Andrey Lukin

CS 4641 Machine Learning

Project 3: Unsupervised Learning and Dimensionality Reduction

Professor Charles Isbell

Project 3: Unsupervised Learning and Dimensionality Reduction

Introduction

Unsupervised Learning

Unsupervised Learning is the area of Machine Learning that focuses on the ability to "learn without a teacher". For the purpose of this assignment, we will be focusing on the idea of clustering: when data is grouped together based on the similarity of its attributes. We will be discussing two different clustering algorithms, k-means clustering and Expectation Maximization.

Dimensionality Reduction

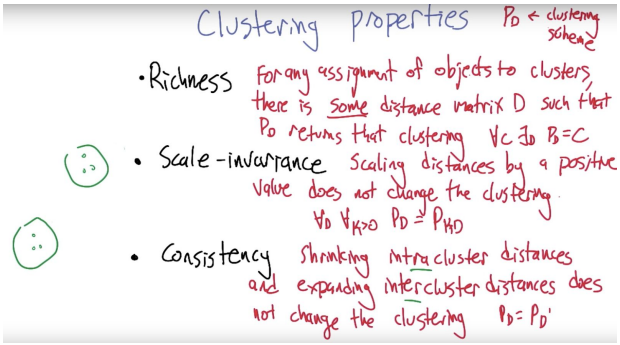
The curse of dimensionality makes it so that as we use more attributes in the data, the amount of data necessary to train a theoretical model that would accurately predict something increases exponentially. Especially, in the space of Unsupervised Learning, where we don't know the attributes necessary, and what we are trying to predict, the more attributes we have the better we will be able to generalize the data. Dimensionality Reduction algorithms like PCA, ICA, Randomized Projections, and Chi-squared feature selection allow us to generalize multiple attributes into one. With this project, we will be testing how Dimensionality Reduction aids the Unsupervised Learning process, and whether or not there is an increase or loss of performance.

Data Sets Used

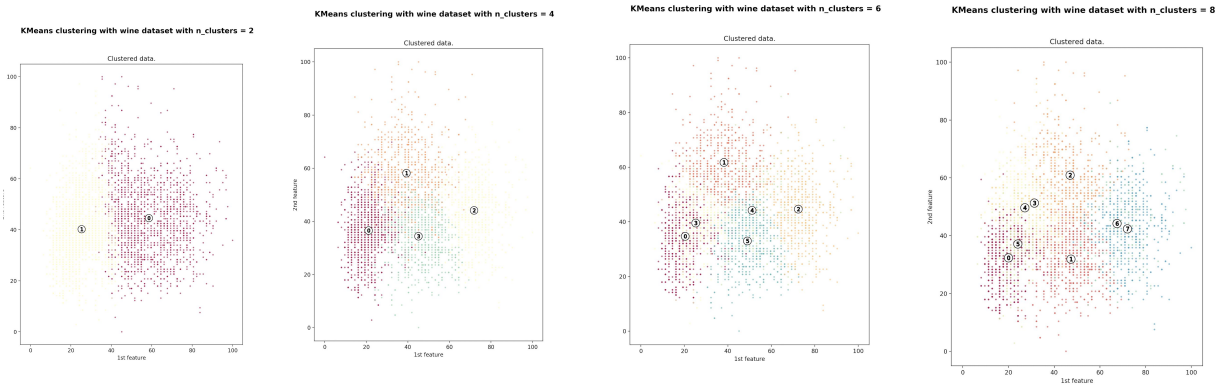
For the purpose of this assignment, I used two datasets:

1. The first one was the Titanic dataset which contains a description of most of the passengers in the Titanic, including their class, sex, ticket fare, the location of embarkment, whether they are sibling/spouse, whether they are parent/child of a person, and this was used to predict whether or not they survived in the Titanic catastrophe.
2. The second dataset that I used was the wine quality data set that provides the description of the wine's acidity, sugar contents, chlorides, sulfur dioxide, density, pH, alcohol content, and finality the quality. This is personally interesting to me because I grew up in a family of wine lovers.

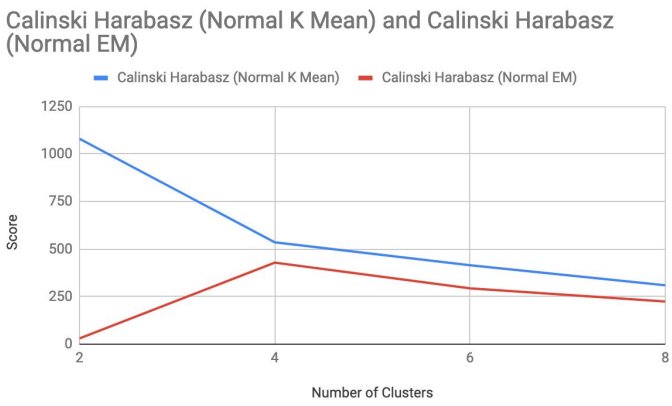
I used normal Euclidean distance for my distance metric because of the three main concepts of clustering, which states that scale invariance shouldn't affect the clusters. Euclidean was the most simple and hence was the best course of action in the beginning.



Data Set 1: Wine Quality

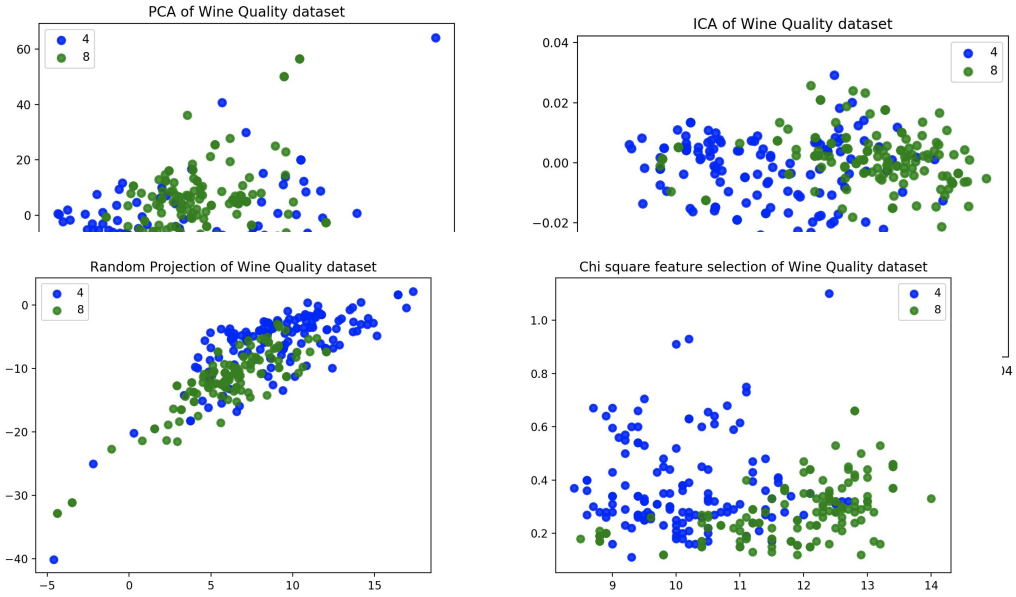


I ran the clustering algorithm with 4 different k's, [2, 4, 6, 8]. The reason for these numbers were that I initially started running 2 up to a high number N and saw that clusters are becoming subclusters after 4 clusters. As seen on this graph, the 6 cluster graph already has clusters being broken up , like cluster "0" in the n_clusters = 4 which is broken up into "0" and "3", even though the centers of those clusters are really close together, and there is not an increase in generalization. The number does not back me up unfortunately: the NMI score is super low (no more than 0.1, and that's at 2 clusters), and the silhouette score is also only about 0.24 at 2 clusters. Another good metric to look at is the Calinski Harabasz score, where the higher the score the more dense each cluster is, and the more spread out the clusters are from each other. This is the score over the number of clusters:



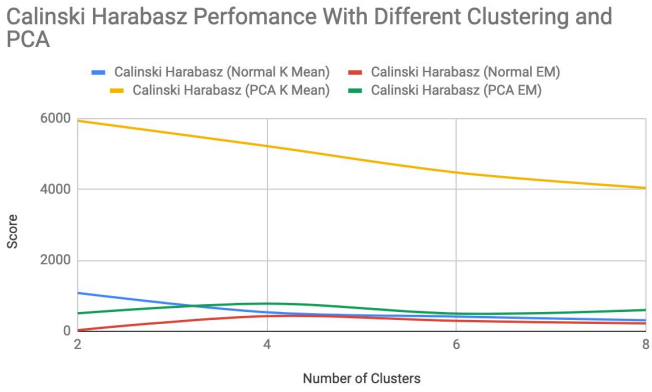
Here we see that the score is best with two clusters. This is very confusing, because in the data that we are using, there are 8 different 'quality' possibilities, but only

6,7,8, and 9 are used. The reason why it might be two now is because we are using way too many attributes. We also see here that K Mean completely outperforms EM clustering, and so I will not be adding graphs to that to save space.



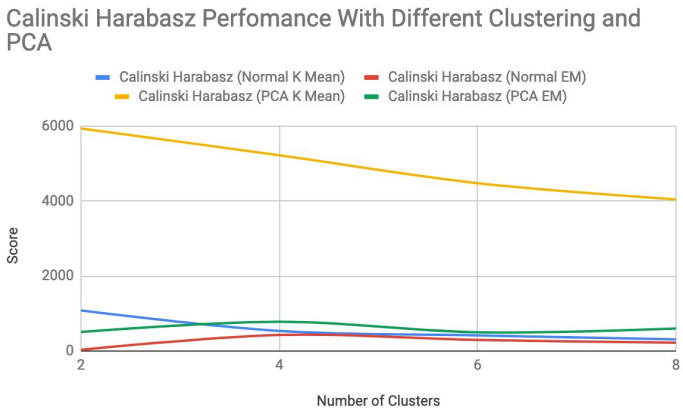
This is the PCA, ICA, Random Projection and Chi^2 Feature selection ran on the wine quality dataset.

As seen here, based on the Calinski Harabasz scores, PCA definitely creates better clusters.



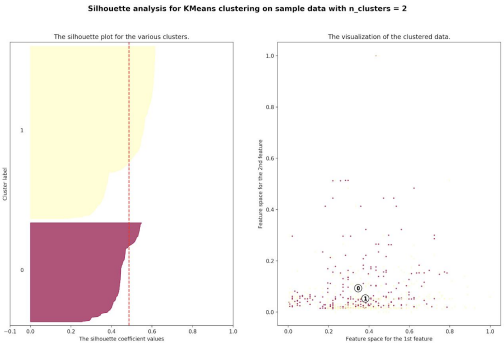
Data Set 1: Titanic

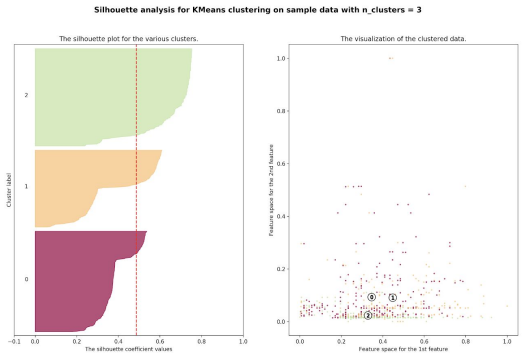
For this data set, let's start off with the Calinski Harabasz score right away. We will run the PCA on both the K Mean and EM clustering methods, and based on the score see how many clusters is a good amount to look more in depth into. Based on the graph here, again the PCA algorithm definitely improves the clustering algorithms. We also see that normal K mean does the best with a cluster size of 2 and Normal EM does best with a cluster size between 4-6.

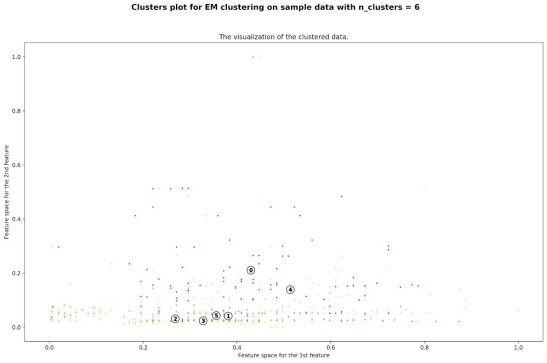


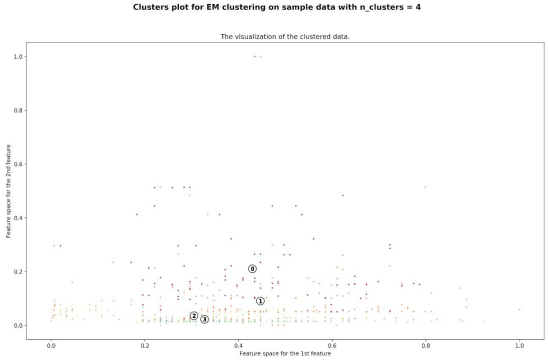
Let's look at the graphs.

One important thing to think about is the context of this data. We are trying to predict whether or not the passenger was able to survive based on the attributes. This is the reason why two clusters is the best amount of K Mean Clustering; you either survive the crash or you don't, there is no in between.





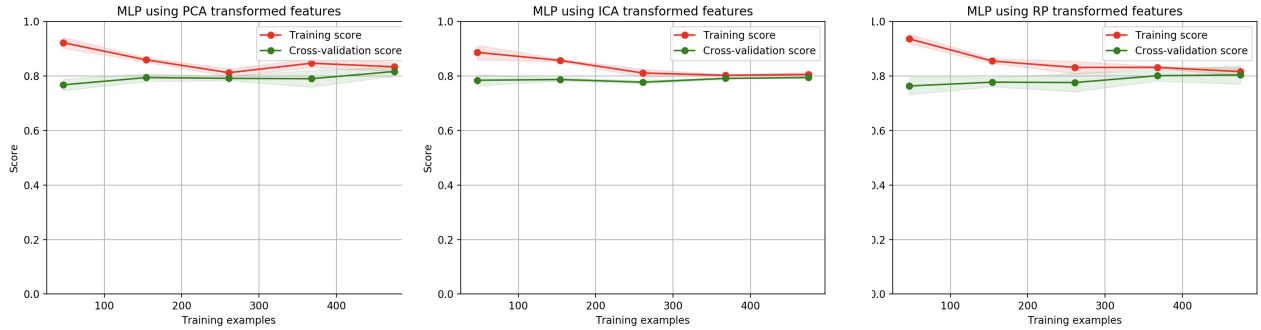




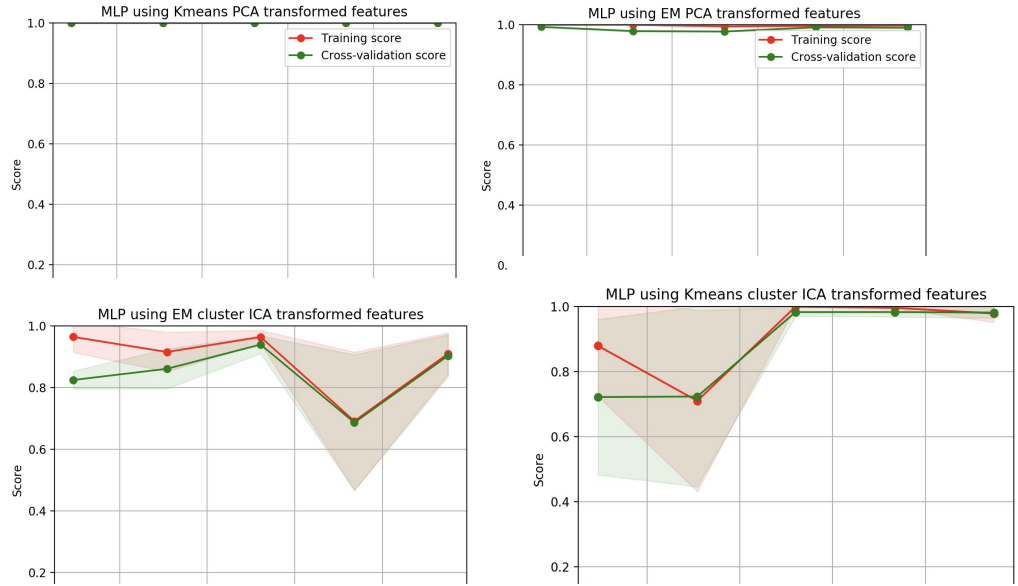
But why does Normal EM perform best on k being between 4 and 6? One possible reason is because EM uses probability whereas K Mean clustering using Euclidean distance.

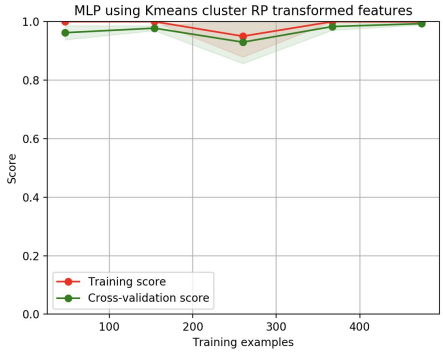
Neural Networks

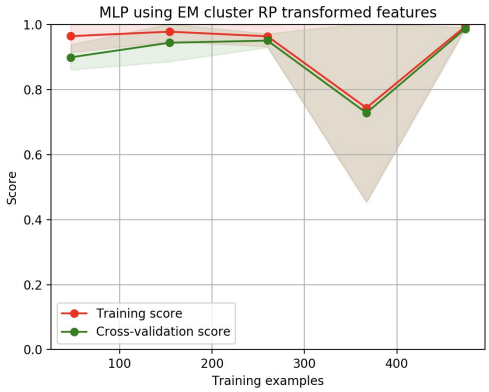
First, let's look at the feature selection using the different algorithms like PCA, ICA and RA, and feed it into the neural net. This is what we get:



There isn't much of a difference compared to the training and cross-validation score that was in the project 1. This might be because the answer is binary, and predicting takes only two clusters. The labels needed are also easy to predict, and with a low amount of data, not much is changed. Now what happens if cluster after we have done dimensionality reduction, and then used those as features in the neural networks?







With the clusters becoming a lot more distinct from each other, the neural network is able to a lot easier be able to categorize it. With that, the cross validation score jumps to almost perfect, or perfect in some cases. We see that in some situations ICA take longer than PCA to train, and the reason behind this is the fact the ICA actually runs PCA in the background before running PCA.

Conclusion

With Dimension Reducibility, we can try and work on the curse of dimensionality. The number of potential attributes for any analyzable situation is always super high, and it's always about the amount of data that you are able to collect, but now we are able to generalize the data in a better way. We have demonstrated the pros of this reducibility with the Neural Networks performing better in this project than they did in project 1.