CS7641 ML- Project 3 report

Chenxi Yu

Saturday, Apr 1, 2017

Performance measure:

Evaluating the performance of algorithms is crucial at this assignments. Thanks to sklearn in python (2.3. Clustering), I will start up by briefly discussing the approaces I used as follows:

Homogeneity score:

It measures each cluster contains only members of a single class. A clustering result satisfies homogeneity if all of its clusters contain only data points which are members of a single class. This metric is independent of the absolute values of the labels: a permutation of the class or cluster label values won't change the score value in any way.

Completeness score:

It measures all members of a given class are assigned to the same cluster. A clustering result satisfies completeness if all the data points that are members of a given class are elements of the same cluster. This metric is independent of the absolute values of the labels: a permutation of the class or cluster label values won't change the score value in any way.

Random score:

It measures Rand index adjusted for chance. The Rand Index computes a similarity measure between two clusterings by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the predicted and true clusterings.

The raw RI score is then "adjusted for chance" into the ARI score using the following scheme:

ARI = (RI - Expected_RI) / (max(RI) - Expected_RI)

The adjusted Rand index is thus ensured to have a value close to 0.0 for random labeling independently of the number of clusters and samples and exactly 1.0 when the clusterings are identical (up to a permutation).

Silhouette scores:

Compute the mean Silhouette Coefficient of all samples.

The Silhouette Coefficient is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample. The Silhouette Coefficient for a sample is (b - a) / max(a, b). To clarify, b is the distance between a sample and the nearest cluster that the sample is not a part of. Note that Silhouette Coefficent is only defined if number of labels is 2 <= n_labels <= n_samples - 1.

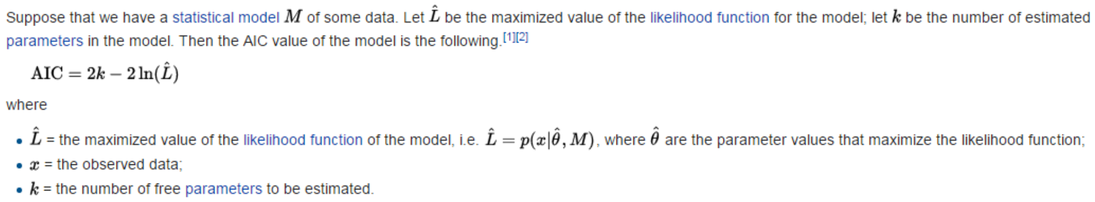
For the Silhouette scores to be a good value for numbers of cluster, one should consider the following points

1. Firstly, The mean value should be as close to 1 as possible
2. The plot of each cluster should be above the mean value as much as possible. Any plot region below the mean value is not desirable.
3. The width of the plot should be as uniform as possible.

BIC and AIC:

Both criteria are based on various assumptions and asymptotic approximations. Despite various subtle theoretical differences, their only difference in practice is the size of the penalty as illustrated in the formulas shown as below. As the dimension population increases, the numeric value in BIC gets worse faster than AIC. The best model in the group compared is the one that minimizes these scores, in both cases. Clearly, AIC does not depend directly on sample size.

Moreover, generally speaking, AIC presents the danger that it might overfit, whereas BIC presents the danger that it might underfit, simply in virtue of how they penalize free parameters (2*k in AIC; ln(N)*k in BIC). Diachronically, as data is introduced and the scores are recalculated, at relatively low N (7 and less) BIC is more tolerant of free parameters than AIC, but less tolerant at higher N (as the natural log of N overcomes 2). BIC penalizes model complexity more heavily. The only way they should disagree is when AIC chooses a larger model than BIC.



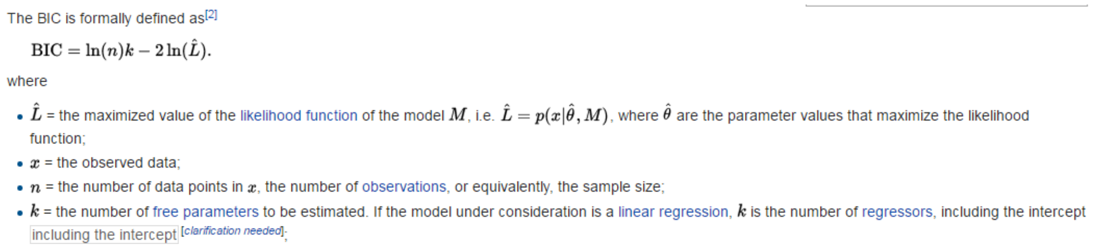
Suppose that we have a statistical model M of some data. Let $\hat{L}$ be the maximized value of the likelihood function for the model; let k be the number of estimated parameters in the model. Then the AIC value of the model is the following.[1][2]

$$\mathrm{AIC} = 2k - 2\ln(\hat{L})$$

where

- $\hat{L}$ = the maximized value of the likelihood function of the model, i.e. $\hat{L} = p(x|\hat{\theta}, M)$, where $\hat{\theta}$ are the parameter values that maximize the likelihood function;
- x = the observed data;
- k = the number of free parameters to be estimated.

(From https://en.wikipedia.org/wiki/Akaike_information_criterion)



The BIC is formally defined as[2]

$$\mathrm{BIC} = \ln(n)k - 2\ln(\hat{L}).$$

where

- $\hat{L}$ = the maximized value of the likelihood function of the model M, i.e. $\hat{L} = p(x|\hat{\theta}, M)$, where $\hat{\theta}$ are the parameter values that maximize the likelihood function;
- x = the observed data;
- n = the number of data points in x, the number of observations, or equivalently, the sample size;
- k = the number of free parameters to be estimated. If the model under consideration is a linear regression, k is the number of regressors, including the intercept including the intercept [clarification needed];

(From https://en.wikipedia.org/wiki/Bayesian_information_criterion)

Reconstruction errors:

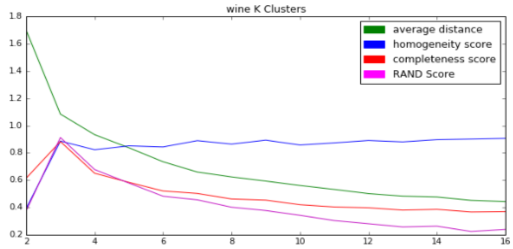
I use it as a way to find the best number of components because PCA finds projection that minimizes reconstruction error while I need to find some small reconstruction errors for RP in order to get appropriate results for RP.

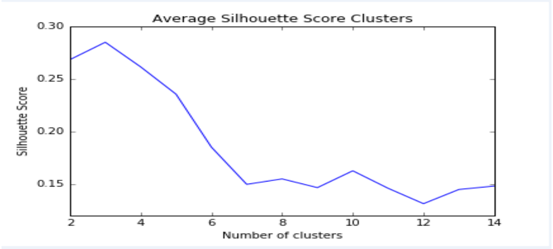
Part 1

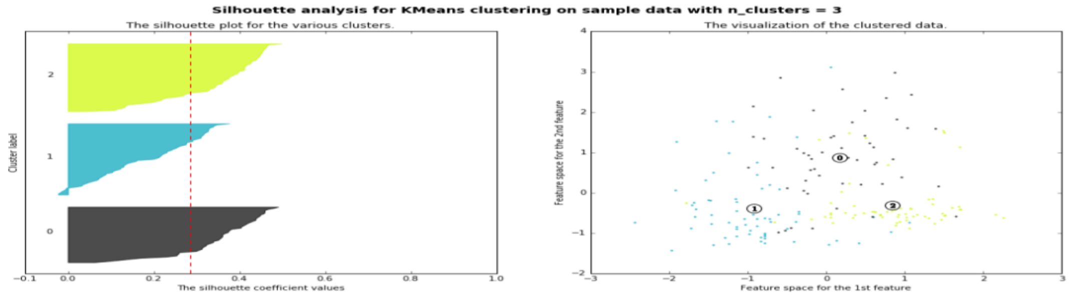
The wine data set by C.Blake from UCI Machine Learning repository is chosen as the additional data set for the clustering and dimensional reduction techniques exploration. There are 12 features and 1 categorical outcome with 179 samples. I picked this data set for simplicity reason to explore the 4 dimensional reduction and clustering method.

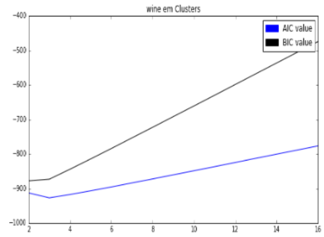
1. Clustering analysis:

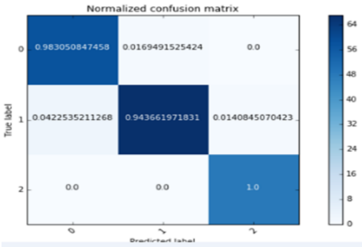
Exploring the K means on the wine data set, both the elbow method and silhouette analysis indicates k = 3 is the

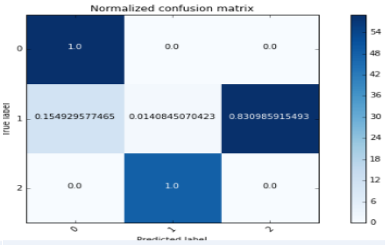








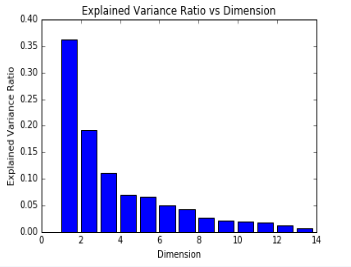


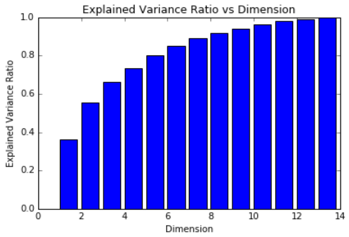


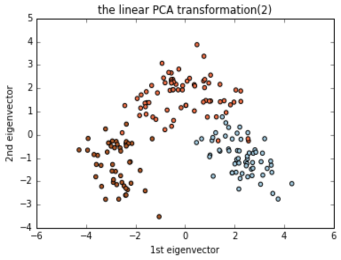
best cluster among all. The confusion matrix (in the stage_1 folder) tells the labels line up pretty well with the clusters result. Thus, k=3 is preferred but the confusion matrix tells the labels does not line up well with the clusters result shown in the right bottom corner.

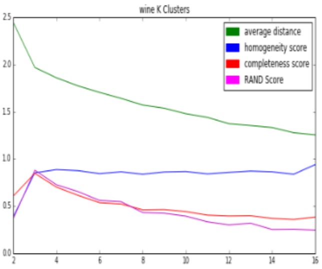
2. Dimensional Reductions and Clustering

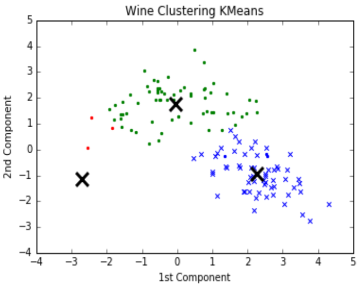
A. PCA

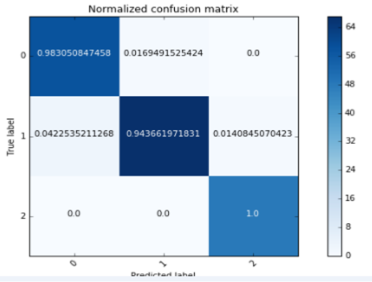




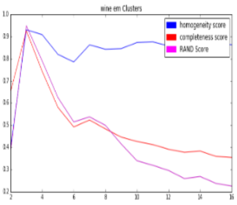


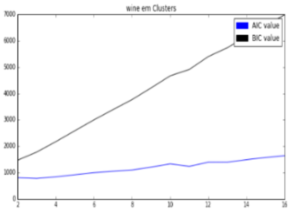


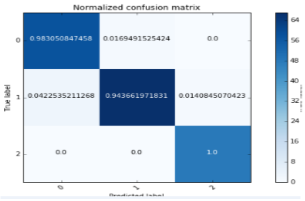


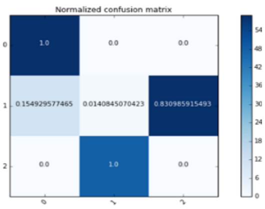


The upper left 2 bar plots shows that the explained variance was more than 80% explained by the first 5 components, while the linear PCA components indicates a fair separation. The reconstruction error for the first 9 components is 0.198. Next, looking at the distance measures used in the above lower graph, it indicates a stable homogeneity score after k=3 and the scatter plots and confusion matrix identifies the result and it does improve from the original clusters.





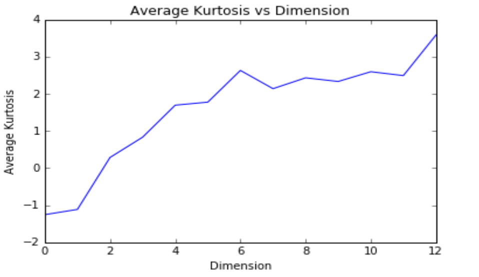


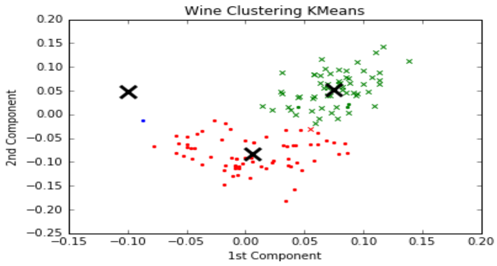


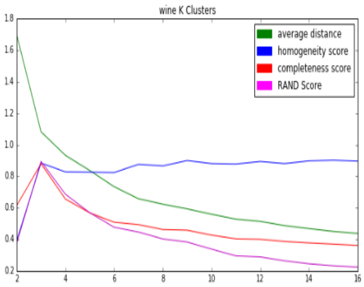
The above 2 graphs indicates some disagreements for the EM clusters after applying PCA. The BIC and AIC seems increase overtime and diverge away. Thus, I believe this PCA model does not yield sound results for K means clustering and it is agreed by the result shown in the confusion matrix.

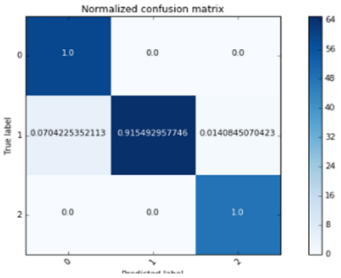
B. ICA

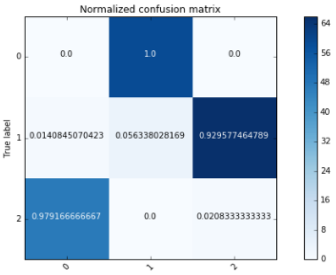
The below graph indicates ICA's kurtosis curve is widely falling in the range of -1 and 3 which means that the data set in the dimensions area tend to be independent from each other. The class labels in the 2nd scatter plot shows a different result than the PCA's. Next, the K means clustering algorithm is proceeded and noticed that the average distance score was decreasing along with completeness and Rand, while the homogeneity score remained high over the increasing clusters in the middle of the graph below. This is similar to the case in PCA's, yet ICA's the average distance measure's converges faster. However, the BIC and AIC curves still look abnormal like PCA's. However, the Kmean's clustering is well lined up with the labels whereas the EM's has not performed well again.

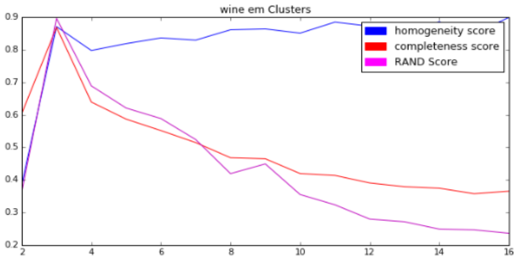


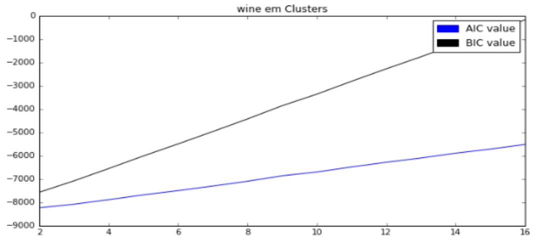






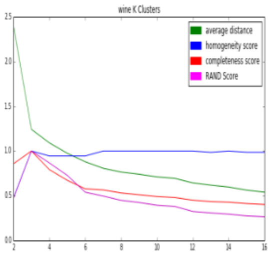


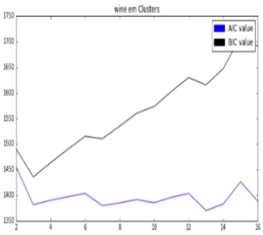


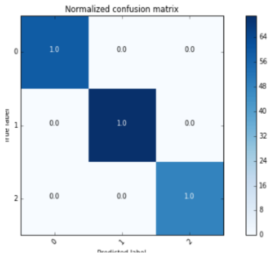


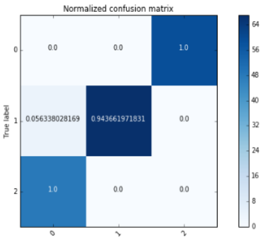
3. LDA:

So far, I am short of a way to measure the LDA, probably the best way is to measure it in a supervised learning context which I will do in the next part in the report. As shown below, both clustering algorithms perform well after LDA, even stunning, the K-mean just have the perfect labels. plus the dimensions number remained is only 2.



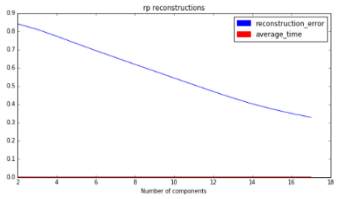


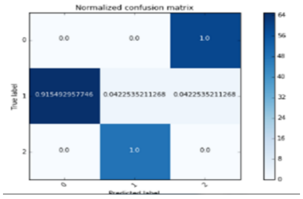


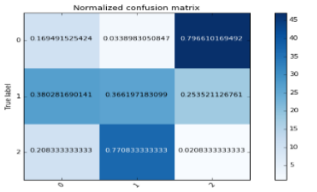


The K-mean clustering results after LDA is shown in the left above and it looks similar to the other 2, whereas the EM clustering looks different from the others as it finally indicates k=3 is the best clustering it gets after which it tend to overfit as BIC surges fast.

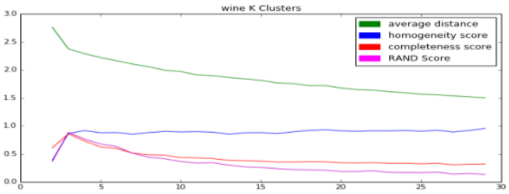
4. Random Projection:

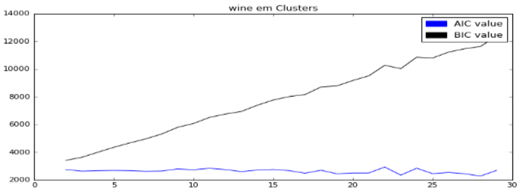






The random projection’s performance could be measured by the average reconstruction errors. I have tried some random run on it and decided n_component is optimal with 11 and 0.086 as the reconstruction error. In the flipped side, added in the searching time, RP does take more time than any other methods. Below are the clustering algorithms run correspondingly. Yet, both K-means and EM have not labeled well after RP’s process.





Now, the K-means clustering graph looks similar to the PCA’s yet it may perform worse than PCA as it was based on random run yet it keep more components than the PCA which is capturing more explained variance. Meanwhile, the EM clustering fails to make sense again because it means the whole clustering model chain keeps overfitting with more clusters.

In terms of the run time performance for the dimensional reduction algorithms, RP is the fastest per run (0.0039s), traced by PCA(0.025s) and ICA(0.033s), finally LDA(.037s) . It makes sense intuitively as RP is based on randomness, PCA on explained variance, ICA on capturing non-Gaussian signals and LDA on linear separators. Overall, in the above labelling tasks, PCA, LDA and ICA have helped transformed the data, this is especially true in K-means clusters.

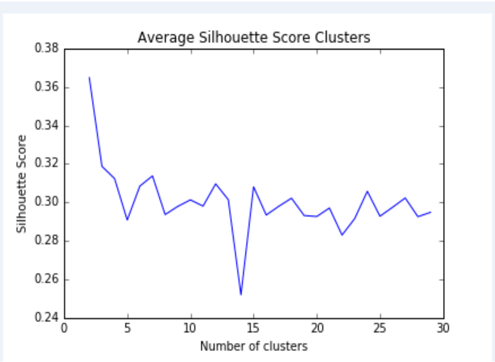
Part 2

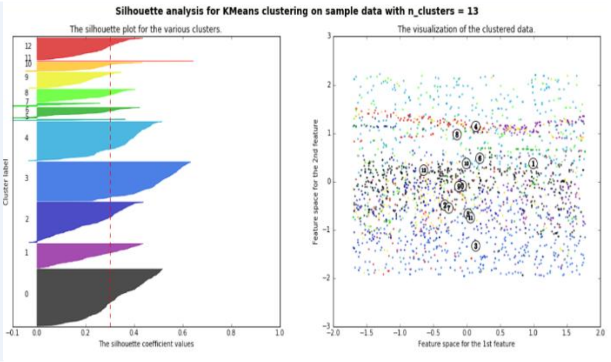
The image Segmentation classification for every pixel from UCI Machine Learning repository is used again for this portion of the analysis. There are 18 features and 1 categorical outcome with 2310 samples in the data sets.

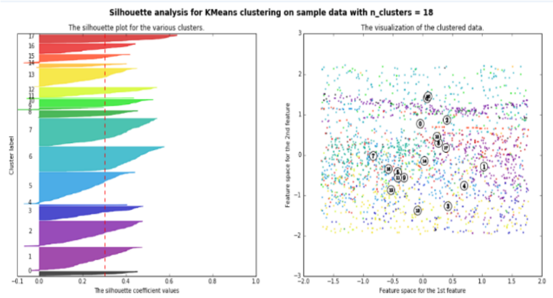
1. Clustering analysis:

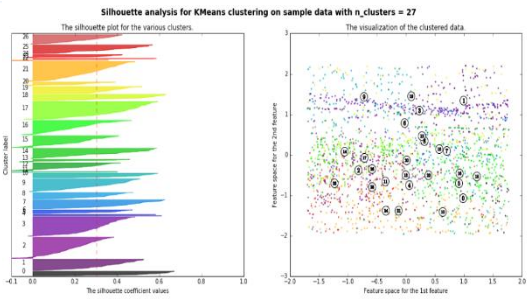
On the one hand, I applied K-mean clustering to the data set and in the silhouette scores plot and their detail analytic graphs shown as below, we find that the clustering number equal to 18 or 27 or 13 are the best according to the previous criteria stated when seeking the optimal clusters model.

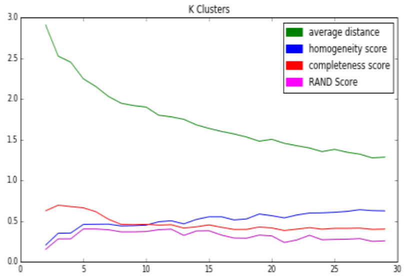
There are other peaks in the below graphs when the curve purges. Yet, those clusters were not chosen because the width of the coefficient distribution is not as uniform as others.



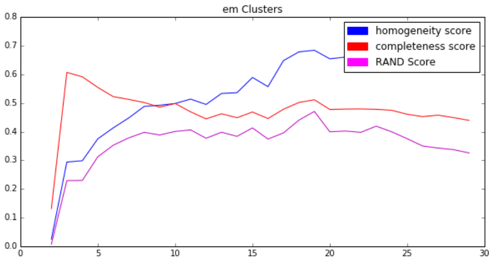


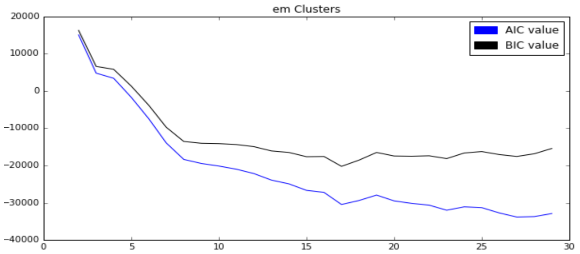






On the other hand, EM clustering is applied to the data set and in the silhouette scores plot and their detail analytic graphs shown as below, we find that the clustering number equal to 17 is the best because both the elbow method and the information criteria (AIC and BIC) indicates peaks and valley in the graph, despite no obvious convergence trend is shown. Yet, it is apparent that homogeneity score, rand score and completeness score decreases after clusters size runs down to 17 while the model tends overfitting as BIC gradually increases after that threshold.



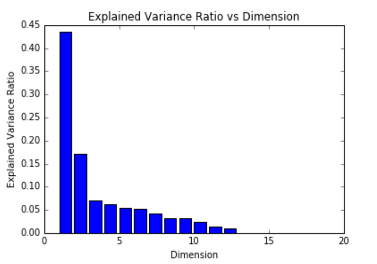


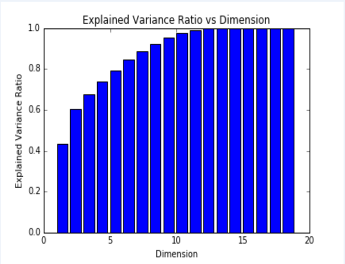
Finally, when it comes to run time, on average, it takes almost 10 times as long for running EM clustering as the K means clustering. Specifically, EM takes 0.90 for 28 runs while K-means for 0.09. This is due to their algorithmic difference. EM based on the movement of the centers and clusters without the assistance of gradient method or others dooms to be slower than the distance approach applied by K-means.

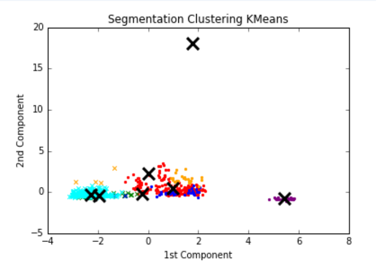
Overall, I would pick k = 13 on the k-mean clustering for further exploration to the neural net model construction later discussed in the report as it looks better in all categories for our silhouette analysis while agreeing to our elbow method indicated by the 4 measures in the last figure above. Meanwhile, I choose n = 17 on the EM clustering for neural net model construction.

2. Dimensional Reductions and Clustering

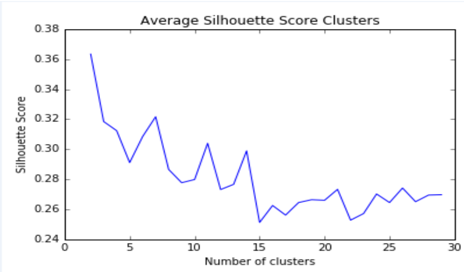
A. PCA:

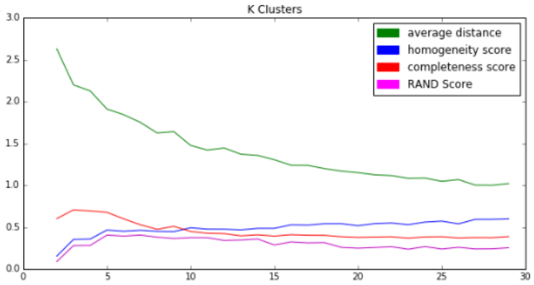


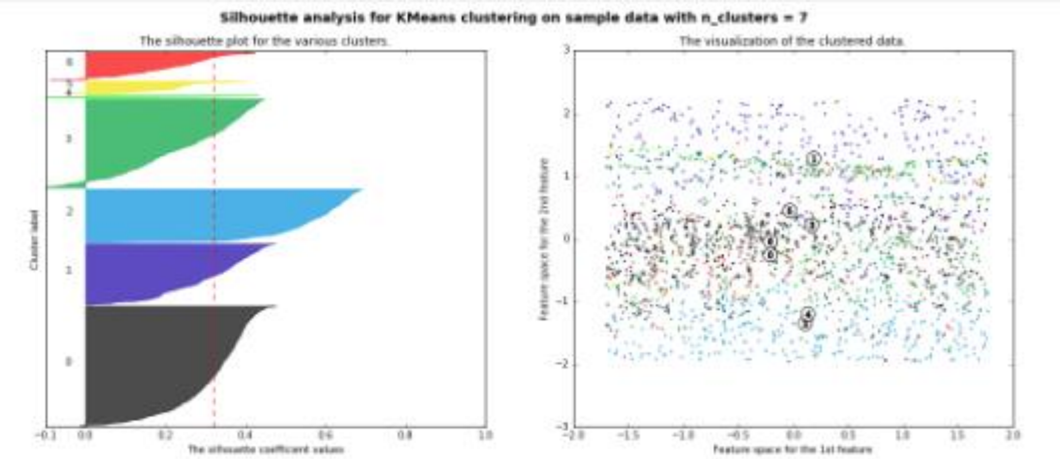




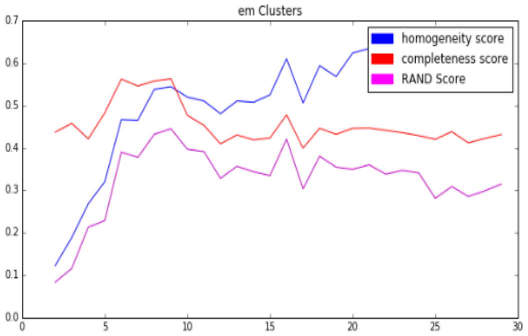
As is shown above, almost 91% of variance if explained by the first 7 components while 66.5% was by the first 3 components. The reconstruction error is 0.11, which is not bad overall. It is obvious that only 12 components are explaining the variance. However, if we refer to the principal components plots, it seems do pretty well except for the upper center. So I proceeded to have the number of components as 7 and ran the 2 clustering algorithms.

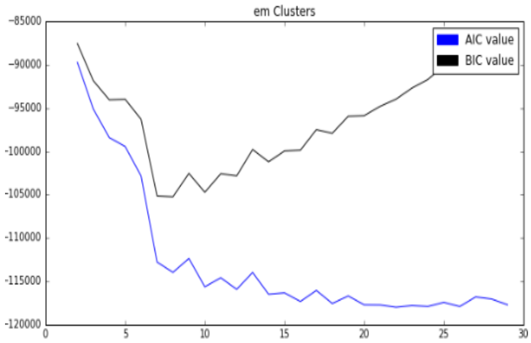






After applying the k means clustering on the data transformed by PCA, I choose the k=7, because the average silhouette score seems indicating a peak while its distribution looks promising, though it does not look obvious in the elbow method curves shows in the upper right graph. Yet, the clustering plots on the right do not look obvious and there are some discrete variables distorting the results.

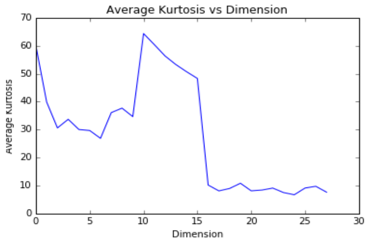


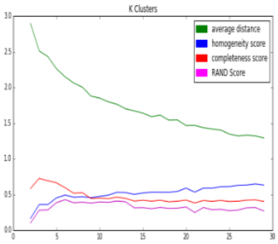


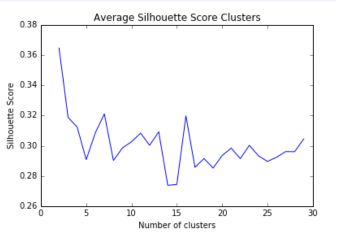
Next, the results from the EM clustering shows that k =8 is the best as both methodologies agree. Although it seems k=16 a good position, the divergence of AIC and BIC curves fail horribly, thus indicating a heavy overfitting for the model construction. Interestingly, the IC curves fast divergence after n = 15 indicated PCA's variance reduction merit.

B. ICA:

The mean Kurtosis for the ICA running is around 28.5 and its range is from 7 to 66 implying that the data set is not filled widely by random noise and it tells the number of components is optimal by 11 which means this 11 components is more statistically independent than the other models. And when that score falls badly by 16 components.

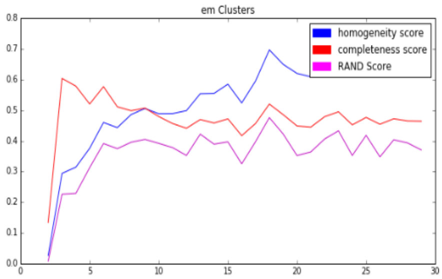


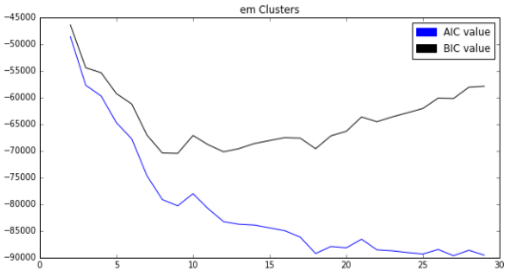




However, it may be better to have negentropy score to see if outliers may heavily distort the results as kurtosis is not sensitive to capture outliers.

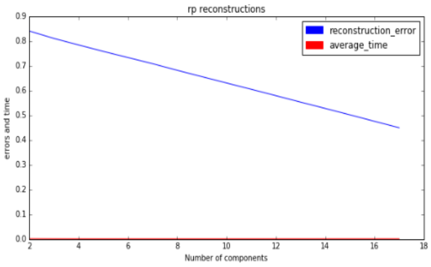
Next, in terms of the K-means clustering applied to the data set transformed by ICA, via the elbow method, it seems that k=13 is the best which is agreed by the average silhouette graph. Meanwhile, when it comes to the EM clustering, the below graphs show that the information criteria tells k = 8 is the best though may not be the case if measured in the distance approach shown in the left graph below. However, we stick with this result if the outliers distorting effect is trivial in this case. But as per the EM clustering information criteria graph, this has not violated my intuition though negentropy method is on the demand for further analysis.

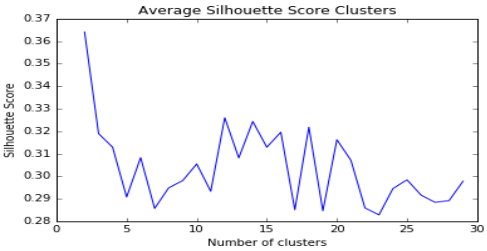




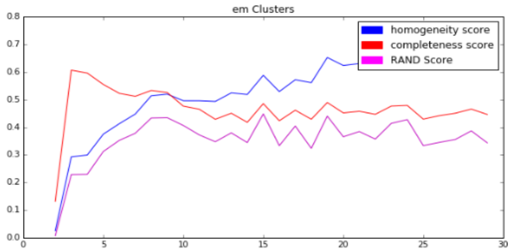
C. Random Projection:

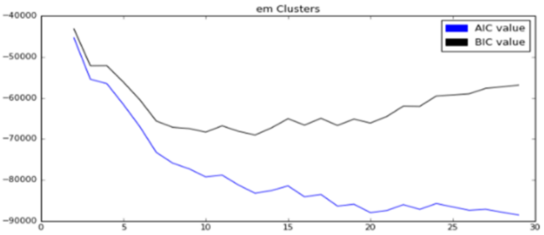
This one has not been easy to set up for the evaluation as its randomized nature has not allowed variance measure without difference comparison to make sense. However, I have managed to measure it with reconstruction errors curve based on the approach having 100 runs per component unit as it expands. As it is shown in the lower left, the reconstruction errors decreases linearly as the components number increases, however, it really does not converge fast due to its random nature. Probably it look better after the running rounds increases for per component unit. However, after several tries, I found k=17 stands a high chance for a low reconstruction error--0.081.Yet, the time difference per run is pretty short and varies little, which is the advantage of this algorithm. If computational timing and complexity could not spare room for this model in running, it exhibits a lack of stability. However, if those is ignored, it could reach global optima and thus perform better than many other algorithms.





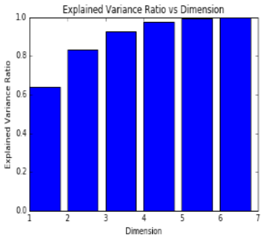
However, taking in the consideration from the k-mean clustering, the silhouette score curve indicates worse stability than the other dimensional reduction techniques result as indicated in the upper right graph. Meanwhile, the IC curves looks overfitting after 11 in the lower right but k=10 or 11 are preferred which is not agreed with the elbow approach. However, a random approach does not necessarily include superior result than PCA. However, it may be improved with more runs.

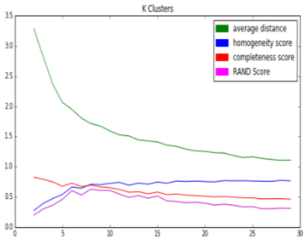


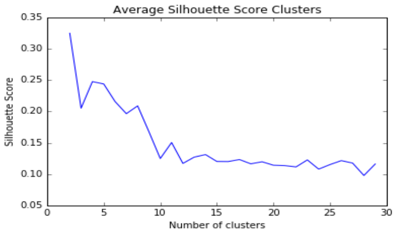


D. Linear Discriminate Analysis

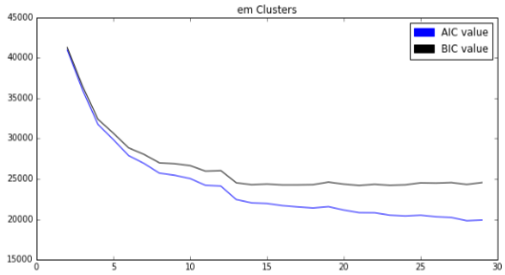
As indicated in the literature, LDA does not take variance into account for dimensional reduction and the one in the left below tells a lot. Upon so far, I have not found a way to effectively measure its performance.

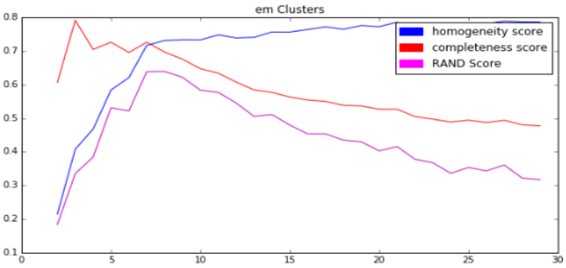


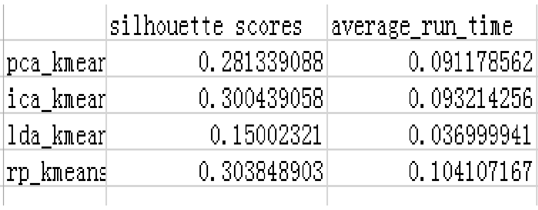




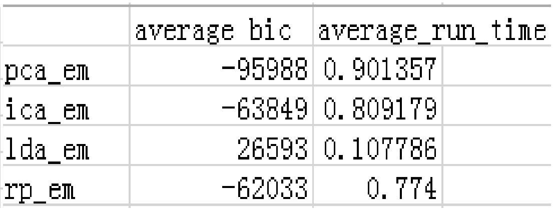
Considering the performance of the k-means clustering after LDA, I found both the elbow approach and the Silhouette approach agree that k = 8 is the best it gets though the silhouette score prefer k= 4 if only the score amount is concerned. Meanwhile, below is the EM clustering after LDA and it does not look like the other clustering graph's BIC patterns who tend to over-fit as the clustering number increase and it has high values for cluster number =6.







	silhouette scores	average_run_time
pca_kmear	0.281339088	0.091178562
ica_kmear	0.300439058	0.093214256
lda_kmear	0.15002321	0.036999941
rp_kmeans	0.303848903	0.104107167

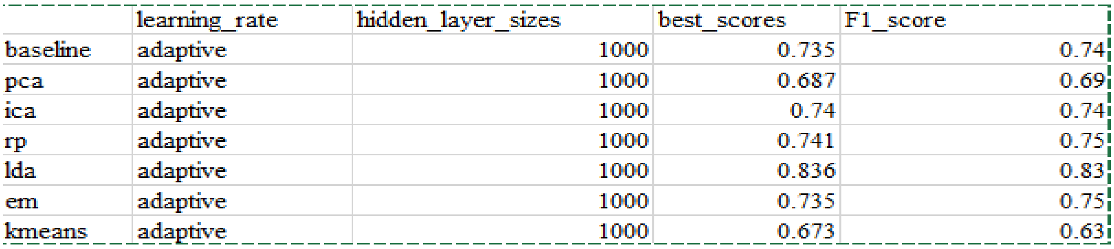


	average bic	average_run_time
pca_em	-95988	0.901357
ica_em	-63849	0.809179
lda_em	26593	0.107786
rp_em	-62033	0.774

Overall, there are some average performance scores for each algorithm, it seems that K-means after ICA is the best in silhouette score, while EM after PCA is the best in average BIC. When it comes to run times, LDA is the best, however, it seems not do well on the dimensional reduction task in our data set as both high average BIC and low silhouette score indicates that. Yet, our data set is not the case as ICA has shown that our data set's kurtosis component-wised is not closed to 3 in the first 30 components. Yet, comparing the run time for the above approaches, we notice that EM clustering does poorly while K-means is genuinely fast in each scenario.

E. Neural Network

After applying all 4 dimensional reduction techniques and 2 clustering techniques, we have 6 datasets for Neural network testing. Of all the dataset, I find it weird to have per 2 columns filled with the same values for the clustered data sets. However, for the sake of performance evaluation on the “dimensional reductions tasks”, I keep them intact. To save time, I applied the previously simple parameter tuning where only tuned “hidden_layer_sizes” and “learning rates” in grid search thus having hidden_layer_sizes =1000 and learning rates=adaptive as the baseline for examining the performance of Neural network after data transformations. Below is the result after running script in Gridsearch approach. Yet, coincidentally, all the optimal parameters are “adaptive” and hidden layer sizes =1000. It seems obvious that both clustering algorithms fail to make improvement while LDA makes a significant improvement in best score and F1-score. In terms of the running speed, all the algorithms make improvement somewhat. Yet, RP, LDA and the clustering algorithms do better at the run time. Thus, this data sets features are presumably more linear than I thought, while reducing trivial explained variance by PCA and capturing non-Gaussianity by ICA have not performed as nicely as expected. Meanwhile, all the other clustering methods have not shown significant power in dimensional reduction.



	learning_rate	hidden_layer_sizes	best_scores	F1_score
baseline	adaptive	1000	0.735	0.74
pca	adaptive	1000	0.687	0.69
ica	adaptive	1000	0.74	0.74
rp	adaptive	1000	0.741	0.75
lda	adaptive	1000	0.836	0.83
em	adaptive	1000	0.735	0.75
kmeans	adaptive	1000	0.673	0.63

Conclusion:

In summary, all the dimensional reductions algorithms and clustering algorithms have been great assets to transform features for supervised learning and to describe data set in unsupervised learning. In terms of the 4 dimensional reductions methods, it is easy to understand and interpret PCA and LDA, while the former is capturing explained variance, the latter seeking linearity. However, it is worthwhile to notice that RP could perform well if enough computation and time is consumed. Furthermore, it may be hard to utilize ICA and LDA in real world application. It is not easy to interpret the results in ICA while LDA relies on the preprocessing of a data set as it requires Gaussian distributed input variables. It is also difficult to evaluated ICA since kurtosis measure may be less sensitive to outliers. Taking speed performance into account, RP is intuitively the best due to its complete randomness nature while PCA captures the maximum of explained variance. As for the clustering algorithms, EM performs well statistically yet with a lack of speed performance though it could be improved by integrating with gradient methods. K-means has the best descriptive power as all the distance measures applied in the first data set make sense, whereas it is not the same case for EM’s IC criteria measure. Most importantly, it is simple to understand K-means and fast to run clustering analysis on it.

"2.3. Clustering¶." 2.3. Clustering — scikit-learn 0.18.1 documentation. N.p., n.d. Web. 02 Apr. 2017. <http://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation>.