Assignment 05

Preetam Kumar Ghosh
CS17M033

Amar Vashishth
CS17M052

November 23, 2017

Abstract

Table 1: Model Accuracy Table

Dataset	Accuracy (in %)
Parzen Window	64.0
Fisher Discriminant Based Classifier	76.6
Perceptron Based Classifer	60.8
SVM (OCR)	41.6927
SVM (Speech)	83.6015
Neural Network (OCR)	40.4
Neural Network (Speech)	87.0

1 Parzen Window

In this method, we define a function such that

$$\phi(u) = \begin{cases} 1 & \text{if } |u_i| \leq 0.5,\ i = 1, 2, 3, \ldots, d \\ 0 & \text{otherwise} \end{cases}$$

where, $u = (u_1, u_2, \ldots, u_d)^T$. The function ϕ is a mapping such that, $\phi : \mathbb{R}^d \to \mathbb{R}^d$

This is a definition of a hypercube in $\mathbb{R}^d$, centered at the origin; The function ϕ is an indicator of a data point being inside that hypercube. It is also called as the Kernel Density Estimator. We have used a Gaussian Kernel for the given data. The Kernel density estimator used is given as:

$$\phi(u) = \left(\frac{1}{\sqrt{2\pi}}\right)^d \exp\left\{-\frac{1}{2}||u||^2\right\}$$

with this the hypercube volume, $V = h^d$. The density estimator is:

$$f(x) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{h\sqrt{2\pi}}\right)^d \exp\left\{-\frac{||x - x_i||^2}{2h^2}\right\}$$

This is a kind of mixture density where, instead of choosing a gaussian mixture, we choose exactly n gaussians and centering gaussian at the data point. The advantage of using such a Kernel density estimator is that it gives us a smoother density estimate.

Problems with this kind of estimator: Decision for choosing a value for h is difficult, we can also try choosing different values of h at different regions of the feature space.

	Forest	Opencountry	Tallbuilding
Forest	**72**	**25**	**16**
Opencountry	**10**	**77**	**28**
Tallbuilding	**17**	**21**	**63**

Figure 1: Confusion Matrix for Parzen Window @h=0.08

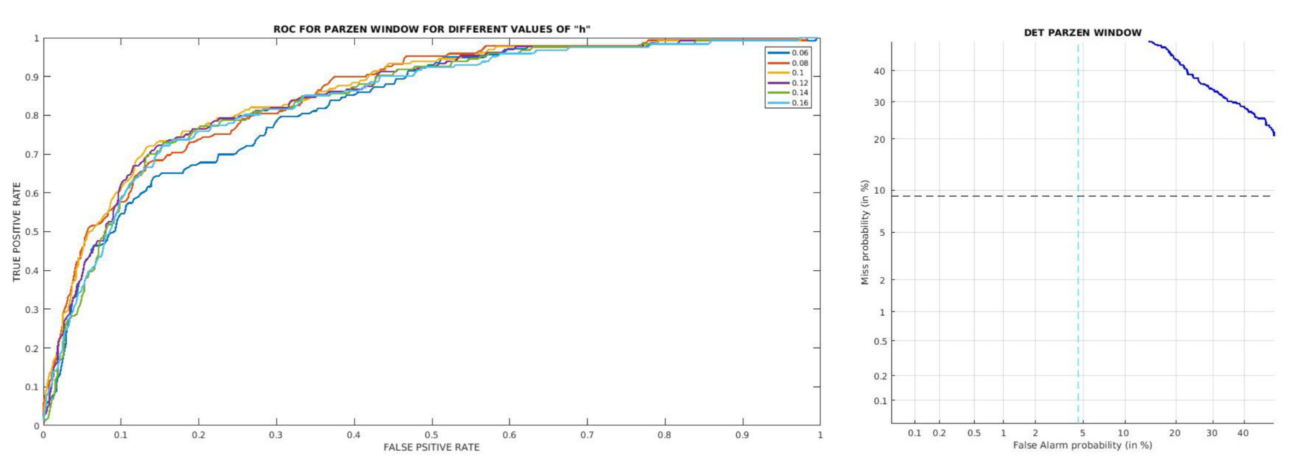

Figure 2: ROC with different values of h & DET(with h = 0.08) Plots for Parzen Window

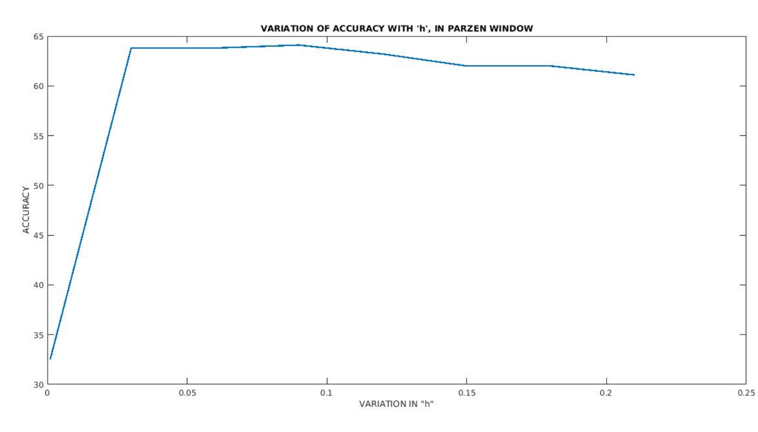

Figure 3: Parzens's Side of Cube, h Vs Accuracy Plot

2 Fisher Discriminant Based Classifier

It is a way of constructing linear classifier. It decides that a data item $X \in$ Class 1 if $W^T X + w_0 \geq 0$. Here, W is a vector hence, we can take it as a direction in $\mathbb{R}^d$ space. $W^T X$ is projected value of X onto the direction W.

The question which are interested to answer here is - how to find that W which maximizes the separation between the classes. We can understand it by this figure below:

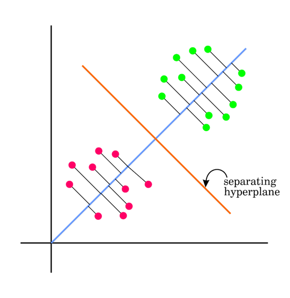

Figure 4: Fisher Discriminant Based Classifier

To compute such a W, we consider a binary classification problem. So we have two classes, namely Class 1 and Class 2. Let, M_1 and M_2 be the means for each class. These are computed using the training data items, as:

$$M_1 = \frac{1}{n_1}\sum_{X_i \in C_1} X_i \qquad M_2 = \frac{1}{n_2}\sum_{X_i \in C_2} X_i$$

Also, we compute a real symmetric matrix(hence, would be invertible) which is given as,

$$S_w = \sum_{X_i \in C1}(X_i - M_1)(X_i - M_1)^T + \sum_{X_i \in C2}(X_i - M_1)(X_i - M_2)^T$$

finally, the W matrix is computed as:

$$W = S_w^{-1}(M_2 - M_1)$$

We have a binary classifier, but we have to classify data into multiple classes. To do this we have reduced this problem to subproblems of binary classification.

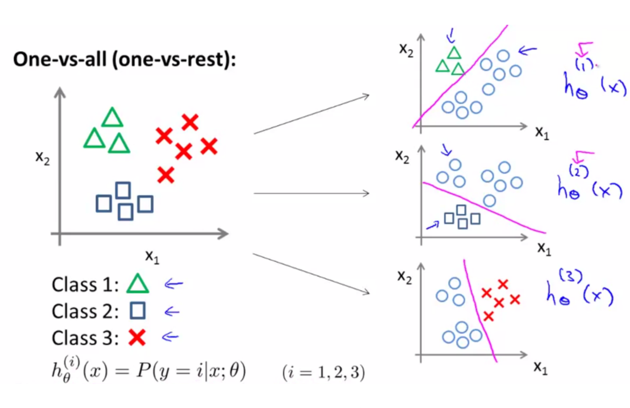

Figure 5: Multiclass Classification

	Forest	Opencountry	Tallbuilding
Forest	**90**	**54**	**1**
Opencountry	**4**	**61**	**5**
Tallbuilding	**5**	**8**	**101**

Figure 6: Confusion Matrix for Fisher Discriminant Based Classifier

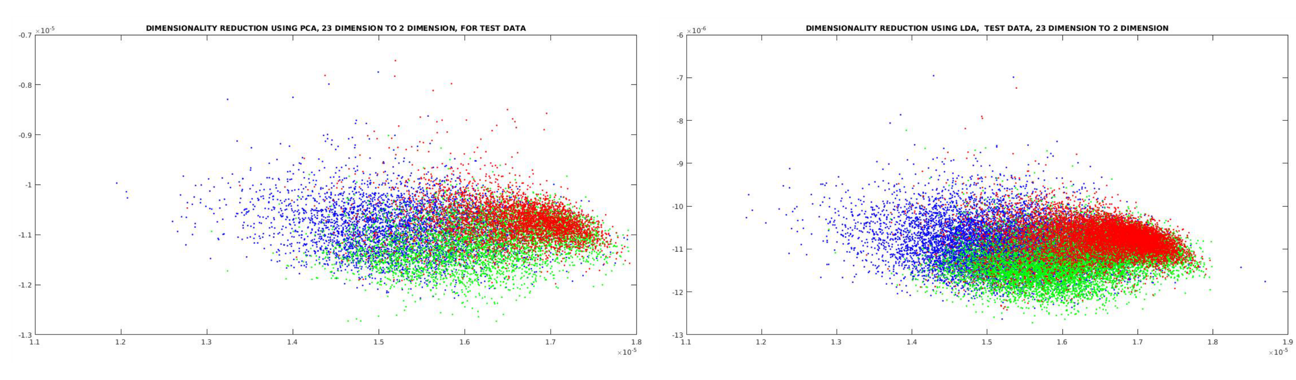

Figure 7: Projected Testing and Training data of all Classes onto 2D

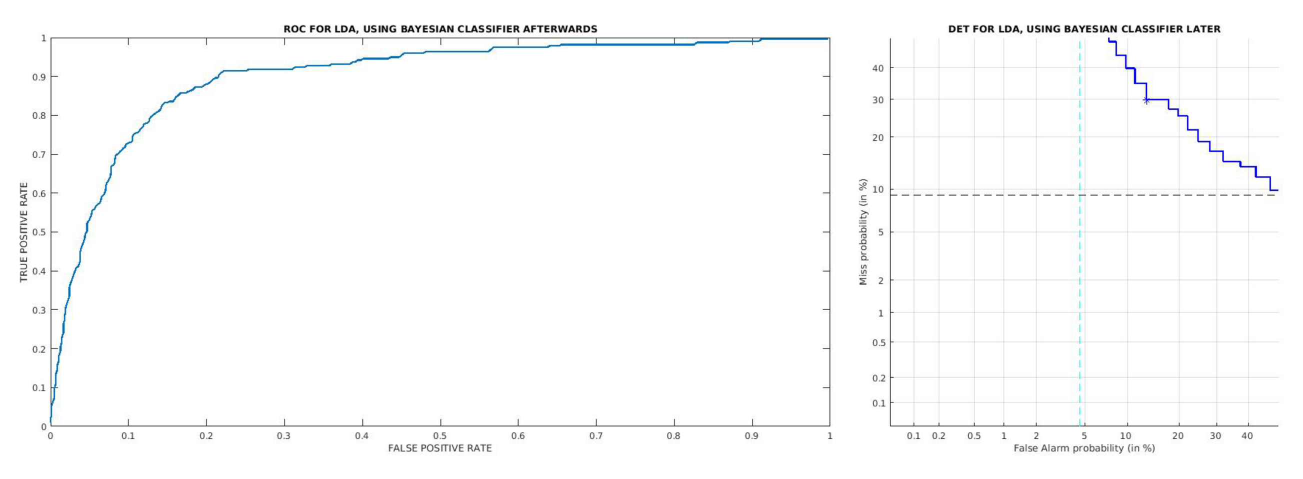

Figure 8: ROC & DET Plots for Fisher Discriminant Based Classifier

3 Perceptron Learning Algorithm

It is an iterative algorithm to learn W corresponding to a hyperplane. By adding the incorrectly classified vector to the current hyperplane. We are actually rotating the hyperplane about the origin towards that point, hopefully which could also lead to its classification in its appropriate group.

We continue to add such data points until we obtain a separating hyperplane.

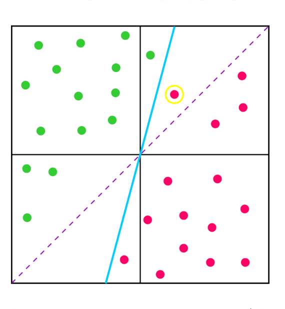

Figure 9: Perceptron Learning Algorithm

Design of Weight Vector W: Assume two category of linearly separable classes, not in homogeneous form. The discriminant function takes the form:

$$g(x) = W^T X + w_0 = a^T\ y$$

a = modified weight vector has components of W and w_0.
y = augmented feature vector x

$$y = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \\ 1 \end{bmatrix} \quad \text{and} \quad a = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \\ w_0 \end{bmatrix}$$

Update Step:

$$y_i = \begin{cases} \text{Augmented } x_i & \text{if } x_i \in \omega_1 \\ -\text{Augmented } x_i & \text{if } x_i \in \omega_2 \end{cases}$$

$$a(k+1) = a(k) + \eta(k)\sum y \quad \forall y \in \text{misclassified}$$

	Forest	Opencountry	Tallbuilding
Forest	**61**	**27**	**11**
Opencountry	**31**	**57**	**14**
Tallbuilding	**7**	**39**	**82**

Figure 10: Confusion Matrix for Classifier using Perceptron Learning Algorithm, took 1700 Iteration

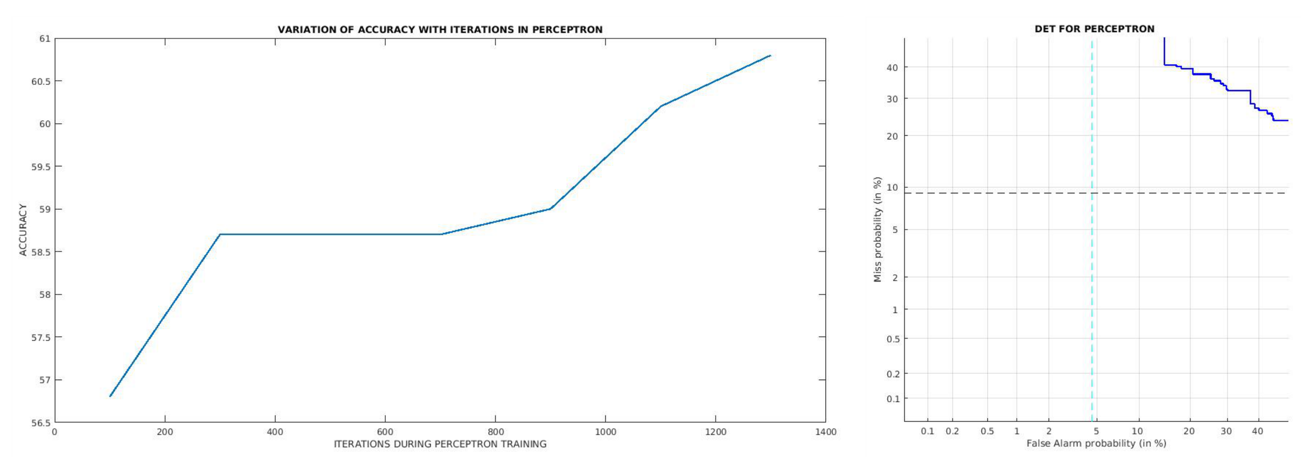

Figure 11: Perceptron Variation of Accuracy and DET Plot

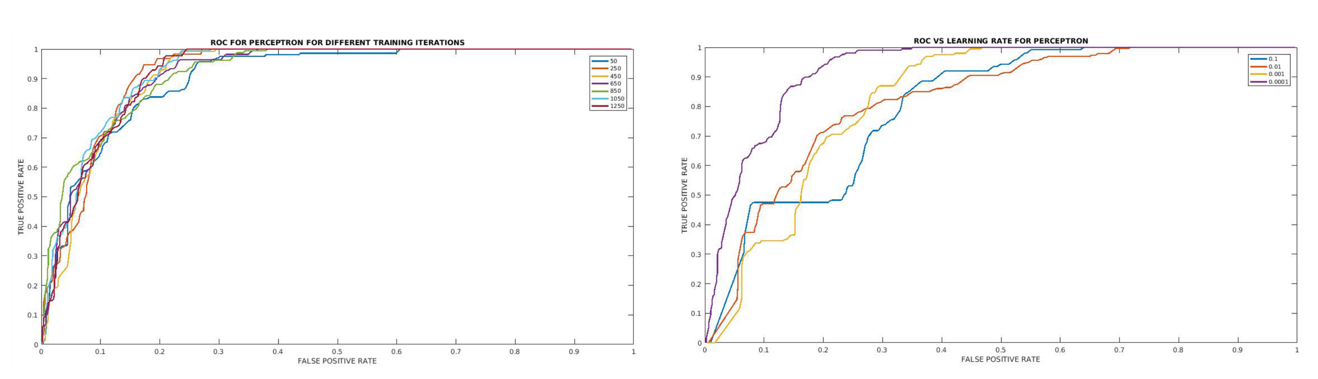

Figure 12: Perceptron ROC Vs Iteration & ROC Vs Learning Rate

4 Support Vector Machine

These are non linear classifiers. The idea is to map the feature vectors non linearly into another space and learn a linear classifier there.

$$y_i = \begin{cases} +1 & \text{for +ve samples} \\ -1 & \text{for -ve samples} \end{cases}$$

Mathematically, we can write the classifier as:

$$y_i(\vec{w}\vec{x_i} + b) - 1 \geq 0$$

for $\vec{x_i} \in$ gutter are: $y_i(\vec{w}\vec{x_i}) - 1 = 0$

Then, we try to find out how to arrange the line such that we get the widest gutter(street). That's why the svm approach is also called widest street approach. To get the widest street:

$$\text{minimize} \frac{||\vec{w}||^2}{2}$$

answer to which we get,

$$\vec{w} = \sum_i \alpha_i y_i x_i = 0$$

where, α_i is the lagrange's multiplier there.

Finally on putting things back to our decision rule equation we obtain its final form:

$$\boxed{\sum \alpha_i\ y_i\ \vec{x_i} \cdot \vec{u} + b \geq 0 \quad \text{Then, +ve sample}}$$

	a	chA	tA
a	**1343**	**0**	**905**
chA	**945**	**0**	**1231**
tA	**902**	**0**	**1409**

Figure 13: Confusion Matrix for OCR data

	Digit 1	Digit 2	Digit 5
Digit 1	**1394**	**67**	**150**
Digit 2	**166**	**1277**	**94**
Digit 5	**248**	**70**	**1382**

Figure 14: Confusion Matrix for Speech data

5 Multi-layer Feed-forward Neural Network

Here we provided two sets of data, we have used a neural network for classifying them, 100 neurons were used in the process.

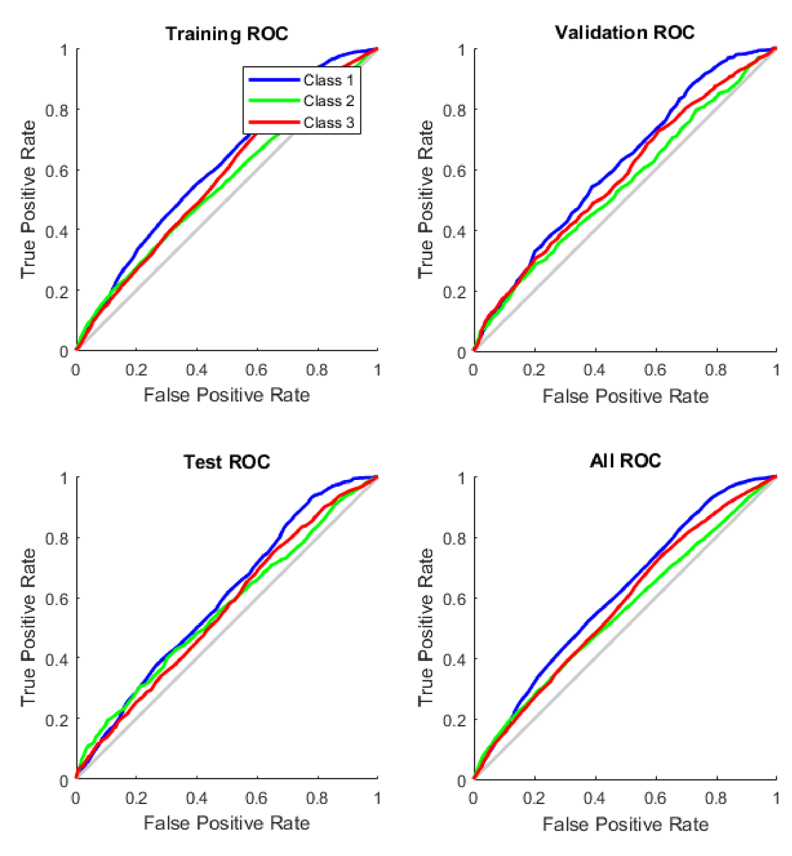

Figure 15: ROC Plot for OCR Data

	a	chA	tA
a	**325**	**271**	**248**
chA	**4**	**79**	**49**
tA	**465**	**475**	**606**

Figure 16: Confusion Matrix for OCR data

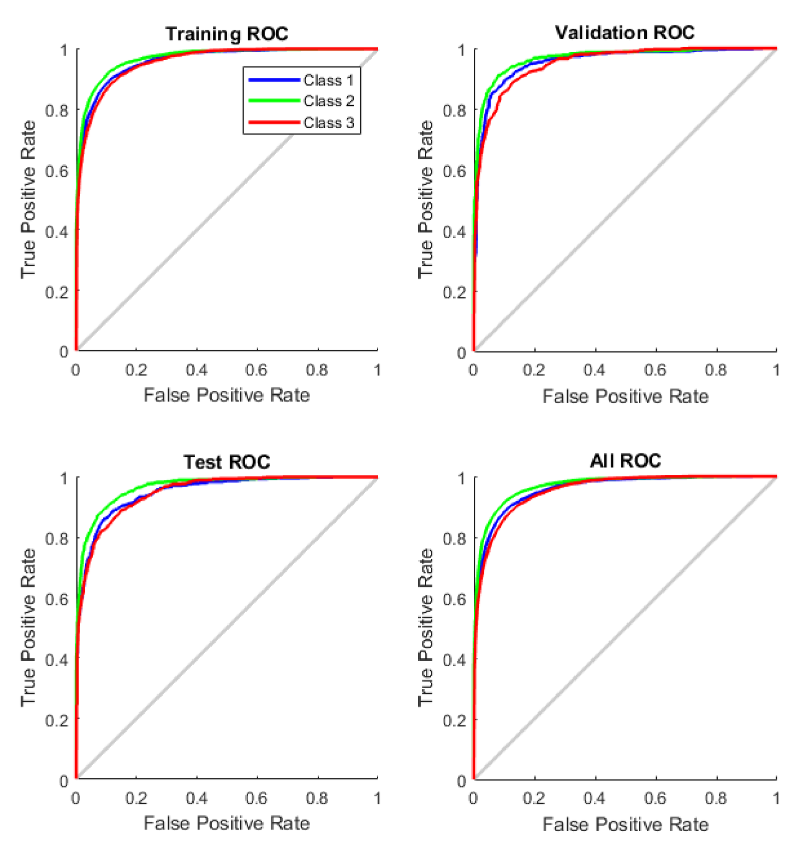

Figure 17: ROC Plot for Speech Data

	Digit 1	Digit 2	Digit 5
Digit 1	**656**	**45**	**86**
Digit 2	**45**	**674**	**59**
Digit 5	**85**	**66**	**708**

Figure 18: ROC Plot for Speech Data

6 Inference

- In support vector machines, we utilized features which were local to each point hence the accuracy went down to ≈40%
- In Perceptron, the accuracy obtained was not good enough because we attempted to separate real data using hyperplanes.