CS7641 Machine Learning HW3

Winnie Yeung
GTID: 903390457

March 18, 2019

Introduction

This is an analysis of using clustering and dimensionality reduction techniques to gain domain knowledge and solve two classification problems. Two clustering techniques are used: k-means, Expectation-Maximization(Gaussian Mixtures). Four dimensionality reduction algorithms are used: Principal Component Analysis, Independent Component Analysis, Randomized Projection, Feature Selection using Chi-square test. The analysis is divided into 5 parts: clustering, DR, clustering + DR, using DR outputs as features in Neural Network (ANN), and using DR outputs with clustering labels as features in ANN. The first three parts are discussed with the wine quality dataset and forest cover type and the last two parts with only the wine quality dataset.All the Clustering and dimensionality reducation are implemented in Python 3.5 using the scikit-learn and ANN is run through Abigail with Jython.

Datasets

The first dataset is white wine quality. It has 11 attributes related to physicochemical properties of the wine. It is a binary classification problem between good and bad wine with about 4900 observations. **F1-Score** is used as metrics to put both false positives (recall) and false negatives (precision) into account.

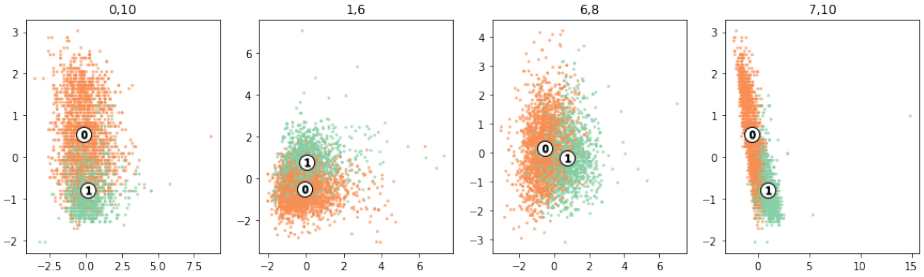
The forest cover dataset contains tree observations from national forest in Colorado. The dataset has 15120 observations and 53 geographical attributes wiht some of them booleans. There are seven types of covers. **Accuracy scores** is used as the measure of accurate prediction of forest cover types.

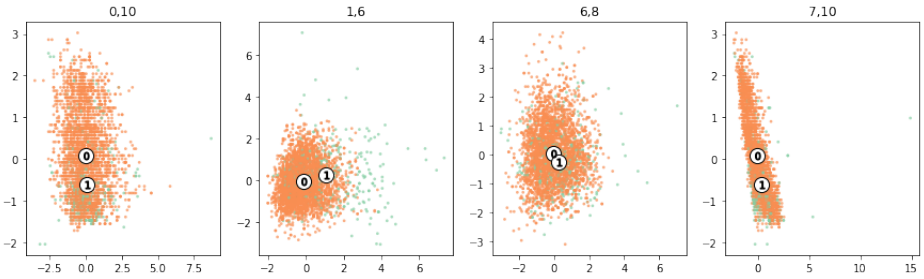
Clustering

I used **k-means** and **Expectation Maximization** algorithms to conduct clustering for the two datasets. Kmeans is hard assignment while Gaussian mixtures is a soft assignment for each observation to be a mixture of n components. **No. of clusters:** I used k=2 for wine quality datasets and k=7 for forest cover types. This naturally corresponds to the labels of the data whereby wine quality is divided into good wine and bad wine and there are 7 forest cover types. **Scoring:** Since we know the output labels, we can compare the clustering with the actual output and use **mutual information score**to measure how useful are the clustering in providing insights into the actual grouping of the data.

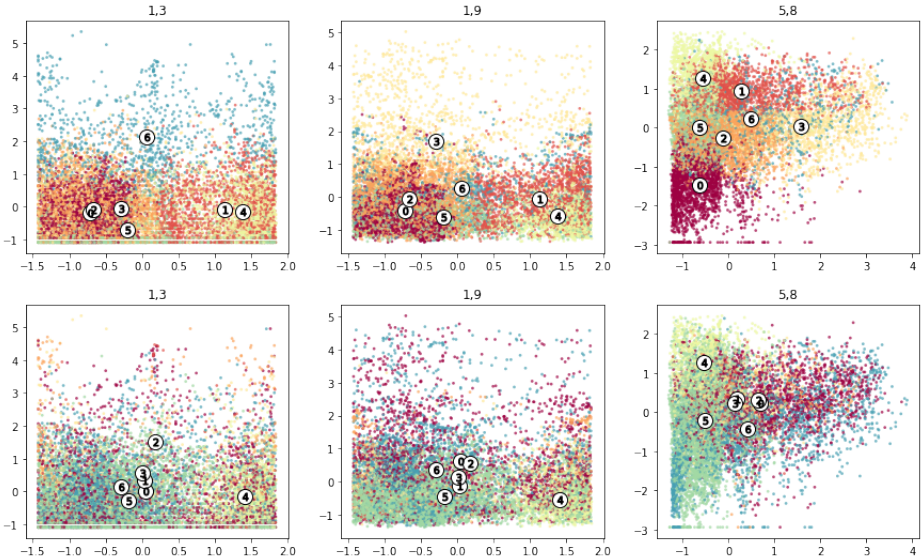
Mutual Information Score		
Dataset	k-means	EM
Wine	0.057	0.036
Forest Cover	0.229	0.337

Wine Dataset: (Top Panel: Kmeans, Bottom Panel: EM)

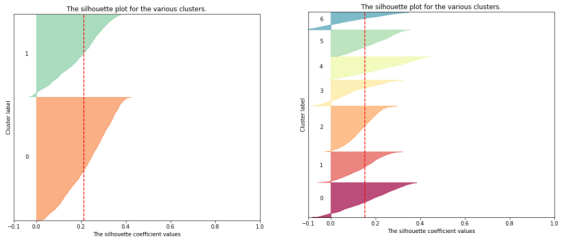




Forest Type Dataset Feature Plots with respect to clustering: (Top 3 Panels: Kmeans, Bottom 3 Panels: EM)



Silouette Scores Plot for Kmeans (Right: Wine , Left: Forest Cover)



Analysis:

Wine Dataset: From the Mutual Information Score, we can see that for wine dataset, kmeans cluster assignment offers more useful information related to the labels than EM. The observation is supported by the feature plots. While we can see the clusters separates more clearly between the orange and green group, with the two centroids far from each other. The Gaussian Mixture returns centroids that are closer to each other looking across different features. Another thing is that the weight of the Gaussian Mixture assigns 89% weight to cluster 0 and 11% to cluster 1, implying that most points follow closer to one cluster. But from our domain knowledge, we know that the two groups are about 60-40 split so Gaussian Mixture offers weaker insights into the real groupings of the data.

Forest Cover: I was surprised to see that the mutual information score is 6 times better than the wine dataset. I think it is because the feature offers rich information to the groupings as compared to the wine dataset. It is also surprising to see that EM outperforms k-Means in the score, which indicates the grouping generated by Gaussian Mixture is better than in k-Means. This can be the case since **33 out of 53 features** of Forest Cover types are booleans features which does not have a Gaussian distribution of data. K-means assumes the clusters as spherical, so it does not work as well with non-linear data. With 7 clusters, it is different to visualize the clusters bu we can see that the clusterings separates some groups more distinctively than the other groups.

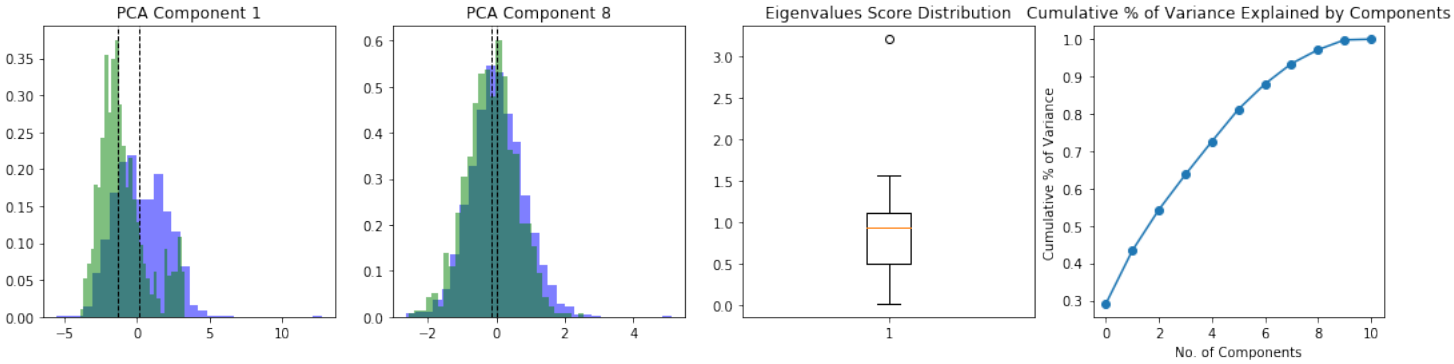
Justification of choice of number of clusters: I used Silouette scores to verify if picking clusters of 2 for wine dataset and clusters of 7 for forest cover dataset is a good choice. Each observation will receive a silouette score. If we group the scores by cluster, and if the max silouette score is below the overage overage silouette score, it indicates bad assignment of clusters. Running the silouette score plots indicates maximum score for each cluster is well above the average silouette score, meaning that the cluster is well justified.

Dimensionality Reduction

I used Principal Component Analysis (PCA), Independent Component Analysis (ICA), Randomized Projection (RCA) and Feature Selection by Chi-Square Score (FS) as dimensionality reduction techniques.

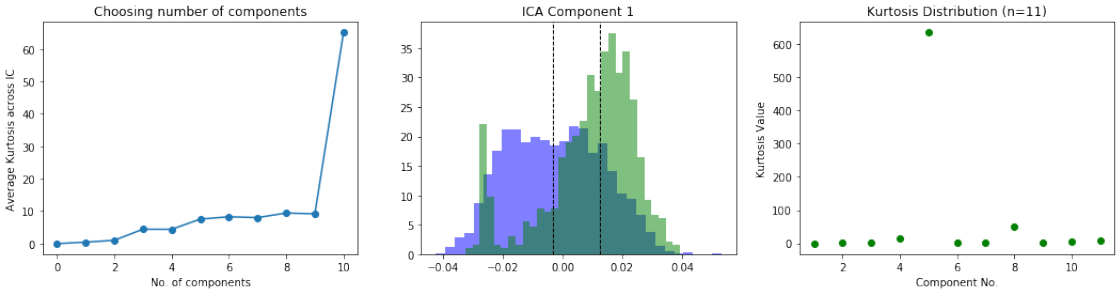
Wine Dataset:

Principal Component Analysis:

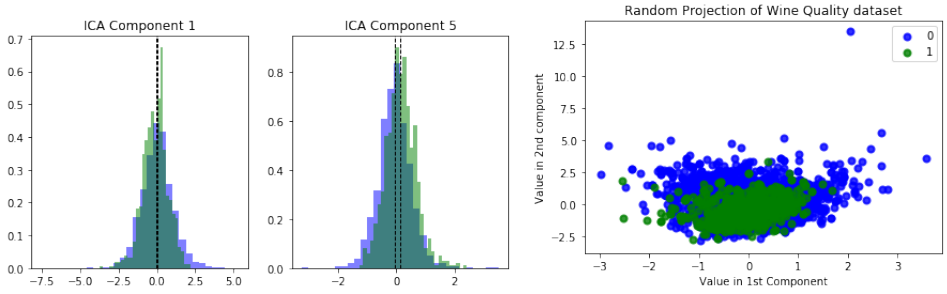


Left 1,2: Components Density Plot**Middle:** Cumulative Explained % of Variance **Right:** Eigenvalues Distribution
PCA Analysis: I used ratio of variance explained to understand how much each component explains variation in the data. First component has high eigenvalues thus after normalizing the eigenvalues, we see that the first component explains about 40% of variance in data. While the others' eigenvalues ranges between 0-1.5, explaining 10% of variations. To represent 95% of the variations, I need at least 9 dimensions to represent my 11-dimensional data. Plotting the density distribution by groups for the first component, we can see that the means are different, showing that component 1 is a good diferentiator for the two groups. The later components like no. 8 is weaker in differentiation, since densities for both groups are approximately normal with similar mean.

Independent Component Analysis:



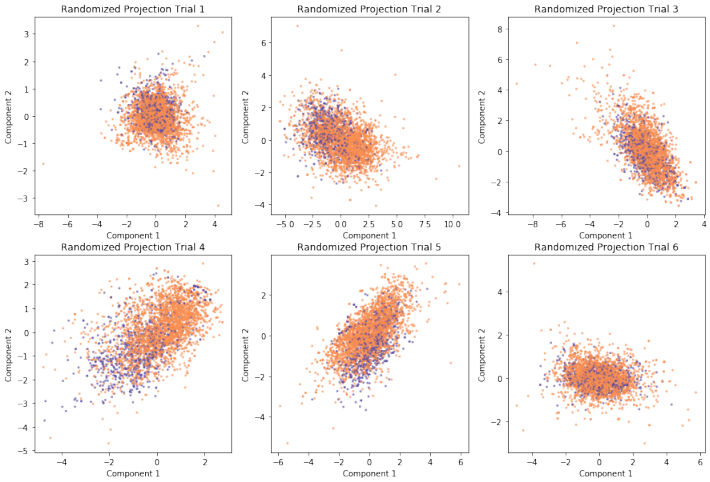
Left: Avg. Kurtosis Across Components, **Middle:** ICA Component Density, **Right:** Kurtosis Distribution
ICA - Analysis: To choose the optimal number of components, I calculated the median kurtosis across all components after projection. Median is used to avoid being affected extreme values at tails. It returned 10 components as the optimal number. To examine the effectiveness of the projection, I look at the distrbution of values for each classified groups. We can see that after transformation, two groups actually significant difference in distribution. In terms of kurtosis, we can see kurtosis in 10-40, most extreme get to 600.
Randomized Projection Left,Middle: Distribution of two groups for Component 1 and 5 Right: Scatter Plot of 2 features from RCA



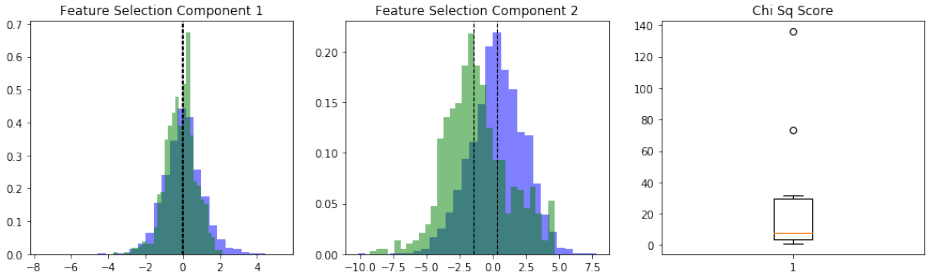
Analysis: I choose to project the 11-dimensional data into 5-dimensional space.It is really noisy as the distribution of

different group all centered around mean showing little differentiation.

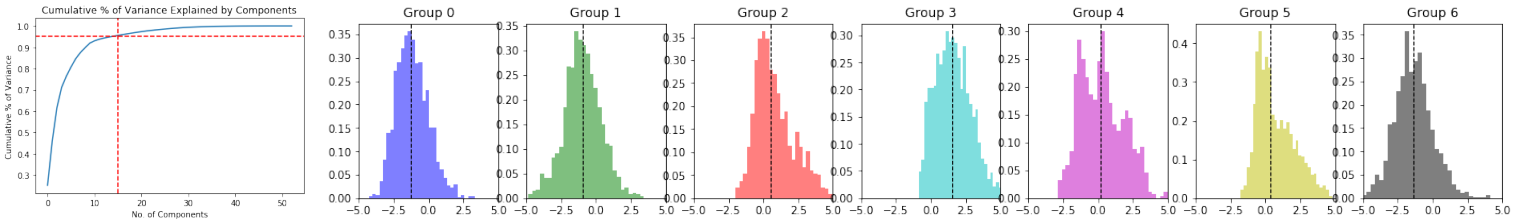
Different random states: I run the projection 6 different times, from the plotting we can see that the projection is different. Similar to randomized hill climbing, the projection can stumble upon an effectively dimensionality reduction or being random, adding more noise rather than differentiation. This is the plotting of the result of 6 different trials:



Feature Selection: I used Chi-square scores to compare attributes against output labels to select the best 4 attributes. Chi-square measure the similarity between the two series, the higher the score, the better one series in representating the other. Looking at box plot, two features are much stronger than the others who range between 10-40. Some differentiation between groups can be seen in density of best two attributes where means of component two are significantly different.



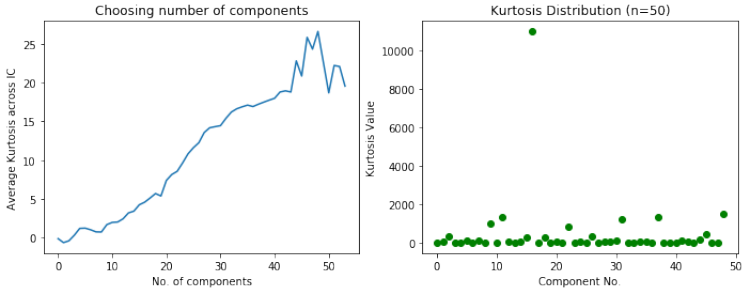
Forest Cover Type: Left: Cumulative Explained Variance Right 1-7: Distribution in first component



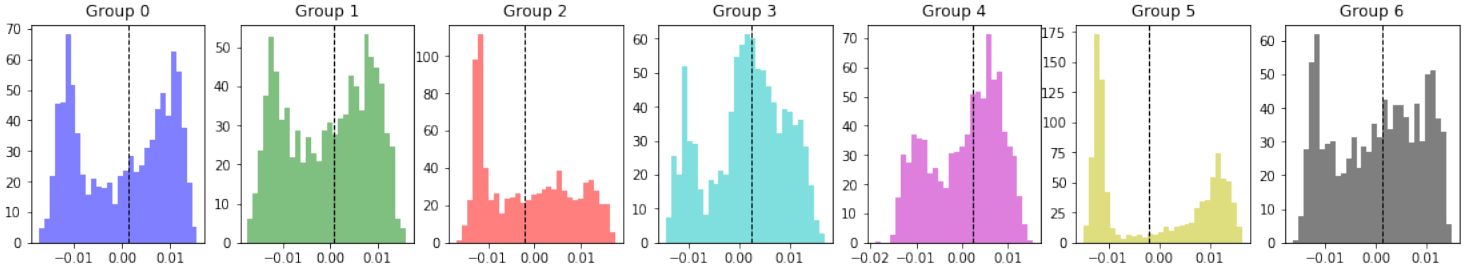
PCA Analysis: With 53-dimensional data, using PCA, we can see that 15 components can explain about 95% of the variations of the original dataset. Looking at the first component, we can see each forest cover type exhibits different distribution, some groups are more right-skewed and some other left-skewed, showing that the component is helpful in making the differentiation.

Independent Component Analysis:

Left: Average Kurtosis, Right: Kurtosis Distribution



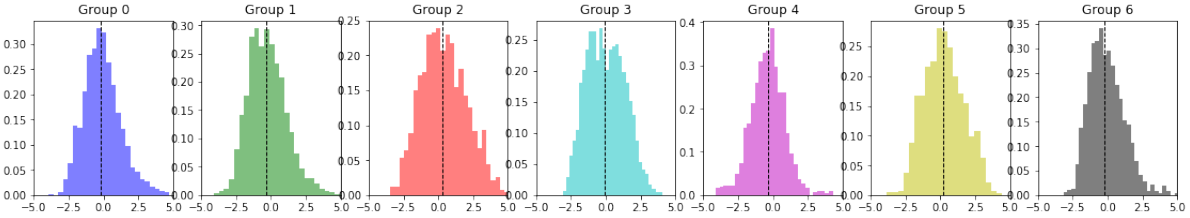
Looking at the median kurtosis, 49 components for 53-dimensional dataset returns me the highest average kurtosis. We can see that one component peaked the kurtosis while the others stay in with a value of 25.



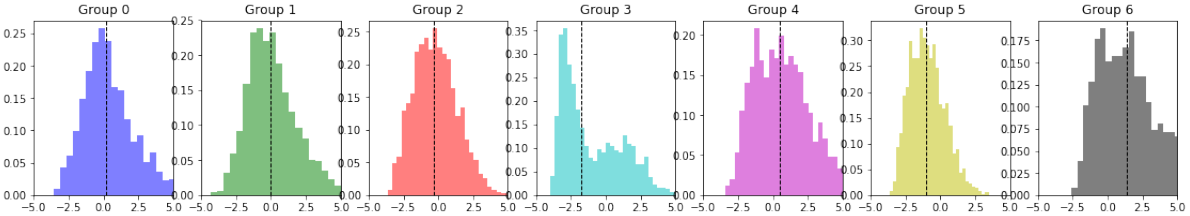
Looking at distribution of 7 groups for component 25, we can see that distribution of groups looks highly non-gaussian, with some more left-skewed and some more right-skewed and differences in means across groups.

Randomized Projection:

Component 1 Distribution for trial no.1

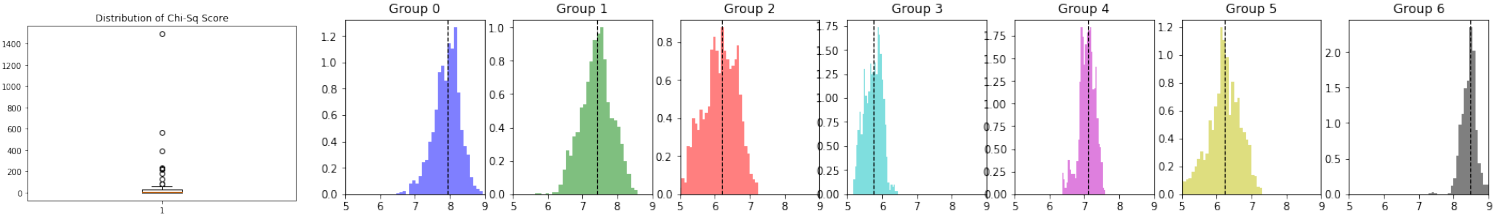


Component 1 Distribution for trial no.2



Analysis: I chose to project the 53-dimensional data into 5 dimensional space to see the effects. I compared the distribution for trial 1 and 2. Sometimes random projection lakcs differntiation power like in trial 1. However, for trial 2, the projection matrix successfuly separates group 3 and group 6, having significantly difference of mean and non-gaussian distribution.

Feature Selection:



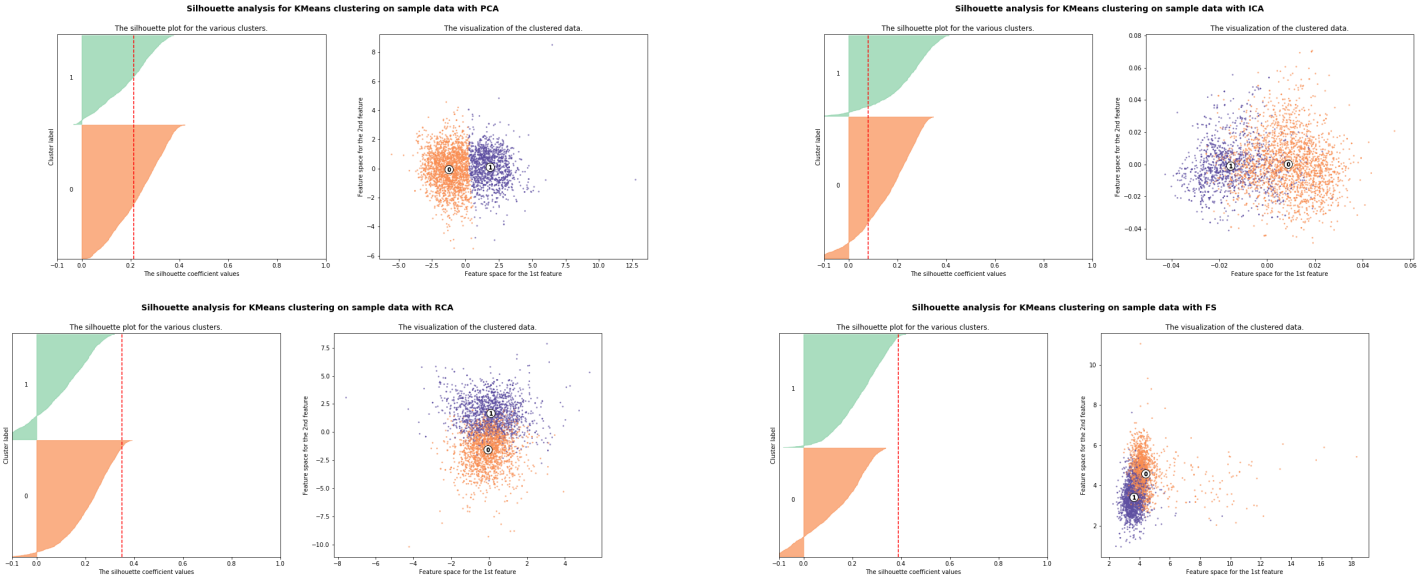
Analysis: Similar to procedure for wine dataset, I chose best 5 features with highest chi-sq square to represent the data. Looking at the boxplot, there are obvious outliers in chi-sq square to pick the most informative attributes. First attribute actually has the highest chi-sq square. Looking at the distribution, we can clearly see that between value of 5-9, different group concentrates on certain spectrum of values and are non-overlapping, demonstrating this feature is significant in separating the groups apart similar to what we would do in decision tree.

Clustering After Dimensionality Reduction

Mutual Information Score for Wine Dataset		
Dataset	k-means	EM
Baseline (n=11)	0.057	0.036
PCA (n=11)	0.057	**0.054**
ICA (n=11)	0.023	0.036
RCA (n=5)	0.057	0.028
4-Best Feature (n=4)	**0.088**	0.021

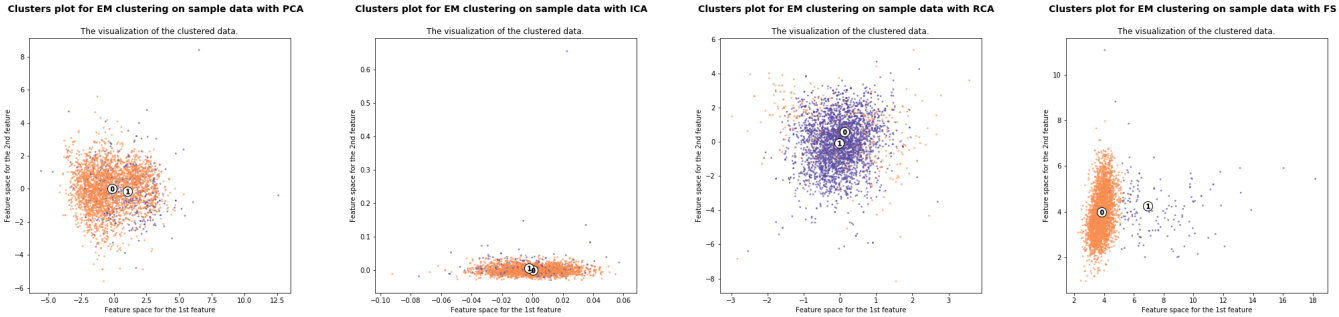
KMeans:

Left-Top: PCA, Right-Top: ICA, Left-Bottom: RCA, Left-Right: Feature Selection



Gaussian Mixture EM:

Left to Right: PCA,ICA, RCA, and Feature Selection

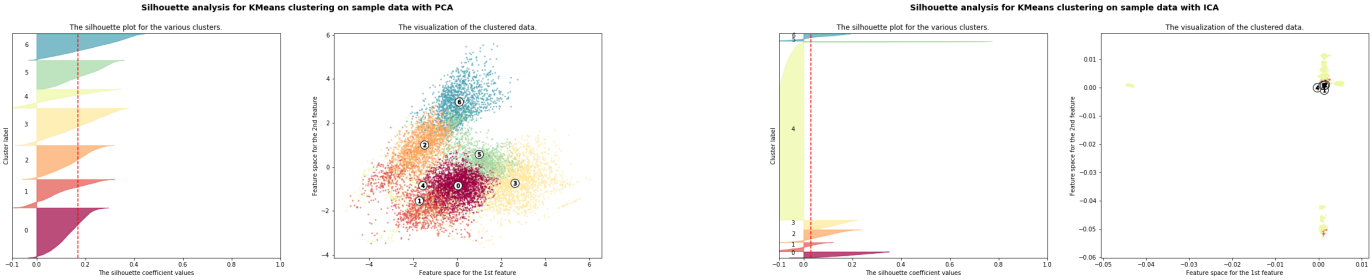


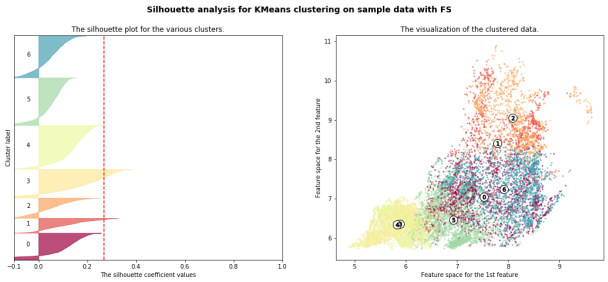
Mutual Information Score for Forest Cover Dataset		
Dataset	k-means	EM
Baseline(n=53)	0.229	**0.337**
PCA (n=15)	0.227	0.248
ICA (n=48)	0.112	0.206
RCA (n=5)	0.065	0.078
Feature Selection (n=4)	**0.336**	0.329

Forest Cover Dataset

KMeans:

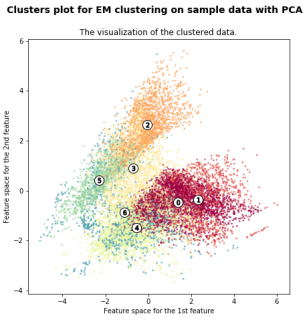
Left-Top: PCA, Right-Top: ICA, Left-Bottom: RCA, Left-Right: Feature Selection

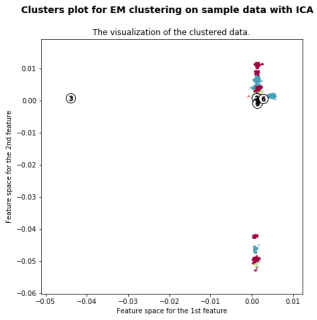


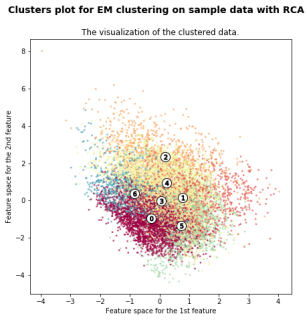


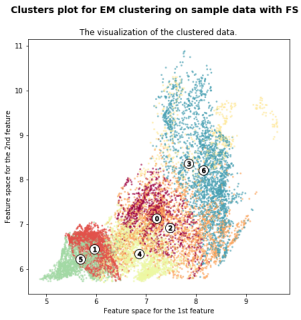
Gaussian Mixture EM:

Left to Right: PCA,ICA, RCA, and Feature Selection









Analysis:

Looking across both Wine and Forest Cover Type result, some DR algo has improved the mutual information score and some decreases. **The one that decreases across the board is Randomized Projection (RCA)**, which is understable, by some random projection matrix, the distribution of groups is added with some noise and make the groups more difficult to differentiable.

Visualization: Comparing between the scatter plot between part 1 clustering and clustering in this part, I notice that examining the feature it looks like the clusters have become more distinct and less overlapping in kmeans. I would think it is due to dimensionality reduction to separate out some of the noise in the original data.

PCA: Looking at k-means visualization after PCA, the projection separates the variation along two axes more clearly, which means visualization of groups becomes clearer too, even with 7 groups! However, for mutual information score for wine set, since I picked 11 components, which is the same as original dimension, same amount variations remain in the projected data, thus the mutual information did not change much. For Forest Cover type, since I picked n=15, which captures 95% of the variation in the data, the kMeans algorithm receives less information from the PCA-reduced attributes hence the Mutual Info score decreases.

ICA: ICA is about breaking down independent components that span the feature space. Instead of linear combinations of all attributes, it is to have subset of attributes in one components and some in the others. Interstingly,in all scenarios ICA decreases the mutual information score for both k-means and EM across the two dataset. I think one reason is that ICA components are too complex for k-Means or EM to understand well, especially given very non-Gaussian irregular distribution. To draw a good separation line, one might need something like ANN instead of EM algo (if labels exist).

Feature Selection: k-means can handle simple but not complex attributes is further support by the FS result whereby, FS with only 4 attributes in wine dataset and in forest cover type **improves** the mutual information score. I think this is the simplicity of kmeans algorithm. These features have outsized correlation with the output labels when they were selected, they capture much of the information of the dataset, while they are all equally important, kmeans are about to generate meaningful clusters that improve the information score. On the otherhand, EM is not improved much because the soft clustering already assigns weights to clusters that taking features importance into account. Having fewer features actually bring in less information than the baseline.

How to improve the performance of the algorithm?

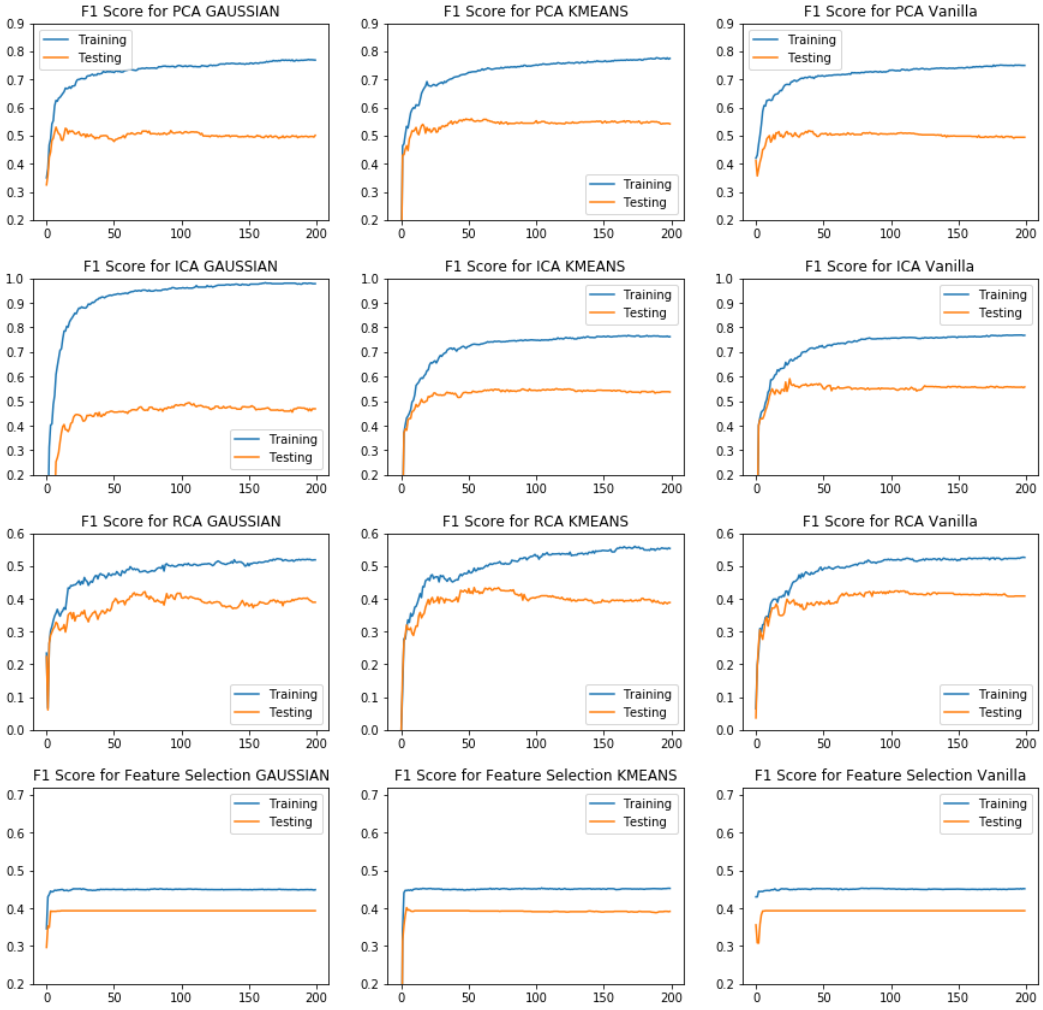
Clustering after PCA: I suggest **adding weights relative to eigenvalues of components** since 40% of variation is in the first component and less so in the others. Clustering after ICA: I suggest **increasing number of iteration** as the data might be complex to understand. **Weighing attributes by kurtosis** might also help.

ANN After DR and Clutsering

From Homework 2, I showed that putting in the entire wine dataset in ANN returns a test score of 0.451. This will be used as the baseline for comparing results with using ANN for dimensionality-reduced data and adding clustering labels as an additional feature. The table shows the relative test score compared to the baseline (=0.451). I have kept the hidden layer size and number of iteration the same and only modifying the input layer size to hold consistency in experiment setting.

Relative F1 Score for ANN from Baseline (Baseline score = 0.451)			
Method	Vanilla	k-means	EM
PCA (n=11)	0.04	0.09	0.05
ICA (n=11)	**0.11** (BEST)	0.09	0.02
RCA (n=5)	-0.04	-0.06	-0.06
Feature Selection (n=4)	-0.06	-0.06	-0.06

Learning curves for each trial is displayed below:



It is motivating for me to see that using dimensionality reduction algorithm would boost the ANN performance by 20% from 0.451 to 0.55. PCA and ICA has improved the performance across the board with or without clustering labels. However, RCA and Feature Selection led to decreasing performance, likely because dimensionality reduction from 11 to 5 or 4 has reduced the variation in the data to make group classification more difficult to tackle than the original data. For RCA, adding k-Means or EM actually worsens the performance, I think it is because the clustering based on RCA results actually created more noise than a good feature as we see that the mutual information score of the RCA cluster label is really low.

Looking into PCA and ICA further: I think what I have done in PCA and ICA using all the components is that instead of reducing the variation, the DR-algorithms have kept all the variations but recombine the attributes as such it is easier to differentiate from one another. When putting it into a simple ANN model with only one hidden layer, the DR is doing the legwork to make the perceptron distingish the groups more easily using the components (almost like dumbing it down), like how we visualize on the plots earlier. For PCA, since it is a linear combination of all attributes, each

attribute might be still more complex to read, putting a clustering attribute from k-means or EM helps tell additional information about the data. Since the cluster labels has good information score, it acts as a meaningful attribute to improve the model performance.

As for ICA, for components to be independent subsets of attributes is super powerful and makes the perceptron a lot easier to differentiate hence the best boost in performance. However, recalling from before that the k-means or the EM cluster labels have weak information score, hence adding the labels as additional features decreased model performance as it is adding more noise.

Leanring Curves: For PCA and ICA, the training performance is much better than the other two methods. I think this represents how much easier the data is interpretable by the perceptron to understand the distribution of the data. However, it was learning more noise as well, hence we see the bigger gap between training and testing curves in PCA and ICA compared to RCA and FS.

Time Performance: running PCA and ICA models takes the same time as in HW2 (20 seconds). Since RCA and FS has only 4-5 attributes, it took only 10 seconds to run these two implementations.