Unsupervised Learning and Dimensionality Reduction Techniques

Swati Prasad
Cs7641: Machine Learning

"Ownership of the following report developed as a result of assigned institutional effort, an assignment of the CS 7641 Machine Learning course shall reside with GT and the instructors of this class. If the document is released into the public domain, it will violate the GT honor code."

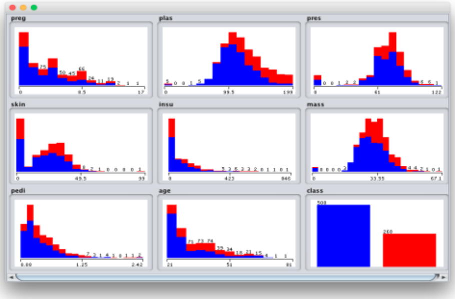
Introduction:

The purpose of this assignment is to explore two clustering algorithms K means and Expected maximization (EM) and four dimensionality reduction techniques PCA, ICA, Random Projection and Information Gain. There are three sections to implement this. First section is to implement clustering techniques to two datasets. Second section applying dimensionality reduction and reapplying clustering techniques. Third section applying dimensionality reduction and reapplying clustering techniques to neural network.

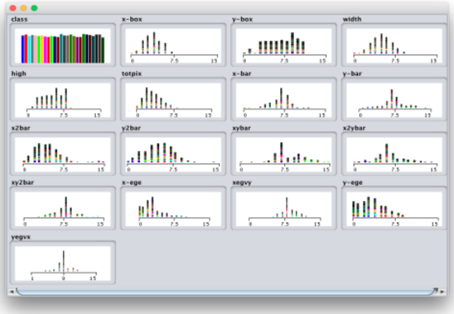
Dataset Information:

Both datasets were analyzed in assignment 1.in WEKA. Here is an excerpt:

Fig(1) Pima Indian Analysis



Fig(2) Letter Recognition Analysis



Dataset -1: Pima Indian Diabetes Dataset

Data Characteristics:

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective is to predict based on diagnostic measurements whether a patient has diabetes. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage. All values are numeric and is classification problem.

From fig(1) Number of Instances = 768. age : Sample population is young and 708/768 instance's data has age < 50. 0 value indicates error or missing data. Plas, pres, mass has errors. There are 500 (65%) positive instances and 258 (35%) negative instances. After reviewing histogram of attributes in weka, Few attributes are normally distributed (plas, pres, mass and skin). Few attributes are exponentially distributed (preg, insu, pedi, age). There is no direct relation between any attribute and diabetes.

<u>Interesting Reason:</u>

Vast amount of data available in health care industry is difficult to handle, hence mining is necessary to find the necessary pattern and relationship among the features available. Medical data mining is one major research area where evolutionary algorithms and clustering algorithms play a vital role. K-Means is used for removing the noisy data and dimensionality reduction technique to further filter relevant attributes. The experimental result proves that, the proposed model has attained an average accuracy of 98.79% for reduced dataset of Pima Indians Diabetes from UCI repository. Dataset is interesting because it is simple dataset to analyze the clustering and dimensionality reduction.

Dataset-2: Letter Recognition

a) Data Characteristics:

Number of Instances: 20000 & Number of Attributes: 17 (Letter category and 16 numeric features). There are no missing values. After reviewing histogram of attributes in weka, Distribution of Sample population is good. Many attributes are normally distributed. e.g x-box, width, x-bar, y-bar , x2bar, y2bar, x2ybar, xy2bar, xegvy. Sample data has even distribution of 26 letter.

Interesting Reason:

The objective is to identify each of a large number of black-and-white rectangular pixel displays as one of the 26 capital letters in the English alphabet. The character images were based on 20 different fonts and each letter within these 20 fonts was randomly distorted to produce a file of 20,000 unique stimuli. Each stimulus was converted into 16 primitive numerical attributes (statistical moments and edge counts) which were then scaled to fit into a range of integer values from 0 through 15. Many industries such as healthcare, finance, law and construction have used optical character recognition(OCR) to help the paperwork reduction, process improvement and task automation. A study of the accuracy of OCR is important to the computer vision industry, as it is more mature and can be used as a guide when developing for more difficult sub-domains like video tracking and object recognition. It is interesting with respect to machine learning because the numeric features and equally occurring classes make letter recognition a good candidate for using clustering and dimensionality to train a neural network. It is also multiclass dataset, compared to previous one was binary class dataset. It is intuitive from the given dataset, 26 clusters should be appropriate for clustering. This will help to verify our experiment results.

Dataset is interesting because numeric features, even distribution of 26 letters And normal distribution of sample population makes interesting and good candidate for learning clustering technique and dimensionality reduction.

Part 1

Multivariate analysis is set of useful methods for analyzing data when there are more than one variable under consideration. Multivariate analysis techniques may be used for several purposes, such as dimension reduction, clustering or classification. Here we are going to explore on clustering techniques and in later part dimensionality reduction.

Clustering Techniques

Two algorithms were examined in clustering technique.

a) K mean Clustering:

The goal of clustering analysis is to establish a set of meaningful groups of similar objects by investigating relationships between objects.

K-means clustering is a type of unsupervised learning, which is used when you have unlabeled data (i.e., data without defined categories or groups). The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable K. The algorithm works iteratively to assign each data point to one of K groups based on the features that are provided. Data points are clustered based on feature similarity. The results of the K-means clustering algorithm are: a) The Centroid of the K clusters, which can be used to label new data. B) Labels for the training data (each data point is assigned to a single cluster)

Distance Measurement in K-means Algorithms:

In K-means algorithm, we calculate the distance between each point of the dataset to every centroid initialized. Based on the values found, points are assigned to the centroid with minimum distance. Hence, this distance plays the vital role in the clustering algorithm.

But, in choosing such techniques, some important points to be noted such as, the property of the data and dimension of the dataset. In this experiment, we take "cityblock", "Euclidean" distance measurement techniques for distance calculation in K-means algorithm. Description about each techniques is mentioned below

City Block Manhattan:The city block distance two points a and b with k dimensions is defined as:

sigma(j=1..k |aj - bj|

Euclidian Distance:The euclidian distance between two points a and b with k dimensions as calculated:
root(sigma(j=1..k (aj-bj)^2

b) Expected Maximization: The concept of the EM algorithm stems from the Gaussian mixture model(GMM). The GMM method is one way to improve the density of a given set of sample data modelled as a function of the probability density of a single-density estimation method with multiple Gaussian probability density function to model the distribution of the data. In general, obtain the estimated parameters of each Gaussian blend component if given a sample data set of the log-likelihood of the data, the maximum is determined by the EM algorithm to estimate the optimal model.
Principally, the EM clustering method uses following algorithm:
Input: Cluster number k, a database, stopping tolerance
Output: A set of k-clusters with weight that maximize log-likelihood function.

1. Expectation step: For each database record x, compute the membership probability of x in each cluster h=1,...,k
2. Maximization Step: Update mixture model parameter(probability weight).
3. Stopping criteria: If stopping criteria are satisfied stop, else set j=j+1 and go to (1).

The iterative EM algorithm uses a random variable and eventually is a general method to find the optimal parameters of the hidden distribution function from the given data, when the data are incomplete or has missing values.
EM clusters and K-means were applied for quality assessment of pima indian diabetes using WEKA and the performance of two algorithms is compared based on logistic classification using the data set.

Finding K (Number of clusters) : To find the number of clusters in the data, need to run the K-means clustering algorithm for a range of K values and compare the results. In general, there is no method for determining exact value of K, but an accurate estimate can be obtained using the following techniques.
One of the metrics that is commonly used to compare results across different values of K is the mean distance between data points and their cluster centroid. Since increasing the number of clusters will always reduce the distance to data points, increasing K will always decrease this metric, to the extreme of reaching zero when K is the same as the number of data points. Instead, mean distance to the centroid as a function of K is plotted and the "**elbow point**," where the rate of decrease sharply shifts, can be used to roughly determine K.

Cluster Analysis on Dataset-1 (K means and EM)

From fig(1) it is clear that as number of clusters increases, SSE(sum of square error) decreases. From cluster 1 to 2, SSE has decreased significantly, after 6 clusters it is almost constant. The k-means chart shows that the nature of graph for both Euclidian and Manhattan are quite similar in value. But Manhattan SSE is greater than Euclidian. There is no significant effect of suing see=10,20,30 on SS value.
In fig(2) Average std dev is calculated for dataset. It is clear that standard deviation has decreased in KM and EM compared to original. Around 6 clusters std dev decreases significantly , but after it starts increasing which shows over fitting of data. From fig(3), using elbow method it is clear that cluster appear to be 2 or 7. Cluster = 2 matches with with our dataset , cluster = 7 is displaying overfitting. As we have total 8 attributes and it may try to create 1 cluster/attributes.

From fig(4) For EM, different seed locations shows similar results. Log likelihood increases over some ranges of k, which is expected. These observations can be explained be the notion that EM uses probabilities (which work most effectively on numeric features) and our dataset is numeric.

From fig(5) and fig(6) it is clear that EM takes much more time compared to K means, which is expected. EM has much more mathematical calculation compared to KM. Time also increases with increasing clusters, which also make sense in terms of computation. Different seeds doesn't affect time for both EM and KM. Fig(7) is cluster analysis, it is evident that solid blue line reveals that having no diabetes criteria is more clearer.

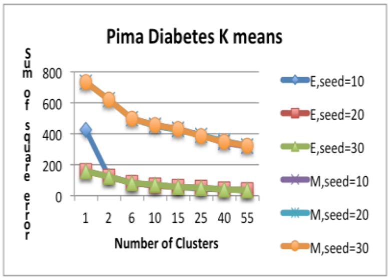
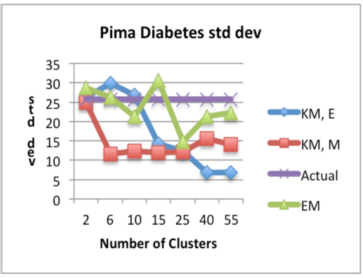
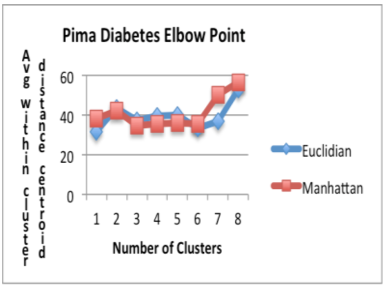


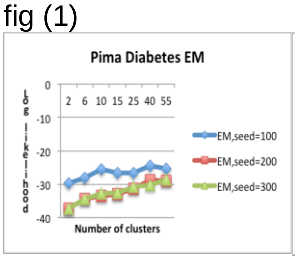
fig (1)



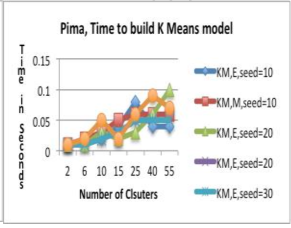
fig(2)



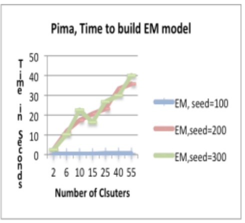
fig(3)



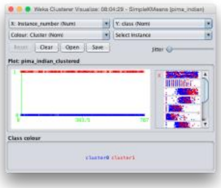
fig(4)



fig(5)



fig(6)



Weka Clusterer Visualize: 08:04:29 - SimpleKMeans (pima_indian)

X: Instance_number (Num)

Y: class (Nom)

Colour: Cluster (Nom)

Select Instance

Plot: pima_indian_clustered

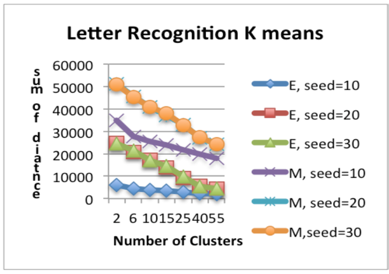
Class colour

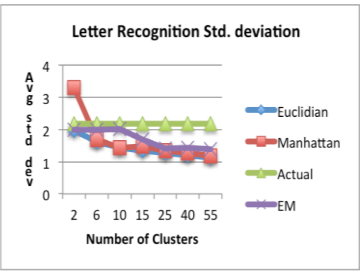
cluster0 cluster1

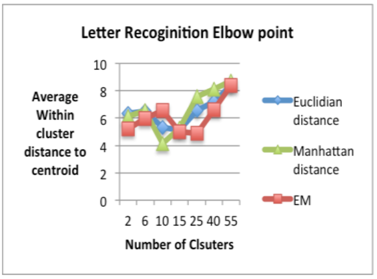
fig(7)

Cluster Analysis on Dataset-2 (Letter Recognition): Both distance functions are implemented in k means to evaluate the performance. From fig(1) both distance function has same curve nature but manhattan distance function has larger SS. This is expected as Manhattan sum the distance in each direction while Euclidian is the shortest distance. From fig(2) it is clear that with increasing clusters std dev is decreasing and after k=26 it is almost constant. it is evident that standard deviation was high for very small clusters but as number of cluster increases, data points gets located around its centroid and hence std deviation decreases. Std. dev. Is calculated based on avg std dev of all clusters.

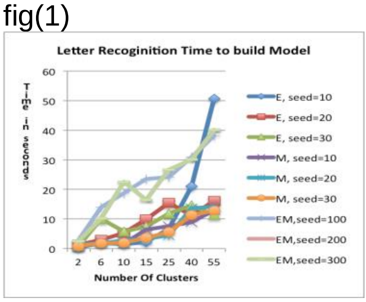
From fig(3) Using the elbow method K value is 26 for both Euclidian and manhattan. There is local minima at k=15, but then k=26 gives clear segregation. From fig(5) it is clear that creating euclidian model takes more time than manhattan model , it make sense manhattan distance calculation is simple compared to Euclidian distance calculation for large dataset. From fig (5) log likelihood increases with increasing clusters. Different seeds doesn't effect the curve and overlapped with each other. Cluster is visualized using fig(6) in K means with K=26. The graph is visualized as X axis with 20000 instances and 26 clusters on yaxis, where color represented letters A-Z. The color is solid or scattered, solid color indicates cluster with class.

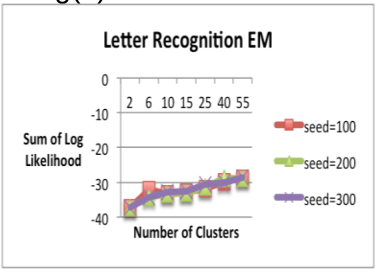


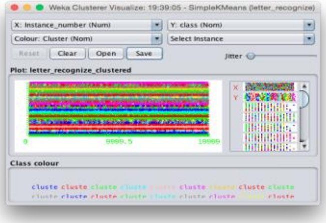




fig(1) fig(2) fig(3)






Weka Clusterer Visualize: 19:39:05 - SimpleKMeans (letter_recognize)

X: Instance_number (Num)

Y: class (Nom)

Colour: Cluster (Nom)

Select Instance

Plot: letter_recognize_clustered

Class colour
Reset
Clear
Open
Save
Jitter
0
9999.5
19999
X
Y
cluste cluste cluste cluste cluste cluste cluste cluste cluste
cluste cluste cluste cluste cluste cluste cluste cluste cluste

fig(4) fig(5) fig(6)

Conclusion: For clustering analysis, the number of clusters, seed locations and distance functions were changed to improve performance of the algorithm. In addition, for EM the performance was further improved using k-means to initialize the best starting location.

PART 2: Dimensionality Reduction Algorithm

The selection of relevant feature and eliminate the irrelevant one is a central problem. Machine learning algorithms are known to degrade (prediction accuracy) when faced with many features that are not necessary for predicting desired output. We need some techniques to recognize the unnecessary attributes. Removing these attributes increases efficiency and reduces over fitting. The feature subset selection techniques try to determine the appropriate features. Transformation by reduction of the set of variable into smaller set while keeping most of the information content. To pursue this there are four unsupervised technique for dimensionality reduction mentioned below:

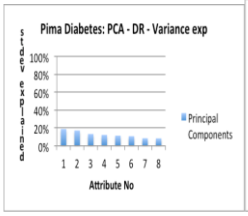
PCA Analysis:

The central idea of PCA is to reduce dimensionality of dataset consisting of large number of irrelevant variables, while retaining as much as possible of the variation present in the dataset. This is achieved by transforming to new set of variables, principal components (PCs), which are uncorrelated

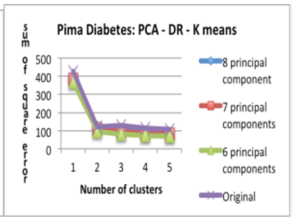
Dimensionality reduction is accomplished by choosing enough eigenvectors to account for some percentage of the variance in the original data---default 0.95 (95%). Attribute noise can be filtered by transforming to the PC space, eliminating some of the worst eigenvectors, and then transforming back to the original space. PCA identifies the pattern of correlation among dependent variable and substitute new variable (component) for the group of original attributes that were correlated. Due the fact this is unsupervised technique, it will not account the output feature, but only the input. Usually most of the variance is just explained by a handful of directions as opposed to the large original dimension space. We basically pick the top k principal components that explain a significant amount of variance in the data. Following charts plot the same along with the corresponding reconstruction errors.

PCA Analysis of Dataset-1(pima indian)

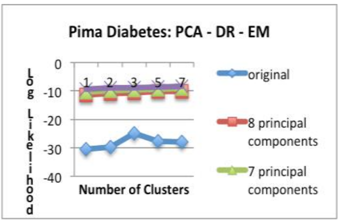
Fig(1)



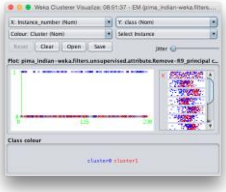
fig(2)



fig(3)



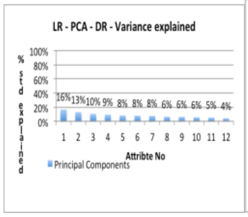
fig(4)


Weka Clusterer Visualize: 08:51:37 - EM (pima_indian-weka.filters...
X: Instance_number (Num)
Y: class (Nom)
Colour: Cluster (Nom)
Select Instance
Reset
Clear
Open
Save
Jitter
Plot: pima_indian-weka.filters.unsupervised.attribute.Remove-R9_principal c...
Class colour
cluster0 cluster1

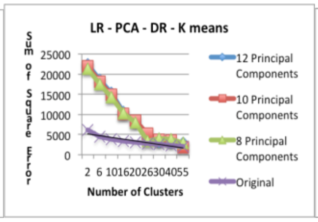
As it can be seen from fig(1), first 6 components is able to explain more than 95% of variance of data So an ideal strategy for dimensionality reduction would be to pick these first six principal components. Clearly, there's an elbow at k = 2 in the reconstruction K means and EM charts fig(2) and fig(3). so the strategy of picking the first 2 principal components is a valid one.

PCA Analysis of Dataset-2 (Letter Recognition):

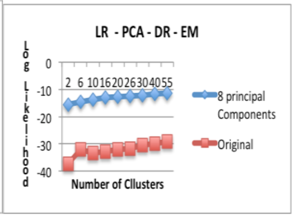
As it can be seen from fig(1), first 8 components is able to explain more than 95% of variance of data. So, first 8 components is picked for further cluster analysis to find K(number of clusters). From fig(2) and fig(3) , it is clear that graph clearly converges at K=26 compared to original dataset (without DR). Graph of 12,10 and 8 components overlapped with each other, which clearly states that there is very low deviation contribution by components 9-12.



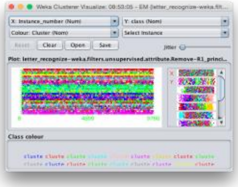
fig(1)



fig(2)



fig(3)


Weka Clusterer Visualize: 08:53:05 - EM (letter_recognize-weka.filt...
X: Instance_number (Num)
Y: class (Nom)
Colour: Cluster (Nom)
Select Instance
Reset
Clear
Open
Save
Jitter
Plot: letter_recognize-weka.filters.unsupervised.attribute.Remove-R1_princi...
Class colour

fig(4)

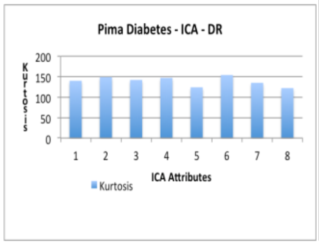
ICA Analysis:

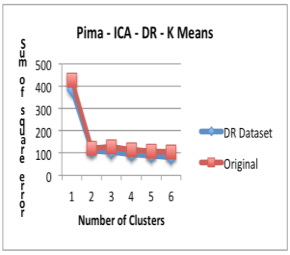
ICA on the other hand identifies features that are statistically independent from each other. The given features here are assumed to be a linear combination of some unknown latent features (the number of these latent features is the lower dimension here). ICA tries to describe the given data in terms of these latent features. The latent features are assumed to be non-Gaussian and mutually independent. In fact, the non-Gaussanity is the key to estimating independent components (from central limit theorem). The classical measure of non-Gaussanity is the kurtosis. Since kurtosis for a Gaussian random variable is zero, we need to pick the independent components that have higher kurtosis values in order to maximize non-Gaussanity. Since ICA is a local algorithm relying on random starting points, it was necessary to run the experiments several times to capture a trend. As there is no ordering in independent components, a value of k on the x-axis corresponds to top k components (by kurtosis) as compared to first k components in PCA. So the correct way to interpret the x axis values is to read it as the top kth independent component, rather than just the kth independent component. The below charts show the average kurtosis values for various independent components.

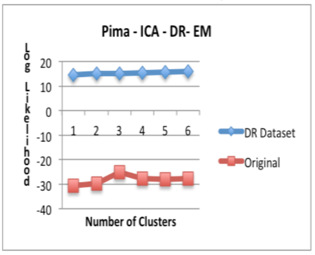
ICA Analysis of Dataset-1:

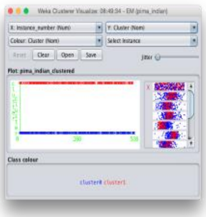
From the fig(1), Kurtosis for all the 8 components is comparable and hence we can't remove any components. From fig(2) and fig(3) It is clear that after selecting all 8 independent components , (number of cluaters) K =2. The EM chart demonstrates a significant improvement in LL, while SSE chart for KM overlaps or shows no improvement.

Fig(1) Fig(2) Fig(3)



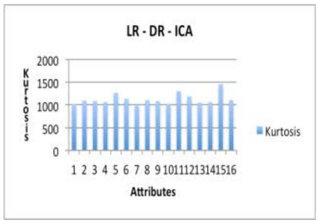


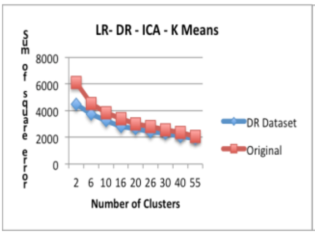


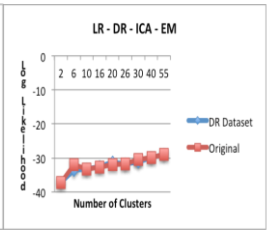


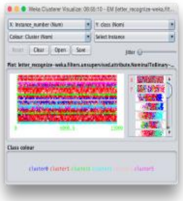
ICA Analysis of Dataset-2:

From the above two charts, it is clear that we should be selecting the top 3 independent components for Abalone and top 2 components for Wine. On using more components as part of the model seems to deteriorate the reconstruction error. This validates our argument saying that we should only be using non-Gaussian components in the model, i.e., components with high kurtosis values.








Weka Clusterer Visualize: 08:58:10 - EM (letter_recognize-weka.filt...
X: Instance_number (Num)
Y: class (Nom)
Colour: Cluster (Nom)
Select Instance
Reset
Clear
Open
Save
Jitter
Plot: letter_recognize-weka.filters.unsupervised.attribute.NominalToBinary-...
Class colour
cluster0
cluster1
cluster5

fig(1) Fig(2) Fig(3) fig(4)

After choosing n=6, From fig(1), It is clear that kurtosis for all ica components are comparable, so all 16 ica components are selected for cluster analysis. From fig(2) the k-means chart shows an improvement in SS, indicating that that the dimensionality reduction has helped reduce noise. From fig(3) ,The EM chart demonstrates a no improvement in LL and graph overlap each other. When comparing the original clustering for kmeans
k=26 , with the clustering for ICA n=6 (by elbow method).

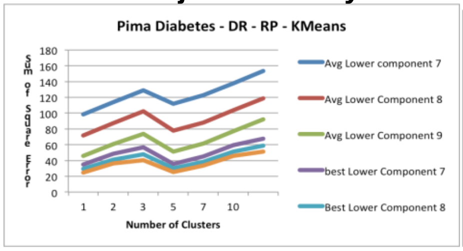
Random Projection Analysis:

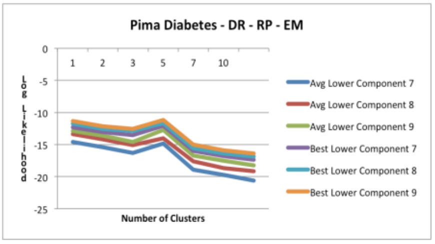
Random projection involves the projection of vectors lying in a higher dimensional space to a randomly chosen lower dimensional subspace. Note that this projection need not be Euclidean, i.e., the subspace need not be aligned with the basis vectors from the original space. A vector is projected by multiplying the vector by a suitable random matrix. First, construct a matrix R1 with each entry chosen independently from the distribution N(0, 1). Then, ortho-normalize the columns of R1 using Gram-Schmidt process (QR-decomposition) and form the required matrix R using these resultant vectors as columns. This special construction of the random projection matrix allows for the Johnson Lindenstrauss lemma to hold. The lower dimension in this case is chosen by looking at the reconstruction errors.
Calculating the reconstruction error for a randomly projected matrix is slightly a tricky thing and in fact an ongoing research topic. For the scope of this analysis, I limited myself to calculating it the simplest way, i.e. inverting the random projection matrix by

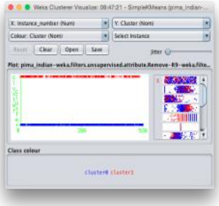
multiplying by its pseudo-inverse. The manner in which the random matrix was constructed in this case causes the pseudo-inverse to just be its transpose.

Average Lower component is chosen based on multiple runs=25 and averaging over values, while best component is chosen with lower reconstruction error among n=25 runs.

Random Projection Analysis of dataset-1:



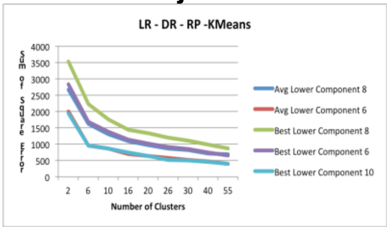


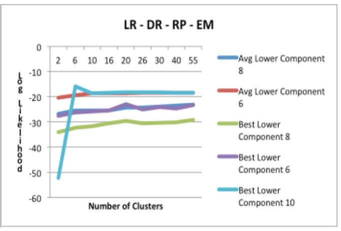

Weka Clusterer Visualize: 08:47:21 - SimpleKMeans (pima_indian-...
X: instance_number (Num)
Y: Cluster (Nom)
Colour: Cluster (Nom)
Select Instance
Reset
Clear
Open
Save
Jitter
Plot: pima_indian-weka.filters.unsupervised.attribute.Remove-R9-weka.filte...
Class colour
cluster0 cluster1

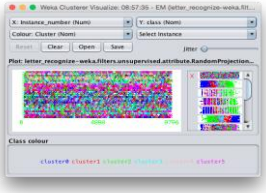
Since it's a randomized procedure, the data in RP are projected randomly into a new space of different direction. Cluster analysis is done for randomly choosing 10,9,8,7 attributes. As the chart shows, the different runs in RP have generally the same trend but vary within ~2% depending on the run and seed location.

Due to a lack of hard/elbow cuts, k=5 is chosen, because afterwards errors curve is almost constant. The EM chart shows much improvement with lower component=5.

Random Projection of Dataset-2:





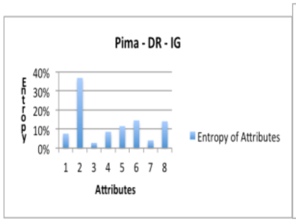

Weka Clusterer Visualize: 08:57:35 - EM (letter_recognize-weka.filt...
X: Instance_number (Num)
Y: class (Nom)
Colour: Cluster (Nom)
Select Instance
Reset
Clear
Open
Save
Jitter
Plot: letter_recognize-weka.filters.unsupervised.attribute.RandomProjection...
Class colour
cluster0 cluster1 cluster2 cluster3 cluster4 cluster5

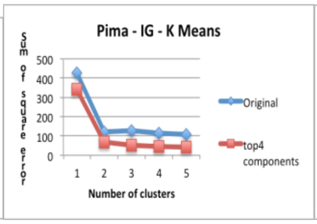
From both charts, it is clear the elbow point is k =6 . A thing to note here is that when k is equal to the original dimension, the error is almost zero, which makes sense. When the dimensions are same, the random projection matrix is basically just randomly rotating the data and thus, no information is lost.

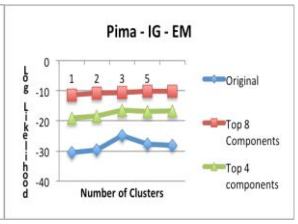
Information Gain Analysis:

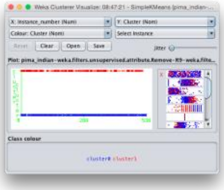
Another popular feature selection technique is to calculate the information gain. Weka supports feature selection via information gain using the InfoGainAttributeEval Attribute Evaluator. Like the correlation technique above, the Ranker Search Method must be used. You can calculate the information gain (also called entropy) for each attribute for the output variable. Entry values vary from 0 (no information) to 1 (maximum information). Those attributes that contribute more information will have a higher information gain value and can be selected, whereas those that do not add much information will have a lower score and can be removed.   This to ensure that most of the features end up showing up based on information Gain using ranking Algorithm. After the training, these features are ranked based on entropy. These values are representative of the relative importance of each of the features for classifying the data according to the given labels. We can then select the top *k* important features to reduce the dimensionality of our data. Following charts plot the feature importance according to the information gain for both the problems.

Info Gain Analysis on Dataset-1: Running this technique on our Pima Indians we can see that one attribute contributes more information than all of the others (plas). If we use an arbitrary cutoff of 0.05, then we would also select the plas, mass, age and insu attributes and drop the rest from our dataset.

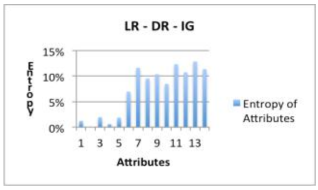


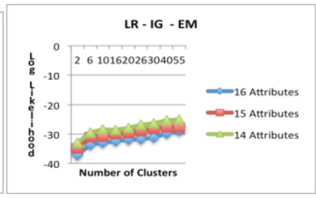


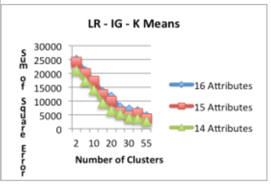


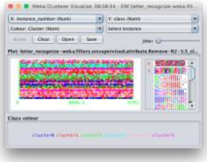

Weka Clusterer Visualize: 08:47:21 - SimpleKMeans (pima_indian-...
X: Instance_number (Num)
Y: Cluster (Nom)
Colour: Cluster (Nom)
Select Instance
Reset
Clear
Open
Save
Jitter
Plot: pima_indian-weka.filters.unsupervised.attribute.Remove-R9-weka.filte...
Class colour
cluster0 cluster1

Info Gain Analysis on Dataset-2:








Weka Clusterer Visualize: 08:58:54 - EM (letter_recognize-weka.fil...
X: Instance_number (Num)
Y: class (Nom)
Colour: Cluster (Nom)
Select Instance
Reset
Clear
Open
Save
Jitter
Plot: letter_recognize-weka.filters.unsupervised.attribute.Remove-R2-3,5_cl...
Class colour

Clearly there are a few features which seem to be a lot more important than the rest. This skewness will help us pick the relevant features and reduce the dimension of the problem. Notice that Feature 3 and 7 for dataset 1 problem and feature 2 and 4 of dataset-2 shows almost zero value in the chart. This might be because the feature does not contain enough information to show up in any of the rules. Using the above charts, we select top 4 features for Pima and top 14 features for LR. Interestingly, unlike the other dimensionality reduction algorithms where the new features were generated by transforming the original features, the new features in this case are simply a subset of the original features.

Conclusion:

From cluster analysis, it is clear that PCA Analysis gives best clusters for dataset-2 (Letter Recognition) and info gain analysis for dataset-1 Pima Diabetes.

Part 3:

Learning Using Neural Network:

This is where it all culminates. This is where we look at how the classification performance is affected when clustering and/or dimensionality reduction algorithms are applied to the dataset at the pre-processing stage. A neural network from assignment-1 is used as the base classifier to compare performances.

Applying dimensionality reduction algorithms on Letter Recognition dataset, which is used in assignment#1. I am adding labels on dimensionally reduced dataset using unix command. Use these dimensionally reduced dataset (without added label) in clustering algorithm and it will add clusters as new labels. Use these datasets in Neural Network Classification:

Neural Network Analysis for Dataset-1 (Pima Diabetes)

<u>Base line From Assignment -1</u>

Training Accuracy : 76.35%
Testing Accuracy: 79.22%

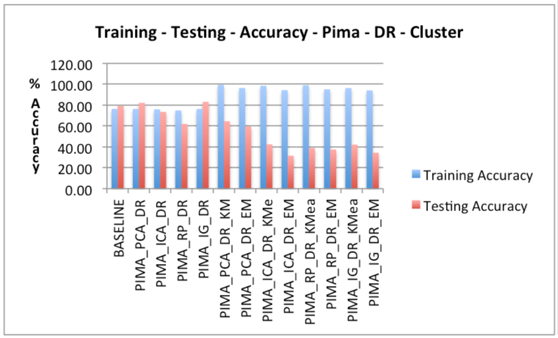
Sr.No	Datasets	Learning Rate	Momentum	Training Accuracy	Testing Accuracy	Query time	Train Time
Dimensionally Reduced Datasets							
1	PIMA_PCA_DR	.3	.3	76.3%	82.19%	.02	23.15

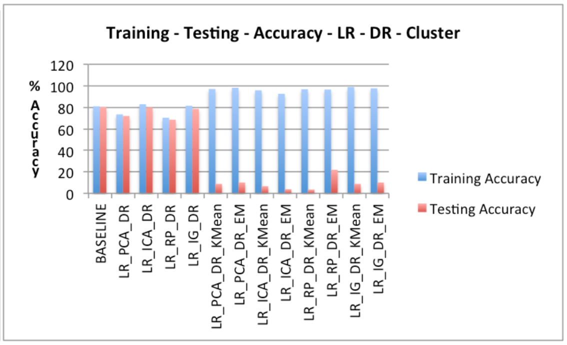
2	PIMA_ICA_DR	.3	.3	75.97%	73.51%	.03	40.12
3	PIMA_RP_DR	.3	.3	74.87%	61.9%	.01	16.79
4	PIMA_IG_DR	.3	.3	76.3%	83.16%	.03	23.45
Applying clustering on DR datasets							
5	PIMA_PCA_DR_KM	.3	.3	99.44%	64.5%	.04	40.19
6	PIMA_PCA_DR_EM	.3	.3	96.45%	59.56%	.05	45.67
7	PIMA_ICA_DR_KMean	.3	.3	98.51%	42.42%	.03	65.12
8	PIMA_ICA_DR_EM	.3	.3	94.35%	31.56%	.04	76.12
9	PIMA_RP_DR_KMean	.3	.3	99.06%	38.9%	.04	37.28
10	PIMA_RP_DR_EM	.3	.3	95.15%	37.45%	.05	43.12
11	PIMA_IG_DR_KMean	.3	.3	96.34%	42.13%	.03	49.12
12	PIMA_IG_DR_EM	.3	.3	94.14%	34.35%	.05	55.14

Neural Network Analysis for Dataset-2 (Letter Recognition Dataset)
Base line from Assignment -1: Training Accuracy = 80.94%, Testing Accuracy = 80.21%

Sr.No	Datasets	Learning Rate	Momentum	Training Accuracy	Testing Accuracy	Query time	Time Train
Dimensionally Reduced Datasets							
1	LR_PCA_DR	.2	.8	73.44%	71.97%	.31	71.11
2	LR_ICA_DR	.2	.8	82.86%	79.83%	.26	143.29
3	LR_RP_DR	.2	.8	70.34%	68.47%	.21	78.44
4	LR_IG_DR	.2	.8	81.44	78.48%	.15	84.23
Applying clustering on DR datasets							
5	LR_PCA_DR_KMean	.2	.8	97.03%	8.89%	.23	169.19
6	LR_PCA_DR_EM	.2	.8	98.12%	10.16%	.16	175.23
7	LR_ICA_DR_KMean	.2	.8	95.8%	6.7%	.17	138.67
8	LR_ICA_DR_EM	.2	.8	92.55%	3.8%	.21	142.9
9	LR_RP_DR_KMean	.2	.8	96.8%	3.52%	.11	79.69
10	LR_RP_DR_EM	.2	.8	96.53%	21.91%	.13	79.61
11	LR_IG_DR_KMean	.2	.8	98.8%	8.89%	.31	204.68
12	LR_IG_DR_EM	.2	.8	97.68%	10.13%	.45	213.16

Neural network Time to train decreases with less number of clusters.





Conclusion:

From the Above charts, it is clear Dimensionally reduced dataset matches the baseline. So clearly, using the cluster features only is a bad idea performance wise. Testing loss increase. Training accuracy is still ok, but testing accuracy is pretty bad. This explains due to over fitting of clustered DR datasets. The test error deteriorates, implying overfitting. So we’re creating clusters out of the training data and then using the same clusters as features to train another model on the same data. Perhaps we’re trusting our training data a bit too much, which is what’s causing the model to overfit.

References:

Assignment-1 and Udacity Lecture
ICA Plugin for WEKA GUI https://github.com/cgearhart/students-filters/raw/master/StudentFilters.zip