Charlie McKnight
CS4641
Charles Isbell
4/2/2017

Analysis of Unsupervised Learning Algorithms

Introduction

This paper focuses on analyzing unsupervised learning algorithms. The Clustering algorithms that I used were the Weka implementations of Expectation Maximization and SimpleKMeans. For the Dimensionality Reduction Algorithms I used the Weka implementations of PCA, ICA (student-filters package), Randomized Projections, and CfsSubsetEvalutation (chosen algorithm). Unless otherwise stated distances are calculated using euclidean distance as this made the most sense for the datasets as you will see below. In addition, All clusters were found without including the classification as an attribute.

CfsSubsetEval is a Feature Selection algorithm that selects attributes (best first) based on their individual predictive ability and their independance from the already selected attributes to arrive at a subset. I chose this algorithm because of its intuitive nature and simplicity as well as the fact it was a feature selection algorithm and did no transformation. This differentiated it from the other three algorithms.

For this paper I will be focusing on the ability of these unsupervised learning algorithms to help with supervised learning. I will do this because many times unsupervised learning techniques are used to help understand the data better and to reduce the curse of dimensionality for supervised learning.

Datasets

Phishing Websites Dataset

For the first dataset I decided to use the dataset that I have used in the past two assignments. It is the Phishing Websites Dataset consisting of 30 binary attributes (-1 = false; 0 = unknown; 1 = true) and a binary classification as a phishing website or not. There are 11055 instances with a roughly 55/45 split between phishing and not. A Neural Network is able to achieve an accuracy of ~96% after some tuning.

It is an interesting dataset for this assignment because of the fact that it has a large number of attributes (as compared to the next dataset), however each attribute contains very little information since they are boolean values.

Fraudulent Banknotes Dataset

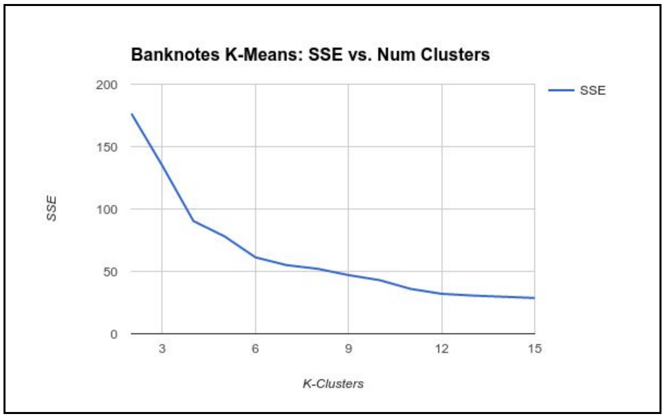
For the second dataset I decided to switch from the other dataset I used in the first project. I decided this because the other dataset I used contained mainly categorical attributes and I decided it would be more interesting to compare the Phishing Dataset with something that contained real valued attributes. I chose the Banknote Authentication Dataset. This consists of 1372 instances containing 4 real valued attributes of measurements taking from photographs of banknotes. These instances are then labeled as fraudulent or not.

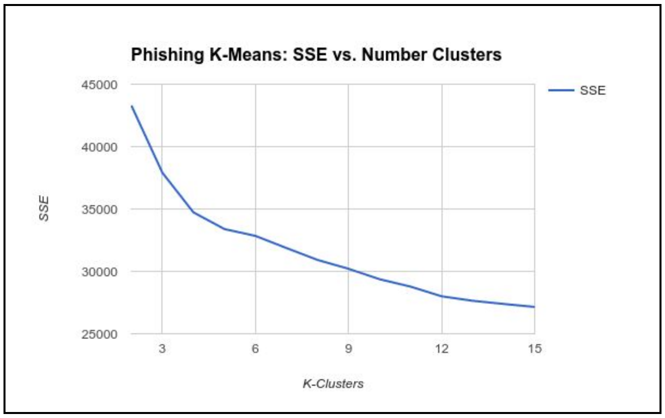
This is an interesting dataset to compare with the Phishing Websites because it has a relatively small number of attributes. While it has a small number of attributes each attribute contains a lot more information seeing as they are real valued. This dataset also lends itself more readily to linear transformation as they are real valued instead of categories or boolean values.

Untransformed Clustering of Datasets

K-Means: Number of Clusters (K)

To decide the ideal number of clusters for K-Means I decided to use the Sum of Square Error (SSE) for each cluster which is calculated by squaring its distance from the center of its assigned cluster. This metric in an of itself cannot be used to decide the quality of a particular K because as K approaches the number of instances, the SSE approaches 0. To combat this I graphed the SSE by K and looked for the point where the Error started to level off, creating a sort of elbow shape. This is the point where the SSE is shrinking simply due to there being a larger number of clusters and not because the clusters describe the data any better.



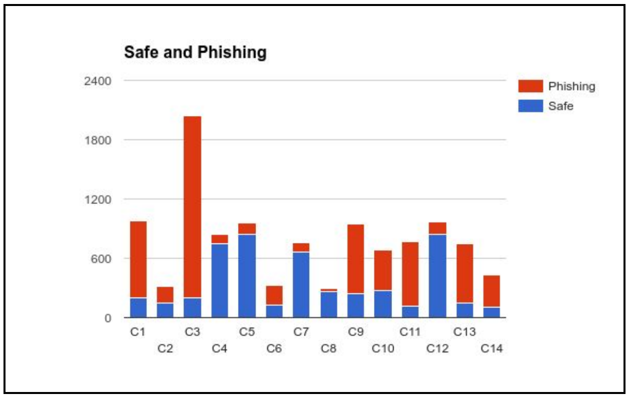


As one can see by looking at the above graphs Banknotes starts to level off at roughly K=12. Phishing data starts to level off at closer to K =14. For the analysis in the next section I will be using those K values.

K-Means Clusters

Phishing Data

Since the data has 30 different attributes excluding the classification, visualization of the clusters can be difficult. However since one of the eventual goals of the clustering is to later do supervised learning the following is a graph of the clusters with the colors indicating the classification.

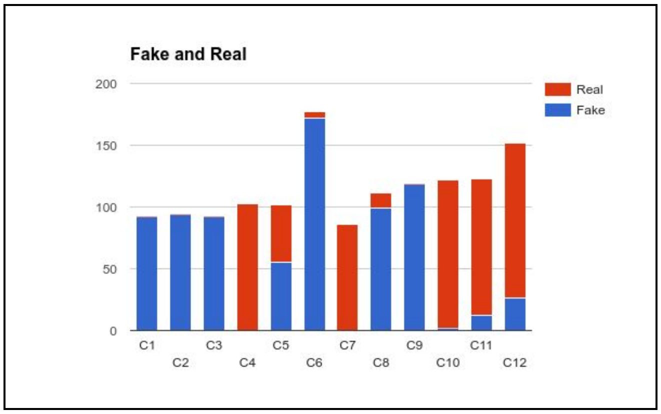


As one can see the clusters do tend to group themselves according to their eventual classification. This is to be expected since more similar websites are more likely to be Phishing Websites or not.

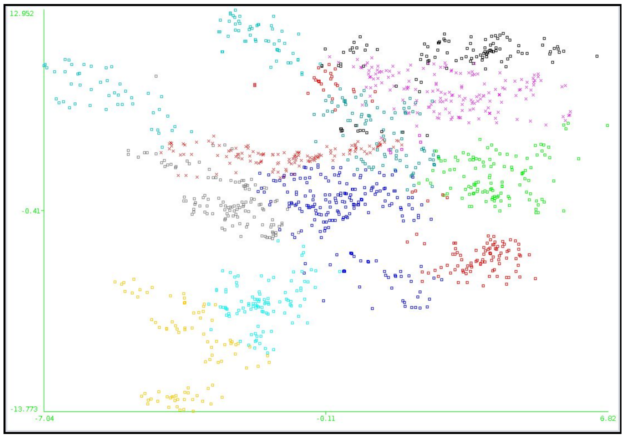
The breakdown of the clusters also was interesting. Even with 14 clusters, one of the clusters ended up having 20% of the instances.

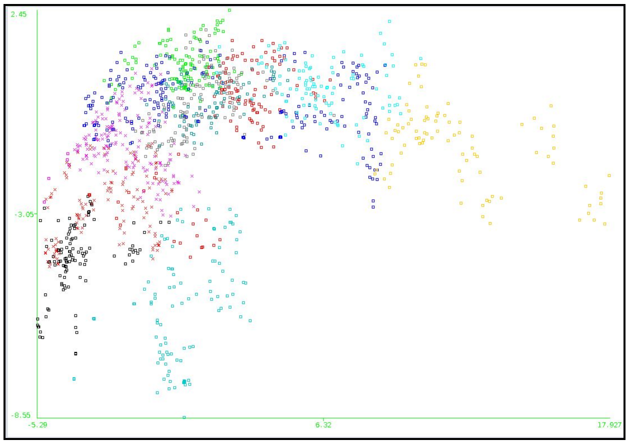
Banknote Data

The same holds true for the banknote data. The clusters that K-Means is able to locate map astonishingly well to a classification. Even better than in the previous dataset with one exception, Cluster 5. Cluster 5 was very difficult to label. This is most likely the cluster of good counterfeits and edge cases.



Since the Banknote data only contains four attributes it is also much easier to visualize the clusters. The first graph represents the first two attributes with the color representing the cluster. The second graph represents the second two attributes. Even with the naked eye it is clear where the clusters are coming from. This is one advantage of K-Means in lower dimensions is that the results are easy to interpret if they can be graphed.





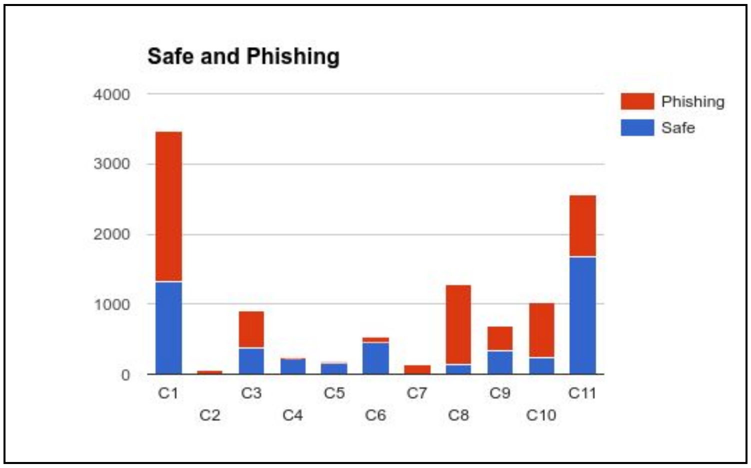
Expectation Maximization: K Selection

For Expectation Maximization K Selection I will be using the default weka implementation to find the ideal K. This process is performed by starting the number of clusters off at one. Performing EM and cross validating the results until the log likelihood no longer increases. For the Banknote Dataset the ideal K was chosen as 21 by the algorithm. For the Phishing dataset the K was chosen as 11.

Expectation Maximization: Clusters

Phishing Dataset

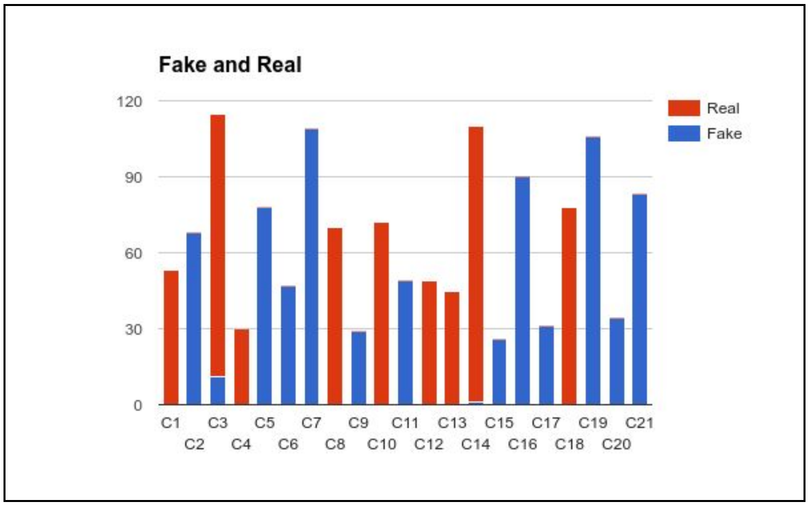
As one can see Expectation Maximization arrives at 11 different clusters and unfortunately these clusters do not align as well with the classes as the K-Means clusters did. There is one exception to this. The small clusters C2 and C7 appear to be made of only



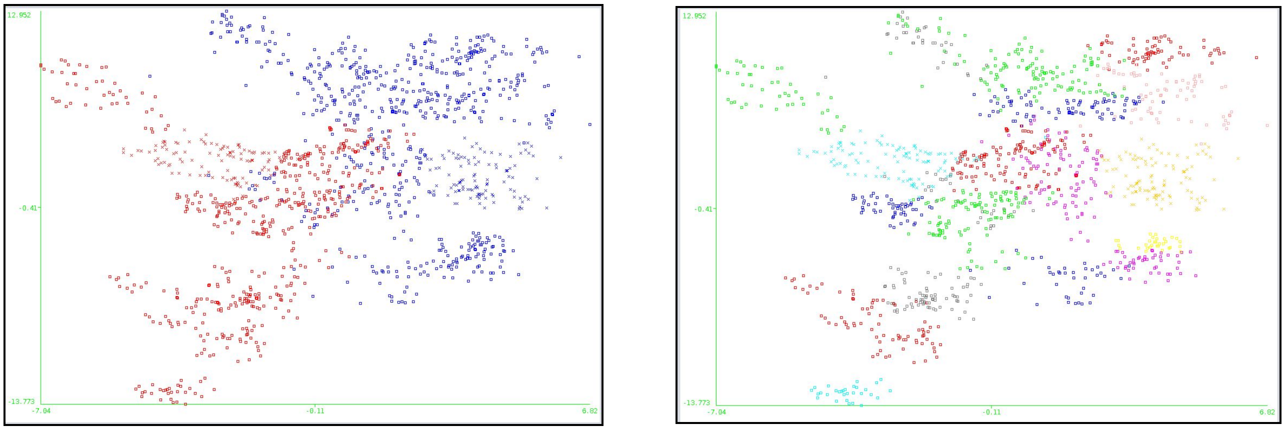
Phishing Websites. I would attribute this to a combination of attributes use in a small subset of Phishing Websites that would not be used in a Non-Phishing website.

I would attribute the fact that the clusters do not align very well to classes because of the large number of attributes. A Phishing Website is often designed to mimic a real website. Thus many of the attributes would be the same and only a few of the attributes would be changed. This would mean that the Euclidean Distance between a phishing and a non-phishing website would be close. This could lead to a phishing website being within a reasonable range for a normal distribution and could be clustered with many legitimate websites.

Banknote Dataset



The banknotes dataset resulted in much more shocking results. As we can see the banknotes clusters almost perfectly aligned with their classification. I would attribute this to the fact that unlike with the previous dataset where there were a large number attributes shared by phishing and non phishing websites, in this dataset there are only four attributes each real valued. In addition each of these attributes were chosen for their ability to differentiate between real and fake notes.



Attribute 1 vs Attribute 2 (color indicates classification and cluster)

As you can see the data seems to be linearly separable by itself. The clusters when grouped in many ways represent a confidence in a notes legitimacy. With C3 representing boundary notes.

Feature Transformations

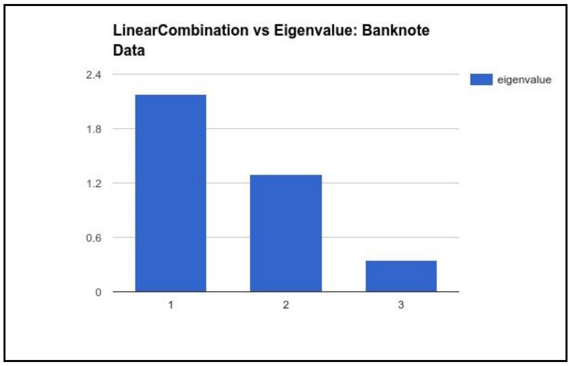
Principle Component Analysis

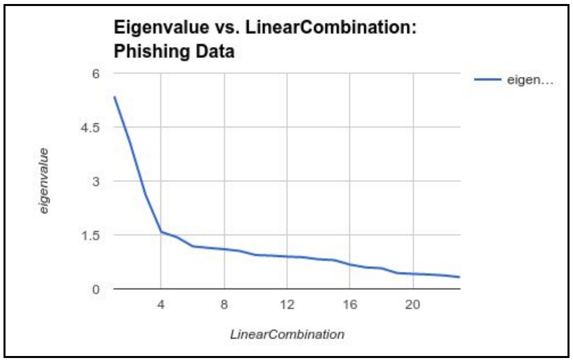
The first feature transformation I performed was Principal Component Analysis which aims do feature reduction by performing linear transformations on the data to maximize variance. By maximizing variance in theory you are able to consider fewer attributes but each attribute is guaranteed to be more informative because PCA maximizes variance.

Datasets

After performing PCA on the two datasets I received some interesting results. For the banknote data, as is to be expected the number of attributes could not be reduced by that many. After analyzing the eigenvalues (2.18, 1.29, 0.35) I decided to select the two former because an eigen value of 0.35 means that Linear Combination was not that informative.

The Phishing data because of its larger number of attributes had more interesting linear combinations. The first being that there were some very informative projections. One with an eigenvalue of 5.36. Then there is a quick falloff and a long tail of combinations. I decided on a cutoff eigenvalue of 1 because with an eigenvalue of less than 1 there is very little information to be differentiate the instances.

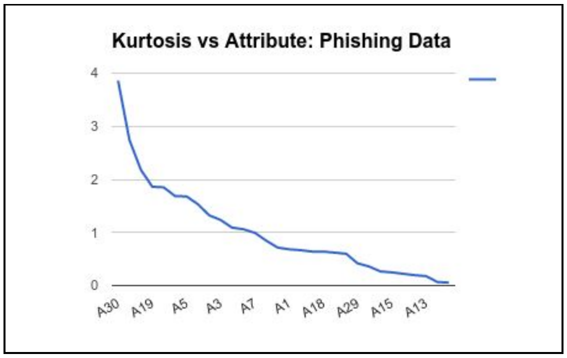


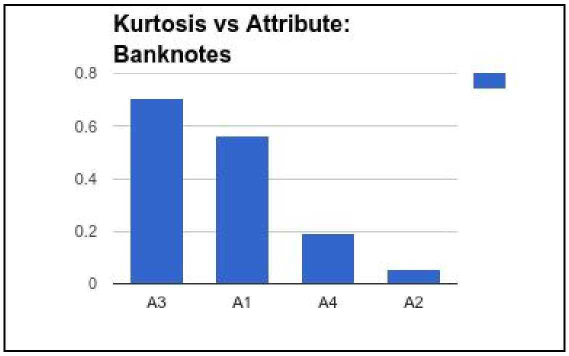


Independent Component Analysis

The second feature transformation that I performed was ICA. ICA attempts to find the hidden attributes which cause the measured attributes. To analyze the results I will be evaluating the projected attributes kurtosis by how far they are from a normal distribution.

Datasets





For the Banknotes data it is clear that the second attribute is very close to a normal distribution and provides little data. The same is true for A4 so I will be using only the 3rd and 4th attributes. When considering the Phishing Dataset the kurtosis makes a nice elbow shape. I chose the cutoff to be at A9 (kurtosis of 0.6 from 3). This leaves me with 14 attributes. Many fewer than the original 30.

CfsSubsetEvalutation

I chose CfsSubsetEvaluluation because it was a feature selection algorithm as opposed the others which perform transformations on the data. The algorithm itself decides the subset of attributes so I did not need to decide how many attributes to include.

Datasets

On the Banknotes dataset CFS found that the ideal subset was the first two attributes. This will provide a good comparison to ICA since I chose two of ICA's projections. CFS on the Phishing Dataset found 9 attributes to keep. Those attributes were {Prefix_Suffix, having_Sub_Domain, SSLfinal_State, Request_URL, URL_of_Anchor, Links_in_tags, SFH, web_traffic, Google_Index}

Feature Projection Clusters

In keeping with the topic of this paper, to analyze the performance of the projections in terms of clustering, I will be analyzing how well the clusters represent the classification. To be more explicit I will assign each cluster a classification based on the mode classification of the instances in that cluster. I will then calculate what percentage of the instances fall within clusters of the same classification. In essence how good are the clusters at predicting the classification of an instance. This accuracy will give an insight into how well those clusters will be when performing supervised learning in the later section.

The reason I am performing this calculation is because for the remainder of the paper I will be looking at unsupervised learning as a way to improve the performance (accuracy, space efficiency, time efficiency) of supervised learning algorithms.

For the randomized projections of the data each each dataset was projected randomly three time and the results were then averaged.

Banknote Data

EM

With expectation maximization I will admit I was disappointed in the results of the projections at least when compared to their baseline. The unprojected clusters were able to predict with an accuracy of 99.1%.

ICA performed abysmally with an accuracy just over random. This seems to indicate that the either the banknote data does not have hidden variable or that the hidden variables do not lend themselves well to predicting fraudulent banknotes. This is supported by the kurtosis measurements in the previous section.

PCA was able to perform slightly better with an accuracy of 70.2%. This was no better than the average randomized projection. This indicates to me that maximizing the variance does not in turn help to create clusters that represent the classifications better than just a random projection.

CFS was the real standout with an accuracy of 90.8% with half the number of attributes as the original data. This means that most of the data relevant to classification is held in the first two attributes. This means one could have high supervised learning accuracies by only collection the first two attributes which could save time and perhaps money in the long run depending on where the instances are coming from.

	Accuracy	Attributes	Clusters
Unprojected	0.991	4	11
ICA	0.635	2	14
PCA	0.702	2	4
CFS	0.908	2	11
RP_AVG	0.7226666667	2	11.33333333

The K-Means data followed the same trends so the chart was omitted to save space but is included in the accompanying spreadsheets.

Phishing Data

EM

This projected data yielded more interesting results for the Phishing Dataset. As we can see the accuracy of the unprojected data was a solid but not outstanding 81.9% accuracy.

ICA performed well. With a reduced number of attributes (20), the clusters were able to achieve only a slightly lower level of differentiation from the original. This gives me hope that when we train the NN, ICA will be able to perform well.

PCA performed even better than ICA by achieving similar levels of accuracy with half the number of attributes that ICA used. However, as with the last dataset, Randomized Projections was able to do almost as well as PCA. I attribute this to the fact that increased variance does not mean that the cluster will better align with the important metrics as opposed to the less important metrics.

That point is what caused CFS to perform so well. Since CFS is a feature selection algorithm based on the more informative features, it is able to select which attributes are more important for classification. Since EM and K-Means do not take into consideration the importance of attributes the clusters group on both unimportant and important attributes. When unimportant attributes are removed the clusters are better aligned with classifications. This causes CFS to create clusters that better align with classifications than even the unprojected data.

	Accuracy	Attributes	Clusters
Unaltered	0.819	30	11
ICA	0.756	20	20
PCA	0.746	9	20
CFS	0.884	9	9
RP_AVG	0.734	9	13

As with the previous dataset, the results I obtained from K-Means are included in the accompanying spreadsheets. They are not included or discussed here because the results followed the same trends as the EM data.

Performance of Neural Networks on Projected Phishing Data

I decided to train my neural networks on the Phishing data for consistency seeing as I used the Phishing data on the two previous assignments as well. The results were collected by running the datasets through weka. 10 fold cross validation was performed to ensure reliability. The parameters were played with to try and achieve the lowest error.

The baseline for this is of course the NN trained on the Unprojected dataset. This was able to achieve an accuracy of 96.02% with the original 30 attributes and a relatively long training time of 56 seconds.

	Accuracy	Attributes	Training Time
Unprojected	96.02	30	56.18
ICA	93.27	20	28.94
PCA	92.7	9	10.03
CFS	93.65	9	9.3
RP	85.11	9	9.67
KMeans: Clusters	81.68	1	15.64
EM: Clusters	68.16	1	10

Each of the feature reduction algorithms performed worse than the unreduced version when it came to accuracy. This is only one part of the story since ability to predict on our training set is only one measure of success.

The first thing that must be taken into account when considering the results of the feature transformation algorithms is the curse of dimensionality. As talked about in class, as the number of features increases linearly, the amount of data needed to train the model increases exponentially. Fortunately in our case, as talked about in the first paper, there is enough data to properly learn the underlying model. However, it is easy to envision a case where we have a smaller amount of data and the reduced attribute sets provide a larger advantage.

The amount of data needed is also only one part of the story. Training time is also a factor to consider. The original dataset took almost one minute to train. ICA took 30 seconds. PCA and CFS took only ~10 seconds. As the number of attributes decreased, The time to train decreased even faster. The accuracy only took a small drop. With CFS, the smaller number of attributes would offer easier data collection; achieving similar results while only having to collect ⅓ the number of attributes.

With the clustering algorithms K-Means was easily able to outperform EM. I attribute this to the interesting problem of applying EM to only boolean attributes because EM performs better with more fleshed out probability distributions such as those found in the Banknotes Dataset. As a whole however, the clustering algorithms were able to achieve a high accuracy in an information dense manner. It's not often talked amount but there are scenarios in which the size of an instance could play a part. K-Means was able to achieve 81.68% accuracy with each instance able to be stored in 4 bits, as opposed to the original 30 bits.

Summary

One of the major applications of unsupervised learning is to assist in later supervised learning. The benefits of which can not be overstated. Unsupervised learning allows one to overcome the curse of dimensionality by reducing the number of attributes. With feature selections algorithms making data collection easier as well. Fewer attributes also allows for faster training time and smaller storage requirements.