November 3, 2019 (F19)
Md Khaled Hassan (ID: mhassan49)
Georgia Institute of Technology

Unsupervised Learning and Dimensionality Reduction

CS7641: Machine Learning (Assignment 03)

Abstract:

The objective of this project is to explore various unsupervised machine learning techniques and dimensionality reduction algorithms, apply them on two different datasets, and understand and analyze their behavior on these datasets (part I and II of this report). In addition, I have also applied these algorithms to the neural network that we explored in assignment 1 and compared the results with the original performance parameters such as cross validation score and fit time (part III). The Introduction section of this report is adopted from my assignment I since I am using the same datasets for all the experiments in this assignment.

Introduction:

The datasets that we explored in this projects are available from UCI machine learning repository [1] . The first dataset that we considered is *EEG (electroencephalography) eye state dataset* (Link: https://archive.ics.uci.edu/ml/datasets/EEG+Eye+State). This data set determines the eye state of a person and a binary class dataset ('1' indicates the eye being closed and a '0' indicates the eye being opened). The dataset has 14980 instances and 14 attributes. The second dataset we explored in this project is the *Letter Recognition dataset* (Link: https://archive.ics.uci.edu/ml/datasets/Letter+Recognition). This is another interesting dataset since image recognition has gained a very wide range of applications over the last decade. This is a multi-class dataset that identifies each of the 26 alphabets based on 16 attributes. There are 20000 instances of letter recognition provided in this dataset. The algorithms that we applied are as follows and their performances were analyzed in the following sections:

- KMeans Clustering Algorithm (KM)
- Expectation Maximization (EM) with Gaussian Mixture Model
- Principal Component Analysis (PCA)
- Independent Component Analysis (ICA)
- Random Projection Analysis (RPA) with Sparse Random Projection
- Linear Discrimination Analysis (LDA)
- Neural Network (Multi-layer Perceptron (MLP))

Data Processing:

I have implemented the algorithms explored in this assignment using Python's scikit learn library [2-5]. For the neural network (NN) analysis, I have used the letter recognition dataset and 75% of which are used for training purpose to be consistent with assignment 1. In addition, we have applied 10 fold cross validation (10 fold CV) on the training data set to determine the accuracies of the neural networks.

1. Run Clustering Algorithms

In this section, two clustering algorithm, namely KMeans (KM) clustering and expectation maximization (EM) are being analyzed by applying them the two datasets we briefly discussed before. These are unsupervised machine learning techniques. The KM algorithm uses Euclidean distances to label the the clusters. The EM is an iterative algorithm to find maximum likelihood of parameters in a statistical model [6]. In this project, we have analyzed EM algorithm based on Gaussian mixture model. We evaluated the KM algorithm based on the sum of squared error (SSE) determined using elbow method the the homogeneity, completeness, and adjusted mutual info (AMI) scores. Homogeneity score indicates if all of the clusters contain only data points that are members of a single class while the completeness score indicates if all the data points that are members of a given class are elements of the same cluster. In addition, the AMI score is an adjustment of the mutual information score. Regardless of the amount of information being shared, this score assumes that mutual information increases with the size of the clusters [2-5]. The EM algorithm was evaluated using the weighted log likelihood of each of the samples in the dataset and the Akaike information criterion (AIC) and the Bayes Information criterion (BIC) are used to evaluate the algorithm. Both AIC and BIC scores measure how close the model is to the true: lower values being closer to the truth [7].

EEG eye state dataset:

In Fig.1 (a) and (b), I have plotted the SSE and KM scores. The EM scores and EM average log likelihood are plotted in (c) and (d), respectively. All of these parameter are plotted as a function the number of clusters or components. For KM algorithm, we observe that the SS Error becomes 0 when the cluster number is greater than 3. In case of the EM algorithm, we see similar trend. When the cluster or components number is greater than 3 or so, the scores and the average likelihood starts to become saturated. We also observe that both AIC and BIC scores in close agreement which is also expected. Since the dataset only contains 2 classes, these results make sense because all these parameters indicates saturation when the cluster or components numbers is close to 2 or higher.

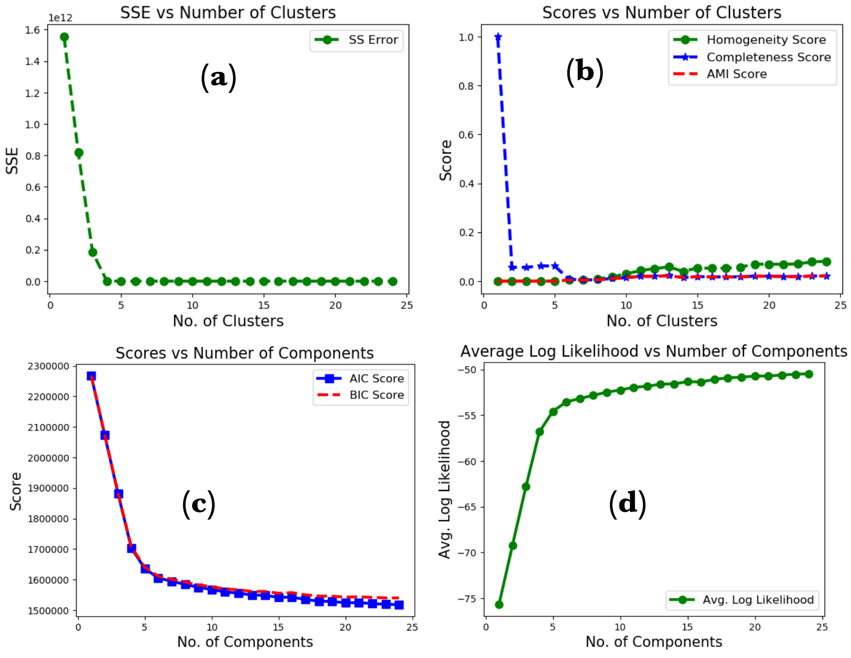


Fig.1: EEG eye state: (a) KM SSE (b) KM Scores c) EM Scores (d) EM Likelihood

Letter Recognition dataset:

We observe similar trend in both KM and EM evaluation parameters as that of the EEG eye state dataset. From the plots in Fig 2, we observe that the parameters start becoming saturated when the cluster or components number is at around 25 or so. The letter recognition dataset contains 26 different classes and therefore a number close to 25 is also expected in this case.

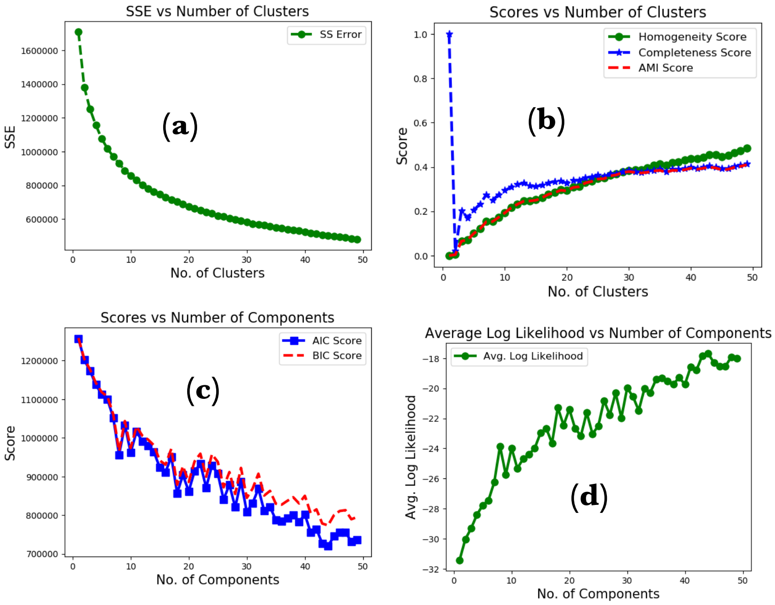


Fig.2: Letter Recognition: a) KM SSE (b) KM Scores c) EM Scores (d) EM Likelihood

2. Run Dimensionality Reduction Algorithms

We explored four different dimensionality reduction algorithms in this section: Principle Component Analysis, Independent Component Analysis, Random Projection Analysis, and Linear Discriminant Analysis. The objective of these algorithms are to reduce the number of feature sizes (dimensions) without losing any accuracy of the output as well as reduce the computation time.

1. Principle Component Analysis (PCA): PCA uses singular value decomposition of data (SVD) and project that to a lower dimensional space[2-5]. The algorithms evaluated using variance and singular values (Eigen values) as a function of the number of dimension.

EEG eye state dataset:

Fig 3(a) shows the variance (left y-axis) and the Eigen values (right y-axis). Both parameters approach 0 when the number of components are greater than 3. Therefore, I have chosen 3 to reduce the dimension of the dataset and using the reduced dataset, the KM and EM algorithms are evaluated. We have explored the same parameters (Fig 3 (b)-(e)) as we did in the previous section of this assignment for these two clustering algorithms and observe no qualitative difference and very little quantitative difference with the parameters for the full dataset (Fig .1). All parameters become saturated when the number of clusters are 3 or so. Therefore, PCA performs great on these dataset to reduce the number of features.

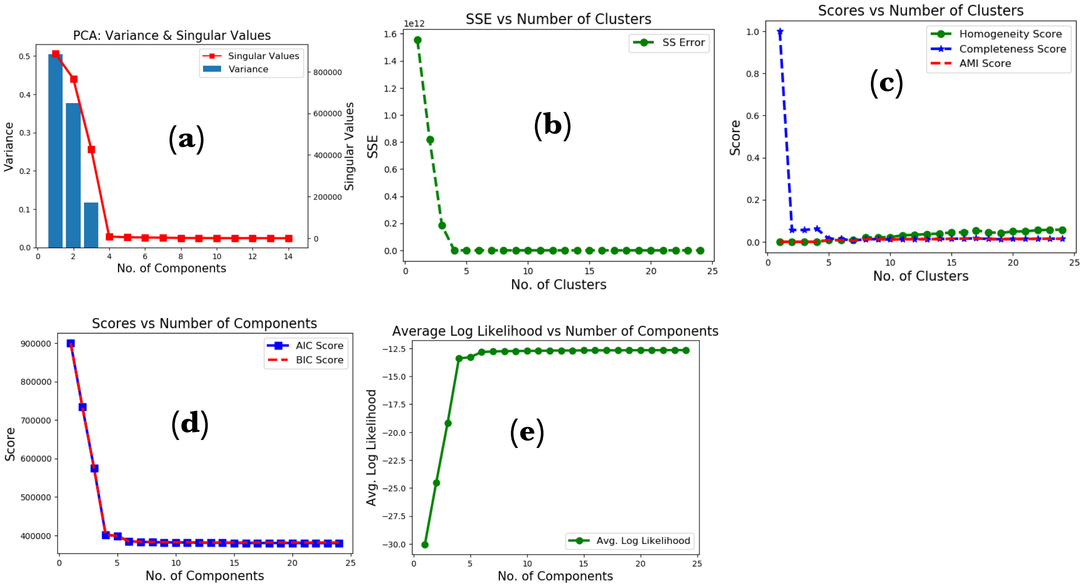


Fig.3: PCA on EEG eye state: (a) Variance and singular values vs dimensions (b) KM SSE (c) KM Scores (d) EM Scores (e) EM Likelihood

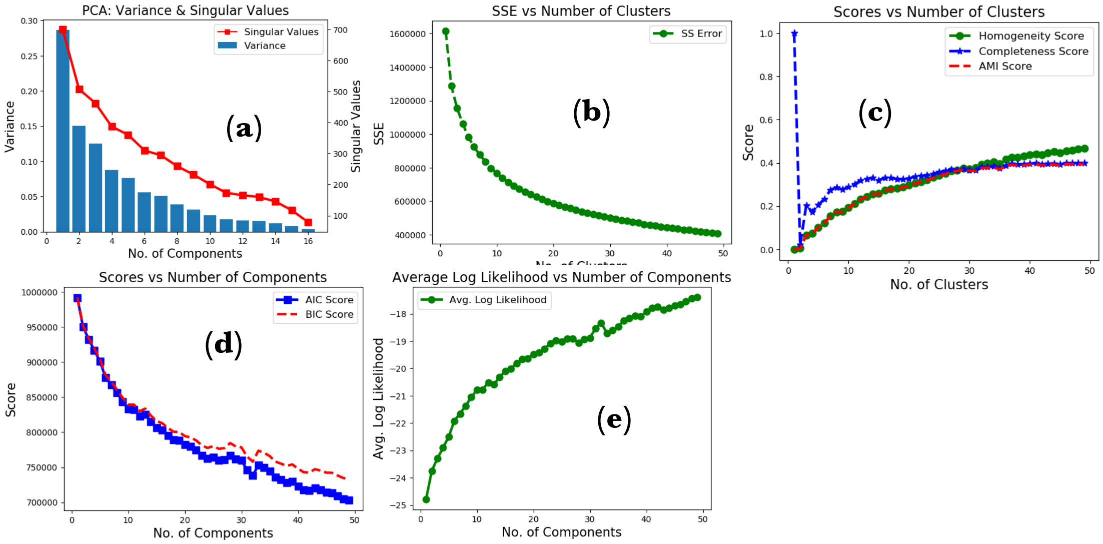


Fig.4: PCA on Letter Recognition: (a) Variance and singular values vs dimensions (b) KM SSE (c) KM Scores (d) EM Scores (e) EM Likelihood

Letter Recognition dataset:

Similar analysis is carried out on the letter recognition dataset. From Fig. 4 (a), we observe that the variance and the Eigen values become very small or negligible when the number of dimensions is close to 10. I have chosen 11 to reduce the dataset and applied the clustering algorithm on the reduced dataset. We observe similar trend in the SSE, Avg. likelihood, and score parameters. A similar saturation level of the parameters is also observed when the cluster size is close to 25. Therefore, PCA does well on both binary and multi-class datasets.

2. Independent Component Analysis (ICA): ICA is a statistical technique that can be used to model the dataset as a linear mixture of non gaussian and mutually independent components [2-5, 7] and reduce the number of dimensions of the original dataset in the process. The algorithm is evaluated using the kurtosis (which is a measure of the non gaussianity in a dataset) of the dataset as a function of the number of independent components.

EEG eye state dataset:

Fig. 5(a) shows the Kurtosis of the dataset as a function of the number of independent components. We observe that the kurtosis doesn't change much when the number of independent components is 9 or greater. I have chosen 10 to reduce the original dataset to re-run the clustering algorithms (KM and EM). We observe that the SSE (Fig.5(b)) does not become saturated at 3. From the score parameters and the average likelihood (Fig. 5(c)-(e)), we see that the number of clusters need to be at least 5 for the parameters to be saturated. Therefore, compared to PCA, ICA does somewhat worse on this dataset.

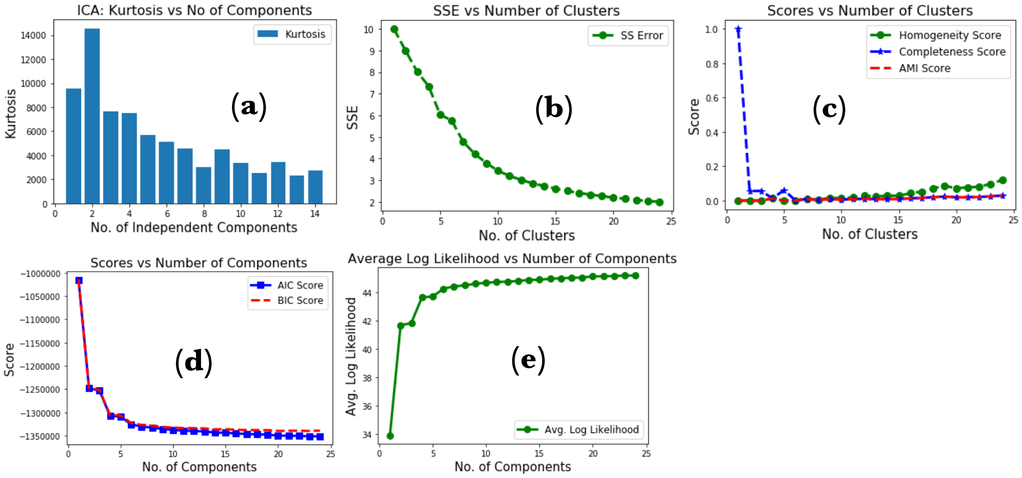


Fig.5: ICA on EEG eye state: (a) Kurtosis vs no of independent components (b) KM SSE (c) KM Scores (d) EM Scores (e) EM Likelihood

Letter Recognition dataset:

Fig 6(a) who's the kurtosis of the dataset as function of the number of independent components. Although we observe negative values the the independent components are lower than 5, the plot reveals that the data is shows additional kurtosis for all independent components. I have chosen 13 as the number of components to reduce the dataset dimension and re-ran the clustering algorithms. From the evaluation parameters of KM (Fig. 6(b)-(c)) and EM (Fig. 6(d)-(e)) that SSE and average likelihood do not become saturated even for a very large number of clusters. Based on these plots, we observe that ICA performs somewhat comparable to PCA.

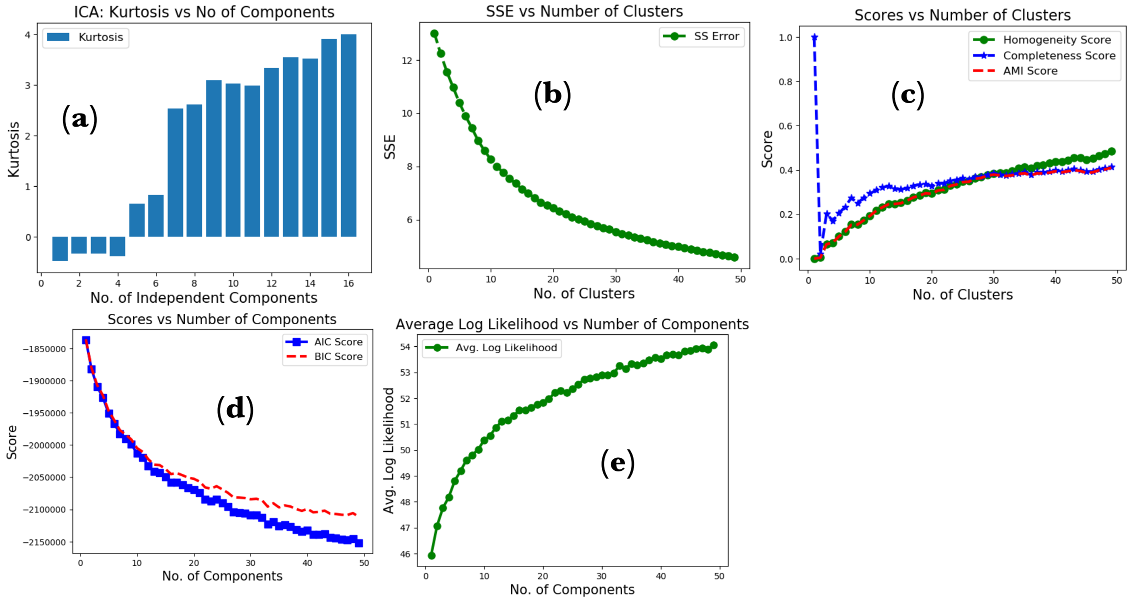


Fig.6: ICA on Letter Recognition: (a) Kurtosis vs no of independent components (b) KM SSE (c) KM Scores (d) EM Scores (e) EM Likelihood

3. Random Projection Analysis (RPA): RPA is dimensionality reduction technique that reduces dimension at the expense of additional variance to the dataset. I have leveraged the sparse random projection model in python's scikit-learn library [2-5] for this section. The algorithm was evaluated using the pairwise squared distance rates as a ruction of the number of components. We have also used 5 deferent random states to evaluate the variation due to different random seeds.

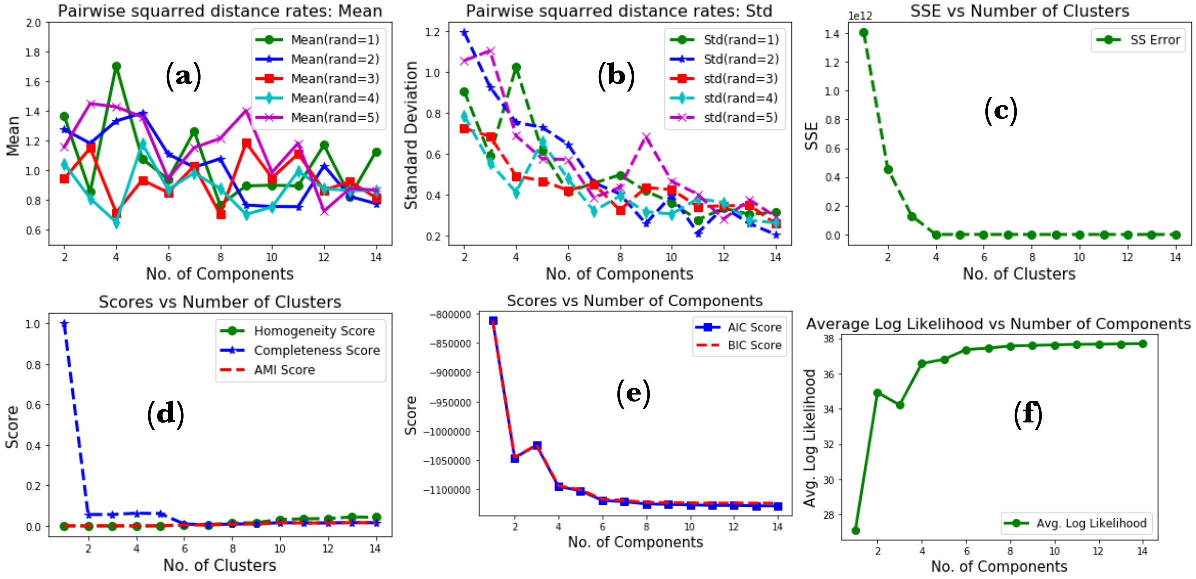


Fig.7: RPA on EEG eye state: (a) pairwise squared distance rates (Mean) (b) pairwise squared distance rates (Standard deviation) (c) KM SSE (d) KM Scores (e) EM Scores (f) EM Likelihood

EEG eye state dataset:

Fig 7 (a) and (b) shows the mean and standard deviation of the pairwise squared distance rates as a function of the number of components. We have also run the same simulation with five different random seeds (rnd=1-5 in the plots). We see that running the algorithm multiple times generate different results. However, the average trend remains same and when the number of components is 8 or above, the standard deviation and also the variation due to random seeds becomes very small. Therefore, we have chosen 8 as the dimension of the reduced dataset and re-ran the KM and EM clustering algorithms. From the SSE (7(c)) and average likelihood (7(f)) plots, we see that the parameters become saturated when the cluster number reaches 3 or above. This is very consistent with what we observed in the original dataset. We observe similar consistency in the score parameters (7(d)-(e)). Based on the analysis, we conclude that RPA does very well in dimension reduction on this dataset.

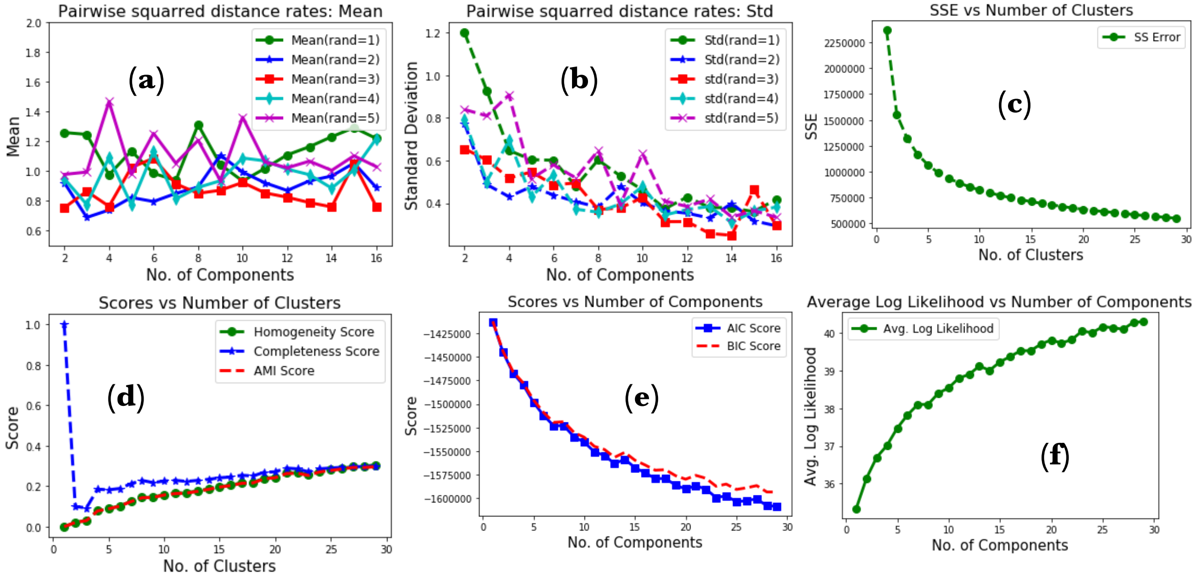


Fig.8: RPA on Letter Recognition: (a) pairwise squared distance rates (Mean) (b) pairwise squared distance rates (Standard deviation) (c) KM SSE (d) KM Scores (e) EM Scores (f) EM Likelihood

Letter Recognition dataset:

Similar variation due to random seed in pairwise squared distance rate is observed in Fig 8 (a)-(b). Based on the standard deviation, we have chosen 10 as the number of components for the reduced dataset and ran the clustering algorithms again. We do observe that the SSE (8(c)) and average likelihood (8(f)) have increased compared to the original dataset. Since the random projection algorithm compromises variance to improve computation speed, the results are somewhat expected. All evaluation parameters (8(c)-(f)) become saturated when the cluster number is close to 26. This indicates, the accuracy of the algorithms are still preserved with RPA in the reduced dataset.

4. Linear Discriminant Analysis (LDA): LDA is another dimension reduction technique and a classification algorithm that leverages Bayes' rule to fit class conditions densities [2-5]. The reduction algorithm is evaluated based on the Eigen values (variance) as a function of the number of components.

EEG eye state dataset:

Plot 9(a) shows that the variance is 1 regardless of the number of components used in this algorithm. This is somewhat interesting. It is impossible to pick up a suitable number for the reduced dataset from this plot. I have randomly picked up 5 and re-ran the clustering algorithm. The dataset shows lower SSE (9(b)) for KM algorithm compared to the full dataset. The sharp decrease in completeness score in 9(c) indicates that all data are clustered in one single label. The AIC and BIC scores (9(d)) disagrees even when the cluster number is very small. The average likelihood with EM algorithm (9(e)) is also very small. Based on all these plots, my conclusion is this dataset is not suitable for this reduction technique possibly because the algorithm has bias.

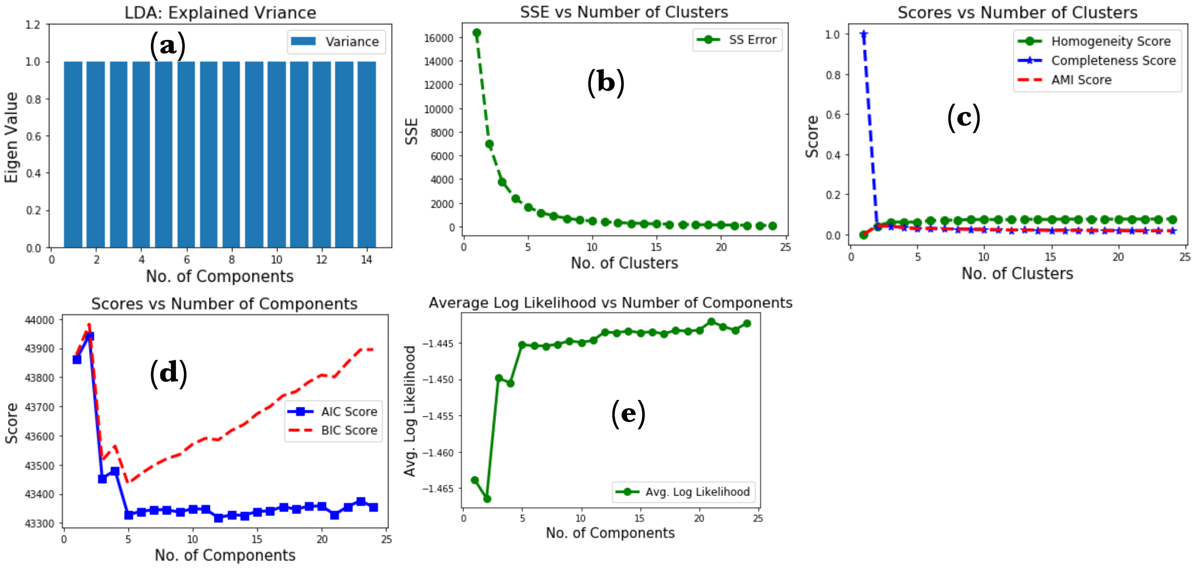


Fig.9: LDA on EEG eye state: (a) Variance (b) KM SSE (c) KM Scores (d) EM Scores (e) EM Likelihood

Letter Recognition dataset:

Using similar analysis, we picked up 10 as the number of dimension for the reduced dataset based on the variance (Fig. 10(a)). After re-running the clustering algorithms, we observe that SSE (10(b)) increases while the average likelihood (10(e)) decreases compared to the full dataset. In addition, the completeness score (10(c)) is much higher when the cluster number is less than 25 or so. The LDA seems to perform better in this case (multi-class) compared to the EEG eye state dataset (binary-class).

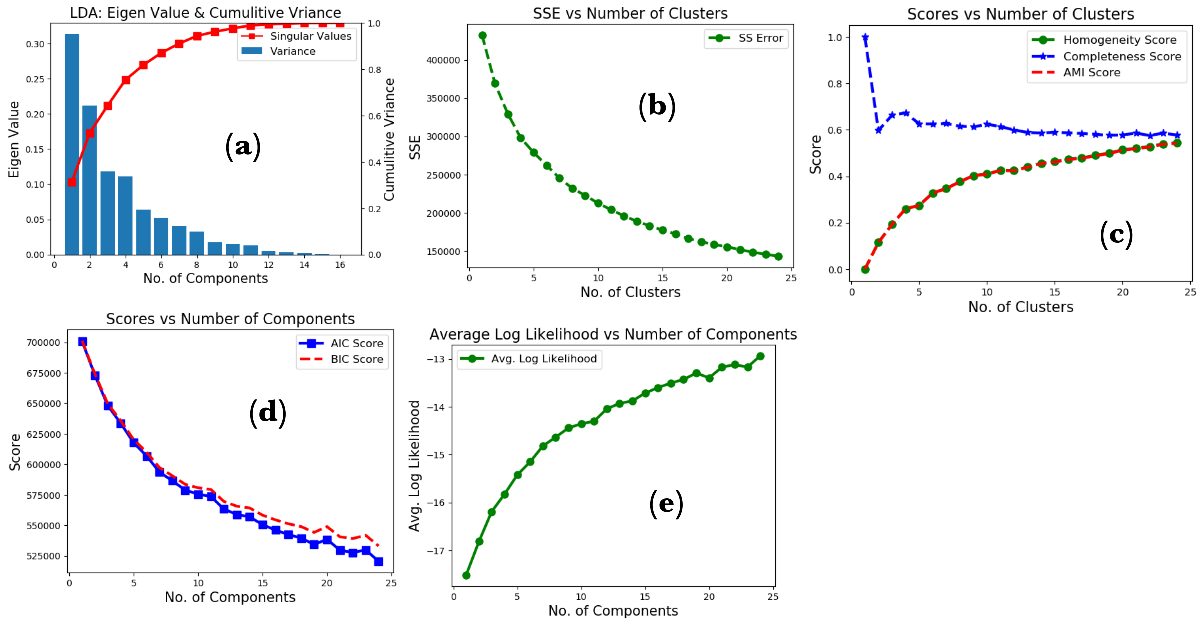


Fig.10: RPA on Letter Recognition: ((a) Variance and cumulative variance (b) KM SSE (c) KM Scores (d) EM Scores (e) EM Likelihood

3. Run Neural Network (MLP Classifier) Analysis

I have chosen the letter recognition dataset for this part and ran various simulations on the neural network (NN) with multi layer perceptron that I tuned in assignment 1.

Letter Recognition dataset:

In this part, Neural network (Multi Layer Perceptron) from assignment 1 was implement using the dimensionally reduced data (only for the letter recognition dataset). The parameter sets are same as that we obtained after tuning the algorithm in assignment 1. The number of hidden layers are set at 100. To be consistent with assignment 1, I am using 75% of the dataset for training purpose. We have explored all four of the dimensionality reduction algorithms that we analyzed in the previous section and compared the 10-fold cross validation (cv-10) score of the training dataset and fitting time with the original tuned network from assignment 1. I have also re-ran the simulations by treating the labels from KM and EM algorithm as new features. I have increased the number of dimensions of the reduced dataset by 2 for each of the algorithms when the label of the clustering algorithms were added to the dataset compared to what we have chosen in the previous section. This is because both KM and EM labels should add useful information in the classification problem. We have listed the accuracies and fit time for each of the cases (with and without the clustering labels in the dataset) in Table 1 and plotted in Chart 01 and Chart 02. We see that PCA performs best among all these algorithms and LDA is second to PCA for both with and without the clustering labels in the dataset. Fit time with the dimensionally reduced algorithms are mostly higher without the clustering labels and somewhat comparable to the original network when the levels from KM and EM are added to the dataset. Based on this analysis, we conclude that adding the labels from clustering algorithms may not add any significant accuracy to the network.

(Table.1: Accuracy and Fit time (Letter Recognition dataset)

Algorithms	Without Clustering Labels		With Clustering Labels	
	CV-10 Accuracy (%)	Fit Time (s)	CV-10 Accuracy (%)	Fit Time (s)
NN (Original)	**88.31**	**9.77**	**86.25**	**8.94**
NN (PCA)	**87.72**	**17.06**	**87.96**	**8.86**
NN (ICA)	**71**	**10.17**	**72.1**	**8.76**
NN (RPA)	**77.28**	**9.59**	**75.63**	**8.31**
NN (LDA)	**84.01**	**10.35**	**85.43**	**9.13**

Chart 01: CV10 Accuracy (Letter Recognition)

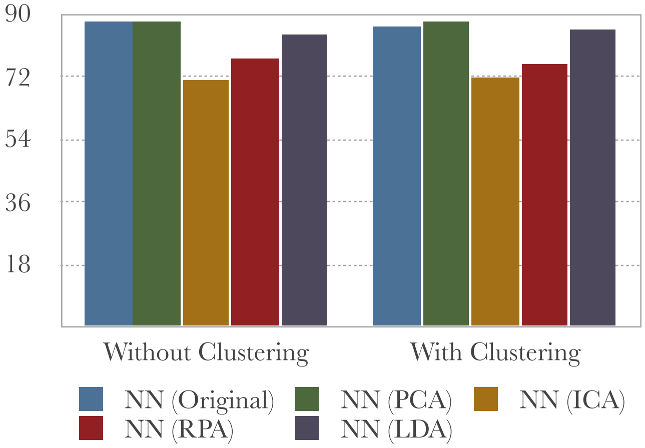
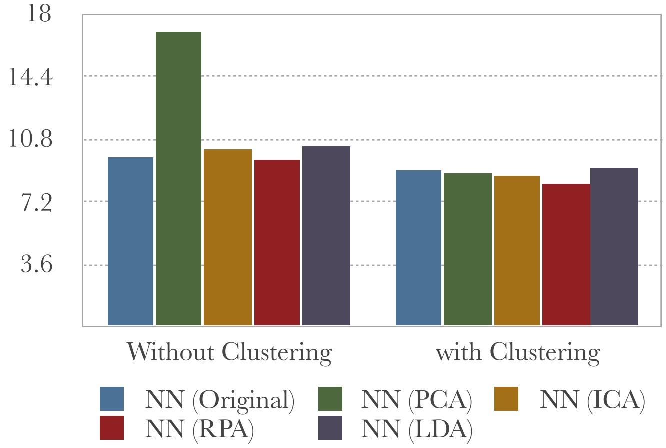


Chart 02: Fit Time (Letter Recognition)



References:

1. UCI repository (https://archive.ics.uci.edu/ml/datasets.php)
2. Scikit-learn (https://scikit-learn.org/stable/index.html)
3. https://stackoverflow.com/questions/19197715/scikit-learn-k-means-elbow-criterion
4. https://scikit-learn.org/stable/auto_examples/plot_johnson_lindenstrauss_bound.html#sphx-glr-auto-examples-plot-johnson-lindenstrauss-bound-py
5. https://scikit-learn.org/stable/auto_examples/classification/plot_lda.html#sphx-glr-auto-examples-classification-plot-lda-py
6. https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithm
7. https://www.methodology.psu.edu/resources/AIC-vs-BIC/
8. https://www.cs.helsinki.fi/u/ahyvarin/whatisica.shtml
9. Udacity lectures of CS7641