=1}^{N} \gamma_{nk} \, y_n \, \boldsymbol{x_n} \right) \tag{26}$$

4. Go to step 2 if not yet converged