CS 7641 Machine Learning
Assignment 3 - Unsupervised Learning
Liyue Hu (lhu81)

Abstract

This assignment explores various methods in unsupervised learning on two distinct datasets. It first compares two clustering algorithms, k-means clustering and Expectation Maximization (EM). Then it explores four dimensionality reduction techniques, Principal Component Analysis (PCA), Independent Component Analysis (ICA), Randomized Projection (RP) and Random Forest (RF), and applies these techniques in conjunction with clustering algorithms. Lastly, it explores using clustering as input feature to Neural Networks and assess the performance.

1. Datasets:

For this assignment, I have selected 2 distinct classification problems: predicating default on credit card and classifying handwritten digits (0-9).

Dataset 1 – Pen Digits Recognition Description (from Assignment 1)

The second dataset contains 10992 instances of information. It contains a collection of samples of handwritten numbers from 44 writers. There are 16 attributes consisting of 8 resampled (x_t, y_t) points, representing digits as constant length feature vectors. The final input attributes are integers in the range of 0 and 100. The classes (y) are the 10 digits between 0 and 9, each representing the corresponding digit.

Pattern recognition is an important field of machine learning that has high applicability in our daily lives. We are increasingly interacting with pen-based digital devices such as Tablets, computers, smart-phones, etc. Theories of user-centric design advocates for "invisible" interfaces as they provide better user experience. Handwriting recognition is an important aspect that will enable more "invisible" interfaces.

Dataset 2 – Vehicle Recognition

The second dataset I chose was classification of silhouette of four types of vehicle, using features extracted from the silhouette. The dataset contains 18 attributes and four labels (van, saab, bus and opel). There are 845 instances. This is an interesting problem since it aims to classify 3D objects within 2D image using shape feature extractors on the 2D silhouettes of the vehicles.

2. Implementation

2.1 Clustering

I implemented two types of clustering algorithm - K-means clustering and EM.

2.1.1 K-Means Clustering

K-Means clustering is an unsupervised learning algorithm that assigns all data points to k centroids based on similarities. The similarity/distance metric that is used for this

assignment is Euclidian. The algorithm ran iteratively where the centers are re-computed by averaging the clustered points until convergence. I ran the algorithm with k-values ranging from 1 to 40 for each dataset to find the optimal number of clusters. To determine the optimal k-value, I used a combination of the elbow method and silhouette method. The elbow method attempts to find the point of diminishing improvement in within cluster SSE. The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation), with higher score corresponding to better cluster.

Pen Digits

According to the silhouette method, see Figure 1, the optimal k-value should be 8, but that is lower than the number of classes so I picked 10, which was the next highest point. Using the elbow method as seen in Figure 2, the optimal value should be 10 or 15. The datasets have 10 different labels corresponding to the 10 different digits, and therefore a cluster number of 10 seems sensible as the optimal k value.

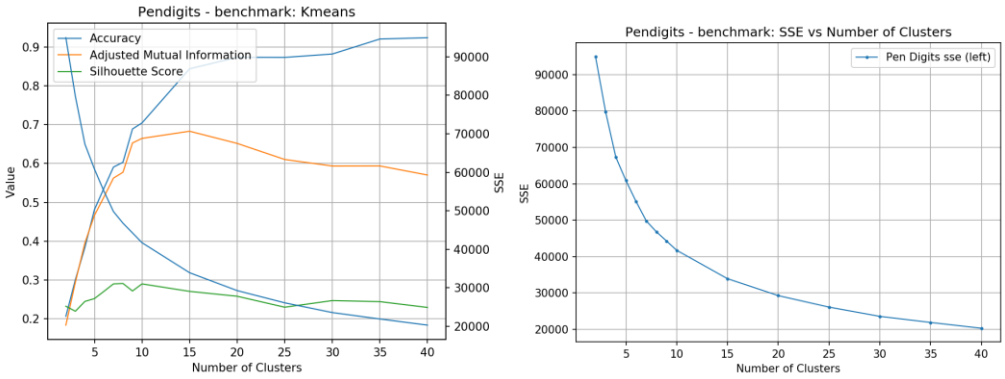


Figure 1 (left), 2 (right)

Vehicle Classification

I also used the same metrics to determine the k-value for the vehicle classification dataset. According to the silhouette method, see Figure 3, the best k-value is 2. Though this doesn't make sense given that there are 4 labels in the data, and for values above 4, k value of 4 has the highest silhouette score. Using the elbow rule, k-value should be 4 as that is where Figure 4 shows diminishing return. Although adjusted mutual information suggests a k value of 9, I decided to go with the consensus of 4.

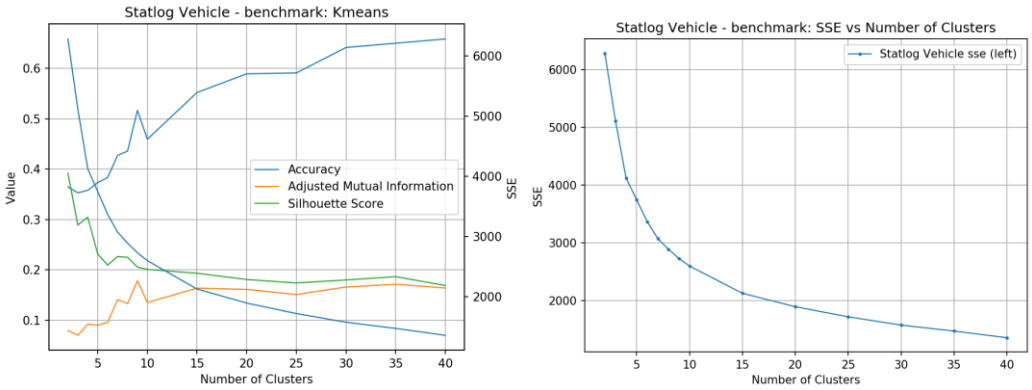


Figure 3 (left), 4 (right)

2.1.2 Expectation Maximization (EM, or GMM)

The EM method leans on probability to creates a probability density functions, and each point can probabilistically belong to multiple clusters. The algorithm tries to find a hypothesis that maximizes the probability of the data by iterating between the "expectation" (computing the soft clustering expectation given the probability distribution) and the maximization (computer the means of the soft clusters) steps. To determine how well similar data points are clustered, I looked at log likelihood, which shows how well the data fit the clusters they are assigned.

Pen Digits:

Looking at Figure 5 and 6 below, the silhouette score shows that the k value should be 2, however given that there are 10 labels, 10 would be the next best option. Looking at log likelihood, the value continues to increase as the number of cluster increases, however, using the elbow method, BIC increases more slowly after 20. The Adjusted MI graph also shows 15 as the highest point. Therefore, a k-value of 15 is selected for this algorithm.

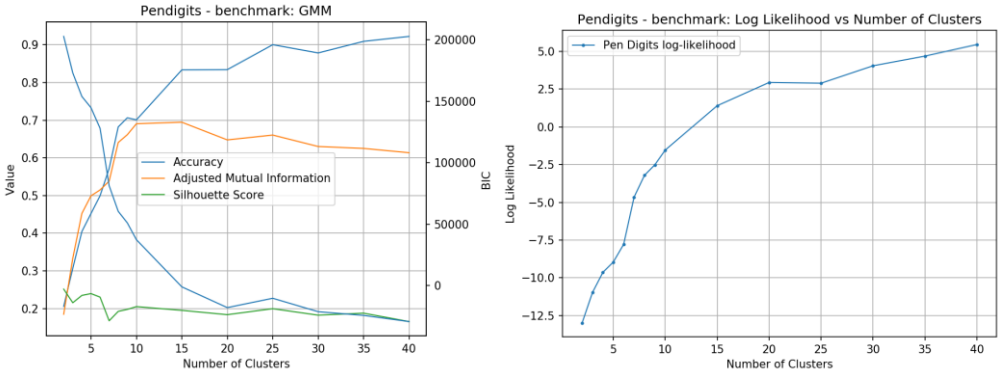


Figure 5 (left), 6 (right)

Vehicle Classification:

Using silhouette method, see Figure 7, although k value of 2 has the highest score, since it is less than the labels, 4 is chosen. Adjusted MI in Figure 8 suggest a k value of 5. Log-likelihood graph shows an elbow at k value of 4. Therefore, I chose k value of 4, which makes sense given that there are four classes.

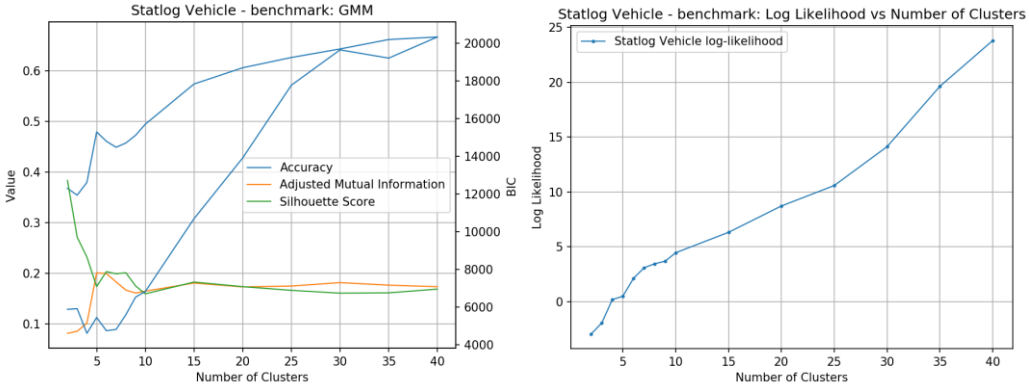


Figure 7 (left), 8 (right)

Performance Comparison:

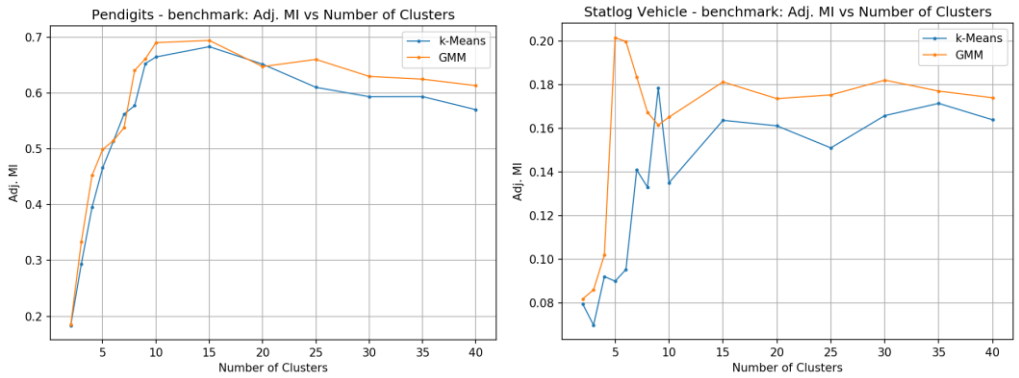


Figure 9 (left), Figure 10 (right)

For both datasets I explored, EM performed better than k-means, as shown in Figure 9 and Figure 10 above. For vehicle classification, EM had a lower optimal k-value than k-means, likely a reflection of the higher representational power of soft clusters. Overall, the vehicle classification problem seems to be more challenging for clustering than the pen digits problem, yielding significantly lower adjusted MI across all k-values.

2.2. Dimensionality Reduction & Re-Clustering:

In this section, I implemented four dimensionality reduction algorithms, PCA, ICA, RP and RF. Following which, the data were re-clustered with the two clustering algorithms (EM and K-means) and analyzed for comparison.

2.2.1 PCA:

PCA aims to maximize variance by performing orthogonal transformation, producing independent components called principal components. The goal is to retain the most important information and simplify the data for interpretability and insights, as well as reducing dimensionality.

After running PCA on the pen digits data, I found the knee value to be 3 components, representing ~65% of all explained variance. See Figure 11 below for the distribution of variance over the 16 components in descending order. It's clear that towards the end, the components add little additional info. As shown in Figure 12, for knee value for the vehicle classification data is 3 as well, as additional components after provides little additional variance.

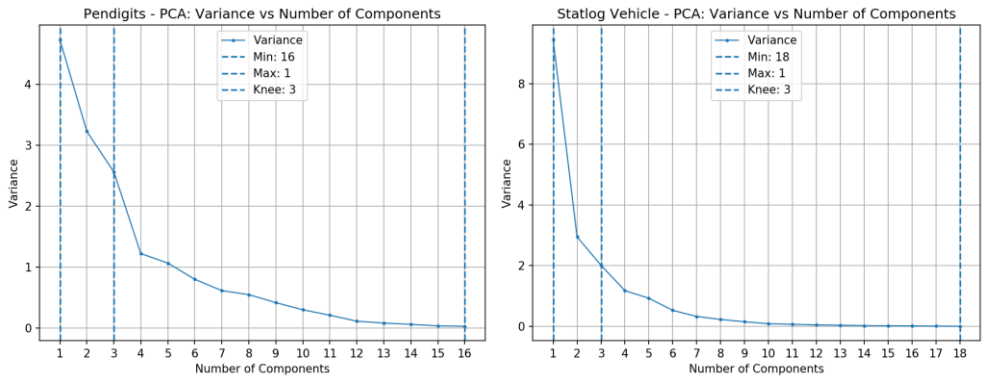
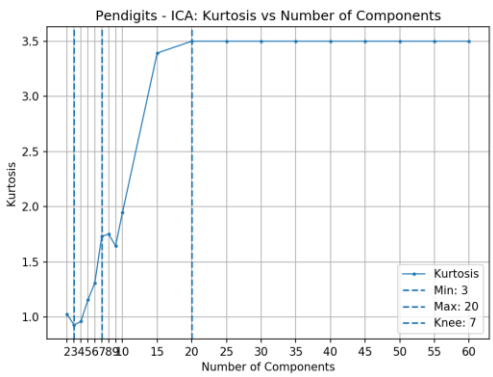


Figure 11 (left), Figure 12 (right)

2.2.1 Independent Component Analysis (ICA):

I then applied ICA to reconstruct a new feature space of independent components through linear transformation, to maximize non-Gaussianity.

For the pen digits recognition dataset, after applying ICA, the knee value by looking at kurtosis was 7. Max kurtosis occurs at 20 components, however that is higher than the original number of components, 16. For the vehicle classification problem, the max kurtosis is actually 3 components, while the knee value was 7.



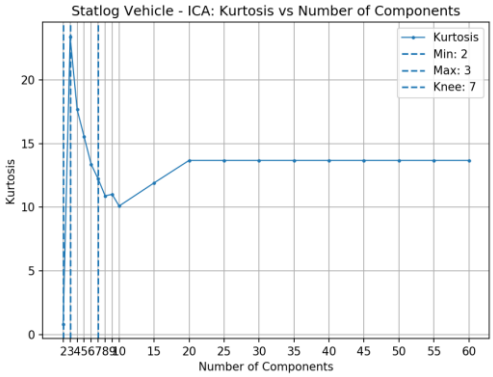


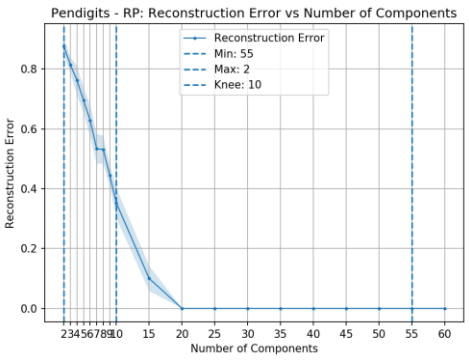
Figure 13 (left), Figure 14 (right)

Kurtosis for pen digits recognition is almost linear. The kurtosis for vehicle classification has a low max kurtosis, implying that there isn't really an underlying combination of distributions.

2.2.1 Randomized Projections (RP):

The third dimensionality reduction method I implemented was randomized projections, where directions are randomly generated and projects data out into those directions into a lower dimensional space. RP is known to work very well with classification as it deals with the curse of dimensionality problems.

As shown below in Figure 15 and Figure 16, the knee value is 2 for both problems.



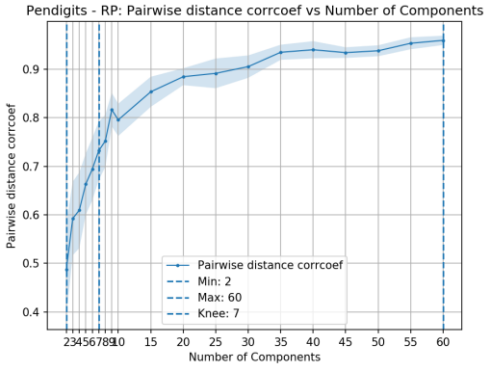


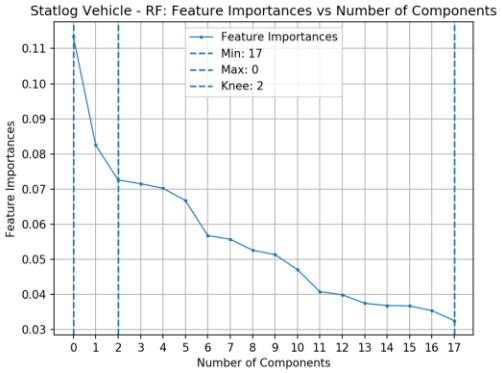
Figure 15 (left), Figure 16 (right)

Although RF is known to be not as efficient as PCA, since we are not projecting to the best direction, but it is also supposed to be faster than PCA.

2.2.1 Random Forest (RF):

The last dimensionality reduction I implemented was feature selection using feature importance in random forests. I compared both feature importance and reconstruction

error against number of components to find the knee component value. I used the values from feature importance in, 2 for both the vehicle classification and pen digits recognition.



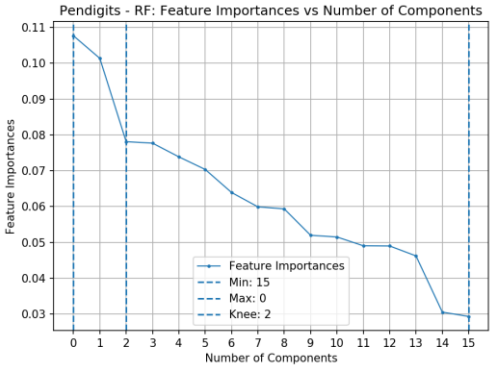
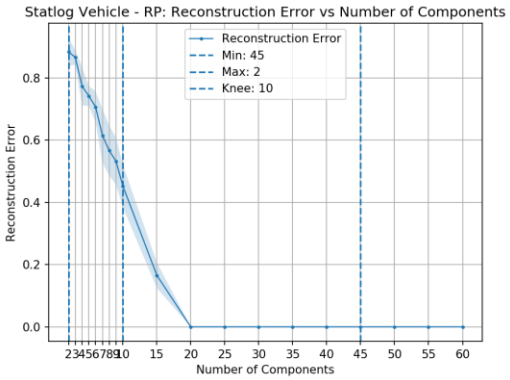


Figure 17 (left), Figure 18 (right)



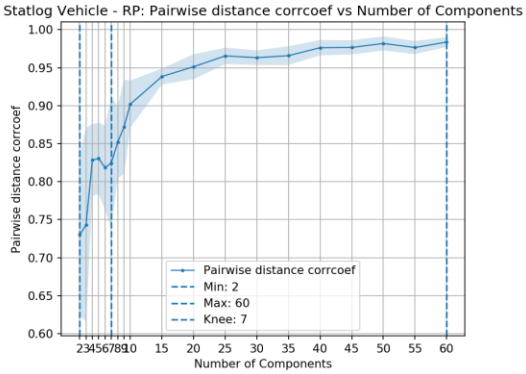


Figure 19 (left), Figure 20 (right)

2.2.2 Clustering Analysis:

Pen Digits Data:

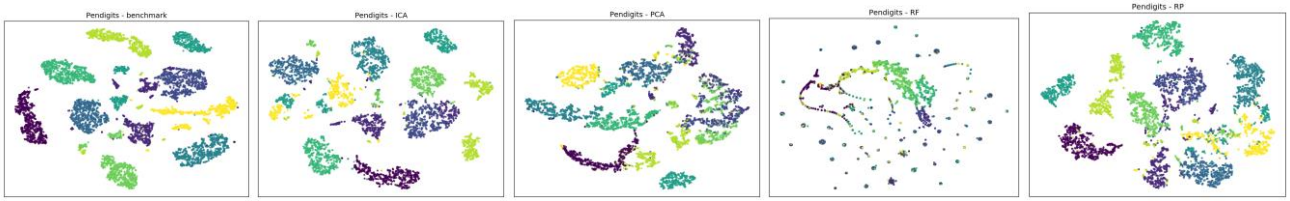


Figure 21 (Benchmark), 22 (ICA), 23 (PCA), 24(RF), 25 (RP)

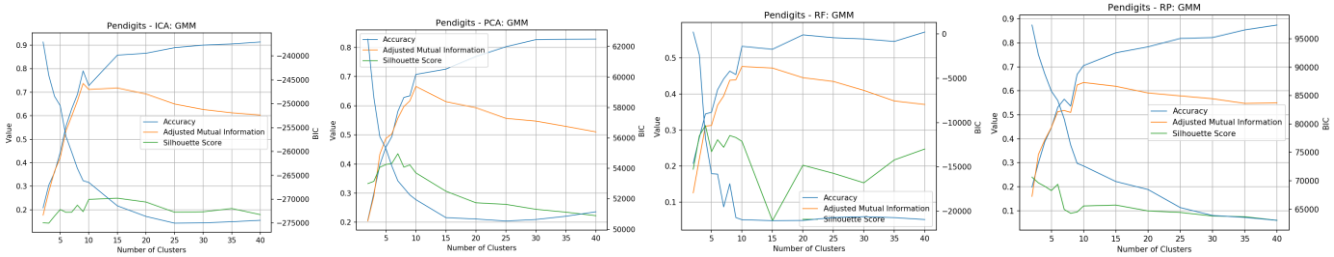


Figure 26 (ICA), 27 (PCA), 28 (RF), 29 (RP)

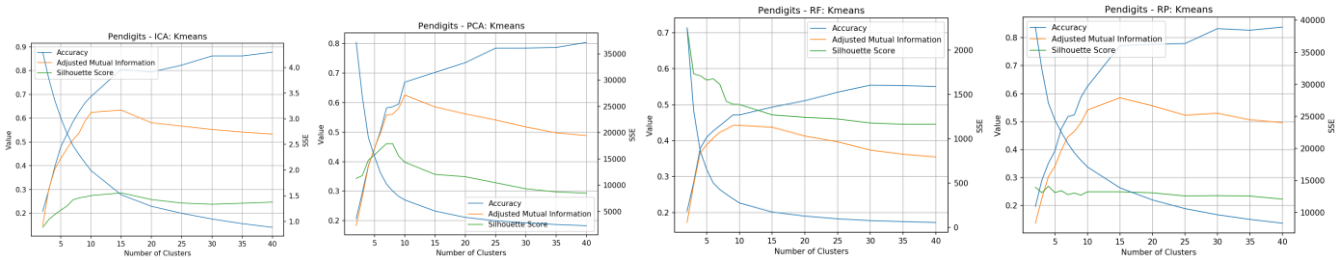


Figure 30 (ICA), 31 (PCA), 32 (RF), 33 (RP)

Clustering after the various DR methods all look fairly different. ICA, PCA and RP maintained similar clusters as before, while RF looked completely different, since it is a very different type of random transformation. RP was a little surprising in that it actually looked very similar to the other principal analysis transformations even though it is random. In terms of performance, RF had the worst accuracy across board, while the other ones had more similar accuracy. This agrees with the visual representations of the clusters where RF did not seem to create good clusters.

Vehicle Classification:

Figure 34 (Benchmark), 35 (ICA), 36 (PCA), 37 (RF), 38 (RP)

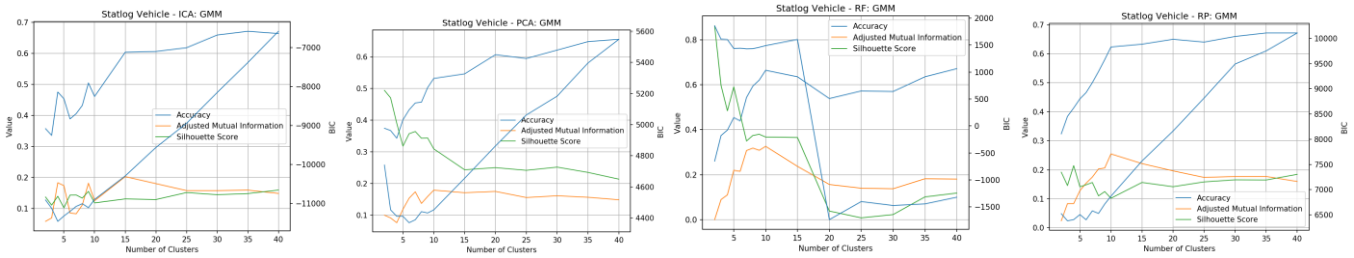


Figure 39 (ICA), 40 (PCA), 41 (RF), 42 (RP)

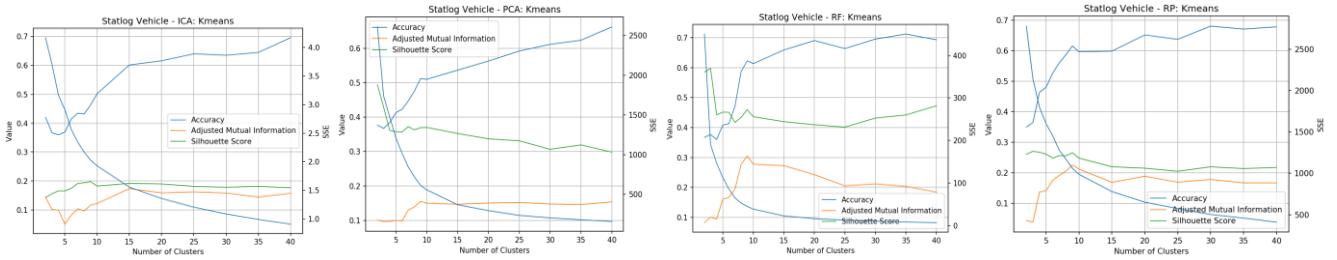


Figure 43 (ICA), 44 (PCA), 45 (RF), 46 (RP)

For this dataset, dimensionality reduction methods improved performance overall. Specifically, in contrast to the pen digits recognition problem, RF performed well when used in comparison with clustering. Vehicle classification is also seemed to be a harder problem since the data were not in very distinct clusters, and overall accuracy was fairly low. Looking at the Adjusted MI across the different graphs, the peak for both EM and K-

means seem to be closer to 10. In order to improve performance in the future, I can try running with larger k-values.

3.1 Dimensionality Reduction and Neural Network (NN)

	No DR	PCA	ICA	RP	RF
Test Score	0.99	0.86	0.94	0.94	0.56
Train Score	0.99	0.86	0.94	0.95	0.56
Fit Time	2.88	3.55	7.70	4.52	3.08
Train Time	0.006	0.006	0.005	0.006	0.010
Parameters		n=3	n=7	n=7	n=2

Table 1

According to Table 1 above, NN still performed the best without applying dimensionality reduction. As well, ICA is noticeably slower in terms of fit time versus the other algorithms. The best performing DR was actually RP, where projections were in random directions. This confirmed what I learned from the lecture video that even though RP does not project to the best directions, it still works remarkably well and is efficient (fast). RF performed the worst.

3.2 Clustering and NN

I got the following results while comparing the effect of using different clustering as an additional attribute to NN to NN without clustering. I ran the experiment on the Pen Digits recognition dataset. The number of clusters was set to the optimal k value of 10 for K-means and 15 for EM.

K-means	NN Benchmark	PCA	ICA	RP	RF
Fit Time	2.87876	2.98581	13.5849	7.83284	4.26379
Score Time	0.00600	0.00718	0.00917	0.00878	0.00778
Test Score	0.98974	0.82503	0.86011	0.84983	0.55679
Train Score	0.99665	0.82799	0.86483	0.86395	0.56286

Table 2

EM	NN Benchmark	PCA	ICA	RP	RF
Fit Time	2.87876	4.35933	4.38049	5.73313	3.86825

Score Time	0.00600	0.00997	0.01097	0.01276	0.01057
Test Score	0.98974	0.79516	0.86050	0.79780	0.54046
Train Score	0.99665	0.79639	0.86691	0.79863	0.54043

Table 3

As seen in tables above, NN performed the best without any dimensionality reduction or clustering. This is the case for both EM and K-means clustering as an additional feature. ICA was the slowest for K-means clustering, taking significant longer than the other algorithms for fit time. It performed the best out of the different DR methods but just marginally. RF is still the worst performing algorithm for this particular problem. Compared with performance without clustering (Table 1), fit time was considerably higher for most methods. For EM, ICA's fit time was not a lot higher than the other methods, and the performance is worse than the K-means method, showing a tradeoff between time and accuracy. However, the best performing method was still the benchmark NN method, indicating that for certain classification problems, dimensionality reduction and clustering algorithms are not appropriate or necessary steps in data pre-processing.

4. Conclusion

Through this assignment, I explored many methods within unsupervised learning. I was able to compare the two clustering algorithms (EM and K-means clustering), as well as the four dimensionality reduction methods (PCA, ICA, RF and RP), and finally combining clustering algorithm and dimensionality reduction with Neural Networks to assess the impact of pre-processing input data with unsupervised learning. The result I got were not as good as the NN on the raw data. This is the case for the pen digits problem that I explored. But for other problems, this may not be the case.

Reference:
https://www.linkedin.com/pulse/finding-optimal-number-clusters-k-means-through-elbow-asanka-perera/
https://archive.ics.uci.edu/ml/datasets/Statlog+(Vehicle+Silhouettes)