Ali Saeizadeh 002766864
EECE5644: Prof. Brady, Machine Learning and Pattern Recognition
December 14, 2022

Homework IV

GitHub link for the codes: GitHub

Question 1.

For four classes, four three-dimensional Gaussian distribution have been considered. Each of them have different means but same covariance.

$$m_0 = \begin{bmatrix} -2 \\ -2 \\ -2 \end{bmatrix}, \quad m_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad m_2 = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}, m_3 = \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix} \tag{1}$$

$$C_0 = C_1 = C_2 = C_3 = \begin{bmatrix} 1.7 & -0.5 & 0.9 \\ 0.5 & 1.8 & -0.5 \\ 0.9 & -0.5 & 1.8 \end{bmatrix} \tag{2}$$

For keeping the probability of error between 10% to 20%, the covariance matrix can simply multiplied a number α. In our case, $\alpha = 0.4$ is good enough and gave us $p_e = 0.1675$. All these Gaussian distribution have the same chance to be selected in the data set: $P(C = 0) = P(C = 1) = P(C = 2) = P(C = 3) = 0.25$.

Train dataset with the size of 100, 200, 500, 1000, 2000, 5000 has been generated as well as a test data set with 100000 samples.

Training dataset has been separated using 10-fold cross validation, meaning that 1/10 of data will be used to find the best classifier by validating the performance of trained classifier with different number pf perceptrons in hidden layer. 10-fold cross validation has been implemented by sci-kit learn library. For this part, 1 to 512 with step of 20 perceptrons have been investigated.

Since theoretically optimal classifier is Bayes classifier, we can find the the best classifier and its performance as baseline.

$$\hat{y} = \underset{k \in \{1,\dots,K\}}{\operatorname{argmax}} p\left(C_k\right) \prod_{i=1}^{n} p\left(x_i \mid C_k\right) \tag{3}$$

By feeding the test dataset to this classifier we get $p_e \approx 0.16$ which is the best minimum error that it can be reached.

The MLP model have simple structure with just 2 layers. The first layer we used ELU activation function and for the next layer softmax activation function has been used. For implementing this part keras library has been used. Structure of MLP is shown in figure 1. Also, Adam optimizer with learning rate of 0.001 has been utilized.

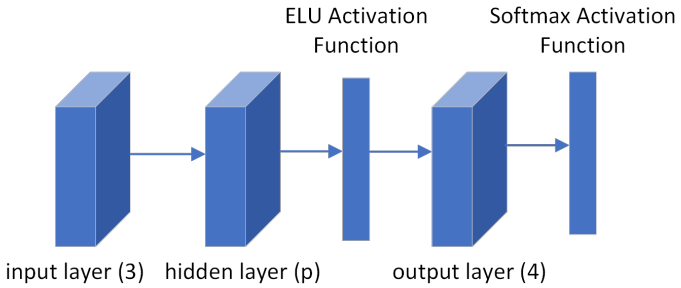

FIGURE 1. Structure of MLP

As it wanted, the optimal number of perceptrons have been found using 10-fold cross validation. These numbers for each training dataset can be found in figure 2.

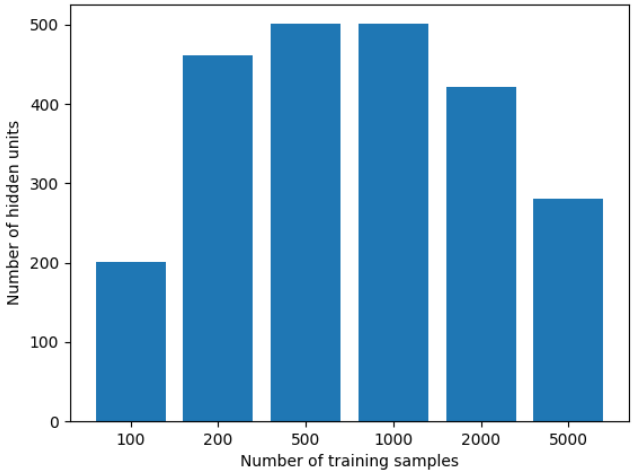

FIGURE 2. Best number of perceptrons for each training dataset.

So far, we can find optimal hyperparameters using k-fold cross validation and we will see that having enough and large training dataset plays the most important rule in classifications.

The results on test dataset can be seen in figure 3. Dash line baseline is showing the probability of error of optimal classifier. As it expected by increasing training dataset size classifiers have better performance and they get close to the optimal classifier which needs prior knowledge about the distribution. For instance, the largest dataset have same error as optimal classifier which shows the power of MLP in which we don't use prior knowledge.

The interesting result happened for the last dataset which reaches lower error than optimal classifier. That is because of our limited test data size. if we used larger dataset for test part our optimal classifier always should have better performance.

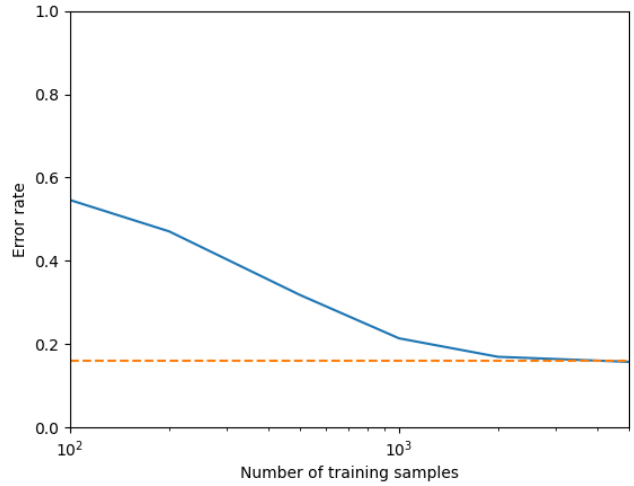

FIGURE 3. Probability of error for each training dataset.

Question 2.

Gaussian mixture model has been used by utilizing sci-kit learn library. To model true GMM we simply fit a 4-component GMM to set of random 100 samples 2-dimensional dataset which specify our true GMM for the rest of the work (without setting its parameters).

Then for each training data we used 10-fold cross validation to find the best number of component for GMM to fit the data by using EM algorithm (it is implemented in sklearn library as fit method). We have 6 candidates from 1 to 6.

Our metric for finding the best fitting GMM is average log-likelihood of the validation dataset which is computed by one of methods of sklearn library. Since it can take negative values we initialized our score at very low number. After that, we will add GMM score in each fold. Then, we pick a GMM with the highest score as our best-fitting GMM. We repeat it for 50 times and number of components is stored in each iteration. Now we can plot histogram of these data that can be found in figure 4.

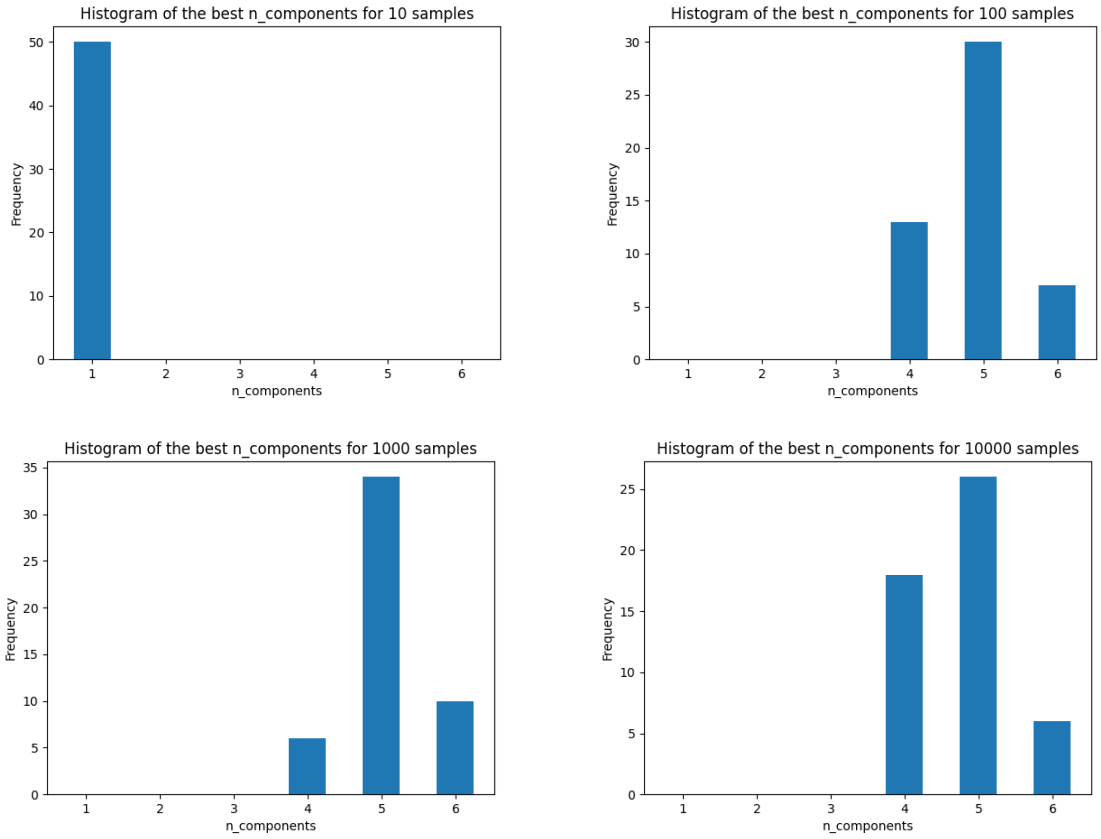

FIGURE 4. Estimated number of components histogram for each training dataset.

By increasing training size we have closer results to the actual number of components. We expect that by more increasing dataset size we eventually found 4 as the most repeated best number.