Homework 3

Luis Vitor Zerkowski - 14895730

October 18, 2023

1 Principal Component Analysis

1.a

We start by computing the sample mean. So for a generic feature $d \in 1, \ldots, D$ we have:

$$\bar{x}_d = \frac{1}{N} \sum_{n=1}^{N} x_{nd} \qquad \text{(I)}$$

The mean vector is then given by $\bar{\boldsymbol{x}} = (\bar{x}_1, \ldots, \bar{x}_D)^T$. Now $\hat{\boldsymbol{x}}_n$ can be expressed as:

$$\hat{\boldsymbol{x}}_n = \boldsymbol{x}_n - \bar{\boldsymbol{x}} \qquad \text{(II)}$$

1.b

To prove that the average of $\hat{\boldsymbol{x}}_n$ over N data vectors is the $\mathbf{0}$ vector, we can simply prove that the average of any given feature of $\hat{\boldsymbol{x}}_n$ is zero. So for a feature $d \in 1, \ldots, D$ we do:

$$\bar{\hat{x}}_d = \frac{1}{N} \sum_{n=1}^{N} \hat{x}_{nd} =$$

From expression (II) provided on **1.a**:

$$= \frac{1}{N} \sum_{n=1}^{N} (x_{nd} - \bar{x}_d) = \frac{1}{N} \sum_{n=1}^{N} x_{nd} - \frac{1}{N} \sum_{n=1}^{N} \bar{x}_d = \frac{1}{N} \sum_{n=1}^{N} x_{nd} - \bar{x}_d =$$

Now from expression (I):

$$= \frac{1}{N} \sum_{n=1}^{N} x_{nd} - \frac{1}{N} \sum_{n=1}^{N} x_{nd} = 0$$

So the average value of any feature $d \in 1, \ldots, D$ of $\hat{\boldsymbol{x}}_n$ is zero, thus the average of $\hat{\boldsymbol{x}}_n$ over N data vectors is indeed the $\mathbf{0}$ vector.

1.c

Say $\hat{\boldsymbol{X}}_i$ indicates the i-th column of the $\hat{\boldsymbol{X}}$ matrix. The covariance of the sample is given by:

$$Cov(\hat{\boldsymbol{X}}_i, \hat{\boldsymbol{X}}_j) = \frac{1}{N} \sum_{n=1}^{N} (\hat{x}_{ni} - \bar{\hat{x}}_i)(\hat{x}_{nj} - \bar{\hat{x}}_j) =$$

But we now from **1.b** that the average value of any feature $d \in 1, \ldots, D$ of $\hat{\boldsymbol{x}}_n$ is zero, so we have:

$$= \frac{1}{N} \sum_{n=1}^{N} \hat{x}_{ni} \hat{x}_{nj}$$

Converting the above expression to matrix notation, we get:

$$\boldsymbol{S} = \frac{1}{N} \hat{\boldsymbol{X}}^T \hat{\boldsymbol{X}}$$

1.d

$\boldsymbol{S}$ is the sample covariance, a matrix indicating how much a feature $i \in 1, \ldots, D$ varies with another feature $j \in 1, \ldots, D$, so $\boldsymbol{S} \in \mathbb{R}^{D \times D}$.

1.e

1.e.1

Considering the eigenvalue decomposition $\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$, we can use the first K columns of $\boldsymbol{U}$ to project the data onto the K-dimensional sub-space. We can also use the first K eigenvalues to normalize the projection, so the first K rows and K columns of $\boldsymbol{\Lambda}$. Denoting them respectively by $\boldsymbol{U}_K$ and $\boldsymbol{\Lambda}_{K \times K}$, we have:

$$\boldsymbol{y}_n = \boldsymbol{\Lambda}_{K \times K}^{-\frac{1}{2}} \boldsymbol{U}_K^T \hat{\boldsymbol{x}}_n \implies \boldsymbol{L} = \boldsymbol{\Lambda}_{K \times K}^{-\frac{1}{2}} \boldsymbol{U}_K^T$$

1.e.2

We start by computing the average of a generic feature $k \in 1, \ldots, K$ of $\boldsymbol{y}_n$ over N. So we have:

$$\bar{y}_k = \frac{1}{N} \sum_{n=1}^{N} y_{nk} =$$

Since the inverse of a diagonal matrix is given by a diagonal matrix with the matrix' reciprocal values, we have $\boldsymbol{\Lambda}_{K \times K}^{-\frac{1}{2}} = \begin{bmatrix} \frac{1}{\sqrt{\lambda_1}} & 0 & \ldots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \ldots & \frac{1}{\sqrt{\lambda_K}} \end{bmatrix} \in \mathbb{R}^{K \times K}$. Now using the equation from **1.e.1** we get:

$$= \frac{1}{N} \sum_{n=1}^{N} \frac{\boldsymbol{u}_k^T \hat{\boldsymbol{x}}_n}{\sqrt{\lambda_k}} = \frac{\boldsymbol{u}_k^T}{\sqrt{\lambda_k}} \frac{1}{N} \sum_{n=1}^{N} \hat{\boldsymbol{x}}_n =$$

But we know from **1.b** that the average of $\hat{\boldsymbol{x}}_n$ is the $\boldsymbol{0}$ vector, so we have:

$$= \frac{\boldsymbol{u}_k^T}{\sqrt{\lambda_k}} \boldsymbol{0} = 0$$

And since we proved the average of any feature k of $\boldsymbol{y}_n$ over N is zero, we know that the average of $\boldsymbol{y}_n$ is also the $\boldsymbol{0}$ vector.

1.e.3

Calling the matrix $\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}_1^T \\ \vdots \\ \boldsymbol{y}_N^T \end{bmatrix}$, and indicating $\boldsymbol{Y}_i$ as the i-th column of $\boldsymbol{Y}$, we compute the covariance of the sample by:

$$Cov(\boldsymbol{Y}_i, \boldsymbol{Y}_j) = \frac{1}{N} \sum_{n=1}^{N} (y_{ni} - \bar{y}_i)(y_{nj} - \bar{y}_j) =$$

But since the average of $\boldsymbol{y}_n$ is the $\boldsymbol{0}$ vector, we get:

$$= \frac{1}{N} \sum_{n=1}^{N} y_{ni} y_{nj} = \frac{1}{N} \boldsymbol{Y}^T \boldsymbol{Y} =$$

Using the equation we got from **1.e.1** again, we have:

$$= \frac{1}{N} \left(\hat{\boldsymbol{X}} \boldsymbol{U}_K \boldsymbol{\Lambda}_{K \times K}^{-\frac{1}{2}} \right)^T \left(\hat{\boldsymbol{X}} \boldsymbol{U}_K \boldsymbol{\Lambda}_{K \times K}^{-\frac{1}{2}} \right) = \frac{1}{N} \left(\boldsymbol{\Lambda}_{K \times K}^{-\frac{1}{2}} \right)^T \boldsymbol{U}_K^T \hat{\boldsymbol{X}}^T \hat{\boldsymbol{X}} \boldsymbol{U}_K \boldsymbol{\Lambda}_{K \times K}^{-\frac{1}{2}} =$$

$$= \left(\boldsymbol{\Lambda}_{K \times K}^{-\frac{1}{2}} \right)^T \boldsymbol{U}_K^T \boldsymbol{S} \boldsymbol{U}_K \boldsymbol{\Lambda}_{K \times K}^{-\frac{1}{2}} =$$

Now using the given eigenvalue decomposition for $\boldsymbol{S}$, we have:

$$= \left(\boldsymbol{\Lambda}_{K\times K}^{-\frac{1}{2}}\right)^T \boldsymbol{U}_K^T \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T\boldsymbol{U}_K\boldsymbol{\Lambda}_{K\times K}^{-\frac{1}{2}} =$$

Since the computed eigenvectors are orthonormal, $\boldsymbol{U}_K^T\boldsymbol{U} = \boldsymbol{A} \in \mathbb{R}^{K\times D}$ with $\boldsymbol{A}$ being the identity matrix on the first K rows and K columns and zero for the rest of the entries. Analogously $\boldsymbol{U}^T\boldsymbol{U}_K = \boldsymbol{A}^T \in \mathbb{R}^{D\times K}$, still containing the identity matrix on the first K rows and K columns. The multiplication $\boldsymbol{U}_K^T\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T\boldsymbol{U}_K$ thus gives us the first K rows and K column of the matrix $\boldsymbol{\Lambda}$, which we already denoted as $\boldsymbol{\Lambda}_{K\times K}$. So we get:

$$= \left(\boldsymbol{\Lambda}_{K\times K}^{-\frac{1}{2}}\right)^T \boldsymbol{\Lambda}_{K\times K}\boldsymbol{\Lambda}_{K\times K}^{-\frac{1}{2}} =$$

Knowing that the multiplication of diagonal matrices is just a diagonal matrix with their entries multiplied and also that a diagonal matrix is always symmetric, we finally have:

$$= \boldsymbol{\Lambda}_{K\times K}^{-\frac{1}{2}}\boldsymbol{\Lambda}_{K\times K}\boldsymbol{\Lambda}_{K\times K}^{-\frac{1}{2}} = \begin{bmatrix} \frac{1}{\sqrt{\lambda_1}}\lambda_1\frac{1}{\sqrt{\lambda_1}} & 0 & \dots & 0 \\ 0 & \frac{1}{\sqrt{\lambda_2}}\lambda_2\frac{1}{\sqrt{\lambda_2}} & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & \frac{1}{\sqrt{\lambda_K}}\lambda_K\frac{1}{\sqrt{\lambda_K}} \end{bmatrix} =$$

$$= \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & 1 \end{bmatrix} \in \mathbb{R}^{K\times K}$$

And we have proved that the covariance of $\boldsymbol{y}_n$ is indeed the identity matrix.

1.e.4

The operator $\boldsymbol{L}$ is known as **whitening** operator, since it centers the features, de-correlates them and cast them to unit standard deviation. Because of the unit standard deviation part, it is also called **sphering**.

1.f

1.f.1

If you make K bigger, you include more eigenvectors to the projection, leading to a higher dimensional sub-space that incorporates more variance from the original D-dimensional space and hence less loss of information. Regarding the complexity, it is natural that increasing K leads to more computation, since finding more eigenvectors is more computationally expensive and also the projection itself involves more inner products.

Analogously decreasing K leads to less variance incorporated to the sub-space, more loss of information and less computational cost.

1.f.2

The greatest trade-off involving K and D is related to dimensionality reduction $\times$ loss of information. The idea of the method is to reduce the number of dimensions of the original data, making it much easier to work with, whether for model training or just for visualization. In this sense, the computational cost reduction play a super important role.

On the other hand, if you reduce the dimensionality of the data too much, you start accounting for too little variance of the original data. The result is a lot of information loss, leading to a bad model or a meaningless visualization. Finding the right K thus can be a challenge on its own.

1.g

1.g.1

A good way to learn these non-linear projections is through neural networks. We could use, for example, an autoencoder. The idea is having a network that encodes the data, projecting it onto a smaller feature space (the encoder), and then another network that uses the encoded features to reconstruct the information, projecting the data onto a feature space with the same amount of features as the initial one (the decoder). By making the reconstruction error as small as possible, we train the encoding network to learn non-linear projections that keeps the core information of the data in a smaller feature space and then train the decoder network to learn non-linear projections that uses the stored core information to reconstruct an output as similar as possibile to the original data.

After training this model, the output of the encoder network should give you a non-linear adaptation of PCA and thus provide a reasonable dimensionality reduction tool.

1.g.2

The natural pro is that you can use the method to project the data non-linearly, which gives you much more power on dimensionality reduction since it's very common to have data for which linear projections don't reduce the dimensionality of the feature space significantly without too much information loss.

For the cons, they are mostly related to the training process. As the autoencoder is a neural network, it is much more computationally expensive to train it and it is also very much sensitive to the quality and amount of data, which can easily lead to overfitting. With an overfitted neural network, the dimensionality reduction is useless since will not generalize well to unseen data.

1.h

Steps **2** and **3** are the slowest in PCA and their complexities are $\mathcal{O}(ND^2)$ for the $\hat{\boldsymbol{X}}^T\hat{\boldsymbol{X}}$ multiplication and $\mathcal{O}(D^3)$ for the eigenvectors decomposition. If $N > D$, the algorithm's complexity is dominated by $\hat{\boldsymbol{X}}^T\hat{\boldsymbol{X}}$, so $\mathcal{O}(ND^2)$. On the other hand, if $D > N$, the algorithm's complexity is dominated by the eigenvectors decomposition, so $\mathcal{O}(D^3)$.

Analyzing the complexities of the most computationally expensive steps, we observe that they increase quadratically and cubically with the number of dimensions. This is very problematic and can represent an impediment for high-dimensional data - images for example. To deal with this situation, we can try choosing a much smaller K, which will decrease cubically the time consumption for step **3**, or use a batch version of PCA, which will decrease linearly the time consumption for step **2**.

1.i

Linear discriminant analysis could be used as a dimensionality reduction method. The idea of the model is to maximize separability between classes by using discriminant projections that maximize the distance between projected means and minimize the projected variance within each class. The number of dimensions of the data is thus reduced to the number of discriminants ($C - 1$ for C classes).

1.j

It is clear that the first projection axis better divides the data since it separates the means for each class as well as the other projection axis, but also gives a much smaller intra-class variance than the other projection axis. Visually speaking, it is also very much noticeable that the data is better split with the first projection axis since the cloud of points for each class don't overlap for this projection and overlaps quite a lot for the second projection axis.

To compute this projection with LDA, we have to define the within-class variation given by the sum of the covariance matrices and the between-classes variation given by the difference between the mean of the classes, so:

$$\boldsymbol{S}_{WC} = \boldsymbol{S}_1 + \boldsymbol{S}_2$$

$$\boldsymbol{S}_{BC} = (\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{m}_2 - \boldsymbol{m}_1)^T$$

With these values in hand, we want to compute the projection that minimizes the within-class variation and maximizes the between-classes variation. This projection is given by the eigenvector with the largest eigenvalue of $\boldsymbol{S}_{WC}^{-1}\boldsymbol{S}_{BC}$, so the eigenvalue problem of:

$$\boldsymbol{S}_{WC}^{-1}\boldsymbol{S}_{BC}\boldsymbol{w} = \lambda\boldsymbol{w}$$

1.k

As said in the previous question, the computation is given by getting the eigenvector corresponding to the largest eigenvalue of:

$$\boldsymbol{S}_{WC}^{-1}\boldsymbol{S}_{BC}\boldsymbol{w} = \lambda\boldsymbol{w}$$

Substituting $\boldsymbol{S}_{WC}$ and $\boldsymbol{S}_{BC}$, we have:

$$(\boldsymbol{S}_1 + \boldsymbol{S}_2)^{-1}\left((\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{m}_2 - \boldsymbol{m}_1)^T\right)\boldsymbol{w} = \lambda\boldsymbol{w}$$

2 Probabilistic PCA - A General Latent Space Distribution

2.a

Using the given expression for $\boldsymbol{z}$, we can express $\boldsymbol{x}$ as:

$$\boldsymbol{x} = \boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}_x =$$
$$= \boldsymbol{W}(\boldsymbol{m} + \boldsymbol{\epsilon}_z) + \boldsymbol{\mu} + \boldsymbol{\epsilon}_x = \boldsymbol{W}\boldsymbol{m} + \boldsymbol{\mu} + \boldsymbol{W}\boldsymbol{\epsilon}_z + \boldsymbol{\epsilon}_x$$

Considering the different ways of writing out a Gaussian distribution and the resulting expression we got, we can understand $\boldsymbol{x}$ as a Gaussian distribution with mean $\boldsymbol{W}\boldsymbol{m}+\boldsymbol{\mu}$ and noise given by $\boldsymbol{W}\boldsymbol{\epsilon}_z+\boldsymbol{\epsilon}_x$.

And now going back to the original expression of $x = \boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}_x$ and considering the given conditional distribution $p(\boldsymbol{x}|\boldsymbol{z}) = N(\boldsymbol{x}|\boldsymbol{W}\boldsymbol{z}+\boldsymbol{\mu}, \sigma^2\boldsymbol{I})$, we know that $\boldsymbol{\epsilon}_x$ must be a Gaussian distribution with zero mean and covariance given by $\sigma^2\boldsymbol{I}$, so $\boldsymbol{\epsilon}_x \sim N(0, \sigma^2\boldsymbol{I})$.

2.b

$$E[\boldsymbol{x}] = E[\boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}_x] =$$

By the linearity property, we know that $E[X + Y] = E[X] + E[Y]$ for two random variables X and Y, and also that $E[\alpha X] = \alpha E[X]$ for the random variable X and the constant α. So we have:

$$= \boldsymbol{W}E[\boldsymbol{z}] + E[\boldsymbol{\mu}] + E[\boldsymbol{\epsilon}_x] =$$

But we know the distribution of both $\boldsymbol{z}$ and $\boldsymbol{\epsilon}_x$, and the expected value of a vector is just the vector, so we get:

$$= \boldsymbol{W}\boldsymbol{m} + \boldsymbol{\mu} + 0 = \boldsymbol{W}\boldsymbol{m} + \boldsymbol{\mu}$$

So the expected value of $\boldsymbol{x}$ is $E[\boldsymbol{x}] = \boldsymbol{W}\boldsymbol{m} + \boldsymbol{\mu}$.

2.c

The covariance of a variable with itself is exactly the variance of the variable. So we proceed by computing the variance of $\boldsymbol{x}$. Using the form we got from exercise **2.a**, we have:

$$Var[\boldsymbol{x}] = Var[\boldsymbol{W}\boldsymbol{m} + \boldsymbol{\mu} + \boldsymbol{W}\boldsymbol{\epsilon}_z + \boldsymbol{\epsilon}_x] =$$

By using the property of variance that $Var[X + \alpha] = Var[X]$ for a random variable X and a constant α, we can get rid of the constant part $\boldsymbol{W}\boldsymbol{m} + \boldsymbol{\mu}$. So we get:

$$= Var[\boldsymbol{W}\boldsymbol{\epsilon}_z + \boldsymbol{\epsilon}_x] =$$

Now considering that the variance of the sum of independent random variables is equal to the sum of the variance of independent random variables, we have:

$$= Var[\boldsymbol{W}\boldsymbol{\epsilon}_z] + Var[\boldsymbol{\epsilon}_x] =$$

Using the given identity $Var[\boldsymbol{A}\boldsymbol{Y}] = \boldsymbol{A}Var[\boldsymbol{Y}]\boldsymbol{A}^T$ where $\boldsymbol{A}$ is a matrix and $\boldsymbol{Y}$ is a random variable along with the fact we know the distributions for both $\boldsymbol{\epsilon}_z$ and $\boldsymbol{\epsilon}_x$, we get:

$$= \boldsymbol{W}Var[\boldsymbol{\epsilon}_z]\boldsymbol{W}^T + \sigma^2\boldsymbol{I} = \boldsymbol{W}\boldsymbol{\Sigma}\boldsymbol{W}^T + \sigma^2\boldsymbol{I}$$

So the covariance of $\boldsymbol{x}$ is $Cov[\boldsymbol{x}, \boldsymbol{x}] = Var[\boldsymbol{x}] = \boldsymbol{W}\boldsymbol{\Sigma}\boldsymbol{W}^T + \sigma^2\boldsymbol{I}$.

Just to make the answer self-contained, we prove some identities we used. Consider a random variable X and a constant α:

$$Var[X + \alpha] = E[(X + \alpha)^2] - E[X + \alpha]^2 = E[X^2 + 2\alpha X + \alpha^2] - E[X + \alpha]^2 =$$

By the linearity of the expected value:

$$= E[X^2] + 2\alpha E[X] + E[\alpha^2] - (E[X] + E[\alpha])^2 =$$
$$= E[X^2] + 2\alpha E[X] + \alpha^2 - E[X]^2 - 2\alpha E[X] - \alpha^2 =$$
$$= E[X^2] - E[X]^2 = Var[X]$$

For the second result, we consider two independent random variables X and Y:

$$Var[X + Y] = E[(X + Y)^2] - E[X + Y]^2 = E[X^2 + 2XY + Y^2] - E[X + Y]^2 =$$

Again by the linearity of the expected value:

$$= E[X^2] + 2E[XY] + E[Y^2] - (E[X] + E[Y])^2 =$$
$$= E[X^2] + 2E[XY] + E[Y^2] - E[X]^2 - 2E[x]E[Y] - E[Y]^2 =$$

But since X and Y are independent, one is a constant to the other, which means we can do $E[XY] = E[X]E[Y]$. So we get:

$$= E[X^2] + 2E[X]E[Y] + E[Y^2] - E[X]^2 - 2E[x]E[Y] - E[Y]^2 =$$
$$= E[X^2] - E[X]^2 + E[Y^2] - E[Y]^2 = Var[X] + Var[Y]$$

2.d

Since we have found that the expected value of $\boldsymbol{x}$ is $E[\boldsymbol{x}] = \boldsymbol{W}\boldsymbol{m} + \boldsymbol{\mu}$ and the covariance of $\boldsymbol{x}$ is $Cov[\boldsymbol{x}, \boldsymbol{x}] = Var[\boldsymbol{x}] = \boldsymbol{W}\boldsymbol{\Sigma}\boldsymbol{W}^T + \sigma^2\boldsymbol{I}$, we simply assume $\tilde{\boldsymbol{\mu}} = \boldsymbol{W}\boldsymbol{m} + \boldsymbol{\mu}$ and $\tilde{\boldsymbol{W}} = \boldsymbol{W}\boldsymbol{\Sigma}^{\frac{1}{2}}$, with $\boldsymbol{\Sigma}^{\frac{1}{2}}$ a matrix such that $\boldsymbol{\Sigma}^{\frac{1}{2}}(\boldsymbol{\Sigma}^{\frac{1}{2}})^T = \boldsymbol{\Sigma}$. With these values, we have:

$$p(\boldsymbol{x}) = N(\boldsymbol{x}|\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{W}}\tilde{\boldsymbol{W}}^T + \sigma^2\boldsymbol{I}) =$$
$$= N(\boldsymbol{x}|\boldsymbol{W}\boldsymbol{m} + \boldsymbol{\mu}, \boldsymbol{W}\boldsymbol{\Sigma}^{\frac{1}{2}}(\boldsymbol{W}\boldsymbol{\Sigma}^{\frac{1}{2}})^T + \sigma^2\boldsymbol{I}) = N(\boldsymbol{x}|\boldsymbol{W}\boldsymbol{m} + \boldsymbol{\mu}, \boldsymbol{W}\boldsymbol{\Sigma}^{\frac{1}{2}}(\boldsymbol{\Sigma}^{\frac{1}{2}})^T\boldsymbol{W}^T + \sigma^2\boldsymbol{I}) =$$
$$= N(\boldsymbol{x}|\boldsymbol{W}\boldsymbol{m} + \boldsymbol{\mu}, \boldsymbol{W}\boldsymbol{\Sigma}\boldsymbol{W}^T + \sigma^2\boldsymbol{I})$$

3 Mixtures of Experts

3.a

Since $\boldsymbol{z}_n$ is one-hot encoded, we just need to compute take the k that maximizes the expression $p(z_{nk} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi}) = \frac{\exp(\boldsymbol{\phi}_k^T \boldsymbol{x}_n)}{\sum_j \exp(\boldsymbol{\phi}_j^T \boldsymbol{x}_n)}$. Considering the fact, they all have the same denominator for a given data point $\boldsymbol{x}_n$, we just need to take the k that maximizes $\exp(\boldsymbol{\phi}_k^T \boldsymbol{x}_n)$. So we have:

$$z_{nk} = \begin{cases} 1, & \text{if } k = \arg\max \exp(\boldsymbol{\phi}_k^T \boldsymbol{x}_n) \\ 0, & \text{otherwise} \end{cases}$$

3.b

Using the i.i.d assumption, we have:

$$p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\Theta}, \boldsymbol{\Phi}) = \prod_{n=1}^{N} p(y_n|\boldsymbol{x}_n, \boldsymbol{\Theta}, \boldsymbol{\Phi}) =$$

Marginalizing over all the experts, we get:

$$= \prod_{n=1}^{N} \sum_{k=1}^{K} p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_k) p(z_{nk} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi}) =$$

Using the given predictive model and the routing mechanism, we have:

$$= \prod_{n=1}^{N} \sum_{k=1}^{K} \exp(\boldsymbol{\theta}_k^T \boldsymbol{x}_n) \exp(-\exp(\boldsymbol{\theta}_k^T \boldsymbol{x}_n) y_n) \pi_{nk} =$$

$$= \prod_{n=1}^{N} \sum_{k=1}^{K} \exp(\boldsymbol{\theta}_k^T \boldsymbol{x}_n - \exp(\boldsymbol{\theta}_k^T \boldsymbol{x}_n) y_n) \frac{\exp(\boldsymbol{\phi}_k^T \boldsymbol{x}_n)}{\sum_j \exp(\boldsymbol{\phi}_j^T \boldsymbol{x}_n)} =$$

$$= \prod_{n=1}^{N} \sum_{k=1}^{K} \frac{\exp(\boldsymbol{\theta}_k^T \boldsymbol{x}_n - \exp(\boldsymbol{\theta}_k^T \boldsymbol{x}_n) y_n + \boldsymbol{\phi}_k^T \boldsymbol{x}_n)}{\sum_j \exp(\boldsymbol{\phi}_j^T \boldsymbol{x}_n)}$$

So the likelihood is given by $p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\Theta}, \boldsymbol{\Phi}) = \prod_{n=1}^{N} \sum_{k=1}^{K} \frac{\exp(\boldsymbol{\theta}_k^T \boldsymbol{x}_n - \exp(\boldsymbol{\theta}_k^T \boldsymbol{x}_n) y_n + \boldsymbol{\phi}_k^T \boldsymbol{x}_n)}{\sum_j \exp(\boldsymbol{\phi}_j^T \boldsymbol{x}_n)}$. Finally for the log-likelihood, we get:

$$\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\Theta}, \boldsymbol{\Phi}) = \log \left(\prod_{n=1}^{N} \sum_{k=1}^{K} \frac{\exp(\boldsymbol{\theta}_k^T \boldsymbol{x}_n - \exp(\boldsymbol{\theta}_k^T \boldsymbol{x}_n) y_n + \boldsymbol{\phi}_k^T \boldsymbol{x}_n)}{\sum_j \exp(\boldsymbol{\phi}_j^T \boldsymbol{x}_n)} \right) =$$

$$= \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} \frac{\exp(\boldsymbol{\theta}_k^T \boldsymbol{x}_n - \exp(\boldsymbol{\theta}_k^T \boldsymbol{x}_n) y_n + \boldsymbol{\phi}_k^T \boldsymbol{x}_n)}{\sum_j \exp(\boldsymbol{\phi}_j^T \boldsymbol{x}_n)} \right)$$

So the log-likelihood is given by $\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\Theta}, \boldsymbol{\Phi}) = \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} \frac{\exp(\boldsymbol{\theta}_k^T \boldsymbol{x}_n - \exp(\boldsymbol{\theta}_k^T \boldsymbol{x}_n) y_n + \boldsymbol{\phi}_k^T \boldsymbol{x}_n)}{\sum_j \exp(\boldsymbol{\phi}_j^T \boldsymbol{x}_n)} \right)$.

3.c

Using Bayes theorem, the responsibility of an expert i for a datapoint n is given by:

$$r_{ni} = \frac{p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_i)p(z_{ni} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi})}{\sum_j p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_j)p(z_{nj} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi})} =$$

$$= \frac{\exp(\boldsymbol{\theta}_i^T \boldsymbol{x}_n)\exp(-\exp(\boldsymbol{\theta}_i^T \boldsymbol{x}_n)y_n)\pi_{ni}}{\sum_j \exp(\boldsymbol{\theta}_j^T \boldsymbol{x}_n)\exp(-\exp(\boldsymbol{\theta}_j^T \boldsymbol{x}_n)y_n)\pi_{nj}} =$$

$$= \frac{\exp(\boldsymbol{\theta}_i^T \boldsymbol{x}_n - \exp(\boldsymbol{\theta}_i^T \boldsymbol{x}_n)y_n)\pi_{ni}}{\sum_j \exp(\boldsymbol{\theta}_j^T \boldsymbol{x}_n - \exp(\boldsymbol{\theta}_j^T \boldsymbol{x}_n)y_n)\pi_{nj}}$$

So the responsibility r_{ni} is given by $r_{ni} = \frac{\exp(\boldsymbol{\theta}_i^T \boldsymbol{x}_n - \exp(\boldsymbol{\theta}_i^T \boldsymbol{x}_n)y_n)\pi_{ni}}{\sum_j \exp(\boldsymbol{\theta}_j^T \boldsymbol{x}_n - \exp(\boldsymbol{\theta}_j^T \boldsymbol{x}_n)y_n)\pi_{nj}}$.

3.d

We start by the parameter $\boldsymbol{\theta}_i$:

$$\frac{\partial}{\partial \boldsymbol{\theta}_i} \log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\Theta}, \boldsymbol{\Phi}) = \frac{\partial}{\partial \boldsymbol{\theta}_i} \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_k)p(z_{nk} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi}) \right) =$$

By the sum rule:

$$= \sum_{n=1}^{N} \frac{\partial}{\partial \boldsymbol{\theta}_i} \log \left(\sum_{k=1}^{K} p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_k)p(z_{nk} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi}) \right) =$$

$$= \sum_{n=1}^{N} \frac{1}{\sum_{k=1}^{K} p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_k)p(z_{nk} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi})} \frac{\partial}{\partial \boldsymbol{\theta}_i} \sum_{k=1}^{K} p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_k)p(z_{nk} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi}) =$$

By the sum rule again:

$$= \sum_{n=1}^{N} \frac{\sum_{k=1}^{K} \frac{\partial}{\partial \boldsymbol{\theta}_i}(p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_k)p(z_{nk} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi}))}{\sum_{k=1}^{K} p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_k)p(z_{nk} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi})} = \qquad \text{(I)}$$

Since $p(z_{nk} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi})$ doesn't depend on $\boldsymbol{\theta}_i$, we can consider it a constant when taking the derivative w.r.t $\boldsymbol{\theta}_i$. Also, by the chain rule, we'll end up with a term $\frac{\partial \boldsymbol{\theta}_k}{\partial \boldsymbol{\theta}_i}$, which we can substitute by the Kronecker delta. So we have:

$$= \sum_{n=1}^{N} \frac{\sum_{k=1}^{K} p(z_{nk} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi})\delta_{ik} \frac{\partial}{\partial \boldsymbol{\theta}_i} p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_k)}{\sum_{k=1}^{K} p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_k)p(z_{nk} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi})} =$$

$$= \sum_{n=1}^{N} \frac{p(z_{ni} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi}) \frac{\partial}{\partial \boldsymbol{\theta}_i} p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_i)}{\sum_{k=1}^{K} p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_k)p(z_{nk} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi})} =$$

Using the provided identity, we get:

$$= \sum_{n=1}^{N} \frac{p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_i)p(z_{ni} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi})}{\sum_{k=1}^{K} p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_k)p(z_{nk} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi})} \frac{\partial}{\partial \boldsymbol{\theta}_i} \log p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_i) =$$

$$= \sum_{n=1}^{N} r_{ni} \frac{\partial}{\partial \boldsymbol{\theta}_i} \log p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_i)$$

So now we know that the derivative of the log-likelihood w.r.t $\boldsymbol{\theta}_i$ is given by $\frac{\partial}{\partial \boldsymbol{\theta}_i} \log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\Theta}, \boldsymbol{\Phi}) = \sum_{n=1}^{N} r_{ni} \frac{\partial}{\partial \boldsymbol{\theta}_i} \log p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_i)$.

We proceed to do the same computations for $\boldsymbol{\phi}_i$ and since we were not computing anything particular to $\boldsymbol{\theta}_i$ until equation (I), we can start from an analogous form. So we have:

$$\frac{\partial}{\partial \boldsymbol{\phi}_i} \log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\Theta}, \boldsymbol{\Phi}) = \sum_{n=1}^{N} \frac{\sum_{k=1}^{K} \frac{\partial}{\partial \boldsymbol{\phi}_i}(p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_k)p(z_{nk} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi}))}{\sum_{k=1}^{K} p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_k)p(z_{nk} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi})} =$$

At this point, because of the same arguments as before, we can treat $p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_k)$ as a constant and use a Kronecker delta to get:

$$= \sum_{n=1}^{N} \frac{\sum_{k=1}^{K} p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_k)\delta_{ik} \frac{\partial}{\partial \boldsymbol{\phi}_i} p(z_{nk} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi})}{\sum_{k=1}^{K} p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_k) p(z_{nk} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi})} =$$

$$= \sum_{n=1}^{N} \frac{p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_i) \frac{\partial}{\partial \boldsymbol{\phi}_i} p(z_{ni} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi})}{\sum_{k=1}^{K} p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_k) p(z_{nk} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi})} =$$

By using the provided identity again:

$$= \sum_{n=1}^{N} \frac{p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_i) p(z_{ni} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi}) \frac{\partial}{\partial \boldsymbol{\phi}_i} \log p(z_{ni} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi})}{\sum_{k=1}^{K} p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_k) p(z_{nk} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi})} =$$

$$= \sum_{n=1}^{N} r_{ni} \frac{\partial}{\partial \boldsymbol{\phi}_i} \log p(z_{ni} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi})$$

So now we know that the derivative of the log-likelihood w.r.t $\boldsymbol{\phi}_i$ is given by $\frac{\partial}{\partial \boldsymbol{\phi}_i} \log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\Theta}, \boldsymbol{\Phi}) = \sum_{n=1}^{N} r_{ni} \frac{\partial}{\partial \boldsymbol{\phi}_i} \log p(z_{ni} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi})$.

3.e

So for $\boldsymbol{\theta}_i$ we have:

$$\frac{\partial}{\partial \boldsymbol{\theta}_i} \log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\Theta}, \boldsymbol{\Phi}) = \sum_{n=1}^{N} r_{ni} \frac{\partial}{\partial \boldsymbol{\theta}_i} \log p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_i) =$$

$$= \sum_{n=1}^{N} r_{ni} \frac{\partial}{\partial \boldsymbol{\theta}_i} \log \left(\exp(\boldsymbol{\theta}_i^T \boldsymbol{x}_n) \exp(-\exp(\boldsymbol{\theta}_i^T \boldsymbol{x}_n) y_n) \right) =$$

$$= \sum_{n=1}^{N} r_{ni} \frac{\partial}{\partial \boldsymbol{\theta}_i} \log \exp(\boldsymbol{\theta}_i^T \boldsymbol{x}_n - \exp(\boldsymbol{\theta}_i^T \boldsymbol{x}_n) y_n) =$$

$$= \sum_{n=1}^{N} r_{ni} \frac{\partial}{\partial \boldsymbol{\theta}_i} \left(\boldsymbol{\theta}_i^T \boldsymbol{x}_n - \exp(\boldsymbol{\theta}_i^T \boldsymbol{x}_n) y_n \right) =$$

By the sum rule and already computing the derivatives:

$$= \sum_{n=1}^{N} r_{ni} \left(\boldsymbol{x}_n^T - \exp(\boldsymbol{\theta}_i^T \boldsymbol{x}_n) \boldsymbol{x}_n^T y_n \right) =$$

$$= \sum_{n=1}^{N} r_{ni} \left(1 - \exp(\boldsymbol{\theta}_i^T \boldsymbol{x}_n) y_n \right) \boldsymbol{x}_n^T$$

So we know that $\frac{\partial}{\partial \boldsymbol{\theta}_i} \log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\Theta}, \boldsymbol{\Phi}) = \sum_{n=1}^{N} r_{ni} \left(1 - \exp(\boldsymbol{\theta}_i^T \boldsymbol{x}_n) y_n \right) \boldsymbol{x}_n^T$. We proceed to compute the derivative w.r.t parameter $\boldsymbol{\phi}_i$:

$$\frac{\partial}{\partial \boldsymbol{\phi}_i} \log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\Theta}, \boldsymbol{\Phi}) = \sum_{n=1}^{N} r_{ni} \frac{\partial}{\partial \boldsymbol{\phi}_i} \log p(z_{ni} = 1|\boldsymbol{x}_n, \boldsymbol{\Phi}) =$$

$$= \sum_{n=1}^{N} r_{ni} \frac{\partial}{\partial \boldsymbol{\phi}_i} \log \left(\frac{\exp(\boldsymbol{\phi}_i^T \boldsymbol{x}_n)}{\sum_j \exp(\boldsymbol{\phi}_j^T \boldsymbol{x}_n)} \right) =$$

$$= \sum_{n=1}^{N} r_{ni} \frac{\partial}{\partial \boldsymbol{\phi}_i} \left(\log \exp(\boldsymbol{\phi}_i^T \boldsymbol{x}_n) - \log \sum_j \exp(\boldsymbol{\phi}_j^T \boldsymbol{x}_n) \right) =$$

$$= \sum_{n=1}^{N} r_{ni} \frac{\partial}{\partial \boldsymbol{\phi}_i} \left(\boldsymbol{\phi}_i^T \boldsymbol{x}_n - \log \sum_j \exp(\boldsymbol{\phi}_j^T \boldsymbol{x}_n) \right) =$$

By the sum rule and already computing the derivatives:

$$= \sum_{n=1}^{N} r_{ni} \left(\boldsymbol{x}_n^T - \frac{\exp(\boldsymbol{\phi}_i^T \boldsymbol{x}_n) \boldsymbol{x}_n^T}{\sum_j \exp(\boldsymbol{\phi}_j^T \boldsymbol{x}_n)} \right) = \sum_{n=1}^{N} r_{ni} \left(1 - \pi_{ni}\right) \boldsymbol{x}_n^T$$

So we know that $\frac{\partial}{\partial \boldsymbol{\phi}_i} \log p(\boldsymbol{y} | \boldsymbol{X}, \boldsymbol{\Theta}, \boldsymbol{\Phi}) = \sum_{n=1}^{N} r_{ni} \left(1 - \pi_{ni}\right) \boldsymbol{x}_n^T$.

3.f

1. Randomly initialize the K D-dimensional vectors $\boldsymbol{\theta}_k$ and the K D-dimensional vectors $\boldsymbol{\phi}_k$.
2. Compute the responsibilities r_{nk} for each K expert.
3. Repeat:
   (a) Update vectors $\boldsymbol{\theta}_k$ using MLE from the derivatives we computed. (M-step)
   (b) Update vectors $\boldsymbol{\phi}_k$ using MLE from the derivatives we computed. (M-step)
   (c) Compute the log-likelihood and break loop if ΔLog-Likelihood $< \epsilon$.
   (d) Update responsibilities r_{nk}. (E-step)

3.g

Once we have our parameters and considering the soft assignment of experts, we have:

$$\hat{y}_n = \sum_{k=1}^{K} \pi_{nk} \hat{y}_{nk}$$

For the computation of the prediction of each expert, we can simply sample from the distributions we fitted. Hence $\hat{y}_{nk}$ will be sampled from an exponential distribution with $\lambda = \exp(\boldsymbol{\theta}_k^T \boldsymbol{x}_n)$.