Unsupervised Learning Report

Datasets

The UCI Wine Quality Data set [4] contains physicochemical inputs which are to be used to classify the wine based on rating which describes quality of the wine. The dataset tested is the specifically wine white from the Portuguese “Vinho Verde” wine. The feature contains 11 attributes and a quality score from 0 to 11. This dataset was chosen to see the results the machine learning algorithms can have on a relatively small set (4898 samples). In addition, the data set is skewed towards the center, so it will allow for exploring the effect of biased data. The distribution is seen below. The data has been normalized to show all attributes cleanly.

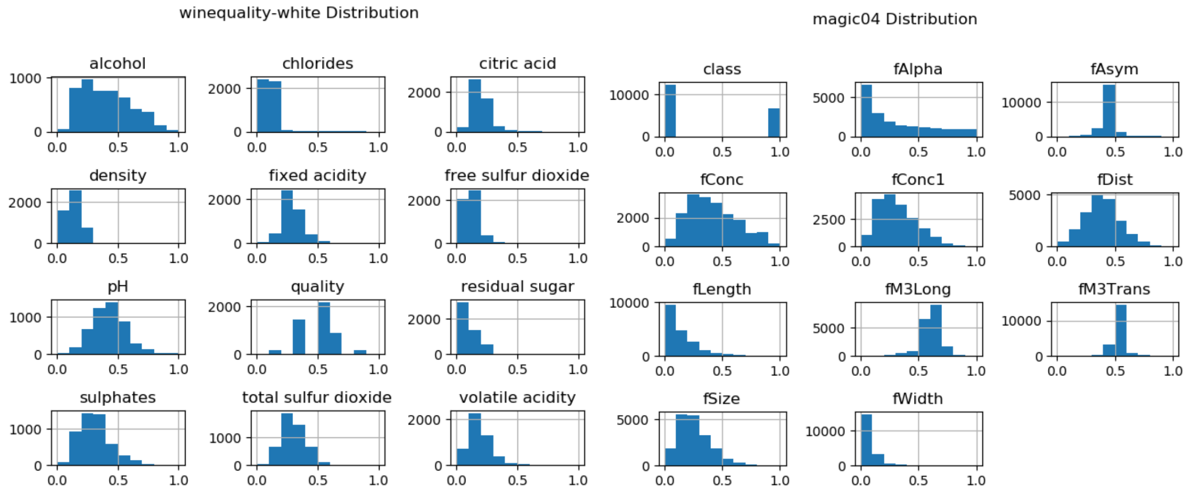


Figure1: Wine Quality Distribution(Left) and Magic Gamma Distribution (Right)

The UCI Magic Gamma Telescope Data set [3] contains data generated from a Monte Carlo program to simulate an atmospheric Cherenkov gamma telescope observing high energy gamma rays from the Earth. If the threshold is correct, the scope takes a pattern of Cherenkov photons called a shower image. Following, the image is put through some pre-processing to generate the 10-dimensional features in the dataset. This dataset offers a different situation for the classifiers as the output is Boolean: gamma or hadron. Additionally, the size of the dataset is an order of magnitude offering differing insights into the how the algorithms will operate on data given a wider range of training set sizes. Above is the distribution for the data. The data has been normalized to show all attributes cleanly. As can be seen, the output holds a much more even distribution, roughly 1 hadron to 2 gamma.

These two datasets together coupled together provides an interesting set of data. First, they are an order of magnitude separated allowing examination into the effect the volume of available training data has on the algorithms. Second, the data have very different spreads among the two sets. Wine being close bundled while magic gamma tends towards a wider spread. Third, wine quality classifies to 12 wine scores while magic gamma is Boolean. The effect the number of classes in respect to the complexity the various algorithms will be interesting. Finally, both datasets have similar number of attributes. This was chosen to introduce less free factors to muddle the comparison of the two datasets.

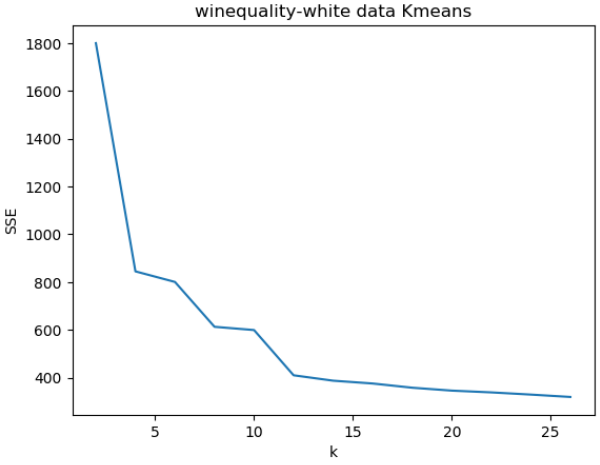
The clustering and dimension reduction steps were completed using the tool provided in weka [1] while the neural networks were created and tested using Sklearn [5].

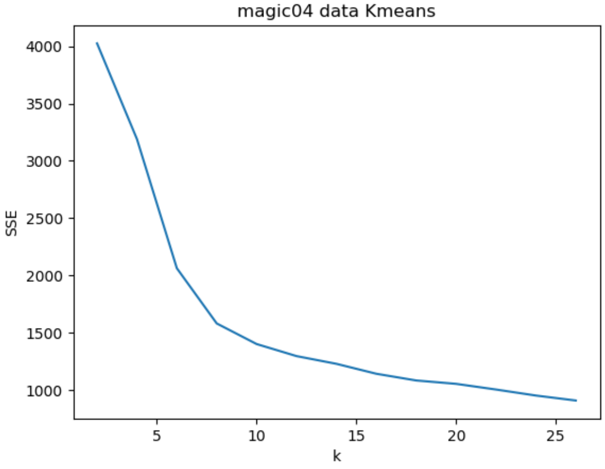
Clustering

Clustering is an unsupervised learning technique to attempt discover some underlaying structure of the data through the assignment of samples to clusters. These clusters in theory represent sets of similar data. The similarity of two samples is defined using distance metrics. Typical distance metrics are Euclidean and Manhattan distances. Euclidean was chosen as the distance metric to be used on the two datasets since the data of both sets contains continuous data which lends itself much more to Euclidean over Manhattan distances. Additionally, the data is well spread in all directions and tightly bunches as discussed in previous papers thus it would make sense to prefer Euclidean as movement is not tied along x-y axis.

In this paper, two methods of clustering will be examined: K-means and Expectation maximization.

K-means is a simple clustering algorithm where K centers are chosen. Each iteration all the closest points to each center are added to that center's cluster. New centers are calculated based on the clusters, and the process repeats. This continues until the centers stop moving. This algorithm was applied to the two datasets. Below are the graphs of standard square error of the k means algorithm run over various k values.





It is interesting to note that both follow similar trends of exponential decay. This makes sense because as k increases the number of points belonging to each cluster becomes smaller. Thus, each cluster becomes a collection of closer and closer-knit points eventually becoming a singular point with no error. This means the plateau being observed of decreasing SSE is due to overfitting. Indeed, this fits with a known method of picking k called the elbow method which selects a k around where the elbow of the curve is located as further increases in k come with marginal benefits in SSE [2]. Thus, k for wine quality should be around 9, and for magic04 is should be 7. Obviously the k for magic04 was far off with the 2 labels in the dataset; however, wine quality was much closer at 7 labels to a k of 9. . It is interesting to note a large difference between the two datasets in the smoothness of the curves. This could have a few likely causes. Magic04 is simply a larger and smoother set thus allowing less variation between k means while wine quality being smaller and less smooth allowed for more local optima for the k means centers to fall into while searching for the global.

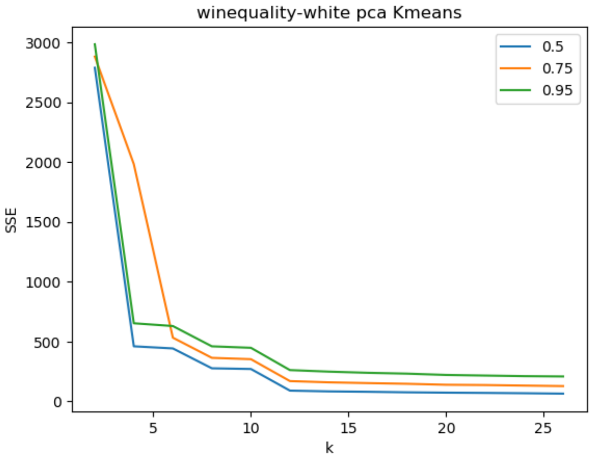
Expectation Maximization is another clustering algorithm. It is similar in purpose to k means, but in place of finding centers based on means, it calculates the probabilities each sample belongs to a cluster. It maximizes the likelihood of the overall assignments. This algorithm was applied to the dataset. Weka completes a cross-validation to choose the value of k [1]. The log likelihood found for the wine and magic04 datasets are -3.413 and -27.69 respectively. This produces a well clustered set of data within the attributes; however, the labels do not correlate to these clusters at all. This is not a huge surprise since these datasets were very difficult for supervised learning.

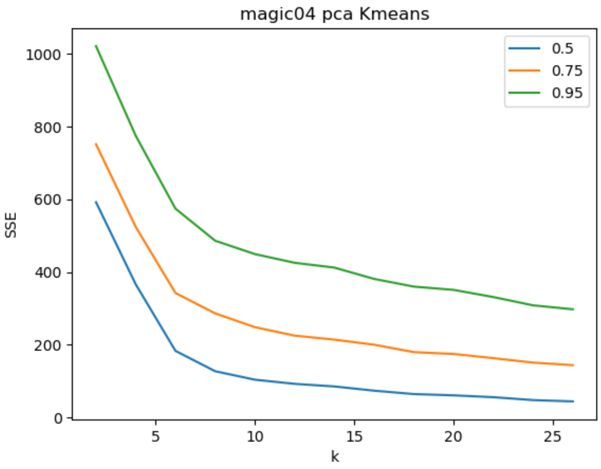
Principal Component Analysis

Principal component Analysis (PCA) is one method of reducing the dimensionality of a dataset. PCA achieves this through looking at transformations of axes and selecting the one that maximize variance in data. From this axis, additional orthogonal projections are added again that maximize variance. The reduction in dimensionality comes when some dimensions produce so low variance that they could be dropped with little effect as the information it originally encoded is now within other dimensions. Interestingly these axes are eigenvectors, and the maximization of variance correlates directly with eigenvalues. Thus the eigenvector with the largest eigenvalue is always selected. For the wine quality dataset with PCA, the largest eigenvalue is 3.22 and covers .29 of the proportion while the next is 1.57 and covers .14. This trend continues for all datasets.

To examine the effects of PCA, it was run on both datasets with a variance cover of 0.5, 0.75, and 0.95. PCA uses variance cover as a threshold to remove less important dimensions. For both datasets, all settings reduce the number of dimensions. But much more interesting is to compare the distribution of the dimensions between the PCA runs and the original data. The PCA dimensions are typically close to normally distributed while the original data varied between normal and heavily skewed.

K means was rerun on the dataset after processing them with PCA at the settings stated above. The experiment from before was repeated. Interestingly, the curves produces are very similar to the values before but shifted downwards on the y axis. The shift is likely partially due to the dimensionality reduction thus each sample in a cluster has less dimensions adding to its distance. Ideally some of the reduction of SSE can be attributed to improvement in the layout of the data. It is important to note, however, that since PCA largely just originates the axis to align in better directions, PCA’s only effect on the distance metric being used, Euclidean, is in the dropped dimensions. A visual look at the assigned clusters provides insight into the effect of these dropped dimensions. The clustering appears much more correct to the actual labels of the data and less sporadic within the data than before especially for low variance cover. It is important to note, the data originally was heavily bundled together thus it would appear PCA helped decouple the different labels.



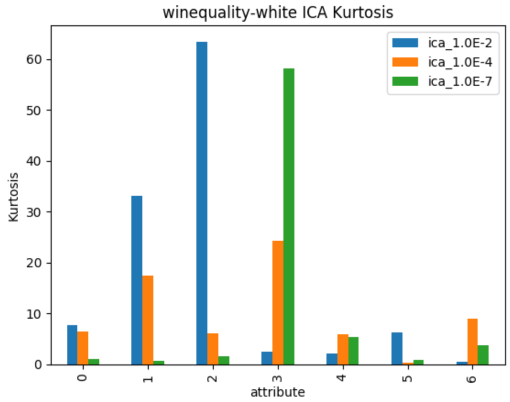


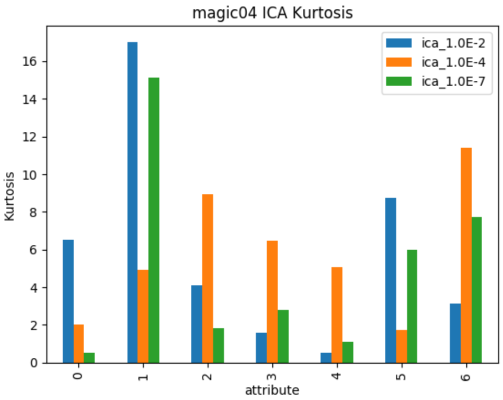
EM was run on the reduced datasets as well. It produced the log likelihoods seen below. Wine quality are reduced below what was found on the overall dataset. However, magic04 was massively increased to around -5 with with 0.5 being the smallest and 0.95 being a the largest over what was found on the overall dataset. This suggests especially when paired with k means that PCA was not very effective on the wine quality datasets likely due to a like of principal axes that could be found. However, magic04 greatly benefits from utilizing PCA due to having useful principal axes with high variance. Visual inspection further corroborates this theory as many of the labels for wine contain multiple clusters.

<u>**Independent Component Analysis**</u>

Independent Component Analysis (ICA) has similarities to PCA. It assumes the data is composed of mixtures of unknown, non-Gaussian, independent variables. It attempts to retrieve these variables. This means the kurtosis of each independent component found can be a very important characteristic to determine how effective ICA was. As ICA is trying to find independent, non-gaussian variables, the kurtosis should be far from 3.0 or the kurtosis of a normal distribution.

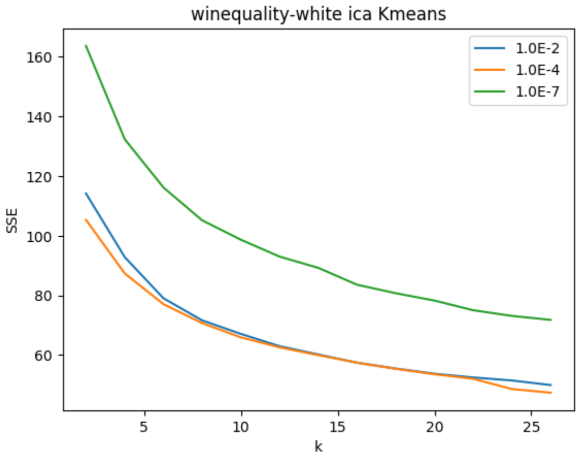
The reduction comes from picking the number of attributes to remove. In this case 7 was chosen for both datasets while the tolerances were varied within 1.0E-7, 1.0E-4, 1.0E-2. As seen below the various runs produced a wide spread of kurtosis values of the attributes. It is interesting to note that the magnitude of kurtosis values of one setting does not seen to have a strong correlations to the others. Additionally, low and high tolerances seem to produce more drastic kurtosis changes between attributes while the middle tolerance seems to be more tightly bunched. Additionally, ICA seems to have found a lot of non-normal components in wine quality while magic04 seems to struggle more especially at some settings. This suggest that the wine quality's underlaying structure is composed of more independent components while magic04 has more covariance between the values.

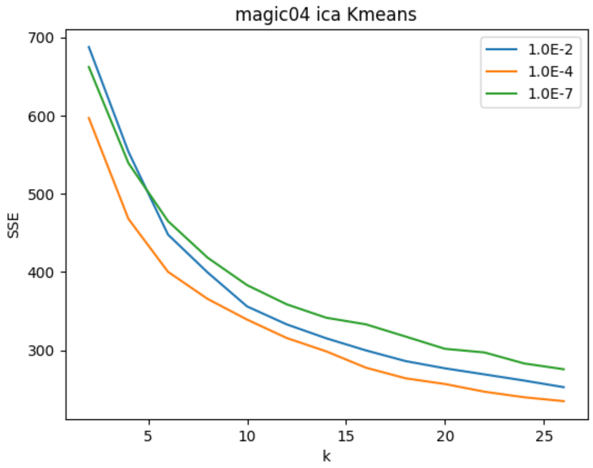




K means was rerun on these reduced settings. This time the results are more significantly different. As was stated above, PCA seemed to find good components for wine quality, so it is unsurprising that this leads to much better performance on the k_means with the SSE being significantly lower than PCA.. Additionally, the curve is much smoother suggesting the surface was smoothed out during the process. However, the elbow is around the same range as before. Magic04 actually produces has a SSE but still close enough that the difference in how the dimensions are defined could make up the difference. Magic04 did not produce as many highly non-normal attributes thus it understandable that it would not see the large drop wine quality did further reinforcing that the more Gaussian nature of magic04 contrasting to wine quality's very much non-normal nature.

Visually inspection shows similar results as far as the spread of clusters between the true labels. ICA however provides some improvement in the smaller sized labels for wine quality. In magic04, it provides a lot more separation to the smaller clusters while maintaining the fidelity of the better clusters from the full clustering,





EM was run on the two datasets after ICA as well. Both datasets had massive increases in log likelihood. Wine into 20 and Magic04 into 38. This suggest especially when take the k means results in mind as well that ICA was able to find good results on the data set. This suggest both datasets have strongly independent components forming its basis. Additionally, magic04 again saw a much larger increases in both log likelihood as well as much larger kurtosis. This further establishes magic04 non-Gaussian structure. Wine quality visually is really well matched by the assigned clusters; however, the assigned clusters compared to the labels is more well matched than k means but has overlaps between clusters and labels.

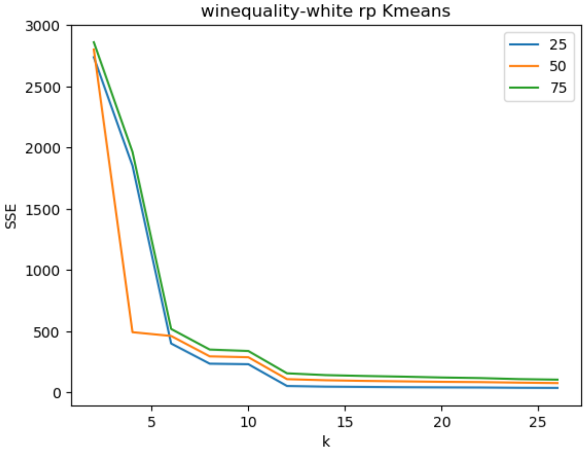
Random Project

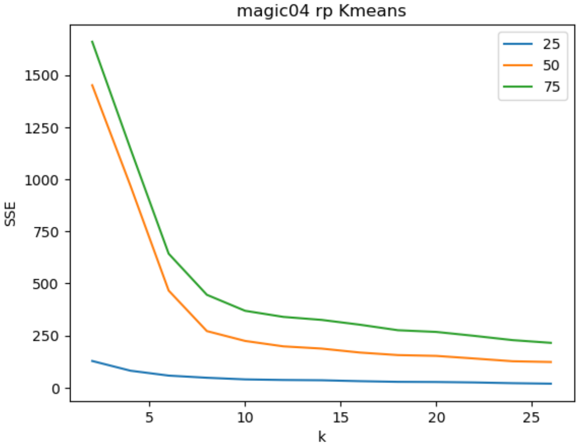
Random projection use random matrix designed to preserve the distances and variation while mapping to a lower dimensional space. Naturally since this is a random algorithm, it is subject to being throw into local optima if they exist which could result in differences between runs. This can be fixed by holding the seed of the randomization as constant as was done. However, this method is very quick and easy to execute compare to the others.

This projection was done on the datasets and maintained 25%, 50%, and 75% of the attributes. Looking at the dimensions formed, overall it appears to produce normal distributions with a few more skewed attributes. Additionally, the dimensions of smaller percent attribute retained appear to match the same numbered attribute in the larger percent runs.

The kmeans run on the datasets produces similar graphs to the full dataset and PCA. Interestingly, the performance was almost akin to PCA, slightly worse in the magic04 case. This is very interesting as it using randomization to perform and achieve these results. Another interesting note is how magic04 for 25% begins near its lowest SSE rather than experiencing a huge drop. This is likely due to the random nature of the algorithms.

Visually, RP produces really well separated clusters over the clusters on the full dataset. In the magic04 dataset, they are well defined zones in each of the dimensions. Wine quality produces shifts some miss cluster samples into the correct samples making the labels more identified with a their own set of clusters.





The EM run on the datasets produced lowest log likelihood on the smaller settings by a significant a

mount akin to the k means. Looking at the clustering of wine quality. It produces really well defined clusters with good; however, the large number of clusters suggest it maybe over fitting with 14 clusters on 25% when there are only two attributes to fit. Beyond this the labels are relatively equally distributed though the clusters suggesting the clusters may not fit to a structure that represents the labels despite producing strong bands of cluster with many of the attributes.

RP

Wine

percent	Likelihood
25	-9.97
50	-19.68
75	-28.73

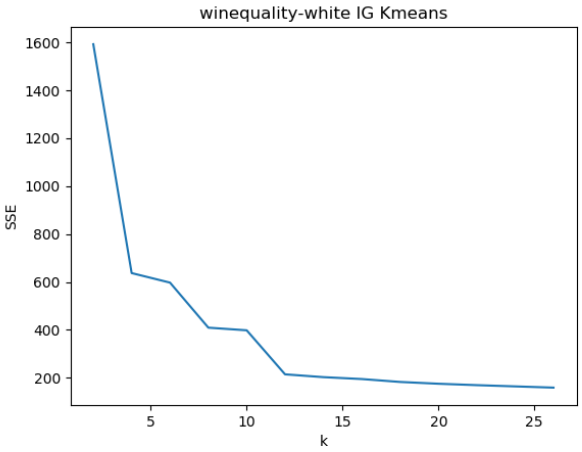
Magic04

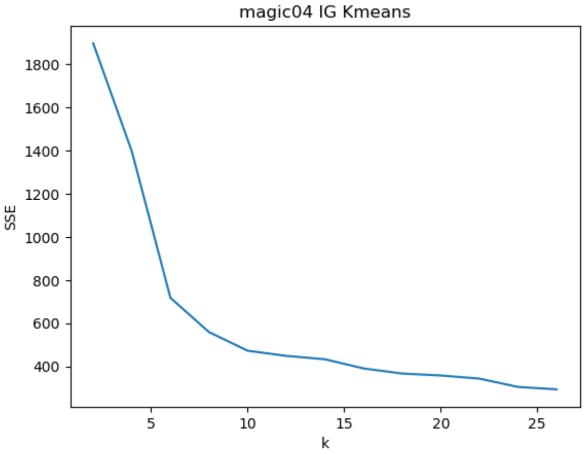
percent	Likelihood
25	-10.96
50	-24.89
75	-34.93

Information Gain

Information gain can be used to reduce dimensions. It follows a similar thought process as ID3: prioritize attributes with high information gain. The information gain of the attributes is calculated and thresholded. All below that threshold are removed. This was run on the datasets. The threshold was set between the largest drop in the information gain. Wine quality was reduced to 8 attributes, and magic04 was reduced to 6 attributes. This suggests that a lot of the data contributed very little to the label of most samples for both datasets.

Kmeans was run on this reduced data. The results were basically the exact same as on the overall set but shifted down. While others had similar curves and trends but still had differences, these stayed the same beyond the SSE shift. This heavily suggest that the removed attributes provided little to the set.





EM was run on this reduced data. The results for this had again very similar results as the original data leading even further evidence that the removed attributes provided little to the set. However, this same likelihood was produced using double the number of clusters. Again the cluster distribution does not match up with the labels with the labels being somewhat evenly distributed throughout the clusters.

Wine
likelihood
-3.13
Magic04
likelihood
-25.69

Neural networks

The magic04 results on the dimension reduction were piped into a neural network to see how these algorithms would effect supervised learning. The neural network settings from the first paper on supervised learning were maintained at a hidden layer of 100, momentum of 0.5, and a learning rate of 0.001. The neural network was iterated up to 700 times as convergence occurred 300 to 400 on the full set.

The result, seen below, were very illuminating on the true effects on the dimensional reduction algorithms on the dataset. PCA is very and away produced the best results seen on the dataset with tiny 0.0133 test error. This is an order of magnitude drop in error compare to all other supervised learning performed on this dataset. This suggests that the data’s underlaying structure is composed on these principal axes with high variance. Additionally, IG performed very well compared to all other non-PCA algorithms. IG simply removed attributes it deemed unnecessary thus it is likely that PCA was doing the same on top of aligning to these stronger axes. Furthermore, the poor performance of ICA makes sense given the many of attributes returned were close to normal distributions while ICA wants highly non-normal. The time distributions vary a lot below; however, the number of iterations before convergence as varies. For instance, PCA was still improving at 700 iterations and 0.0133 test error. Overall, the runtimes of the datasets were similar through the neural network. PCA was able to beat all others with runtime on the same scale just with a slightly higher error.

Dataset	Min train error	test Error	Time
Full	0.3639	0.4036	2.884
PCA	0.0115	0.0133	44.5
ICA	0.568	0.56	3.223
RP	0.55	0.6	6.92
IG	0.15	0.269	8.09

Beyond this, the clustering algorithms were run on these datasets and the clusters were added as an additional feature to see the effect. It provided benefit to the full and IG datasets allowing IG to reach ~0.15 on test error. However, RP and ICA show nearly no benefit. This suggest these algorithms inability to match the structure extended to the clustering thus the cluster provided no new insights. In fact the clustering for these two was very evenly distributed between the labels thus all labels were nearly equally likely for clusters. PCA had little to no change given is little room to improve.

Conclusions

The exploration of unsupervised learning in this paper as uncovered a lot of utility of different algorithms and approaches to these types of problems

Kmeans with the elbow method applied provides decent insight into the provides allowing creation clusters very quickly. However, many of the clusters visually are not very clean. It is a very useful algorithm we clustering needs to be fast or the distance metric matches the data very well. EM gives much cleaner cluster visually however it takes significantly longer to run and creates a long more clusters than k means. EM can be very helpful when the data does not lend itself purely to a distance metric. Again EM takes a lot longer to run especially as sample size increases thus it must really be able to perform much better than k means for it be worthwhile. It is interesting to note for the given datasets k means seemed to match the labels much more closely than EM despite its few clusters.

PCA and ICA are really powerful ways to improve structure of the attributes while reducing them. However, the effectiveness of both is determined by the underlaying structure of the data. Thus both are heavily dependent on domain knowledge or exploring the data itself. This means that for very complex data set where just running a barrage of algoithms maybe way too time intensive using PCA and ICA correctly may not be an options. However other algorithm can be used to good affect still. RP can provide a possible good result depending on the seed with little domain knowledge needed. IG can be a powerful tool to remove attributes were effectively were just noise. It would be an interesting further avenue of study to repeat this study but on the results of IG as it performed very well.

References

[1] Eibe Frank, Mark A. Hall, and Ian H. Witten (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.

[2] Kassambara, A. (2017, September 07). Determining The Optimal Number Of Clusters: 3 Must Know Methods. Retrieved March 31, 2018, from http://www.sthda.com/english/articles/29-cluster-validation-essentials/96-determining-the-optimal-number-of-clusters-3-must-know-methods/#elbow-method

[3] Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[4] P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis.
Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553, 2009.

[5] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.