Homework 4

By

Rishab Goel

MLP:

We apply MLP with supervised learning method as a reference to the other unsupervised learning techniques used further in this assignment. We use a single hidden layer network with 8 hidden PEs with stochastic gradient descent learning method with 500 epochs with adaptive learning rate starting with 1.05. If the new error exceeds the old error while training learning rate decreased by 0.6 otherwise increased by 1.05 while training. We divide the data into two classes as for unsupervised learning methods, of 1/3 initial data as test class while remaining 2/3 as the train class. We observe 100 % accuracy for all the train and test data partition tested, but present here the result for the partition successful for unsupervised learning methods.

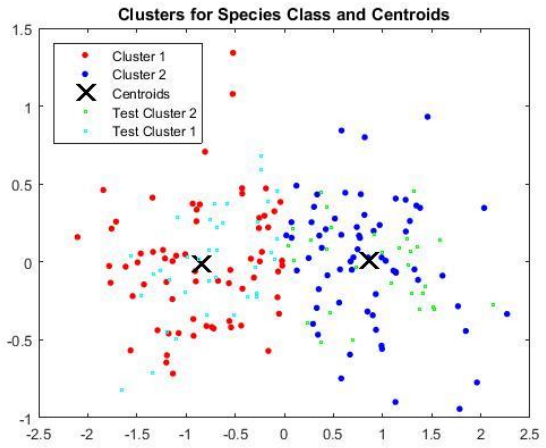
Confusion Matrix = 33 0 Accuracy = 100 %

0 34

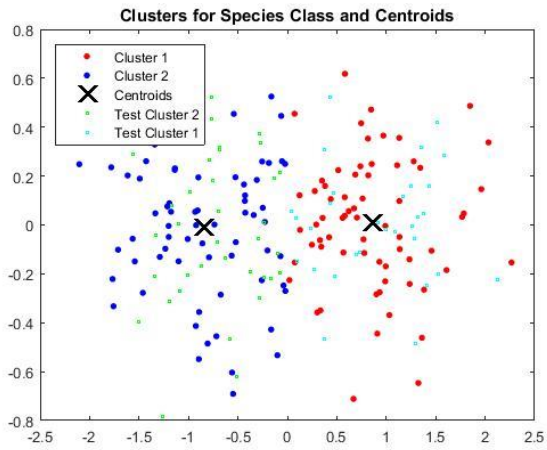
The data is a 5-dimension data, thus could not be plotted to plot a separation surface.

Method 1 :PCA

The principal component analysis is a classification and dimensionality reduction which could be petty efficient to find the separation surface to classify the data. This method is based upon the hypothesis that if we project the data into a separate or new 2D subspace using its covariance we could arrive to a subspace that would allow the data to be linearly separable. We currently apply this method the crab dataset we classify them into two species' classes. We observed best results separating the data into two sets first 1/3 of the input data being the test set and the remaining 2/3 being the train data set. The input data has five dimension so for visualization and effective classification it could be reduced to 2 dimension subspace. We creates two class clusters using kmeans using the projected data from the PCA output and use the Euclidean distance of the test/train data from the centroid to classify the data On applying the PCA algorithm it was found that 1st and 2nd eigenvectors of 5 dimension eigenvector which is generally predicted to separate the data effectively gives a train data accuracy of just 63.5 % and test data accuracy 55.5 %. We found out after testing various 2D subspace that the combinations of 3rd and 4th eigenvectors subspace as well as 3rd and 5th eigenvalue subspace gives us best test and train data accuracy.



Cluster for 3rd and 5th



Cluster for 3rd and 4th

The subspace projected by 3rd and 5th eigenvalue:

(Train Data)

Confusion Matrix = [62 4

5 62]

Accuracy = 93.23 %

(Test Data)

Confusion Matrix = [29 8

4 26]

Accuracy = **82.09 %**

The subspace projected by 3rd and 4th eigenvalue:

(Train Data)

Confusion Matrix=[62 4

5 62]

Accuracy = 93.23%

(Test Data)

Confusion Matrix=[29 8

4 26]

Accuracy = **82.09 %**

Comparison: For many instances and data set it has been observed that PCA gives good results. PCA is even used to preprocess the data before the MLP training for many data sets. Though we are comparing it with MLP on the criteria of effective classification, and its best results lags behind MLP by **18%**. Thus the PCA doesn't perform as well as MLP for classification of this crab dataset and the data set could be said to be non-linearly separable.

Method 2: SOM

Self Organizing Maps is an unsupervised learning method where we project the data into output subspace directly with iterative training of the weights. The classification within different clusters is done on the basis of the Euclidean distance of the test from the weights converging to that particular cluster. SOM creates inherently a two dimensional map of weights with the high Euclidean distances between weights signifying that they belong to a different cluster. The weights with less Euclidean distances would belong to the same cluster. Using SOM feature maps of Matlab we observe that darker regions signify the larger Euclidean distances between weights, and lighter yellow signify the small Euclidean distances between weights. The method, though make it difficult to classify the data and determine the accuracy of the classifications.

We didn't generate the cluster view of the weights on the basis of Euclidean distances for our implementation. We classify the train data into two sets of classes one into gender (Male or Female) and other into species (1 or 2), but were not able to get good accuracy for either of them. We generate the cluster of 2 and map of weights of 5x2 dimension and we increased the number of clusters it didn't help much in classification. We merge the extra clusters into two clusters classes for classification on the basis of the Euclidean distances. We changed the batch size from 1, and increase further found a sweet spot for respective batch sizes.

Gender Predictions with Two clusters batch size 2

Confusion Matrix = [64 61

36 39]

Accuracy = 51.50 %

Species Predictions with Two clusters batch size 4

Confusion Matrix = [56 34

44 66]

Accuracy = 61 %

Though when we come across literature which express SOM is effective for clustering than k-means and to experiment that theory we implement a PCA +SOM combination. While using the cluster using SOM for species classes we get 91 % accuracy and 91.5% accuracy for gender classes. The confusion matrix for species classes and gender classes are enumerated here:

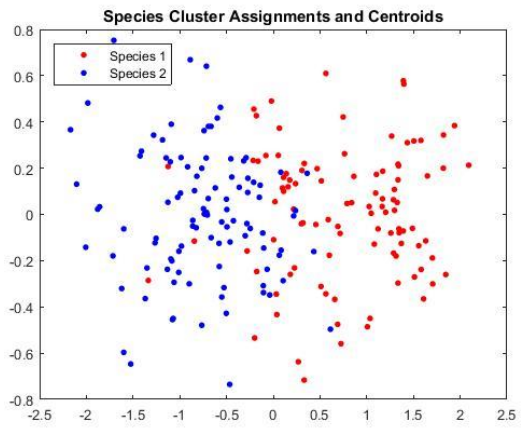
Confusion Matrix = [91 9

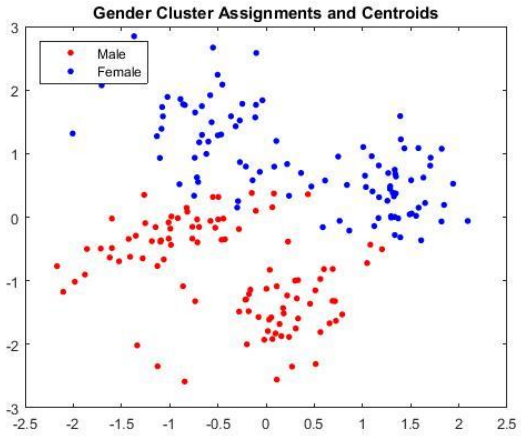
(Species) 9 91]

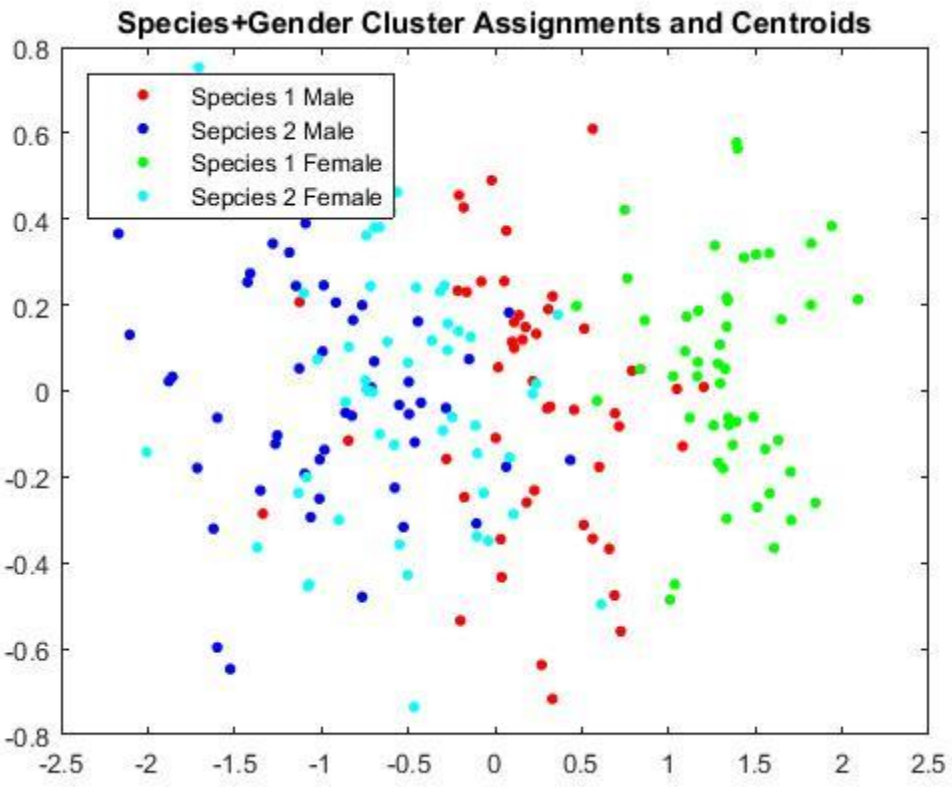
Confusion Matrix = [87 4

(Gender) 13 96]

We initially cluster the data on individual species or gender classes. Since species and gender are intertwined we combine the results to plot the data clusters label them for both male and female species 1 and 2.







From the accuracy results, it is observed the PCA same subspace with SOM clustering gives us much better results than K-means clustering. So SOM can be considered better for K-means clustering.

Comparison with MLP: When we compare this with MLP for either of the two classes either gender or species class. We have an accuracy of 100 % for both train and test dataset while classifying for either gender or species class. We generate a confusion matrix for the test classes for both gender and species class and have the same confusion matrix and accuracy.

Confusion Matrix = 33 0 Accuracy = 100 %

0 34

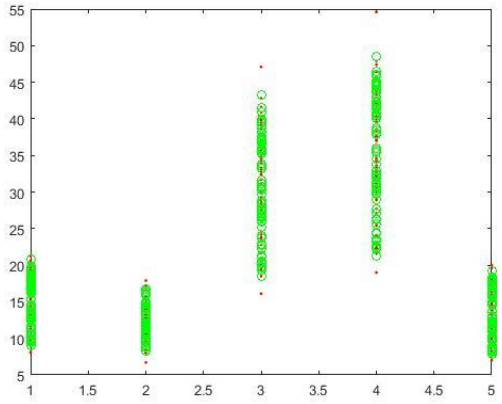
Method 3: AutoEncoder

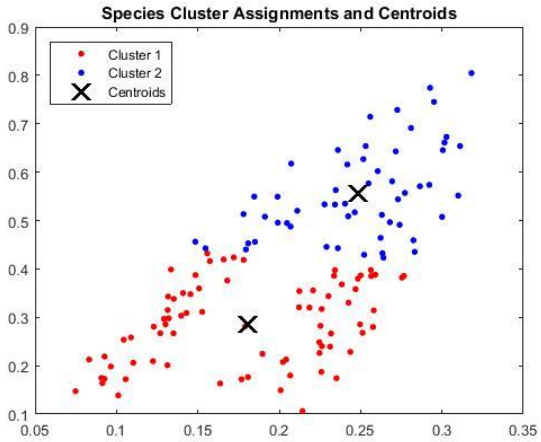
The Autoencoder is very close to MLP network with special layers. The middle layer is the bottleneck layer to project into a new subspace, while the output layer has the same number of processing elements. We are using tied weights for prediction for the output layer. The bottleneck layer is used for clustering the data for the autoencoder method. We divide the dataset into two parts the first 1/3 half is test class, while remaining 2/3 of the data is used it for training it for species classification. We use a softmax classifier using desired classes to classify the data as accurately as an MLP of 100%. The output at the bottleneck layer is used over the test dataset, to generate cluster using k-means clustering.

Confusion Matrix = 33 0 Accuracy = 100 %

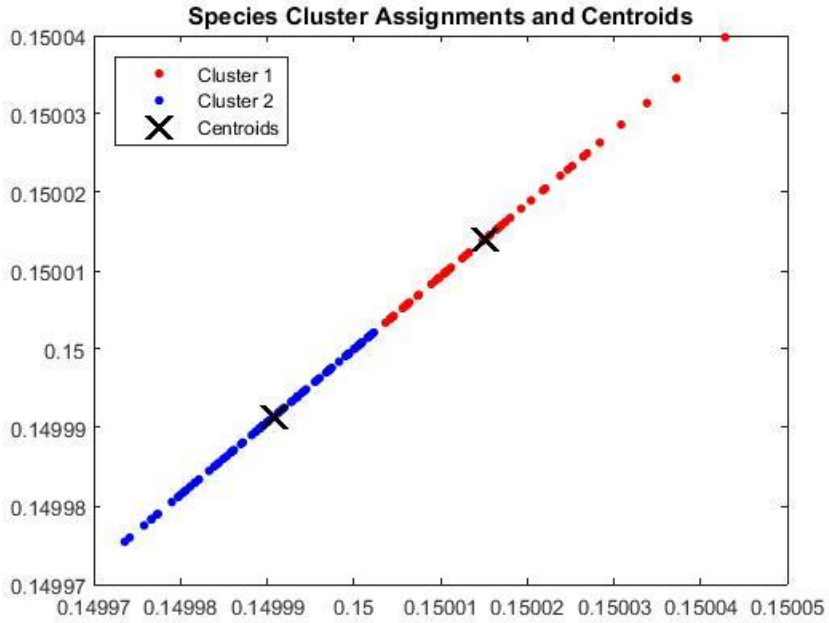
0 34

The predicted data over train data is very close to the actual data in the plot 1 below. The K-means clustering using the bottleneck layer is done is plotted in plot 2.





We also experimented with the number of hidden layers in the autoencoder network and increased it to 3. We observe a similar good performance and compared to MLP. The hidden layers have 4, 3, and 2 PEs consecutively per layer in the autoencoder with tied weights. The bottleneck layer is third layer of 2 PEs which generates not too great a cluster plot though it clearly shows a separated projected data.



Comparison with MLP with 3 hidden layer network gives similar results as both methods again achieve an accuracy of 100 % for the dataset of species class.

Conclusion: We could conclude that unsupervised learning could be efficient classifier for some datasets. PCA could be used as method very successful tool for dimensionality reduction and is used a lot in combination with neural nets achieving much better results when neural nets are used alone. SOM were didn't find to be much effective for classification, though for clustering the present enthusiastic

results. They perform better than Kmeans for clustering the projected data output of the PCA subspace. The autoencoder seems to cluster effectively with a clearly separated data, but they were found to be really effective in classification when paired with softmax classifier. They were as effective as MLP for classification, and that is why they form the basis of deep learning convolutional networks.