CS7641 Machine Learning Assignment III by Georgia Tech: Unsupervised Learning

T. Ruzmetov

October 29, 2017

Assignment Description

In this assignment we are asked to apply and explore the role of two unsupervised learning algorithms such as K-means Clustering and Expectation Maximization on two data sets. Along with that, we are to investigate how dimensional reduction techniques such as PCA, ICA, RP, FA help speeding up performance by educing the dimension of feature space. In addition, we are expected to apply the dimensional reduction algorithms to one of the data sets(done already) from assignment 1 and rerun neural network learner on the newly projected data. Then by applying the clustering algorithms to the same data set to which we just applied the dimensional reduction algorithms, treating the clusters as if they were new features. In other words, we need to treat the clustering algorithms as if they were dimensionality reduction(or feature extraction) algorithms. Once more, we need to rerun neural network learner on the newly projected data. (Last sentence was really confusing to many!)

Data Sets

Lending Club Data Lending Club is the world's largest online marketplace connecting borrowers and investors. Getting loan has become common practice in developed countries, which makes it interesting to learn how banks determine customer eligibility or credit worthiness given some information. In this problem, we apply unsupervised learning methods to predict if customer is going to pay or default on their loan based on provided set of features. After cleaning and feature extraction, the data set contains 15 features and 285373 samples. 90% of entire data is allocated to training set and 10% is separated out for testing. Then only only 30% of the training set is used for model training due to large size resulting in high time consumption. "One-hot_encoding" is applied to all categorical features which increased n features from 15 to 28.

Adult Data The adult data set contains census information from 1994. Our task is to predict whether a person makes more than $50K/year. After preprocessing is applied, there are 11 features and 45000 remaining instances. "One-hot_encoding" is applied to all categorical features which increased n features from 11 to 65. The target feature "income" is labeled as "1" when income is greater than $50K and "0" when it is less.

K-means Clustering

In K-means algorithm we randomly initialize K centroids in a given space (2D, 3D, ..), where N-features define dimension. Then for each data point we calculate distances to all centroids and choose the closest centroid to assign a label. After labeling is complete, we move each centroid to the geometrical center of corresponding class(label). We proceed by reiterating until locations of centroids converge to some position.

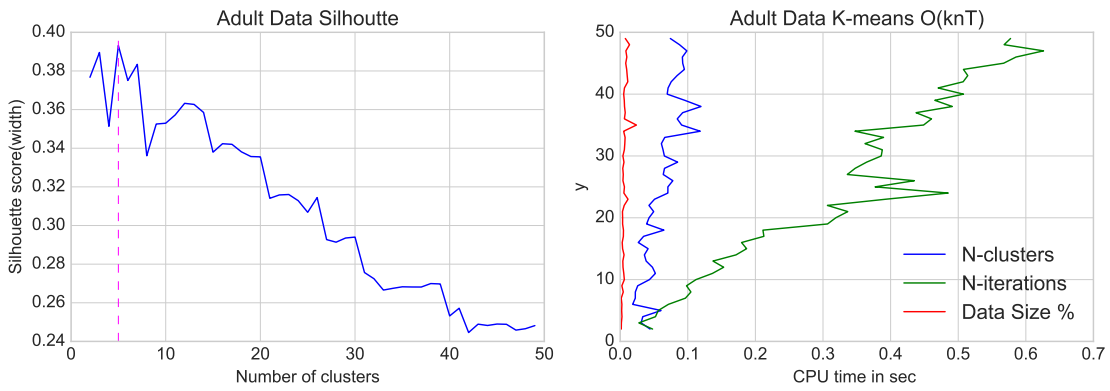

Figure 1:

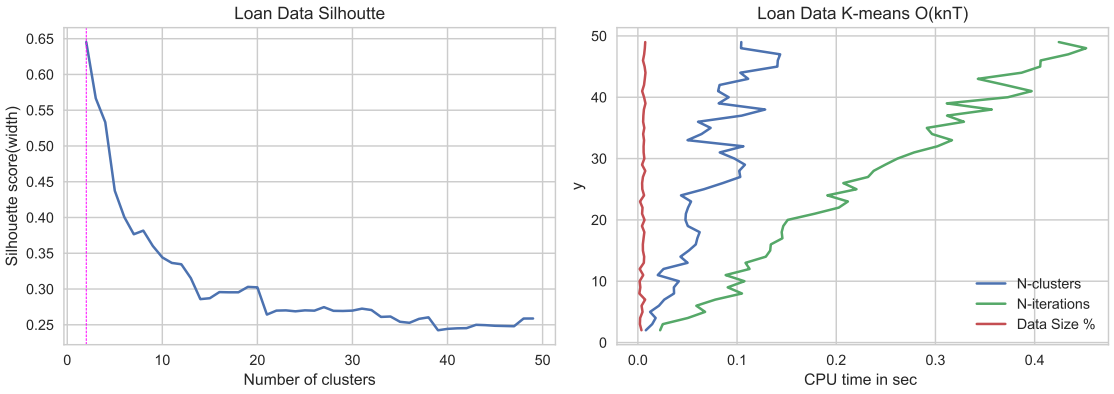

Figure 2:

In practice, the k-means algorithm is one of the fastest clustering algorithms available, but it is prone to fall into local minima. That's why it can be useful to restart it several times. I used k-means++ algorithm, which offers a smart procedure to initialize the cluster centers that guarantees to find a solution that is O(log k).

For both of my data sets target class in known, which is actually helpful to check the prediction accuracy. I separated them out at the beginning from the data frame. Although target classes for my data sets are binary, it is intriguing to check how many clusters k-means can find. So, I used Silhouette Algorithm to evaluate number of clusters for both data sets with SciKitLearn under Python. The Silhouette value is a measure of how similar an object is to its own cluster compared to other clusters. By calculating silhouette score for different cluster sizes one can choose n-clusters with the highest silhouette score. For Adult Data (Fig.1 & Fig.2 left plots highest score is achieved at cluster size 5, and exactly 2 cluster are found with Lending Club data. Since evaluation uses euclidean distance metric, categorical feature dominated data should be less accurate compared to data with mostly numerical features.

Model complexity for k-means algorithm is as described in SciKitLearn O(knT), where k is number of clusters, n sample size and T is number of iterations. As shown in Fig.1 and Fig.2 model complexity analysis

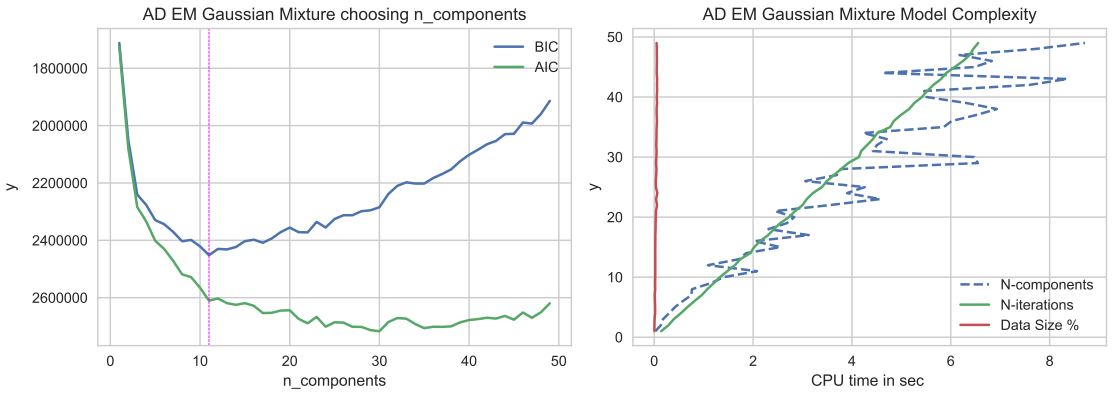

Figure 3:

results reveal linear dependence for K-means algorithm on all above mentioned parameters, but slopes are different. Iterations are most time consuming and sample size almost did not affect the performance time. Performance of K-means on Adult data set was quit astonishing with 74% accuracy on training set and 73% on testing set with n_seed = 100. Since model does not need actual label for training similar accuracy on training and test sets is not surprising. Performance on Lending Club data gave 63% accuracy on both training and test sets.

Expectation Maximization Algorithm and Gaussian Mixture Models

Expectation maximization (EM) is a numerical method for maximum likelihood estimation. EM guarantees to approach a local or global optima by increasing maximum likelihood of the data with subsequent iterations. Here we use EM with Gaussian mixture model. The algorithm basically consists of two main steps, where in a first step known as Expectation step, we calculate expectation of the component assignment(labels) for each data point given the model parameters(Gaussian parameters: means μ_k, weights ϕ_k, standard deviations σ_k). The second step is Maximization step, which consists of maximizing the expectations calculated in a first step with respect to the model parameters. This step involves updating above mentioned parameters iteratively. The entire process repeats until the algorithm converges to maximum likelihood estimate.

I used Gaussian mixture model for EM, which is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. Then naturally one would expect descent performance on data with Gaussian distributed numeric features. For evaluation of optimum number of clusters I tested BIC(Bayesian information criterion) and AIC(Akaike information criterion) methods with Gaussian Mixture Model as implemented with SciKitLearn package. For both data sets BIC score yielded smaller value for n_components than AIC score. For Adult data 11 clusters are found optimum via BIC and AIC gave nclusters=30. Wheres for Lending Club data nclusters(BIC)=2 and nclusters(AIC)=6. Interestingly this cluster size optimization methods are finding right number of clusters(same as what we have) for Lending Club data.

Model complexity analysis conducted by altering parameters such as cluster size, number of iterations, and sample size. Linear relationship is found against sample size and n-iterations, but cluster size show nonlinear behavior with time with zig zags. It will converge to linear if one does take average over multiple iterations

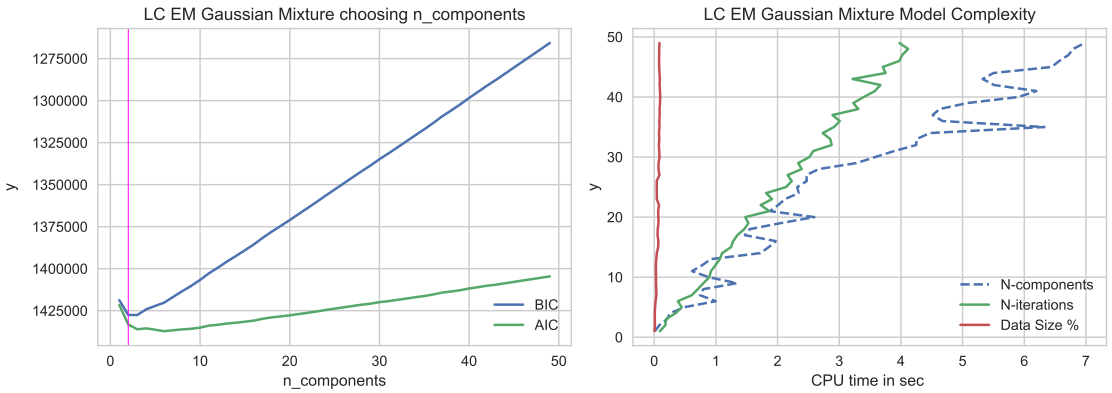

Figure 4:

per cluster size. Increasing nclusters is found most time consuming for both data sets. EM GMM gave 71% prediction accuracy on both training and test sets for Adult Data with nseed=100. For Lending Club Data prediction accuracy is 59% on training set and 53% on testing set. Although Lending Club data have mainly numeric features and normalization is applied to all features, it is showing low prediction accuracy compared to Adult Data. One can look at distribution of all features separately, they may not be normal, or maybe clusters are so much mixed that it is hard to separate. Thus we need to apply dimensionality reduction to eliminate noise or capture variance.

Dimentionality Reduction: PCA, ICA, RPA, FA

Principal Component Analysis

PCA is a dimentionality reduction technique which converts a set of possibly correlated variables(features) into principal components which happens to be directions that capture most variance in data. For instance, if I have a data set composed of features $X_1 = f(x)$, $X_2 = g(x)$ and $X_3 = a * X_1 + b * X_2$ that is linear combination of 1st and second features. Then, applying PCA to above data can transform it from 3D to 2D. It is obvious that 1st and second features are sufficient to explain all variance in the model and 3rd feature is just a linear combination of those two. So, PCA projects original dimensions into new dimension that will capture most of the variation in a data such that all those new features are orthogonal.

In order to choose optimum number of components which retains most variance in the model I plot eigenvalues per components and cumulative sum os eigenvalues vs components Fig.5(right side). One can see that dimension of Adult Data reduced to 4 and Lending Club Data reduced to 3 components preserving 99% of variance. Then k-means is applied to output of PCA with reduced dimension. Prediction accuracy on training set for Adult Data gave 74% and for Lending Club Data 63%. On the other hand prediction of EMGMM is 81% for Adult Data and 61% for Lending Club Data. It was amazing experience to see enhanced prediction accuracy after reducing the dimension of data by order of magnitude. Beside, data in reduced dimension can be easily visualized as shown in Fig.5(left figs). In these plots I plotted best 3 components as 3D scatter plot. One can notice that classes are really mixed for both data sets even after applying PCA. I noticed outliers for Lending Club Data, which could be the result of projection.

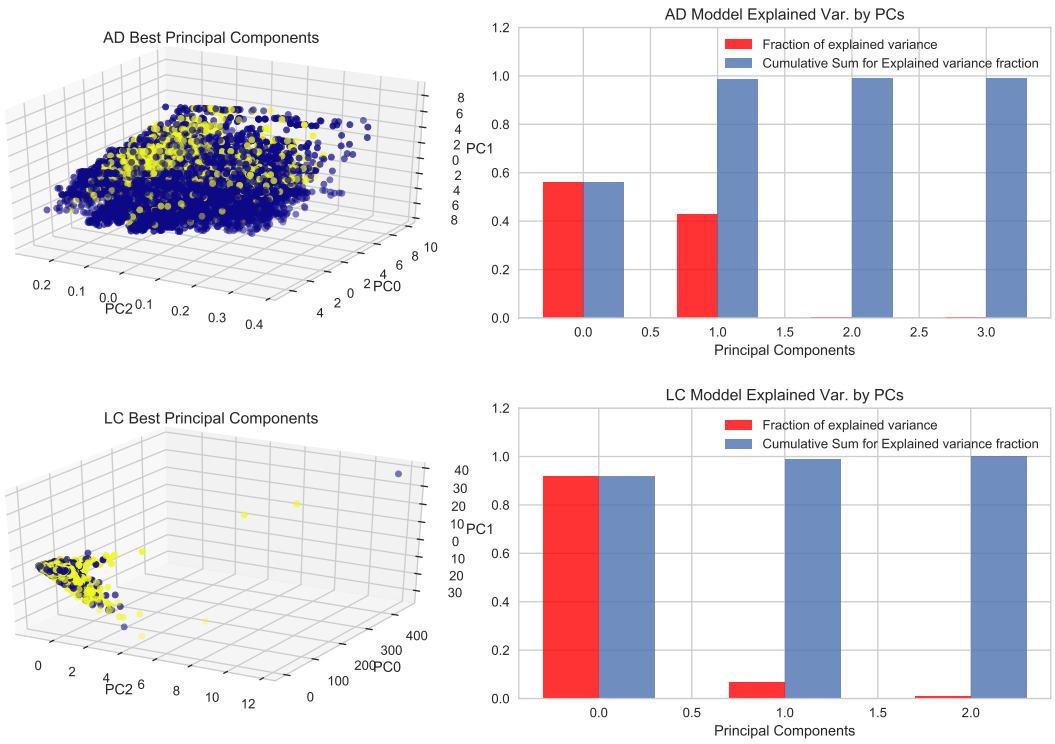

Figure 5:

Independent Component Analysis

ICA is a powerful technique to separate linearly mixed sources. It transforms original features, assuming they are linearly mixed, into independent components with zero mutual information. The latent variables are assumed nongaussian and mutually independent, and they are called the independent components of the observed data. After applying ICA to whole data number of components can be reduced by choosing components with highest kurtosis, which is the fourth central moment divided by the square of the variance. I used FastICA model implemented in SciKitLearn under Python.

By kurtosis for each component I sorted out all components wish kurtosis value greater than 1/3 of maximum kurtosis value. For Adult Data components 20,30,48,59 survived and for Lending Club Data 4,14,19 are kept for further processing Fig.6(right hand plots. On reduced by ICA, Adult Data gave 56% accuracy for K-means and 68% accuracy for EMGMM when applied to training set. Lending Club Data resulted in 56% accuracy for K-means algorithm and 61% accuracy for EMGMM on training set. In practice ICA is good at separating linearly mixed features into independent components. Since prediction accuracy on both data sets with both unsupervised learning methods is less that original prediction accuracy, one can conclude that ICA may not be the best tool for problems considered. Also, 3D scatter plots depicted, where I choose 3 best components with highest kurtosis, don't seem to have well separated clusters.

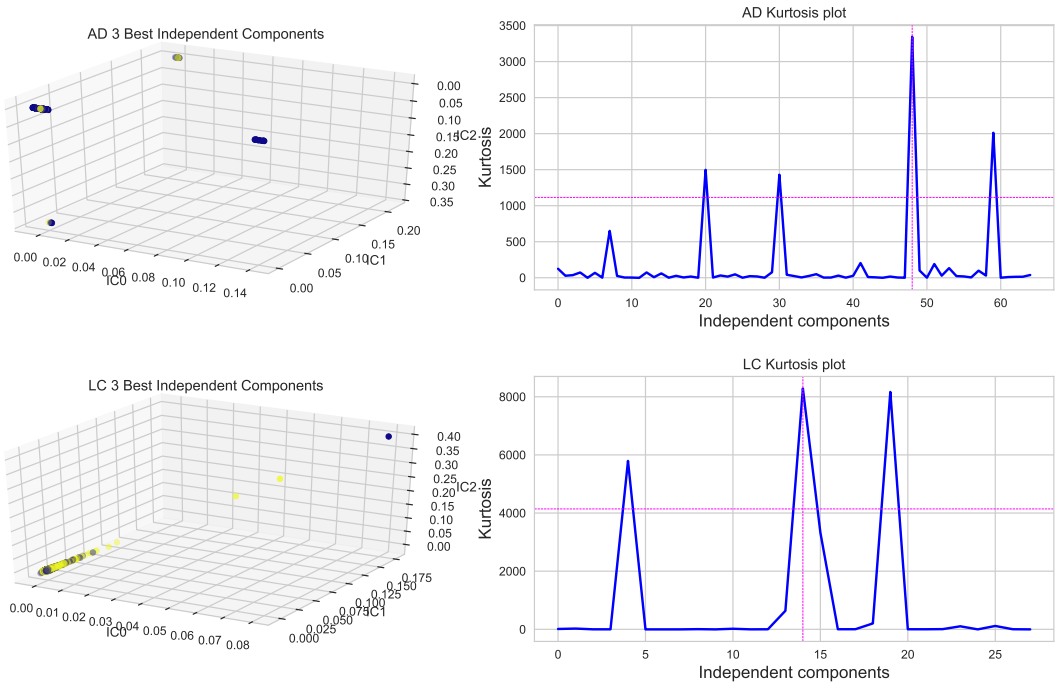

Figure 6:

Random Projection

Random projection is a technique used to reduce the dimensionality. Method is powerful, simple and good at preserving distances well after projection. In random projection, the original d-dimensional data is projected to a k-dimensional (k << d) subspace, using a random k × d dimensional matrix R whose rows have unit lengths. I used Gaussian random projection implementation in SciKitLearn, where Gaussian distributed random matrix R constructed of rows with random unit vectors orthogonal to each other. In order to pick best number of components for D-reduction, I used so called 'reconstruction loss' method suggested in piazza, where after reducing the detention matrix is reconstructed and by subtracting the norms of original and reconstructed matrices error is calculated. Lower error means that n_components was able to preserve distance information well. Unfortunately analysis I did suggest that I must keep all features in order to have lowest error Fig.7. So, this D-reduction technique was not so helpful after all.

Then I decided to keep all components (ncomponents=nfeatures) which gave similar prediction accuracy with minor 2% improvement on training set for Adult Data for both methods. In contrast training accuracy for Lending Club data reduced by 7% for K-means and no change was observed for EMGMM. How to interpret this? Mapping original data into new data via multiplying it by a matrix with Gaussian distributed unit length rows we expect to get more like Gaussian type new features. Then we expect at least EMGMM model to have similar accuracy as original prediction, which actually happened. Both data sets confirm this observation.

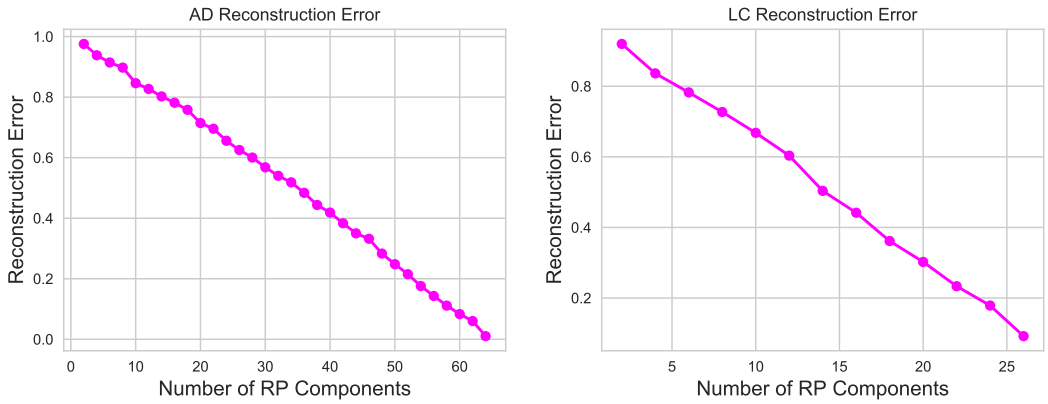

Figure 7:

Factor Analysis (FA)

Factor Analysis is probabilistic models. FA uses likelihood of new data for model selection and covariance estimation. The observations are assumed to be caused by a linear transformation of lower dimensional latent factors and added Gaussian noise. Optimum number for n_components is found via evaluating max cross validation score and picking corresponding ncomponent. Based on cross validation score optimum n-components kept for Adult Data is 22 and for Lending Club Data 15.

Both k-means and EMGMM methods gave 65% training accuracy on Lending Club Data set. For Adult Data accuracy of prediction on training set is 54% for k-means and 71% for EMGMM models. All this predictions and fittings are done using same nseed as originaly used. Once again, EM resulted in same prediction accuracy after reducing the dimension for both data sets, while k-means is suffering by showing lower accuracy. Finally, I performed model complexity analysis for all above dimensionality reduction techniques by altering ncomponents and sample size Fig.9. As shown, FA method was most time consuming as expected since it uses EM internally to transformation of the latent variables to the observed ones. Its ncomponent dependance is linear on time, but stil has the lowest slope meaning the most time consuming among all other DR methods. FA showed dramatic increase in CPU time with sample size. PCA and RP was constant with changes in ncomponents or sample size. Deviation from linearity is observed at large ncomponents or data size for ICA method.

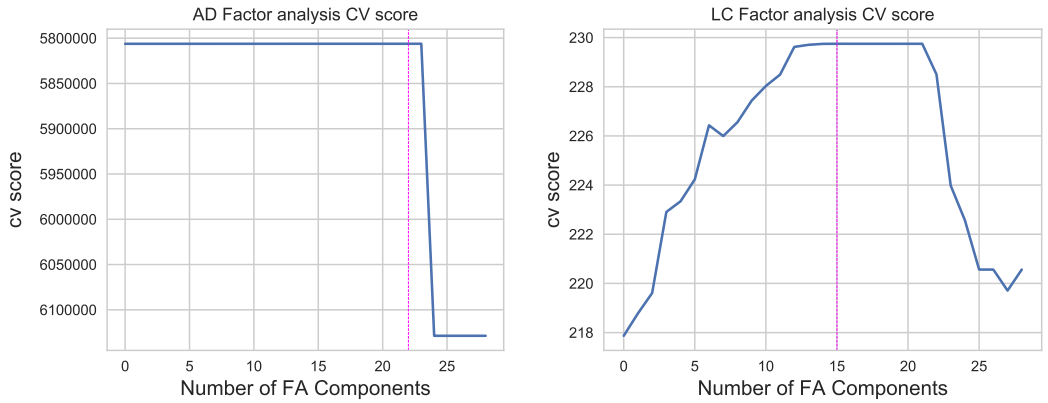

Figure 8:

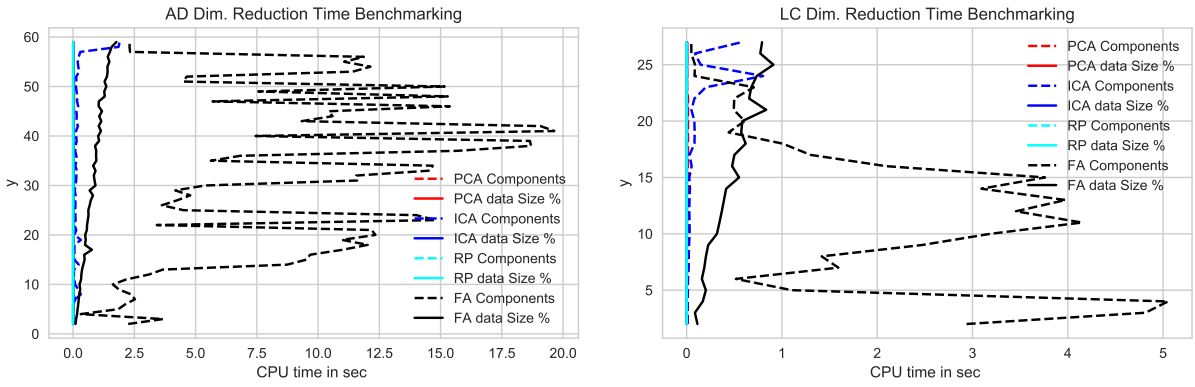

Figure 9:

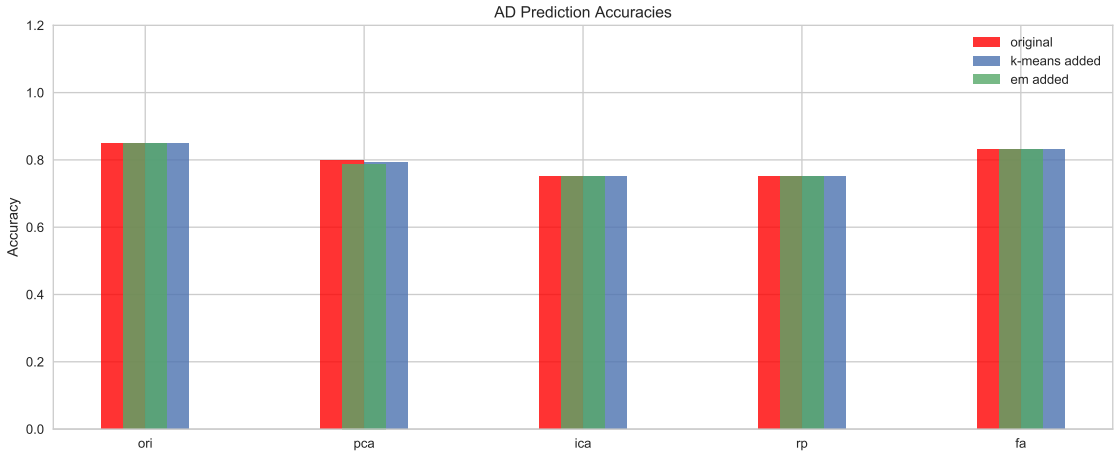

Figure 10:

Neural Network Prediction on all Original Projected and Engineared Data

In this section I tested supervisied neural network model on all differently modified versions of Adult Data: Original + Projected via 4 Dimensionality reduction methods(red), all above with k-means predicted labels added as new feature(blue), all above + EM predicted labels added as new feature(green) Fig.10. For neural network training I used MLP(multi layer perceptron) function built into scikitlearn. Hyperparameters are set similar to once in 1st assignment.

NN on original data resulted in best performance with 84.8% training accuracy. From projected data sets output of FA method scored as best with 83% accuracy. Adding labels generated by k_means and EM as new feature did affect prediction accuracy for any model, except for pca, where original data showed better pridinction.

Conclusion

This assignment was exhausting! But I learned a lot of practical as well as theoretical ML methods and concepts. K-means algorithm is very fast and robust in predicting unknown labels. EM is somehow slower than k-means but it was little more accurate in label prediction. Dimensionality reduction techniques can be really handy with high dimentional data of course if projected data can preserve most information(explained variance in the model, distance information).