Assignment: 3
CS7641 - Machine Learning
Bishwash Aryal
March 24, 2019

Abstract: This paper explores different clustering algorithms and dimensionality reduction algorithms as well as applies them as pre-processing steps to train neural networks.

1. Introduction

Clustering algorithms is one of the supervised learning techniques, where data points of unlabelled dataset are grouped into clusters based on similarity. In high-dimensional space similarity is defined using a distance measure such as Euclidean, Cosine, Jaccard etc. This paper focuses on two of the most commonly use clustering algorithms:

- K means
- Expectation Maximization (EM)

Feature selection is to find a subset of features or attributes using different strategies such as filter, wrapper and embedded strategy. When a dataset has many features, feature selection is usually performed to avoid the effects of the curse of dimensionality. While feature selection simply selects or removes attributes dimensionality reduction is about linear transformation of data into low-dimensional space in which it still preserves the relevant or useful original information but does not suffer from curse of dimensionality. We will analyze following dimensionality reduction algorithms:

- Principal Components Analysis
- Independent Components Analysis
- Randomized Projections
- Factor Analysis

We will first start by running clustering algorithms in following datasets found in UCI machine learning repository:

- German Credit Data
- Wine Quality Data Set (White Wine)

Next, dimensionality reduction algorithms will be applied to the both the datasets and we will reproduce clustering experiments after dimensionality reduction as well. Finally, we will apply dimensionality reduction in the german credit data and rerun the neural network learner on the newly projected data. We will also experiment with running clustering algorithms as pre-processing steps to train neural networks.

2. Datasets

German Credit Data (credit-g) is a binary classification dataset with which classifies people described by 24 attributes as good or bad credit risks. It has 1000 instances. Numerical version of the dataset will be used. The dataset is unbalanced with 70% of samples belonging to the “good” class.

Figure 1: German Credit Dataset Class Distribution

German Credit Dataset is interesting because it has higher number of attributes and not all those attributes might be equally important, this will let us experiment with dimensionality reduction. It is also a unbalanced dataset so it would be interesting to analyze how it would affect the clustering or dimensionality reduction algorithms. In assignment #1 using Neural Network only 70% accuracy using f1_score was achieved for this dataset. We will get to evaluate if we can achieve higher accuracy by using clustering and dimensionality reduction for pre-processing.

Wine Quality Data (wine_quality) is a multivariate classification dataset which has 12 features with continuous values and 7 classes in total. There are 4898 instances available.

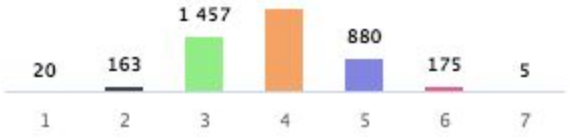

Figure 2: Wine Quality (White) Dataset Class Distribution

This dataset is interesting because it has 7 different classes so we are not looking at just a binary classification problem, it could help analyze the clustering algorithm with high number of clusters rather than just two of them, also this has higher number of instances than the previous dataset which gives us more samples to work with. Using Neural networks without dimensionality reduction 50% accuracy was achieved in this dataset.

3. Clustering

Clustering is a method of grouping data points based on similarity. Analysis of K-Means clustering and Expectation Maximization was done using sklearn.cluster.KMeans and sklearn.mixture.GaussianMixture from Python library scikit-learn.

The Datasets used for analysis contains features with highly varying magnitude, since the algorithms are going to use Euclidean distance between two data points which is sensitive to magnitudes. To give equal importance to all features scikit-learn MinMaxScalar is used so that magnitudes for all feature values are scaled between 0 and 1.

3.1 K-Means Clustering

When number of clusters K and the dataset is provided K means algorithm initially starts with estimates for k centroids either randomly generated or selected from the data set. The data points are then assigned to the nearest centroid based on squared Euclidean distance. Once all data points are assigned, the Centroid is recalculated taking the mean of all data points assigned to that centroid's cluster. The steps for data point assignment and centroid update iterates until the centroid position does not change any more or within-cluster sum-of-squares is minimized or limit of maximum number of iteration is reached.

How many clusters?

To begin clustering we must know the value for k, which is unknown in an unlabelled dataset. To choose a value for k, a plot for within cluster sum of squared distances for a given k is generated where as number of k increases to a point where each sample forms its own cluster the sum of squared distance will be zero. The plot will look like an arm and the optimal k value is the elbow on the arm, which indicates a point where increasing number of clusters is not going to help anymore.

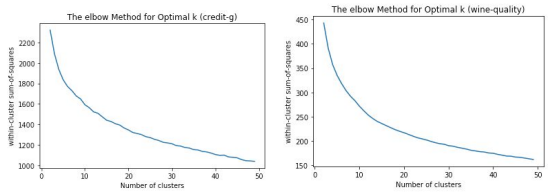

Figure 3: Elbow Curves for both datasets

For both of the datasets we cannot see an clear indication of an elbow therefore it is difficult to choose a k value. Expectation was for credit-g to have 2 cluster and wine-quality to have 7, which is not obvious from the elbow curves shown in figure 3.

Silhouette analysis is another popular method of estimating k as well as the performance of the cluster. When the labels for datasets are unknown, performance can be measured by two aspects cohesion between objects in a cluster and separation between the clusters. The silhouette coefficient is calculated using the mean intra-cluster distance and the mean nearest cluster distance for each sample. The best value if 1 (high separation) and worst value is -1, 0 value indicates overlapping cluster.

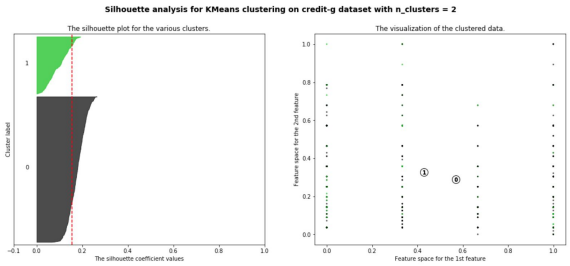

Figure 4: Silhouette analysis for k means with 2 clusters (credit-g)

In credit-g dataset, the highest average silhouette score of 0.156 was noticed for 3 clusters but it also had some negative silhouette score which indicated overlapping cluster, but 2 cluster plot has no negative values and silhouette score is 0.155, all other cluster values upto 10 clusters had lower average silhouette score, hence we can conclude based on average silhouette score that k=2 is the optimal k for credit-g dataset,which seems reasonable as we are dealing with binary classification dataset.

The cluster visualization for credit-g does not look as obvious separation of clusters, and the silhouette score itself is very low, the dataset may not be clustered we will look at few more performance metrics later. The straight lines in visualization is because of categorical values. The visualization is for first two features only so it doesn't represent the entire dataset.

In case of the wine dataset, the silhouette score for 2 clusters was 0.24 while the expected 7 clusters had just 0.14 score. This was an unexpected outcome, but the distribution of classes in the dataset has very high instances of two classes in comparison to other classes so clustering the dataset in 2 classes does make sense, although 7 would match our expectation as we have a labelled dataset. We will select 2 as the optimum value for k, in this dataset as well based on silhouette score.

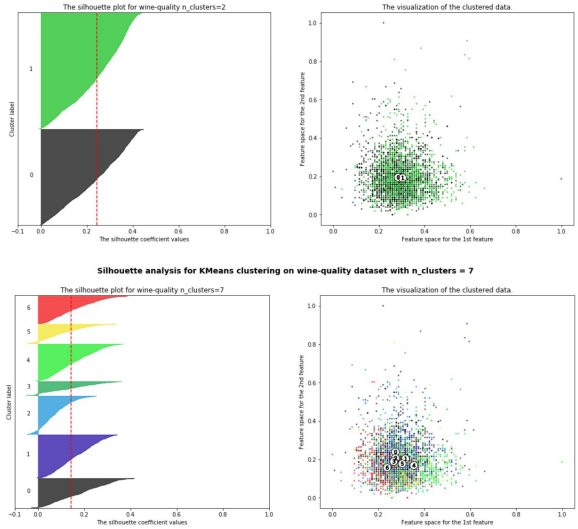

Figure 5: Silhouette analysis for k means with 2 and 7 clusters (wine-dataset)

The visualization for wine-quality dataset looks more spreaded due to continuous valued attributes, but it does not have clear separation as we would like to and very low value of silhouette score highlights that fact. This dataset also doesn't seems to form good clustered using k means.

3.2 Expectation Maximization (EM)

In expectation maximization algorithm, the goal is to maximize the overall probability likelihood of that data belonging to a cluster. While K means creates hard clusters, or assigns data points to one and only one particular cluster, EM does soft clustering by providing probability or cluster memberships based on one or more probability distributions. It starts with a random gaussian model centered on data points or learned from k-means, then computes probability of each point belonging to that probability distribution model, parameters such as the mean and variance is then adjusted to maximize the likelihood of the data given those assignments. It repeats the process until convergence to a local optimum.

How many components?

In case of Gaussian Mixture Model which uses EM algorithm, we need to know the number of mixture components to use for clustering. The

optimal number of clusters is the value that minimizes the Akaike information criterion (AIC) or Bayesian information criterion (BIC) of the current model.

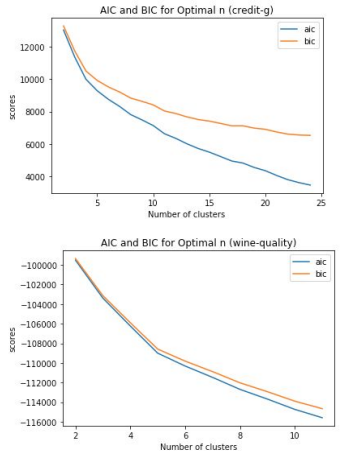

Figure 6: AIC and BIC curve for credit-g and wine-quality dataset

It seems like the for both dataset AIC and BIC scores keeps decreasing indicating a cluster equal to total number of features would be minimal but that does not make sense, we do see slightly hard slope change ast 3 clusters for credit-g and 5 for wine-quality.

Gradient curves were also constructed for both datasets, in case of wine-quality dataset as shown in figure 7 indicated that there is not much gain with increasing number of cluster higher than 6.

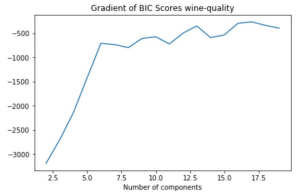

Figure 7: Gradient curve for BIC scores (wine-quality)

The evaluation in terms for wine-quality does make sense as we have 7 classes, 6 clusters would make sense.

We can conclude that 3 mixture components for credit-g and 6 mixture components for wine-quality dataset if optimal for EM clustering. But still it is not obvious, we will now look at performance metrics and see if we can gain more insights on the number of cluster that would result in the highest internal as well as external performance.

3.3 Clustering Performance Analysis

Performance for supervised learning is easy to measure as we have the true labels but for unsupervised learning measuring performance is dependent on internal measure the goodness of the cluster structure itself, focusing on cohesion and separation. We will use silhouette coefficient to measure the goodness of the cluster created by both of the algorithms.

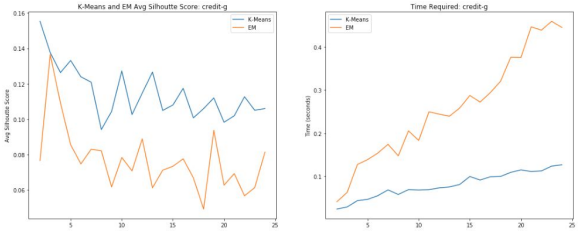

Figure 8: K-Means EM comparison Silhouette Score and Time (credit-g)

In credit-g dataset K-Means resulted in higher performance as well as faster time for convergence. At 3 clusters both K-Means and EM has similar score while the expected 2 cluster in credit-g has lower score with EM. For credit-g dataset K-Means seems to be the better algorithm to use as it converges faster and results in better cluster.

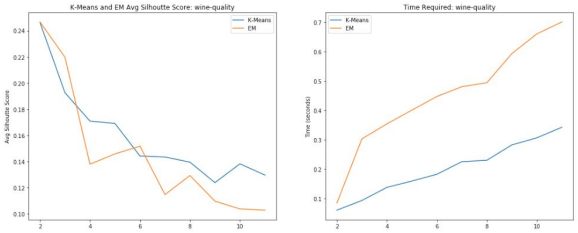

Figure 9: K-Means and EM comparison Silhouette Score and Time (wine-quality)

The Silhouette score for wine-quality dataset is similar weather we use K-Means or EM as clustering algorithm while K-Means converges faster. Both algorithm indicates k=2 is the optimal choice based on silhouette score, which was also discussed earlier. One reason for similar outcome could be that spherical cluster was used for EM, which is similar to how K-Means would cluster as well. There is not

much difference in the cluster quality and convergence is faster, K-Means would be the better algorithm to use with respect to internal measure of the cluster goodness for these datasets.

Since we have labels available for both the datasets we can use external measures and validate against the ground truth using metrics such as homogeneity, completeness and mutual information etc. there are half a dozen or more available. The behaviour of different external metrics in credit-g dataset using k-means is highlighted below.

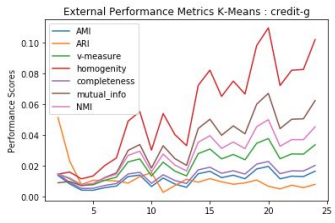

Figure 10: K-Means External Performance metrics for credit-g dataset

Normalized Mutual Information (NMI) is often used but is not normalized against random labeling. Homogeneity, Completeness and V-Measure is not normalized against chance, when cluster size is closer to the number of samples they increase significantly. Metrics that are adjusted for chance can only be safely used to evaluate the stability of clustering algorithms. ARI works better when there is large equal sized cluster while AMI is good when ground truth clustering is unbalanced and there exists small clusters. Since we have an unbalanced dataset, for this analysis we will select AMI to measure cluster performance based on ground truth and compare performance with K Means and EM.

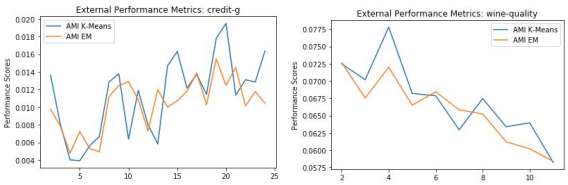

Figure 11: AMI comparision K-Meand and EM

As identified earlier if we were to select 2 clusters for both dataset based on silhouette score, the AMI would not be optimal for either dataset. Based on AMI wine-quality dataset with 4 clusters would perform best either we use K-Means or EM. In case of credit-g dataset, around 20 clusters is the highest but we know that silhouette score would suffer with that choice.

In Summary, the performance score for either datasets were not satisfactory, hence both datasets would not perform well in unsupervised learning with clustering. But, if there were not labels and only choice was do perform clustering using K-Means with 2 cluster for credit-g and wine-quality both, seems to be a good choice in terms of internal and external cluster metrics. But for wine-quality choosing 4 cluster would result in higher AMI or accuracy based on ground truth which makes sense, while we might compromise a little on the silhouette score itself.

4. Dimensionality Reduction Algorithms

Feature selection is to find a subset of features or attributes using different strategies such as filter, wrapper and embedded strategy. When a dataset has many features, feature selection is usually performed to avoid the effects of the curse of dimensionality. While feature selection simply selects or removes attributes dimensionality reduction is about linear transformation of data into low-dimensional space in which it still preserves the relevant or useful original information.

To give equal importance to all features in both datasets, scikit-learn MinMaxScalar is used so that magnitudes for all feature values are scaled between 0 and 1.

4.1 Principal Components Analysis (PCA)

PCA is a linear transformation algorithm, which can be used to reduce the dimensionality of a dataset by projecting it to a lower dimensional space, where the eigenvectors will form the

axes. Reducing dimensionality or the number of features in a dataset helps visualize it in 2 or 3 dimensions, but most importantly it increases computational efficiency which we know increasing exponentially is most of the algorithms due to curse of dimensionality. PCA helps by reducing the dimensionality while retaining most of the information. Using sklearn.decomposition.PCA from Python scikit library which uses Singular Value Decomposition following analysis were done.

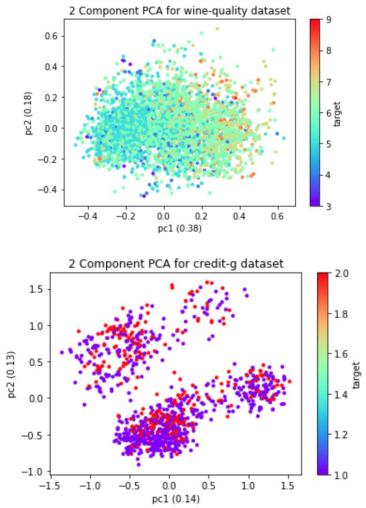

***Figure** 12 Component PCA Visualization for wine-quality and credit-g dataset*

Figure 12 shows 2 dimensional visualization using PCA for both datasets, we can see that for wine-quality dataset, first principal components contains 38% of variance while second component contains 18% variance, total 56% of variance. In the credit-g dataset only about 27% of variance was captured by first two components. It was interesting to see that only 2 dimensional PCA can contain about 60% of the information for the wine-quality dataset. The second part of PCA analysis where we analyze for increase in computational efficiency will be discussed later when we analyze using Neural Networks from previous assignment.

Optimal number of Principal Components can be chosen by inspecting % variance explained and the eigenvalues. As a rule of thumb if cumulative explained variance ratio is above 85% no additional components is required. When eigenvalues are significantly low they do not add information anymore hence additional principal components can be ignored.

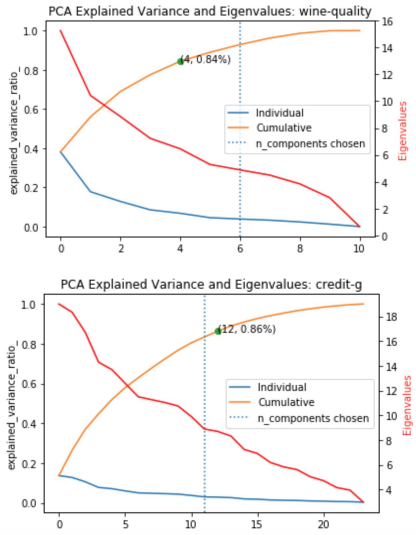

***Figure 13**: PCA Explained variance and Eigenvalues distribution for wine-quality and credit-g dataset*

In the above given plots, explained variance ratio of individual component and cumulative variance is shown, in case of credit-g 85% cumulative variance can be achieved by 4 principal components, for wine-quality 11 components are required for to capture 85% of variance. First two components contribute significantly in case of wine-quality. The red line indicates the eigenvalues distribution. In both cases components with eigenvalues less that 0.4 can be safely ignored as they do not provided additional information.

The blue dotted line in the graph represents highest cross validation accuracy (f1_score). Since we had true labels for the dataset, additional analysis was done using sklearn pipeline where a decision tree was used to measure accuracy score using cross validation based using PCA transformed reduced dataset, it helped in analysis as it gives information of whether using PCA for dimensionality reduction will actually improve

performance (f1_score) or not. The cross validation accuracy seems to agree very closely to the 85% rule for cumulative explained ratio, taking the first k eigenvectors that capture at least 85% of the total variance

4.2 Independent Component Analysis

ICA algorithm tries to separate linearly mixed source and find independent components of the original data by maximizing the difference between components . We will use FastICA algorithm available in scikit library which uses kurtosis as the cost function. Kurtosis measures the degrees of non-gaussianity, kurtosis value of 3 would be a gaussian distribution.

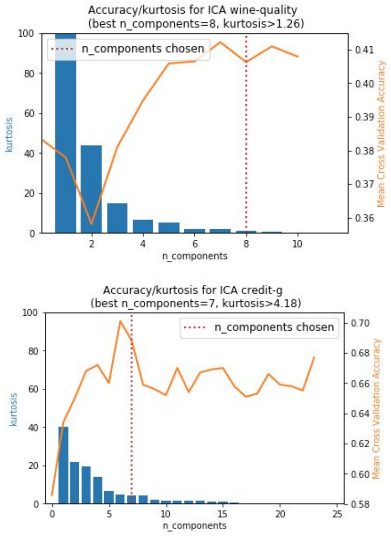

***Figure 14**: ICA Kurtosis distribution for wine-quality and credit-g dataset*

The kurtosis distribution of each component was not sorted from larger to smaller with ICA algorithm the sorting was done to help select components with higher kurtosis, it required some work in addition to the algorithms output as the order of components would not always be the same. The kurtosis value for the first component and second component in wine-quality dataset was much higher than others, this indicates similarity with PCA were we had similar behaviour with first two components. With help of labelled dataset mean cross-validation accuracy is also shows in the diagrams with the best mean accuracy represented by dotted lines. Based on the kurtosis distribution for wine-quality dataset components with kurtosis value lower than 1.26 will be ignored and lower than 4.18 will be ignored in case of credit-g. The best value for number of components is 8 and 7 respectively.

4.3 Randomized Projections

Randomize projections is a computationally efficient way to reduce the dimensionality by projecting the original input space on a randomly generated matrix where components are drawn from a gaussian distribution with variance which are equal to inverse number of components and zero mean. It is used for distance based method by varying dimensions and distributions of random projections matrices the pairwise distance between any two instance of dataset is preserved.

In figure 15 reconstruction error of Randomized project and cross validation accuracy is shown for both datasets. The number of components will be selected based on the least reconstruction error.

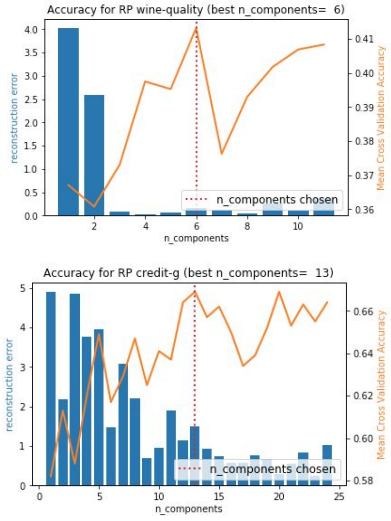

Figure 15: Reconstruction Error and accuracy for RP

It was interesting to see that as the number of components increased the reconstruction error decreased. Based on cross validation accuracy 6 components for wine-quality dataset and 13

components for credit-g dataset gives the optimal accuracy, the reconstruction error is also comparatively lower at that point. The results varied for each execution hence a mean value for 10 executions for reconstruction error was used.

4.4 Factor Analysis

Factor analysis performs linear transformation to lower dimensions with added gaussian noise. Similar to PCA explained variance ratio is measured. The difference is that noise in PCA is assumed to be spherical while in Factor Analysis noise can have an arbitrary diagonal covariances. The figures below show noise variance for both of the datasets.

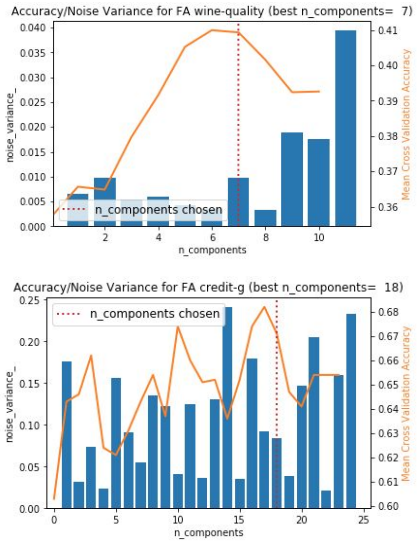

Figure 16: Noise Variance and Accuracy for FA

In wine-quality dataset out of 11 components 3 components have very high noise variance, taking components with noise variance below 0.010 gives us 7 components to be a good estimate. In credit-g dataset taking components with noise variance below 0.10 gives 15 components which filters out the components with very high noise variance. The cross validation accuracy is also the highest at those values for number of components.

5. Clustering after Dimensionality Reduction

Using the number of components chosen from analysis in section 4 for both datasets clustering experiments were reproduced on the transformed dataset using the dimensionality reduction algorithms.

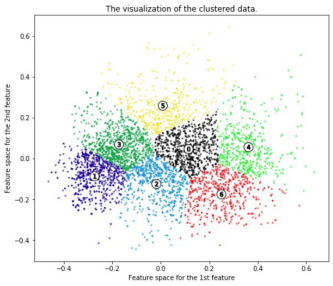

Figure 17: PCA transformed cluster wine-quality 7 clusters and 6 PCA components

Figure 17 shows the clusters formed using PCA reduced dataset using 6 components in the wine dataset. The visualization of clusters were much better for both the dataset when ran after the dimensionality reduction, must be because of the first two principal components which are visualized retained the most variance and also clustering is easier with less features.

Analysis was done to create the optimal cluster selected from section 3, ie 2 clusters for both dataset, using transformed data from dimensionality reduction with optimal number of components chosen from section 4. One internal measure - silhouette score, external measure - AMI and time was measured.

	FA	ICA	PCA	RP	_K-Means
AMI	0.013600	0.013600	0.013600	-0.000106	0.013600
silhouette	0.150881	0.227792	0.185598	0.172175	0.155310
time	0.025861	0.021041	0.026784	0.029483	0.024192

	FA	ICA	PCA	RP	_EM
AMI	0.006250	0.007495	0.013580	-0.000637	0.009766
silhouette	0.034576	0.072786	0.186053	0.180894	0.076612
time	0.097933	0.085216	0.060514	0.060905	0.041288

***Table 1**: Performance metrics for K-Means and EM Clustering credit-g dataset*

In credit-g with K-Means clustering the best silhouette score was achieved with ICA which took about the same about the time to

converge. Interesting observation was the AMI score for all is same, except for RP which is negative in both K-Means and EM. In terms of time efficiency in K-Means clustering RP is the worst while clustering after ICA seems to be most efficient. With EM algorithm in credit-g dataset, PCA resulted in highest AMI as well as highest silhouette. Most efficient in terms of time was without transformation.

ICA reduction before K-Means and no reduction for EM seems to perform better with credit-g dataset.

	FA	ICA	PCA	RP	_K-Means
AMI	0.021810	0.072666	0.072334	0.059298	0.072488
silhouette	0.139480	0.255510	0.276686	0.342815	0.246191
time	0.118364	0.062990	0.084926	0.062370	0.060914

	FA	ICA	PCA	RP	_EM
AMI	0.012269	0.072810	0.069232	0.058002	0.072652
silhouette	0.071003	0.255875	0.271558	0.340667	0.246652
time	0.198344	0.108662	0.192190	0.122485	0.085181

***Table 2**: Metrics for K-Means and EM Clustering wine-quality dataset*

With wine-quality dataset and K-Means clustering AMI for FA was the lowest while ICA, PCA and original dataset performed almost similar. Interestingly RP had the highest silhouette score and all transformation resulted in slight time increase. With EM clustering best AMI was achieved from ICA reduction but it is similar to one without reduction, silhouette for RP is the highest and without transformation is the fastest.

PCA reduction before K-Means and PCA reduction before EM resulted in better results for wine-quality dataset.

Overall significant improved in terms of accuracy or time efficiency was not achieved in both of these datasets from using dimensionality reduction before clustering but the clustering had low performance to begin with due to the dataset itself, dimensionality reduction did not help much. We kept the number of clusters static, if we had varied the number of cluster as well as component like grid search we might have achieved better performances.

6. Neural Network Performance

I have selected credit-g dataset to train a neural network and analyze the performance with or without dimensionality reduction.

6.1 Dimensionality Reduction and Neural Network

Neural Networks was implement using Scikit learn MLPClassifier, highest accuracy of 70% was achieved using single layer with 40 nodes and logistic activation. I decided to not use the component counts selected from before but rather analyze across components counts equal to number of features. Since the dataset is unbalanced, the performance metrics used is f1 score from 3-fold cross validation in testing set.

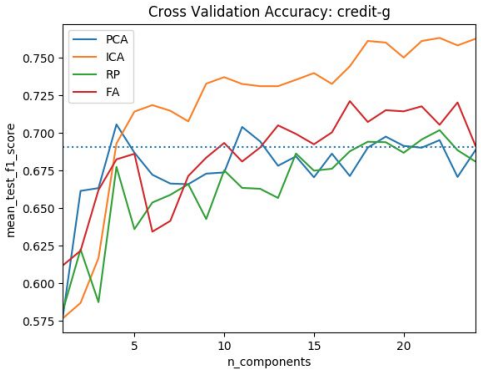

Figure 18 Mean cross validation score credit-g dataset

From the above shown analysis, where the blue dotted lines represent performance without dimensionality reduction, we can see that if ICA was to be used for dimensionality reduction with number of components higher than 5 neural network would yield higher accuracy not a lot but 2-5% gain. PCA with 5 components also would yield slightly better result but not significant enough, FA with component higher that 12. Overall using any other techniques than ICA would not improve the performance of neural network.

It was interesting to see that just 7 components from ICA transformed data, or 5 components from PCA transformed would result in better or equivalent performance instead of using all 24 features/dimensions to train the Neural Networks.

Figure 19 Neural Networks training time vs number of components

Using 5 components any of the dimensionality reduction method resulted in higher than 65% accuracy, dimensionality reduction would definitely help this dataset to be evaluated much faster, which can be confirmed by the figure shown above, where the dotted blue line for dataset without reduction is much higher than the time to train after reduction using 5 components using any reduction method out of the four, which would result in similar accuracy of 65% or higher.

6.2 Clustering and Neural Networks

In this section we will analyze how performance of the dataset varies if the clusters were treated as if they were new features with lower dimensions. To achieve this setup, first clusters were created from all data samples and the original y values were combined to create a dataset. From the transformed dataset then training and testing sets were split for cross-validation. In case of expectation maximization transform method was not available in sklearn library but instead random samples drawn from the fitted distribution was used for analysis.

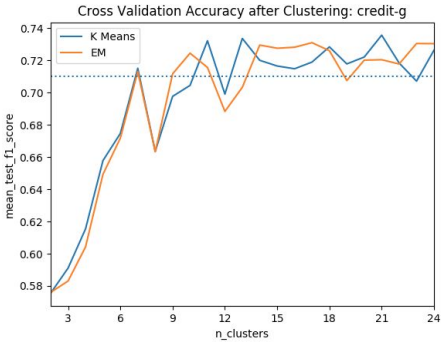

Figure 20: Mean Cross validation score after clustering credit-g dataset

From figure 20 we can see that using 12 clusters with K-Means and 15 clusters with EM, could result in a slight improvement in performance but the interesting observation is that from a dataset with 24 features, if 10 clusters are created using EM or KMeans, that is more than half reduction in dimensionality we could achieve similar accuracy while training a neural network.

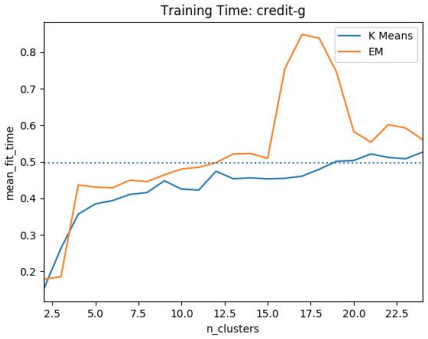

Figure 21: Neural Networks training time vs number of componentst

As expected the time required to train the model also reduces when using 10 clusters, not by a lot but 0.1 second.

If the parameter for neural networks were to be adjusted to fit the dimensionality reduced data, we should get better performance within much less time, In this analysis we did not tune the hyperparameters for neural network, yet the effectiveness of dimensionality reduction using either clustering techniques or dimensionality reduction algorithms could train a neural network in much faster time with similar or even better accuracy in the German Credit dataset.