EM for Document clustering:
Q1.1:
For this algorithm, it will be used for incomplete data (documents clusters are not given)

we can identify these unknown as latent variables
$Z \rightarrow Z_1, Z_2, \ldots Z_n$ (unseen data)

the algorithm is initialised by setting parameters as the follow:

$$\theta^{old} = (\varphi^{old}, \mu_1^{old}, \ldots \mu_K^{old})$$

- $\varphi = (\varphi_1 \ldots \varphi_K)$ is cluster proportion, with $\varphi_K \geq 0$ and $\sum_{k=1}^{K} \varphi_k = 1$
- $\mu_K = (\mu_{K,1}, \ldots \mu_{K\|A\|})$ is word proportion for each cluster, with $\mu_{K,w} \geq 0$ and $\sum_{w \in A} \mu_{K,w} = 1$

For Hard-EM, each data is assigned to the class with largest probability (likelihood)

$$Z^* = \text{argmax}_z \, r(z_n, k) = \text{argmax}_z \, p(z_{n,k} = 1 | d_n, \theta^{old})$$

Before getting that value, probability of the observed documents is:

$$p(d_1, \dots d_n) = \prod_{n=1}^{N} p(d_n) = \prod_{n=1}^{N} \sum_{k=1}^{K} p(z_{n,k=1}, d_n)$$

$$= \prod_{n=1}^{N} \sum_{k=1}^{K} \left(\varphi_k \prod_{w \in A} \mu_{k,w}^{c(w,d_n)} \right)$$

Log-likelihood of the above will be:

$$\ln p(d_1, \cdots d_n) = \sum_{n=1}^{N} \ln p(d_n) = \sum_{n=1}^{N} \ln \sum_{k=1}^{K} p(z_{n,k} = 1, d_n)$$

$$= \sum_{n=1}^{N} \ln \sum_{k=1}^{K} \left(\varphi_k \prod_{w \in A} \mu_{k,w}^{c(w,d_n)} \right)$$

To maximise the likelihood we will use Q function as basis of EM to find largest posterior probability.

Q function:

$$Q(\theta, \theta^{old}) = \sum_{n=1}^{N} \sum_{k=1}^{K} p(z_{n,k} = 1 | d_n, \theta^{old}) \ln p(z_{n,k} = 1, d_n | \theta)$$

$$= \sum_{n=1}^{N} \sum_{k=1}^{K} p(z_{n,k} = 1 | d_n, \theta^{old}) \left(\ln \varphi_k + \sum_{w \in A} c(w,d_n) \ln \mu_{k,w} \right)$$

$$= \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{n,k}) \left(\ln \varphi_k + \sum_{w \in A} c(w,d_n) \ln \mu_{k,w} \right)$$

where $\gamma(z_{n,k}) : p(z_{n,k} = 1 | d_n, \theta^{old})$ are responsability factors

Since there is no expectation in over latent variable latent variables in the definition of Q function.

Therefore, Q function

$= Q(\theta, \theta^{old}) = \sum_{n=1}^{N} \ln p(z_{n,k=Z*} = 1, d_n | \theta)$

After the intial setting is set, the process will be executed in a loop, while it is still the not converge:

In the E-step we will set $\forall n$ and $\forall k$ a z^* such as: $Z^* \leftarrow \arg\max_z r(z_n, k)$

$= \arg\max_z p(z_{n,k} = 1 | d_n, \theta^{old})$

For M-step: Maximization of the Q function will be performed by using lagrangian multiplier on $\sum_{k=1}^{K} \varphi_k = 1$ and $\sum_{w \in A} \mu_{k,w} = 1$ and setting the partial derivative to zero we will get:

$\varphi_k = \frac{N_k}{N}$ where $N_k = \sum_{n=1}^{N} r(z_n, k)$

being the cluster proportion $\mu_{k,w} = \frac{\sum_{n=1}^{N} r(z_n, k) c(w, d_n)}{\sum_{w' \in A} \sum_{n=1}^{N} r(z_n, k) c(w', d_n)}$

being word proportion for each cluster

based on the former equations, we will get

$$\theta^{new} \leftarrow \arg\max_\theta Q(\theta, \theta^{old}) = \arg\max_\theta \sum_{n=1}^{N} \ln p(z^*_{n,k=z^*} = 1, d_n | \theta)$$

$$= \arg\max \sum_{n=1}^{N} (\ln \varphi_{k=z^*} + \sum_{w \in A} c(w, d_n) \ln \mu_{k=z^*, w})$$

When partial derivate to zero, it's leading to the following solutions:

- $\varphi_k^{new} = \frac{N_k}{N}$ where $N_k = \sum_{n=1}^{N} z_{n,k=z^*}$ → being the cluster proportion
- $\mu_{k,w}^{new} = \dfrac{\sum_{n=1}^{N} z_{n,k=z^*} c(w, d_n)}{\sum_{w' \in A} \sum_{n=1}^{N} z_{n,k=z^*} c(w', d_n)}$

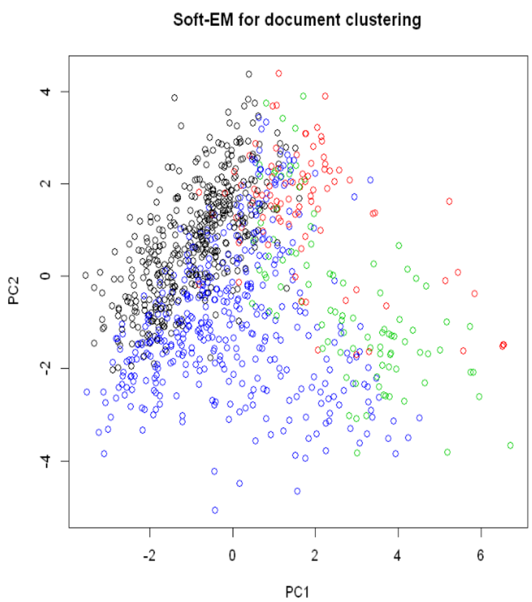
  being the word proportion for each cluster

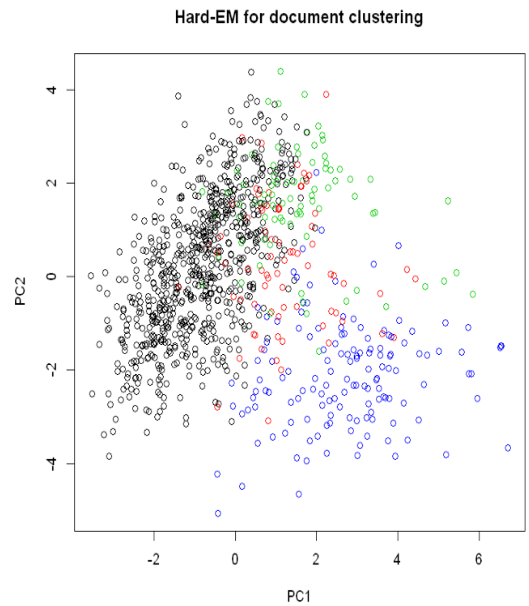
Lastly θ^{new} will be replace θ^{old} until we find optimal θ which reach converge.

Napoj Thanomkul 32338589

FIT 5201 Assignment 2 Question 1. Report

Note: The derivation of EM algorithm will be presented in a handwriting on the page before this page



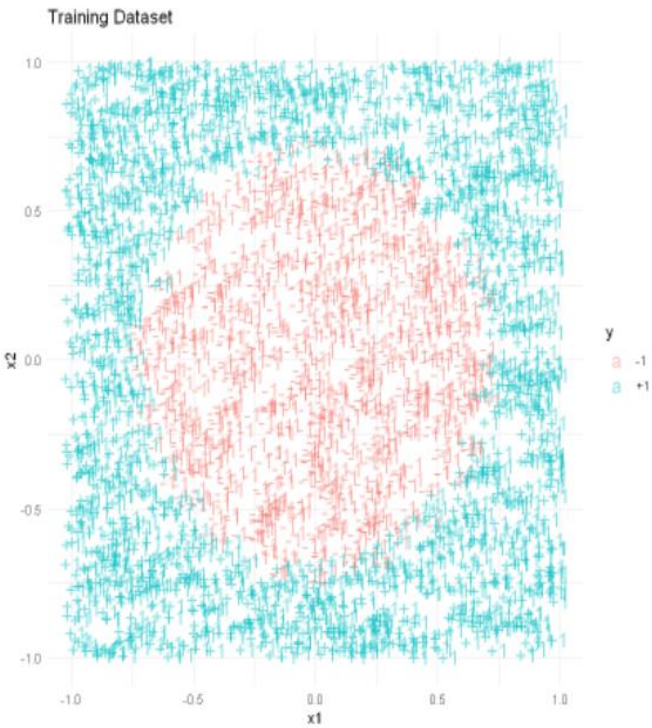


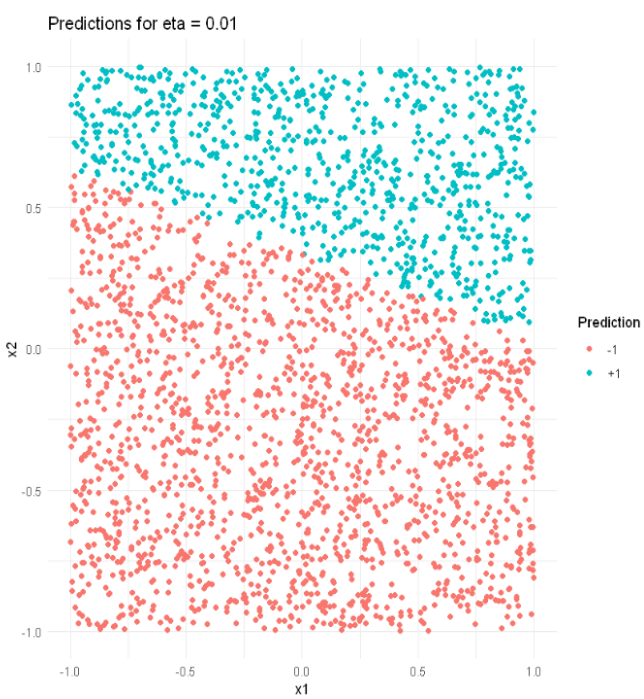
Based on the PCA plot between Hard-EM and Soft-EM for document cluster, it seems that there is some data point overlapping between each cluster in Soft-EM cluster (blue in the black area and red in between green and black area). On the other hand, there is a little overlap between each cluster. The reason for these differences might be the difference during the E-step in these two EM algorithms. The Hard-EM assign the optimal likelihood(probability) to each word clustering or belonging to the certain distribution, while Soft-EM always trade each data a result causing multiple distribution which can be changed based on boundary (can be mixture) or change of based weight of vector (Zhuang, 2022a; Zhuang,2022b).

Napoj Thanomkul 32338589

FIT 5201 Assignment 2 Question 2 report.

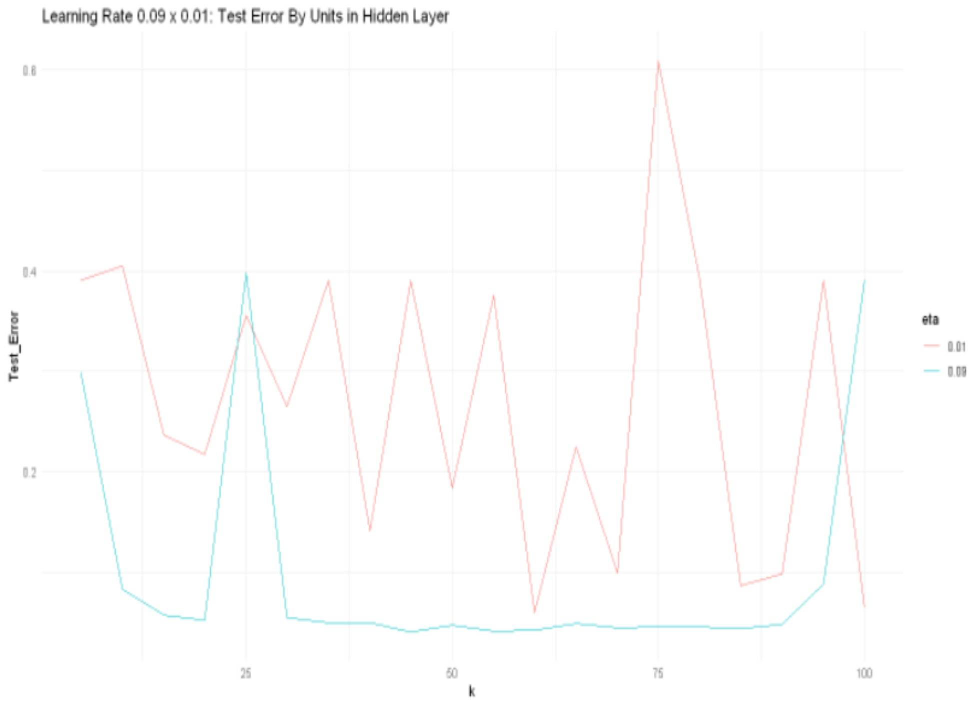
Perceptron with eta = 0.01 with error 44.84

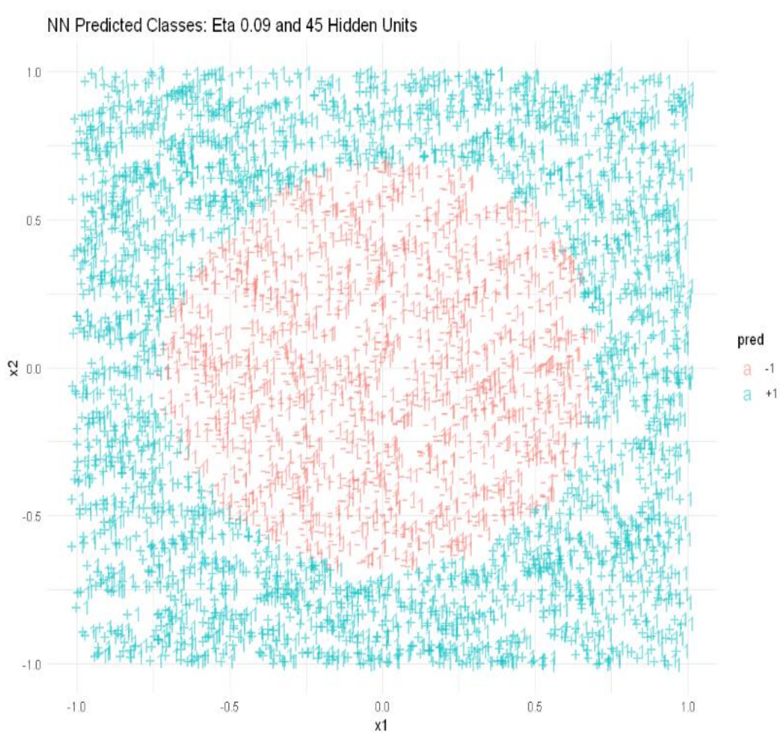




Napoj Thanomkul 32338589

Neural Network with optimal number of hidden layers (45) and eta = 0.09





Napoj Thanomkul 32338589

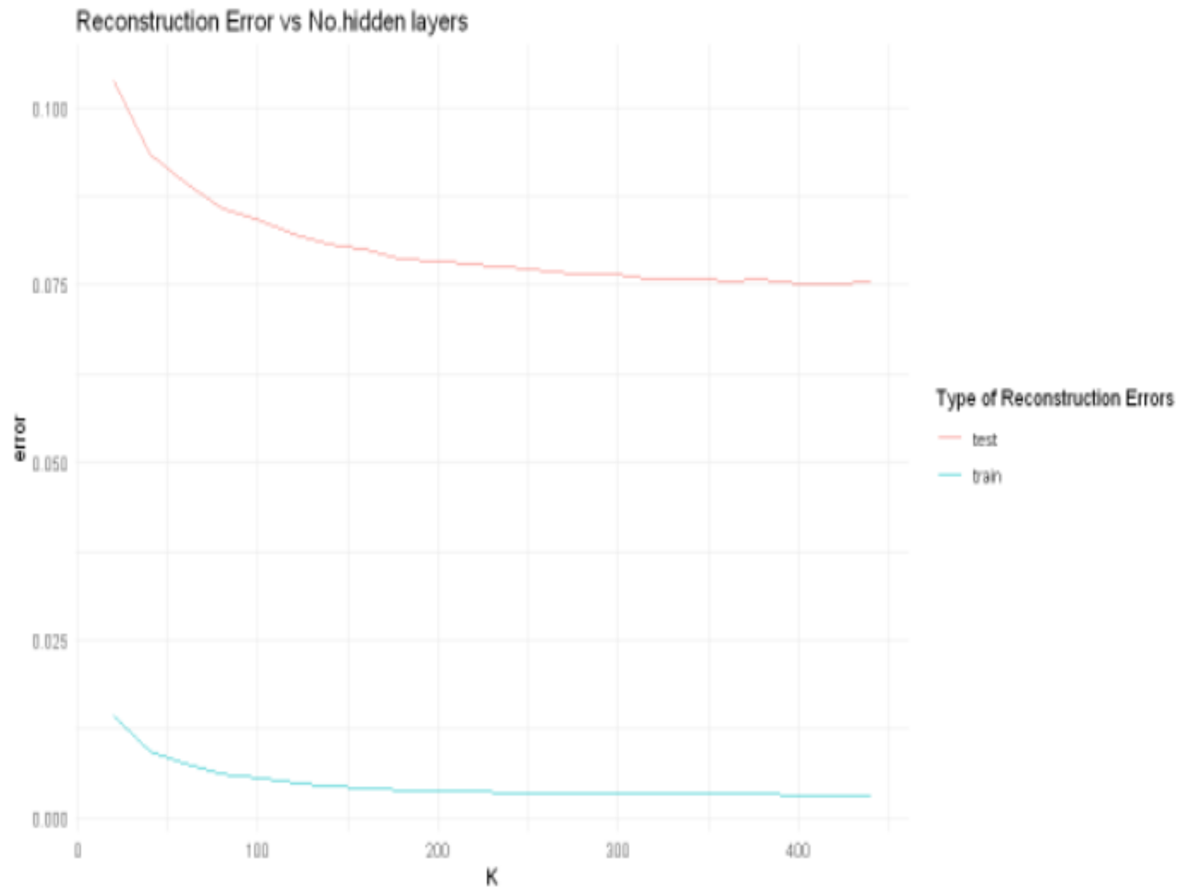
Based on the plot of Neural Network and Perceptron, there are several different between these two models as the following:

The learning rate (eta) for reaching the minimal between these two models is different. Perceptron reaches the converge point with eta = 0.01, but neural network got the model with minimal error with eta 0.09 with 45 hidden layers. The reason for the different might be the different in the processing within these two models. Neural Network uses continuous nonlinearities in the hidden units which is differentiable with respect to the network parameters, while perceptron uses step-function nonlinearities (Haffari,2017b). Due to the following reason, it makes a different result during the network training. The other reason is that perceptron is a sensitive to the initializer which mean the result of the training model will be different each time based during the training process (sometime eta 0.09 might demonstrate a better result than eta 0.09) (Haffari and Kazimipour, 2017), whereas the result of the neural network will be changed based on the eta and the hidden layers.

Regarding the difference between the two models, this is the report of the result based on the plot above: On the neural network plot, it shows some fluctuations on the graph which might be due to including the noise during the training process which may cause an overfitting. Also, the eta 0.09 tends to converge faster and less fluctuate than the eta with a rate of 0.01. On the other hand, the perceptron plot result will not stay the same for each time the model is trained. For this time, it shows that eta 0.01 is reaching the convergent point better than the perceptron with eta 0.09.

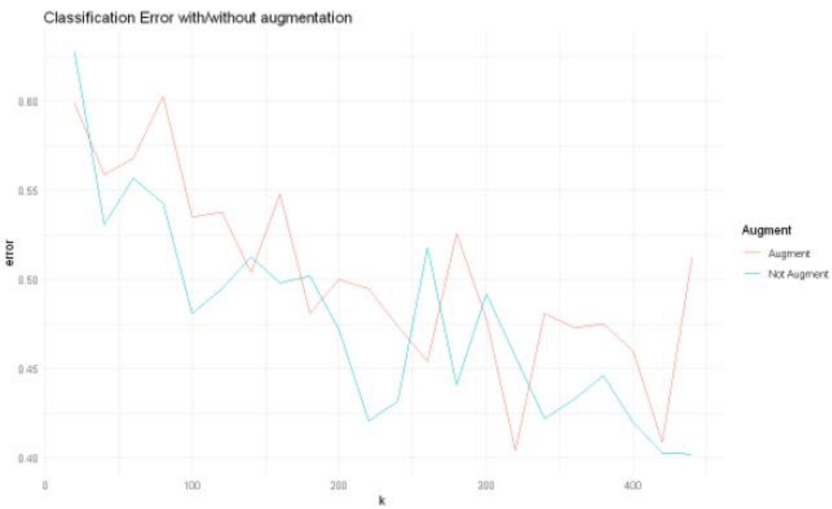
Napoj Thanomkul 32338589

FIT 5201 Assignment 3. Question 3. Report



Based on the plot, the reconstruction error for both testing and training dataset illustrated the same result pattern. The only difference is that the error of test dataset is more than the training set. However, the main point of the plot is that is include some small fluctuation on the plot when the number of hidden layers increase, which mean that it includes a noise during the model training process leading to overfitting on some certain datapoints. Also, when the number of hidden layers increase, the model will be more converge and the graph will stay stable.

Napoj Thanomkul 32338589



Based on the plot, the model with an augmentation is getting to converge faster than the model without an augmentation. This may be because the model with an augmentation is more flexible in prediction since it gets more optimize with an adding of autoencoder to the model. With the adding, it also makes the model more complex, and it shows a fluctuation when the number of hidden neurons increase which mean it is overfitting when it reaches a point where it is considered optimal or converge and beyond that point it will including the noise in prediction as well.

Napoj Thanomkul 32338589

References

Haffari, G. & Kazimipour, B. (2017, May 22nd). *Linear Models for Classification*.
https://lms.monash.edu/mod/resource/view.php?id=10133933

Haffari, G. (2017a, March 11th). *Latent Variable Models and EM*.
https://lms.monash.edu/mod/resource/view.php?id=10172218

Haffari, G. (2017b, March 11th). *Neural Networks.*
https://lms.monash.edu/mod/resource/view.php?id=10251988

Zhuang, B. (2022a). *Week 7.: Module 4 - A | Latent Variable Models* [PowerPoint slides].
https://lms.monash.edu/mod/resource/view.php?id=9895015

Zhuang, B. (2022b). *Week 8.: Module 4 - B | Latent Variable Models* [PowerPoint slides].
https://lms.monash.edu/mod/resource/view.php?id=9895032