CS 4641 - Machine Learning
Assignment 3 – Unsupervised Learning

Datasets

For this assignment, I decided to use one of the datasets that I have been using before which is the letters dataset. Briefly, the letters dataset is formed of 20000 data points that each point represents a written letter. 16 attributes describe these letters such as X and Y dimension, the degree of the curves available and many more that describe the physical shape of the written letter. The question is basically doing the OCR (Optical Character Recognition) having all the features and guessing the letter based on the given attributes.

For the second dataset, I decided to use a new one since the old one I was using covered the entire feature space (all possible feature combinations were given) and this lead to very similar cluster that doesn't allow us to infer anything. Instead, I used the "Wine Quality" dataset, which has 4898 data points that each of them belongs to a certain wine and covers 11 physical features such as acidity, density and the label, which is a score between 1 and 10. The score is basically a subjective score given by human testers and they have not been exposed to the attributes we've given; they only tasted the wines to label it with a score. The question here is to understand if any of those chemical or physical features of the wine actually contribute to the quality of the wine.

Why?

The reason I am using the letter dataset is the number of labels it has. Since the dataset is trying to capture all the letters in the alphabet, there are total of 26 different labels, and compared to any binary classification problems or low number of labeled datasets, this is much harder to cluster into since a lot of labels will be somehow interesting with each other in a given dimension. Also we know that each attribute physically builds the optical character image, so each attribute is important and should be distinctive in order to distinguish the label. Also, the number of labels is almost-equally distributed throughout the dataset, meaning that there are around 700-800 instances belonging to each of the letters. This will be an important point when it comes to clustering since the algorithms will try to create the clusters in similar sizes.

The wine dataset contrasts with the letter dataset in couple different senses. First of all it has less number of labels (10 labels) and the labels are given in a more arbitrary sense without the features are regarded. So in this paper, I will also be trying to determine if a feature actually is important or not, whereas in the letter dataset we know that each feature is important. In this aspect, dimensionality reduction algorithms will show if a feature is important and always give valuable information or if it can be reduced without really causing any decreases in the final accuracy value. Another difference of wine dataset is the distribution of the labels. There are less number of labels of really poor (ranked 1-3) and excellent wines (8-10), meaning that if we want to cluster each

ranking, some clusters will be expected to be larger however the algorithms might not be able to capture those differences.

Clustering Algorithms

K-means

The first clustering algorithm I tested was K-means clustering. To explain the algorithm briefly, it takes an input n, which is the number of clusters to produce in the end, and starts with random cluster centers that have components on each of the given dimensions (attributes) of the data. Using a distance function, it distributes each instance to the closest cluster center and in the end, takes the mean of the clusters to calculate the new cluster center and restarts the distribution process. This goes on until the centers' coordinates on the hyper-dimensions stops changing. I used the Euclidian distance as my distance function and I will be discussing its consequences in the next section.

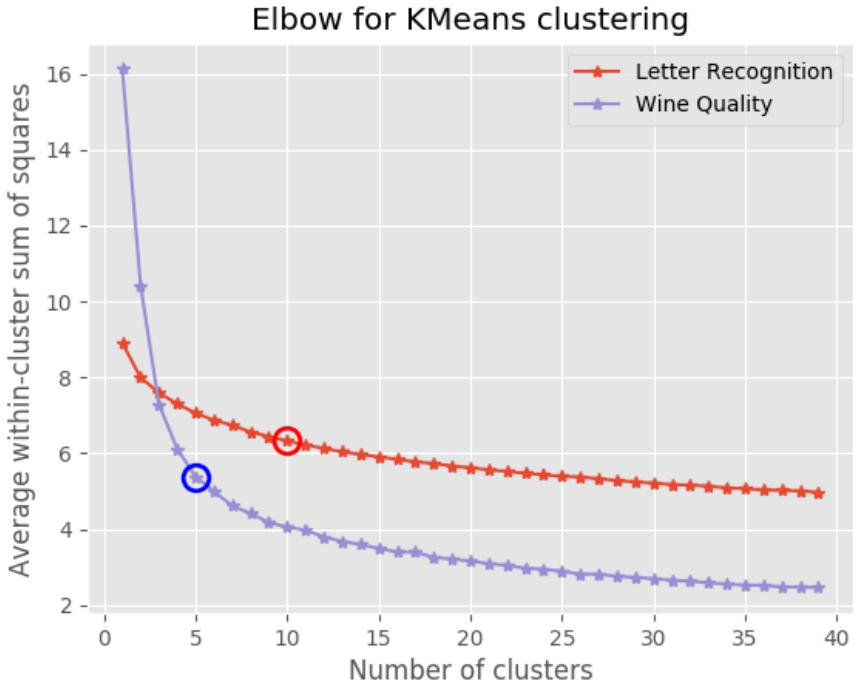

Figure 1: SSE (sum of squares), given each K (number of clusters) for both letter and wine datasets for K-Means Clustering.

In *figure 1*, the plot shows the average SSE values of clusters and instances for increasing K values. The plot shows the elbow method to choose the right k value that captures the best representation of the dataset in clusters. SSE basically means that how instances are far away from the cluster centers in total. As it can be expected, the more cluster centers are there, the lower error we get, however increasing the number of clusters mean that we increase the complexity as well by introducing more segments. Elbow method is used to find the right spot for this tradeoff situation. The circled data points are the elbows of the curves, meaning that this is the k-value that gives relatively low SSE values and have a decent amount of complexity (number of clusters) to represent the data.

Figure 1 suggests that the optimal cluster value is 10 for the letter dataset whilst it has 26 labels and 5 clusters for the wine dataset whilst it has 10 labels. We see a trend here suggesting that a good representation of the dataset happens with clusters almost half the number of labels we have. I was expecting to have the same number of clusters and labels to see if each cluster was coinciding with a label. This will be experimented in the next section.

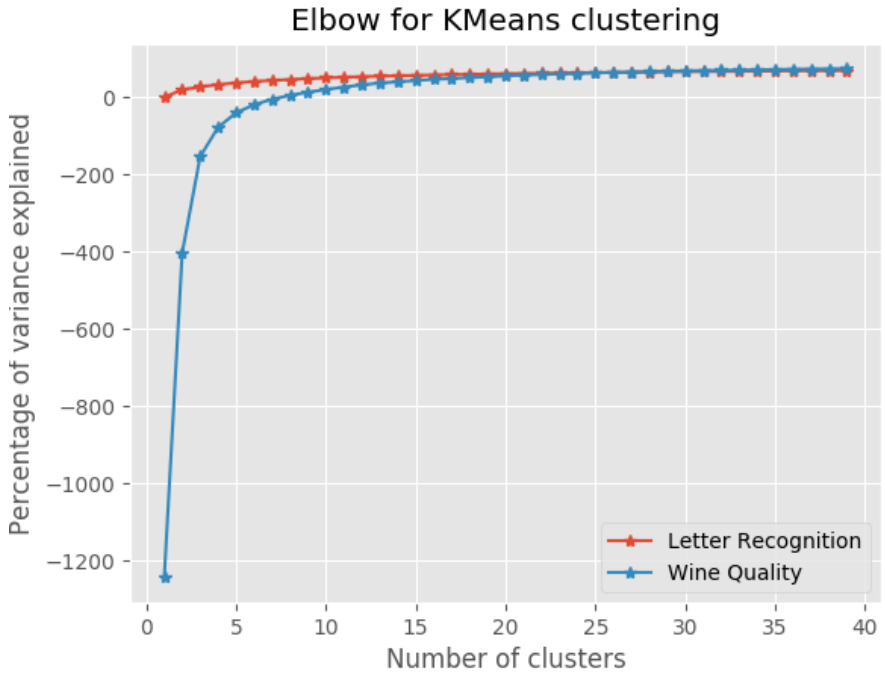

Figure 2: Variance in the distances of instances to clusters, given each K for both letter and wine datasets for K-Means Clustering.

Figure 2 shows the variance in the distances of instances to the cluster centers. We see a difference in both dataset's variances suggesting that the variance changes quickly for the wine dataset. This shows that the data in the Wine dataset are far away from other and with a low cluster number; everything seems far away from the cluster center. However, as the k increases, the variance goes to 0 for both datasets since after a point, the high number for clusters become very inclusive and cover all data points.

Expectation – Maximization

Expectation – Maximization (EM) is another clustering algorithm that is similar to k-mean clustering. In fact EM is a more general algorithm that KM is just a specific version of it. In EM, the values are assigned to the cluster centers with probabilities rather than being directly placed into a certain cluster. This allows instances to belong to different clusters at the same time with different weights. The result of EM is again a list of cluster centers that cover different areas of all the hyper-dimensions and the list of all the instances with their cluster that probably include them. The shapes of the clusters formed by KM and EM will be discussed in the later section with the necessary graphs.

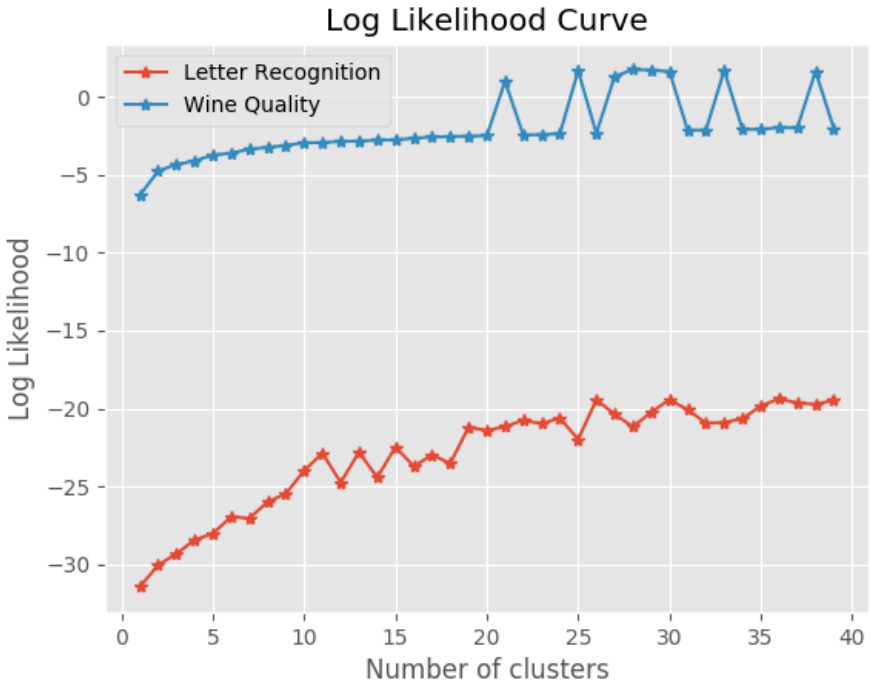

Figure 3: Log Likelihood values of increasing number of clusters for both datasets for the EM Clustering algorithm.

Figure 3 shows the average Log Likelihood values of instances for increasing number of clusters. Log likelihood is basically the natural logarithm of the maximum likelihood function. It describes how the instance is related to the cluster center. The more the log likelihood value is, the closer it is to a certain cluster center in this case. We would expect the log likelihood to increase as we get more and more clusters since more clusters means closer the instances will get to any of the centers. Looking at *figure 3*, we can see that this value is getting higher and slowing down towards the end.

Dimensionality Reduction & Effects of it on Clustering

In this section, I will be exploring different dimensionality reduction and feature selection algorithms to see if the datasets I use can be reduced to simpler forms that have less number of attributes without losing any information. In order to see the differences of these algorithms, I ran the algorithms on both datasets, and used KM and EM on each algorithm to see if reduction helps to the clustering process.

Principle Component Analysis

Principle Component Analysis (PCA) is a major dimension reduction algorithm that tries to reduce the trends in the data to eigenvectors that represent the major correlations of a given dataset. While doing it, the algorithm tries to find the eigenvector that has the highest variance in order to account for the maximum variability in the data to cover as much instances as possible. However, it cannot cover the entire spectrum usually and causes some degree of information loss.

Dataset	Dim. Removed	Number of Dimensions	Covariance
Letter	0	16	0
Letter	2	14	0.49
Letter	4	12	3.33
Letter	6	10	6.21
Letter	8	8	10.92
Letter	10	6	18.62
Letter	12	4	29.92
Letter	14	2	48.10
Wine	0	11	0
Wine	2	9	0.0003
Wine	4	7	0.026
Wine	6	5	0.078
Wine	9	2	19.84

Table 1: Noise Covariance values for newly formed dataset after reduced by the PCA algorithm to different dimensions.

Looking at *table 1*, we see that noise covariance values after the each reduction done by the PCA. Covariance basically explains that how dimensions are related to each other. If two eigenvalues are orthogonal to each other, the covariance will be 0, so in the cases that the datasets are not reduced to lower dimensions, the covariance values are 0, meaning that all eigenvalues that explain each hyper dimension are accepted as orthogonal to each other. As the number of dimensions decrease due to PCA's dimension reduction, we see that covariance increases, suggesting that the eigenvalues formed are more and more dependent, or in other words, related to each other.

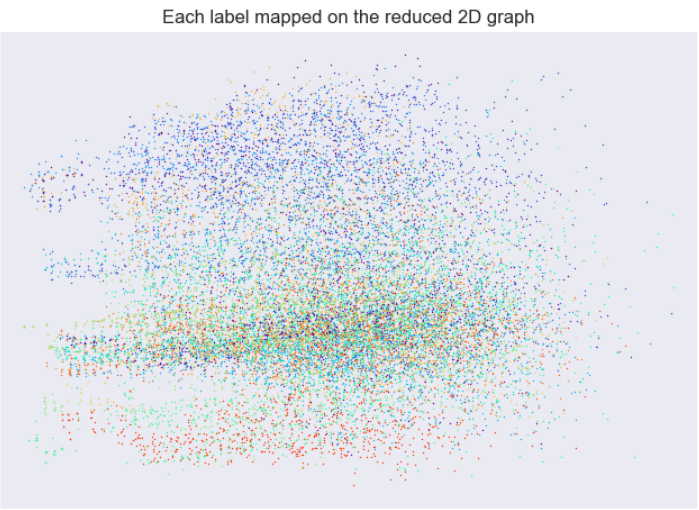

Figure 4: All letters of the letter algorithm represented on a graph after reduced to 2 dimensions by PCA. Each color belongs to a label.

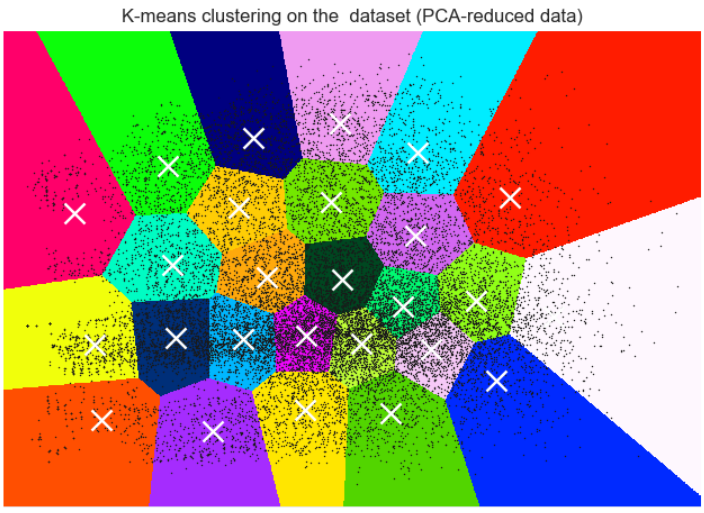

Figure 5: All clusters made by KM after reduced to 2D are shown with white crosses as cluster centers for the letter dataset. Black dots are instances.

Figures 4 and 5 show the experiments I ran in order to understand if the clusters and labels match when the cluster number is set to the label number. The data belongs to the letter dataset and contains 26 labels and 26 clusters (for each letter in alphabet). *Figure 4* shows all the instances, x and y values are the 2 dimensions that the 16 attributes reduced to by PCA, and each instances is

colored for its label. *Figure 5* shows the k-means clustering applied on the reduced dataset and each color on the plot shows a different cluster. Even though some cluster colors look similar or the same, each region is a different cluster. The same data points are shown as black dots. When looked closely to *figure 4*, it can be seen that colors are grouped in almost horizontal ovals and most of the cover almost all x-axis but a small portion of the y-axis. However, in *figure 5*, clusters are circular or hexagonal regions that are evenly distributed over the graph. This experiment shows that when reduced by PCA, k-means clustering does not mimic the labels properly. Since k-means uses Euclidian distance as its distance function, it creates circular regions/clusters. However, clustering would need another distance function that would capture the shape of the given labels in *figure 4.*

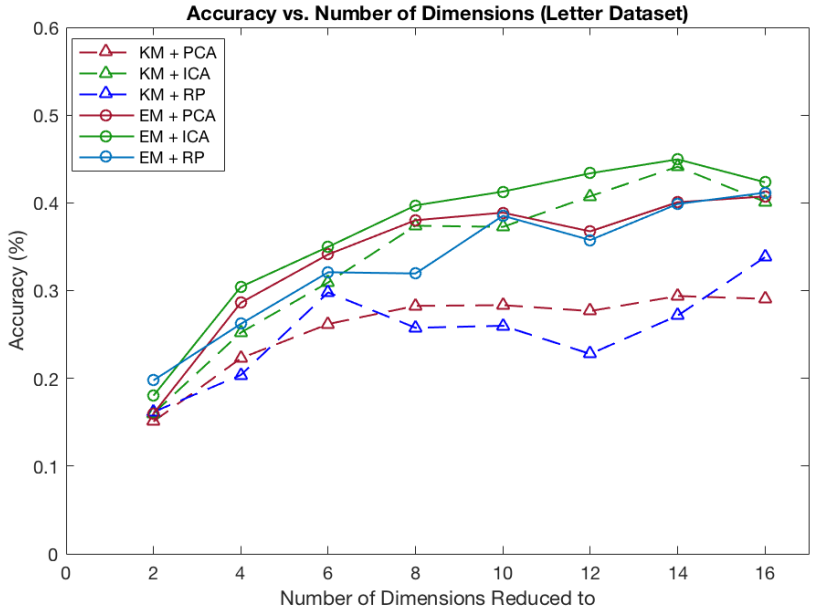

Figure 6: Accuracy values for each clustering and reduction algorithm combination for increasing number of dimensions for the letter dataset.

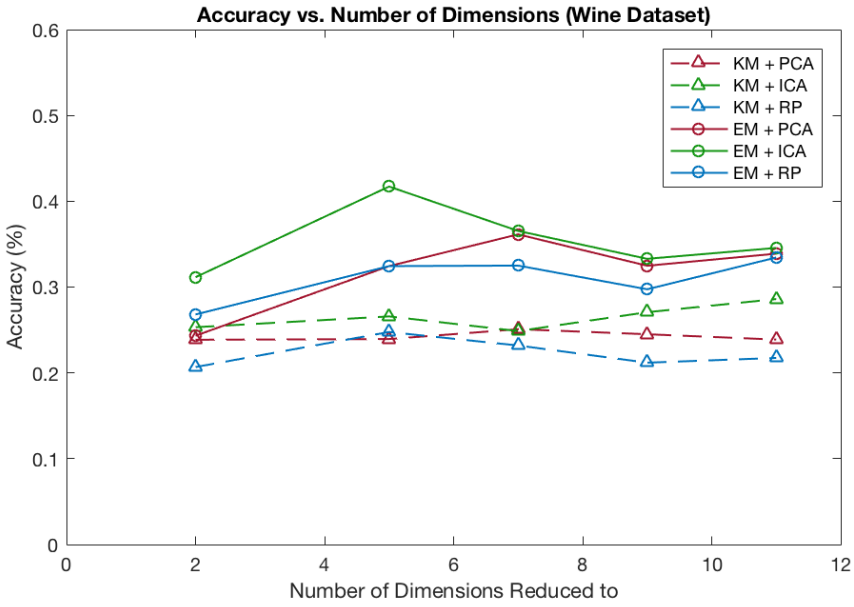

Figure 7: Accuracy values for each clustering and reduction algorithm combination for increasing number of dimensions for the wine dataset.

Figures 6 and 7 show the obtained accuracy values for all combinations of clustering and 3 dimension reduction algorithms. The accuracy values were found by matching clusters to most likely labels and dividing the number of assumed-rightly classified instances by total number of instances. Even though this is not the most accurate way to find the accuracy, the relative values we get out of helps us to understand the behaviors of the dimension reduction. In this case, looking at the plots, we see that PCA performs with 30% accuracy for both datasets.

Looking at *figure 6*, we see that with lower number of dimensions, we get a lower accuracy value for almost all algorithms. This suggests us that reducing the dimensions causes an information loss in the letter dataset. This means that when reduced the lower dimensions, some attributes lose their descriptive features, suggesting that most of the attributes for the letter dataset are important and directly correlated to the label itself.

Looking at *figure 7*, we see a different situation. The accuracy value stays constant or similar as the number of dimensions increase for all algorithm and clustering combinations. This shows that some attributes are not that related when it comes to explain the label and dimension reduction algorithms do a good job of finding the right eigenvectors to capture the trends to describe the entire dataset. Using the domain knowledge we have, we know that the labels of the wine dataset were given without regarding the attributes and this graph proves this.

Independent Component Analysis (ICA)

Independent Component Analysis is another dimension reduction algorithm that tries to reduce the data into meaningful linearly independent vectors just like PCA. However, while PCA was trying to cover the best variability, ICA tries to find the independent vectors that explain each trend or signal in the given data.

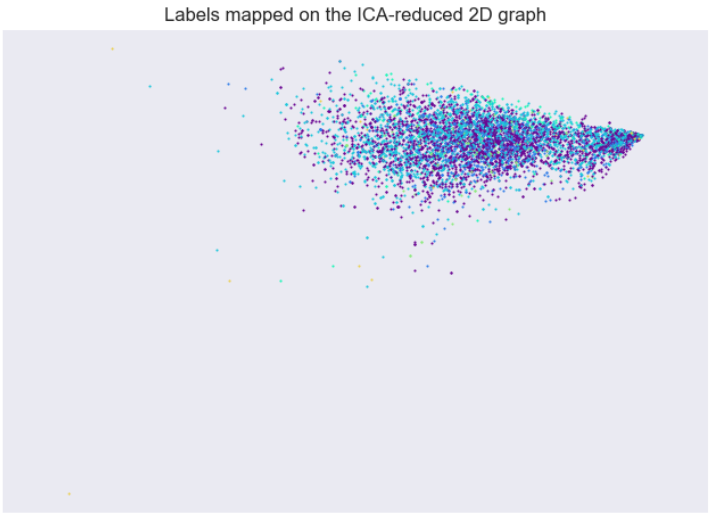

Figure 8: Each wine ranking (label) shown with a different color on the graph that is 2D reduced by the ICA algorithm

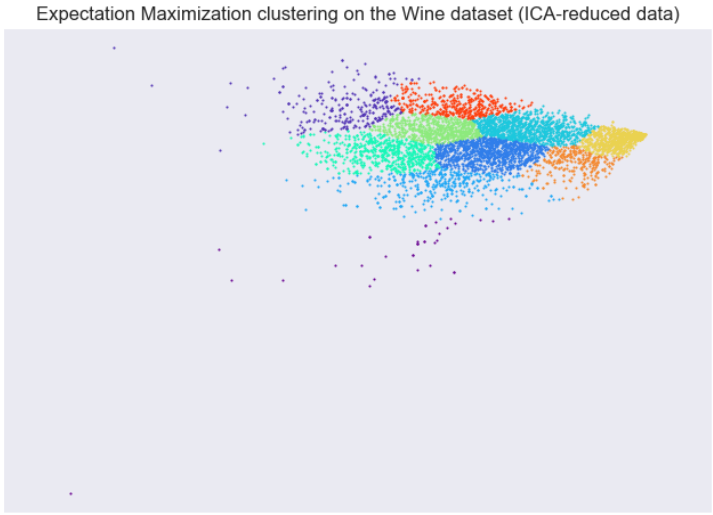

Figure 9: Wine dataset reduced to 2D by ICA and clustered into 10 groups by the Expectation Maximization algorithm show on graph

Figures 8 and 9 show how labels and clusters map into 2D space when the wine dataset is reduced to 2 dimensions by ICA. The first plot is the distribution of the labels of all instances and it can be seen that there is not a uniform distribution but some labels (5,6) dominate other labels and it's clear that the same labels are not clustered together. This is why the clustering shown in *figure 9* doesn't capture the labels well. The low accuracy (around 30% from *figure 8*) is a proof that this clustering is doesn't represent the labels. Lastly, I need to note that the shapes of the clusters here in EM are slightly different than the KM cluster regions by being more horizontal and narrower. This might be happening because of the differences in the distance functions both algorithms use.

Random Projections (RP)

Random projections is another dimension reduction algorithm that works a little different than PCA and ICA and in a sense, RP is a simpler algorithm. It works by trading a controlled amount of error (sacrificing small amount of info) for processing time. It simple gets rid of features and reshapes the data without losing the pairwise distances between instances to carry the relative differences to lower dimensions. In this experiment, I used the special case of Gaussian Random Projection. In *figure 6* and *7*, the effects of the RP dimension reducing can be seen. In both dataset, it manages to keep the similar levels of accuracy as the other reduction algorithms.

Variance Threshold – Feature Selection (VR)

For the last dimension reduction algorithm, I decided to use a very simple one, which is Variance Threshold algorithm. It simply looks over the features and if the variance of a feature is lower than the given input threshold value, it removes it, thus reduces the dimensions. I was curious to see if such a simple algorithm would perform as well as other complex dimension reducing algorithms. *Table 2* below shows the accuracy values I have received. The only noticeable trait is EM works better compared to KM when the data is reduced.

	Accuracy			
	Letter Dataset		**Wine Dataset**	
Variance Threshold	**KM**	**EM**	**KM**	**EM**
0.05	28.9 %	37.2 %	25.1 %	34.2 %
0.10	28.4 %	42.5 %	24.7 %	42.2 %
0.15	28.6 %	38.5 %	24.2 %	29.5 %
0.20	29.0 %	39.8 %	24.0 %	30.6 %
0.25	30.2 %	38.1 %	24.9 %	28.1 %
0.30	30.2 %	36.9 %	24.0 %	44.6 %
0.35	29.6 %	38.5 %	25.2 %	29.7 %
0.40	28.7 %	36.5 %	24.2 %	30.0 %
0.45	29.2 %	40.2 %	24.6 %	43.6 %
0.50	29.6 %	41.5 %	23.9 %	42.2 %

Table 2: Accuracy values for Variance Threshold for both clustering algorithms on bot datasets.

Neural Net Tests

In order to test the results of the clustering and dimensionality reduction algorithms, I prepared two tests that runs reduced and clustered new datasets on neural nets and compare them to the first assignment where I ran the neural net on the letter dataset without reducing the data or adding new attributes I took from clustering. I was able to use only 3 (PCA, ICA, RP) reduction algorithms only since Variance Threshold algorithm didn't reduce the dataset to the number of features I wanted but it was dependent on the features and their variances itself.

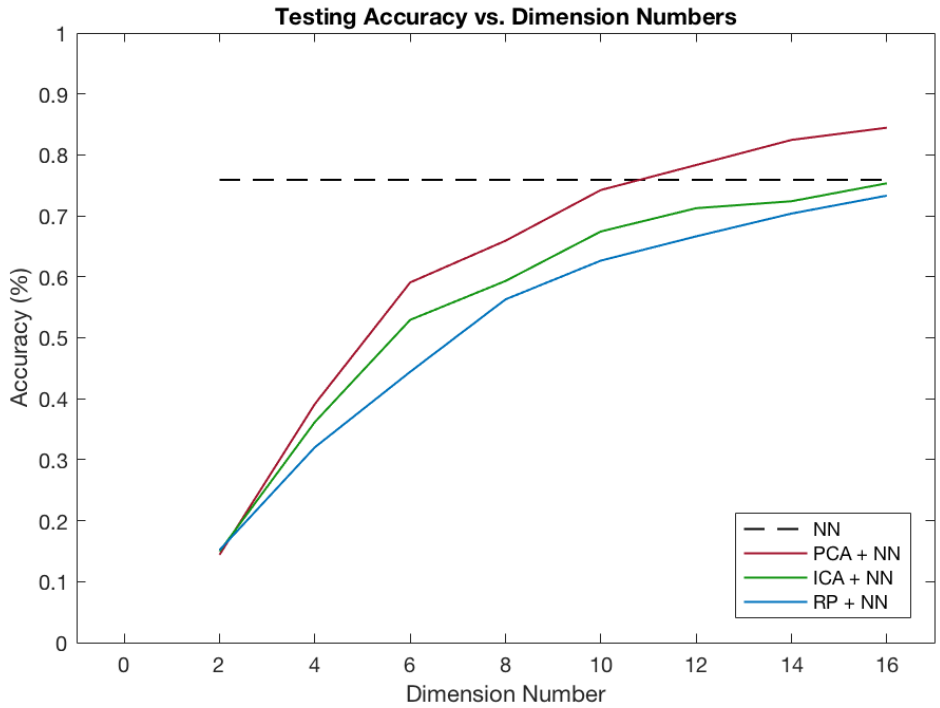

Figure 10: Neural Net accuracy plot where each curve belongs to reduced datasets by different algorithms.

Figure 10 shows us the neural net results for the letter dataset and its reduced versions. The black dashed line is the baseline that I have measured in the first assignment. With 0 hidden layers and 16 perceptrons, the neural net was performing with 75.9% accuracy. On the x-axis of the graph, we see the increasing dimension numbers and it seems like at 16 dimensions all of the algorithms (not reduced but still optimized in eigenvectors) except PCA converge to the same value. The reason for that is, the attributes of this dataset are so important and characteristic for the label that reducing it causes losses in information. However, when the dimension number gets closer to the actual feature number, the difference in loss decreases. Slightly higher PCA result tells us that PCA manages to come up with better eigenvectors to represent the data that it manages to get rid of some possible noise found in the dataset. Getting rid of this noise increases the accuracy in the end.

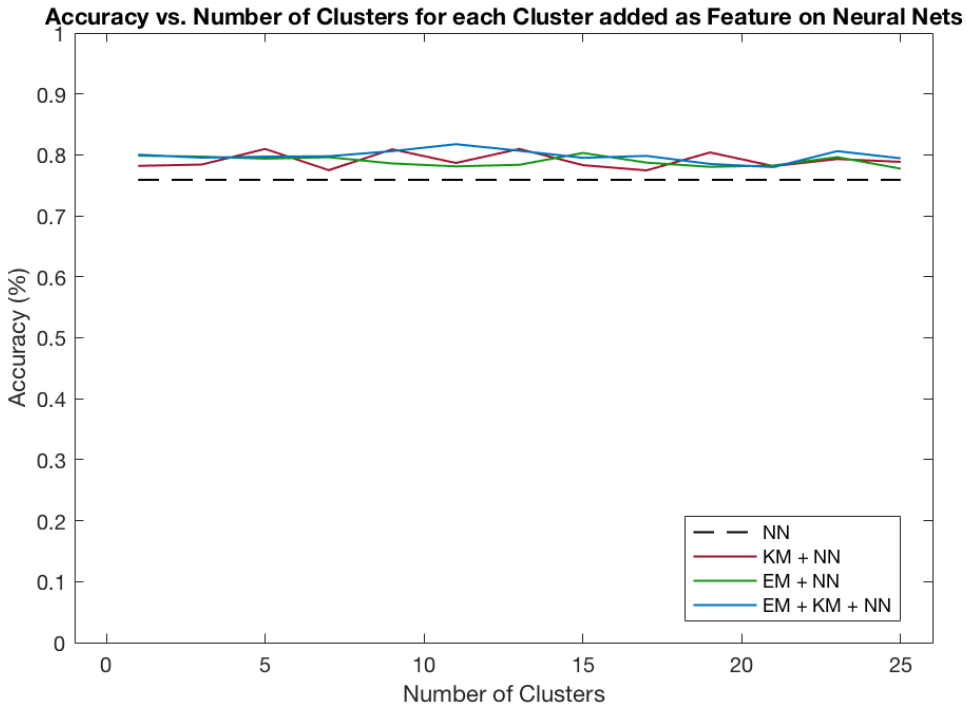

Figure 11: Neural Net accuracy plot where each curve belongs to reduced datasets by different algorithms.

Figure 11 shows the neural net results for the letter dataset. Each curve belongs to a different version of the dataset that has an extra feature that contains the clusters they belong to by various clustering algorithms (KM, EM and both). The results don't really show which clustering algorithm superior since for difference clustering values, the highest accuracy belongs to difference versions. However, we see that all these new versions of the datasets are doing better than the baseline. This can be explained with the nature of the clusters. Even though in the previous sections that we found the clusters and labels don't coincide perfectly; clusters still suggest hints about the label. Adding which cluster the instant belongs to as an extra feature to it, I was able to make the neural net to increase its accuracy slightly.