Unsupervised Learning

Jarrett Alexander

November 5, 2022

1 Introduction

To determine the behavior of different unsupervised learning techniques the following experiments were conducted. These evaluate the performance of 2 different clustering algorithms and 4 different dimensionality reduction techniques. Scikit Learn was used for all algorithms in the following experiments. [PVG+11]

2 Datasets

2.1 Credit Card Default Prediction

This dataset is 30000 rows and has 23 features. Some of these features are highly correlated so Dimensionality Reduction should prove valuable on this dataset. This dataset is imbalanced with only 22% of the rows being defaults. F1 Macro was the metric used here because of this imbalance. [DG17]

2.2 Wine Quality Prediction

This is a smaller dataset with only 1599 samples. Each sample has 11 features all of which are numeric. There are 10 possible categories of wine ranging from 0 to 10 but the majority of scores 5 and 6 with .45 and .4 of the samples having these scores respectively. Accuracy is the metric used for evaluating this dataset. [DG17]

3 Clustering

The two clustering algorithms below take different approaches at doing this and thus yield similar but different results. The labels in the above datasets are already known. However, for these experiments, the labels were only looked at after determining the number of clusters.

3.1 K Means

Experimentation When evaluating the number of clusters for KMeans, 2 approaches were taken. First a plot of the sum-of-squared error is taken at various number of clusters. An elbow is found and that determines a baseline for where to begin looking further (chart omitted, look at EM section for an example). The silhouette score is taken of a few different clusters. This is a measure of how similar a sample is to the rest of the samples in the same cluster. The highest average is used to determine the final number of clusters for K Means.

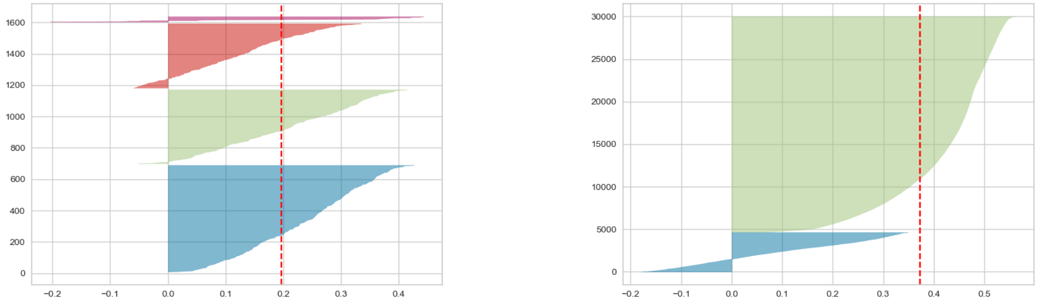

Figure 1: K Means Silhouette (Wine Left, Defaults Right)

For the defaults dataset, the elbow plot determined that the elbow could be 2, 3, or 4. These were all tried and an N of 2 was determined to have the highest average silhouette score and therefore was used for all later validation. Therefore, an K=2 is chosen for this dataset. Further, the silhouette showed an interesting pattern for all N=2,3,4. For all of them, there was N-1 highly related clusters. There is a singular cluster that has a silhouette score of roughly 0 for all of them as well. This cluster is always the smallest as well. K Means is grouping the majority of the data into one or more clusters and then putting all outliers in the other cluster.

For the wine dataset, the elbow plot yielded a range of 3 to 8 different clustering options. This makes sense because this dataset could really be stated as ”good vs bad” wine or ”really bad vs bad vs okay vs good vs really good” wine. Both problems are valid and that explains the wide elbow. After evaluating the silhouette score for these, K=4 was the largest silhouette score.

Further experimentation showed that K=3 was the last time that the clusters were all of similar size. At k=4, a single small cluster with an extremely high silhouette score appears. This same thing happens at all 10¿K¿4 as well This single cluster with high score is likely bringing up the average for the whole metric. However, this is the metric and therefore a K=4 is chosen for the validation stage.

Validation Before looking pairwise charts, I looked at the labels and the differences in the features of each cluster for both datasets.

The defaults dataset showed 2 very imbalanced clusters which appeared to loosely resemble the underlying class that it is trying to predict. If this is true, there would be a high correlation between the label and the cluster. Not in this case. The clustering algorithms output has a 0 correlation to the labels of the data. The clusters were mostly identical except for the limit balance on the card and the remaining balance. This means that the clustering algorithm distinguished between the magnitude of the credit card debt and not the actual underlying class. This also means that the size of the credit card does not determine whether a person will default on that credit card or not. While it would not affect the N chosen above, this same experiment was ran at N=3 and N=4. The same results appeared. Each was a cluster representing the size of the credit card. Therefore, N=2 still appears to be valid. It is grouping samples by a large or small credit card.

For the wine dataset, there was also little correlation between the clusters and the labels. For all clusters, the average wine score was between 5.4 and 6.0 which is in line with the majority of the dataset. Next the small cluster was evaluated as it had the highest silhouette score. The main feature that stood out was the chlorides contained in those wines. Each other cluster had a mean chloride content near 0 standard deviations from the mean. The smallest cluster however had a mean chloride content 5.962 standard deviations from the mean. It appears that clustering values of N=2 and N=3 do not allow for these highly specialized clusters based on 1 or 2 features. However, as N increases, we are able to see an increase in the similarity of small groups. While N=4 appears to make sense here due to this, increasing N could yield other results if the desired result is small groups of very similar wines.

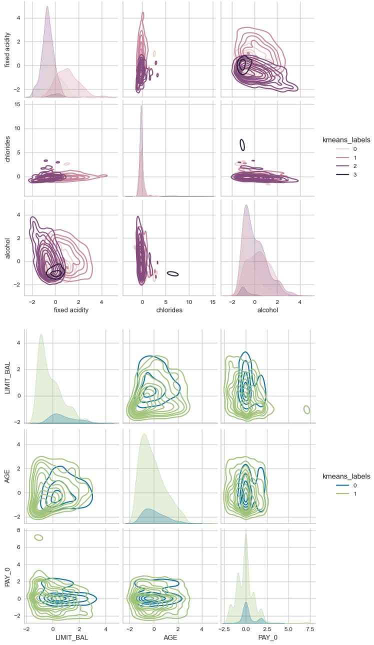

Figure 2: K Means Pairwise (Wine Top)

The pattern with the chloride content can be seen in figure 2. The figure shows the small cluster in black being separated from the rest on the chloride feature. There is also some decent clustering between clusters with the fixed acidity. The clusters for defaults are shown to be similar but distinguished on limit balance and payment due. This is much more subtle than the wine dataset because this dataset seems to be much more clustered around a single point.

3.2 Expectation Maximization

When evaluating the number of clusters for Expectation Maximization the elbow method was used. This works the same as it did above. However, in this case, the average log-likelihood metric is used instead of the sum-of-squared errors. This indicates how likely, on average, a sample is generated by the label that it is given.

Experimentation For the defaults dataset, an N=5 elbow was chosen. This is already very different from K Means. K Means showed decent clustering at N=4 and N=5 but it was clear that N=2 was best. This difference seems to be because EM does not draw hard lines. If the data is grouped together closely, it is easier for many different centers to generate a given point. In K Means, there was a single large cluster as well. This is not the case here. The sizes of the clusters are 3749, 4389, 5794, and 10370. While there is one large cluster, the difference is not as great as K Means which had clusters of size 4636 and 25364.

There was also a very low correlation between the clusters created by K Means and by EM. The correlation between the clusters was pretty much 0. On different runs, this correlation ranged from .12 to -.14. This also means that each time EM was run it landed in a different location. This further exploits the difference between EM and KMeans. The soft boundaries seem to be more affected by the starting positions on datasets that are noisy.

The wine dataset had an N=3 due to the elbow method. N=4 could also be valid but N=3 seems to be more likely. Interestingly, this is lower than was found with K Means. There was a much larger difference in cluster size however. The clusters were 1075, 348, and 176 in size. The largest cluster has the most ”normal” values. Central Limit Theorem still holds true. Most of the features in this cluster are centered near or at the mean. The smaller clusters have different centers however. Acidity and chlorides once again seem to be the distinguishing features.

Validation The labels for EM also had no correlation to the underlying labels in the data. Plotting the same features with EM shows a major difference between K Means and EM. All of these clusters are centered near the same point for both datasets. The lack of hard lines allows it to maximize log likelihood by allowing each sample to be generated by all clusters almost equally. While this seems like a limitation, it shows that without DR these datasets are not split well just on untransformed features.

3.3 Runtimes

Refer to table 1. While EM is faster for small cluster sizes, it get slower much faster than K Means does. Also, as the dataset increases in size, the run time does as well. The default dataset is much larger and EM seems to be hurt by this as well. The EM algorithm suffers because it has to be calculate the likelihood of each cluster generating each sample at each iteration. K Means just has to find the distance to each cluster center and that determines the hard cluster on its own.

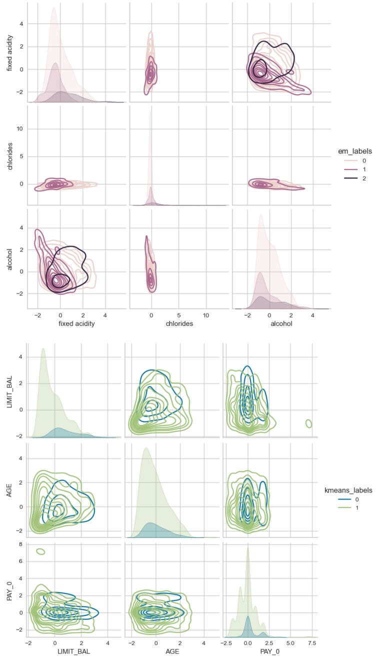

Figure 3: EM Pairwise (Wine Top)

4 Dimensionality Reduction

4.1 Principle Component Analysis

For the following experiments, the total explained variance and the individual explained variances are used to determine a choice for components. For each dataset, all components 1-N (where N is the number of features) were tested.

The wine dataset is known to be noisy both from the clustering experiments above and also from Assignment 1. To reduce this, the experiment needs to determine what number of components is best to reduce or remove this noise from the data. It is hard to classify the amount of noise in the dataset without training a model on the PCA data so we must evaluate the different eigenvalues and vectors created by PCA. As can be seen in figure 4, the first component makes up 28.2% of the explained variance of the data for the wine dataset. Looking into this first component, fixed acidity, citric acid, and pH were the most represented features with coefficients of .48, .46, and -.43. This is interesting because these are all measurements of acidity. The different measurements of acidity tend to explain the most about the of the data.

The first component is able to explain 30.9% of the total variance in the default dataset. The second component is also better than that of the wine dataset. The wine dataset has a steadier drop off to the

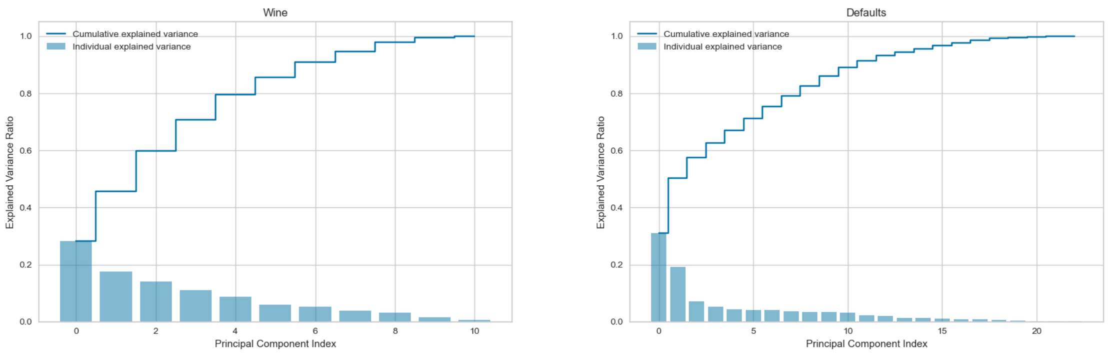

Figure 4: PCA Explained Variance (Wine Left, Defaults Right)

eigenvalues for its components. After the 2nd component, all values are higher for the wine dataset for each corresponding compoenent. This means that the wine dataset needs more components to explain a majority of the data leading to the conclusion that it is a noisier dataset even when dimensionality is reduced. However, the defaults dataset is not able to create components that maximize explained variance that well. This implies that while the majority of the data is explained well by just 2 vectors, the rest is nuanced and hard to summarize in a few vectors.

Wine		
Clusters	Algorithm	Fit Time
1	K Means	0.328
1	EM	.048
3	K Means	.344
3	EM	.075
5	K Means	.359
5	EM	.085
Defaults		
Clusters	Algorithm	Fit Time
1	K Means	.201
1	EM	.063
3	K Means	.263
3	EM	.923
5	K Means	.323
5	EM	2.156

Table 1: Clustering Run Time

Next, to remove noise, we select a number of components such that the total explained variance high and the individual contributions of each component are also still high. This is a heuristic but this seems to occur for the wine dataset at N=7 components. The total explained variance is 90.8% but the individual explained variance is 5% at the lowest. Going forward, N=7 will be used for the wine dataset. For the default set, the same approach was taken and 11 components were chosen. This represents 89.1% of the total variance. Both of these make sense as there are highly correlated features in both datasets. The wine dataset has 4 different features representing acidity and default has 15 representing credit card size.

4.1.1 Clustering

Using the number of components given above, both clustering algorithms were ran again on each dataset and the results are compared to those clusters given in the first section. Further experiments were ran with lower number of DR components and showed that typically meant that the number of clusters would be reduced as well.

K Means Very similar results are achieved after running PCA with 7 components on the wine dataset. At K=4, the highest silhouette score is .21 which is the same as previously found. The smallest cluster has 28 samples instead of 29 and the rest are broken down as 715, 374, and 482. The smallest cluster shows the largest deviation from the means as well. There are 2 compoents (1 and 4) that are 3.6 and 6.0 standard deviations from the mean. This is nearly identical to the previous clustering. The difference is that it is actually a larger difference in this case. Because those components are making up a larger amount of the overall variance than the corresponding features, this cluster is even further away along those vectors.

The default dataset showed a similar similarity to the pre-reduced clustering. A silhouette score was taken and the best option was found to be N=2. The silhouette score again shows to be the same at .37. As N increased, the clusters became more disjoint and the score dropped. The distribution was the same as well with one large cluster and one smaller cluster. The smaller cluster was almost identical in all components except the first 2 components. This makes sense again because these are the most important components for explaining variance in the data.

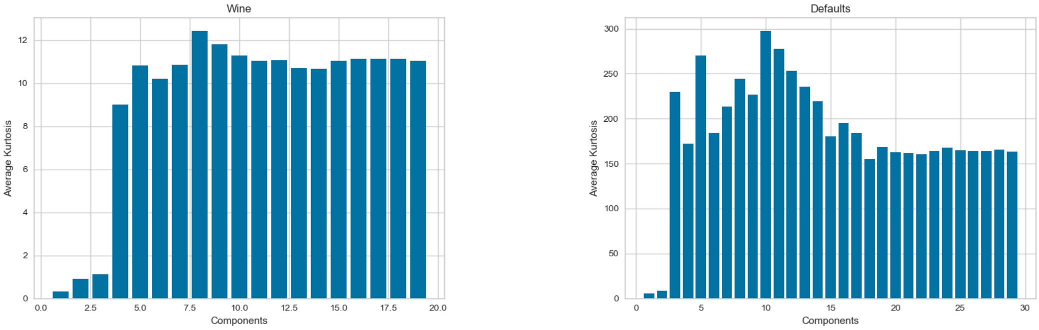

Figure 6: ICA Average Kurtosis

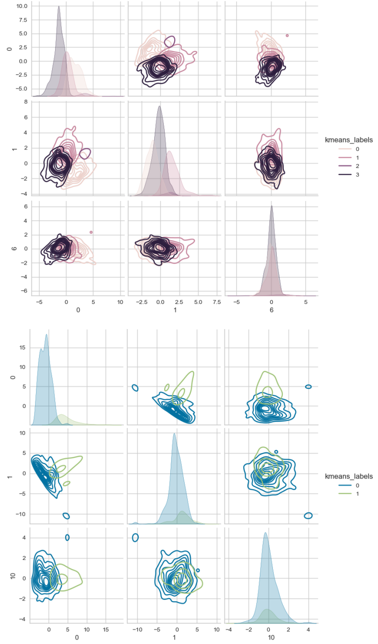

Figure 5: PCA KMeans Pairwise (Wine Top)

The pairwise plots for this clustering shows the 2 first components and the final component for both datasets. The clustering is the best in the first 2 components for both datasets because those explain the most variance. The final component shows very little clustering in both cases as well. This component makes up very little of the explained variance and therefore is unable to improve clustering.

The defaults plot shows an artifact from PCA. The first 2 clusters appears to be distributed orthogonal to each other. This does not appear in later DR techniques.

EM For the wine dataset, EM showed an elbow at N=3 which appears to be identical. The clustering looks similar as well. Different to K Means, the components dont seem to correlate to the clustering strength. The only component that showed the ability to separate the data is the 5th component. The smallest cluster is almost separated here but not much. Also, the distributions are much more normal than the clustering done by K Means. That is just because of the soft clustering done by the algorithm. In general, it appears that EM is not clustering as well on this dataset because of how noisy it is.

The default dataset shows a similar outcome. N=5 is chosen from the elbow method. Clustering using EM provides a better look at each distribution inside the data. As is seen in the first 2 components, 3 of the clusters are very well defined but they are still overlapping in these components. These 3 distributions appear to be laid perpendicular to each other as well which is an outcome of PCA. The rest of the components seem to compress all distributions into a single distribution gaining very little of the explained variance. The first 2 components once again show most of this separation.

4.2 Independent Component Analysis

To determine the number of components for ICA each dataset was fit and projected using ICA with N components. N ranged from 1-20 for the wine dataset and 1-30 for the default dataset. At each iteration, the kurtosis of the resulting data is normalized based on the number of components that are given. This allows us to pick a set of components that on average have the highest individual kurtosis as well. High kurtosis means that the distribution of the data along that vector is very tail heavy and therefore is a valuable component in ICA.

For the wine dataset the highest average kurtosis of the dataset after transformation took place when ICA used 8 components. The average kurtosis was 12.41. The components with the highest kurtosis were component 3, 6, and 8. When looking at the normalized value of the coefficients for these vectors, the first component was comprised on mainly the chloride feature with its chloride coefficient 2.2 standard deviations above the average. This corresponds to some of the clustering results above. Related to component 3, component 6 was made up of chlorides as well as citric acid, pH, and sulphates at 1.67, 1.36, and 2.3 deviations from the mean. This is similar to the number 1 component created by PCA. Finally, component 8 was the sugar vector and it was 2.4 deviations from the mean.

The default dataset had a maximum average kurtosis when given 10 components. The transformed data had an average kurtosis of 295.78. Doing the same evaluation as above shows that many more components have high kurtosis for this dataset. However, only 2 had a kurtosis higher than the average. The default dataset has 6 columns representing payment amounts over past months. For some reason the component with the highest kurtosis focused on the second month payment. The coefficient on this was 2.78 deviations away from the average of the others. However, this is the only feature like this. The others are statistically negligible. The other component did something similar but with the third month instead. We saw that these two columns are highly correlated though. This is where ICA differs from PCA. PCA would not be able to create 2 components like this because each has to be orthogonal. These are highly correlated features and ICA is able to create a component for both of them. These components are still independent but are not orthogonal and therefore both of these components represent similar ideas.

4.2.1 Clustering

Similar to PCA, the number of components are given by the outcome of the previous step.

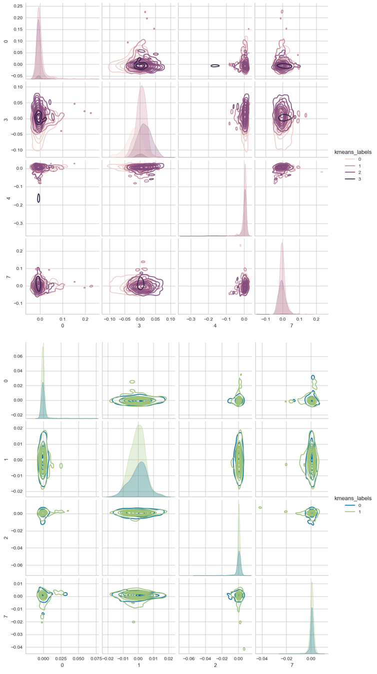

Figure 7: ICA KMeans Pairwise (Wine Top)

K Means Differing from the previous experiment, no elbow was shown for ICA when using K Means on the default dataset. Silhouette was used exclusively to determine number of clusters. This result was the same as before showing 2 clusters as best. 4 was found to be best again for the wine dataset.

Differing from before, the clusters created by ICA were of similar size. This indicates that K Means might be splitting data arbitrarily on certain components if no clear line is able to be found. This is seen in the pairwise plot where there seems to be a line through what would typically be considered a single normal distribution. I believe that ICA might be transforming the data into a space that is more difficult to draw lines between for the default set.

Plotted are some components with high kurtosis and others with low kurtosis. These do not show any improvement in clustering for either dataset. The performance seems to be the worst of any of the algorithms when it comes to performance on clustering.

Differing from PCA again, there is not any orthogonality shown between different clusters.

EM Using the same methods as in PCA, N=3 was chosen for the wine dataset and N=2 was chosen for the defaults dataset. Again, plotting the high and low kurtosis value components shows very little clustering based on those components. EM struggled similarly to K Means and all clusters appeared to be centered in the same location again for both datasets. The chart is omitted here as it looks nearly identical to K Means.

Highly correlated features seem to pose a problem for clustering after using ICA. These features can be the defining feature for a specific component which then gives high kurtosis to all of those components. If these highly correlated features are not useful in splitting the data then it causes an issue with clustering afterwards. This is seen for both K Means and EM for ICA.

4.3 Randomized Projections

To determine the number of components for RP, the reconstruction error was measured by doing a transformation of the data and then an inverse of that transformation. This was then compared using mean squared error to the original data to see how much information was lost. A baseline of 10% reconstruction error is used to determine the number of components. This is at 21 for the defaults dataset and 10 for the wine dataset.

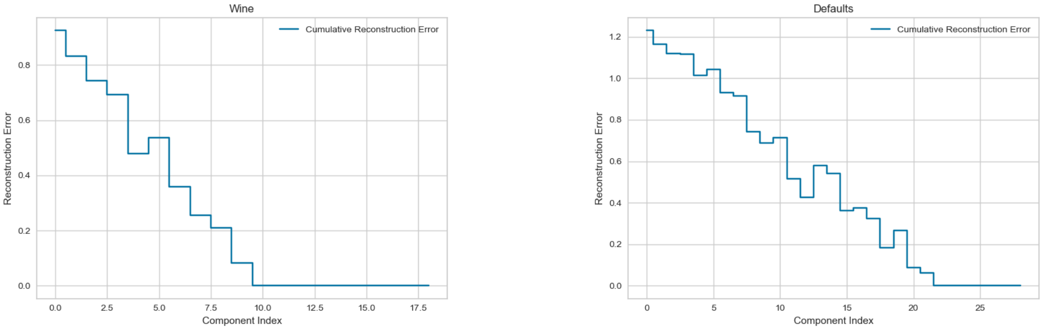

Figure 8: RP Reconstruction Error

For both datasets the overall reconstruction error decreased steadily as more projections were added. This makes sense as each projection is likely to represent some percentage of the data. Some projections add more (these would likely correspond to some of the important components in PCA/ICA) than others but this tends to average out as more projections are added.

This pattern did not change on differing runs. Each run had a small variance in which new components would lead to large or small reductions in error but the overall outcome was always the same. The reconstruction error reached 0 as the number of components equaled the number features. This is guaranteed as long as all of the vectors are linearly independent. The trend always appeared to be increasing steadily for both datasets on every run.

4.3.1 Clustering

The same component number as found above will be used for these experiments.

K Means The wine dataset had a highest silhouette score at N=2. Default was the same. This is the first DR technique that showed this for wine. However, the difference between the score of 2 and 4 clusters was very minimal and would differ on different runs of RP. This makes sense as RP is a random technique. Also, I chose 2 simply to change things up.

The pairwise plot shows all for this algorithm. This plot was slimmed down to just a few features that summarize the behavior. Unlike PCA and ICA where some components were better than others, RP shows decent clustering on all components. Some components create great splits in the data and some do not but on average, each splits the data pretty well. This ends up giving a very good result with 2 nearly equally sized clusters on the wine dataset.

The pairwise plot for the default dataset shows a simliar result. It actually has the best clustering performance of all so far for this dataset (visually). For each component there is a visual ”center” of the data where most of the samples are coming from. However, there are some noisy outliers which are all classified as the smaller cluster. Sadly, these have no correlation to the labels of the real dataset.

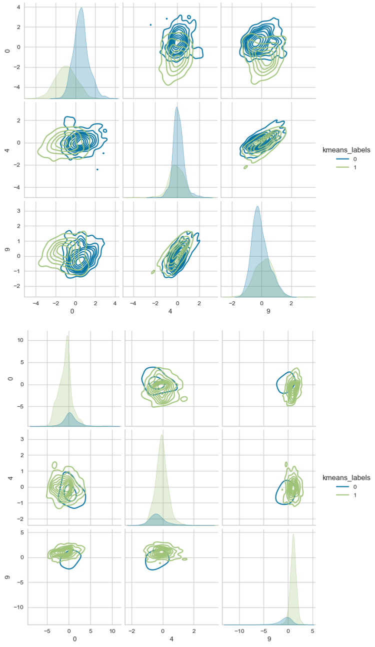

Figure 9: RP KMeans Pairwise (Wine Top)

EM EM had a similar result due to RP. The default dataset showed an elbow at 3 instead of 2 this time. This was subject to change based on each run but 3 was chosen to evaluate this dataset with 3 clusters for the first time. Wine had an elbow at N=3.

The wine dataset showed very little clustering with EM and RP. Because each component was randomly chosen, each represents a roughly equal amount of the underlying variance in the data. As has been seen with all experiments using EM so far, it is difficult for it to cluster when the data is already roughly normally distributed.

EM with 3 clusters on the default dataset did not provide any new information from the previous attempts with 2 clusters. Similar to the results with K means, we see a central cluster that contains most of the data and then a cluster that represents the outliers. However, in EM's case, all 3 clusters are somewhat stacked for each component. There is a single cluster that branches out some to cover the outliers but it is still centered near the centers of the other clusters. This is the same problem that EM has seemed to struggle with for every experiment so far. Random Projections are not able to help EM distinguish between different distributions.

4.4 Linear Discriminant Analysis

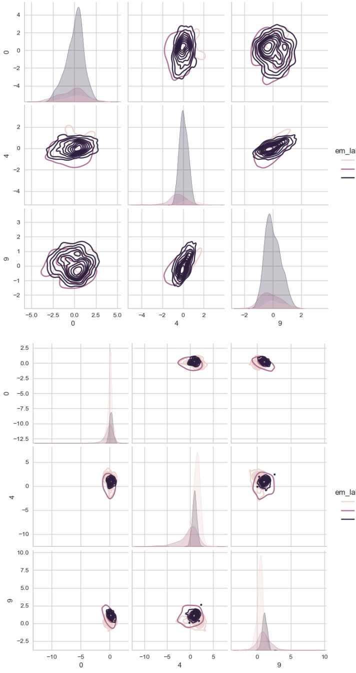

Figure 10: RP EM Pairwise (Wine Top)

LDA is different from other DR techniques because it allows the algorithm to look at the samples in the data. This allows it some performance gains that other algorithms don't gain from. By looking at the labels the algorithm is picking up a bias that others do not. It is assuming that the labels actually mean anything in regards to the data. We hope that is the case for most labels but by assuming it, the DR may be hindered from truly unsupervised exploration of the data.

LDA works in 3 steps. First, it calculates the mean of each feature in X based on samples with the same labels. Each mean is compared with means of the other classes. Then, the difference between each label's mean and each sample with that label is calculated. Finally, the lower dimensional projection is created that maximizes the difference in inter-labeled means but lowers that of intra-labeled means. This reminds me of a combination of some sort of boosting mixed with clustering turned into a DR technique. [LDA]

LDA is limited by the number of labels that you have. It is only able to create a projection with n-1 components where n is the number of different output classes. For binary classification this means that it can only create a single component transformation.

Cumulative explained variance is the metric used for evaluating number of components. For the default dataset there is no choice. A single component is used and it has an explained variance of 1.0. For the Wine dataset, we see that 84.9% of the variance is explained in the first component. 10.2% in the next and therefore 2 components are used. This is dramatically different than any other algorithm we have seen. Because it is able to split data based on labels, single components are able to account for a majority of the variance needed to explain the dataset.

The default dataset transformation component determined that the most important features were the initial status of the repayment as well as the size of the bill. LDA is able to look at labels and defines a vector that is likely to determine something about those labels. PCA/ICA are using vectors that seem to mean something in relation to the whole dataset instead. This creates bias for LDA.

Volatile acidity was the most important feature for LDA's first component. Thinking back to wine quality, volatile acidity would lead to a poor tasting wine in most cases. Once again, the ability of LDA to look at the labels does seem to have some affect on the component choices it makes due to its bias. Other important features included the amount of alcohol and fixed acidity. Alcohol content is inversely related to the others implying that higher alcohol means a better wine.

4.4.1 Clustering

K Means While N=4 showed a high silhoutte score for the wine dataset, N=2 was slightly higher and will be used here. N=2 showed the highest score for the defaults set.

The wine dataset was almost completely split by the first component and not much was added with the second component. This leans back towards the algorithm and how it splits the data. It splits on an already known difference in the data and can explain ¿80% of this variance with a single component. This is shown clearly within the first component splitting the 2 clusters with very little overlap. The second component shows almost no ability to cluster these 2 distributions.

The same is shown for the defaults dataset. There are 2 distributions that are split nearly perfectly by a single component. The larger cluster seems to be split into 3 smaller clusters as well when looking at the chart. Another experiment was ran and K Means + LDA were able to split this into 4 distinct distriubtions with very little overlap. LDA seems a bit like cheating because we are looking at the answers beforehand.

EM Nearly identical results arise when looking at EM. Both datasets ended up with the same number of clusters as in K Means. The outputs of those experiments is identical as well. For the wine dataset, we see a near perfect split due to the first component and then negligible split from the second. This could easily be clustered with a single component. Defaults behaved the same as well. The clustering output from K Means and EM were almost entirely the same. LDA is able to split the data so well using a single component that both EM and K Means tend towards the same result. This is the first time we see that on the defaults dataset. This dataset has struggled with data that is too similar for EM in every other experiment. It would always just find 2 or 3 clusters that are nearly identical in distribution. LDA was able to actually split this large distribution up into smaller distributions and that allowed EM to find different clusters.

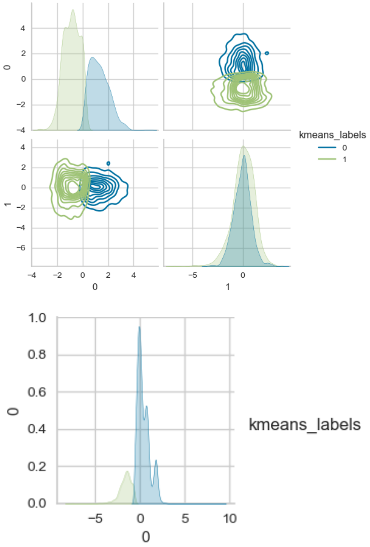

Figure 11: LDA KMeans Pairwise (Wine Top)

4.5 Runtimes

LDA and RP were the fastest at 5ms each for fitting on the wine dataset. ICA and PCA were next at 8ms and 17ms respectively. They fit the defaults dataset with 3ms for RP, 51 ms for LDA, 68ms for PCA, and 108ms for ICA. RP is fastest as it determines the components without doing any other computation. The rest perform similarly because they are all doing some computation to determine components. LDA is fastest here because it is using information to split that the other algorithms cannot which gives it an advantage. It also requires much fewer components than PCA or ICA.

5 Neural Networks

The wine dataset is used to determine the effects of dimensionality reduction and clustering on the performance of a neural network. This network is 4 hidden layers with 10 nodes in each layer, relu activation for all nodes, an adam optimizer, 400 epochs, and a learning rate of .001. Accuracy is the metric used for evaluation when using the wine dataset. The input layer is changed to have N nodes where N is the number of components for the given algorithm. A validation split of .2 was used and a hold out test set of size .2 was also used.

5.1 Dimensionality Reduction

The following experiments use the following component counts for each algorithm: PCA with 7, ICA with 8, RP with 10, and LDA with 2. These are the same number of components that were found before and the number of components that all clustering was run on after DR.

PCA and RP also seem to be able to be overfit. This is not as strong as the overfitting seen with no DR but that makes sense because of the removed noise. At its best, PCA performed slightly better than no DR and better than RP in all situations. RP has better performance on the training set as well. RP selects components at random which allows it to retain some of the noise that PCA does not. This induces more overfitting than with PCA.

ICA performed well most of the time. In a few runs, it struggled to make it through local optima. The new transformation makes the local optima more prominent because it is maximizing independence of different features within the data. This could exaggerate different optima depending on the dataset and therefore cause this issue. The loss value of ICA was the best of all algorithms.

LDA also performed extremely well. It had no troubles with local optima like ICA. There was little to no overfitting with this algorithm. This is because it maximizes the separability of the data with respect to its labels. Therefore, the variance required in the model to fit the data is lower because it is separated more cleanly. This reduces memorization of the data because it is able to perform well without such memorization. With only 2 components, this outperformed all other algorithms (including no algorithm).

The test set results showed LDA to be the best. It had an accuracy of .645, followed by no DR with .616, .609 for both ICA and PCA, and finally RP with .572. RP suffered the most from overfitting as it maintained the majority of the noise in the dataset simply due to component choice. LDA was able to split the data by the labels (note, this was only done on the training set so there was no leakage. The number of components were also determined only on the training set).. It seems to have removed the most variance while maintaining the most information about the dataset because it was able to bias its transformation with label information. PCA and ICA performed similarly on the test set but ICA performed better during training and validation. Therefore, if forced to decide winners, the algorithms come in this order: LDA, ICA, PCA, and RP.

For runtimes, all of them performed faster than no DR. They were all about the same speed at 14.7, 14.9, 14.9, and 14.9 seconds for PCA, ICA, RP, and LDA respectively. No DR completed training in 15.15. This is likely because there are less weights to update in the network as well as a less complex feature space to explore when finding an optima.

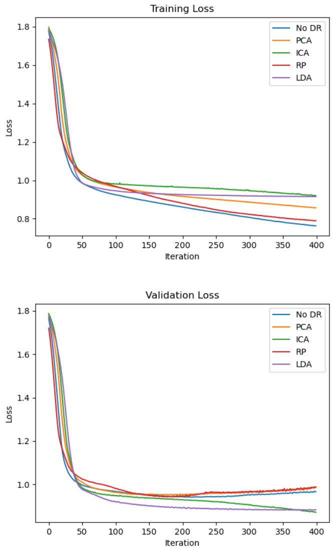

Figure 12: DR Loss Curves

5.2 Clustering

The same network structure described above was used again. The number of clusters remained fixed at the numbers found in section 3. The clusters were fit on the training data only. After fitting, the output was added as a feature to the model. This was tested with K Means, EM, and then both.

There was not any noticeable difference in the results between the control and the 3 experiments. The loss curves show that there is a similar level of overfitting between all 4 runs. Clusters were shown to have no correlation with the labels of the data and this follows here. It will only allow the model to memorize another piece of information and increase the problem of overfitting .

As for runtimes, these were identical as well. After running EM and K Means, the run times of training were all the same at 15.1, 15.2, 15.1, and 15.0 for none, K Means, EM, and both respectively. Adding a single feature only adds 10 more connections in the network and does not add any information to be learned and therefore does not affect training time.

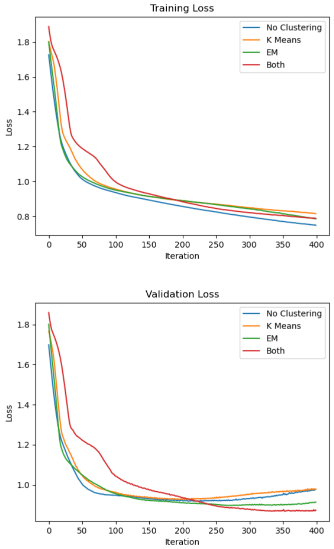

Figure 13: Clustering Loss Curves

6 Conclusion

Overall, K Means seemed to perform better when the data did not have hard lines in it. EM would tend to just stack the clusters on top of each other. After PCA, ICA, RP, and LDA had been ran, both performed much better as the distribution of the data was easier for these algorithms to fit.

PCA and ICA seemed to perform very similarly when clustering and training a NN. These are simlar algorithms with components that make up large amounts of the variance in a dataset. RP was fast and was able to improve clustering but required more components to do so. Finally, LDA was able to use the bias from the labeled data to create near perfect clusters. However, this bias limits the amount of dataset exploration that is possible with PCA and ICA.

References

[DG17] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[LDA] `https://www.knowledgehut.com/blog/data-science/linear-discriminant-analysis-for-machine-learning`. Accessed: 2022-11-5.

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.