Unsupervised Learning Comparisons

Nirave Kadakia

Data

We will work with two different datasets, wine quality and the NFL dataset. Both of these datasets are explained in assignment 1, but will be briefly touch upon again.

Red Wine quality data, and a complete description of features, are located at https://archive.ics.uci.edu/ml/datasets/Wine+Quality. This dataset contains 11 attributes based on physiochemical tests, ranging acidity, pH level, alcohol level and more. This is used to predict one attribute ranging from 3 to 8 describing the quality.

Red Wine quality is important because wine is expensive to procure and choosing the wrong wine can cost a lot of money to both an individual and wine merchants.

The NFL statistics gathered have 33 different types of data, ranging from whether who is the home team, the Over/Under (the bettor's average of how many total points), to yards gained via passing, rushing, etc. The list is exhaustive and provides a plethora of information. Two seasons of data are gathered, and for each week, the last 16 games (one season) of combined stats is used. This means that statistics are gathered between seasons, which may pose a problem as teams do tend to change year by year. However, the variance of team victories pales in comparison to having a limited sample size in the beginning of the year – thus a 16 game rolling average is used. To obtain this data and thorough description of the attributes: http://www.repole.com/sun4cast/data.html.

The NFL, and gambling on the NFL, is a multi-billion business and it is important for many to choose who will win or lose.

Clustering.

The R kmeans library and Mclust libraries are used to provide k-means and expectation maximization.

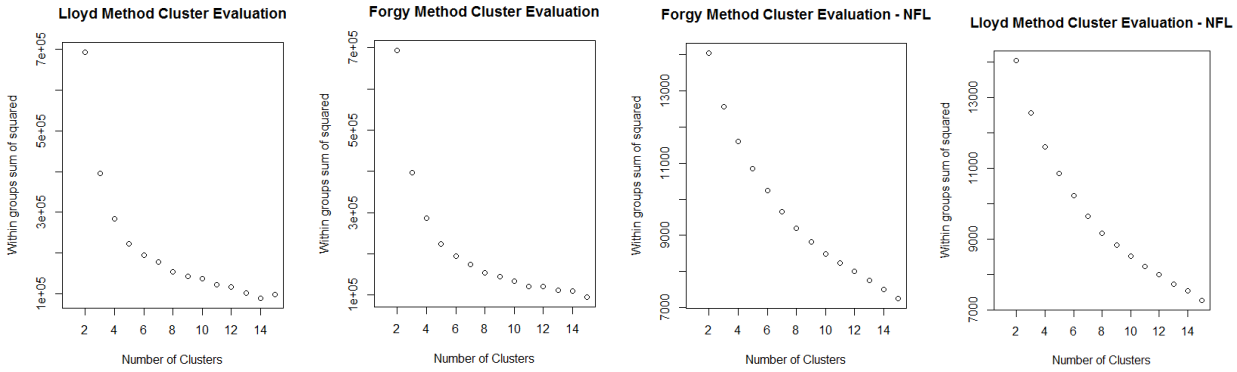
K-means

The first method used to classify or cluster this data is through K-means. K-means is an iterative method to group data to a set number of centroids. The basic algorithm is to first randomly initialize centroids, cluster around those points, then re-center the centroid amongst the cluster. This clustering is a hard clustering where each instance is in exactly one cluster.

Elbow method.

Determining the right number of clusters or centroids is critical. Too many clusters, and the interoperability is lost or meaningless. Too few, and the number of clusters would be too arbitrary and not represent the data accurately. There are many methods to determine the right number of clusters The elbow method allows us to find the best number of clusters that provide adequate information without providing a proper summary of information (i.e. too many clusters are meaningless).

This is done with two different types of methods to determine optimal clusters. One is the Forgy method, and one is the Lloyd method. The Lloyd's method is the simplest method that minimizes within-cluster sum of squares. The Forgy method is similar to Lloyd's algorithm except that it considers the data distribution continuous instead of discrete. Using both methods and measuring the within group of sum of squares for both, it is apparent to see that the "crook", or the rapid reduction of within groups of sum of squares, levels off at around 6 clusters for both methods (two-left most pictures).



For that reason, with the wine quality, 6 clusters are chosen. While that does equal the range of quality is also 6 (3 to 8), this, unfortunately, does not map well.

Similarly, for the NFL (two right-most pictures), a similar result yields a crook at about 6-10 clusters. For the sake of visual interpretation, 6 clusters will be used.

Clustering results

Below are the centers for the clusters for wine:

Cluster	fixed acidity	volatile acidity	citric.acid	residual. sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
1	8.204423	0.525676	0.241745	2.266953	0.083415	12.76659	32.11057	0.996641	3.334742	0.647469	10.47424	5.727273
2	8.754915	0.514658	0.297564	2.475641	0.085517	6.508547	16.30128	0.996677	3.288868	0.646731	10.65865	5.74359
3	8.05	0.557427	0.327206	3.457353	0.089618	32.44853	139.3529	0.997135	3.236176	0.706912	9.805882	5.132353
4	7.768263	0.583174	0.24988	2.980838	0.086623	23.02994	96.88623	0.996744	3.311377	0.627904	10.11407	5.371257
5	8.216611	0.519917	0.250631	2.444352	0.088292	20.83555	49.49834	0.996722	3.330199	0.683821	10.44336	5.69103
6	8.237766	0.518005	0.299043	2.710904	0.099739	25.62766	69.40957	0.997051	3.311649	0.677819	10.19034	5.5

We can see that clusters yield very little about quality – all clusters ranging from 5.3 to 5.7. A rand projection, a computation of agreement between quality and clusters (ranging from -1 to 1, where 1 being perfect agreement), yields a poor result of 0.07 for k-means, meaning it is only slightly above chance determining quality via clustering. This is most likely due to the fact the interactions of differing features can cause interesting behaviors. For example, comparing cluster 2 with cluster 3, wine quality is essentially the same, but cluster 2 has low residual sugar and high alcohol, while cluster 3 has the exact opposite with high residual sugar and low alcohol. Meaning that sugar and alcohol may have a trade-off balance.

And looking at the centers of the cluster yields interesting results. Wine, while not being able to be split on quality, does split on sulfur dioxide, both free and total, and sugar, but everything else is relatively constant.

For the NFL, again, the results were not that interesting. The values are too numerous to show, but the column of win/loss is crucial. The data is too large to show, but there is some relation to wins or losses. Wins and Losses are represented as 1 and -1 respectively. The centers had the following Win or Lose values (.23, .11, -.18, -.49, .07, .27) – meaning that center revolved around a clear loss, a minor loss, two very minor wins, and two somewhat major wins. Meaning that team that falls into one of those clusters will range from large loss to somewhat major win. In other words, there is some interoperability.

And these clusters intuitively make sense. The cluster with the WinOrLose value of -.49 had statistics that were consistent of “losing” teams. Low scoring defense (a scaled -1.49 value while the rest range from 0.04 to 1.01) is one clear example.

Timing was relatively quick, with NFL at 0.13 seconds for k-means and wine at 0.22 seconds for k-means.

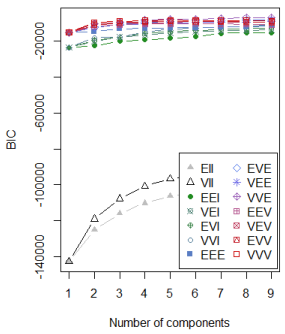
Expectation Maximization

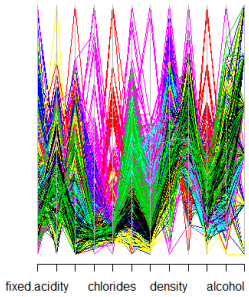
Expectation Maximization (EM) is a clustering method for estimating maximum likelihood. It is an iterative process that computes log-likelihood for a current posterior, then solves for the maximum likelihood parameters. It is a soft clustering method where instances are given probabilities or scores on whether they are in the cluster or not.

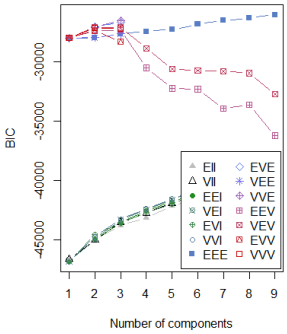
BIC evaluation of number of clusters

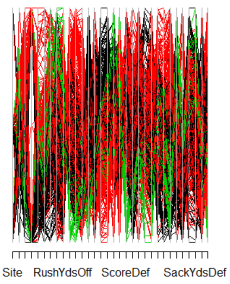
The Bayesian information criterion, that approximates Bayes factors that, is an information criteria method that balances model complexity and precision. The key is to maximize this value.

For wine, while there is no fall off, there is a leveling off of EEV at 7 clusters, so that is what is used. This makes intuitive sense since the dataset is sparse and on the surface, does not seem to be clear cut distinctions. Looking at the parallel coordinate chart below, it is easy to see the randomness. For example, the black line low chloride levels, but high alcohol levels. While the purple cluster does not follow any such pattern.









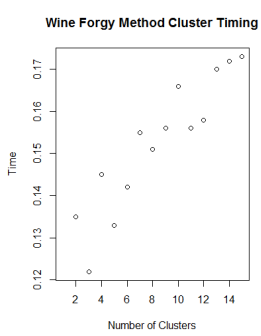
For the NFL, there is a fall off at 3, meaning that 3 clusters is a good fit, as there is a fall off. This makes intuitive sense if you think about football teams. There are clear winners, clear losers, and those that fall in the middle (e.g. close games).

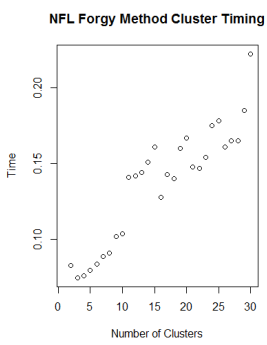
The time for the NFL was 180 seconds for EM. The time for the wine data was 107 seconds for EM. This means that Expectation Maximization is much longer than k-means.

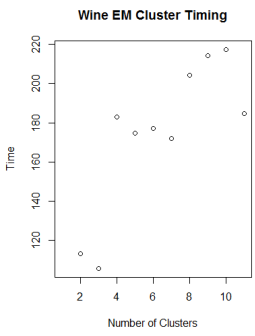
Evaluation of clustering.

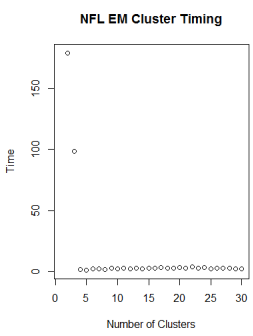
Evaluating clusters is based on multiple criteria. Time, the cleanliness of the clusters, the evenness of the clusters, whether they map to the expected values that we may wish to predict, and how high the intra-cluster similarity and how the inter-cluster similarity is.

In terms of time, k-means is much quicker than Expectation Maximization. Even as the number of features increase, k-means is much quicker (factor of 1000). Below are graphs of wine and NFL with k-means (left), and wine and NFL with EM (right). K-means looks nearly linear, meaning there is high correlation between the number of clusters and time. EM, however, behaves more erratically. In general, EM performs better around the optimal number of clusters, but timings can dramatically increase outside of that.









Below are the plot cluster graphs showing the cluster plots. These plots show the clusters in by compressing the features down to two-dimensional space to allow easy interpretation.

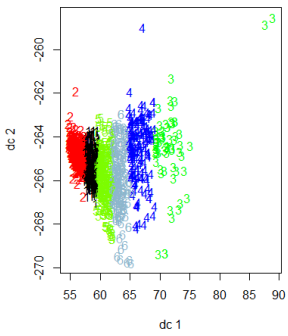


Figure 1 – K-means wine cluster

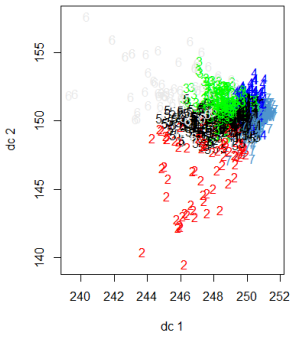


Figure 2- EM Wine cluster

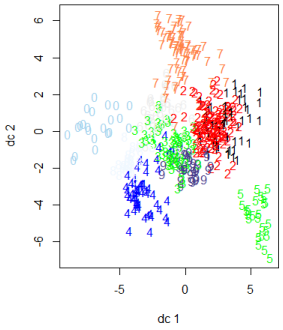


Figure 3-NFL k-means cluster

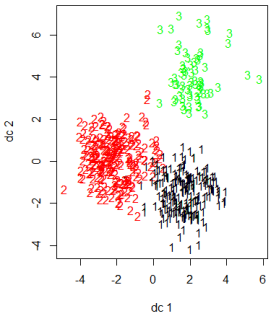


Figure 4 - NFL EM Cluster

The clustering of K-means with wine is very interesting in that it looks like most of the clustering can be done with just 1 dimension. The high intra-cluster similarity (essentially staying within a range in dc1) with k-means signals that PCA does a good job in separating out the clusters. <<TO DO – redo kmeans cluster>

EM does not do as good a job, however, with wine clusters intermingling (high inter-clustering) and large space away from the centers (low intra-clustering). This may be due to the fact that EM uses all components it has access to and thus may not work as well with higher dimensions. EM, however, did a better job. But that's because the ideal K was just 3 for EM.

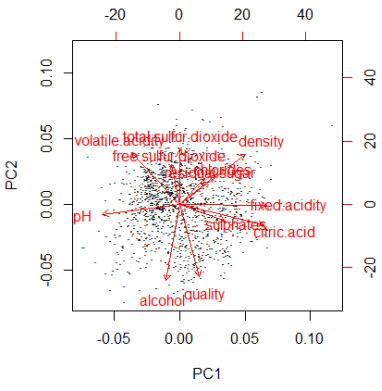
Dimension Reduction

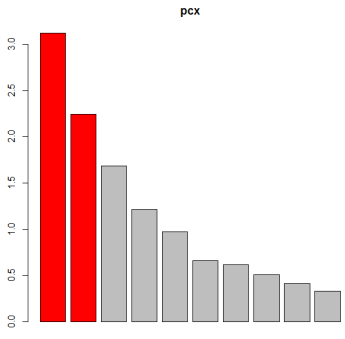
The wine and NFL dataset have a large number of features that could be reduced down for speed and to reduce the curse of dimensionality. Below are 4 methods used to reduce these dimensions.

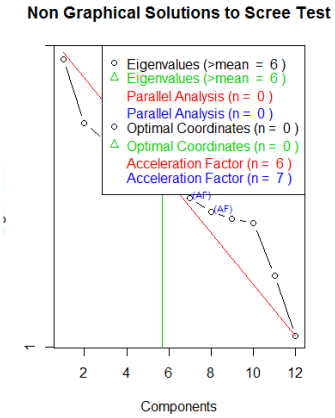
PCA:

Principal component analysis is a linear model technique that maps data to lower-dimensional space such that variance of the data is minimized. The library used is R's prcomp library.

For the wine results, we get the following scatterplot (left) showing the relationship between the first two principal components and the effect on dimensionality. We also see the variance for each principal component in the second graph (middle), and finally, the eigenvalues in the third graph.



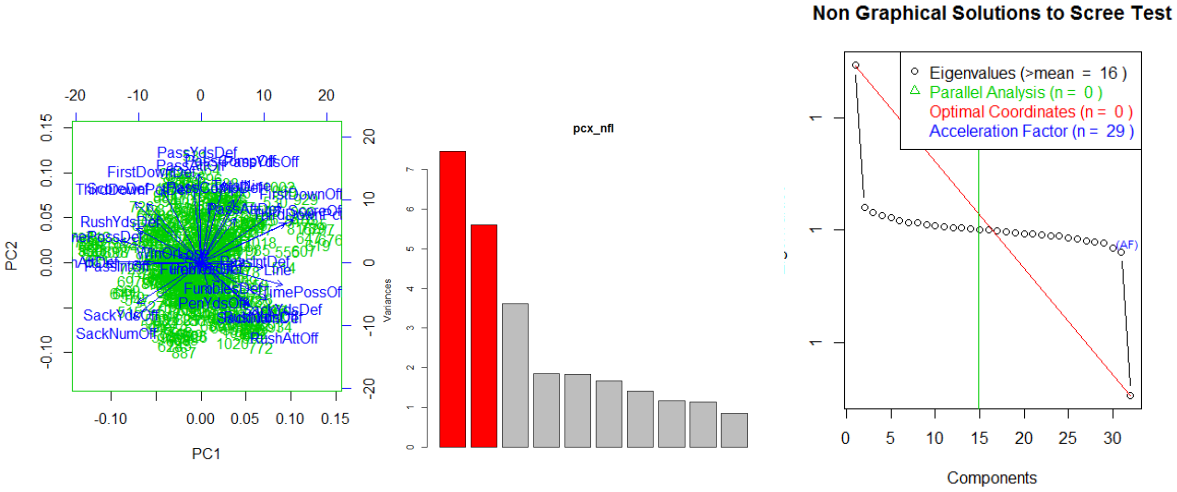




For wine, we can see various attributes being projected, with items like alcohol having a large impact, which makes intuitive sense. We see that the variance goes down gradually, with a steep drop off after 6. This is agreement with the appropriate eigenvalues, which, through the library R library nFactor, yields a value of 6 dimensions as ideal. The

eigenvalues are shown to tell how many principal components are needed to reconstruct a large of a fraction of the dataset.

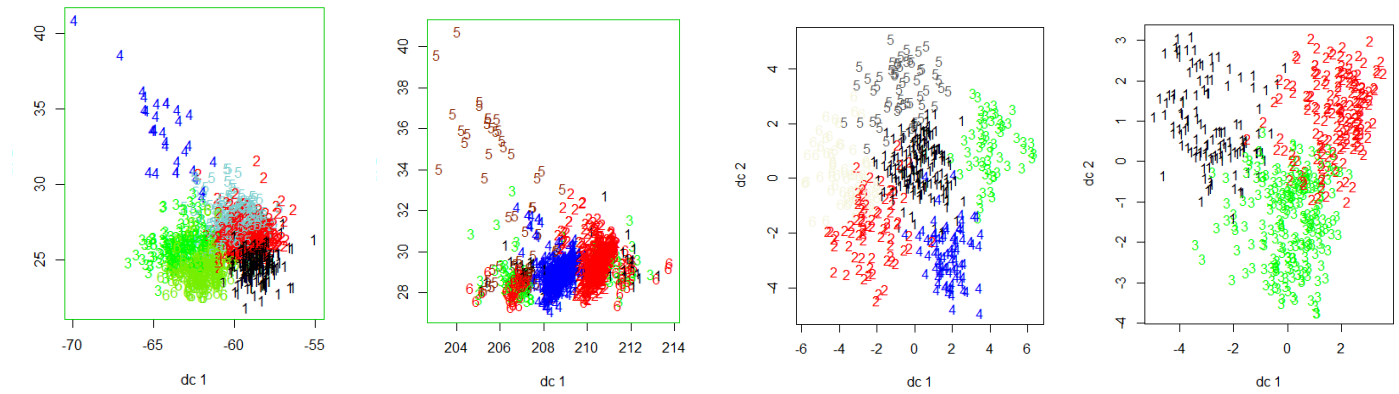
For the NFL, we get the following:



For NFL fans, this makes sense. Sacks, which are a great indicator of defensive prowess (and defense tends to win games) have a large magnitude. Also, in the variance model, there is a large amount drop off after 3. This, however, does not agree with the ideal eignenvalue, 16, given by the library nFactor. With so many features (>30), reducing it down to 16 itself is quite a feat. And this reduction makes sense. Many offensive and defensive statistics are repetitive or subsets of one another. For example, sacks have a high correlation with sack yards.

The time took for wine was a mere 0.02 seconds. The time took for PCA for the NFL dataset was also 0.02 seconds. In other words, this is fast.

Clustering with PCA:



For the wine, k-means (left most) is 0.15 seconds. EM (second to the left) is 7.01 seconds. While there is a slight slowdown in k-means, this may or may not be attributable to general variance in computing times. The difference between 0.15 and 0.02 may be negligible. However, the difference in EM is quite large. While performing EM on raw features took 180 seconds, performing PCA and EM together takes only 7.03 seconds – making it a viable alternative to EM clustering when time is a factor.

In terms of evaluating the clusters, the wine quality does not work as well compared to raw data. Both the k-means (left-most) and EM clusters (second left-most picture) have lower intra-cluster similarity and higher inter cluster similarity – essentially, clustering was not as good. While rand tables are not a great tool, it does yield slightly improved results. K-

mean's clustering for wine was 0.1381, and EM's clustering for wine was 0.919. While slightly better than chance, it was the best out of all dimension reduction algorithms.

For the NFL dataset, similar results occur. K-means (second to the right) and EM (rightmost) also have lower intra-cluster similarity and higher inter cluster similarity.

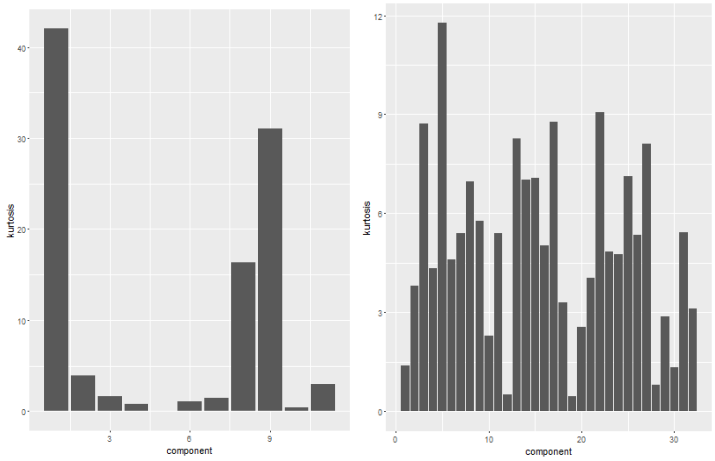
Neural Network with PCA for Wine Quality:

The wine, with PCA used, was used to recalculate the Neural Network to find wine quality from assignment 1. To recap the results with raw data, the out of test accuracy was 59.62% and time was 2.02 seconds. The PCA, with dimension reduction only returns of the best 3 components 42.90% accuracy within 2.08 seconds.

In other words, using PCA does significantly worse than using raw data. This may be through many factors which is not known. One possible answer is that PCA assume linear relationships between variables and the wine data and NFL may have non-linear dependencies. This makes intuitive sense. For example, in the NFL, a statistic like rushing yards is not linearly depending on offensive scoring. Sometimes, teams run to score, sometimes, teams run to run out the clock and not score. Or, it could be that the class variance is very high compared to between class variance and PCA would result in discarding information that separates classes. Finding the exact reason would require further investigation.

ICA:

Independent Component Analysis, like PCA, finds components so that the number of dimensions can be reduced. However, unlike PCA, it finds independent (e.g. orthogonal) components. These components, however, are unranked, and to determine which components, and the optimal number of components, requires looking at kurtosis. In basic terms, kurtosis is a measure of how close points are to a point. Graphing this around the clusters can yield the best components to use. This is implemented via R's fastICA library.

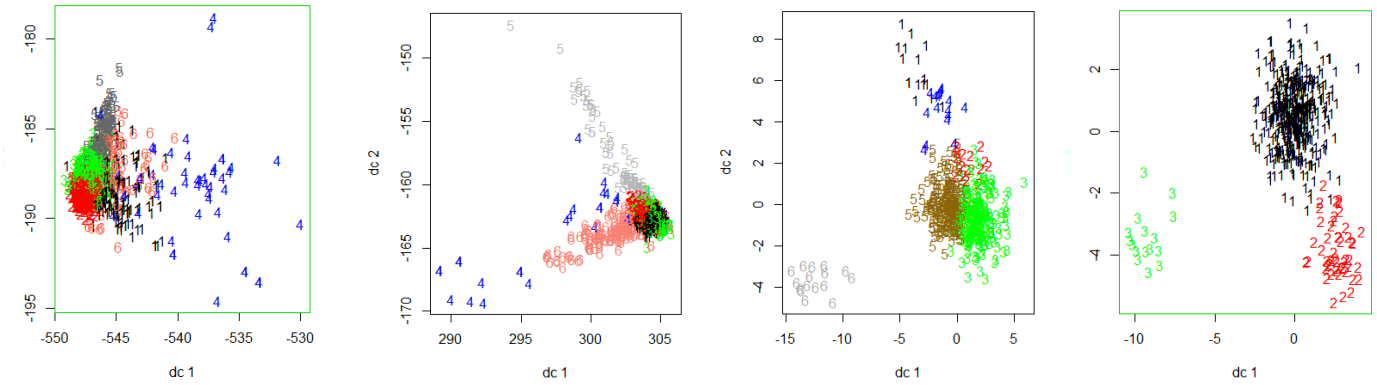


For wine, on the left, there are 3 large components that signify a large proportion of information needed to reconstruct the original dataset. For the NFL, on the right, there are a large number of components with high kurtosis, meaning that selecting the right components is difficult, and dimension reduction does not work too well. This may be due to the fact that NFL statistics are complex, intertwined, and have strange behavior. For example, a statistic like Passing Yards is often times very high for winning teams, but also for losing teams (who throw the ball deep to catch up). However, some teams with good defenses rush the ball instead of passing, so their Passing Yards may skew it down. In other words, it is tough for many components to be independent of one another in the NFL. While, wine, on the other hand, can have independence.

For wine, the top 3 components were used. And for the NFL, though there are quite a few dimensions, only the top 4 components were used for simplicity's sake.

Clustering with ICA:

Rerunning our cluster plot with FA and 3 and 4 dimensions for our wine (left) and NFL (right), with cluster sizes of 6, 6, 6 and 3, respectively, for comparison purposes, are as follows:



The results are significantly works than PCA for wine (K-means at the left, EM second to the left) and the NFL (K-means second to the right, EM to the right). Intra-cluster similarity is much lower, and inter-cluster similarity is much higher. Also, it is apparent in the wine example that there is significant overlap in the cluster.

These bad results make intuitive sense as ICA assumes linear independence amongst the signals. Since the same growing atmosphere and varietal of grape affects the features, independence cannot be assumed. Trivially speaking, statistics in sports can also never be assumed to be independent.

Again, rand tables, while not a great too for this application, does mirror the results. K-mean's clustering for wine was -0.0279, and EM's clustering for wine was 0.03. Meaning, combined, nothing was gained in terms of clustering in terms of finding quality.

Neural Network with ICA for Wine Quality:

Running it against the Neural Network. The ICA, with dimension reduction only returns of the best 3 components 39.11% accuracy within 2.02 seconds.

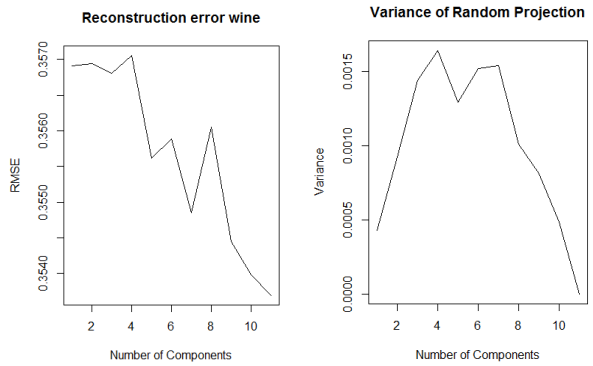
This is worse than raw and PCA because ICA itself is a poor choice of dimension reduction.

Random Projection

Random Projection is a faster algorithm that transforms the data into a random rotation matrix while guaranteeing a maximum distortion.

Unfortunately, due to time constraints and a bug that could not be solved in time, only wine's variance and reconstruction plot will be shown.

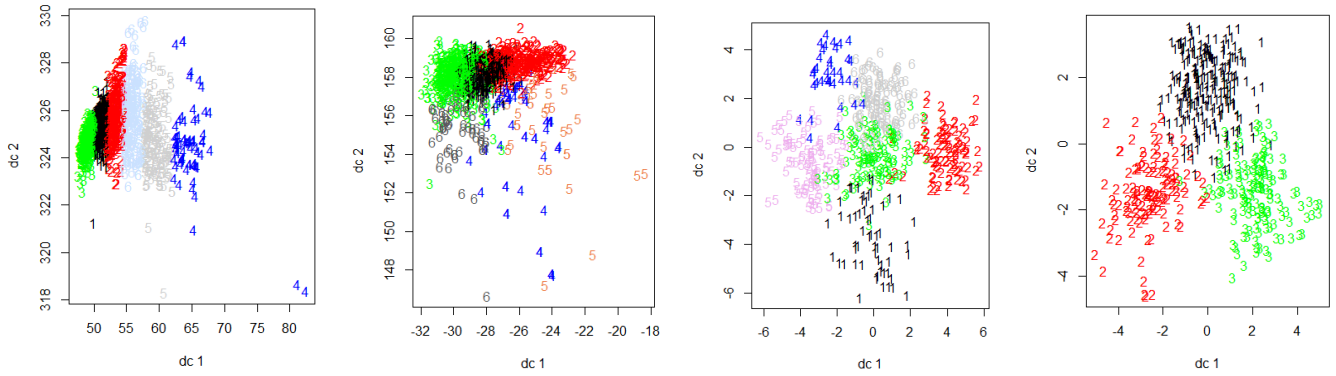
Again, there is a trade-off between number of dimensions and accurate representation. To determine the ideal number of dimensions, a crook is needed to be found where the accuracy (represented by root-mean square error between the original dataset and the Random Projection's reconstructed dataset using the average of 20 runs) – similar elbow method as discussed before. Unfortunately, as pictured (bottom right), there is no obvious answer – but 6 seems like a good choice.



In addition, since Random Projection is random in nature, the variance of the average of 20 runs is shown above. The dimension reduction, around the middle, show the most variance. This makes sense, as a few dimensions would clump obvious clusters together. And too many dimensions would essentially yield a mimic of the dataset.

Clustering with Random Projection:

Rerunning our cluster plot with FA and 4 and 9 dimensions for our wine (left) and NFL (right), with cluster sizes of 6, 6, 6 and 3, respectively, for comparison purposes, are as follows:



The results are interesting for wine (K-means at the left, EM second to the left) and the NFL (K-means second to the right, EM to the right) compared to the raw data. Intra-cluster similarity is slightly lower, and inter-cluster similarity is slightly higher. Wine, for k-means, does reasonably well, but for EM, not so much. RP, by the way of randomly fitting without as many assumptions, is one of the better dimension reduction methods.

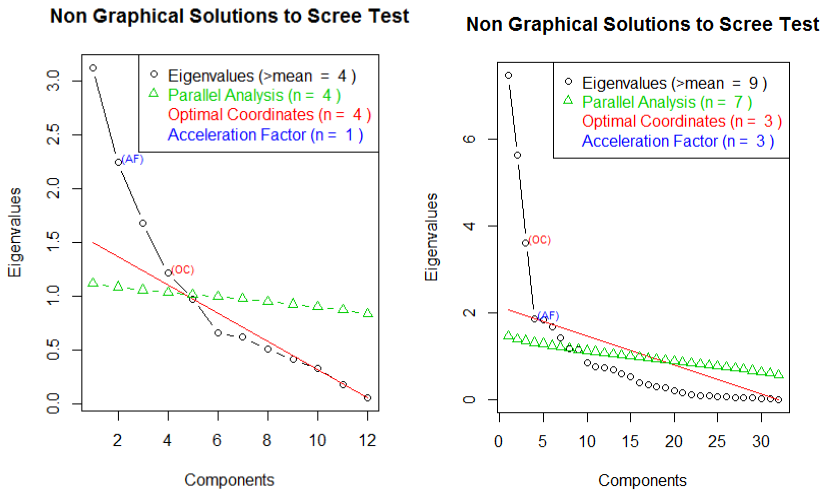
Again, rand tables, while not a great too for this application, does show the clusters lack of correlation with quality. K-mean's clustering for wine was -0.0053 and EM's clustering for wine was 0.08. Meaning, combined, nothing was gained in terms of clustering in terms of finding quality.

Neural Network with Random Projection:

Running it against the Neural Network. The RP, with dimension reduction only returns of the best 6 components, running 10 iterations, has a median of 42.34% accuracy within 1.77 seconds. Median was used because a few runs averaged extremely low accuracy results (e.g, < 35%), and this is partially due to the random nature of random projection.

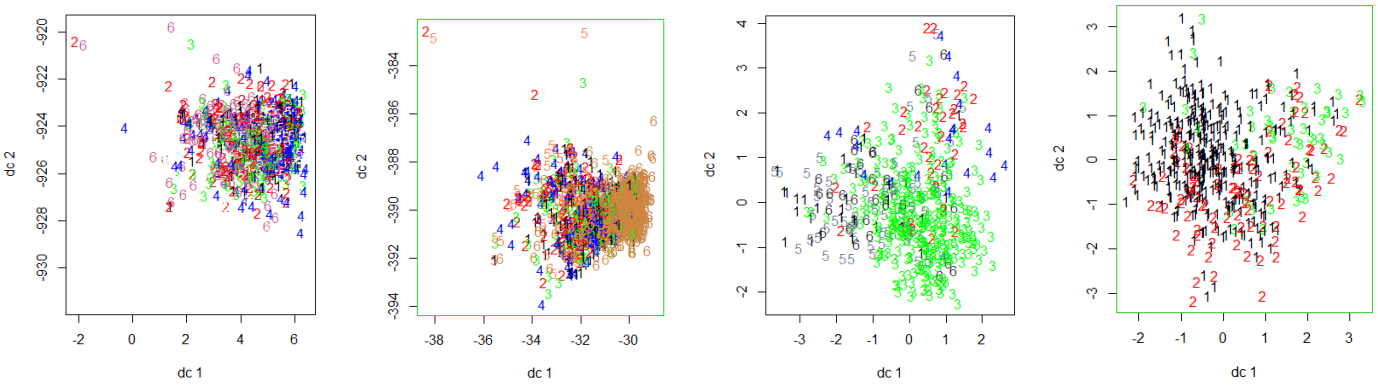
Factor Analysis

Factor Analysis (FA) is very similar to ICA, assuming observed random variables are a linear combination of independent components. However, ICA assumes non-gaussian components while FA assumes Gaussian components. This was implemented using R's factanal library. Eignenvalues using R's nFactors can yield the desired number of dimensions used. In this case, wine is 4 and EM is 9.



Clustering with Factor Analysis:

Rerunning our cluster plot with FA and 4 and 9 dimensions for our wine (left – K-means and EM) and NFL (right – K-means followed by EM), with cluster sizes of 6, 6, 6 and 3, respectively, for comparison purposes, are as follows:



This is by far the worst results. The clustering is not very exact for either situation, and very spread out. This makes intuitive sense because FA assumes independence, which is explained in the ICA section above, and it assumes a Gaussian distribution. This is not the case. For example, looking through the data, factors, like alcohol for wine, is evenly distributed.

Again, rand tables, while not a great too for this application, does mirror the poor results. K-mean's clustering for wine was -0.022, and EM's clustering for wine was 0.003. Meaning, combined, nothing was gained in terms of clustering in terms of finding quality.

The time took for wine was a mere 0.08 seconds. The time took for FA for the NFL dataset was also 0.3 seconds. In other words, this is fast, but not as fast as PCA.

Neural Network with Factor Analysis:

Running it against the Neural Network. The FA, with dimension reduction only returns of the best 4 components 43.53% accuracy within 1.6 seconds.

This is more in line with what is expected. Accuracy is worse compared to raw data, but the amount of time is approximately 25%-30% lower.

Interestingly enough, FA returns the best results out of all 4 dimension reduction algorithms.

Clusters as Features in Neural Networks for wine quality.

With this complex clustering, a natural question is to see if these clusters can help the final goal – determining wine quality. The clusters for K-means and EM were added as features to the Neural Network and run again. The following are the times and out of sample accuracy for 10 runs, with the median chosen:

Clustering/Dimension Reduction	Accuracy	Speed (seconds)
Normal	59.62%	2.02
K-means	58.66%	1.80
Expectation Maximization	58.35%	2.30

Adding clusters as features has negligible effect on the accuracy, slightly lowering the accuracy. K-means had a slight improvement in terms of times, but EM had a lower effect. K-means did prove more effective in terms of classification, which may explain that the clusters improve the timing. The timing effect of EM slowing down the Neural Network makes intuitive sense as adding an extra feature would slow down a normal Neural Network in general.

Improvements

Various improvements could have been made to ensure better performance on both datasets. A thorough cleaning of data would be beneficial and a reduction in known correlated or dependent features. For example, for the NFL, total yards is just total rushing yard plus total passing yards. Consolidating those would have made a small difference perhaps. In addition, experimenting with various parameter (e.g. kmeans' nstart), and rerunning cross validation against the Neural Networks may have boosted results.

Conclusion

Two different clustering methods and various dimension reduction algorithms were used on wine quality and NFL datasets. The clustering methods were able to classify data, but not line up with the supervised learning goal. These two datasets also did not lend itself well to dimension reduction, as the plot clusters were slightly worse and Neural Networks proved less accurate with very little speed gain. However, a slight bump in speed was gained by including clusters as features with only a slight loss of accuracy for k-means.

References:

EM in R:
https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Clustering/Expectation_Maximization_(EM)#Analysis

http://stackoverflow.com/questions/20446053/k-means-lloyd-forgy-macqueen-hartigan-wong

The k-means clustering technique: http://www.tqmp.org/RegularArticles/vol09-1/p015/p015.pdf

https://en.wikipedia.org/wiki/Dimensionality_reduction#Principal_component_analysis_.28PCA.29

http://stats.stackexchange.com/questions/35319/what-is-the-relationship-between-independent-component-analysis-and-factor-analy

http://web.engr.oregonstate.edu/~xfern/classes/cs534/notes/Unsupervised-model-11.pdf - – high intra- cluster similarity vs – low inter- cluster similarity

https://www.quora.com/What-is-the-difference-between-K-means-and-the-mixture-model-of-Gaussian

https://www.quora.com/For-which-cases-is-it-not-preferable-to-use-principal-component-analysis-PCA

http://www.statmethods.net/advstats/factor.html

http://stats.stackexchange.com/questions/52773/what-can-cause-pca-to-worsen-results-of-a-classifier