An Analysis of Unsupervised Learning and Dimensionality Reduction

In this analysis, we will be looking at unsupervised learning algorithms. These algorithms are used for drawing inferences from unlabeled data. The most common method of unsupervised learning is clustering. Two types of clustering will be reviewed here: K-means clustering (KM) and Expectation Maximization (EM). K-means clustering works by partitioning the data into k clusters into which each observation belongs to the cluster with the nearest mean. The mean, or centroid, of the cluster serves as a prototype of the cluster. Expectation Maximization is similar, but builds its clusters based on prior probability distributions.

We will also look at four dimensionality reduction algorithms (DR): Principal Component Analysis (PCA), Independent Component Analysis (ICA), Randomized Projections (RP), and feature selection by information gain (IG). The experiments above will be performed on two datasets: the Iris dataset I used in Assignment #1, and the Spambase dataset I used in Assignment #2. We will discuss the performance and characteristics exhibited by the above algorithms, the influence of DR on clustering, and the influence of DR on neural net training. For this analysis, all algorithms were run using WEKA 3.8.1. Unless otherwise specified, all parameters are the default parameters in WEKA.

Datasets

The Iris dataset contains 150 instances with 4 attributes, belonging to 3 classes of Iris plant. This provides a simple and well-behaved dataset that will show the expected behavior of the above algorithms. The Spambase dataset is more complex. It contains 4601 instances with 57 attributes, classifies as either spam or not-spam. This is a relatable problem that should accentuate the effects of size on how the clustering algorithms perform.

Methodology

I. Clustering

KM and EM will be applied to the Iris and Spambase datasets. The number of clusters to be built was varied for both datasets. The number of clusters used for Iris was in the range (3, 6) and in the range (2, 20) for Spambase. These are both inclusive. For KM we use the Euclidean distance function. Euclidian distance is the shortest distance along the hyperplane containing the two points of interest.

II. Dimensionality Reduction

Again, PCA, ICA, RP, and IG will be used for dimensionality reduction on the datasets. The parameters were varied according to the table below:

	PCA	ICA	RP	IG
Parameter	Total variance	N/A	Number of Attributes	
Value(s)	{0.75, 0.9, 1}	N/A	Iris: {1, 2, 3} Spambase: {5, 15, 20, 30, 40}	

III. Clustering After Dimensionality Reduction

KM and EM will be applied again to dimensionally reduced datasets. We will use the same number of clusters for each dataset as outlined above.

III. Neural Net Training after Dimensionality Reduction

We will compare a typical neural net's performance on the unreduced datasets to each reduced by KM and EM. The results of cross validation will be compared to the clustering analysis. We select the number of hidden layers using $H = \frac{a+c}{2}$, where a is the number of attributes and c is the number of classes. The best parameters from each will be selected and compared. I used a 70/30 split for training. For our typical neural net, we set the following parameters:

Learning Rate (L)	Momentum (M)	Epochs (E)	Number Hidden Nodes (H)
0.3	0.2	500	Iris: 4 Spambase: 30

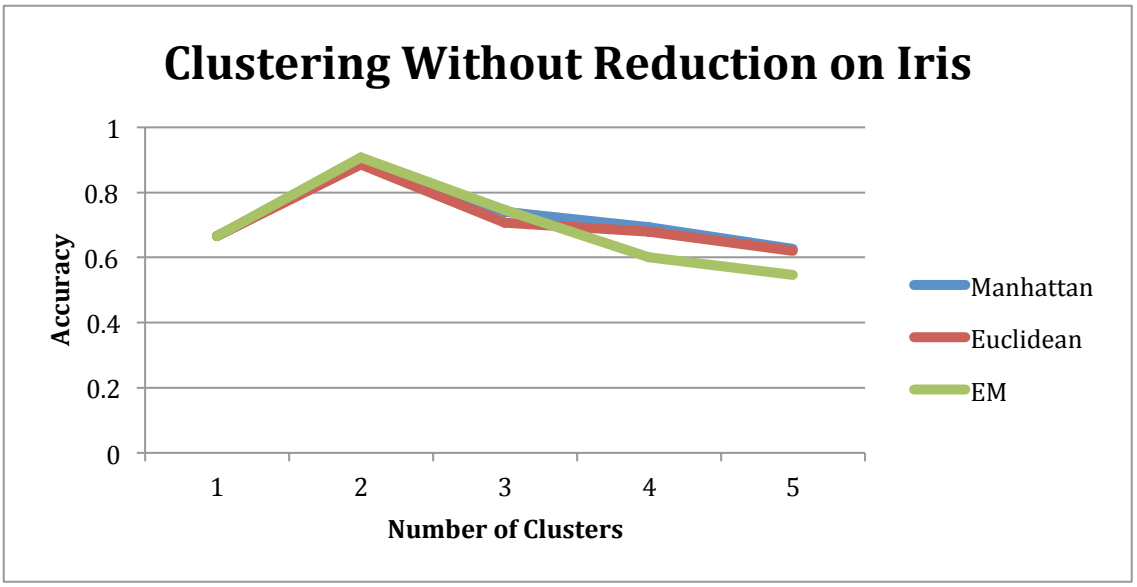
Results

I. Benchmark

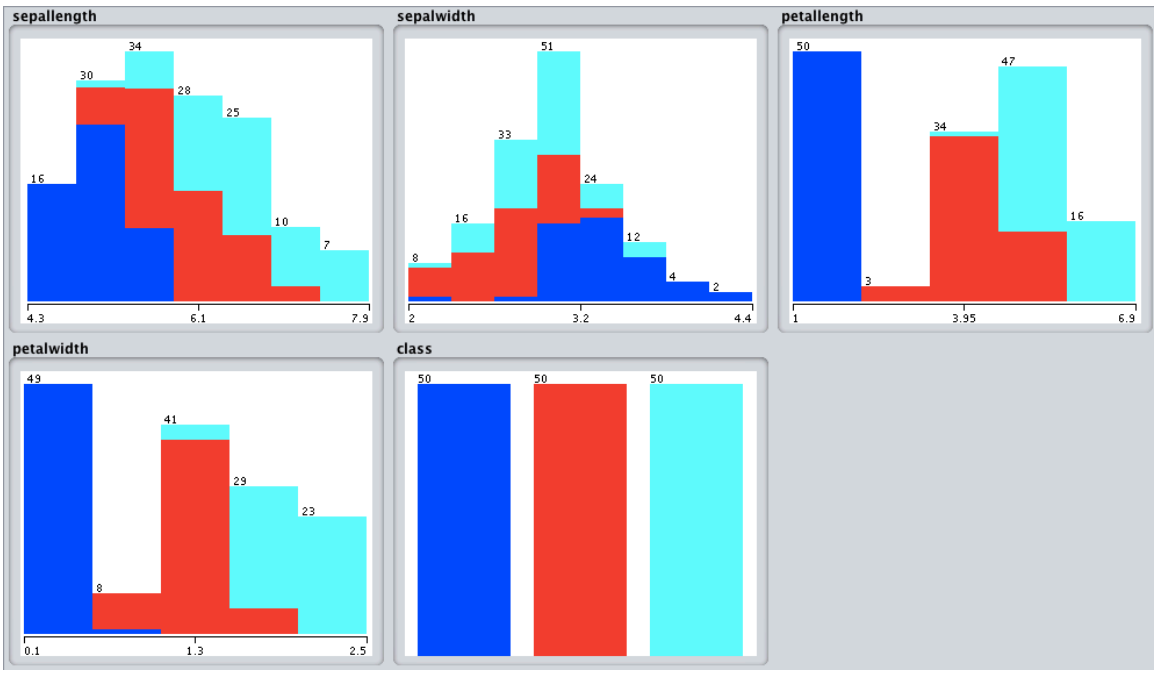
Our benchmark neural net completed training in 98.7 seconds with 92.174% accuracy. This is the standard we will compare our clustering and dimensionality reduction algorithms to.

I. Clustering without Dimensionality Reduction

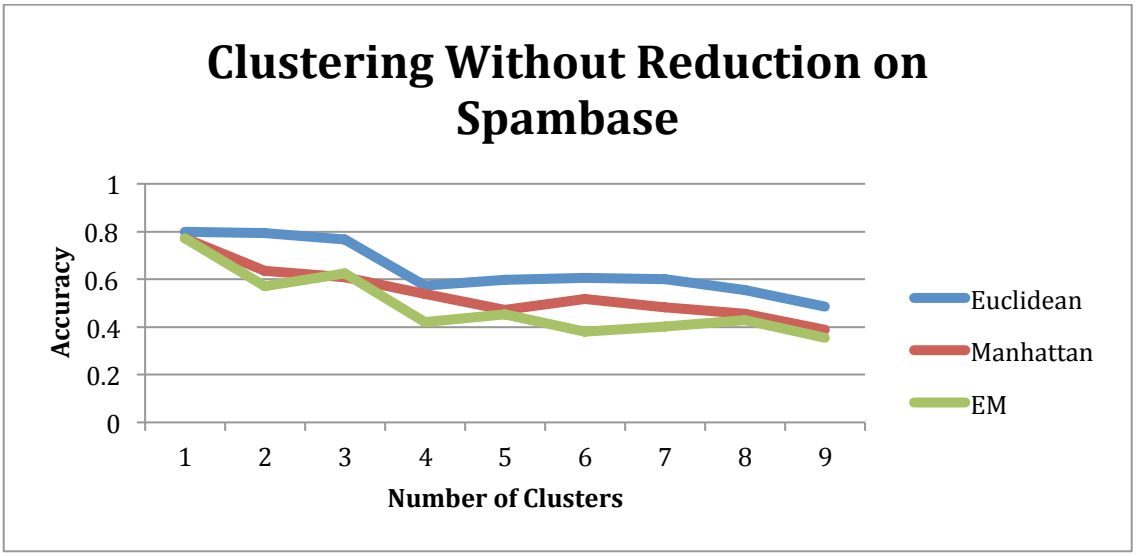
The performance of KM and EM on the Iris dataset is shown below. It is evident that the clustering does not perform naturally well on this dataset as compared to our benchmark. However, it does take much less time, with each algorithm taking less than second.



The graph would seem to suggest that the algorithms perform very similarly, however the similarities are most likely due to the simplicity of the Iris dataset. With only 150 instances, 4 attributes, and 3 classes, the Iris dataset does not push these clustering algorithms. It is apparent that if we were to continue to increase the number of clusters, then the accuracy would drop near 50%. The figure below



shows the distribution of the iris dataset. It is clear there is significant similarity between classifications. This leads to both clustering algorithms doing poorly on this dataset.



Clustering suffers on Spambase as well. However, unlike Wine, the behavior is more differentiable. KM with a Euclidean distance function performs the best. However, it still does not compare to our benchmark neural net. As the number of clusters increases, the accuracy decreases, but the sum of the squared error decreases. The clusters get more accurate themselves, but they do not do a better job of representing the data.

II. Component Analysis

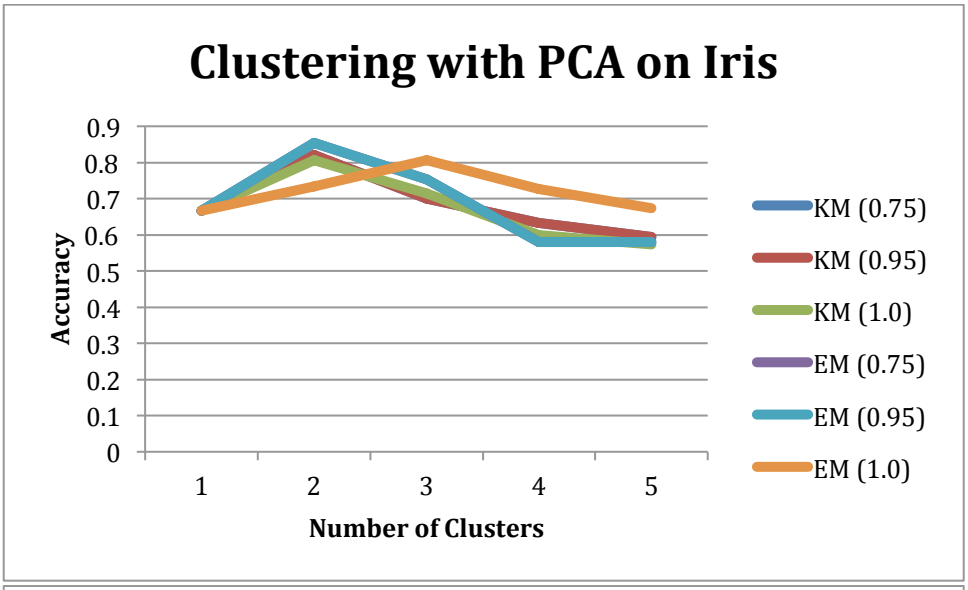
PCA uses orthogonal transformation, which maximizes the total variance, to convert the attributes into linearly independent variables. PCA does not work on discrete data types. Thus, PCA should help both datasets. PCA covers 95% of the variance in Iris with 2 attributes. However, PCA requires 48 attributes to cover 95% of the variance in Spambase. PCA is expected to enhance Iris.

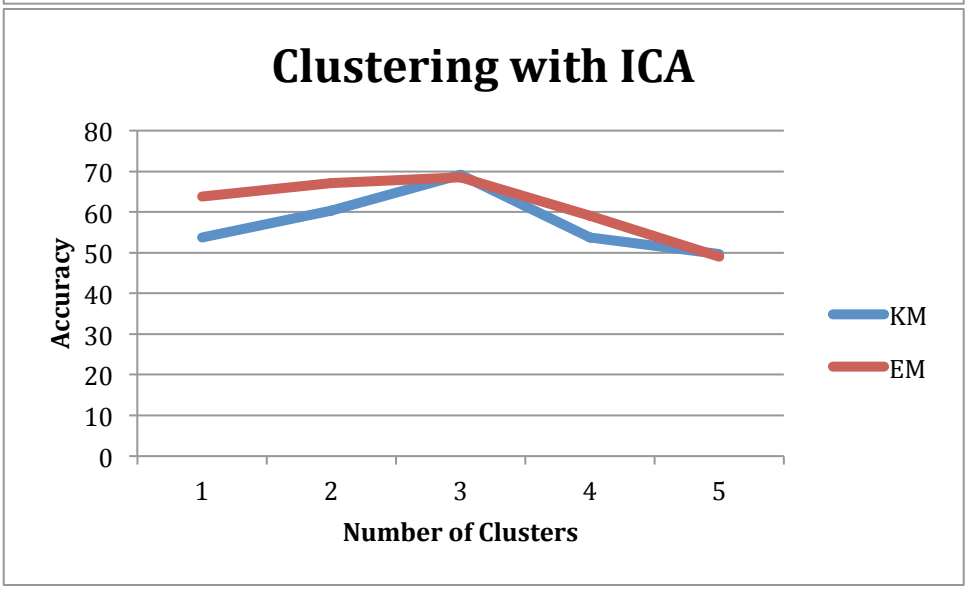
ICA separates multivariate data into additive subcomponents, with the assumption that all attributes are non-Gaussian. Performance is measured by the kurtosis value. In a normal distribution, kurtosis should be equal to 3. The kurtosis values for the Iris database are [3, 2, 2] and have a kurtosis sum = 2. This means the new distribution is similar to a Gaussian. For the Spambase database the kurtosis values is an array with length 56, and a kurtosis sum of 21. Spambase is not well represented by ICA

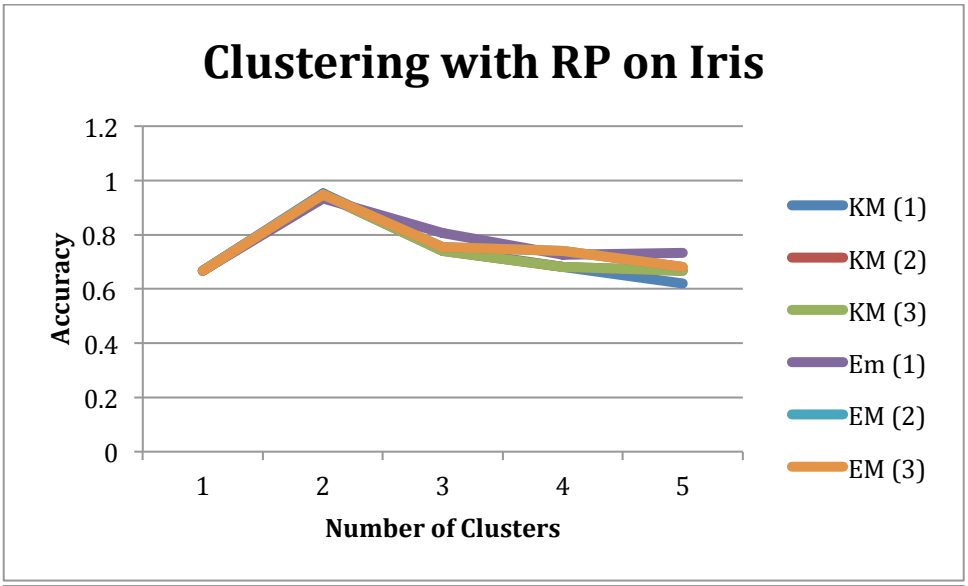
RP reduces the dimensionality of random variables into a suitable lower-dimensionality space. IG is a supervised algorithm that ranks attributes based on Gain. We retained a set number of attributes for both.

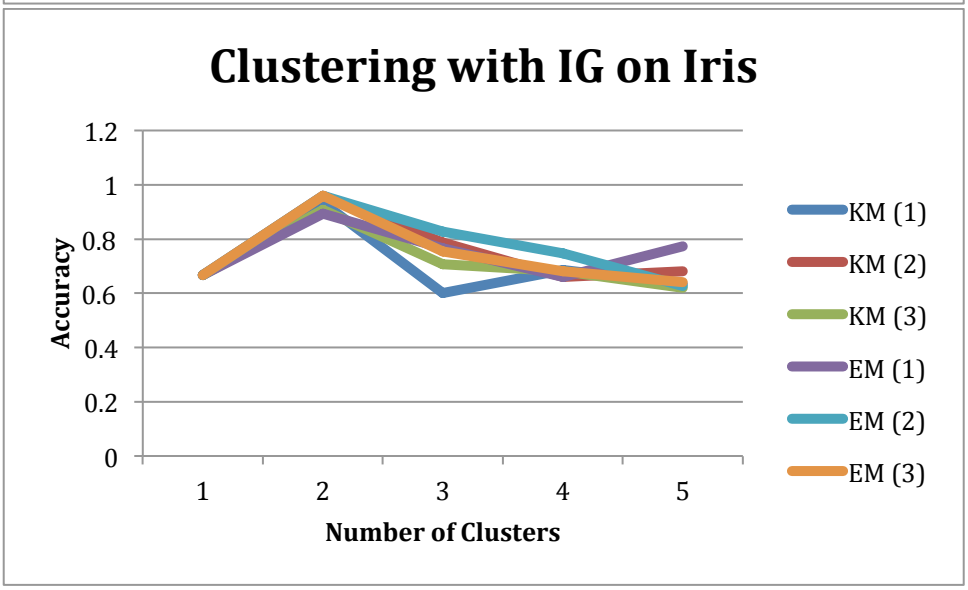
III Clustering After Dimensionality Reduction

KM with Euclidean distance and EM clustering algorithms is applied to the dimensionally reduced datasets. On the Iris Dataset, 2 clusters do very well, especially for RP and IG. It is almost comparable to our benchmark neural net. Performance trends upwards to 2 clusters, then drops. As Number of clusters passes two it is feeding too much information and loses accuracy.

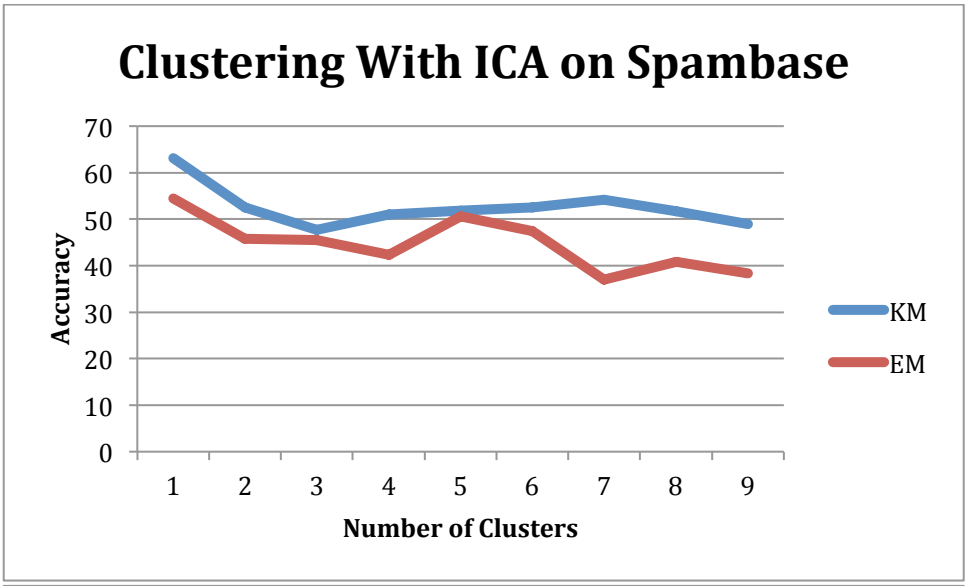


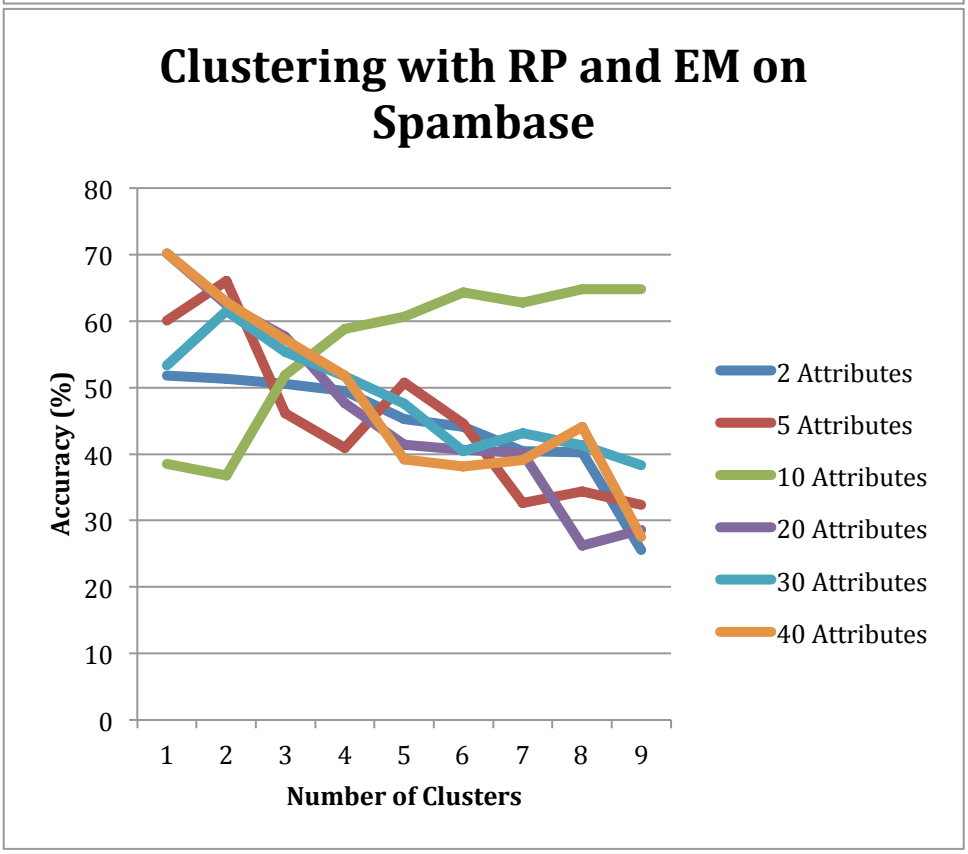


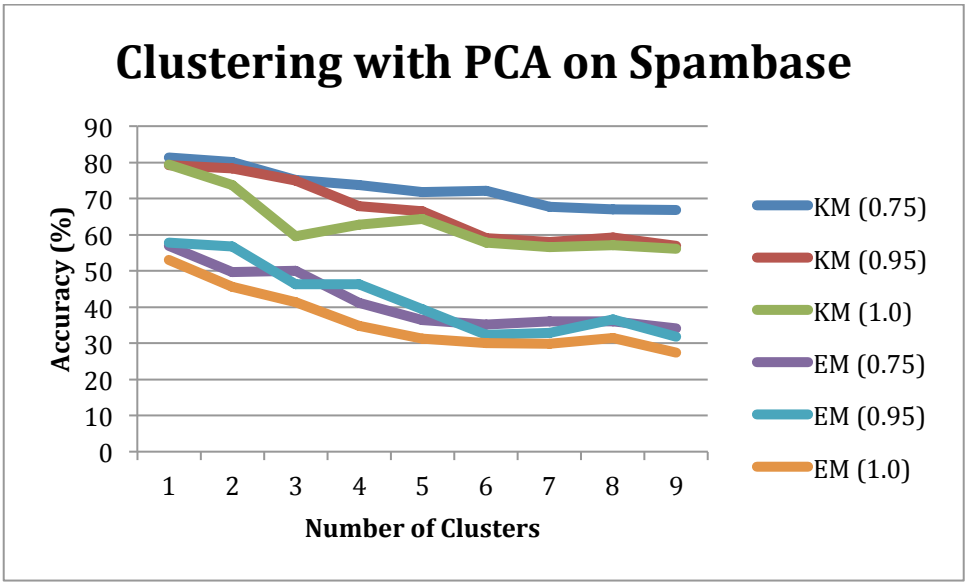


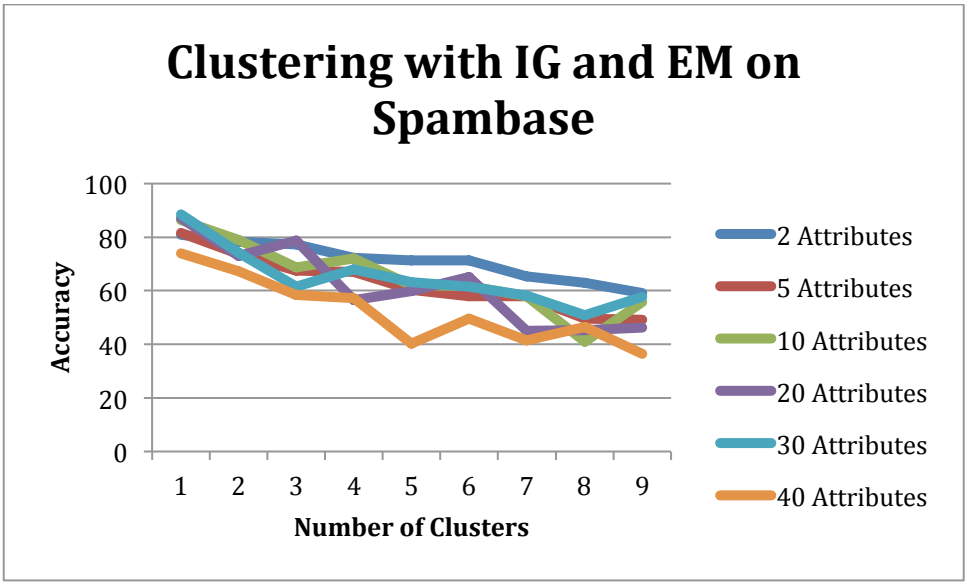


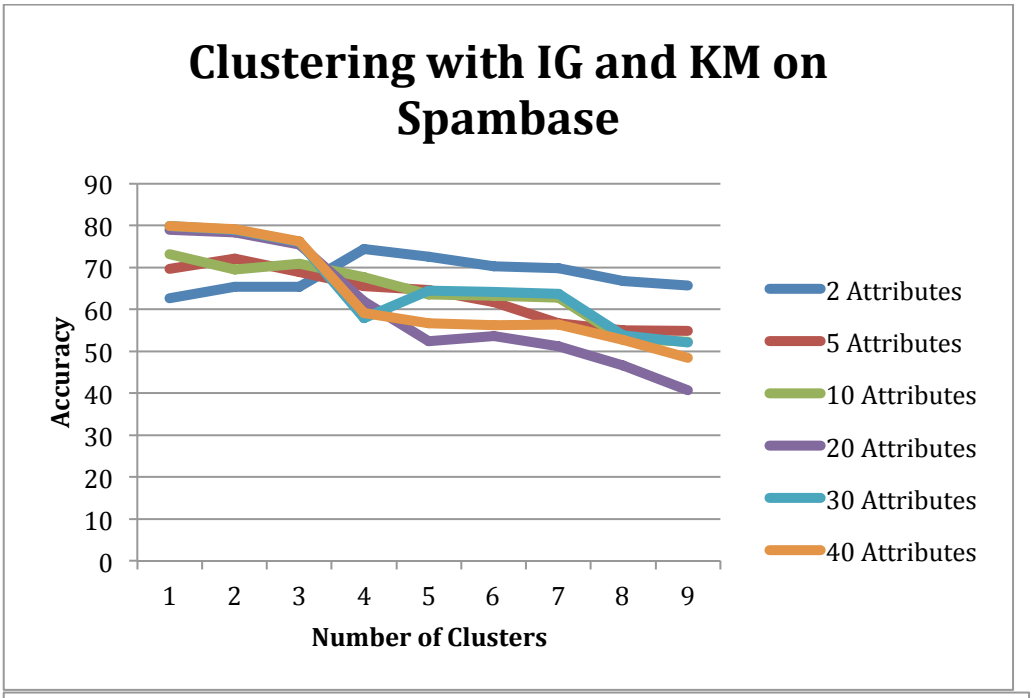
The Spambase dataset is dimensionally reduced by the same processes as Iris. Due to Spambase's nature, DR will probably not have a substantial effect on Spambase. If the previous trend continues, then we can expect spambase accuracy to be best at the outset. It is interesting to note that for clustering with RP and EM, using 10 attributes managed to have a positive slop, being the only clustering method to for spambase.

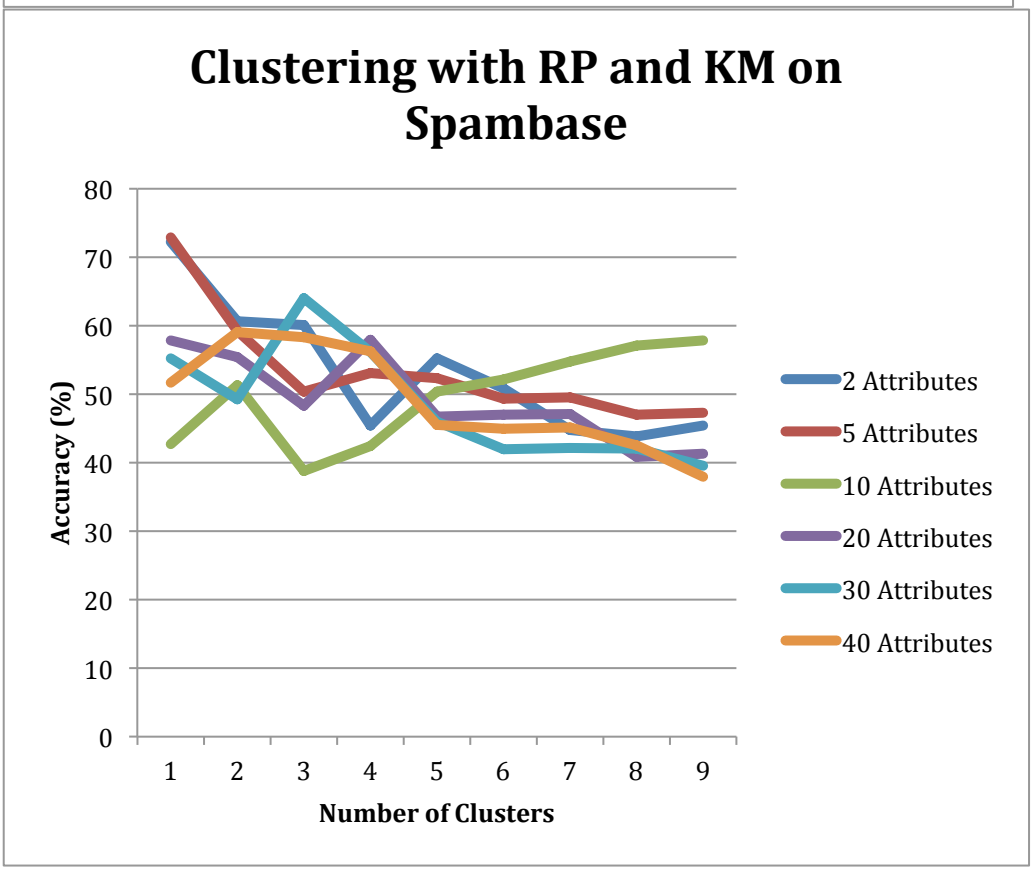

Clustering With ICA on Spambase
Accuracy
70
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
9
Number of Clusters
KM
EM


Clustering with RP and EM on Spambase
Accuracy (%)
80
70
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
9
Number of Clusters
2 Attributes
5 Attributes
10 Attributes
20 Attributes
30 Attributes
40 Attributes


Clustering with PCA on Spambase
Accuracy (%)
90
80
70
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
9
Number of Clusters
KM (0.75)
KM (0.95)
KM (1.0)
EM (0.75)
EM (0.95)
EM (1.0)


Clustering with IG and EM on Spambase
Accuracy
100
80
60
40
20
0
1
2
3
4
5
6
7
8
9
Number of Clusters
2 Attributes
5 Attributes
10 Attributes
20 Attributes
30 Attributes
40 Attributes



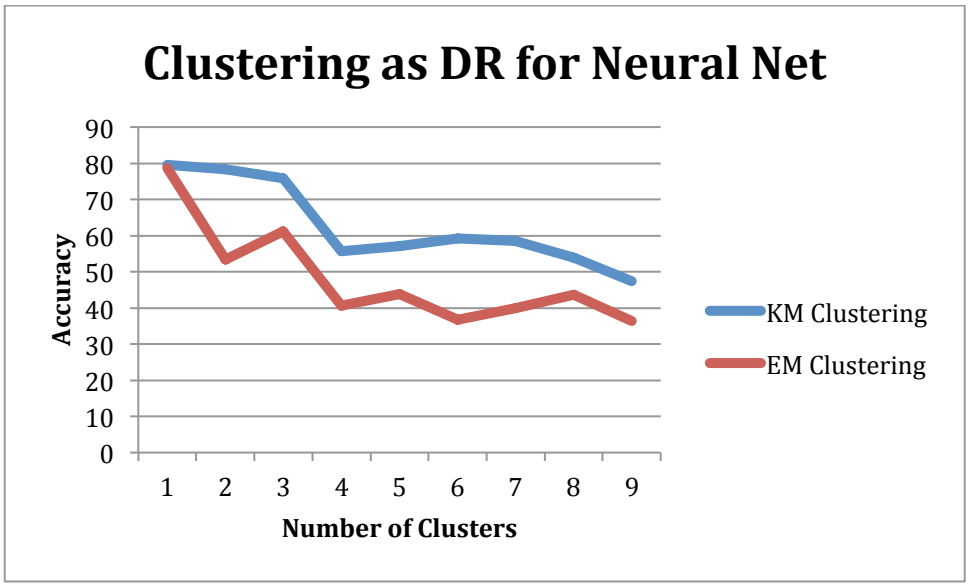


It is shown that RP with 10 attribute performs the best for spambase, but only barely.

IV Neural Network Learning After DR

NN Training After DR on Spambase												
Benchmark NN												
NN Acuracy	92.174	Training Time	98.7									
PCA												
Variances	0.75		0.95		1							
NN Accuracy/Training Time	92.5362	41.24	91.8116	82	91.2319	97.01						
ICA												
NN Accuracy/Training Time	92											
RP												
# Attributes	2		5		10		20		30		40	
NN Accuracy/Training Time	69.7826	2.21	78.9855	3.22	83.7681	6.85	81.3043	18.19	87.3913	33.22	85.2174	53.82
IG												
# Attributes	2		5		10		20		30		40	
NN Accuracy/Training Time	83.1159	2.13	84.7826	3.23	90.2174	7.08	91.8841	17.26	91.5942	32.52	91.7391	53.07

V Neural Net Learning After Clustering



The advantages to dimensionality reduction in NN comes from the ability to simplify the data and run the neural net faster.