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:

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

where $\mathbf{W}_i = \mathbf{\Sigma}_i^{-1}\boldsymbol{\mu}_i$ and $w_i = \frac{1}{2}\boldsymbol{\mu}_i^\top\mathbf{\Sigma}_i^{-1}\boldsymbol{\mu}_i - \ln(p_i) + \frac{1}{2}\ln(\det\mathbf{\Sigma}_i)$. The quantities $\frac{1}{2}\mathbf{x}^\top\mathbf{\Sigma}_i^{-1}\mathbf{x} - \mathbf{W}_i^\top\mathbf{x} + w_i$ 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 $\mathbf{\Sigma}_i$ 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:

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

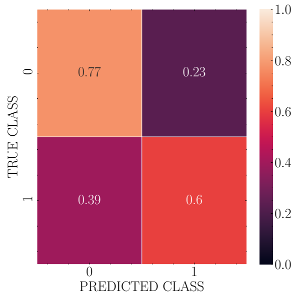
where n_i are the number of samples in class i. The priors are taken to be the ratios n_i/n.

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.

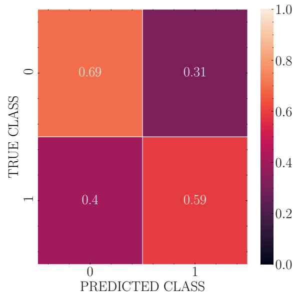
Figure 1 shows the results for testing the classifier on P1a dataset. Figures 1a, 1d, 1g, 1j depicts the accuracies in the confusion matrices for the Bayes' classifier, Figures 1b, 1e, 1h, 1k depicts the accuracies in the confusion matrices of the nearest-neighbour classifer and Figures 1c, 1f, 1i, 1l 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 $\boldsymbol{\mu}_0 = [0\ 0]^\top$ and $\boldsymbol{\mu}_1 = [1\ 1]^\top$, 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.

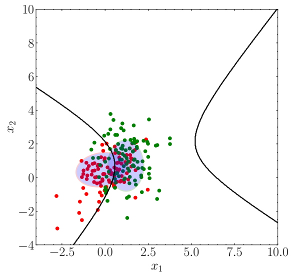
Inferences on P1b: The class means are $\boldsymbol{\mu}_0 = [0\ 0]^\top$ and $\boldsymbol{\mu}_1 = [3\ 3]^\top$, and the covariances are in the same order. The class distributions are farther apart, i.e., the discriminability is higher than in P1a. This is seen with the training samples from each class 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.



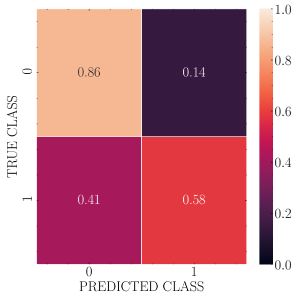
(a) QDA on P1a, size 10.



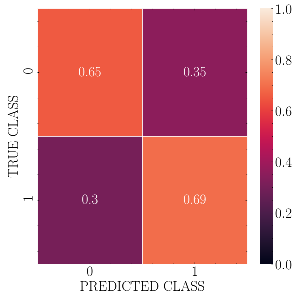
(b) KNN on P1a, size 10.



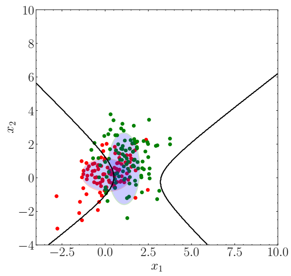
(c) Discriminant, size 10.



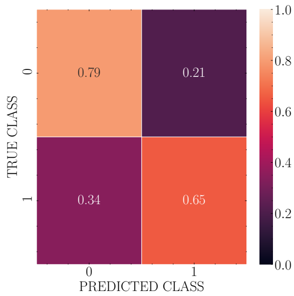
(d) QDA on P1a, size 25.



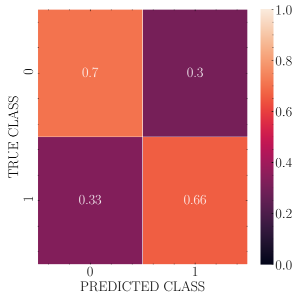
(e) KNN on P1a, size 25.



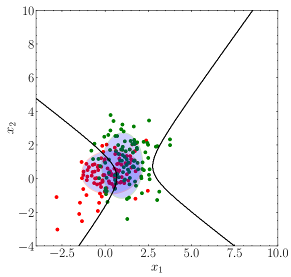
(f) Discriminant, size 25.



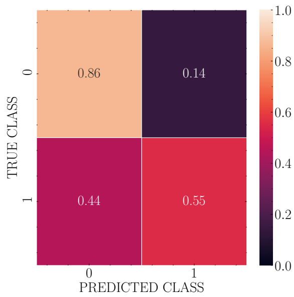
(g) QDA on P1a, size 75.



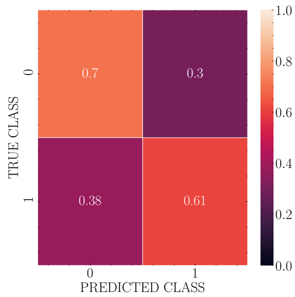
(h) KNN on P1a, size 75.



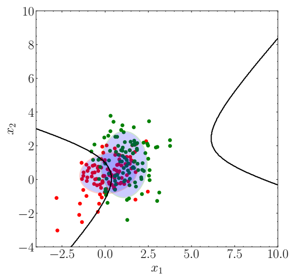
(i) Discriminant, size 75.



(j) QDA on P1a, size 199.

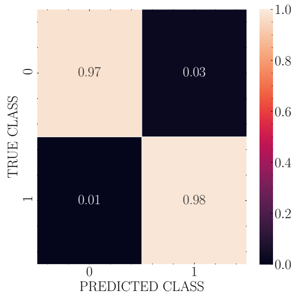


(k) KNN on P1a, size 199.

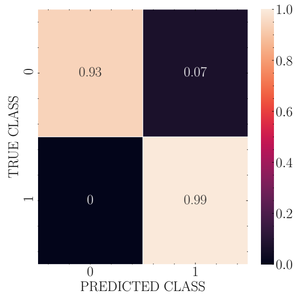


(l) Discriminant, size 199.

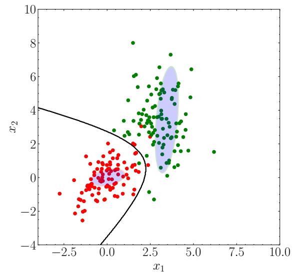
Figure 1: Bayes classifier and Nearest neighbour classification on P1a dataset.



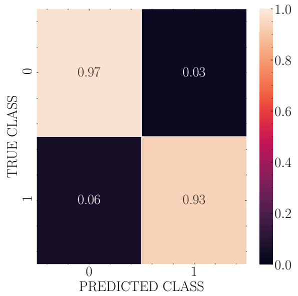
(a) QDA on P1b, size 10.



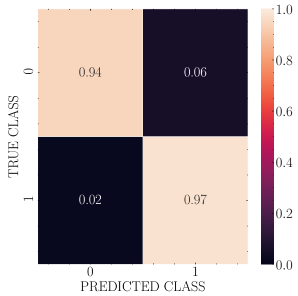
(b) KNN on P1b, size 10.



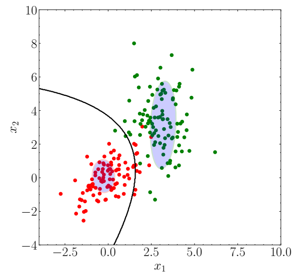
(c) Discriminant, size 10.



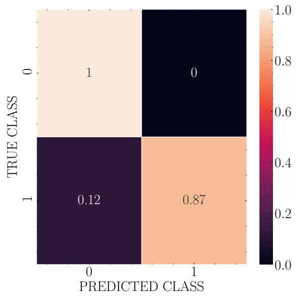
(d) QDA on P1b, size 25.



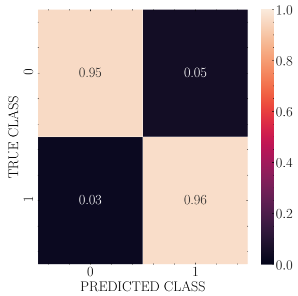
(e) KNN on P1b, size 25.



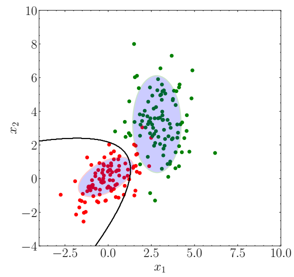
(f) Discriminant, size 25.



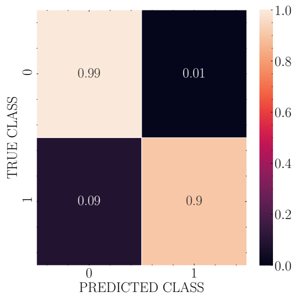
(g) QDA on P1b, size 75.



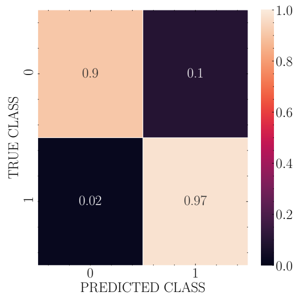
(h) KNN on P1b, size 75.



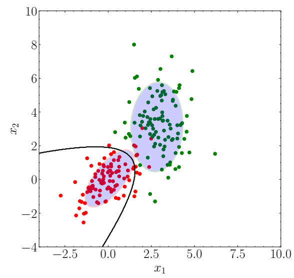
(i) Discriminant, size 75.



(j) QDA on P1b, size 199.

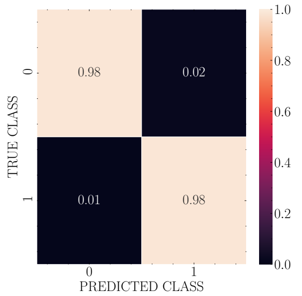


(k) KNN on P1b, size 199.

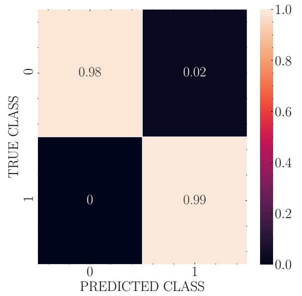


(l) Discriminant, size 199.

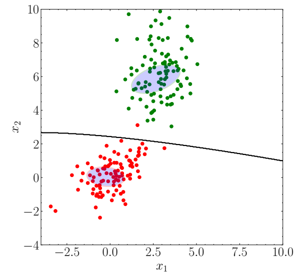
Figure 2: Bayes classifier and Nearest neighbour classification on P1b dataset.



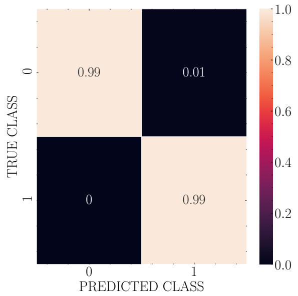
(a) QDA on P1c, size 10.



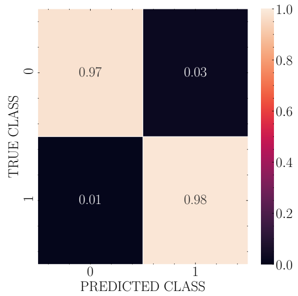
(b) KNN on P1c, size 10.



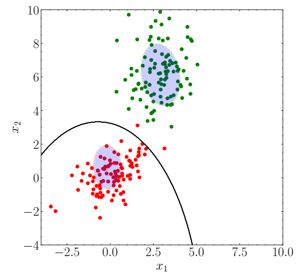
(c) Discriminant, size 10.



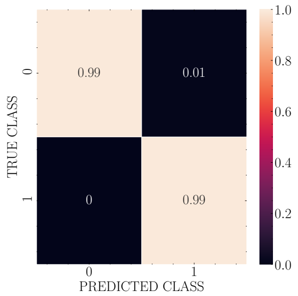
(d) QDA on P1c, size 25.



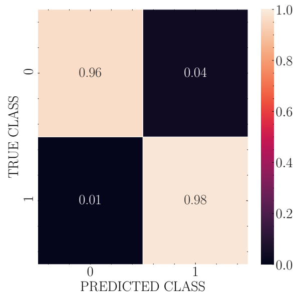
(e) KNN on P1c, size 25.



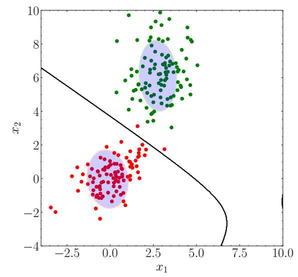
(f) Discriminant, size 25.



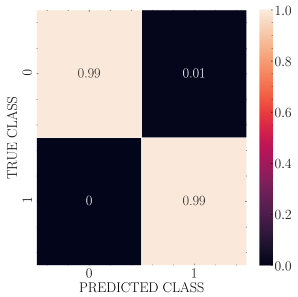
(g) QDA on P1c, size 75.



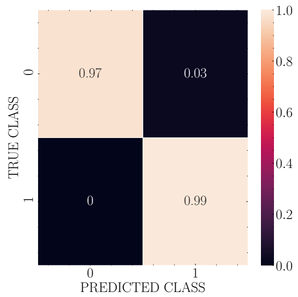
(h) KNN on P1c, size 75.



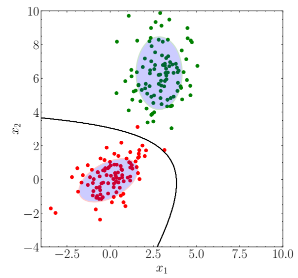
(i) Discriminant, size 75.



(j) QDA on P1c, size 199.



(k) KNN on P1c, size 199.



(l) Discriminant, size 199.

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

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.

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

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

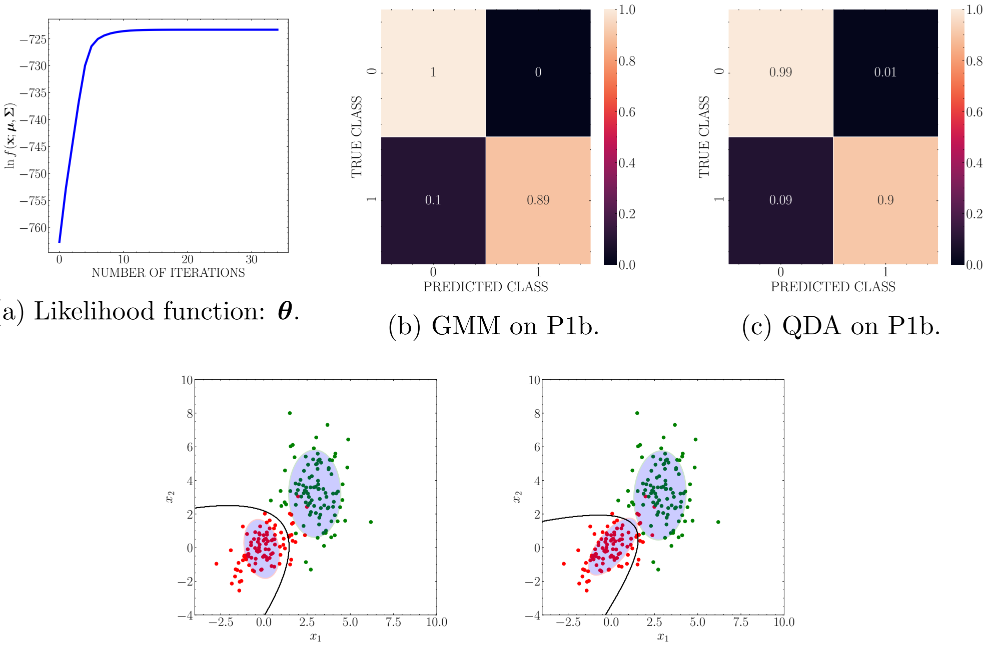
where the $(k+1)$st update for the priors, means and covariances are obtained in the M-step with:

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

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.

Figure 4 shows the results for testing the classifier on P1b dataset. Figure 4a shows the log-likelihood function 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) Likelihood function: $\boldsymbol{\theta}$.

(b) GMM on P1b.

(c) QDA on P1b.

(d) Discriminant, GMM.

(e) Discriminant, 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

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. 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).

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

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

4 Naive Bayes' Classifier for Document Classification