GTID: ywang3564

Assignment 3: Unsupervised Learning and Dimensionality Reduction

Summary

This report includes following analysis: 1) I first used unsupervised methods to cluster on dataset1 and dataset2 (*K-mean and EM algorithms*); 2) I projected dataset1 and dataset 2 to different dimension, with four different dimension reduction technique: (*PCA/ICA/RF/RP*) 3) Re run the clustering with dimension reduced data; 4) Rerun NN with dimension reduced dataset1; 5) Rerun NN with cluster labels of dataset1; Then the result was compared with the value from assignment 1 .

Introduction of background and dataset

The dataset I am using is **Titanic problems from Kaggle**, which includes demographic and detailed information for each passengers. Among all 2,200 passengers in Titanic, only ~700 were rescued after the sinking of Titanic. It was surprisingly to find that some factors greatly increased the rescue possibility of each passenger. So the problem itself is very interesting to explore. Furthermore, the size of the dataset is moderate and the number of features is low. Therefore, it's easier to apply different algorithm on it and compare their performance. Additionally, a lot of people working on Kaggle is improving the performance of this problem; it could be very helpful to bring their experience in my analysis.

The whole dataset has 892 records with ground truth. After removing the NA values, the total number is 714. To make it convenient for all algorithm, I recorded following factors to binary: Cabin class, Gender. Specifically for age and ticket fare, I re-scaled the value with (x-min)/(max-min) to make sure the range is from 0 to 1. The dataset1 is consist with factors of: Age, Gender, Fare, Class, Survival Rate. Those factors were selected based on previous analysis, with high weight on the final survival rate. On the other hand, dataset2 includes more factors than dataset1: Embarked, SibSp, and Parch. Since these factors are skipped in dataset1, it's very interesting to re-check the importance of these factors with unsupervised learning.

Method

- Apply python sklearn package and use Cmaron pipeline.

Following evaluation methods are preferred:

- Silhouette Score;
  - "*The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from −1 to +1, where a high value indicates that the*

    object is well matched to its own cluster and poorly matched to neighboring clusters."[1] The model with the highest Silhouette score is preferred.

- Bayesian information criterion (BIC);
    - *"BIC is a criterion for model selection among a finite set of models; the model with the lowest BIC is preferred. It is based on the likelihood function and introduces a penalty term for the number of parameters in the model to balance the performance and overfitting."* [2]
- Adjusted mutual information (AMI);
    - *"Adjusted Mutual Information (AMI) is an adjustment of the Mutual Information (MI) score to account for chance. It accounts for the fact that the MI is generally higher for two clusterings with a larger number of clusters, regardless of whether there is actually more information shared. For two clusterings and , the AMI is given as: AMI(U, V) = [MI(U, V) - E(MI(U, V))] / [avg(H(U), H(V)) - E(MI(U, V))]"*[3]

I Clustering algorithm

1.1 K-means clustering (KM)

K-means clustering is an unsupervised learning technique. The algorithm iteratively assign data point to one of the K clusters based on distance of the point and the K cluster. The number of cluster K should be clarified first, and the data points in each cluster are updated dynamically. Then with the centroids of the clusters, each train data or new data can be assigned to specific clusters. The KM was implemented with the python sklearn package.

1.2 Expectation Maximization (EM)

The Expectation Maximization algorithm is designed to find the best parameters for a model, with maximum-likelihood estimates. It iteratively updates the parameters with two steps: expectation step and maximization step. For the expectation step, it creates a function for the expectation of the log-likelihood evaluated with current parameters. While for the maximization step, it computes parameters maximizing the expected log-likelihood found on the expectation step. The EM was implemented with the python sklearn package.

[1] https://en.wikipedia.org/wiki/Silhouette_(clustering)
[2] https://en.wikipedia.org/wiki/Bayesian_information_criterion
[3] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_mutual_info_score.html

2 Clustering analysis

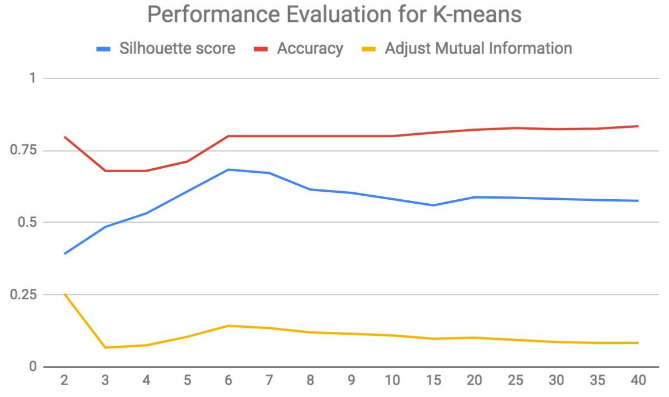
In this topic, I will compare the performance of two algorithms: K-means (KM) and Expectation Maximization (EM).

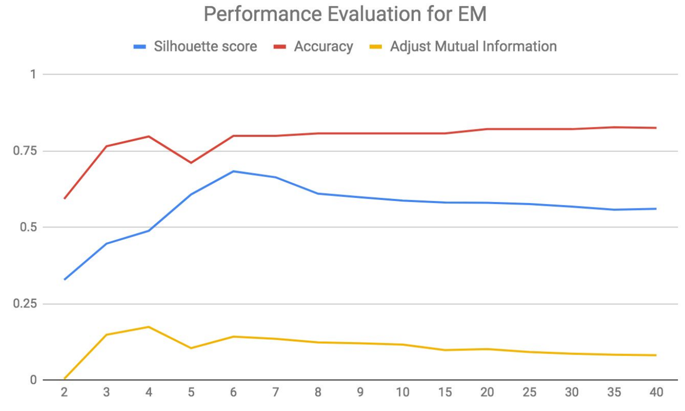
2.1 Dataset1

Figure 1 shows different evaluation metrics for dataset 1 on two clustering algorithms. KM includes three different evaluation methods: silhouette score, accuracy and AMI. 1) The silhouette score reached a peak when cluster number is 6. 2) The accuracy had a dramatic decrease when the cluster number change from two to three. Then it slowly increased when the cluster number increase and displayed a "V" curve.. The extreme condition will be one data is assigned to on clusters. 3) The AMI had a similar trend with accuracy: with the increase of the cluster number, the AMI have a lower score.

EM includes five different evaluation: silhouette score, accuracy, adjust mutual information(AMI), BIC and loglikehood. 1) The Silhouette score had a highest score when cluster number equal to 6. 2) The BIC also had a lowest value when K= 7. 3) For other three evaluation, they have a similar trend: have a better performance when cluster number increased. Those evaluations didn't balance the parameters number and performance.

Conclusion: KM and EM gave the best K: 6. The number is little higher than expected, because we only have two prediction results: survival or not. But it may indicate the real passenger groups in Titanic. And the survival rate for each groups may have be different





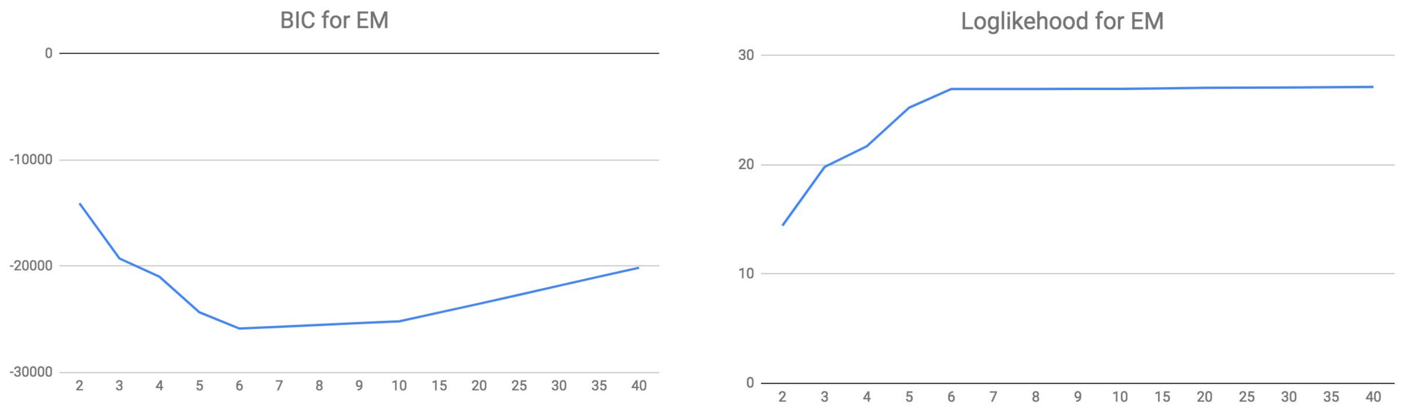


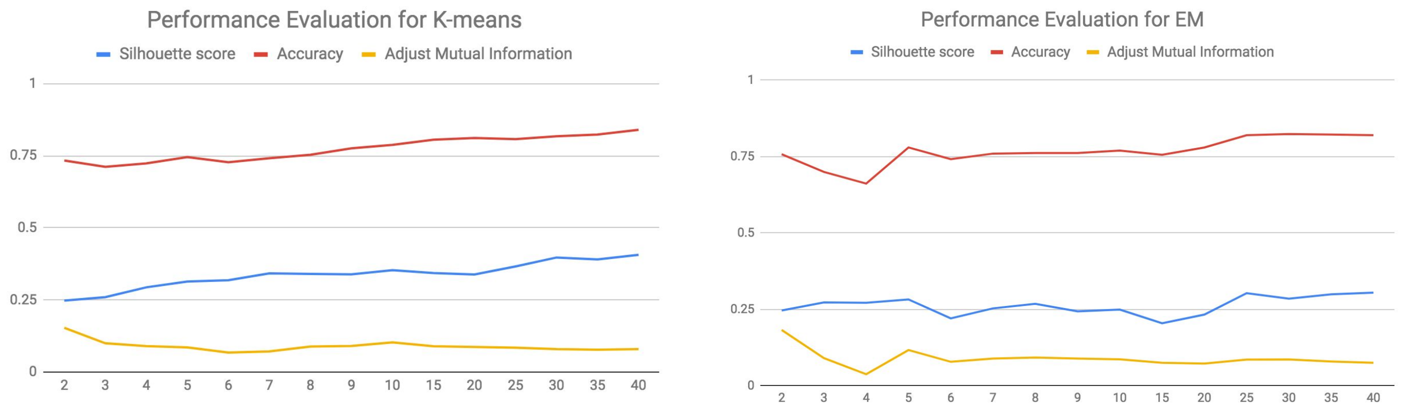
Fig1 Clustering Evaluation for K-means and EM (dataset1)

2.2 Dataset 2

Interestingly, dataset2 showed a different pattern with dataset1. In general, dataset2 had bad performance on unsupervised learning methods. This is true because features in dataset2 are considered as noise and we already removed it. It's very similar to the process of dimension reduction.

Specifically, for each evaluation feature, dataset 2 showed a similar pattern with dataset1, except BIC. I can not find the best BIC before 40 and the BIC may decrease when the cluster number increases.

<u>Conclusion: Both KM and EM didn't give the best K because the silhouette score continue to increase when number of cluster increased.</u>



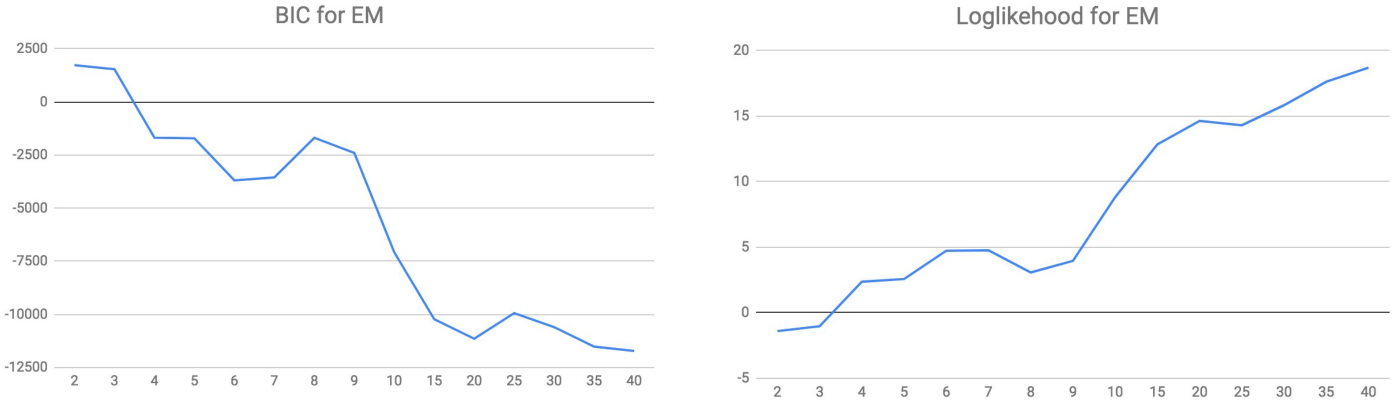


Fig2 Clustering Evaluation for K-means and EM (dataset2)

3. Dimension Reduction

In the first part, I did dimensionality reduction of the two datasets with four different algorithms. Let me start with some background of these algorithms.

Principal Component Analysis (PCA)

Principal Component Analysis is a linear dimension reduction technique. It is a projection based method which transforms the data by projecting it onto a set of orthogonal axes. In practice, the covariance matrix of the data is constructed and the eigenvectors on this matrix are computed. The eigenvectors that correspond to the largest eigenvalues can be used to reconstruct a large fraction of the variance of the original data.[4]

Independent Components Analysis (ICA)

Independent Components Analysis is a linear dimension reduction method, which transforms the dataset into columns of independent components. It assumes that each sample of data is a mixture of independent components and it aims to find these independent components. A simple application of ICA is the "cocktail party problem", where the underlying speech signals are separated from a sample data consisting of people talking simultaneously in a room.[5]

Randomized Projections (RP)

Random projection is a technique used to reduce the dimensionality of a set of points which lie in Euclidean space. Random projection methods are powerful methods known for their simplicity. It is used for distance based method and pairwise distance between any two instance of dataset is preserved. This is done by varying dimensions and distributions of random projections matrices.[6]

Random Forest (RF)

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the

[4] https://en.wikipedia.org/wiki/Principal_component_analysis
[5] https://en.wikipedia.org/wiki/Independent_component_analysis
[6] https://en.wikipedia.org/wiki/Random_projection

classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.[7]

3.1 Dataset1

Figure 3 shows the projection results by PCA, ICA, RP and RF for dataset 1. The number of dimensions are chosen based on the highest accuracy/weight/Kurtosis. Specifically, PCA gave a different best K: 4, all others gave best K: 5. In the PCA analysis, the top 4 components has explained the majority of variance, and component 5 and 6 have no contribution at all. In the ICA analysis, kurtosis is a measure of the "tailedness" of the probability distribution of a real-valued random variable. The biggest kurtosis was observed when K equals to 5. In RP analysis, I plotted the pairwise distance correlation coefficients of each component and chose the one with the biggest value. In the RF analysis, the feature importance of each component suggested that best K is 5.

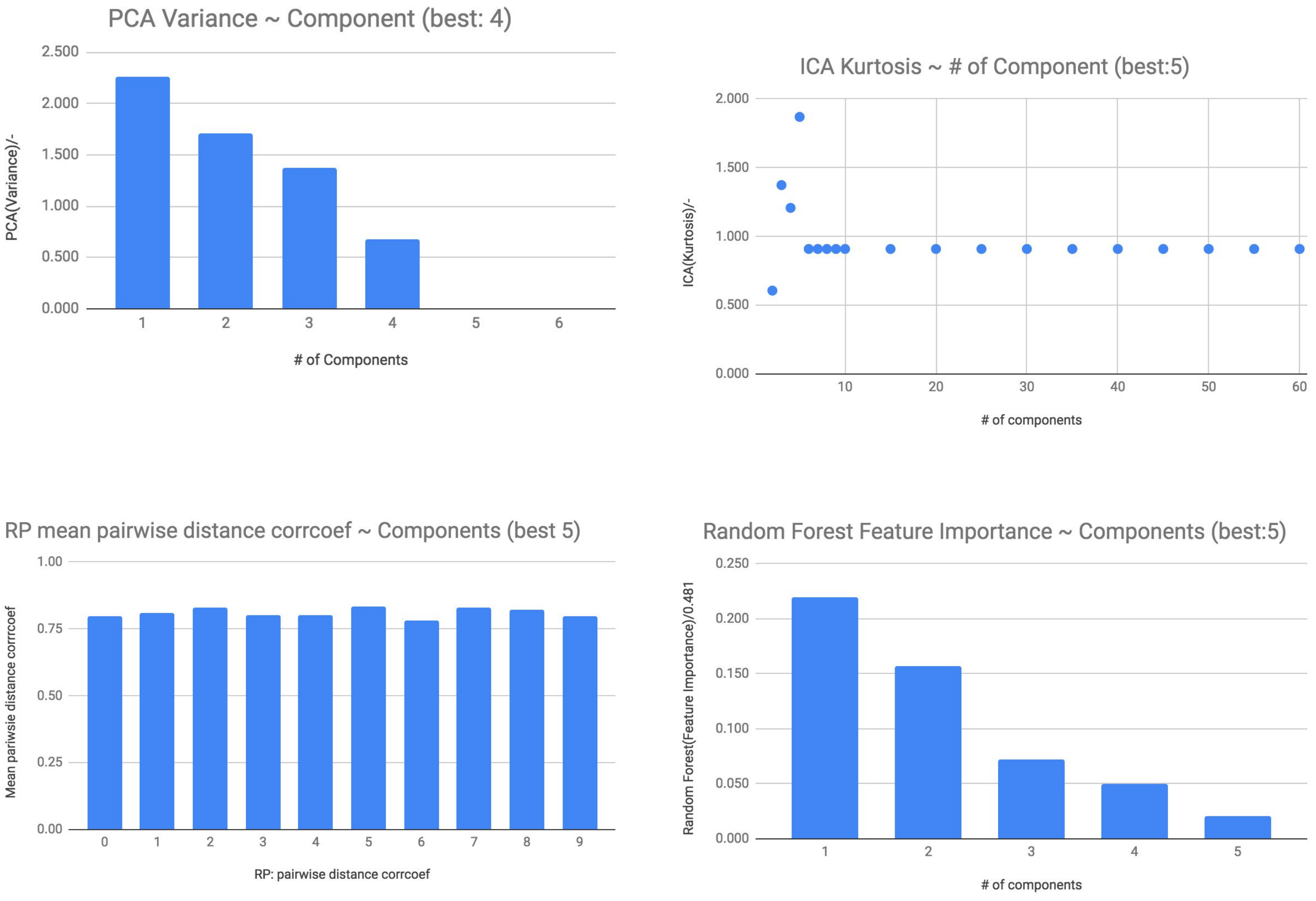


Fig3 Dimension reduction for dataset1

Similar to the dataset1, I used the same evaluation parameters for dataset2, and the best K are listed as:

- PCA: 6;
- ICA: 5;

[7] https://en.wikipedia.org/wiki/Random_forest

- RP: 8
- RF: 5

The result is not consisted of different methods, indicating the internal heterogeneity of dataset2.

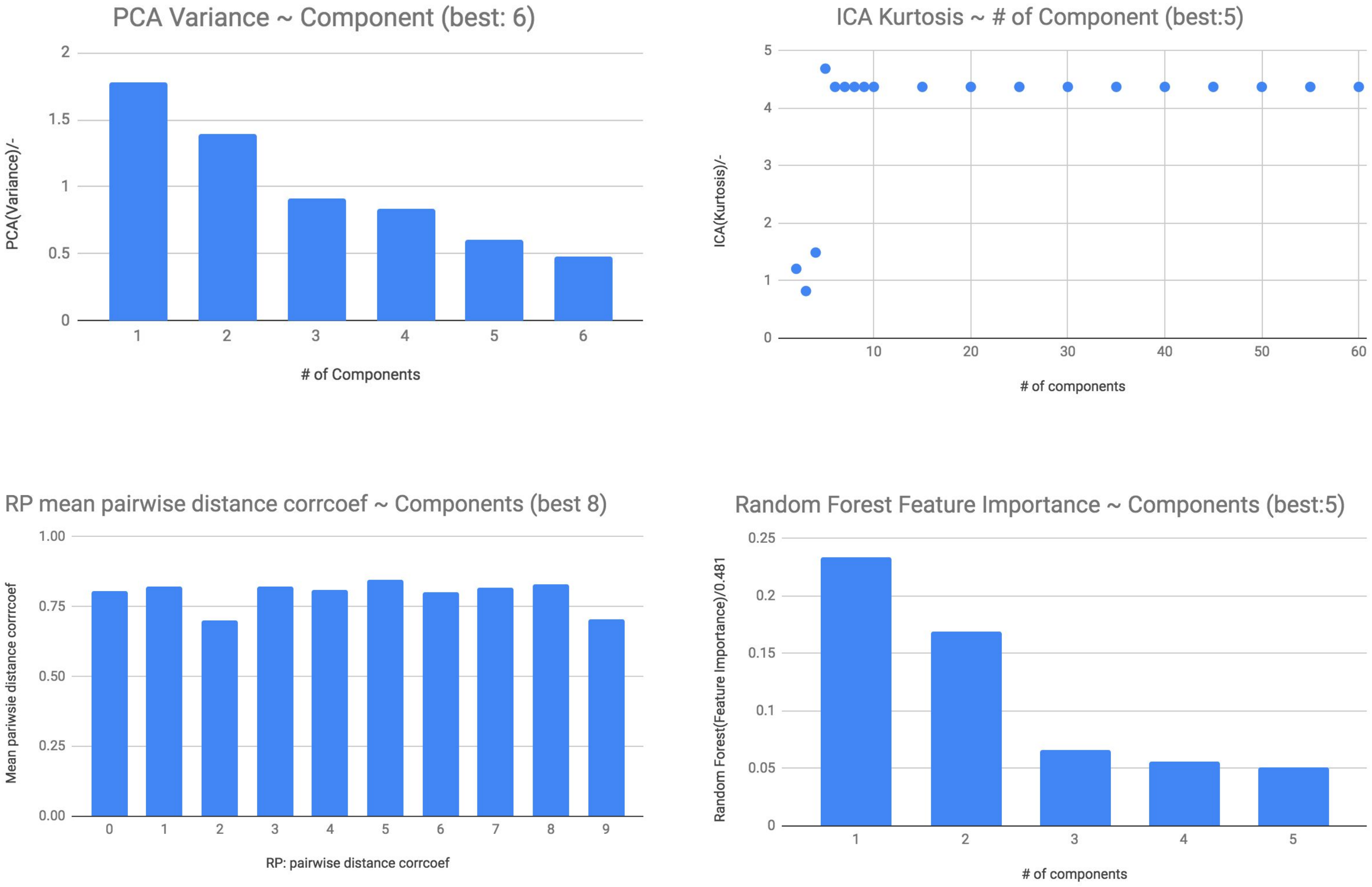


Fig4 Dimension reduction for dataset2

4. Re-run Clustering after dimension reduction

With the inferred dimension from part 3, I changed the dimension for the clustering algorithm and re-run the cluster for dataset1 and dataset2.

The dimension used for dataset 1 clustering:

- PCA: 4
- ICA: 5
- RP: 5
- RF: 5

4.1 Dataset1

Figure 3 shows different evaluation metrics for dataset 1 on two clustering algorithms. KM includes three different evaluation methods: silhouette score, accuracy and AMI. EM includes five different evaluation: silhouette score, accuracy, adjust mutual information(AMI), BIC and loglikehood.

1) The silhouette score (KM and EM) reached a peak when cluster number is 6.
2) The BIC (EM) had a lowest value when K= 6.
3) For other three evaluation, they have a similar trend: have a better performance when cluster number increased. Those evaluations didn't balance the parameters number and performance. For example, the accuracy may still increase when the cluster number increase. The extreme condition will be one data is assigned to on clusters.

<u>Conclusion: By comparing my result before dimension reduction and after, the best K value and evaluation didn't change so much. The main reason maybe: the dimension reduction has dramatic effect on high dimension data, but my dataset has only six dimension</u>

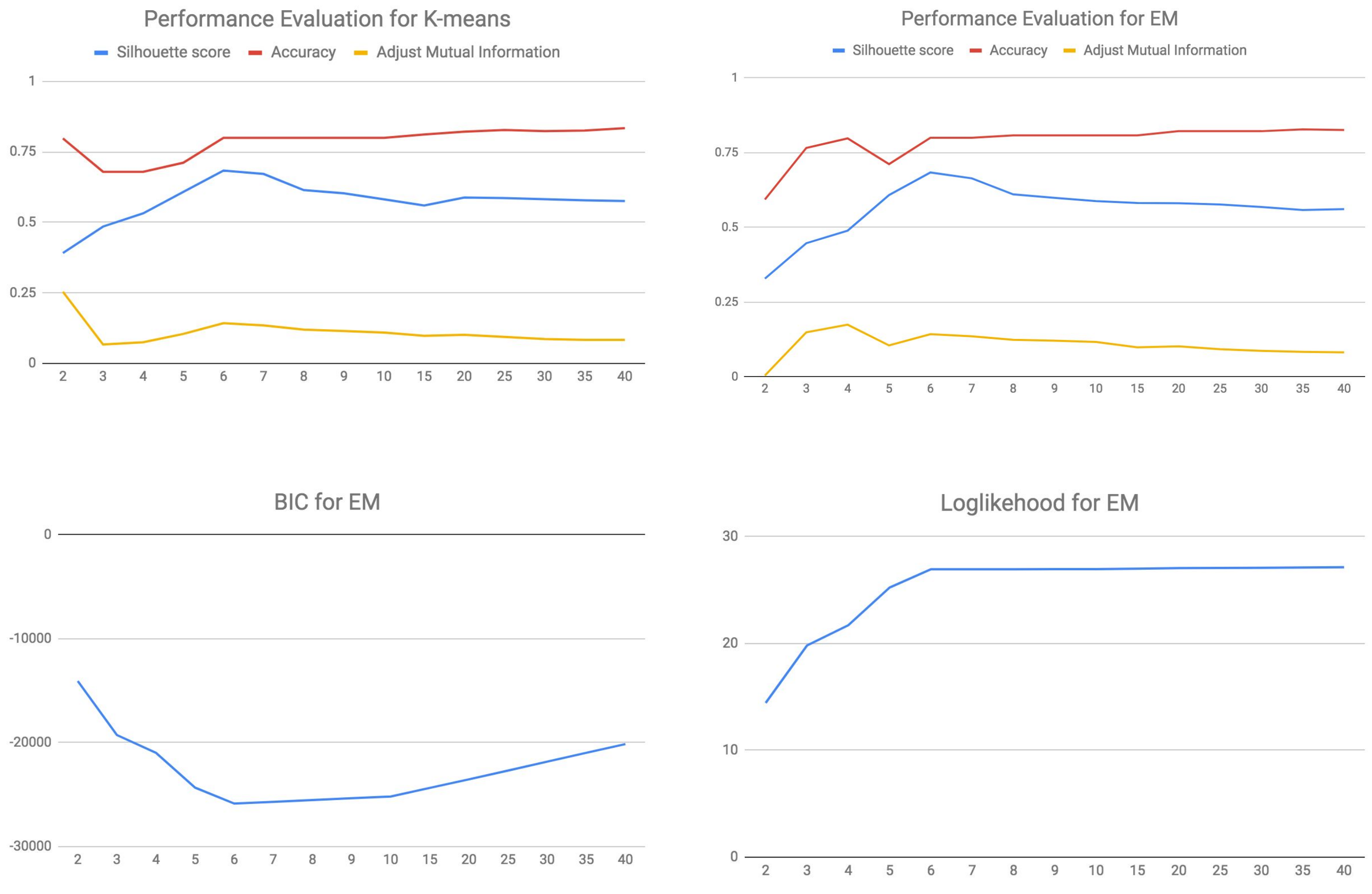


Fig5 Re-Clustering Evaluation for K-means and EM (dataset1)

<u>The dataset 2 has a similar result with dataset1, which means the dimension reduction didn't improve the clustering.</u>

5 Neural network learner on dimensionality reduced dataset and clustered dataset

In the final part, I re-run the neural network analysis with dataset1 from two different scenarios: 1) dimension reduced from four methods; 2) cluster labels from two clustering methods.

The topology and parameters are indicated in the previous assignments: two hidden layers, with five and three nodes respectively, ReLU activation. In this experiment, I used the mean test score to compare different methods. Among all methods, PCA, RP, and RF all have a similar result with labeled clustering result, when K equals to 6. Which means, labeled result from EM or KM actually have a similar power with dimension reduction. EM does not perform very vell as KM and so does ICA. But they can acuqire the best performance when K= 6.

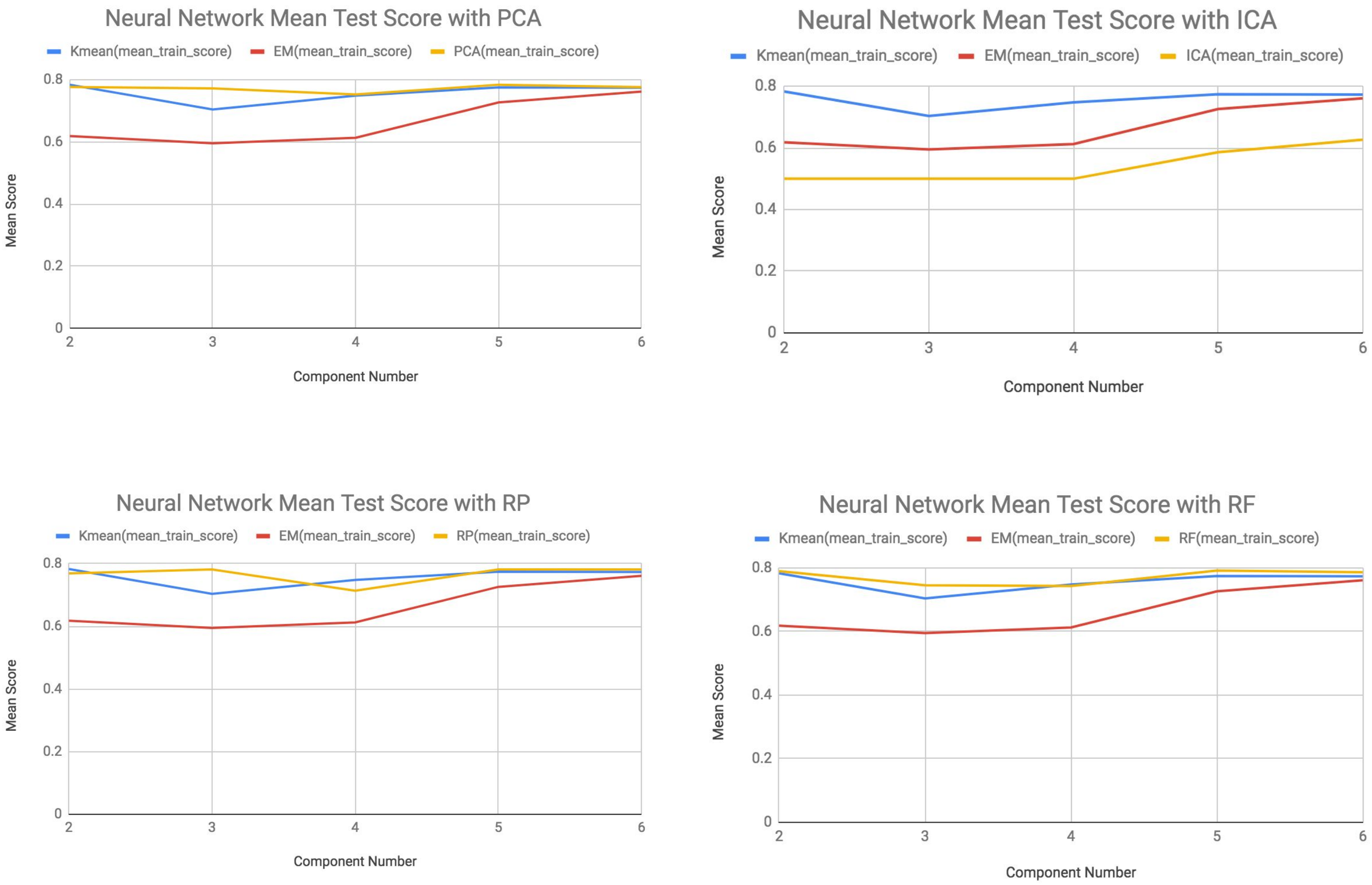


Fig6 Mean training values by neural network learner with different parameters for dataset 1.

Conclusions

In this experiment, I tried the analysis with different clustering and dimensionality reduction algorithms for two datasets. Very interestingly, my result does not acquire a better clustering performance after dimension reduction. But all methods gave me the same K value. For the last part, PCA, RP and RF are more reliable even K is not the best.