CS 7641: Assignment #3
Unsupervised Learning and Dimensionality Reduction

Due on Sunday, November 6, 2016

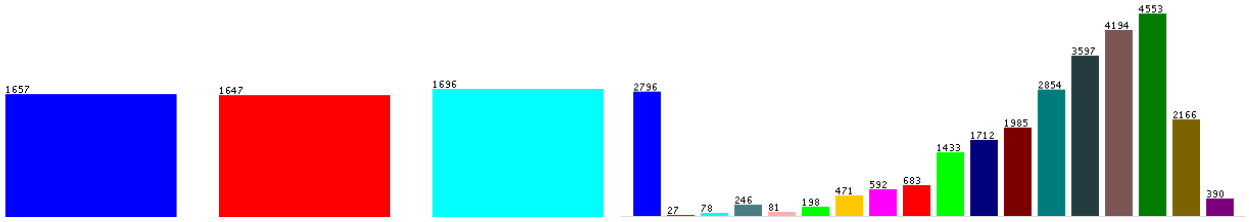
Dr. Charles Isbell

Ryan Chow

Intro

This report explores techniques in clustering and dimensionality reduction by comparing performance of two datasets. The clustering algorithms used in this paper are k-means clustering and Expectation Maximization. The dimensionality reduction algorithms used in this paper are Independence Component Analysis (ICA), Principal Component Analysis (PCA), Random Projection (RP), and Feature Agglomeration (FA). In addition, the clustering and dimensionality reduction algorithms are used in conjunction with the neural network to see how these new techniques interact with earlier work in supervised learning.

The first dataset is Waveform Database Generator and contains 5,000 instances. Each instance represents a waveform consisting 21 attributes of continuous values between 0 and 6. The goal of this dataset is to classify each waveform into 3 base wave categories. The second dataset is Chess King-Rook vs. King and contains approximately 28,000 instances. Each instance represents a chess board status consisting 6 nominal attributes (the column and row of three chess pieces). The goal of this dataset is to classify each chess board status into optimal depth-of-win for the King-Rook, ranging from zero to sixteen moves or a draw. Both datasets are available from the UCI Machine Learning Repository online. A histogram of each dataset's output variable is shown in Figure 1 below.



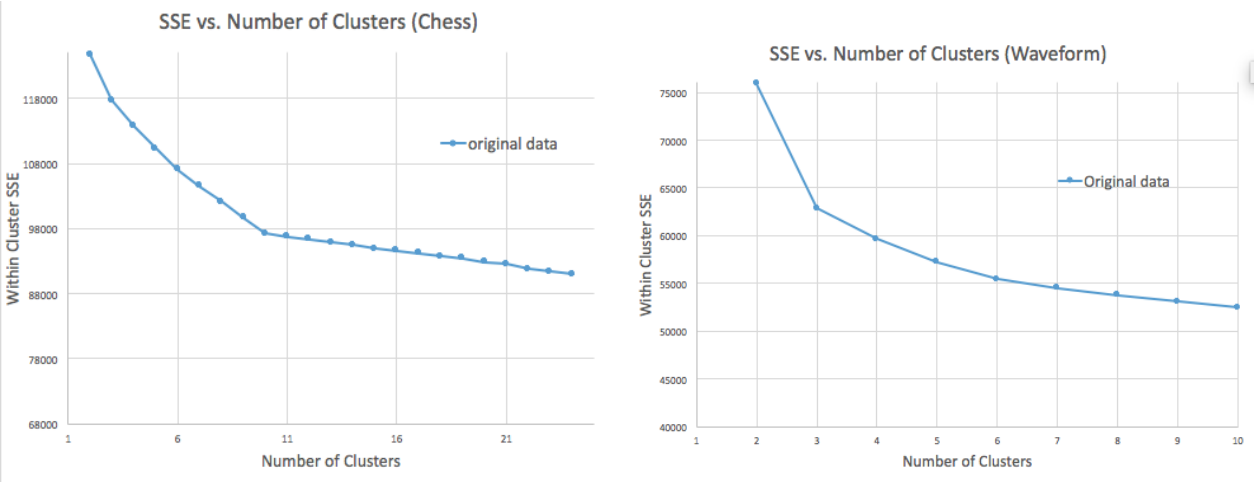
(a) Waveform dataset (5,000 instances) (b) Chess dataset (28,056 instances)

Figure 1: Classification histograms

Together, the datasets provide interesting classification problems. They are almost dichotomous in how the data are represented; the Chess dataset contains only nominal attributes, which are used to predict 18 possible classifications. The Waveform dataset contains only continuous attributes, which are used to predict just 3 possible classifications. In addition, the Chess dataset contains nearly five times as many instances. The underlying structure of these datasets illuminate many differences in performance between algorithms. A final analysis at the end of this report summarizes the lessons learned and key factors affecting the performance of algorithms for each dataset.

Clustering

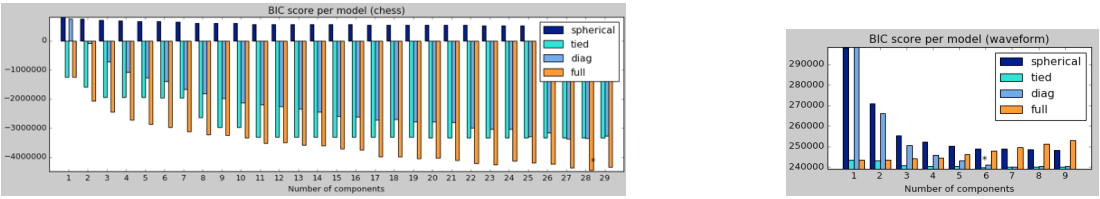
K-means clustering and Expectation Maximization were used to explore clusters with the datasets. For k-means clustering, the within-cluster sum of squared error (SSE) was used to evaluate the performance of the chosen quantity of clusters and position of cluster centers. As shown in Figure 2 below, both charts clearly illustrate that increasing the number of clusters results in lower SSE. However, the elbow method was used to examine the optimal number of clusters based on the graph. We can see that the diminishing return of SSE reduction occurs around k=10 for chess dataset and k=3 for waveform dataset.



(a) Chess dataset (b) Waveform dataset

Figure 2: k-means Clustering SSE

BIC score was used as the performance metric for Expectation Maximization. Bayesian Information Criterion is a criterion for model selection based on the likelihood function. Because more clusters added results in lower error (as seen in Figure 2 above), a penalty term exists in BIC score to penalize larger numbers of clusters. The lowest BIC scores found result in k=6 for the waveform dataset and k=28 for the chess dataset.



(a) Chess dataset (b) Waveform dataset

Figure 3: Expectation Maximization BIC score

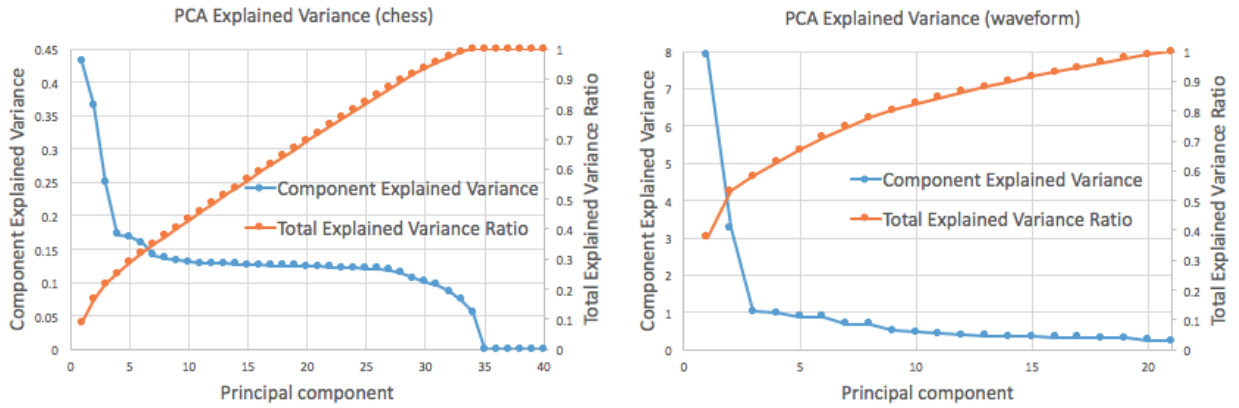
The clusters for the waveform made intuitive sense. Although the k clusters did not match the original labels for the data (which equals 3 classes), almost all of my analysis in this paper resulted in a k=6 for the waveform dataset. This could mean that the waveform dataset is actually grouped into 6 distinct groups in the space, but some groups are joined in terms of classifying them.

The chess made much less sense in terms of the number of clusters found. I think this is due to the fact that the underlying chess data is much more complex than the waveform data, and the original data had 18 classification categories. This leads to a much higher dimensional clustering, which is hard for the simple algorithms to identify.

Clustering continued on next page...

DR

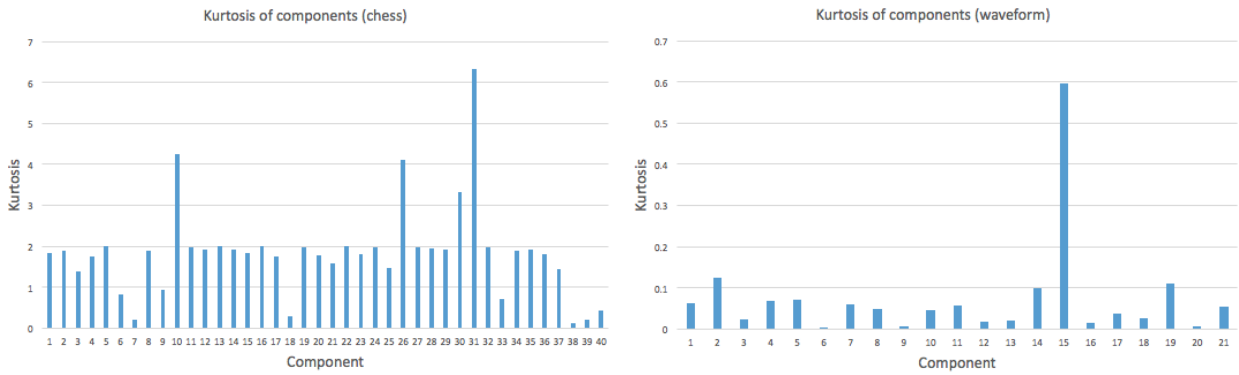
PCA: The distribution of eigenvalues for PCA in Figure 4 below demonstrates the explained variance of each component very clearly. The majority of variance in the dataset can be explained by approximately 10 components for the chess dataset and only 3 components for the waveform dataset. The elbow method was used to identify the number of components at which there is a diminishing return on explained variance by adding more components. Thus, the top few components result in a large affect on variance of the data, as expected from the PCA results.



(a) Chess dataset (b) Waveform dataset

Figure 4: PCA Explained Variance

ICA: This algorithm seemed to capture important information about the non-gaussianity of the components in each dataset. As shown below in Figure 5, three components in the Chess dataset have significantly non-normal distribution behavior, and one component in the Waveform dataset has significant non-normal distribution behavior. The average kurtosis of the chess dataset seems higher than the average kurtosis of the waveform dataset, but the waveform dataset has a single components that has a magnitude higher kurtosis. This means that the components of the chess dataset, on average, have higher non-gaussianity and the waveform components may be more likely to come from a single source. Note that a perfectly normal distribution is represented in my charts below with kurtosis=0 instead of kurtosis=3.

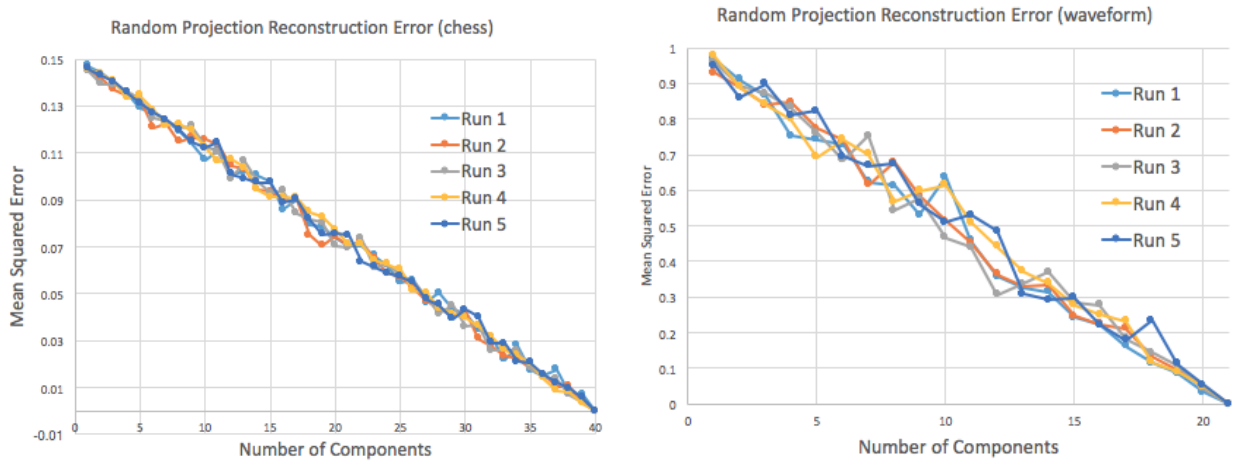


(a) Chess dataset (b) Waveform dataset

Figure 5: ICA Kurtosis

DR continued on next page...

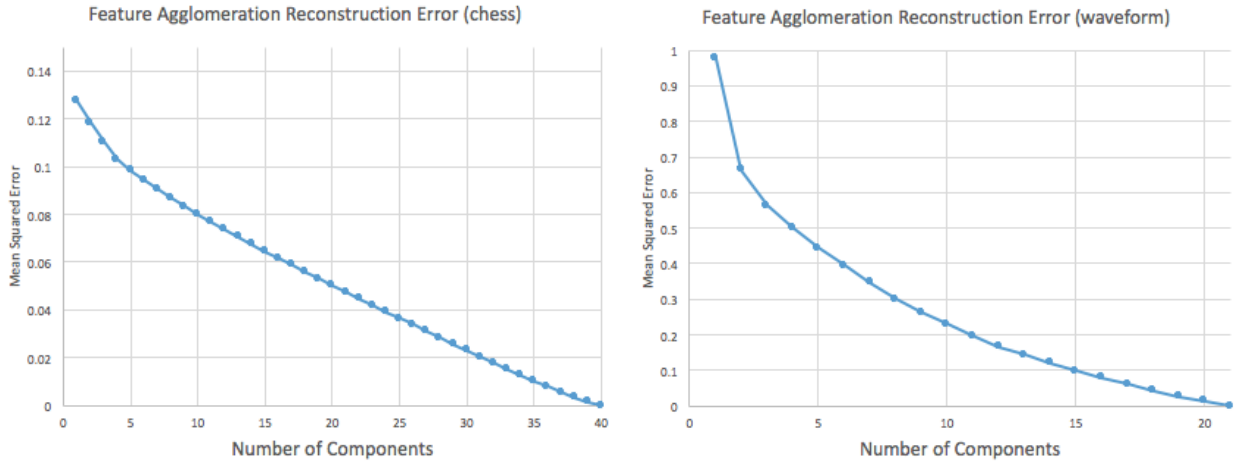
Random Projection: The mean squared error between the original dataset and reconstructed dataset after random projection was used to evaluate performance. As shown below in Figure 6, as more components are added, the MSE decreases until there is zero error between the reconstructed and original datasets when the reconstructed dataset retains all elements from the original dataset. There was no significant variation between runs, as shown by each color in the plots below. However, it is interesting to note that adding features does not always reduce MSE. In addition, the underlying feature vectors are complex and some combinations of components represent the data much better than other combinations of the same number of components.



(a) Chess dataset (b) Waveform dataset

Figure 6: Random Projection MSE

Feature Agglomeration: My fourth dimensionality reduction technique of choice was Feature Agglomeration, a method that tries to merge together similar features by clustering in the feature direction. It was very interesting to see the result of FA compared to RP. The MSE curve for FA was seemed to be much more smooth and consistently decreasing, as opposed to RP which can decrease seemingly erratically.



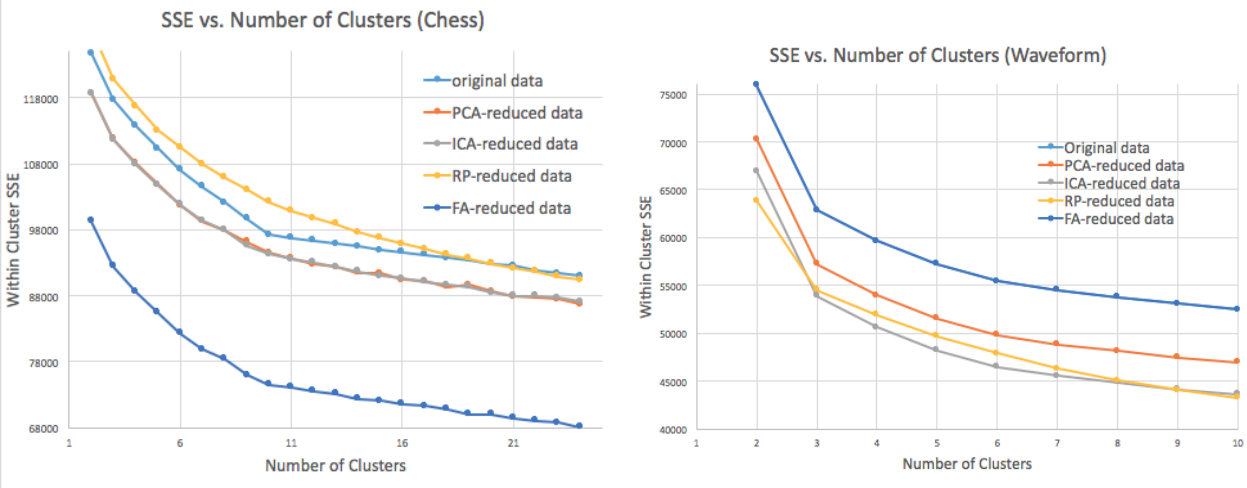
(a) Chess dataset (b) Waveform dataset

Figure 7: Feature Agglomeration MSE

DR continued on next page...

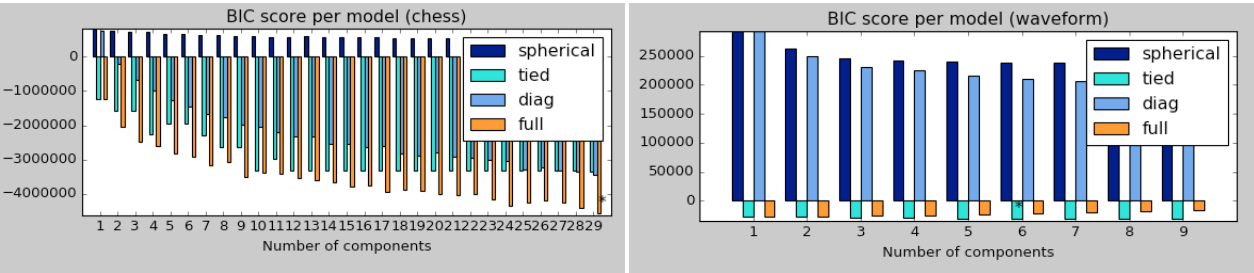
Cluster + DR

Clustering was performed with dimensionality reduction using all the previous methods. The results were interesting. Even after dimensionality reduction, the clustering algorithms resulted in the same number of clusters as on the original data. These clusters are k=3 for the waveform dataset and k=10 for the chess dataset. This means that the dimensionality reduction techniques were successful in retaining the important information in the data that is relevant to clustering the data into distinct groups. I was surprised to see the consistency of curve behavior regardless of the DR algorithm, however, it is clear that some DR techniques result in lower overall sum of squared error. This could be due to the fact the some of the DR techniques take advantage of the data structure.



(a) Chess dataset (b) Waveform dataset

Figure 8: k-means Clustering SSE with Dimensionality Reduction



(a) Chess dataset (b) Waveform dataset

Figure 9: Expectation Maximization BIC Score with PCA

Cluster + DR continued on next page...

NN + DR

It is interesting to note that neural network performance using the clusters as features resulted in performance that was nearly equivalent (or even better than) on the original dataset as calculated in previous assignments. However, ICA did not perform nearly as well. This could be due to the fact that the waveform dataset is not in fact from separate sources, as ICA assumes.

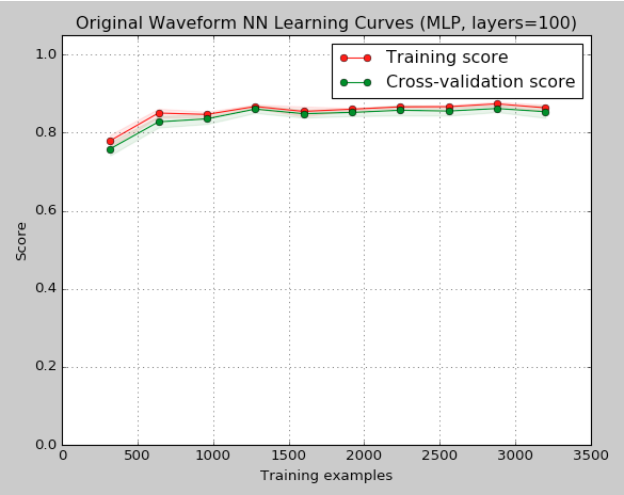
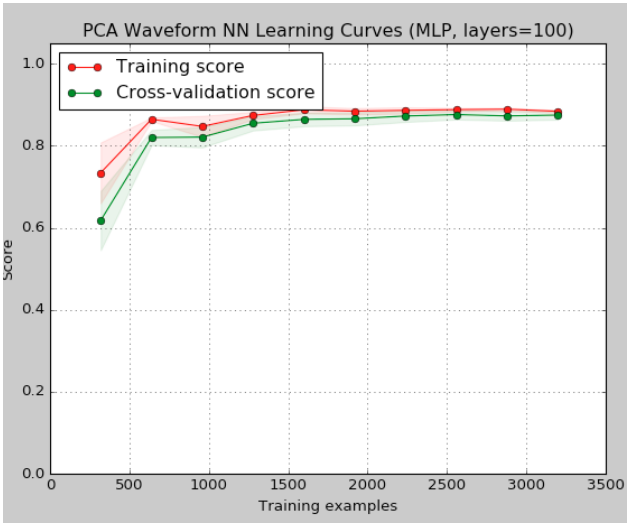
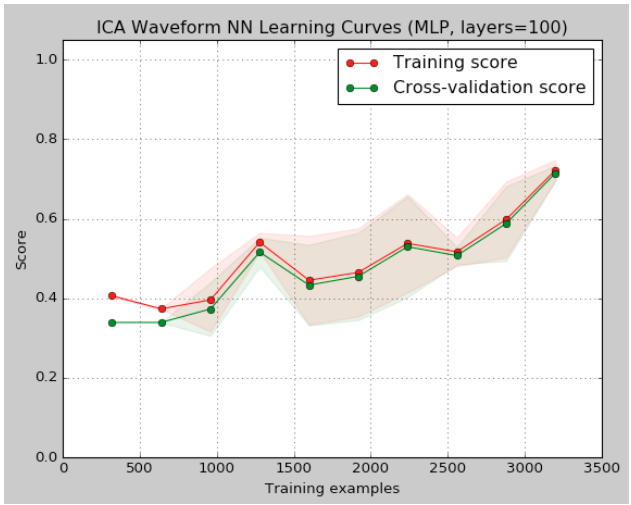


Figure 10: Original Neural Network Performance



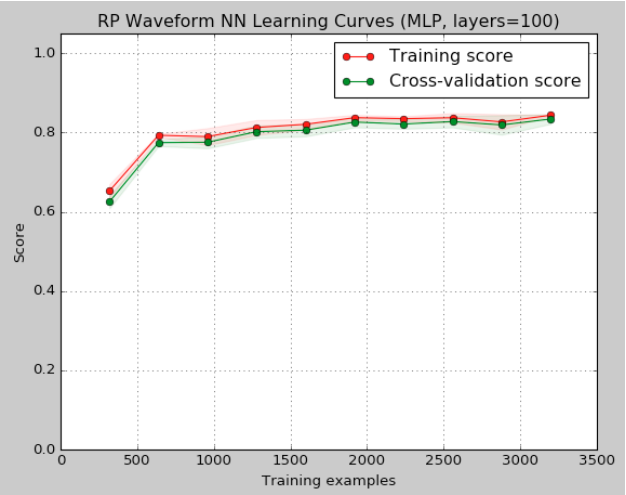
(a) Independent Component Analysis



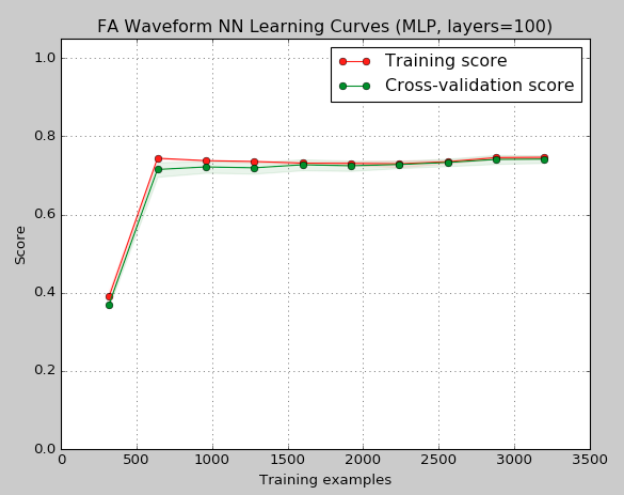
(b) Principal Component Analysis

Figure 11: Waveform Neural Network with ICA and PCA

NN + DR continued on next page...



(a) Random Projection

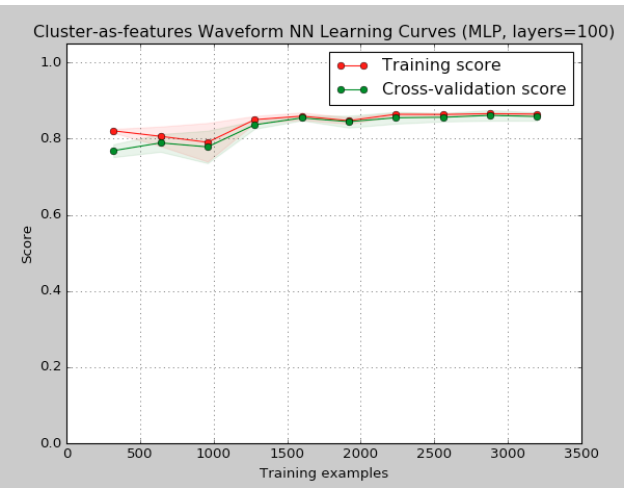


(b) Feature Agglomeration

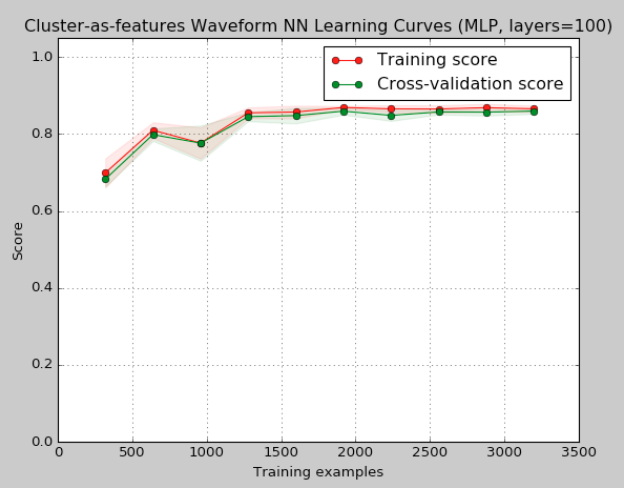
Figure 12: Waveform Neural Network with RP and FA

NN + Cluster

Clustering with neural networks performed nearly as well as on the original dataset. Treating the clusters as new features, the newly re-run neural network achieved the same accuracy as the original dataset. This was not surprising, given that the previous experiments showed how dimensionality reduction can be performed to retain almost all the important information that distinguishes clusters, and clustering can be performed to identify separate groups within the data. However, the neural networks seemed to train much faster on average compared to the original dataset. This may be because the additional clustering information provided an accurate heuristic and valuable information for the neural network to utilize.



(a) k-means



(b) Expectation Maximization

Figure 13: Neural Network with Clusters as Features

Final Thoughts: Dimensionality reduction and clustering are techniques that are highly valuable when dealing with data that has high dimensionality. Dimensionality reduction can be performed to retain an adjustable percentage of useful information, trading off data size for accuracy. This is especially helpful in cases such as the curse of dimensionality where machine learning is performed on datasets with only a few samples. In addition, dimensionality reduction techniques are extremely useful in extracting the important information from a dataset and reducing the feature space so that training is more manageable. I can definitely see cases where dimensionality reduction is not only faster, but necessary in order to achieve realistic training times for certain learning algorithms.