K-Nearest Neighbours, Support Vector Machines, Logistic Regression, and Artificial Neural Networks

Programming Assignment 4 Report
CS5691 - Pattern Recognition and Machine Learning

Team 29 - Abhigyan Chattopadhyay (EE19B146) & Nihal John George (EE19B131)

May 3, 2022

1 Preprocessing for Time Series Data

We used the resample() function provided by Scipy to make all time series examples of the same length. Resample internally uses frequency information in the signal to extrapolate signals. This approach was chosen over padding since padding inherently changes the nature of the signal, especially in the case of handwriting where any sort of padding implies a position of the pen which changes the character.

Padding vs Resampling Time Series Data: We found that the system scored a surprisingly good accuracy by merely padding the data with extra zeros. While we were able to improve upon it significantly by using resampling, it would probably have been OK to just use padding in a system with stricter memory or time constraints.

2 Principal Component Analysis

2.1 Inferences

- **Character Dataset**: Interestingly, PCA doesn't affect the accuracies obtained in the character dataset at all. This is because the data is inherently the 2D position of the n points even though it is arranged as an $n \times 2$ dimensional input (n being the number of points sampled for a given character)
- **Synthetic Dataset**: Similarly, PCA doesn't affect the accuracy of the synthetic dataset by much due to it already being 2D.
- **Spoken Digits Dataset**: It appears that the accuracy on this dataset increases upon using PCA, while it remains the same or reduces while using LDA. This is consistently observed across all 4 classifiers that we used.
- **Image Dataset**: PCA slightly reduces the accuracy on SVM and ANN while it increases the accuracy while using Logistic regression. This is probably due to the fact that we are leaving important components while choosing the top 18 principal components of such a high-dimensional (828-D) dataset.

3 Fisher's Linear Discriminant Analysis

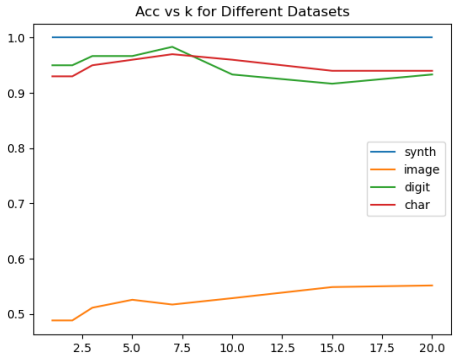
3.1 Inferences

- **Character Dataset**: LDA and raw had the same results in this case, once again due to the fact that it is based on inherently 2D data which is already low in dimensionality.
- **Synthetic Dataset**: Slight improvements are seen on using LDA in SVM and ANNs with different hyperparameters, but in general it is negligibly better in other cases.

- **Spoken Digits Dataset**: It works better than PCA when we look at the padded data, possibly due to random overfitting, but in the resampled data, it doesn't do much better.
- **Image Dataset**: Images are again classified better by PCA than LDA in general. However, LDA always gives competitive scores with respect to PCA, and in the case of using a smaller learning rate in the Neural network case, it performs better than PCA.

4 K-Nearest Neighbours

Here, we compute the Euclidean distance between the test feature vector and each train feature vector. The closest k vectors are used in a voting process to decide the predicted class. The DTW + Majority voting scheme used in Assignment 3 (results not shown here) had outperformed KNN.



Hyperparameters used	Dim Red Technique	Synthetic	Image	Character	Spoken Digits
KNN @ k=1	PCA	100	48.85	93	95
KNN @ k=2	PCA	100	48.85	93	95
KNN @ k=3	PCA	100	51.15	95	96.67
KNN @ k=5	PCA	100	52.59	96	96.67
KNN @ k=7	PCA	100	51.72	97	98.33
KNN @ k=10	PCA	100	52.87	96	93.33
KNN @ k=15	PCA	100	54.88	94	91.67
KNN @ k=20	PCA	100	55.17	94	93.33

Figure 1: Variation of Accuracy with Learning Rate on different datasets

4.1 Inferences

1. As k increases from 1 to 20, initially the accuracy is lower since the model is sensitive to noisy examples, as seen in the image dataset performance. This sensitivity can be interpreted as overfitting
2. After a point, increasing k decreases accuracy. This can be interpreted as high bias, since the model is considering more global information than local

5 Logistic Regression

Logistic Regression with Polynomial basis functions was used. Each feature was raised to a powers upto a max-degree to create new features. We changed hyperparameters such as learning rate, regularization factor and iteration count.

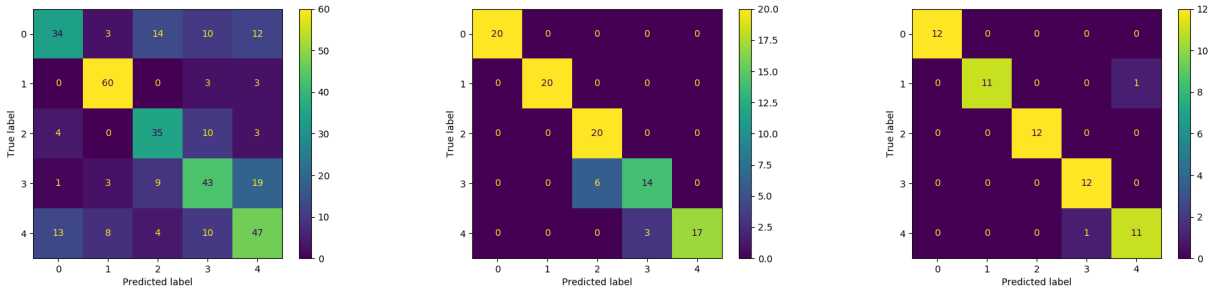
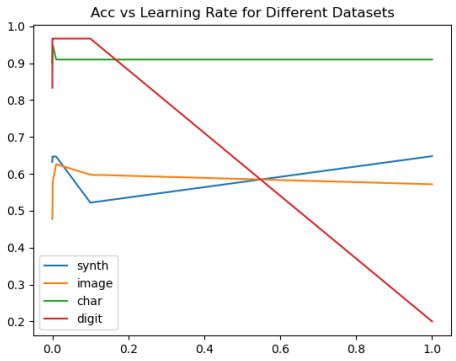


Figure 2: Confusion Matrices of Image, Characters and Digit Datasets respectively



Hyperparameters used	Dim Red Technique	Synthetic	Image	Character	Spoken Digits
Log Reg @ alpha=1e-6	PCA	63.2	47.7	90	83.33
Log Reg @ alpha=1e-5	PCA	63.2	47.7	90	93.33
Log Reg @ alpha=1e-4	PCA	64.7	49.13	91	96.67
Log Reg @ alpha=1e-3	PCA	64.7	57.75	95	96.67
Log Reg @ alpha=1e-2	PCA	64.7	62.64	91	96.67
Log Reg @ alpha=1e-1	PCA	52.2	59.77	91	96.67
Log Reg @ alpha=1e0	PCA	64.8	57.18	91	20

Figure 3: Variation of Accuracy with Learning Rate on different datasets

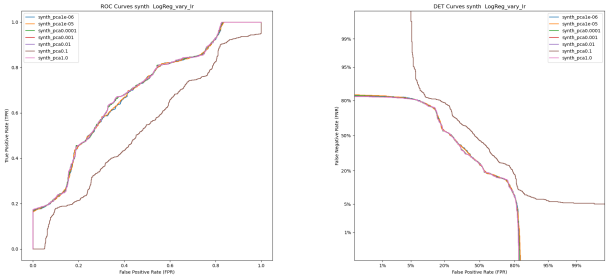


Figure 4: ROC and DET Curves on Synthetic Dataset

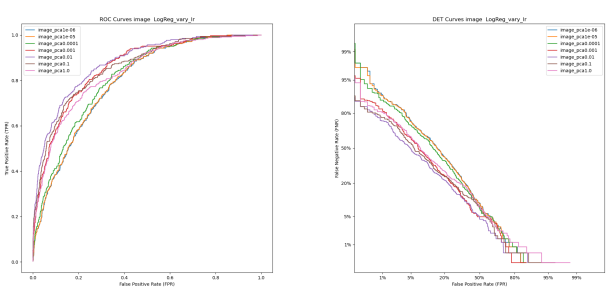


Figure 5: ROC and DET Curves on Images Dataset

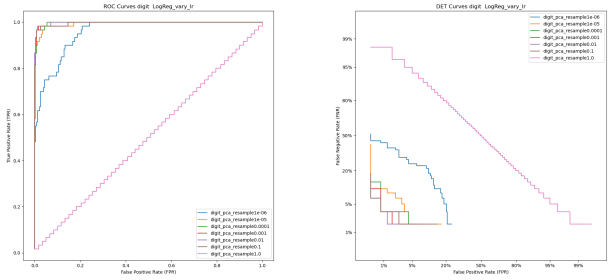


Figure 6: ROC and DET Curves on Spoken Digits Dataset

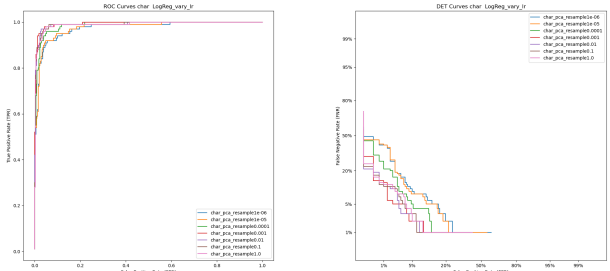


Figure 7: ROC and DET Curves on Handwritten Telugu Characters Dataset

5.1 Inferences

1. Logistic Regression finds linear decision boundaries in the feature space. Without extra transformation, this provides rather poor classifiers. Adding polynomial features helps to some extent, but for spiral data (synthetic), it was noticed that it was not enough, and a transformation involving polar coordinates and a sine function on distance to origin can be a good feature.

2. On increasing the learning rate, it first reduces the iterations required to reach the optimum value. However, if it becomes too large, the descent fails since the point is taken to the other side of the convex surface farther away from the goal. The gradient is even higher here, causing the distances to the goal to increase further.

3. On increasing iteration count, if the learning rate did not cause explosion in loss, the model got closer to convergence of loss

6 Support Vector Machines

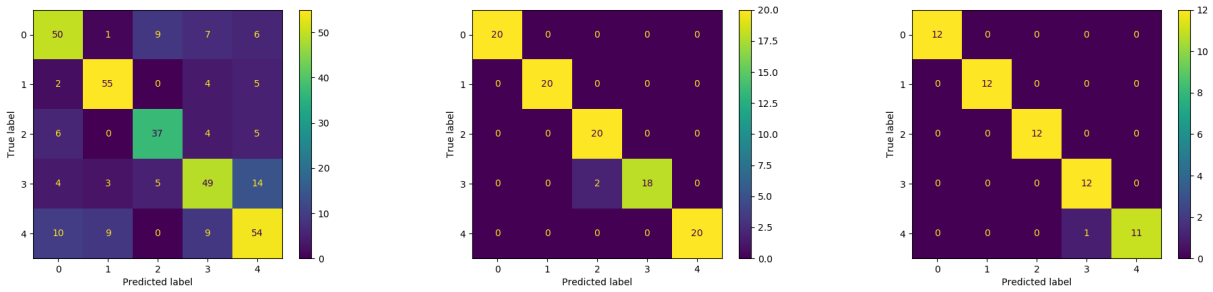


Figure 8: Confusion Matrices of Image, Characters and Digit Datasets respectively

Hyperparameters Used	Dim Red Technique	Synthetic	Image	Character		Spoken Digits	
				Padding	Resampling	Padding	Resampling
SVM @ C=1e7, rbf	Raw	100	77.59	92	98	91.67	98
	PCA	100	71.55	92	98	91.67	95
	LDA	100	70.4	92	98	93.33	95
SVM @ C=1e7, sigmoid	Raw	45.8	60.92	52	83	91.67	96.67
	PCA	49.4	54.31	52	83	80	98.33
	LDA	50	56.32	52	83	93.33	96.67
SVM @ C=1e3, rbf	PCA	97.8	77.59	92	98	91.67	95
	LDA	97.8	71.55	92	98	91.67	95
	Raw	97.8	70.4	92	98	93.33	95

Figure 9: Accuracies (in %) on classifying different datasets using SVMs

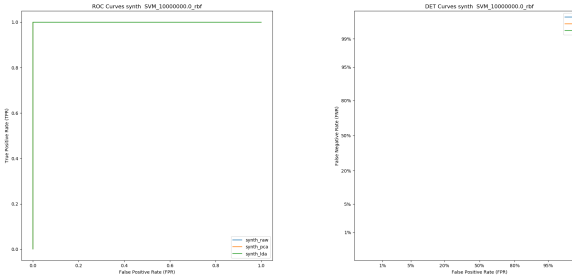


Figure 10: ROC and DET Curves on Synthetic Dataset

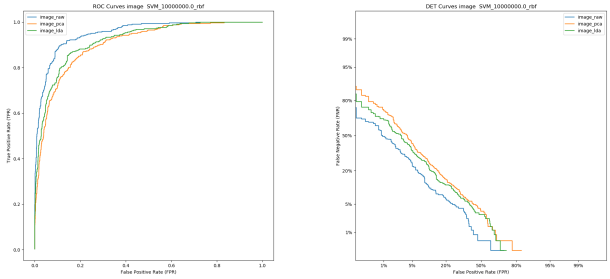


Figure 11: ROC and DET Curves on Images Dataset

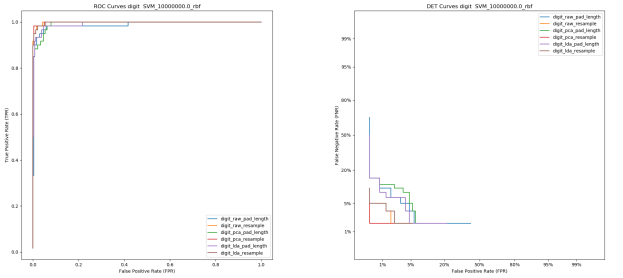


Figure 12: ROC and DET Curves on Spoken Digits Dataset

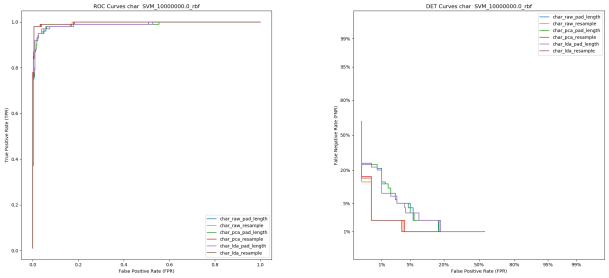
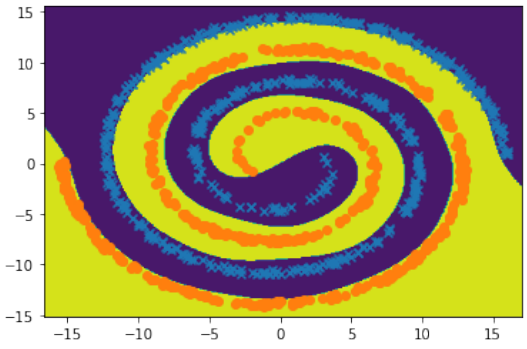


Figure 13: ROC and DET Curves on Handwritten Telugu Characters Dataset

6.1 Inferences

- **Basis Functions**
  - We are able to get good results on all datasets using the Radial Basis Function.
  - On using the Linear or Polynomial Basis functions resulted in Python freezing and becoming unresponsive, hence it was not possible to check with them.
  - Using a Sigmoid function, results were worse than the baseline in the synthetic dataset, and in general produced worse values than what we got using the Radial Basis Function.
- **Regularization Parameter**
  - The Regularization parameter has little influence on most datasets except the synthetic one.

In fact, SVM gives us some of the best results among all the algorithms we used, doesn't overfit the data as visualized using the synthetic dataset (shown below), and runs decently fast.



7 Artificial Neural Networks

Hyperparameters used	Dim Red Technique	Synthetic	Image	Character		Spoken Digits	
				Padding	Resampling	Padding	Resampling
ANN @ hidden_layer = (30,25,20), alpha = 1e-6, optimizer = "adam"	Raw	100	71.84	81	97	91.67	95
	PCA	100	61.78	81	97	80	98.33
	LDA	100	64.94	81	97	93.33	95
ANN @ hidden_layer = (30,25,20), alpha = 1e-6, optimizer = "sgd"	Raw	67	74.43	79	95	88.33	95
	PCA	65.3	58.62	79	95	88.33	96.67
	LDA	66	63.51	79	95	93.33	93.33
ANN @ hidden_layer = (20,15,10), alpha = 1e-6, optimizer = "adam"	Raw	100	69.25	83	98	85	91.67
	PCA	99.9	65.52	83	98	90	98.33
	LDA	99.9	65.52	83	98	88.33	96.67
ANN @ hidden_layer = (30,25,20), alpha = 1e-3, optimizer = "adam"	Raw	100	70.69	81	97	91.67	95
	PCA	100	58.62	81	97	88.33	98.33
	LDA	100	65.52	81	97	93.33	95

Figure 14: Accuracies (in %) on classifying different datasets using ANNs

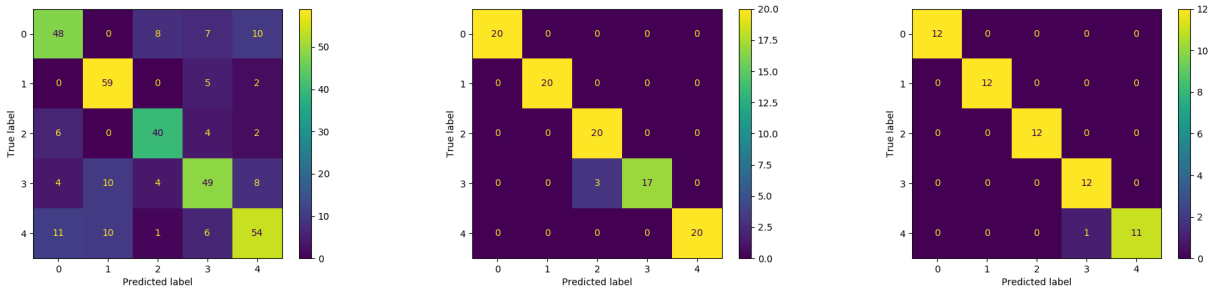


Figure 15: Confusion Matrices of Image, Characters and Digit Datasets respectively

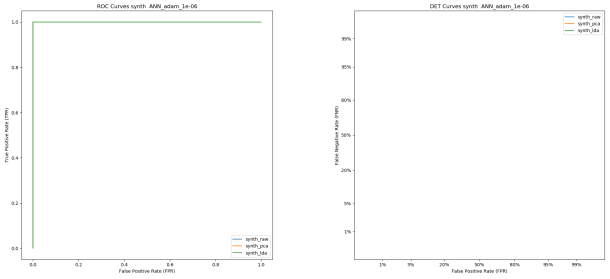


Figure 16: ROC and DET Curves on Synthetic Dataset

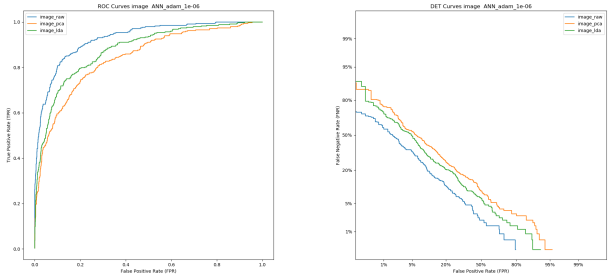


Figure 17: ROC and DET Curves on Images Dataset

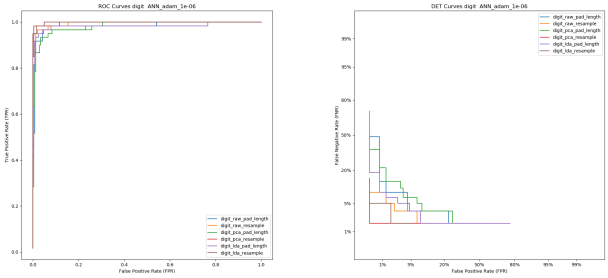


Figure 18: ROC and DET Curves on Spoken Digits Dataset

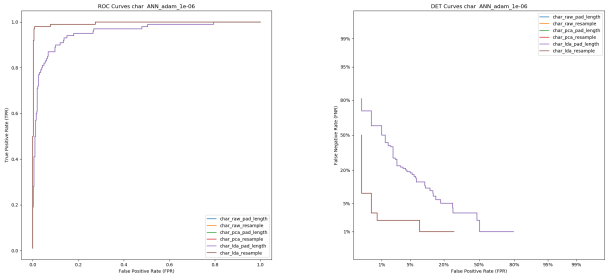


Figure 19: ROC and DET Curves on Handwritten Telugu Characters Dataset

7.1 Inferences

- **Network Size**
  - We found that using a network of shape (25,20,15) gave the best results in terms of time as well as accuracy. Other sizes tried included (50,25,12) and (10,5,2), the former was much slower and showed an overfit plot on the synthetic data, while the latter was underfitting the data.
- **Optimizing Function**
  - 3 optimizing functions were used through scikit-learn, i.e., Adam, Stochastic Gradient Descent and LBFGS. Of the three, Adam gave the quickest and most consistent results, while Stochastic Gradient Descent (sgd) had a massive underfit at any given neural network size and learning rate.
  - LBFGS was not as quick as Adam and wasn't possible to run on other datasets due to high dimensionality.
- **Learning Rate**
  - The optimal learning rate was found to be around 1e-6 or so, anything larger led to more underfit while larger values led to a very slow computation time.