E1 213: Pattern Recognition and Neural Networks **Spring 2021**

Assignment 1: Implementing Bayes' Classifier

Author: Abijith J. Kamath *Email:* abijithj@iisc.ac.in

Introduction

Let patterns be in $\mathbb{R}^d$ and consider the 2-class classification problem – given a pattern $\mathbf{x}$, classify the pattern into a class with label y. In Bayesian classification theory, the training data $\{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^n$ is viewed to be i.i.d samples of the random vector $(\mathbf{X}, Y)$. The training samples are used to model the joint distribution of the random vector $(\mathbf{X}, Y)$.

Let $\Pr(C_i) = p_i$ be the priors associated with the ith class, $i = 0, 1$; and let $f_i(\mathbf{x}) = p_{\mathbf{X}|Y}(\mathbf{x}|y = i)$ be the class-conditional densities. Using the Bayes' rule and the training data, the update on the probabilities of the classes as:

$$q_i(\mathbf{x}) = \Pr(y = i|\mathbf{X} = \mathbf{x}) = \frac{p_{\mathbf{X}|Y}(\mathbf{x}|y = i)\Pr(C_i)}{\sum_j p_{\mathbf{X}|Y}(\mathbf{x}|y = j)\Pr(C_j)} = \frac{f_i(\mathbf{x})p_i}{\sum_j f_j(\mathbf{x})p_j},\ i = 0, 1. \tag{0.1}$$

The two-class Bayes classifier that equally penalises misclassification can now be defined as:

$$h_B(\mathbf{x}) = \begin{cases} 0, & q_0(\mathbf{x}) > q_1(\mathbf{x}), \\ 1, & \text{otherwise.} \end{cases} \tag{0.2}$$

The decision $q_0(\mathbf{x}) > q_1(\mathbf{x}) \implies p_0 f_0(\mathbf{x}) > p_1 f_1(\mathbf{x})$, and each of the prior, class-conditional pairs describes the joint distribution of $(\mathbf{X}, Y)$. Hence, this is a generative model. The set of points $\{\mathbf{x}|q_0(\mathbf{x}) = q_1(\mathbf{x})\}$ is called the decision boundary. The problem of implementing this classifier is to learn the class-conditional densities and the prior densities from the training data.

1 Bayes' Classifier in $\mathbb{R}^2$

Problem (1.1) Bayes' Classifier with Gaussian Class Conditionals

Let the class conditional distributions be modelled to be multivariate Gaussians. In the training phase, the unknown means and covariances of the Gaussians are estimated from the training data. Let the class conditional densities be parametrised as:

$$f_i(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \frac{1}{(2\pi)^{(d/2)}(\det \boldsymbol{\Sigma}_i)^{1/2}} e^{(\mathbf{x}_i - \boldsymbol{\mu}_i)^\mathsf{T} \boldsymbol{\Sigma}_i^{-10} (\mathbf{x}_i - \boldsymbol{\mu}_i)}, \ i = 0, 1. \tag{1.1}$$

The resulting Bayes classifier, using (0.2) results in a quadratic decision boundary as:

$$\frac{1}{2}\mathbf{x}^\mathsf{T} \left(\boldsymbol{\Sigma}_1^0 - \boldsymbol{\Sigma}_0^{-1}\right) \mathbf{x} - (\mathbf{W}_1 - \mathbf{W}_0)^\mathsf{T} \mathbf{x} + w_1 - w_0 = 0, \tag{1.2}$$

where $\mathbf{W}_i = \boldsymbol{\Sigma}_i^{-1}\boldsymbol{\mu}_i$ and $w_i = \frac{1}{2}\boldsymbol{\mu}_i^\mathsf{T}\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\mu}_i - \ln(p_i) + \frac{1}{2}\ln(\det \boldsymbol{\Sigma}_i)$. The quantities $\frac{1}{2}\mathbf{x}^\mathsf{T}\boldsymbol{\Sigma}_i^{-1}\mathbf{x} - \mathbf{W}_i^\mathsf{T}\mathbf{x} + w_i$ are defined to be the score functions for class i, such that the multiclass extension of the 2-class model picks the class with the least score.

To implement this classifier, the means $\boldsymbol{\mu}_i$ and the covariances $\boldsymbol{\Sigma}_i$ are to be estimated from the training data. The maximum likelihood estimators (MLE) for the mean and the covariance, given training samples $\{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^n$ are given by the sample mean and the sample covariance:

$$\begin{aligned} \boldsymbol{\mu}_i &= \frac{1}{n_i} \sum_{y^{(j)}=i} \mathbf{x}^{(j)}, \\ \boldsymbol{\Sigma}_i &= \frac{1}{n_i} \sum_{y^{(j)}=i} (\mathbf{x}^{(j)} - \boldsymbol{\mu}_i)(\mathbf{x}^{(j)} - \boldsymbol{\mu}_i)^\mathsf{T}, \end{aligned} \tag{1.3}$$

where n_i are the number of samples in class i. The priors are taken to be the ratios n_i/n.

Implementation: The classifier is trained on three datasets (P1a, P1b, P1c) with varying training sizes. The trained classifiers are tested on the full test dataset and the confusion matrix and the discriminant functions are plotted. The classifier is compared with the nearest-neighbour classifier.

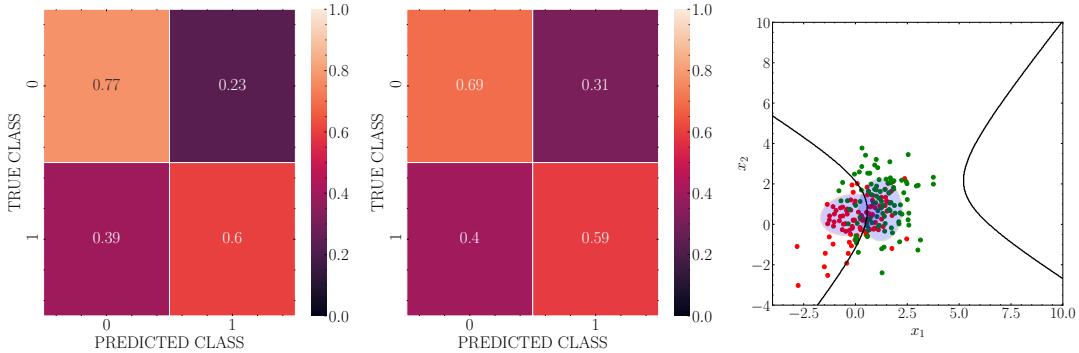
Results: Figure 1 shows the results for testing the classifier on P1a dataset. Figures 1a, 1d, 1g, 1j depicts the accuracies in the confusion matrices for the Bayes' classifier, Figures 1b, 1e, 1h, 1k depicts the accuracies in the confusion matrices of the nearest-neighbour classifer and Figures 1c, 1f, 1i, 1l shows the training samples (red samples in class 0, green samples in class 1), the $3-\sigma$ Gaussian learnt from the training samples

(ellipses of the corresponding colours), and the quadratic discriminant function (black). Figures 2 and 3 shows the corresponding figures for datasets P1b and P1c, respectively.

Inferences on P1a: The class means are $\boldsymbol{\mu}_0 = [0\ 0]^\mathsf{T}$ and $\boldsymbol{\mu}_1 = [1\ 1]^\mathsf{T}$, and the covariances are in the same order. The class distributions are "very close". i.e., they have low discriminability. This is seen with the training samples from each class overlapping into the other class. Within this setting, the classifiers trained with training sizes 10, 25, 75, and 199 on the average show, show an increasing trend in test accuracy. The nearest-neighbour classifier performs comparably to the Bayes' classifier. In no case, the error in classification using nearest-neighbour classifier exceeds twice the error in the Bayes' classifier. With increasing size of the training set, the learnt Gaussians are observed to learn "better", i.e., the means and covariances are closer to the true values. This shows that MLE is consistent.

Inferences on P1b: The class means are $\boldsymbol{\mu}_0 = [0\ 0]^\mathsf{T}$ and $\boldsymbol{\mu}_1 = [3\ 3]^\mathsf{T}$, and the covariances are in the same order. The class distributions are farther apart, i.e., the discriminability is higher than in P1a. This is seen with the training samples from each class barely overlapping into the other class. Within this setting, the classifiers trained with training sizes 10, 25, 75, and 199 on the average show, show an increasing trend in test accuracy. The nearest-neighbour classifier performs comparably to the Bayes' classifier. In no case, the error in classification using nearest-neighbour classifier exceeds twice the error in the Bayes' classifier. With increasing size of the training set, the learnt Gaussians is observed to learn "better", i.e., the means and covariances are closer to the true values.

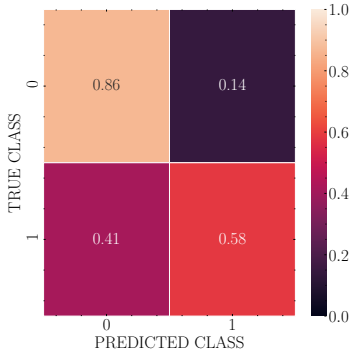
Inferences on P1c: The class means are $\boldsymbol{\mu}_0 = [0\ 0]^\mathsf{T}$ and $\boldsymbol{\mu}_1 = [3\ 6]^\mathsf{T}$, and the covariances are in the same order. The class distributions are far apart, i.e., the discriminability is high. This is seen with the training samples from each class not overlapping into the other class. Within this setting, the classifiers trained with training sizes 10, 25, 75, and 199 on the average show, show an increasing trend in test accuracy. In the case where all the training samples are used, the accuracy in classification on the test data is 99%. When the data distributions are well separated, risk minimisation is achieved with training sample size as small as 199 pairs. The nearest-neighbour classifier performs comparably to the Bayes' classifier. In most cases, the error in classification using nearest-neighbour classifier does not exceed twice the error in the Bayes' classifier. With increasing size of the training set, the learnt Gaussians is observed to learn "better", i.e., the means and covariances are closer to the true values.



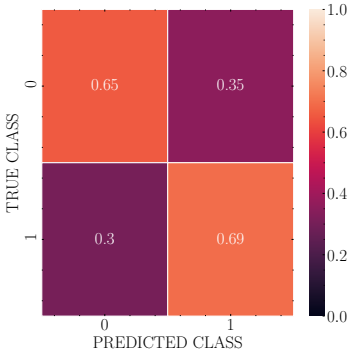
(a) QDA, size 10.

(b) KNN, size 10.

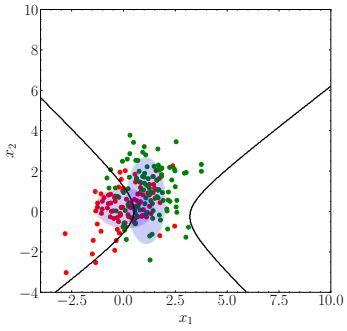
(c) Samples, size 10.



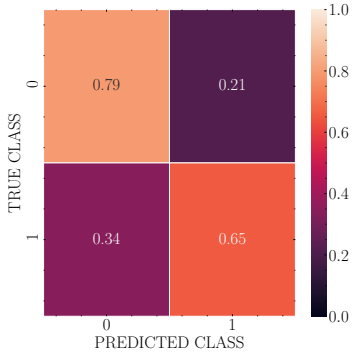
(d) QDA, size 25.



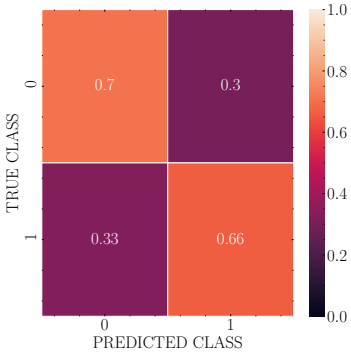
(e) KNN, size 25.



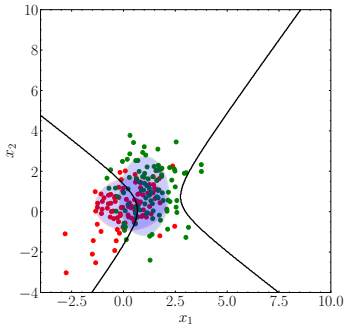
(f) Samples, size 25.



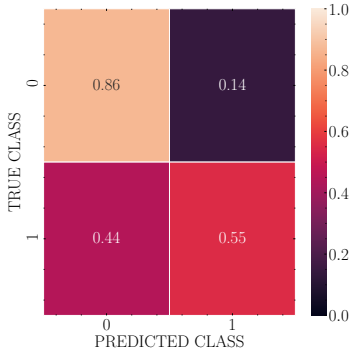
(g) QDA, size 75.



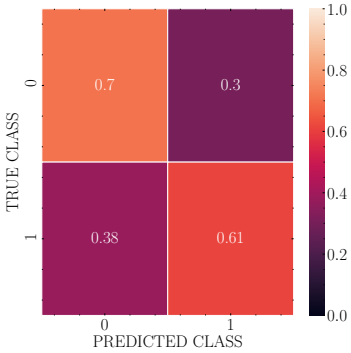
(h) KNN, size 75.



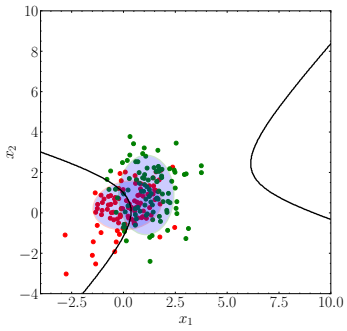
(i) Samples, size 75.



(j) QDA, size 199.

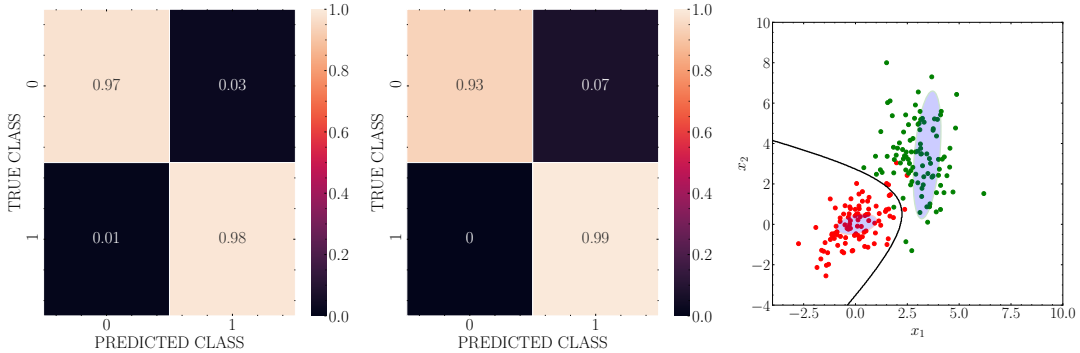


(k) KNN, size 199.

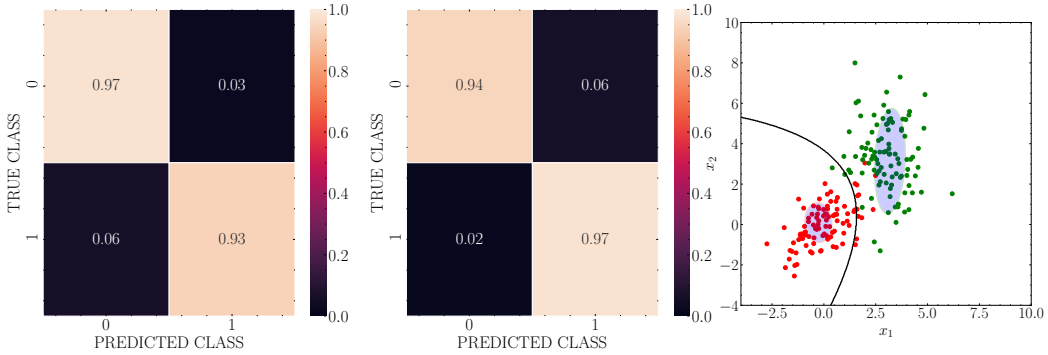


(l) Samples, size 199.

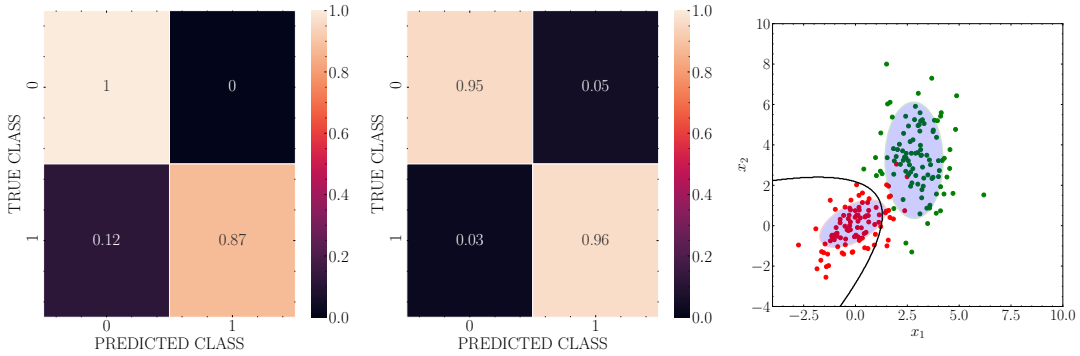
Figure 1: Bayes classifier and Nearest neighbour classification on P1a dataset.



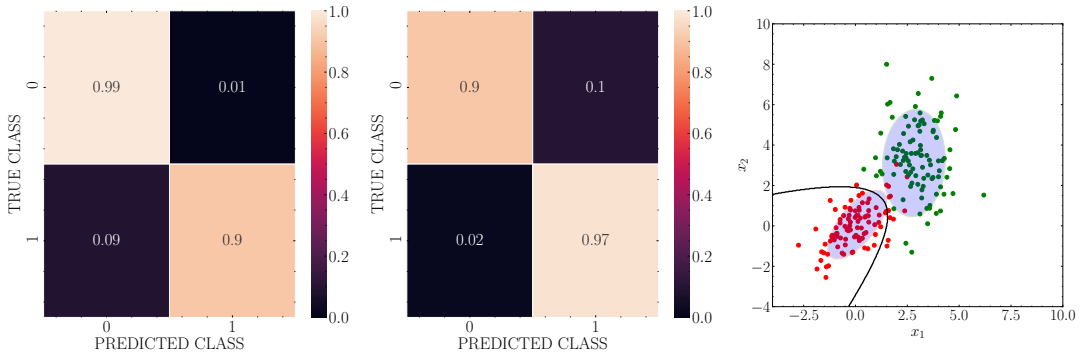
(a) QDA, size 10. (b) KNN, size 10. (c) Samples, size 10.



(d) QDA, size 25. (e) KNN, size 25. (f) Samples, size 25.

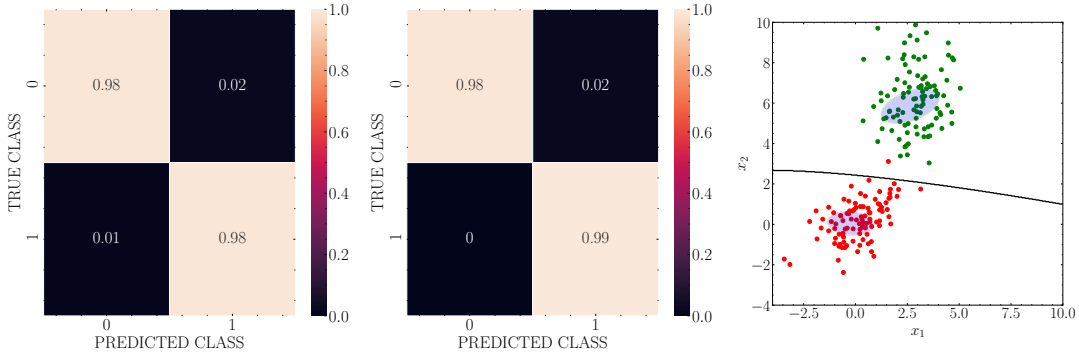


(g) QDA, size 75. (h) KNN, size 75. (i) Samples, size 75.

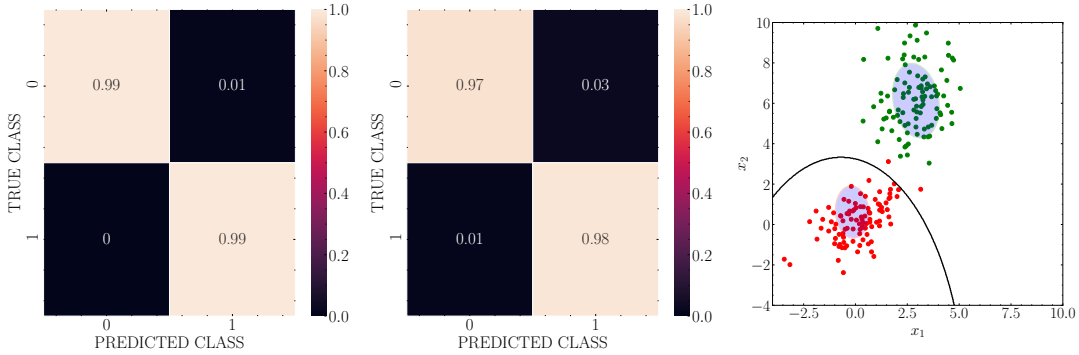


(j) QDA, size 199. (k) KNN, size 199. (l) Samples, size 199.

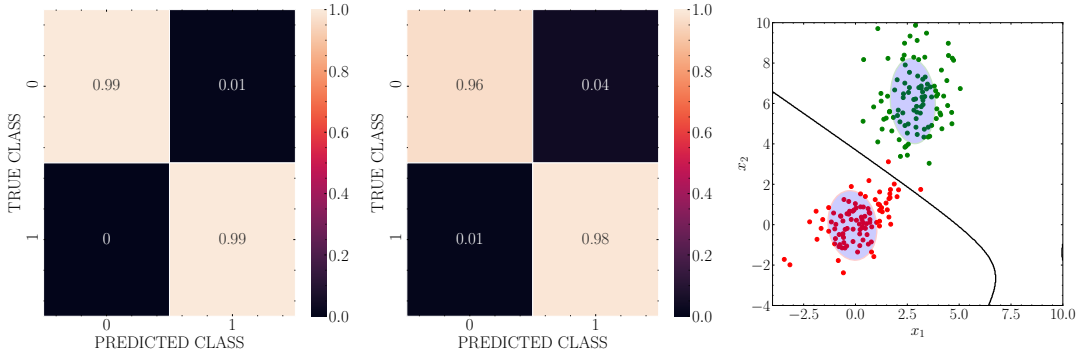
Figure 2: Bayes classifier and Nearest neighbour classification on P1b dataset.



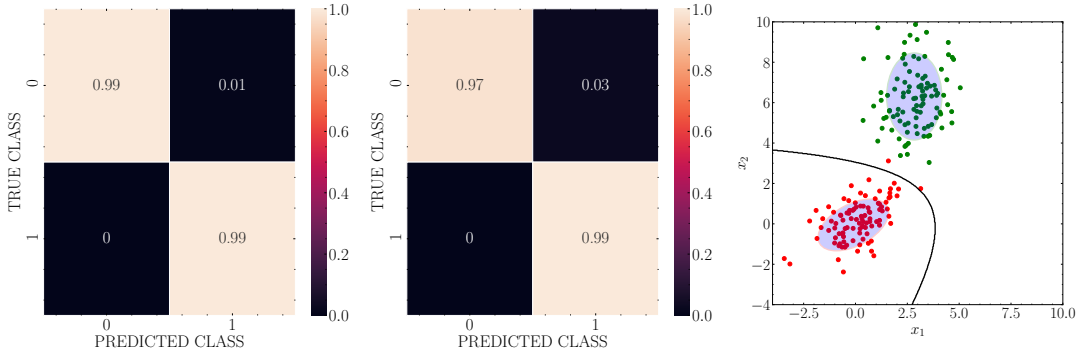
(a) QDA, size 10. (b) KNN, size 10. (c) Samples, size 10.



(d) QDA, size 25. (e) KNN, size 25. (f) Samples, size 25.



(g) QDA, size 75. (h) KNN, size 75. (i) Samples, size 75.



(j) QDA, size 199. (k) KNN, size 199. (l) Samples, size 199.

Figure 3: Bayes classifier and Nearest neighbour classification on P1c dataset.

Problem (1.2) Bayes' Classifier Trained using Gaussian Mixture Model

Let the distribution of the data be modelled to be a mixture of two Gaussians. Here, the class labels are not important to learn the Gaussian mixtures as both classes are learnt together. The data distribution has the density function:

$$f(\mathbf{x};\boldsymbol{\theta}) = \sum_{i=0}^{1} \lambda_i f_i(\mathbf{x};\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i), \tag{1.4}$$

where f_i are Gaussians as in (1.1) and $\boldsymbol{\theta} = [\lambda_0\ \lambda_1\ \boldsymbol{\mu}_0\ \boldsymbol{\mu}_1\ \boldsymbol{\Sigma}_0\ \boldsymbol{\Sigma}_1]$ is the vector of unknowns to be estimated during the training phase using the training samples.

Implementation: The maximum likelihood estimation (MLE) is performed using the expectation maximisation (EM) algorithm. The E-step updates the weight matrix:

$$\gamma_{i,j}^{(k+1)} = \frac{\lambda_j^{(k+1)} f_i(\mathbf{x}^{(i)};\boldsymbol{\mu}_j^{(k+1)},\boldsymbol{\Sigma}_j^{(k+1)})}{\sum_{m=0}^{1} \lambda_m^{(k+1)} f_i(\mathbf{x}^{(i)};\boldsymbol{\mu}_m^{(k+1)},\boldsymbol{\Sigma}_m^{(k+1)})}, \tag{1.5}$$

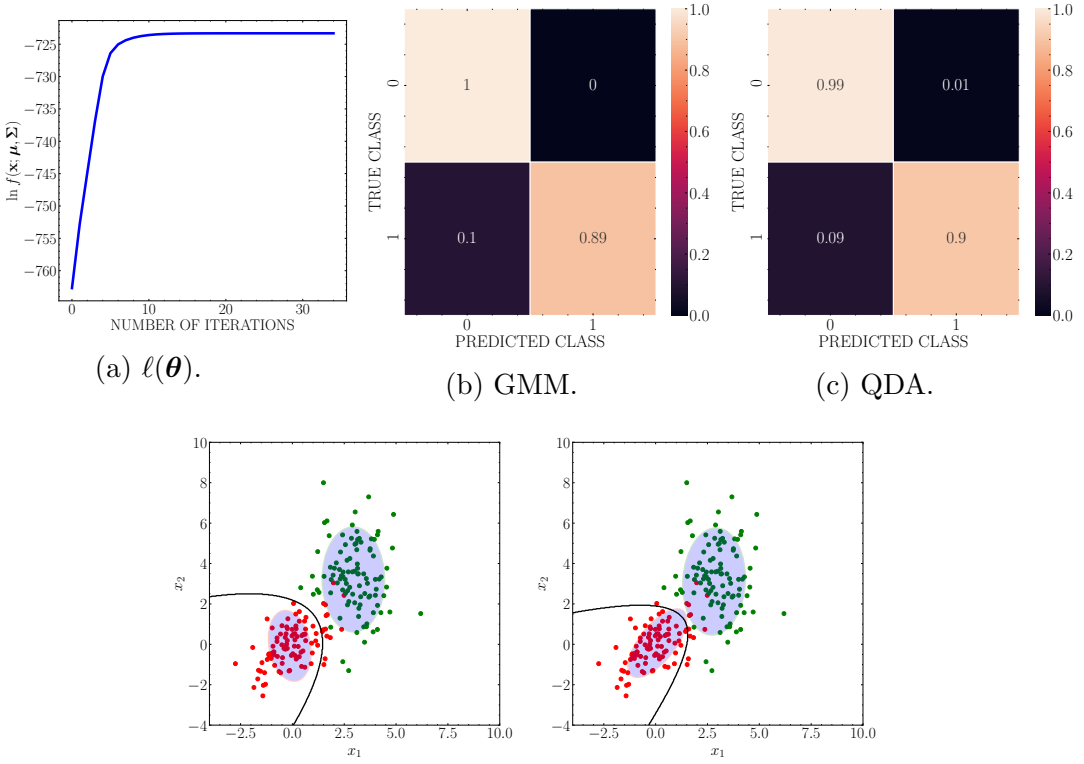
where the $(k+1)$st update for the priors, means and covariances are obtained in the M-step with:

$$\begin{aligned}
\boldsymbol{\mu}_j^{(k+1)} &= \frac{\sum_{i=1}^{n} \gamma_{i,j}^{(k)} \mathbf{x}^{(i)}}{\sum_{i=1}^{n} \gamma_{i,j}^{(k)}}, \\
\lambda_j^{(k+1)} &= \frac{1}{n} \sum_{i=1}^{n} \gamma_{i,j}^{(k)}, \\
\boldsymbol{\Sigma}_j^{(k+1)} &= \frac{\sum_{i=1}^{n} \gamma_{i,j}^{(k)} (\mathbf{x}^{(i)} - \boldsymbol{\mu}_j^{(k)})(\mathbf{x}^{(i)} - \boldsymbol{\mu}_j^{(k)})^{\mathsf{T}}}{\sum_{i=1}^{n} \gamma_{i,j}^{(k)}}.
\end{aligned} \tag{1.6}$$

Each component in Gaussian mixture is considered to be the class conditional and the corresponding Bayes' classifier in (0.2) is tested on the test dataset. The classifier is trained on the P1b dataset using all training samples. The trained classifier is tested in the full test dataset and the confusion matrix and the discriminant functions are plotted. The classifier is compared with the class conditional modelled as Gaussians as in Problem 1.1.

Results: Figure 4 shows the results for testing the classifier on P1b dataset. Figure

4a shows the log-likelihood function $\ell(\boldsymbol{\theta}) = \ln f(\mathbf{x}; \boldsymbol{\theta})$ for the parameter vector $\boldsymbol{\theta}$. Figures 4b and 4c shows the accuracies in the confusion matrix for the classifier trained with GMM and individual Gaussian classes, respectively. Figures 4d and 4e show the training samples (red samples in class 0, green samples in class 1), the $3-\sigma$ Gaussian learnt from the training samples (ellipses of the corresponding colours), and the quadratic discriminant function (black).



(a) $\ell(\boldsymbol{\theta})$. (b) GMM. (c) QDA.

(d) Samples, GMM. (e) Samples, QDA.

Figure 4: Bayes classifier learnt using Gaussian mixture model and each class modelled as Gaussian on P1b dataset.

Inferences: Figure 4a shows the log-likelihood function for $\boldsymbol{\theta}$ and it can be observed that the EM algorithm maximises the log-likelihood and convergence is achieved in 34 iterations. Since the entire data distribution is learnt together using GMM, the learning is unsupervised; whereas modelling each class as a Gaussian requires the class labels and is hence supervised. However, the knowledge of the number of components is necessary for training using GMM. The accuracies on the test data using GMM are comparable to the accuracies on the test data using supervised training, and the discriminant functions obtained are also similar.

Problem (1.3) Bayes' Classifier with Exponential Class Conditional

Let the class 0 be modelled as a Gaussian, identical to Problem 1.1 and class 1 be modelled as independent Exponentials, i.e.:

$$f_1(\mathbf{x}; \lambda_1, \lambda_2) = \lambda_1 \lambda_2 e^{-\lambda_1 x_1 - \lambda_2 x_2}, \tag{1.7}$$

where the parameters λ_1, λ_2 are unknowns that are estimated in the training phase using the training data.

Implementation: The maximum-likelihood estimate for the parameters is given by:

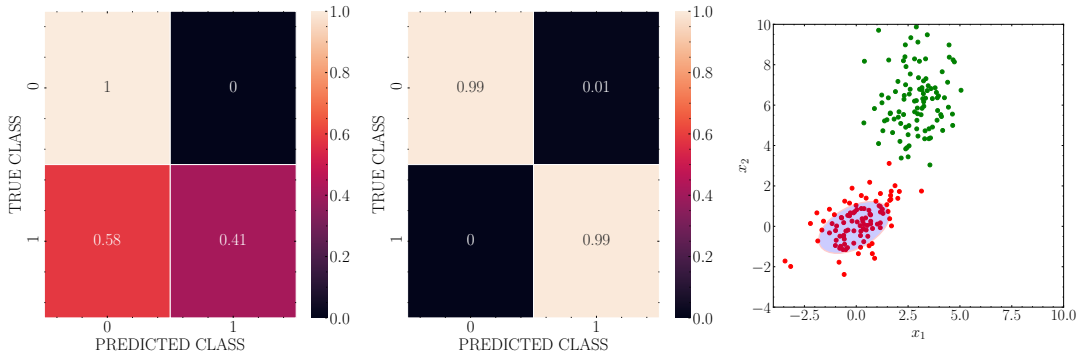
$$\lambda_j = \frac{n}{\sum_{i=1}^{n} x_j^{(i)}}, \; j = 1, 2, \tag{1.8}$$

where n is the number of samples in the class. The priors are computed similar to 1.1, as ratios of number of samples in the class to total number of samples. The classifier is trained on the P1c dataset using all the training samples. The trained classifier is tested on the full dataset and the confusion matrix are plotted. The classifier is compared with both class conditionals modelled as Gaussians as in 1.1.

Results: Figure 5 shows the results for testing the classifier on P1c dataset. Figure 5a shows the accuracies in the confusion matrix for the classifier with Gaussian and Exponential class distributions. Figure 5b shows the accuracies in the confusion matrix for the classifier with both classes modelled as Gaussians. Figure 5c shows the training samples along with the learnt Gaussian for class 0.

Inferences: The performance of the classifier with exponential distribution is poor. The accuracy of classification into class 0 with Gaussian class conditional is 100% in comparison to the accuracy of classification with both class conditionals as Gaussians and with GMM. However, the accuracy of classification with the exponential distribution is very poor and comparable to a coin-toss classifier. This is expected as the mean of class 1 is at $[3\ 6]^\mathsf{T}$ and since the class is exponential, the samples are not drawn from a bell-shape. The samples are drawn around $[0\ 0]^\mathsf{T}$ with high probability.

The exponential distribution is not a good model for the class conditional, in this



(a) Exp-Gaussian. (b) QDA. (c) Samples.

Figure 5: Bayes' classifier with class conditionals as Gaussian and Exponential; and both class conditionals as Gaussians on P1c dataset.

case. This is apparent as we know a priori, the class conditionals are Gaussian. Model mismatch in the modelling of the class conditional explains the poor performance of the classifiers.

2 Bayes' Classifier in $\mathbb{R}^{20}$

Problem (2) Bayes' Classifier with Gaussian Class Conditionals

Let the class conditional distributions be modelled to be multivariate Gaussians. In the training phase, the unknown means and covariances of the Gaussians are estimated from the training data.

Implementation: The model is identical to the setting in Problem 1.1 and the updates for the unknown means and covariances are identical to (1.3). The classifiers are trained on three datasets (P2a, P2b, P2c) with varying training sizes. The trained classifiers are tested on the full test dataset and the accuracies in confusion matrices are plotted. The classifier is compared with the nearest-neighbour classifier.

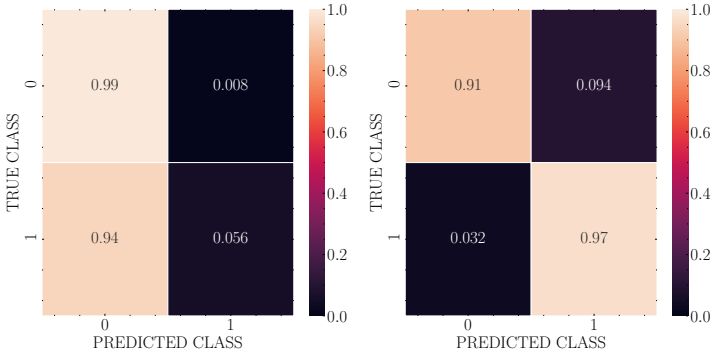
Results: Figure 6 shows the results for testing the classifier on P2a dataset. Figure 6a 6c 6e 6g depicts the accuracies in confusion matrices for the Bayes' classifier, Figures 6b 6d 6f 6h depicts the accuracies in confusion matrices for the nearest-neighbour classifier. Figures 7 and 8 shows the corresponding results for datasets P2b and P2c, respectively.

All the datasets have the same means for the classes, $\boldsymbol{\mu}_0 = \mathbf{0}$ and $\boldsymbol{\mu}_1 = \mathbb{1}$, but different covariances.

Inferences on P2a: The covariance matrices of the classes are both equal to the identity matrix. Hence, the discriminability in the data is sufficient and the classifier performs well with sufficient training samples. 10 samples are insufficient to completely capture the model and the Bayes' classifier performs poorly. The estimated means are not accurate and hence the class conditionals are not accurate. The nearest-neighbour classifier performs well with 10 samples. With few training samples, the nearest-neighbour classifier is more reliable than the Bayes' classifier. As the number of training samples increase from 10 to 50, 100 and 300, the performance of the Bayes' classifier beats nearest-neighbour, as in these cases, the MLE converges to the true values. The error in nearest-neighbour classification is at most twice the error in Bayes' classifier, in most cases.

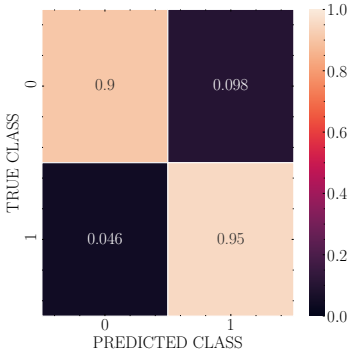
Inferences on P2b: The covariance matrices of the classes are both equal to three times the identity matrix. Hence, the discriminability in the data is insufficient. The data from either of the classes overlap into the other class. The classifiers perform poorly in relation to P2a, where the discriminability is better. The number of samples taken in the range of 50-100 are not sufficient of risk minimisation for the data with overlapping classes. In this range of training samples, the nearest neighbour classifier performs better than the Bayes' classifier. As the number of training samples increases to the full training data size, the estimation accuracy improves and the accuracy of the classifier improves. Again, when the estimation of the parameters improves, the error in the nearest-neighbour classifier is less than twice the error in Bayes' classifier.

Inferences on P2c: The covariance matrices of the classes are both random matrices with ones on the diagonal. The data in the classes can have different orientations, in which case the discriminability in the data is large. The classifiers perform well suggesting that classes indeed have different orientations. In this case, the classifier performs well with 99% accuracy with training size as low as 100 samples. The nearest-neighbour classifier does perform better in the case with 50 training samples. With larger training size, the error in classification using nearest-neighbour classifier is around twice the error in Bayes' classifier.

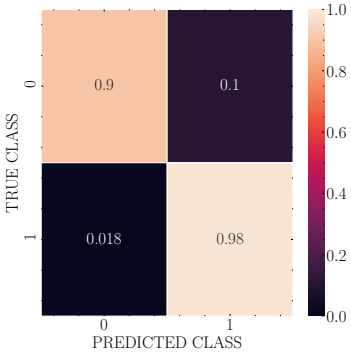


(a) QDA, size 50.

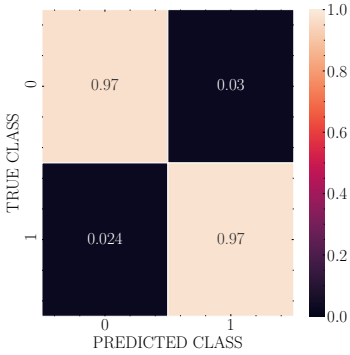
(b) KNN, size 50.



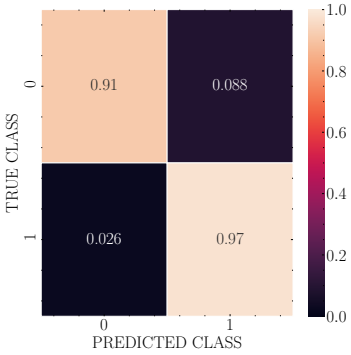
(c) QDA, size 100.



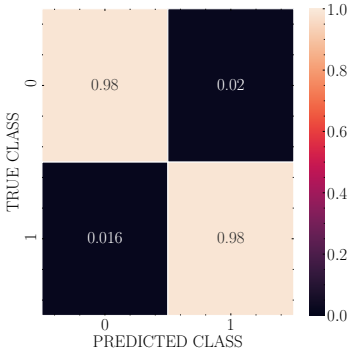
(d) KNN, size 100.



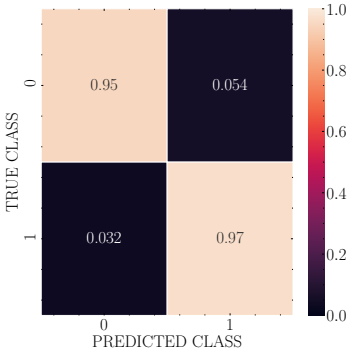
(e) QDA, size 300.



(f) KNN, size 300.

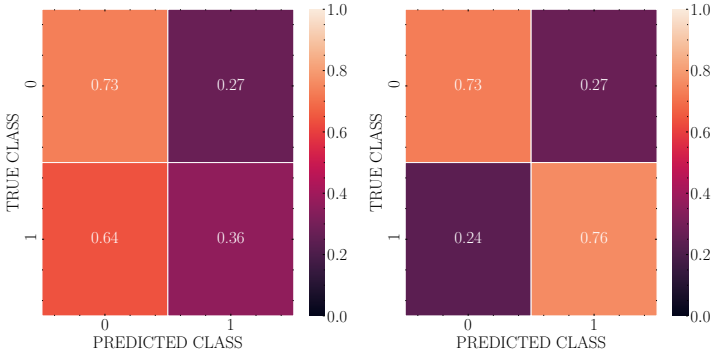


(g) QDA, size 999.



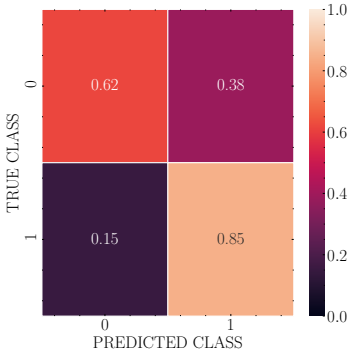
(h) KNN, size 999.

Figure 6: Bayes classifier and Nearest neighbour classification on P2a dataset.

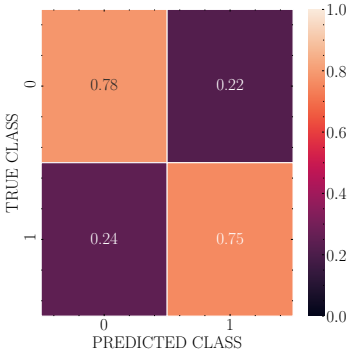


(a) QDA, size 50.

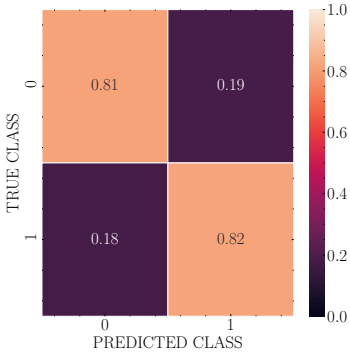
(b) KNN, size 50.



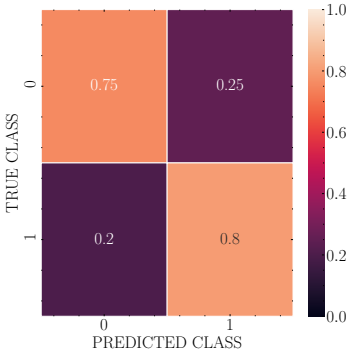
(c) QDA, size 100.



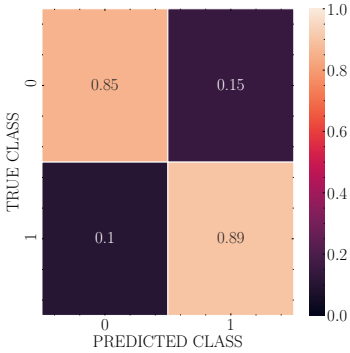
(d) KNN, size 100.



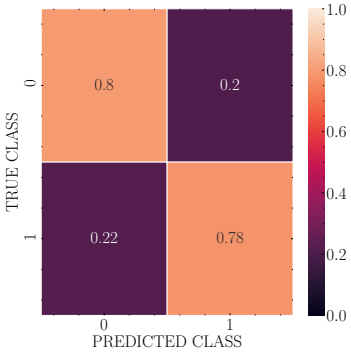
(e) QDA, size 300.



(f) KNN, size 300.

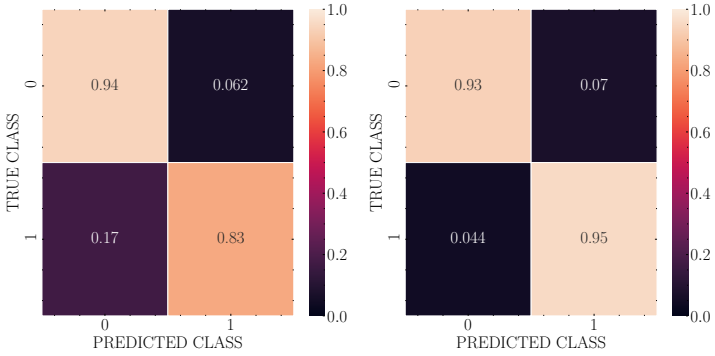


(g) QDA, size 999.



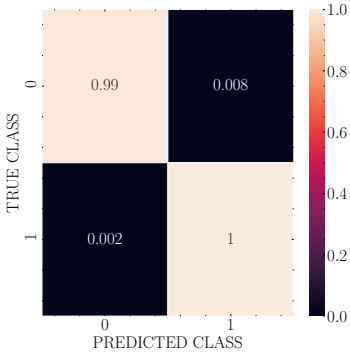
(h) KNN, size 999.

Figure 7: Bayes classifier and Nearest neighbour classification on P2b dataset.

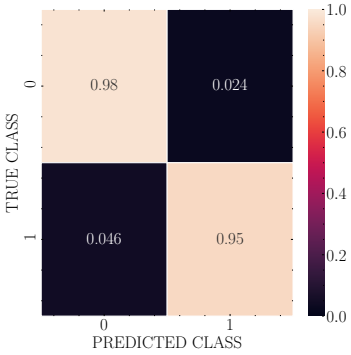


(a) QDA, size 50.

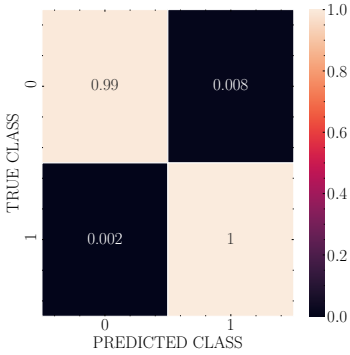
(b) KNN, size 50.



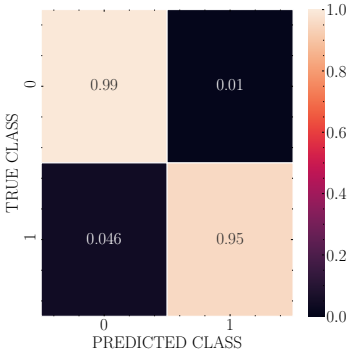
(c) QDA, size 100.



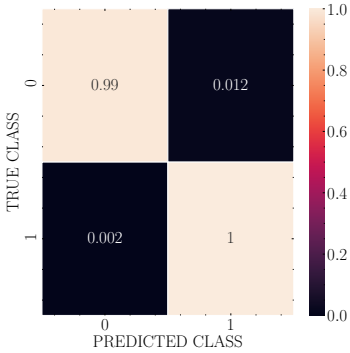
(d) KNN, size 100.



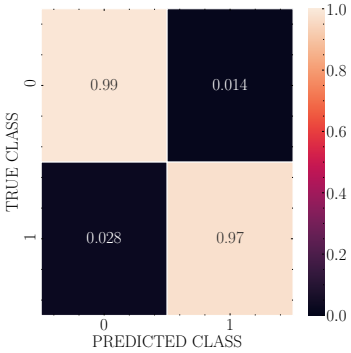
(e) QDA, size 300.



(f) KNN, size 300.



(g) QDA, size 999.



(h) KNN, size 999.

Figure 8: Bayes classifier and Nearest neighbour classification on P2c dataset.

3 Gaussian Mixture Model in $\mathbb{R}$

Problem (3) Bayes' Classifier Trained using Gaussian Mixture Model

Let the class conditionals be modelled as a mixture of Gaussians identical to (1.4). Here, each class is modelled as a mixture of Gaussians, i.e., the learning is supervised. In the training phase, the unknown weights, means and covariances are estimated from the training samples using the EM algorithm, identical to (1.5) and (1.6).

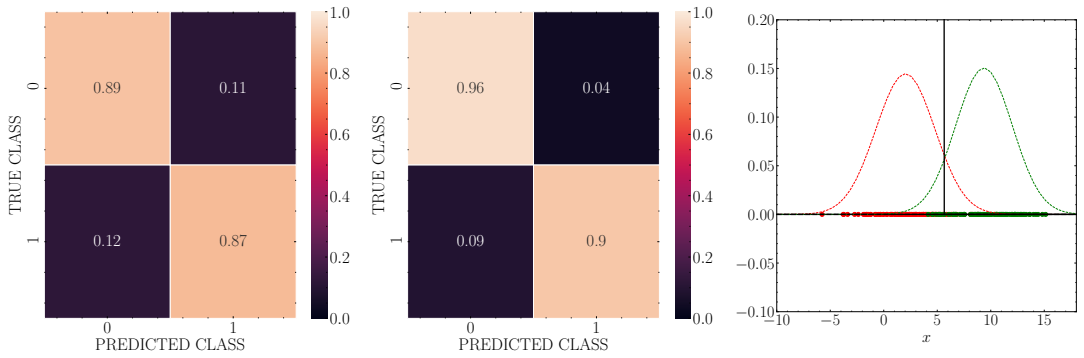
Implementation: The classifiers are trained on two datasets (P3a, P3b) with the full set of training samples. The trained classifiers are tested on the full test dataset and the accuracies in the confusion matrices are plotted. The classifier is compared with the Bayes' classifier with Gaussian class conditional model and the nearest-neighbour classifier.

Results: Figure 9 shows the results for testing the classifiers on P3a dataset. Figure 9a shows the accuracies in the confusion matrix for the nearest-neighbour classifier. Figure 9b and 9c shows the accuracies in the confusion matrix for the Bayes' classifier with Gaussian class conditionals and the learnt class conditionals, the samples and the discriminant function, respectively. Figure 9d and 9e shows the accuracies in the confusion matrix for the Bayes' classifier with Gaussian mixture class conditionals and the learnt class conditionals and the samples, respectively. Figure 10 shows the corresponding results on P3b dataset.

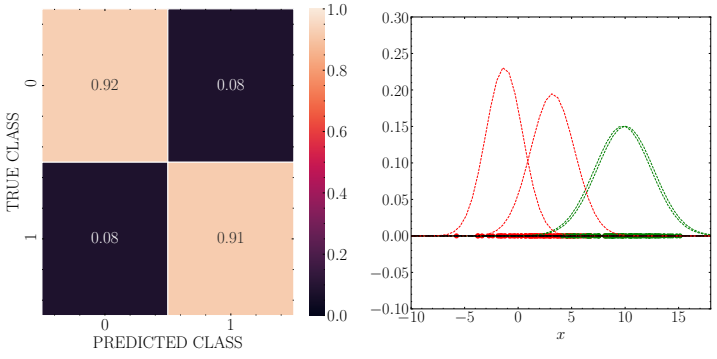
Inferences on P3a: The true distributions are indeed a mixture of two Gaussians with means $\{0, 4\}$ and $\{8, 12\}$. The variances are of the same order. The discriminability in the data is large. The Bayes' classifier in both cases, with Gaussian class conditionals and Gaussian mixture class conditionals performs identically. The Bayes' classifier with Gaussian class conditionals in $\mathbb{R}$ is a threshold based classifier. The priors are approximately equal to half and the variances are the same. Hence, the threshold passes through the point of intersection of the PDFs of the class conditionals as in Figure 9c. In the case of Bayes' classifier with Gaussian mixture class conditionals, it can be observed that the mixture of Gaussians in class 1 (green samples in Figure 9c) collapses into a single Gaussian, in a drawback dubbed *mode collapse*. However, since the classifier in 1D is only a threshold based classifier, the performance of the Bayes' classifiers are similar. The

nearest-neighbour classifier performs well where the errors are not larger than twice the error in the Bayes' classifier.

Inferences on P3b: The true distributions are indeed a mixture of two Gaussians with means $\{0, 8\}$ and $\{4, 12\}$. The variances are of the same order. The Bayes' classifier with Gaussian class conditionals has largely overlapping Gaussians with variances $\{18, 20\}$ which is 4 times the true variance. The threshold is a bisector that is weighted by the priors. In the Bayes' classifier with Gaussian mixtures as class conditionals, since the mean of one component of the Gaussian mixture in one class is between the mean of the components in the other class, the mixture component model has low discriminability. The overall performance of any of the classifiers is poor. The Bayes' classifier with Gaussian mixture class conditionals performs slightly better than the Bayes' classifier with Gaussian class conditionals as it represents the true data. Mode collapse in class 1 is still observed. The nearest neighbour classifier performs as expected with bounded errors.

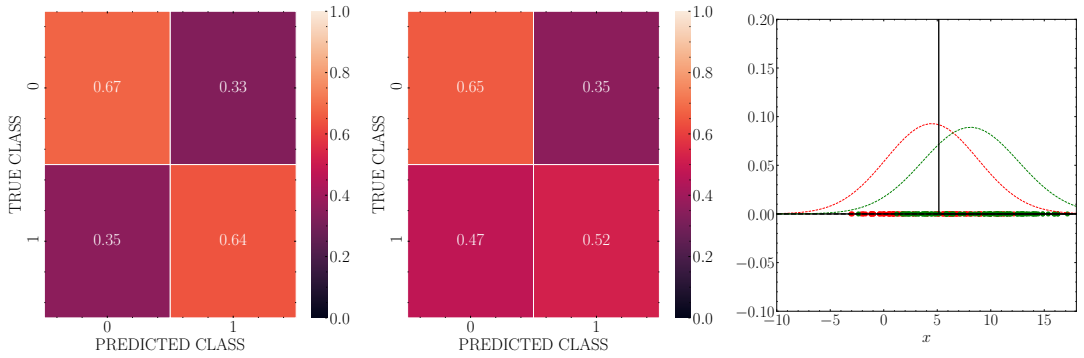


(a) KNN. (b) QDA. (c) Samples, QDA.

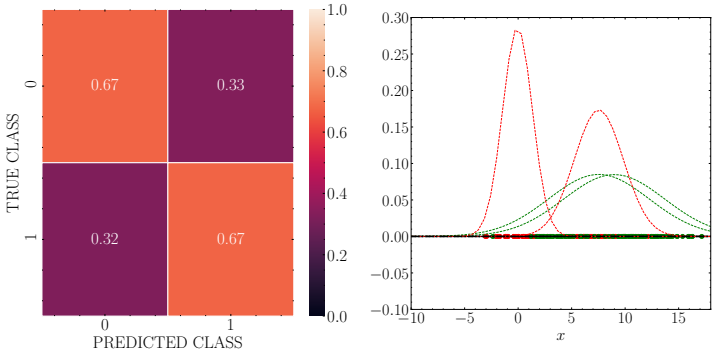


(d) GMM. (e) Samples, GMM.

Figure 9: Bayes classifier with Gaussians, GMM and nearest neighbour on P3a dataset.



(a) KNN. (b) QDA. (c) Samples. QDA.



(d) GMM. (e) Samples, GMM.

Figure 10: Bayes classifier with Gaussians, GMM and nearest neighbour on P3b dataset.

4 Naive Bayes' Classifier for Document Classification

The bag of words (BoG) model considers words to be binary features $\mathbf{x} \in \{0,1\}^d$ with $x_i = 1$ if the ith word is present, and $x_i = 0$ otherwise. Alternatively, the TF-IDF models are dubbed to be better features to represent documents. Let g_{ij} denote the number of times word i appears in document j. The term frequency is defined as $\text{TF}_{ij} = \frac{g_{ij}}{\max g_{ij}}$. The inverse document frequency of word i is defined as $\text{IDF}_i = \log_2\left(\frac{N}{n_i}\right)$, where N is the total number of documents and n_i is the number of times the word i appears. The TF-IDF of document j is the vector whose ith entry is given by $\text{TF}_{ij}\text{IDF}_i$.

Implementation: The naive Bayes' classifier assumes the joint class conditional to

be independent in x_i i.e.,

$$f(\mathbf{x}; y = c, \boldsymbol{\theta}) = \prod_{j=1}^{d} f(x_j; y = c, \theta_{jc}), \tag{4.1}$$

where, which is either the BoG features or TF-IDF, and $f(x_j; y = c, \theta_{jc})$ is a Bernoulli random variable with parameter θ_{jc}. The parameter can be estimated using maximum likelihood estimation:

$$\theta_{jc} = \frac{1}{n} \sum_{y^{(i)}=c} x_i^{(j)}, \; c = 0, 1, \tag{4.2}$$

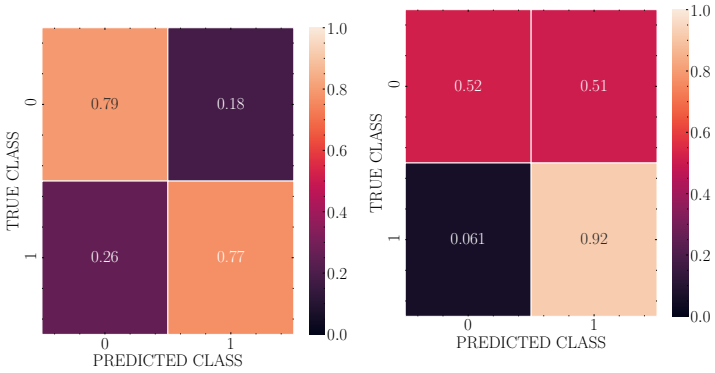
where n is the number of samples in class c. The `sentiment analysis` dataset is used for training and testing. The training-testing split is taken to be 30%, 50% and 80%, and the accuracies in the confusion matrices are plotted. The classifiers are trained and tested using the `scikit learn` toolbox in Python.

Results: Figure 11 shows the accuracies in the confusion matrices with training a naive Bayes' classifier using the BoG model (Figure 12a) and TF-IDF (Figure 12b) with 50% train-test split; and Figure 12 and 13 shows the corresponding results with 50% and 80% train-test splits.

Inferences: In the case with training using as few as 30% training samples, it can be seen that the TF-IDF features does not perform as well as the BoG model. With higher training samples 50% and 80%, the TF-IDF model performs better than the BoG model. In both cases, with BoG model and TF-IDF features, it can be seen that the accuracy improves with a larger training dataset. With a fixed training size, it can be seen that the accuracy improves with using TF-IDF rather than using the BoG model.

5 Code Repository

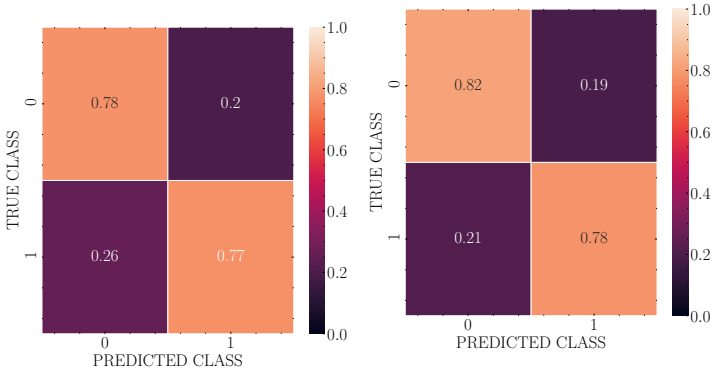
The Python codes to reproduce the results can be found in the GitHub repository `https://github.com/kamath-abhijith/Bayes_Classifier`. Use `requirements.txt` to install the dependencies.



(a) BoG model with 30% training data.

(b) TF-IDF features with 30% training data.

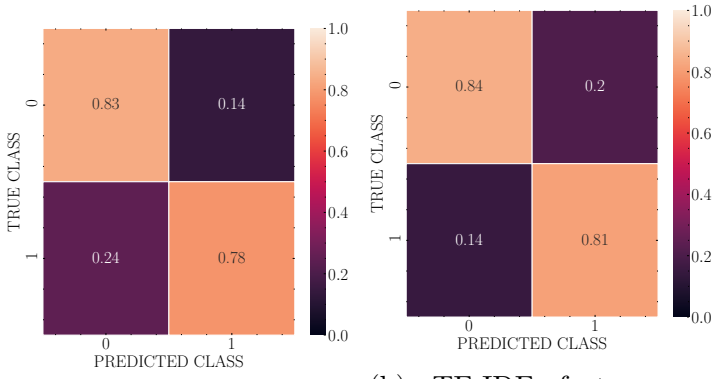
Figure 11: Document classification using naive Bayes' classifiers with BoG model and TF-IDF features, 30% train-test split.



(a) BoG model with 50% training data.

(b) TF-IDF features with 50% training data.

Figure 12: Document classification using naive Bayes' classifiers with BoG model and TF-IDF features, 50% train-test split.



(a) BoG model with 80% training data.

(b) TF-IDF features with 80% training data.

Figure 13: Document classification using naive Bayes' classifiers with BoG model and TF-IDF features, 80% train-test split.