CS 7641 Machine Learning Assignment 3

Unsupervised Learning and Dimensionality Reduction

Tao Peng, Novermber 2016

1. Introduction

This assignment is to explore and study unsupervised learning and dimensionality reduction. I ran two clustering algorithms (k-means and EM) and four dimensionality reduction algorithms (PCA, ICA, RP, Info Gain) on the same two data sets as Assignment 1. I also studied how clustering behaves after using new features from dimensionality reduction, and how neural network behaves after using new features from dimensionality reduction and from taking clusters as new features. Different algorithms are compared. For each algorithm, modifications of the parameters are made in order to see how to improve. Weka was used to all the algorithms.

2. Datasets

2.1 Wine Quality Data Set

This data set is also from the UCI Machine Learning Repository [3]. It is to determine the sensory quality of the Portuguese “Vinho Verde” wine. There are 11 classes of the wine quality (score from 3 to 9), which are determined from the 12 attributes. The attributes are the physicochemical feature of the wine, such as fixed acidity, residual sugar, pH. In order to reduce the running time, I used 70% of the white wine data set, which originally contains 4898 instances.

This data set is interesting. The quality of the wine is based on the sensory outcome, so if we do not use machine learning, we may have to destroy the wine by tasting it in order to determine its quality, which is not realistic in actual production process. Moreover, it may be quicker to determine the quality by running our models in computers than spending time tasting them one by one.

2.2 Letter Recognition Data Set

This data set is from the UCI Machine Learning Repository [2]. Its purpose is to recognize distorted 26 distored English capital letters. Each letter is displayed by a large number of black-and-white rectangular pixels. Each letter has many different shapes, which are based on randomly distorted fonts. To identify those letters, 16 numerical attributes were used. Each attribute describes one feature of the shape of the letter. For examples, the overal horizontal position of the letter, the total width, the mean x of the pixels, etc. The data has 20,000 instances and 16 attributes. There is no missing values. In order to reduce the running time, I used 70% of the instances in this assignment.

This dataset is interesting because it is very similar to the Captcha, which is a combination of distorted letters used to test whether or not the user is human. It is very interesting to find a way for computer to recognize Captcha so that they can pretend to be human.

3. Clustering

3.1 K-means

For k-means, I studied its behavior with both the Euclidean and Manhattan distance definitions. The metric for Euclidean is within cluster sum of square errors, and for Manhattan is sum of within cluster distances. I varied the number of clusters (k) and Figure 1 is its relation with the metrics. To find the best k, I used the elbow method. The purpose of this method is to keep the error small while still maintain a small enough k. From the figure, it seems that the best k is 6 for Wine Quality Dataset, and 8 for Letter Recognition Dataset.

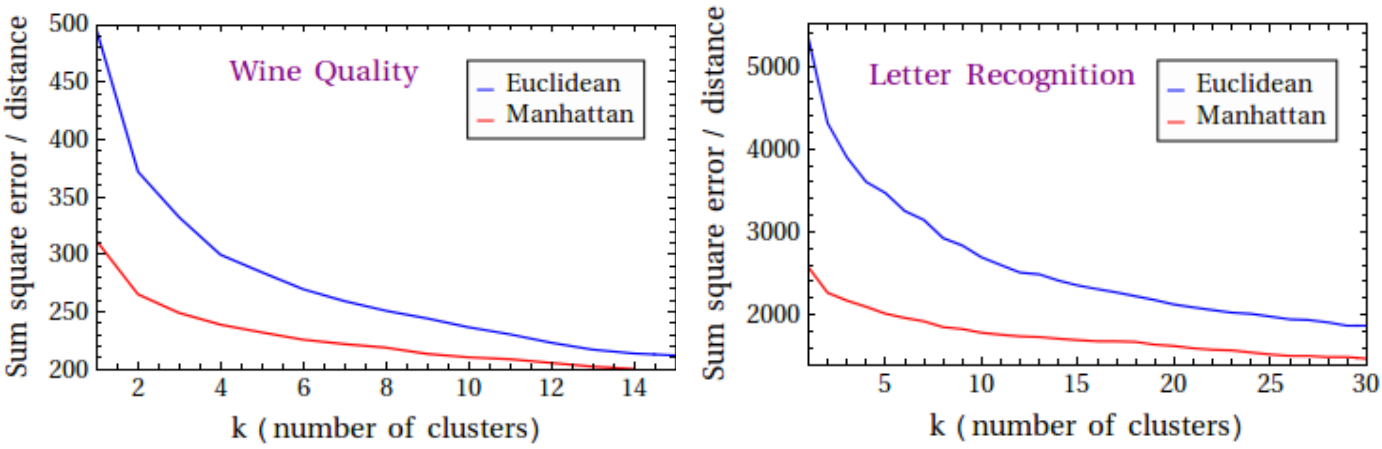

Figure 1: Sum of square error or sum of distances vs k (number of clusters) for k-means. I divided the sum of distance values of Mahattan by 10, in order to make it appear in the same figure as Euclidean.

We can also compare the clustering with the original classes. The result of comparing is shown in Figure 2. The first interesting thing to notice is the computing time for the original Letter Recognition Dataset which has 26 classes (one for each English letter). It had been running for more than six hours and still did not stop. After I reduced the number of classes to 10, by removing the instances of some classes, the running time reduced significantly to several seconds.

The second interesting thing is how the clusters line up with the original classes. To compare, I chose k to be the same as the number of original classes. For Wine Quality, the incorrectly clustered instances is 73% (Euclidean) and 74% (Manhattan). For Letter Recognition, it is 58% (Euclidean) and 61% (Manhattan). I think this low accuracy is due to the nature of the datasets.

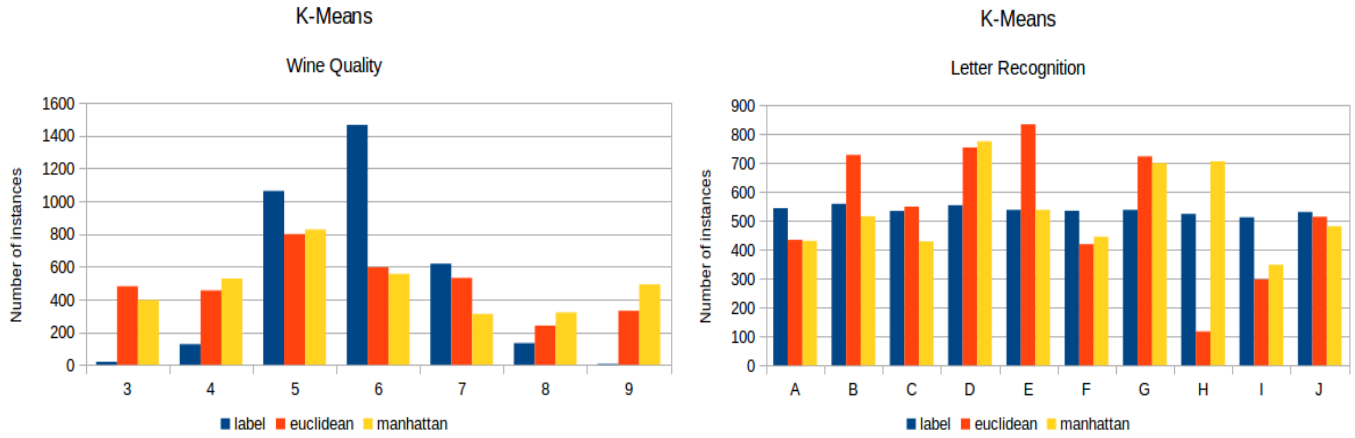

Figure 2: Comparison between the clusters and the original classes for k-means. For letter recognition, I reduced the number of classes in the dataset to 10 in order to save computing time.

3.2 Expectation Maximization (EM)

For EM, I studied the relation between the log likelihood and the number of clusters, as shown in Figure 3. I used different seeds, but it seems that it did not make much difference. The best k determined by Weka is 12 for Wine Quality and 7 for Letter Recognition. But from the elbow method, it seems it is more like 10 for both Wine Quality and Letter Recognition.

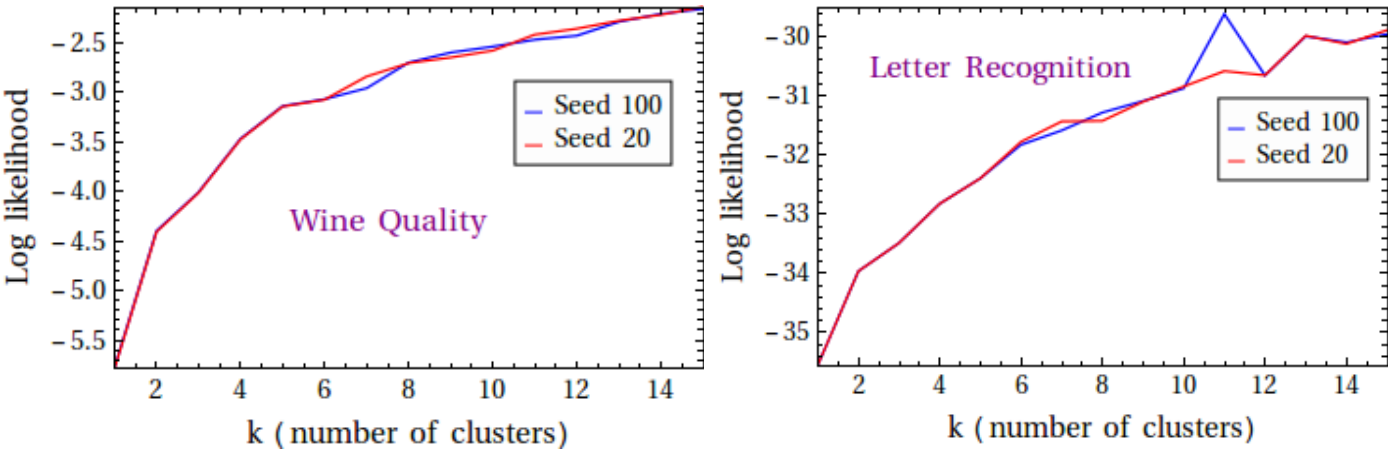

Figure 3: Log likelihood vs k (number of clusters) for EM.

The comparison between clusters and the original classes is similar as the k-means case, as shown in Figure 4. The incorrectly clustered instances for Wine Quality is 73% (seed 100) and 74% (seed 20), and for Letter Recognition is 62% (seed 100) and 61% (seed 20). The cluster distribution seems to be more uniform than the original class distribution.

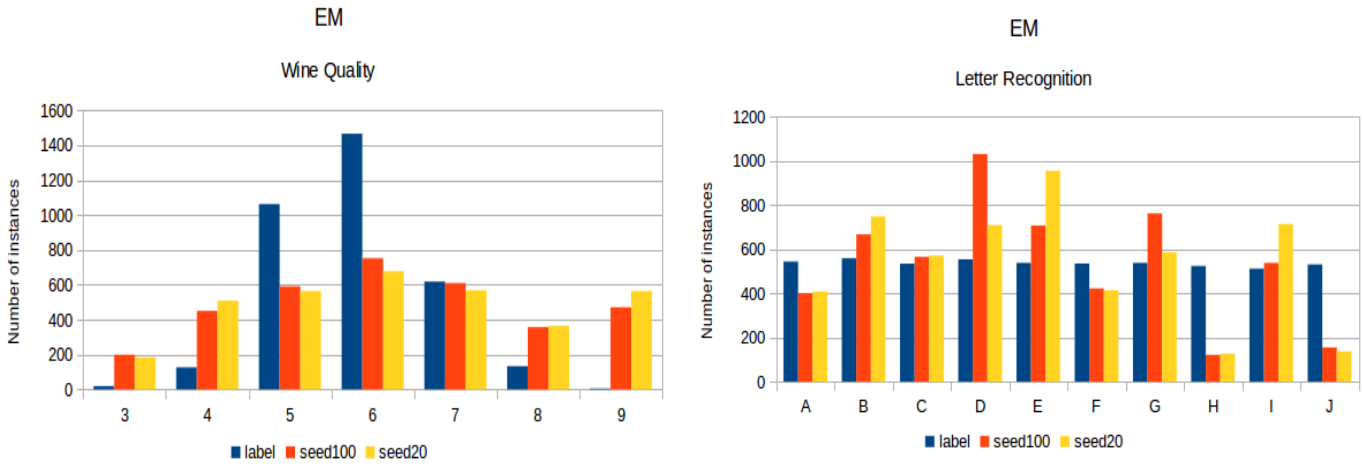

Figure 4: Comparison between the clusters and the original classes for EM. For letter recognition, I reduced the number of classes in the dataset to 10 in order to save computing time.

4. Dimensionality Reduction and Its Effect On Clustering

4.1 Principal Component Analysis (PCA)

In PCA, to find which principal components to keep after dimensionality reduction, I studied the the eigenvalue and cumulative variability of each principal component, as shown in Figure 5. We like to select the components with largest eigenvalues and smalles cumulative variability. From the figure, I would chose the first 3 principal components for Wine Quality and first 5 for Letter Recognition. I chose more for Letter Recogintion because it has much more attributes than Wine Quality.

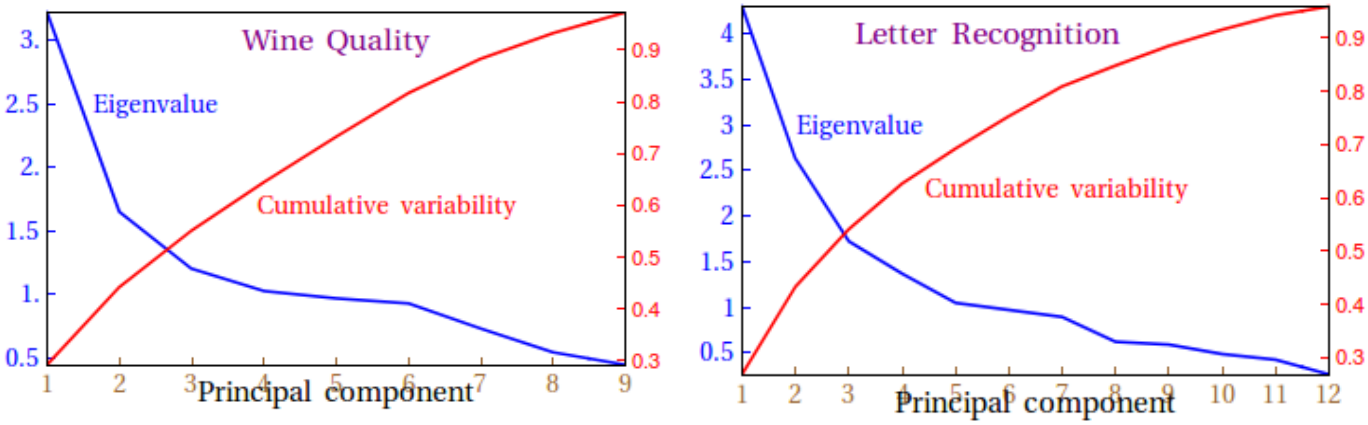

Figure 5: Eigenvalue and cumulative variability vs principal component. Each number in the horizontal axis denotes a principal component.

After selecting the principal components, we can use them to do the clustering again. Figure 6 shows the distributions of clusters of different datasets after using different clustering algorithms. In order to show them in a 2-D plane, I used only two components in the plot. We can see that the clusters in Wine Quality ae mostly well seperated, while for Letter Recognition there are some overlap.

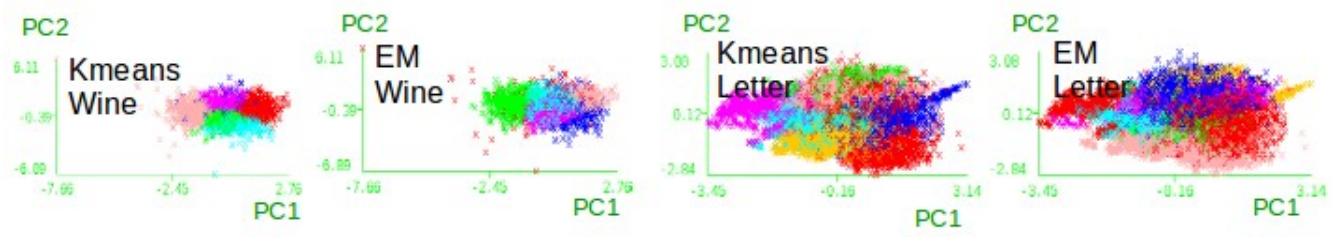

Figure 6: Cluster distrubutions after using PCA to reduce dimenstion.

4.2 Independent Component Analysis (ICA)

For the ICA, I select the components by the kurtosis. The kurtosis describes the tailedness of the distributions. Figure 7 and Figure 8 show the distributions of independent components and their kurtosis for the two datasets respectively. We can see that the shapes of the distributions are consistent with their kurtosis values. To do clustering, I select the components with 3 largest kurtosis for Wine Quality and 5 largest kurtosis for Letter Recognition.

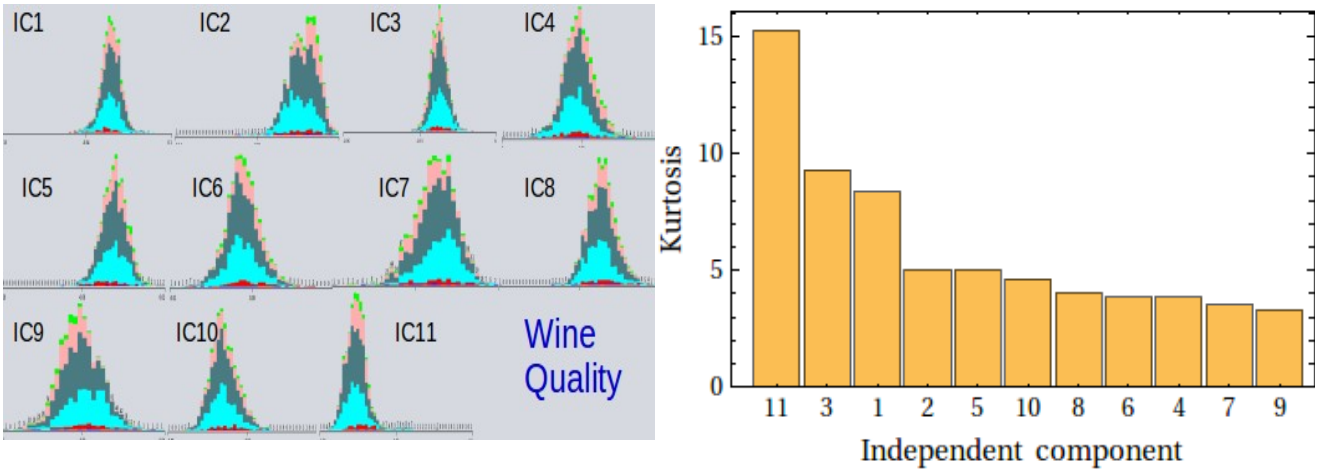

Figure 7: Distributions of independent components (left) and their kurtosis (right) for Wine Quality DataSet.

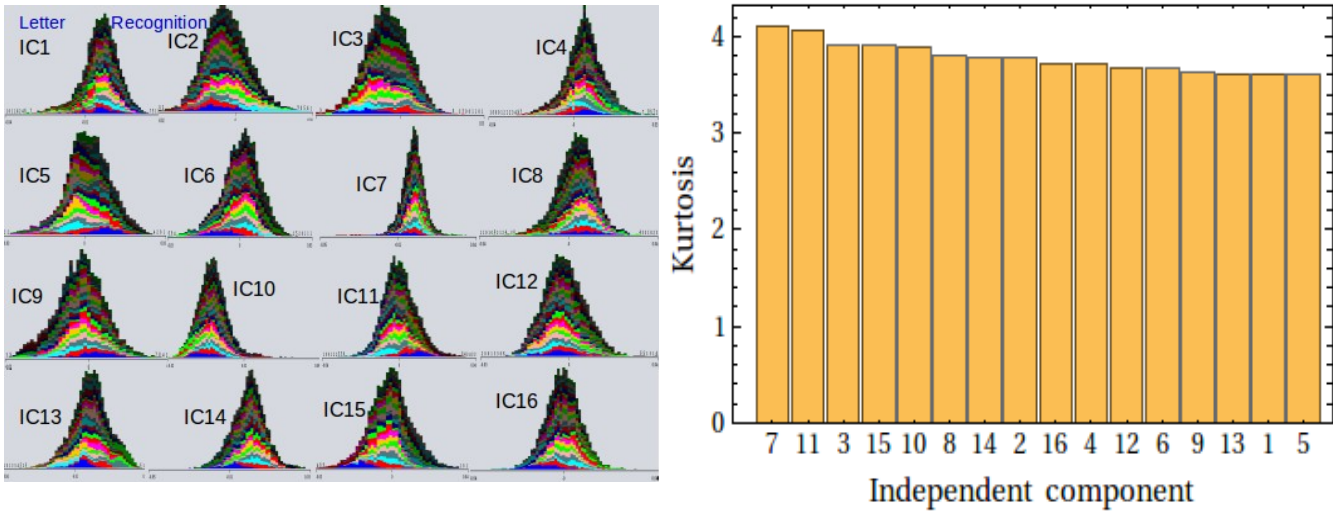

Figure 8: Distributions of independent components (left) and their kurtosis (right) for Letter Recognitoin DataSet.

I then used the selected independent components to do the clustering. The results are in Figure 9. Again, the Wine Quality data are better seperated than the Letter Recognition data. I think this is due to the nature of the dataset. The Letter Recognition Dataset is more uniform and thus harder to make clusters from them without knowing the labels.

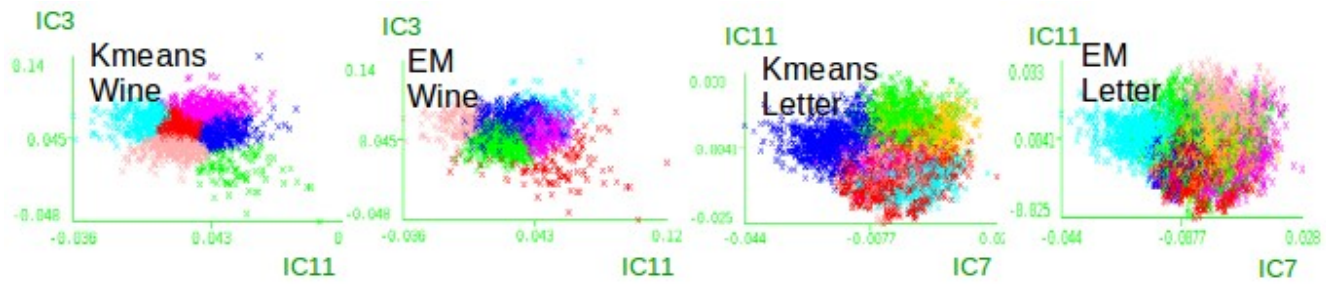

Figure 9: Cluster distrubutions after using ICA to reduce dimenstion.

4.3 Ramdom Projection (RP)

I re-ran the RP several times with different seeds and saw some difference in the curves, as shown in Figure 10. For different run, the curves have different overal variance. In other words, some curves are always above other curves. So I selected curves based on this. For Wine Quality, I selected the curve with seed 50 (red), and for Letter Recognition, I selected the curve with seed 30 (blue). Then among the selected curves, I selected 3 random components with largest variance for Wine Quality, and selected 5 with largest variance for Letter Recognition.

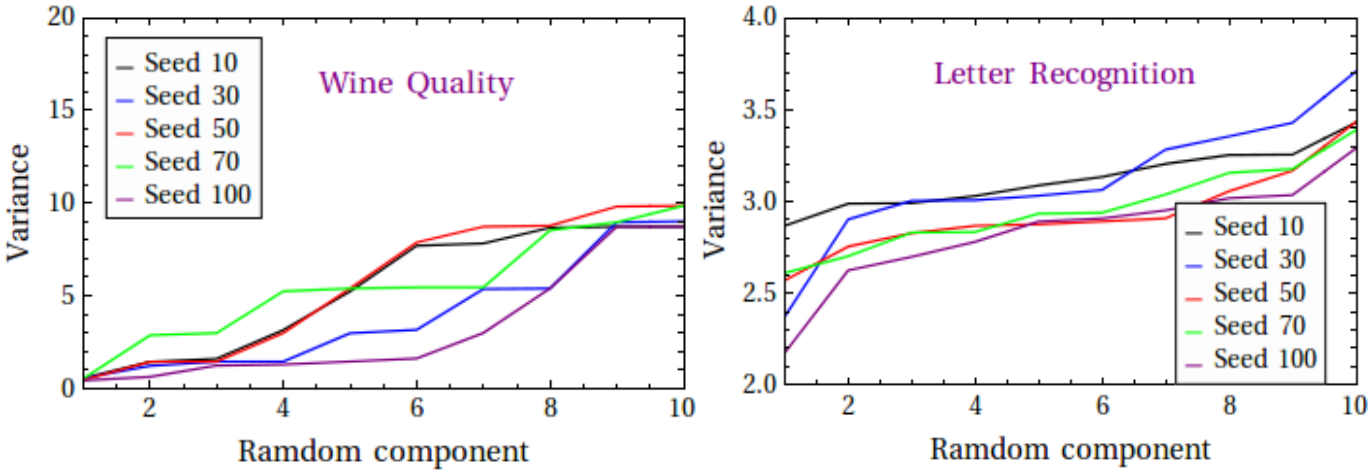

Figure 10: Variance vs ramdom components. Diffrent curves are results of re-running RP several times with different seeds.

I then used the selected components to do clustering, using both kmeans and EM methods. Figure 11 shows the result of clustering. What I noted is that the Wine Quality clusters are very well seperated. This means that RP may be a good choice to reduce dimensionality for the Wine Quality data.

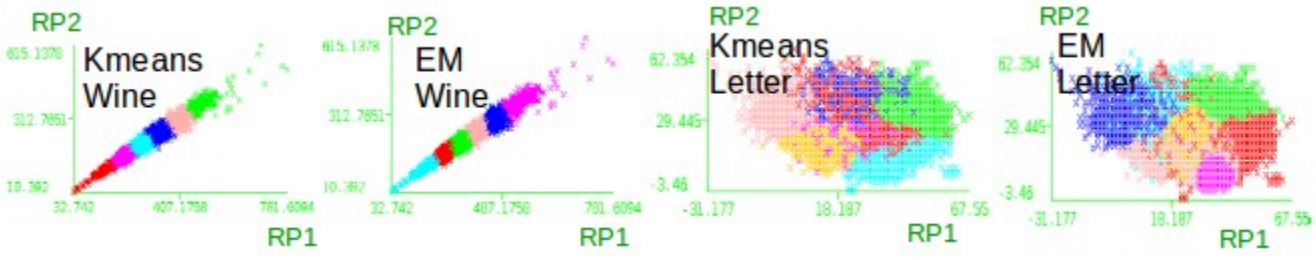

Figure 11: Cluster distrubutions after using RP to reduce dimenstion.

4.4 Information Gain Attribute Evaluator

The Information Gain Attribute Evaluator is an attribute selection method provided by Weka. It is based on the information gain of each attribute on decision trees. We can reduce the dimensionality by selecting the attributes with largest information gain. Figure 12 shows the information gain for each attributes in the two data sets. I select the three attributes (alcohol, density, chlorides) with largest informatin gain for Wine Quality and the five attributes with largest information gain for Letter Recognition.

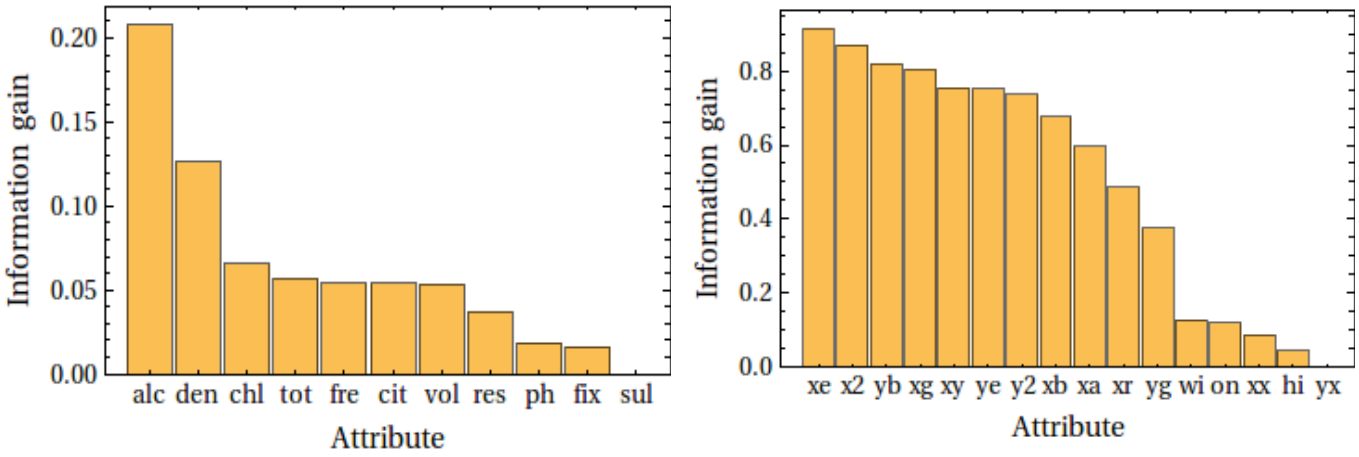

Figure 12: Information gain for each attribute of Wine Quality (left) and Letter Recognition (right).

Similar as other algorithms, we can use the selected attributes to do clustering. The results are shown in Figure 13. We can see that it does well for Wine Quality, but bad for Letter Recognition. I think information gain does worse for Letter Recognition because it just selects the original attributes, not making any linear combinations as what other algorithms do. So it has a smaller space to choose and thus may do worse for some datasets.

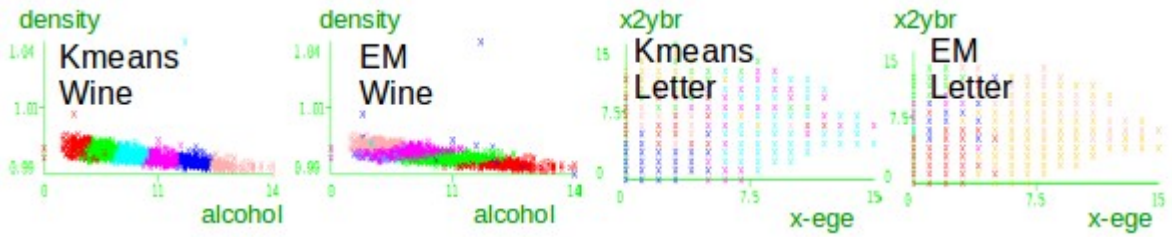

Figure 13: Cluster distrubutions after using Infomation Gain to reduce dimenstion.

5. Effects of Dimentionality Reduction on Neural Networks

5.1 Cross Validation Accuracy

I used the dimensionality reduction algorithms to reduce attributes of the Wine Quality Dataset, and use thed the newly projected data as inputs of neural networks. Figure 14 shows the effect of

dimensionality reduction on the cross validation accuracy, compared to neural networks with the original data (red dashed line). From Assignment 1, we know it is best to use 8 nodes in the hidden layer. In order to learn more about how the reduction of dimension affects the performance of neural network, I show the accuracy under different combinations of components. First we see that after dimension reduction, the accuracy is always smaller than the original data. That's expected because we have less information to use. We can then use this to find the best number of components. For PCA, ICA, and Info Gain it is around 6, while for RP it is close to 3.

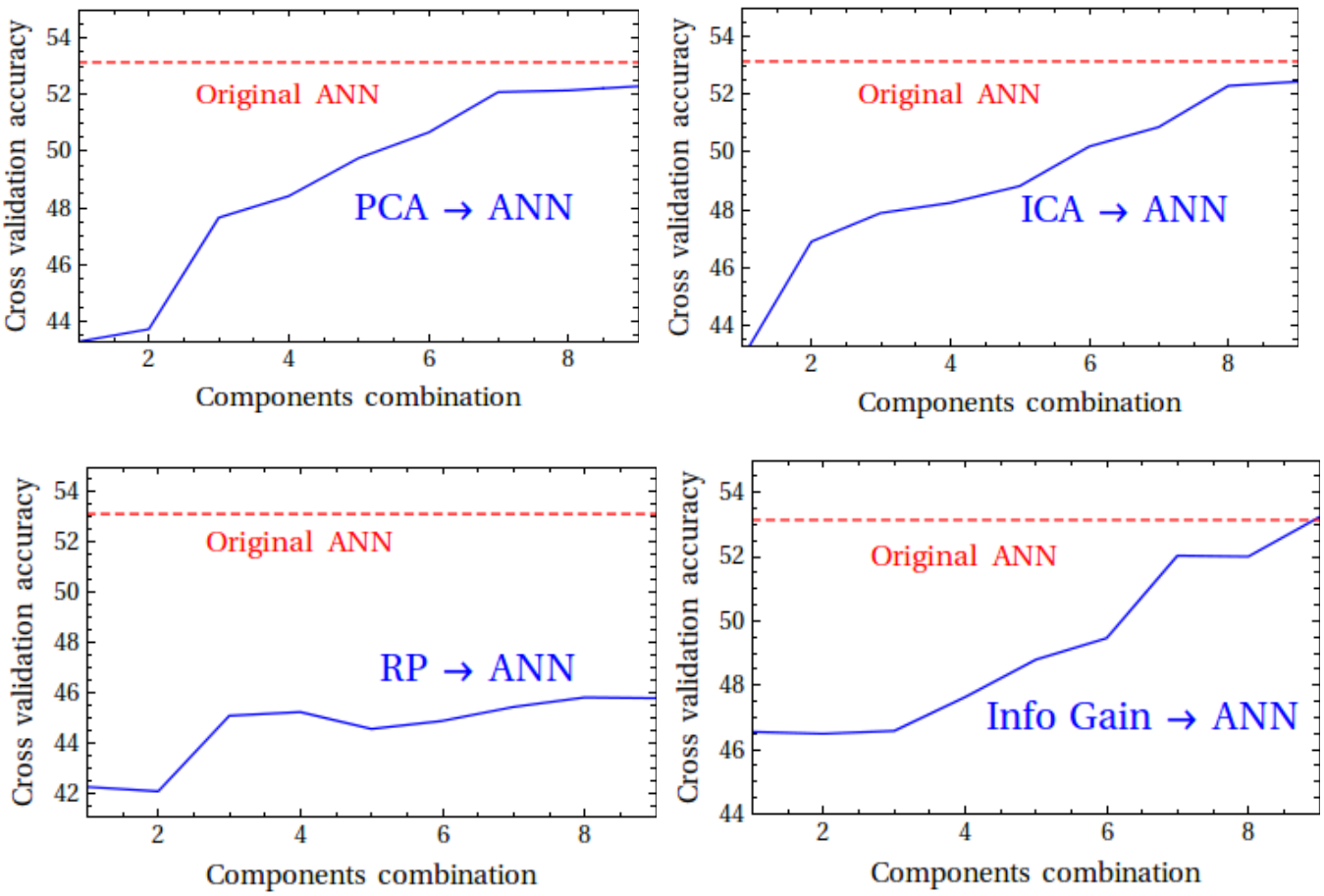

Figure 14: Cross validation accuracy vs components combination for neural network, after dimensinality reduction of each algorithm. The horizontal axis is different combinations of components. For example, 2 means the best 2 components, 4 means the best 4 components.

5.2 Running time

We can also study the effect on running time of neural networks. We see that as number of components grows, the running time almost grows linearly, regardless of the dimension reduction algorithm used. And the running time is always less than the neural network on the original data. This is also expected because every time we add one more attribute, the neural network has the same amount of extra work to do. So the running time grows linearly. This also shows the advantage of dimention reductoin: shorter running time.

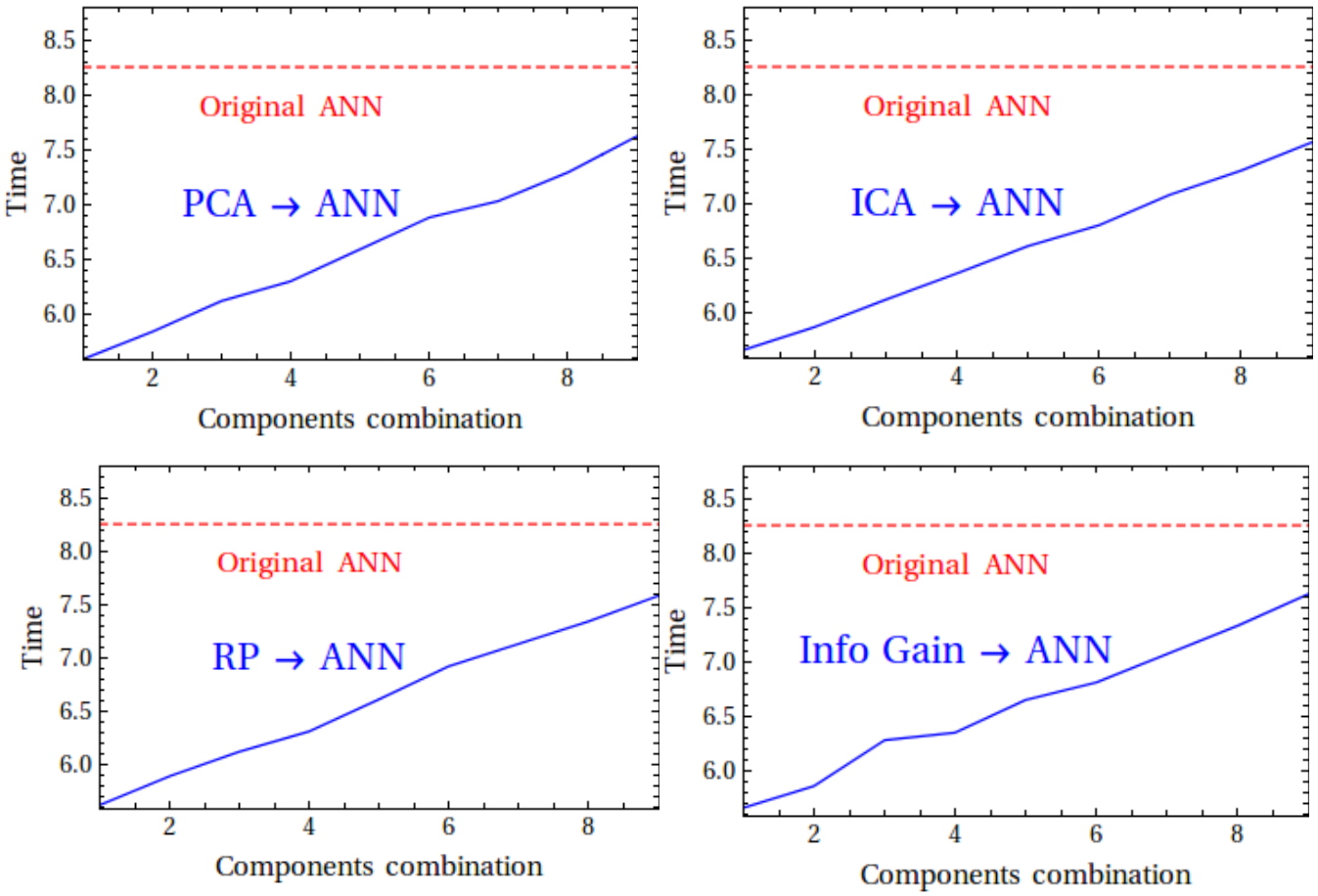

Figure 15: Computing time vs components combination for neural network, after dimensinality reduction of each algorithm. The horizontal axis is different combinations of components. For example, 2 means the best 2 components, 4 means the best 4 components.

6. Treating Clusters As New Features for Neural Networks

We can also first do clustering with different algorithms, and then take the cluster as new features for neural network. In this way, a cluster feature means which cluster each instance belongs to. Here I generate 4 clusters using each clustering algorithm. So that there are 4 new feature. For clustering, I used k-means with 4 different k (2, 4, 6, 8) to make 4 clusters, and did it with both Euclidean distance and Manhattan distance. I used used EM with one group of 4 seeds (10, 30, 50, 70) to make 4 clusters, and another group of 4 seeds (80, 100, 120, 140) to make another 4 clusters. Then I trained the neural network on these 4 new featuers.

The results are shown in Figure 16. We can see that the cross validation accuracy is smaller than the neural network with original data. That is because we have only 4 clusters as new features, which is less than the original number of attributes. So there is less information to use. However, the benefit is that the running time is reduced.

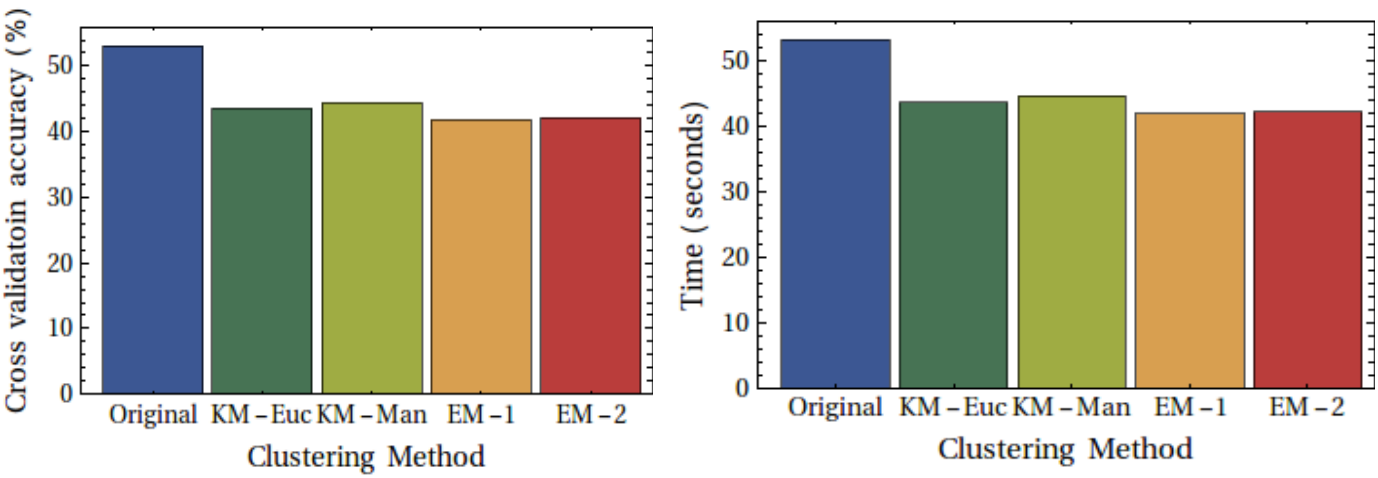

Figure 16: Cross validation accuracy (and running time) of neural network aftering treating clusters of each clustering method as new features.

7. Summary

In this assignment, I used the clustering and dimensionality reduction methods on the two data sets, and studied how the clustering performs after dimensionality reduction, and how neural networks performs after dimensionality reduction and treating clusters as new features. We see that different clustering algorithms behaves differently. And we see that the accuracy of classificiatoin is reduced after dimensionality reduction, however, the running time is shorter, which is the benefit of dimensionality reduction.

From this assignment, I learned more about the how clustering and dimensionality reduction algorithms behave under different circumstances, and had a deeper understanding of their properties. I also learned how to use the corresponding Machine Learning tools.

References

[1] https://archive.ics.uci.edu/ml/datasets/Wine+Quality
[2] https://archive.ics.uci.edu/ml/datasets/Letter+Recognition