Assignment 3: Unsupervised Learning and Dimensionality Reduction

Introduction

In this report we explore results of running multiple dimensionality reduction and clustering algorithms on two different datasets. In the first part of this report, the original datasets will be run through dimensionality reduction algorithms to create new datasets. The algorithms explored are Principle Component Analysis (PCA), Independent Component Analysis (ICA), Random Projection (RP), and Random Forest Feature Selection (RF). In the Second part of this report, we will be looking at two types of clustering algorithms, K-means and Expectation Maximization (EM). We will use these algorithms to cluster the original dataset as well as the dimensionally reduced datasets and compare results. In the third part of this report, neural networks are implemented to determine and compare the results of the dimensionality reduction algorithms and the clustering algorithms in a classification scenario.

Datasets

Digits - 1797 instances, 64 features, 10 classes

Each instance in this dataset is an 8x8 image depicting a handwritten image. Each feature represents a pixel in an 8x8 grid and the value corresponds to the grayscale value of each pixel.

This is an interesting dataset to use for experiments involving dimensionality reduction because there are some pixels in the corners of the grid, for instance, that are rarely used and do not provide much useful information creating the possibility to produce a more compact feature space through feature transformation and selection. Relative to the other dataset used here, there are a greater number of features and it will be interesting to see how clustering and dimensionality reduction perform in comparison.

Segmentation – 2310 instances, 19 features, 7 classes

Each instance in this dataset represents a 3x3 pixel segment of a randomly selected outdoor picture. The features are continuous values containing information about the visual characteristics of the segment. The 7 classes correspond to what is depicted in the segment.

This dataset is interesting because it serves as a contrast to the digits dataset. It will be interesting to see how clustering and dimensionality reduction perform in comparison given it has fewer features to work with and fewer classes.

Part 1 – Dimensionality Reduction

In this section the original Digits and Segmentation datasets are run through four dimensionality reduction algorithms. A main criterion for each algorithm is used to determine the best number of components, typically incorporating the elbow method. Neural networks were also run and serve to validate and, in difficult cases, inform the selection of the number of components/features.

Principal Component Analysis

PCA is a feature transformation algorithm that seeks to produce a new basis that succinctly captures the original data. It does this by finding components that correspond to eigenvectors that account for the maximum amount of variance in the covariance matrix of the data.

Results can be seen for the PCA implementation on both datasets in Figure 1 & Figure 2. Explained variance is the metric used to determine how many components to keep. Explained variance is how much variance in the data each component accounts for. The higher the explained variance, the more information is captured about the underlying dataset by the component. In the Digits implementation there are elbows at around 5 and 8 components but at this point the cumulative explained variance was fairly low suggesting that there is a fair amount of additional information of the dataset that is not captured. 20 components present a happy medium: there is an elbow and about 80% of cumulative variance is explained. The neural network results confirm that we have retained a good amount of information from the original dataset with a greater than 90% test accuracy. The Segmentation implementation we see that on only the first component we can explain more than 40% of the variance. There is an elbow in explained variance around 5 components but considering the number of features is already low we can opt to use 13 components which is able to explain nearly 100% of variance while removing 6 of 19 features! Neural network results show a greater than 90% accuracy at this point as well.

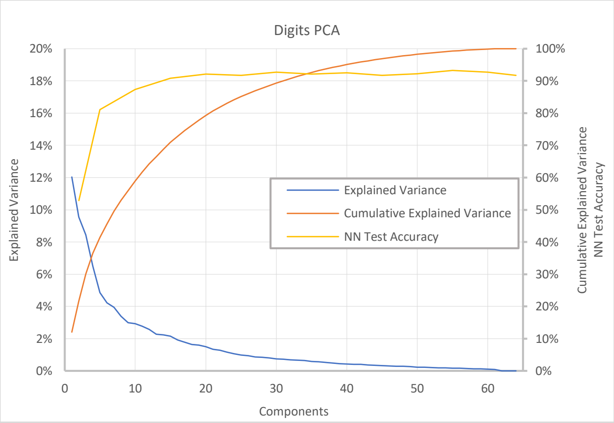

Figure 1: Results of PCA dimension reduction on Digits dataset by number of components

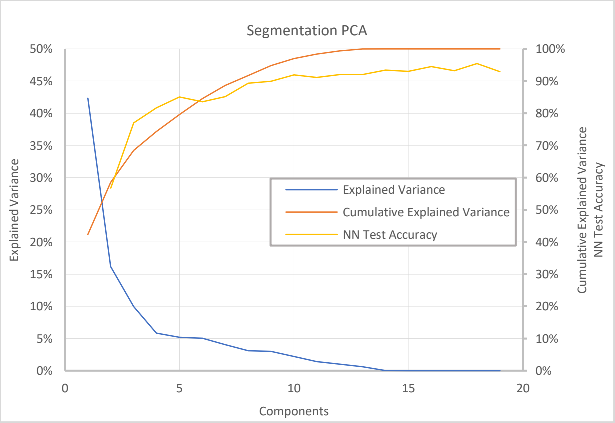

Figure 2: Results of PCA dimension reduction on Segmentation dataset by number of components

Independent Component Analysis

Like PCA, ICA also seeks to find a new set of basis vectors to better capture the data allowing for reduced dimensionality. ICA, however, attempts to find this new basis by finding vectors that are independent of each other. This is done by finding a vector that maximizes the non-gaussianity of the projected data.

Results can be seen for the ICA implementation on both datasets in Figure 3 & Figure 4. The metric used to choose the number of independent components to use is kurtosis. The plots here show the average kurtosis taken over the independent components. Greater values for kurtosis represent greater non-gaussianity and thus independence. For the Digits implementation we can see we have a local maximum for average kurtosis at 20 components. The decent performance of the neural network further confirms that this is a good number of components to keep. The Segmentation implementation has a peak kurtosis at 12 and if we choose 13 components we also see we have a high value for NN test accuracy.

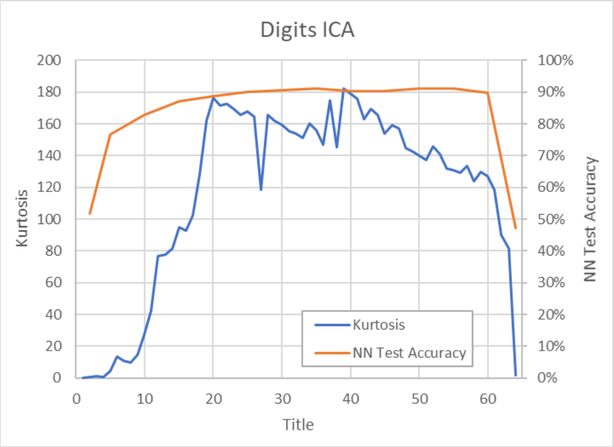

Figure 3: Results of ICA dimension reduction on Digits dataset by number of components

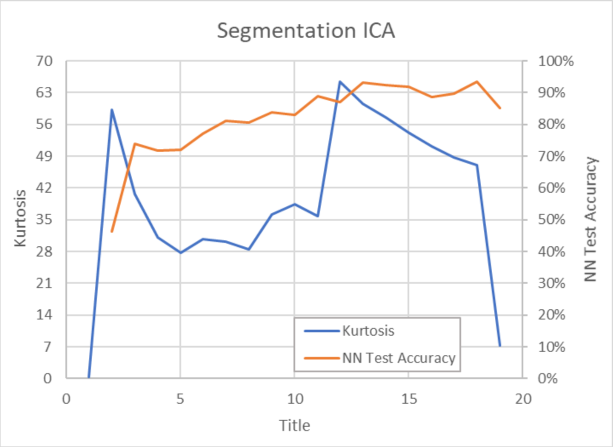

Figure 4: Results of ICA dimension reduction on Segmentation dataset by number of components

Random Projection

Like PCA and ICA, the RP algorithm also seeks to create a new set of basis vectors to project the data onto. Unlike PCA, however, RP generates random vectors rather than finding vectors that maximize variance. This can result in faster computation compared to PCA and comparable performance.

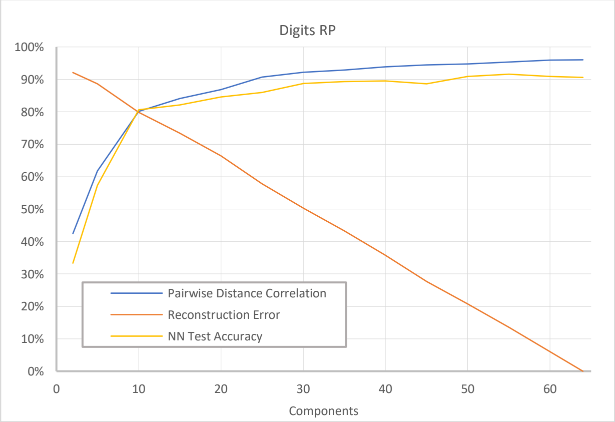

Figure 5: Results of RP dimension reduction on Digits dataset by number of components

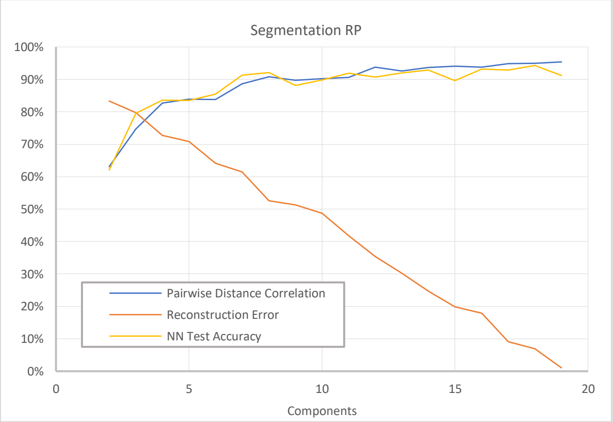

Figure 6: Results of RP dimension reduction on Segmentation dataset by number of components

Results can be seen for the RP implementation on both datasets in Figure 5 & Figure 6. Pairwise Distance Correlation and Reconstruction Error were the metrics used to evaluate the algorithm. These plots were produced by averaging results from 10 iterations of the RP algorithm given its stochastic nature. Pairwise Distance Correlation compares the Euclidian distances between instances in the transformed dataset with the distances in the original. A higher Pairwise Distance Correlation suggests that more information about the underlying dataset is maintained. Reconstruction Error is a measure of how much information from the original dataset is retained when projecting it to the random components. The results from the Digits implementation show a constantly decreasing reconstruction error as components increase that does not give a hint at how many components to keep. Pairwise Distance Correlation however sees a step up at 25 components and crosses the 90% threshold and the acceptable NN performance suggests this is a decent number of components to keep. The Segmentation implementation reconstruction error shows a prominent elbow at 8 components. At this point Pairwise Distance Correlation passes 90% and the NN accuracy is high, further suggesting this is a good number of components to keep.

Random Forest

Unlike the other dimensionality reduction algorithms explored here, RF is a form of feature selection that determines the most important features of a dataset to keep. This algorithm for feature selection is wrapped with a Random Forest classifier to inform feature selection. This classifier consists of many decision trees that split the data according to the measure of Gini impurity. Importance is assigned based on how greatly a feature reduces the impurity of a tree, with the algorithm returning an average importance for each feature across all trees.

Results can be seen for the RF implementation on both datasets in Figure 7 & Figure 8. The metric used to select the number of features to keep was feature importance. In the feature importance curve of the Digits results we can see a prominent elbow at around 8 and 35 features. Since the cumulative feature importance is only around 30% at 8 features, 35 appears to be a better option and the high NN test accuracy achieved at this number of features helps to confirm this. Conversely, the Segmentation results show a prominent elbow around 4 features and although the cumulative feature importance is only 50% at this number of features, the NN achieves a relatively high accuracy. For these reasons we will proceed with the 35 Digits features and 4 Segmentation features with the highest importance and discard the rest.

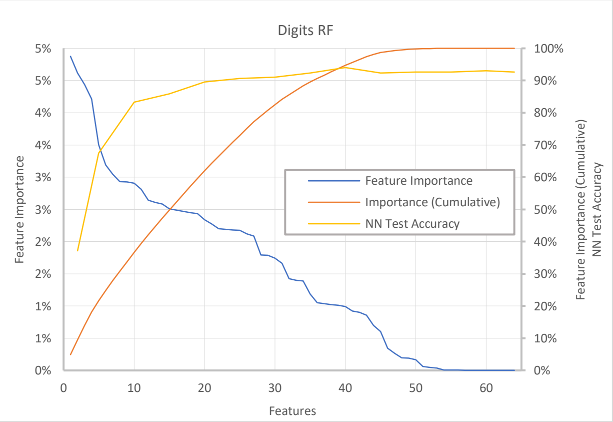

Figure 7: Results of RF dimension reduction on Digits dataset by number of features

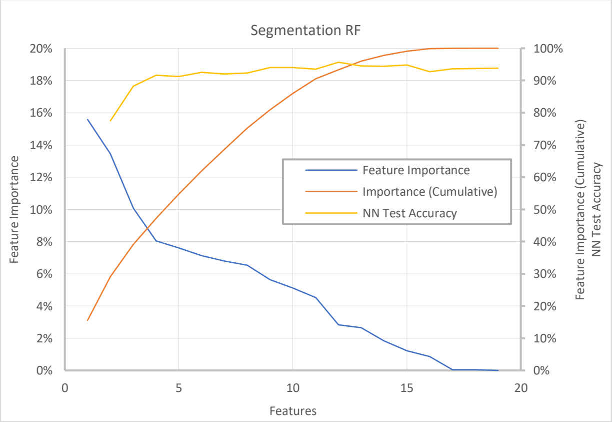

Figure 8: Results of RF dimension reduction on Segmentation dataset by number of features

Part 2 – Clustering

K-means and EM clustering algorithms are used in the following analyses. Similarity is determined by Euclidian distance. For K-means, the elbow method will be used on the sum of squared errors (SSE) metric to select the number of components and for EM, the minimum BIC value will be used. Homogeneity, Completeness and Adjusted Mutual Information metrics will be used to validate number of clusters selected. All these metrics compare cluster assignments to a ground truth (the actual class labels). Homogeneity is the measure of how much of a cluster contains members of the same class and conversely, completeness is the measure of how much of a class contains members of the same cluster. AMI is especially useful as it shows a combination of the information provided by the homogeneity and completeness metrics and it often gives us a maximum where the most amount of information is shared between the cluster and the classification labels.

Original Dataset

The results of running K-means and EM clustering algorithms on the original dataset can be seen in Figure 9 & Figure 10. In the Digits implementation of K-means we can see a slight elbow at 12 clusters suggesting this is a good number of clusters to have. The AMI plot peaks at 11 clusters and confirms that we have selected a good number of clusters. In the EM implementation BIC is lowest at 8 clusters but we can see we have a relatively low AMI at this value and 11 or 12 may have been a better choice. In the Segmentation implementation of K-means we see an elbow suggesting we select 6 clusters. AMI is at its peak here and confirms this to be a good choice. The EM implementation has a minimum BIC at around 13. AMI is relatively low here and we might be better off using about 8 clusters.

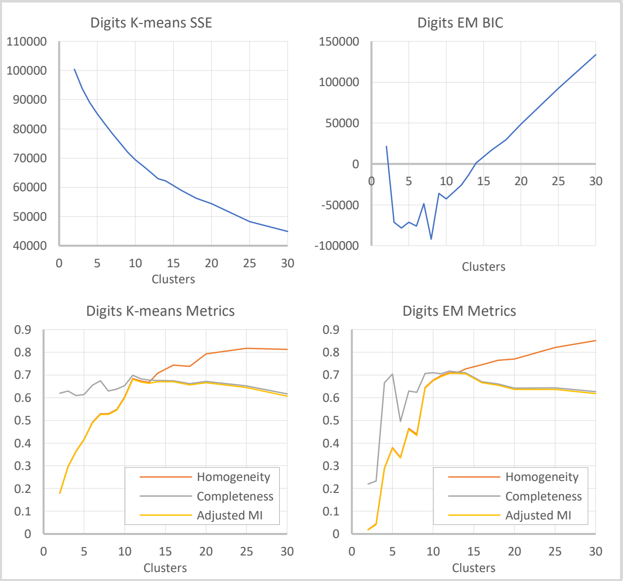

Figure 9: Results of clustering algorithms on PCA-reduced Digits dataset

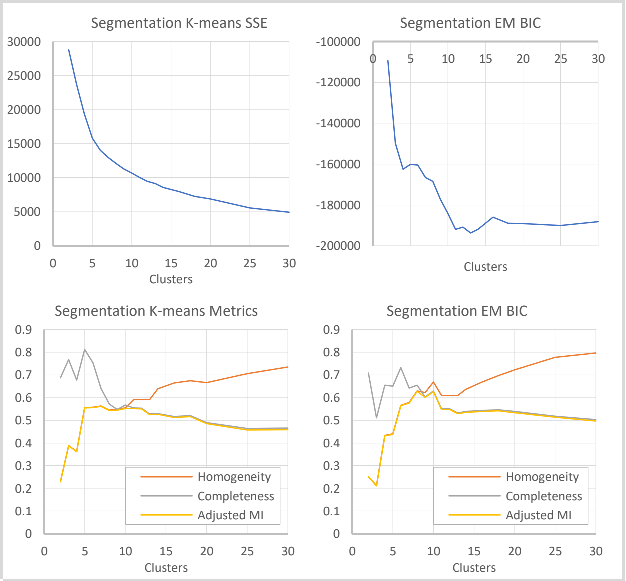

Figure 10: Results of clustering algorithms on PCA-reduced Segmentation dataset

Principal Component Analysis

The results of running K-means and EM clustering algorithms on the PCA reduced dataset with 13 components can be seen in Figure 11 & Figure 12. In the Digits implementation of K-means it is difficult to pick out an elbow though we see two slight ones at around 12 and 15 clusters. The peak in AMI suggests 15 is a good number of clusters. For the EM implementation BIC is at a minimum at 13. The AMI plot shows we are close to a peak at this number. The Segmentation implementation of K-means sees a slight elbow at 8 and AMI at this number of clusters is at its peak confirming this is a good choice for number of clusters. Finally, the Segmentation EM implementation shows a minimum BIC at 13 clusters, however we find that AMI is not near its peak here but is still relatively high.

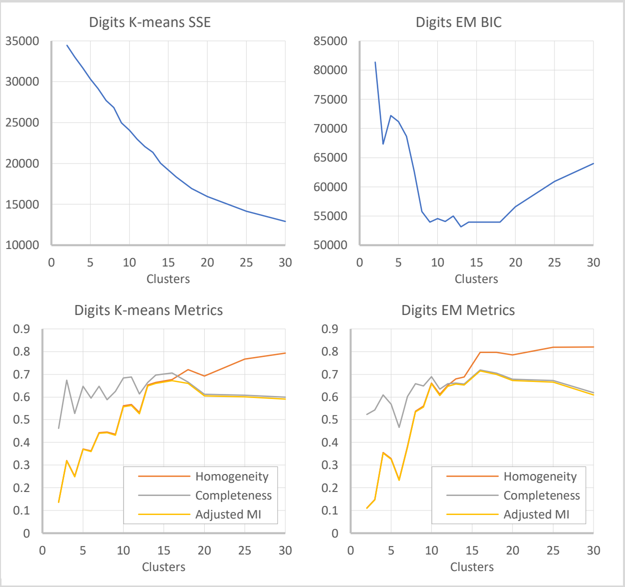

Figure 11: Results of clustering algorithms on PCA-reduced Digits dataset

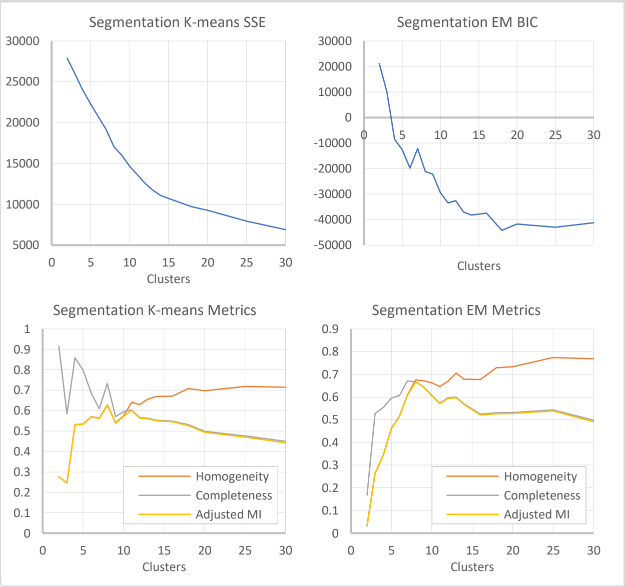

Figure 12: Results of clustering algorithms on PCA-reduced Segmentation dataset

Independent Component Analysis

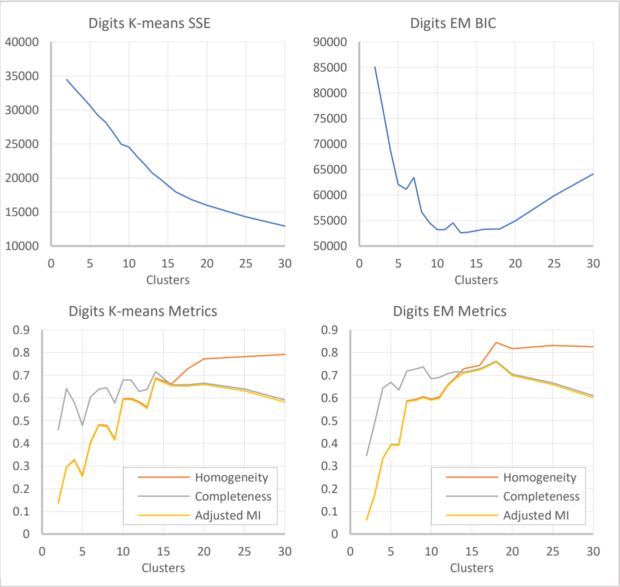

Figure 13: Results of clustering algorithms on ICA-reduced Digits dataset

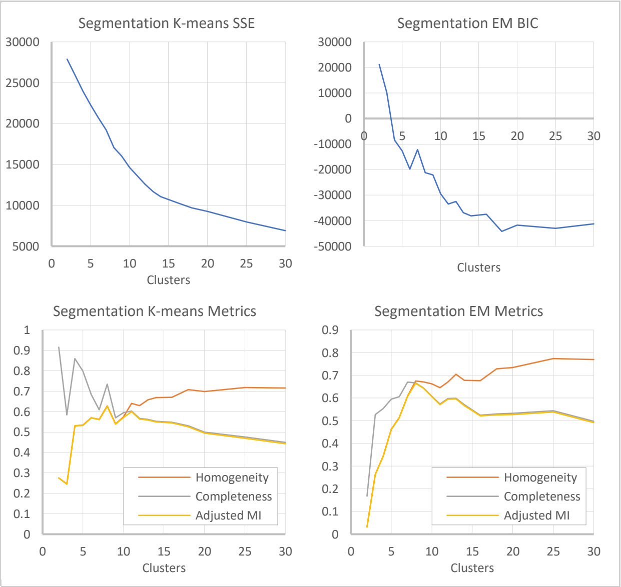

Figure 14: Results of clustering algorithms on ICA-reduced Segmentation dataset

The results of running K-means and EM clustering algorithms on the ICA reduced dataset with 13 components can be seen in Figure 13 & Figure 14. In the Digits implementation of K-means there is a slight elbow at 9 and another more prominent one around 16. AMI suggests 16 clusters is the preferred choice. The EM implementation of Digits shows a minimum BIC at 13. This is confirmed to be a good number of clusters as AMI is near its peak. Surprisingly, the Segmentation implementation for both clustering algorithms reveals identical results to the PCA implementation. The same number of components, 13, were selected for both algorithms in Part 1 and the raw data appears to be different. The explanation here is that both algorithms produced components that represent the same basis vectors. From this we can gather that, for this number of components, the vectors that capture the most variance are also the vectors that are most non-gaussian and thus most independent.

Random Projections

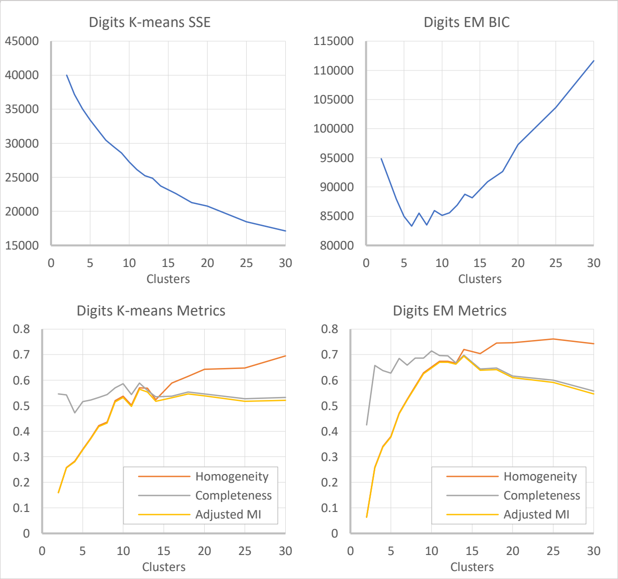

Figure 15: Results of clustering algorithms on RP-reduced Digits dataset

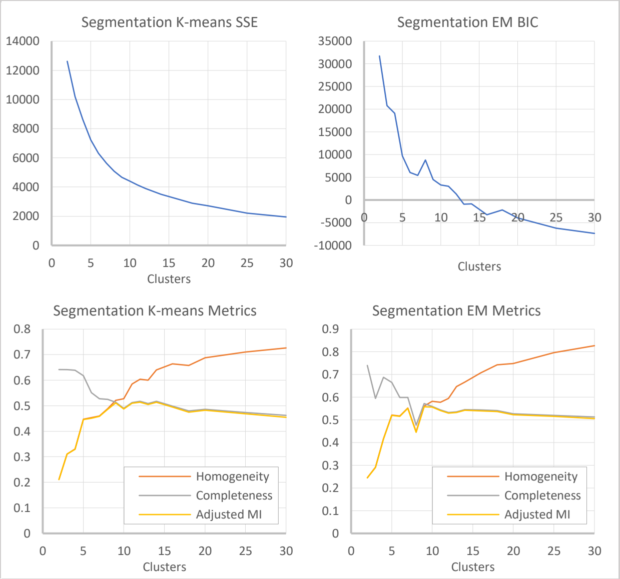

Figure 16: Results of clustering algorithms on RP-reduced Segmentation dataset

The results of running K-means and EM clustering algorithms on the RP reduced dataset with 8 components can be seen Figure 15 & Figure 16. The Digits implementation of K-means shows a relatively smooth SSE curve with a slight elbow at 8 and 11. AMI shows 11 clusters to be the better pick. The EM implementation of Digits has a minimum BIC at 8 clusters and AMI is decently high at this value. The Segmentation implementation of K-means shows an elbow at 7 clusters which the high AMI confirms to be a good choice. For the EM implementation of the Segmentation dataset, BIC is less for higher cluster numbers with a minimum at 30. This may be because the dimensionality reduction algorithm transformed the data in such a way that the data is clustered without clear boundaries resulting in more clusters. The AMI plot shows a relatively high AMI value here, though the peak is at 9 clusters.

Random Forest

The results of running K-means and EM clustering algorithms on the RF reduced datasets with 4 features can be seen in Figure 17 & Figure 18. The Digits implementation of K-means shows a slight elbow at 10 clusters. AMI is near its peak here and confirms this is a good number of clusters. The EM implementation of Digits has a minimum BIC at 13 clusters and AMI is near its peak here confirming this choice. The Segmentation implementation of K-means shows an elbow at 5 clusters which the high AMI confirms to be a good choice. For the EM implementation, again we see that BIC is generally decreasing for higher cluster numbers with a minimum at 30, perhaps for similar reasons as in the RP implementation. The plot shows lower relative AMI at 30 clusters compared to its peak at 7 clusters.

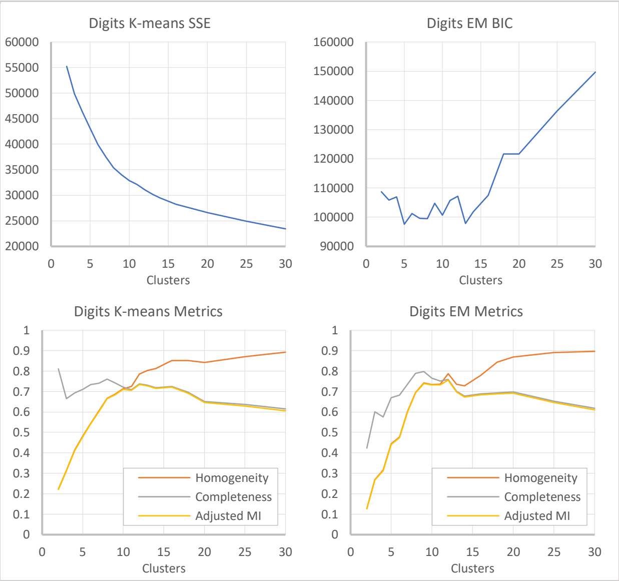

Figure 17: Results of clustering algorithms on RF-reduced Digits dataset

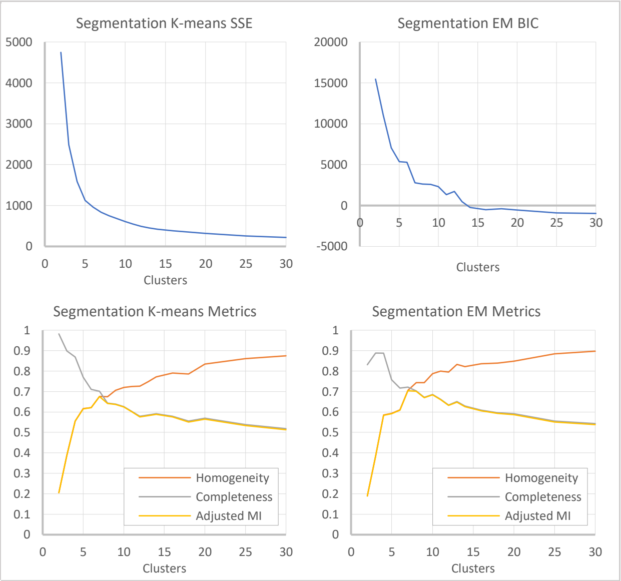

Figure 18: Results of clustering algorithms on RF-reduced Segmentation dataset

Clusters Comparison

In Table 1 we can see how the number of clusters chosen after dimensionality reduction compare to the number chosen on the original dataset. Given the Digits dataset has 10 classes and the Segmentation dataset has 7, we can see that in general that the number of clusters selected falls close to the number of classes.

	Digits		Segmentation	
	K-means	**EM**	**K-means**	**EM**
Original	12	8	6	13
PCA	15	13	8	13
ICA	16	13	8	13
RP	11	8	7	30
RF	10	13	5	30

Table 1: Number of clusters selected based on dimensionality reduction and clustering algorithms

Some exceptions are the K-means implementation Digits PCA and ICA datasets. We can see that a greater number of clusters is chosen here. This may be because the dimensionality reduction algorithms found components that split the dataset it to a greater number of more clearly separated clusters. Additionally, the EM implementation of Segmentation RF and RP datasets found a much larger number of clusters to be preferred. This was discussed in the earlier analyses.

Finally, a run-time analysis (plots not included here for brevity) showed mostly negligible advantages in time for clustering the original datasets vs the dimensionally reduced datasets. The time difference is likely to be more pronounced in datasets with much more features. K-means was found to be faster to implement than EM however across both datasets.

Part 3 – Neural Network Implementations

Datasets dimensionally reduced by the various algorithms seen in Part 1 were run through the neural networks across a range of selected components/features. After this the K-means and EM clustering algorithms were run on dimensionally reduced datasets that were created based on the results of Part 1 for a range of numbers of clusters. The assigned cluster of each instance was tacked on as an additional feature and the new datasets were run through neural networks. As in Assignment 1, the neural networks were implemented over a range of parameters including learning rate and layer architecture. The neural networks that achieved the best accuracy are included in the plots here.

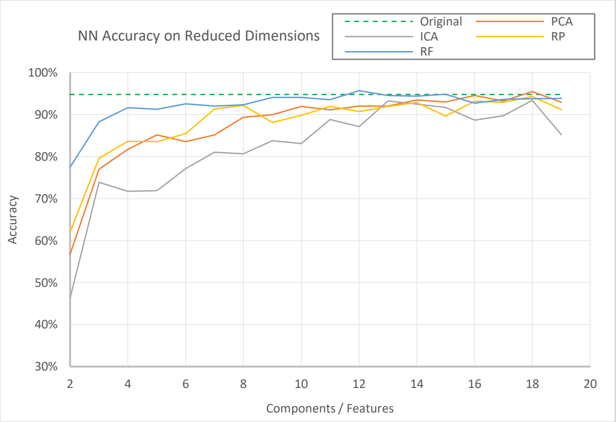

Figure 19: Neural network accuracy vs number of components/features in dimensionally reduced datasets

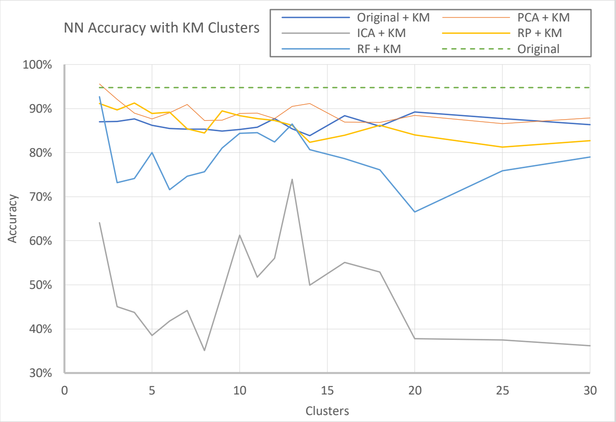

Figure 20: Neural network accuracy on dimensionally reduced dataset with K-means cluster feature added

Figure 19 shows that each of the dimensional reduction algorithms can achieve a high accuracy and, in some cases, beat the original dataset in NN accuracy, especially in higher numbers of selected components/features. We were able to strike a balance in selecting the number of components/features to keep in Part 1 for each algorithm that allows us to reduce the feature space while maintaining as much information as possible from the original dataset. As a result, 13 PCA components, 13 ICA components, 8 RP components and 4 RF features were selected each corresponding to high NN accuracy shown here.

In Figure 20 & Figure 21, we see the NN results from running the selected dimensionally reduced datasets with an addition feature corresponding to clusters assigned by K-means and EM algorithms. Surprisingly the results show that we struggle to achieve greater accuracy than the original dataset regardless of number of clusters. This may because of information from the original dataset being lost through dimensionality reduction and the clustering algorithms may have compounded this issue by using this incomplete data to assign less than ideal clusters. The other interesting finding here is that the neural networks benefit more by fewer clusters, with some of the highest accuracies found at only 2 clusters. We see this characteristic especially in the dimensionally reduced datasets and this again may be due to missing information compared to the original dataset. As a result, the NN prefers fewer clusters that generalize more. An analogy would be that the added cluster feature is only providing the NN a hint of what the true classification is rather than a firm answer, which allows the NN to consider the rest of the data more.

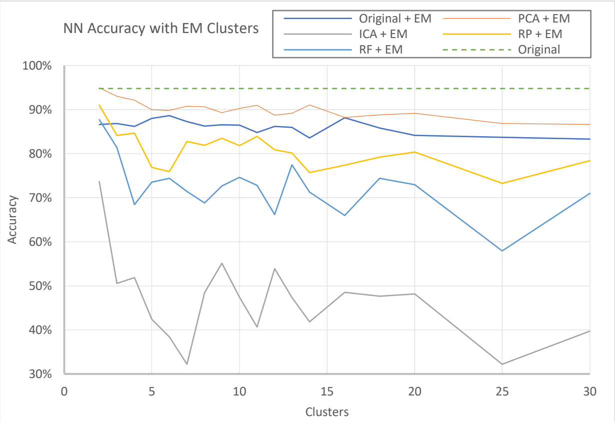

Figure 21: Neural network accuracy on dimensionally reduced dataset with EM cluster feature added

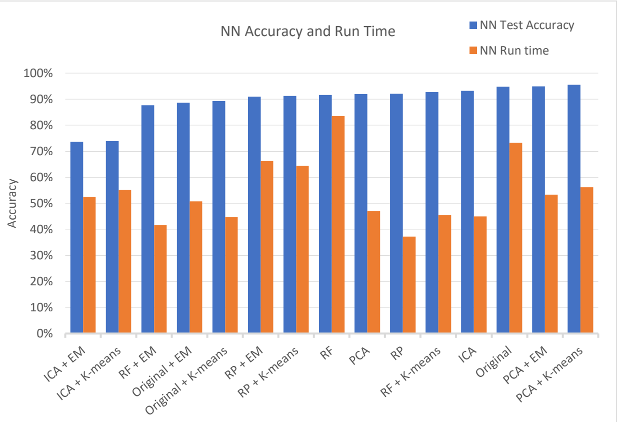

Figure 22: Summary of neural network experiments with accuracy and run-time (ascending by accuracy)

In Figure 22, we can see that between K-means and EM, both reach similar accuracies on the same dimensionally reduced dataset, but K-means appears to have a slight advantage. This could be because the potential clusters are well defined and negate the “soft clustering” advantage in EM. In addition, accuracies tend to be lower in datasets with the additional cluster feature when compared to the datasets that have only been dimensionally reduced. This is surprising but may be explained by the fact that the assigned clusters may not line up with class labels well are misleading the NN. The exception to this is PCA, where the added feature helped it the neural network achieve greater accuracy than was found on the original dataset. In this case, for both clustering algorithms, two clusters provided the best accuracy. The high accuracy can be explained by the initially high PCA accuracy coupled with a helpful ‘hint’ (to use the earlier analogy) provided by the added cluster feature.

Of the dimensional reduction algorithms, PCA generally performs the best, with and without addition cluster features as exhibited by Figures 19, 20, & 21. As seen in Part 1, a large amount of variance in the Segmentation dataset can be explained with only a few components meaning most of the information from the original dataset is captured in the feature transformation and because of this the clusters are also relatively true to the actual classifications. Conversely, the ICA algorithm proved to do poorly, especially paired with clustering. This is surprising and may be due to there being a better number of components to keep from the ICA implementation and it appears that clustering only makes matters worse.

We see that in general the neural networks run with dimensionally reduced datasets have a lower run-time in comparison to the original dataset. This is to be expected since we have reduced the number of features and thus the number of weights the neural network must train. By that same logic, it is understandable that when the cluster feature was added to these datasets that the time generally increased slightly due to the one additional feature and corresponding additional weights to train.