Report: Assignment 2

Submitted by: Arulkumar S, Shitanshu Kusmakar
CS15S023, ED15F003

25-Sep-2015

1 Introduction

1.1 Problem Definition

A bayesian classifier with a simple assumption of independence among different features shows good classification performance in supervised learning tasks [1]. In this assignment we evaluate and build a Bayesian classifier by learning the theory of bayesian networks. Bayesian networks are probabilistic representation of the data points belonging to a particular probability distribution. We further validate the build model by plotting ROC and DET curves. Bayesian classifier showed 82% classification accuracy on the real world multi class data.

2 Approach to the problem

2.1 What is the given information

A Bayesian classifier has to be build and the discriminant function has to be tested for a different number of special cases. The general multivariate normal density is as shown in equation 1.

$$p(x) = \frac{1}{2\pi^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} exp[-\frac{1}{2}(x-\mu^t)\Sigma^{-1}(x-\mu)] \tag{1}$$

where, x is the d-component feature vector, μ is the mean vector with d-components, Σ is the $d \times d$ covariance matrix, $|\Sigma|$ and Σ^{-1} are the determinant and the inverse.

The five cases for which the discriminant has to be tested are

- Bayes with covariance matrix (Σ) same for all classes. So, mean of all covariance matrix is taken.
- Bayes with covariance matrix (Σ) different for all classes.
- Naive bayes, when $\Sigma = \sigma^2$. So, first we took the maximum covariance from all covariance matrices and then converted it to a diagonal matrix of the size of covariance matrix.
- Naive bayes with Σ same for all classes. So, we took the mean and we did element wise multiplication with the identity matrix of the same size to get the covariance matrix.
- Naive bayes with Σ different for all. So, we did element wise multiplication of the covariance matrix for each class with an identity matrix of the same size.

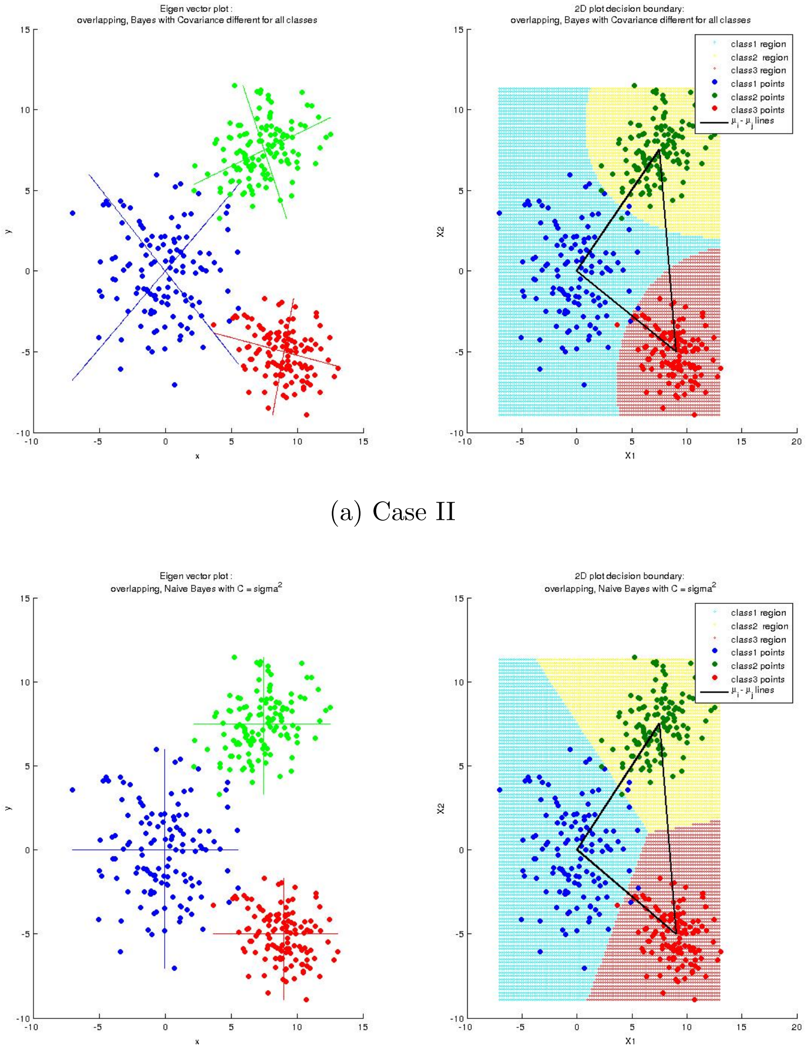
2.2 Approach

The approach to the problem was based on following things

- The data was divided into training and test set. 75% of the data was selected as training set and 25% as test.
- So, to build a Bayesian model we first calculate the gaussian parameters μ and Σ from the training data.
- Then, the covariance matrix is set based on the five different cases.
- Now, we have the model and using the model parameters we test classification accuracy on the test data.
- Then, the model is verified by calculating the confusion matrix, classification accuracy, sensitivity, specificity, precision and F1-score.
- Then, we calculated cosine similarity for the predicted labels and the target labels to understand how similar they are.
- We plotted the decision region boundary with the line connecting the means of different clusters, eigen vectors were also plotted for all the data clusters. The eigen vectors were calculated for the covariance matrix of each cluster. We also, plotted gaussians of different classes alongwith the ROC and DET curves for different classes.

3 Results & Observations

The figure 1 shows the data clusters with the decision boundary and overlapping eigen vectors of the covariance matrix. The eigen vectors points in the direction of maximum variance. However, under case II the eigen vectors are not orthogonal, whereas in Naive bayes the eigen vectors are always orthogonal. The reason, for such a measure can be accounted from the covariance matrix of the data. The covariance matrix in Naive Bayes is a diagonal matrix and hence the eigen vectors are standard basis vectors.



(a) Case II

(b) Case III

Figure 1: Eigen vector and decision boundary (a) Case II (b) Case III

The figure 2 shows the decision boundary plot for the case III in linearly separable data. It can be inferred from the figure

- Naive Bayes with covariance same across all features ($\Sigma = \sigma^2 I$) the line connecting the means of every class cluster is perpendicular to the decision boundary.
- The decision boundary is a perpendicular bisector of the line connecting the means of clusters.
- The same observation is not true for non-linearly separable data as seen in figure 1 (b).
- Therefor, if the covariance is same for all classes $\Sigma_i = \Sigma$ the line connecting the means of different cluster need not be perpendicular to the decision boundary, however it will be bisected by the decision boundary as seen in figure 1 (b).

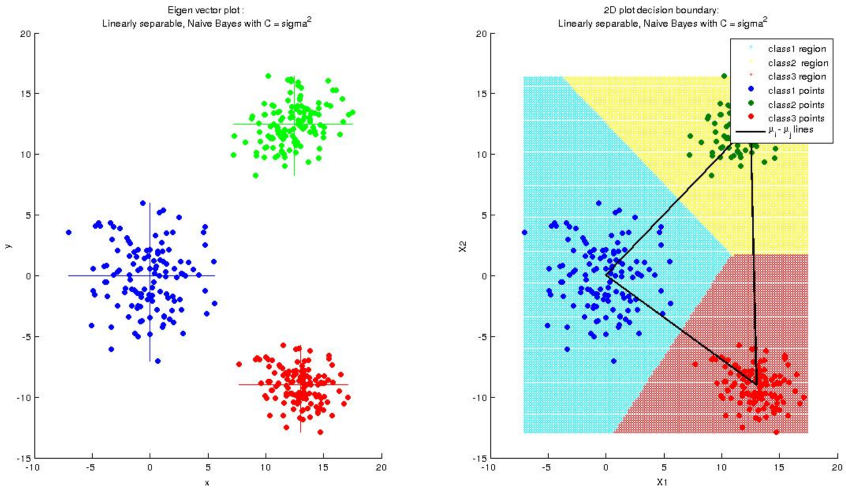
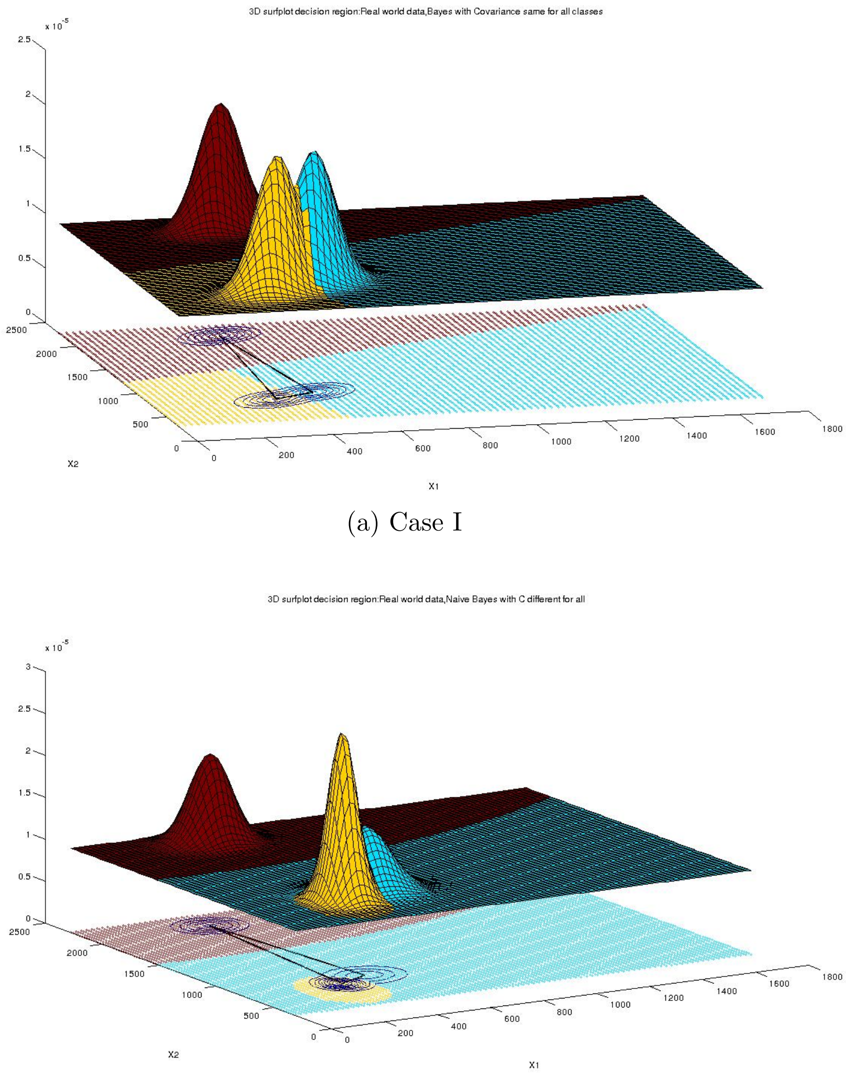


Figure 2: Decision boundary plots for linearly separable data for case I.

The figure 3 shows the probability plot in 3-D for all three classes. It can be inferred from the two probability density plots that when we use different covariance matrices (case II) the decision boundary is non-linear as seen in figure 3 (b) and when the covariance matrix is same for all the classes the decision boundary is linear as seen in figure 3 (a). Also, the high peak of the probability density plot seen as in figure 3 (b) is due to the different prior probabilities observed for the different classes.

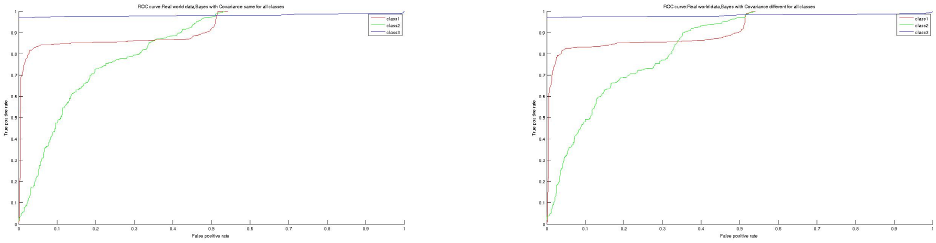


(a) Case I

(b) Case V

Figure 3: Probability density plots

The figure 4 shows the ROC curves. ROC curves are ploted between false positive rate (FPR) and true positive rate (TPR). We used each data points probability as threshold and calculated the number of TP's and FP's according to the threshold and on iterating for all data points we get an array of TPR and FPR. The TPR reaches 1 faster (*i.e* for smaller values of FPR) for class where the data is accurately classified. Also, the ROC curves showed the similar behaviour for the two cases I and II, which means that in both cases the data was separated with fairly similar accuracy.

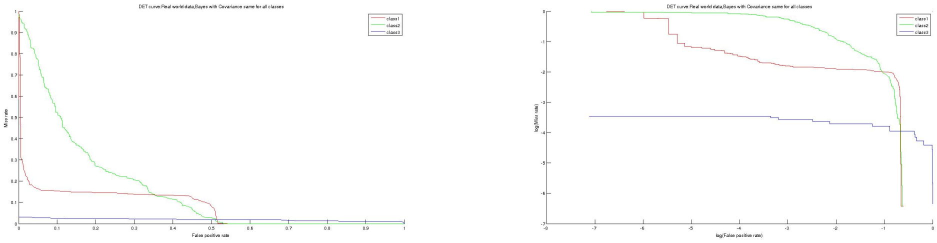


(a) Case I

(b) Case II

Figure 4: ROC plots

The figure 5 shows the DET plots. DET curves are alternatives for ROC curves. DET shows Miss vs False alarms. DET curves are assumed to be better estimates as the values are non-linearly transformed.



(a) Case I-DET Plot

(b) Case I-log plot of DET

Figure 5: DET plots

Table 1 shows the confusion matrix for the real world data set for case II.

Table 1: Confusion matrix.

Class	**ClassI predicted**	**ClassII predicted**	**ClassIII predicted**
ClassI-target	406	208	0
ClassII-target	91	531	0
ClassIII-target	11	8	554

Table 2 shows the performance measure of the classifier for the real world data set for case II.

Table 2: Table showing the performance measures for the classifier.

Measure	**ClassI**	**ClassII**	**ClassIII**
Sensitivity	0.66	0.85	0.96
Specificity	0.92	0.82	1.00
Precision	0.80	0.71	1.00
F1-score	0.72	0.78	0.98
Overall Accuracy	0.824		

4 Conclusion

As seen from the results the Bayesian classifier is a good algorithm for performing classification tasks involving multi class data. We performed various experiments based on which we observed that the nature of the decision boundary is dependent on the covariance matrix. For, a special case when $\Sigma = \sigma^2 I$ the decision boundary is a perpendicular bisector of the line joining the means of the clusters. Using the Bayesian classifier built in case II, we achieved a classification accuracy of 82% on the real world data set.

References

[1] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine learning*, 29(2-3):131–163, 1997.