Anmol Chhabria – achhabria7
Assignment #3
CS 4641

Unsupervised Learning Assignment

Dataset

Movies

In this assignment, I used both datasets for Assignment #1. One of the datasets being examined in this report is an IMDB dataset containing attributes from 5,000 different movies. These movies have release dates within the past 100 years and pertain to 66 different countries. The dataset included the IMDB score of the movie. Based on rankings in IMDB, movies that are on the top of the charts have a rating of 6 or above. The classification variable in this dataset is whether the IMDB score rating is above a 6 or not. A subset of 14 attributes was used to predict the IMDB score classification. This subset removed variables that had a high number of unique instances (actor names, director names, country, title year) and variables that had the same instance (aspect ratio, color, face number in poster. Furthermore, the dataset had nominal values for the 'language' and 'content rating' attributes, so these were transformed into binary attributes to successfully run all algorithms for this assignment. This resulted in a transformation from 14 attributes to 75 attributes.

MOOC EEG Signals

The other dataset examined in this report is from a study conducted on 10 college students while they watched MOOC videos. This dataset analyzes the electroencephalogram (EEG) signals in a student's brain while the student watches videos. The data has approximately 12,000 instances. A total of 13 attributes were used to predict the classification variable. The classification variable used in the algorithms below is the self-reported data from the student- whether the student was confused or not. The prediction of the confusion classifier is not used as an attribute ('Col14' in README.txt).

Background

K-Means

K-means is an unsupervised learning algorithm that is used to solve clustering problems. The algorithm defines k centroids, one for each of the k clusters. The centroids are typically placed far away from each other to be able to group the data well. Each instance in the data is associated with the closest centroid. The centroids are recalculated after one iteration of the grouping is done. This process continues until the k centroids do not move anymore. The algorithm's objective function is to minimize the squared error between each data point and its respective cluster's centroid.

Expectation Maximization

The expectation maximization algorithm uses maximum likelihood to estimate the log likelihood parameter. The E step of the algorithm involves computing probabilities over each completion of missing data. These probabilities are then used to create a new training dataset, and the M step uses maximum likelihood estimation to re-estimate the log likelihood. The algorithm runs until the parameter estimation converges.

Dimensionality Reduction

Dimensionality reduction is an approach used to reduce the number of random variables in a dataset. This can improve performance by reducing noise in the data, improving runtime, and reduce multi-collinearity. The feature reduction algorithms run on the datasets in this assignment are principal component analysis (PCA), independent component analysis (ICA), and random projection. The feature selection algorithm run in this assignment is information gain.

Clustering Algorithms

Approach

In order to measure performance of the clustering algorithms, the sum of squared errors (SSE) for each run was used to calculate % variance explained. Variance explained is (SSE(k) – SSE(k = 1))/ SSE(k = 1) for all k. Subsequently, the elbow method was used to find the optimal number of clusters for K-means and expectation maximization. The optimal number of clusters represents the value for k after which increasing the number of clusters would result in decreasing returns in variance explained.

K-Means

K-means was run on the *Movie* and *EEG* dataset for 1-15 clusters. Looking at the graph, it can be inferred that an increasing number of clusters results in decreasing error. A higher number of clusters is able to represent unique features in more subsets of the data, but can also result in overfitting. As the number of clusters approaches the number of instances in the data, each cluster provides less meaningful information.

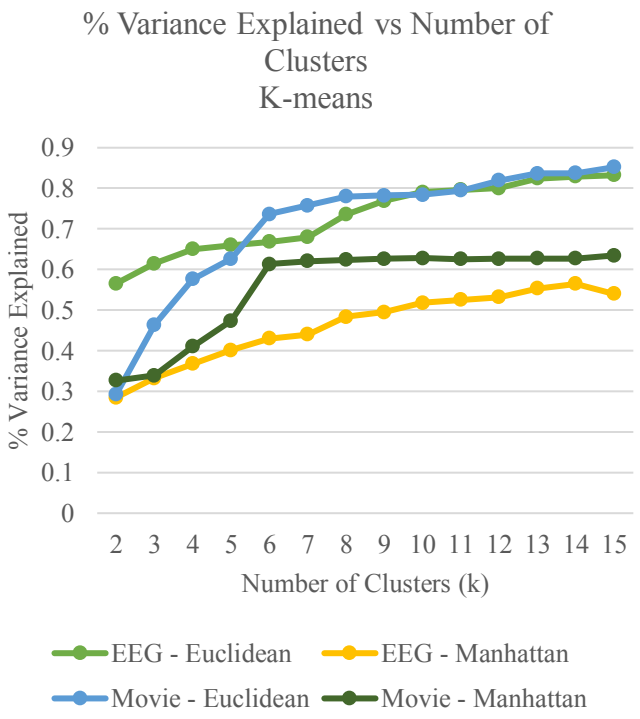

Figure 1 Variance Explained vs k for K-means

The optimal number of clusters for EEG was found to be 8. When doing the classes to cluster evaluation the error with Euclidean distance was 80.2904 %, while it was 78.5965% with Manhattan distance.

The optimal number of clusters for the *Movie* dataset was 8. The classes to cluster evaluation resulted in 68.8281% for Manhattan distance and 57.1287 % error for Euclidean distance. In contrast to EEG, the Manhattan distance metric performs worse for the Movie dataset. The % Variance Explained for both distance metrics with the Movie dataset reflects this as well. Therefore, Euclidean distance adds more information for the Movie dataset, while Manhattan distance adds more information with increasing clusters for the EEG dataset. Even though the class to clusters evaluation for Euclidean distance results in more error, K-means clustering with Euclidean distance explains 74% of the variance as opposed to 48% with Manhattan distance. This leads to the conclusion that Euclidean distance is better for both datasets.

Expectation Maximization

The output for the expectation maximization clustering algorithm is log likelihood. Log likelihood represents the probability that the points will remain in that cluster. The expectation maximization algorithm uses maximum likelihood to maximize this value.

For Expectation Maximization, I ran different 1-15 clusters for the EEG and the Movie dataset. By looking at the graphs and using the elbow method, it can be inferred that the optimal number of clusters for the EEG dataset is 6. The clusters to class evaluation for the EEG dataset results in 70.60% error. The optimal number of clusters for the Movie dataset is 8, and the clusters to class evaluation results in 54.7293% error. The EEG dataset has a log likelihood that remains negative for all values of k 1-15, and a log likelihood of -113.00 for the optimal number of clusters (k = 6).

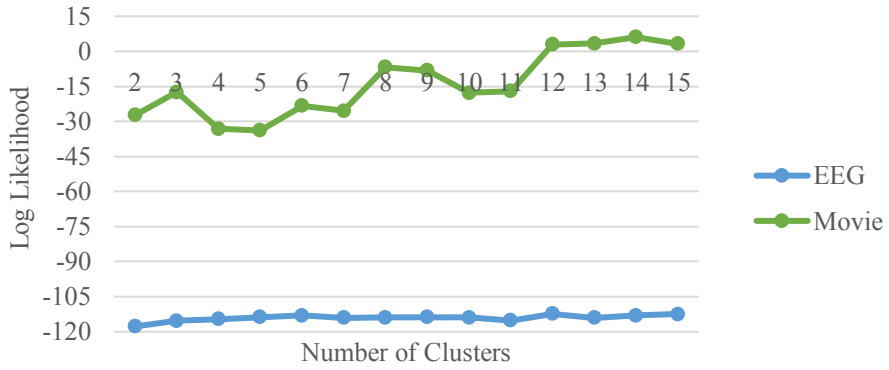

Figure 2 Log Likelihood vs. k for Expectation Maximization

On the other hand, the Movie dataset has a much lower log likelihood at -6.73 for the optimal number of clusters (k = 8). The log likelihood and the error indicate that expectation maximization performs better on the Movie dataset. Since the log likelihood remains negative in both, it would not be recommended for either dataset.

Dimensionality Reduction with Principal Component Analysis

Principal components analysis uses an orthogonal transformation that results in linear combinations of attributes called principal components. It is used to find principal components in a dataset of dependent variables, which, generally, are inter-correlated. PCA uses the eigenvalue decomposition of matrices. The output of the procedure are the linear combinations of original variables that are linearly uncorrelated. The combination with the highest eigenvalue has the highest variance, and so on.

Approach

For both datasets, PCA was run with 0.75, 0.8, 0.85, 0.9, and 0.95 variance covered. As the variance covered increased, the number of principal components increased as well. There was no effect on runtime as the variance covered increased. For the analysis, 0.9 variance covered were utilized for both datasets. After which, variance thresholding was used to filter principal components.

Eigenvalues Analysis

PCA initially reduced the dataset from 13 attributes to 9. The eigenvalues ranged from 4.62 to 0.45, as shown in the bar graph. Only the components with 15% of the variance of the component with the highest eigenvalue (4.62448) were preserved in the dataset. This further reduced the dataset to 6 components with the highest eigenvalues. The thresholds were selected based on the number of components PCA gave as an output.

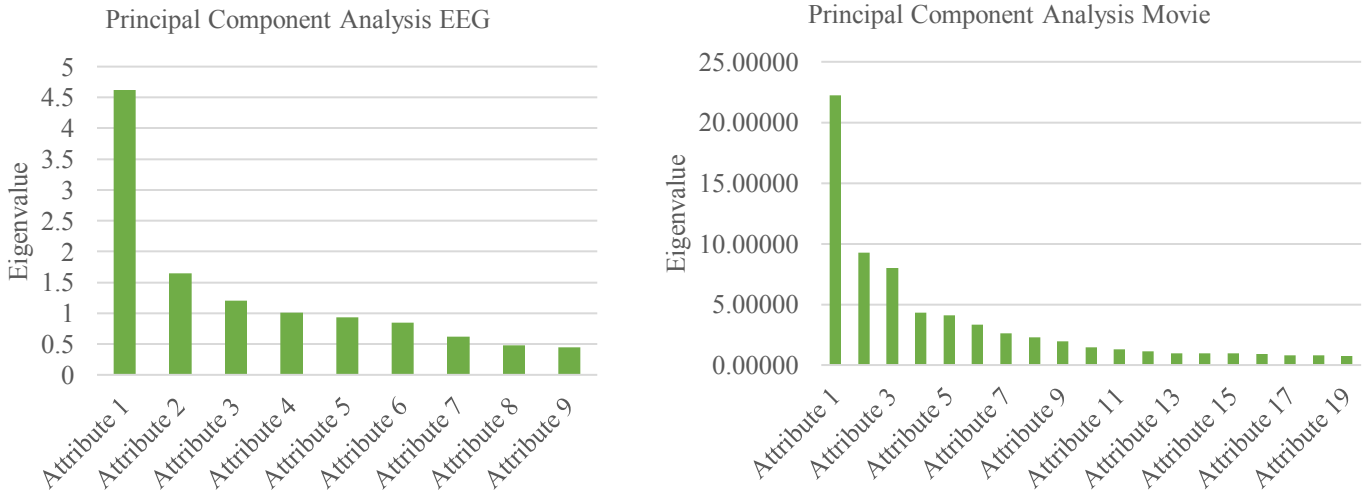

Figure 3 Eigenvalues for linear combinations output by PCA with 0.9 variance covered

The Movie dataset initially had 2 nominal variables, language and content rating. These variables in the dataset were converted to binary values before running any of the clustering and dimensionality reduction algorithms. For the movie dataset with the binary values, PCA with 90% variance covered reduced 75 attributes to 19

attributes. After obtaining the 19 attributes, only those with the 10% of the variance of the component with the highest eigenvalue (22.262) were preserved. This reduced the dataset from 19 components to 8 components.

Clustering after PCA

After PCA, the EEG dataset had a 0.18 increase in the % Variance Explained for the optimal k, 8 clusters, when k means was run on the Euclidean distance metric. For the Manhattan distance metric, there was a 0.15 increase in the % Variance Explained for the optimal k. For both metrics, PCA with 90% variance covered and 15 % thresholding resulted in better results for K-means clustering. PCA was useful in reducing the collinearity among the EEG attributes. For K-means with Euclidean distance, the higher numbers of instances were classified in Cluster 7 (26%) and Cluster 5 (23%), while Cluster 0 (2%) and Cluster 1 (2%) did not have a high number of instances. Most of the clusters had centroids where the class label was 1, which is consistent with the original dataset. These results were pretty similar for the Manhattan distance metric.

The Movie dataset showed a significant improvement after PCA as well. The change in K-means with Euclidean distance an increase of 0.18 and an increase of 0.15 with Manhattan distance for the optimal k, 6 clusters. PCA with 90% variance covered and 10% thresholding resulted in improvements for the Movie dataset as well. For the K-means with Euclidean distance, 54% of the instances were in Cluster 5, while Cluster 3 had the least, 1% of the instances.

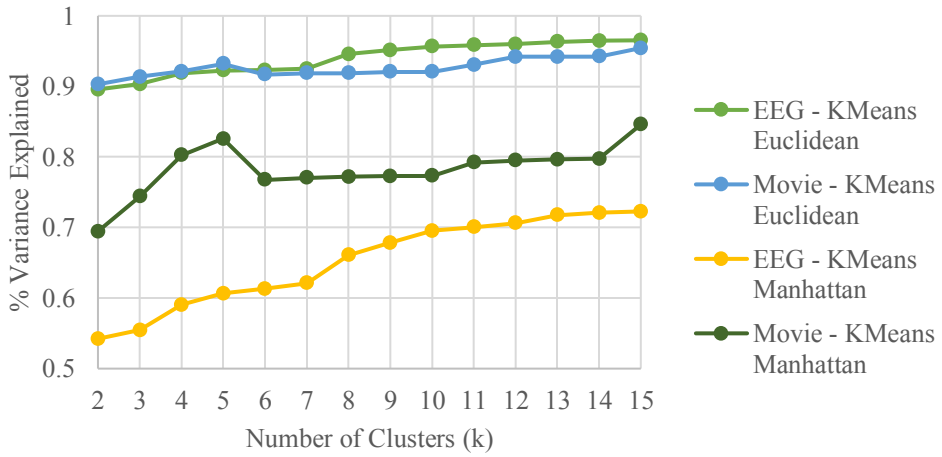

Figure 4 % Variance Explained vs k K-means after PCA

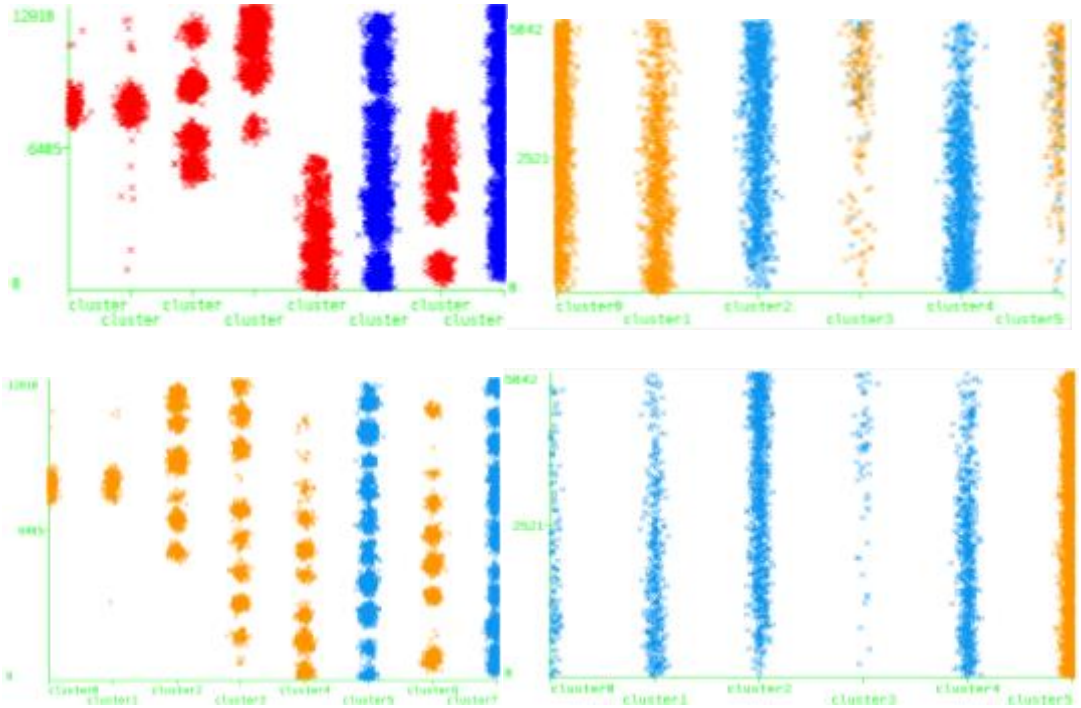

Figure 5 Kmeans EEG (top left) and Movie (top right) noPCA, EEG (bottom left) and Movie (bottom right) with PCA. Red=Orange=1, Light Blue = Dark Blue = 0. EEG (k=8), Movie (k=6)

Overall, reducing the dataset with PCA resulted in better performance for K-means. The cluster plot for the Movie dataset after PCA shows an interesting distribution, which differs from the original cluster plot. The cluster plot for the EEG dataset preserves the similar distribution as the original clusters.

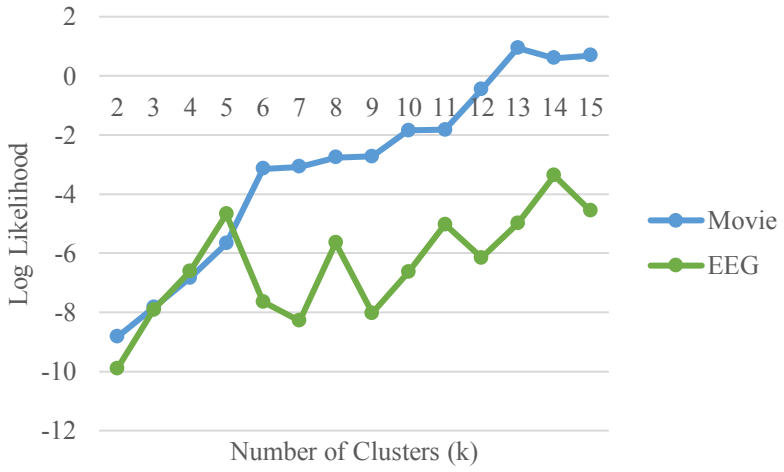

Figure 6 Log Likelihood vs k PCA Expectation Maximization

For the EEG dataset, the log likelihood in the original dataset for the optimal k, 6 clusters, was -113.00245, while the log likelihood after running PCA (0.9 variance covered, 15% threshold) on the dataset was -7.64152. PCA improved the log likelihood for expectation maximization, but the orthonormal transformations were not enough to result in good performance for EM. For the EEG dataset, most the clustered instances were in Cluster 1 (42%) and Cluster 4 (23%) with expectation maximization. The cluster plot for EEG after PCA has a different grouping from EM without PCA and better performance.

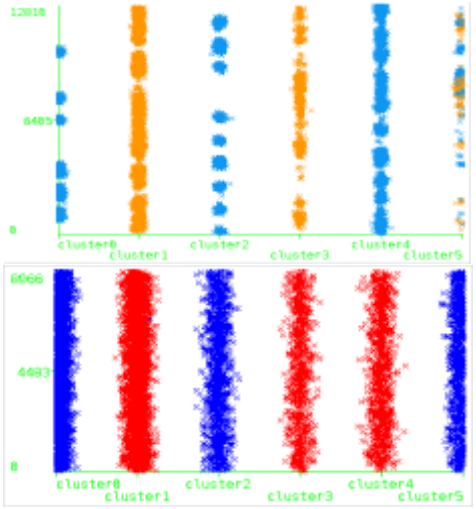

Figure 7 Cluster Plot for EEG EM after PCA (top), before PCA(bottom) Orange/Red = 1, Blue = 0

Moreover, the log likelihood for the Movie dataset did not improve significantly. For 8 clusters, the original dataset resulted in a log likelihood of -6.73362 and after running PCA (0.9 variance covered, 10% threshold), the log likelihood was -2.75717. EM would not be the best clustering algorithm for any of these datasets, given the low log likelihood values. The instances are mainly split across Cluster 4 (26%), Cluster 5 (29%), and Cluster 7 (29%). The cluster plot for the Movie data set did not show any interesting insights like EEG.

Dimensionality Reduction with Independent Component Analysis

Independent component analysis separates the variable into independent subcomponents. ICA assumes that these subcomponents are non-Gaussian. Like PCA, ICA results in a linear representation. After running ICA on the EEG and Movie datasets, the least kurtotic attributes were removed from the resulting dataset to ensure the non-Gaussian nature of the data, and the clustering algorithms were further run on the datasets.

Kurtosis Analysis

For the both datasets, the kurtosis of each resulting attribute was calculated in Excel. The values with low kurtosis were removed from the resulting dataset. As a result, ICA reduced the dataset from 13 attributes to 12 attributes with the highest kurtotic values. As the graph shows, these attributes have a high kurtosis values that are in the range of 291. This shows that the dataset with 12 attributes is highly non-Gaussian. For the Movie data, the kurtosis was within the range of 41 for most attributes, The 5 least kurtotic attributes were removed from the dataset. This reduced the dataset from 75 attributes to 70 attributes.

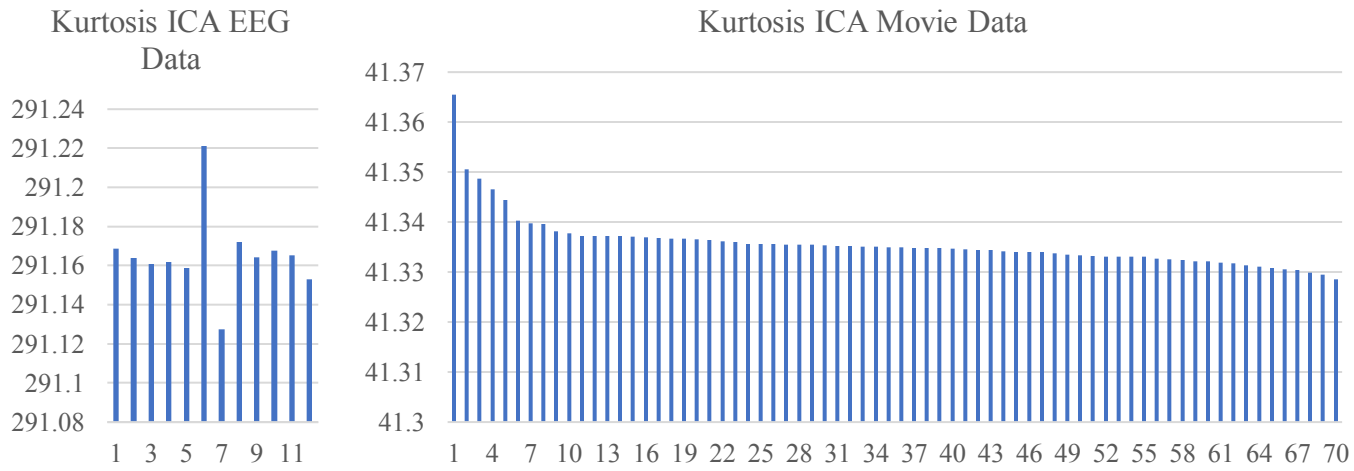

Figure 8 Kurtosis values for Attributes after ICA on EEG (left) and Movie (right)

Clustering After ICA

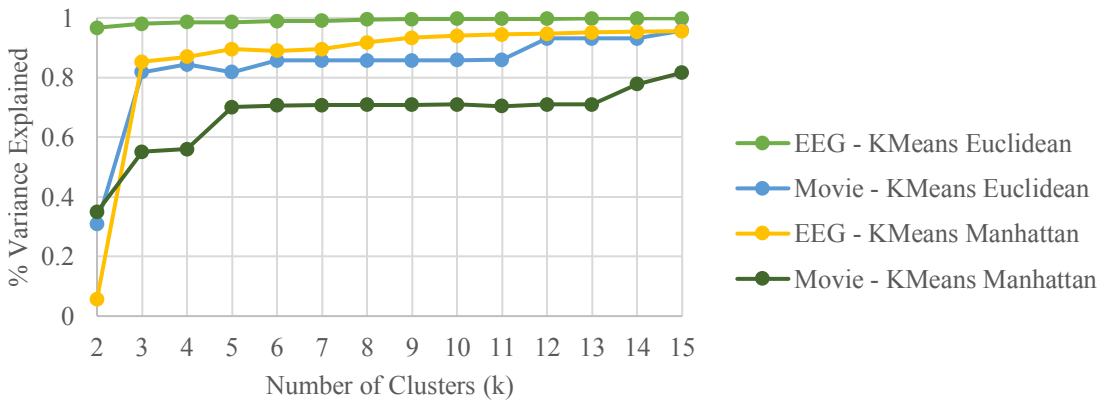

Figure 9 Variance Explained vs k ICA K-means

After reducing the datasets using ICA, k-means was run on EEG. ICA results in an increase in % variance explained of 0.26 for the Euclidean distance metric and 0.30 for the Manhattan distance metric. This is a much better performance improvement than PCA resulted in for the EEG dataset. The instances are mostly grouped in Cluster 4 (39%) and Cluster 5 (47%). All the clusters except Cluster 5 and Cluster 7 have a label of 1, whereas the other clusters have mostly label 2.

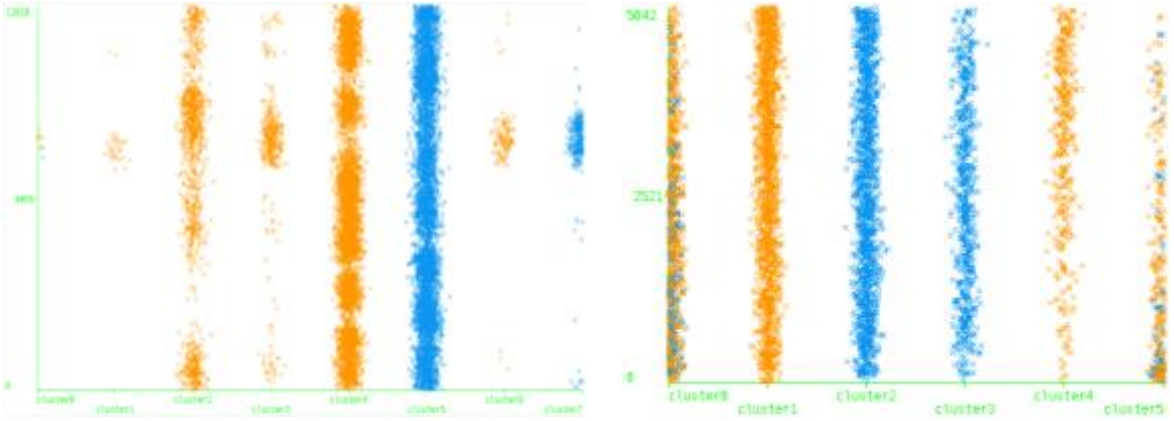

Figure 10 Cluster Plots for EEG (left) and Movie (right) K-means after ICA. Orange/Red = 1, Blue = 0.

The results are similar for k-means with Manhattan distance metric. The instances are mostly grouped in Cluster 5 (44%), Cluster 4 (22%), and Cluster 2 (14%), yielding a well grouped cluster plot. The cluster plot shows the clusters for the k-means algorithm with Euclidean distance. EEG shows a much reduced number of instances in Cluster 7 compared to original k-means without ICA.

For the Movie dataset, running k-means after ICA resulted in an increase of 0.15 in % variance explained for the Euclidean distance metric and 0.09 for the Manhattan distance metric. The instances are mostly grouped in Cluster

5 (51%) and Cluster 1 (40%) with k-means (k = 8) with Euclidean distance. For k-means (k = 8) with Manhattan distance, most instances are in Cluster 1 (30%) and Cluster 2 (28%). The cluster plot for the Movie dataset has a different label distribution compared to expectation maximization without ICA.

EEG has a much better performance improvement than the Movie dataset with k-means. The Movie dataset performs better with PCA than ICA dimensionality reduction for the k-means algorithm with Euclidean and Manhattan distance.

For expectation maximization after ICA, EEG resulted in a log likelihood of -38.39402, which is much lower than the original dataset log likelihood, -113.00245. The improvement is not as significant as PCA had on EEG. The instances are evenly distributed among Cluster 0, Cluster 2, Cluster 3, Cluster 4, and Cluster 5 with Cluster 1 having the least of the instances, 8%.

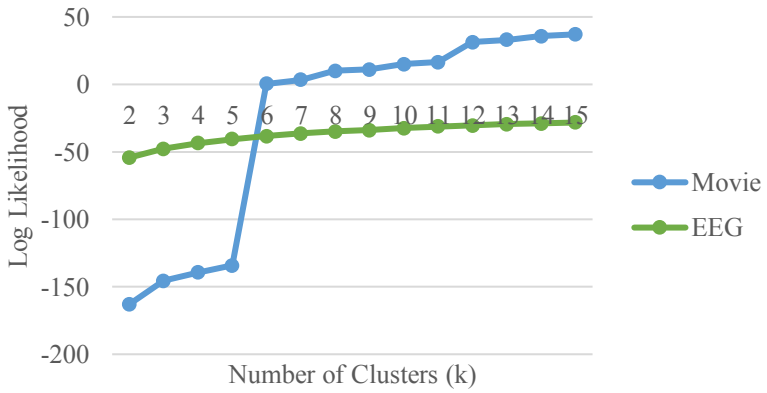

Figure 11 Log Likelihood vs k EM after ICA

For the Movie dataset, expectation maximization after ICA resulted in a log likelihood 10.10137, which is a positive improvement from -6.73362 for the original dataset. A majority 73% of the resulting instances of the Movie dataset are in Cluster 6. The higher log likelihood after running ICA on the dataset indicates that the output is likely for expectation maximization with 8 clusters. Expectation maximization performs well on the Movie dataset after ICA. However, it does not perform well on the EEG dataset after ICA.

Random Projection

In random projection, data with d dimensions is projected onto a subspace with k dimensions, where k is smaller than d. As a result, this results in a reduced- dimension dataset. The algorithm uses the Johnson-Lindenstrauss lemma to project data on a subspace that preserves the distances between the original points.

Clustering after Random Projection

Random projection was run for both datasets with 2 different values for the attributes, corresponding to the number of attributes resulting from ICA (EEG = 12, Movie = 70) and PCA (EEG = 6, Movie = 8). For EEG, having as many attributes as ICA performed better than as many as PCA, with 0.13 and 0.11 increases in % variance explained, respectively. With ~ICA attributes, the instances were mostly in Cluster 7 (37%) and the rest were spread out within 4 other clusters. For ~ PCA attributes, the instances were mostly in Cluster 7 (41%) as well, with a similar spread to ~ ICA.

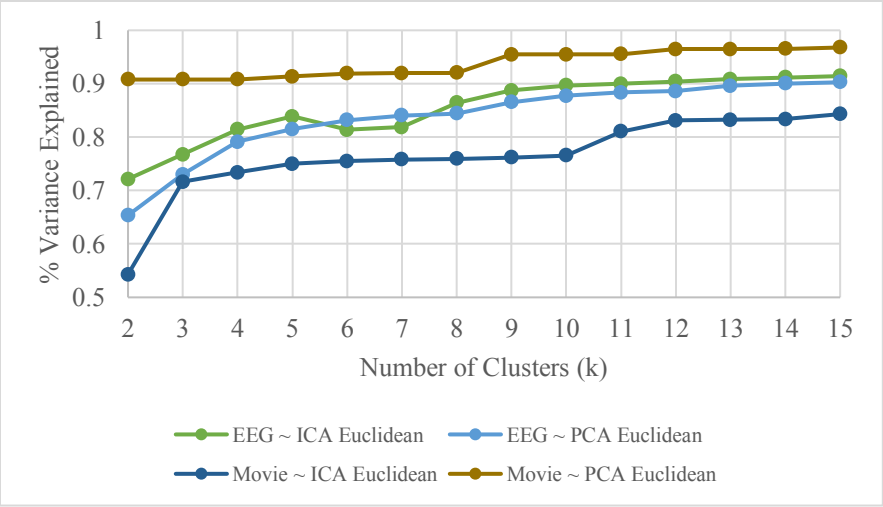

Figure 12 Variance Explained vs. k K-means after Random Projection

For Movie, in contrast to EEG, having as many attributes as PCA performed better than as many as ICA. Having as many attributes as ICA resulted in an low increase of 0.02 for % variance explained, whereas having as many attributes as PCA resulted in a significant increase of 0.18 for % variance explained. This can be attributed to the high number of components in ~ICA (70) and the low number of components in ~PCA (8). In ~ICA 47% of the instances are in Cluster 5, while in ~PCA 54% of the instances are in Cluster 5.

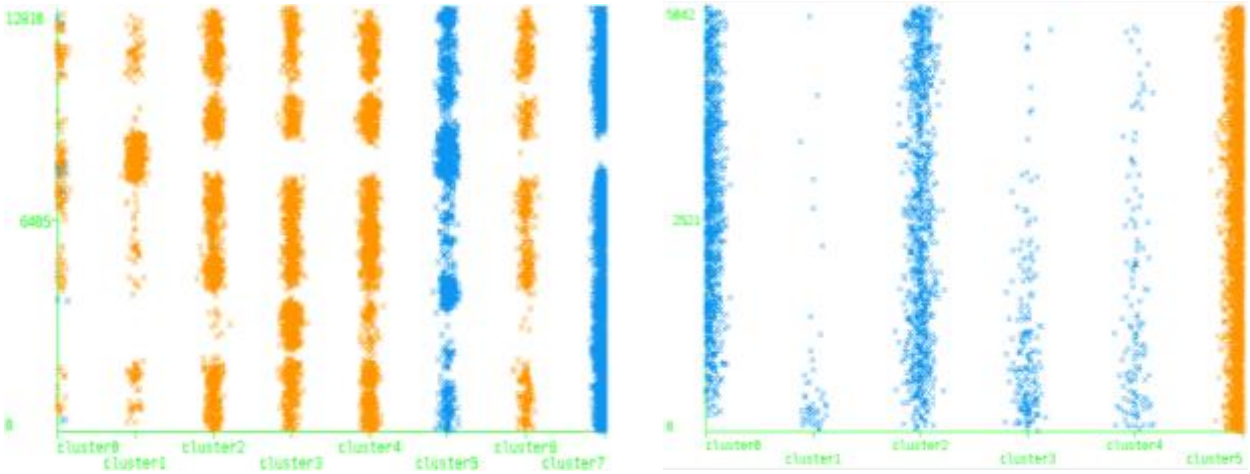

Figure 13 Cluster Plots for EEG (left) and Movie (right) for K-means after Random Projection

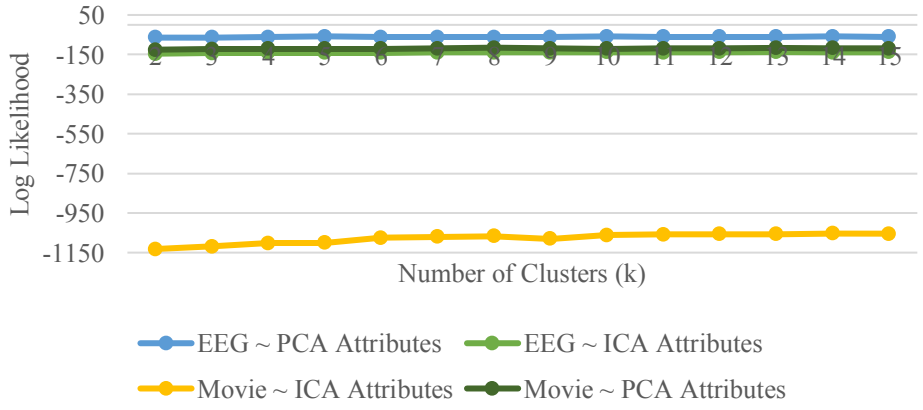

Figure 14 Log Likelihood vs k for EM after Random Projection

Random projection increased the log likelihood from -113.00245 to -61.82237 when the number of attributes was ~PCA. When the number of attributes was ~ICA, the log likelihood decreased to -141.18565. Performing random projection for EEG with ~PCA attributes resulted in a better value for log likelihood. For the Movie dataset, log likelihood for ~ ICA and ~ PCA was much lower than that of the original dataset with 8 clusters (-6.73362) at -1066.96942 and -116.69087, respectively. Having ~ICA attributes performs better than ~PCA, but not better than the original dataset. Overall, Random Projection would not be recommended to run EM on these datasets.

Dimensionality Reduction with Information Gain

Information gain is a feature selection algorithm used to select the most significant attributes in a dataset. It measures the number of bits of information obtained for prediction of the class by knowing the presence or absence of an attribute.

Clustering after Information Gain

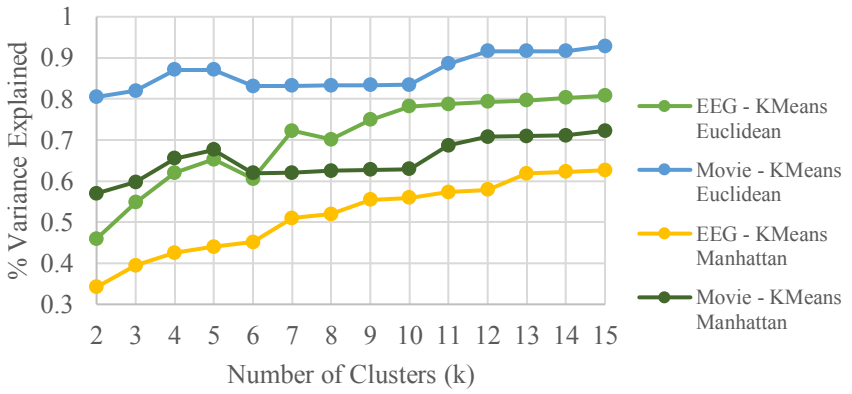

Figure 15 Variance Explained vs k K-means after Information Gain

For EEG, information gain with k-means resulted in a decrease of 0.03 and 0.10 for % variance explained with Euclidean and Manhattan distance metrics, respectively. Information gain performed worse than PCA, ICA, and the original dataset for EEG. The EEG dataset was run with ~70% of the attributes of the original dataset. These included the top 9 attributes ranked by information gain. 86% of the instances were concentrated in Clusters 5 and 6.

For Movie, information gain with k-means resulted in an increase of 0.09 and 0.006 for % variance explained with Euclidean and Manhattan distance metrics, respectively. Information gain did not perform better than PCA and ICA. However, it still performed better than the original dataset. The Movie dataset was run with ~13% of the attributes of the original dataset. These included the top 10 attributes ranked by information gain. With k-means in Euclidean distance, 54% of the instances were in Cluster 5.

Figure 16 Log Likelihood vs k EM after Information Gain

Running expectation maximization for EEG resulted in a log likelihood of -69.81847 with the optimal number of clusters (k = 8). For EEG, this is an improvement from -113.00245. It performs worse than ~ ICA for random projection, but worse than PCA and ICA.

For Movie, this resulted in a log likelihood of -91.87646 with the optimal number of clusters (k = 6). For Movie, this is a much worse performance than the original dataset which had a log likelihood of -6.73362. This can be attributed to the low number of attributes that was selected. Perhaps selecting 40-50% would yield better results for the Movie dataset. Expectation maximization after information gain performed better than random projection but not as well as PCA and ICA. Information gain does not take into consideration the correlation between attributes, whereas PCA and ICA do. The cluster plots for EEG show less concentration in cluster 7 compared to EEG without Information gain. For the Movie dataset, information gain results in a more even distribution of the instances than without information gain.

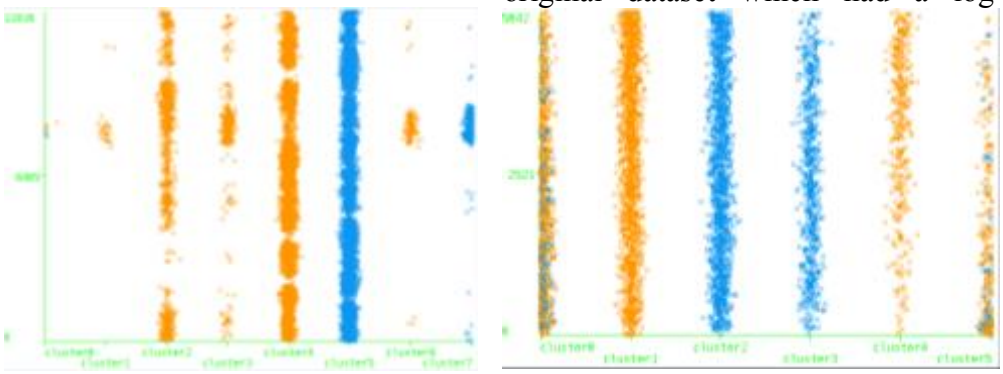

Figure 17 Cluster Plots for EEG (left) and Movie (right) after Information Gain

Neural Networks

Approach

The neural networks were run for the EEG dataset after dimensionality reduction and with clustering as a feature. The optimal hyperparameters found in Assignment#1 for this dataset were 0.1 for learning rate and 0.1 for momentum. The neural networks were run over 2500 epochs.

Results

Dataset	CV Err (%)	CV Acc (%)	Test Error(%)	Test Acc(%)	Training Time	Test Time
Information Gain	29.2327	70.7673	28.6755	71.3245	55.98	0.01
PCA	30.6924	69.3076	32.7609	67.2391	37.76	0.01
ICA	48.7784	51.2216	51.0539	48.9461	77.77	0.01
RP ~ PCA	39.0602	60.9398	40.0989	59.9011	35.97	0.01
RP ~ ICA	38.2718	61.7282	38.1993	61.8007	71.98	0.01
EM	17.2274	82.7726	16.2894	83.7106	120.82	0.02
KM - Manhattan	2.6306	97.3694	3.0445	96.9555	166.57	0.01
KM - Euclidean	2.2325	97.7675	0.7546	99.2454	196.47	0.03
Original	30.185	69.815	28.2852	71.7148	18.22	0.01

With dimensionality reduction, the best results were obtained after running Information Gain on the EEG dataset with an accuracy of 71.32%, which is comparable to the accuracy with the original dataset. This is attributed to the fact that 70% of the attributes were preserved after information gain. Moreover, it is important to notice that information gain results in a similar accuracy as running the original dataset, 71.71%. Information gain, despite lower number of attributes, has a higher runtime. In conjunction with the clustering results after random projection, random projection with ~ICA performs better than random project with ~PCA. For this dataset, having a larger subspace resulted in better results than a smaller subspace. The linear combinations from PCA and ICA did not result in useful information since they did not have better accuracy than the original. Overall, dimensionality reduction did not have a large improvement on the neural network results. From the dimension reduction scenarios, I would select information gain to run neural networks. In this case, the runtime was in the order of seconds, so runtime is not as crucial. If the EEG dataset was larger, it would be recommended to run the original dataset in neural networks.

Using clustering as a feature improves accuracy significantly compared to the original dataset. Clustering with k-means and expectation maximization adds useful features for this dataset. K-means resulted in an accuracy of 99.24%, outperforming expectation maximization with an accuracy of 83.71%. As expected, Euclidean distance performs better for EEG than Manhattan distance. The best accuracy rate corresponds to the neural network with k-means (Euclidean) as a feature, 99.24%. Despite the run time, the high testing accuracy will yield the best results. The training accuracies should be evaluated to ensure there is no overfitting.

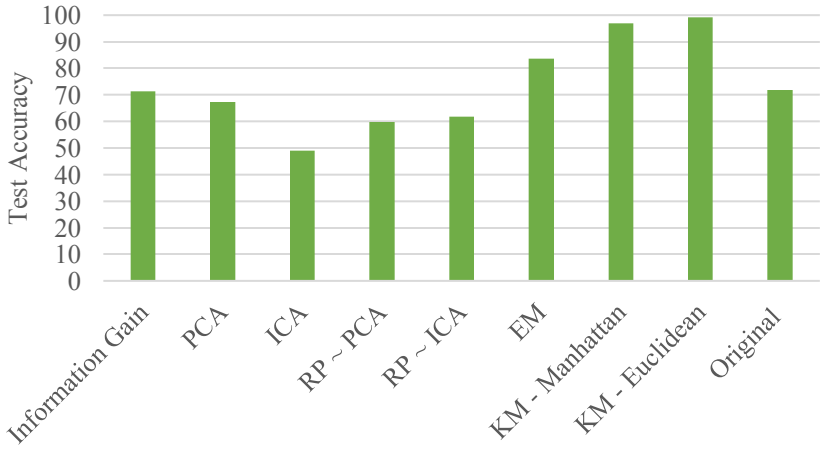

Figure 17 Neural Networks Test Accuracy