High Performance Clustering of Electroencephalogram (EEG) Data for Prediction of Seizure Events

Dustin C McAfee

Department of Electrical Engineering and Computer Science, UT, Knoxville, USA

Introduction

Given a dataset of Electroencephalogram (EEG) data, two clustering algorithms are implemented and compared as predictive models for seizure moments[1].

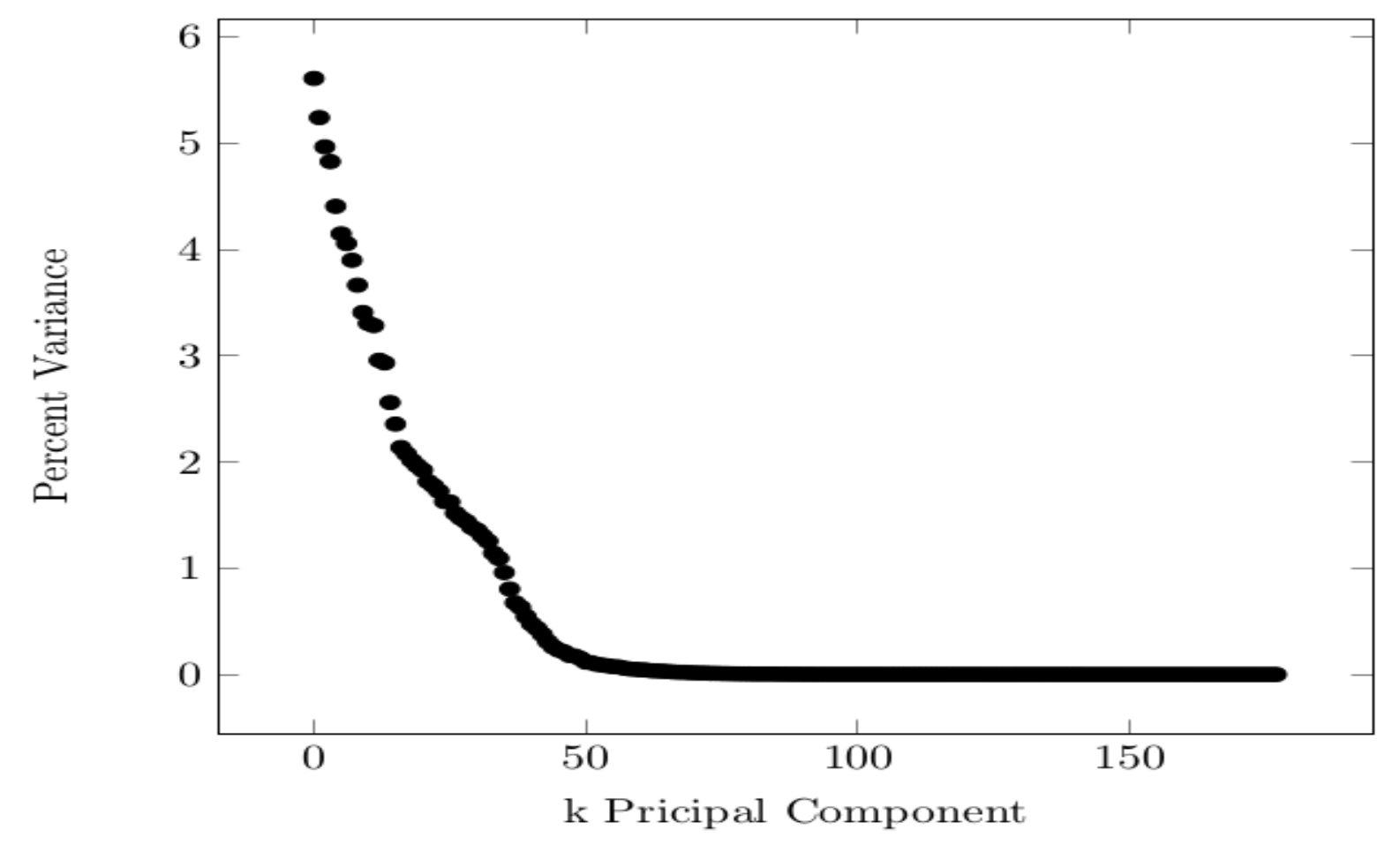
- The data has 178 dimensions, each dimension representing a second of information. This is reduced to the first 40 Principal Components (about 95.88% of the variance).
- There are 11500 observations, representing 500 individual 23 second EEG readings.
- The Testing Dataset represents about 23.5% (2700 observations) of the entire dataset, and the Training Dataset makes up the rest.

Expectation Maximization (EM) and KMeans++ is implemented in pyspark and compared

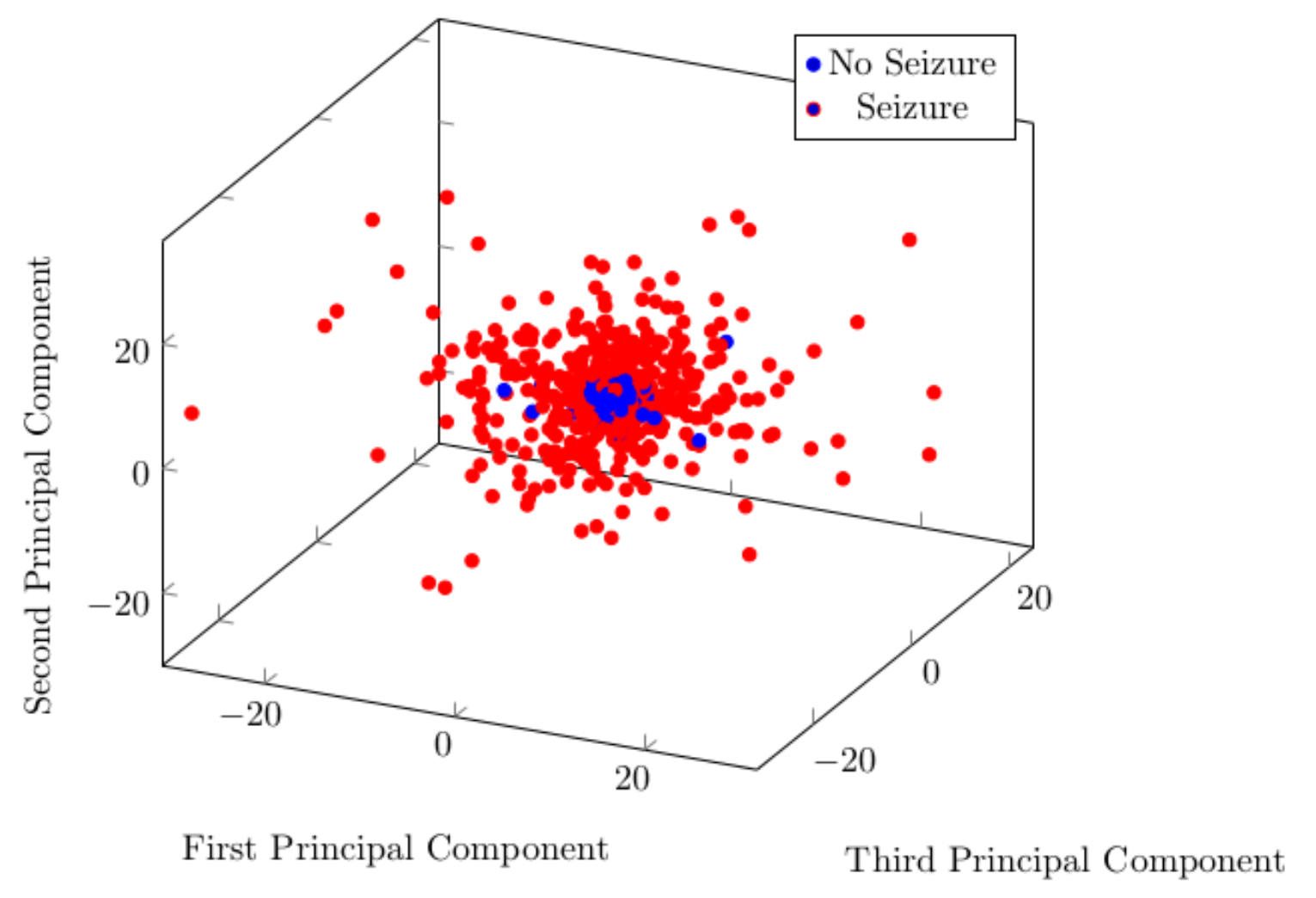
- Kmeans++ is implemented in both Kmeans and EM as an efficient way to pick the starting centroid values of each cluster[2].
- K = 2,3 clusters are considered, as to represent low/high risk categorizations and low/medium/high risk categorizations.

Statistical analysis reports

- Confusion Matrices and metrics such as Accuracy, Sensitivity, Precision, and Specificity are computed for each K and compared/contrasted across K values and algorithms.
- The dataset is projected onto its first three Principal Components and the testing dataset is plotted in a scatter plot against the predicted values for compare/contrast of the K values and algorithms.



Scree graph of EEG dataset



Dataset projected onto first 3 Principal Components

Methods

Environment Used:

- Python 3.4.3 on Ubuntu 14.04 with 16GB of RAM, 4-core (8 logical core) 2.8GHz 7th gen Intel I7
- 8GB of memory and 7 cores are reserved for execution in spark environment

Source Code:

- 'data.py' z-normalizes the dataset and splits the dataset into randomized training and testing dataset partitions.
- 'kmeans.py' takes number of clusters as an argument, trains on the training dataset, and predicts the testing dataset.
- 'exmax.py' takes number of clusters as an argument, trains on the training dataset, and predicts the testing dataset.
- Expectation Maximization assumes Gaussian Clusters.

Different Indices are used for clustering measurements

- The Dunn Index is used to measure how well the Kmeans algorithm seperated the data.
- The Silhouette Index is used to measure how well the Expectation Maximization algorithm seperated the data.
- The Silhouette Index is much more suited for measuring cluster seperation of non-hyperspheroidal clusters, such as Gaussian Distributions.

Tolerance and Convergence:

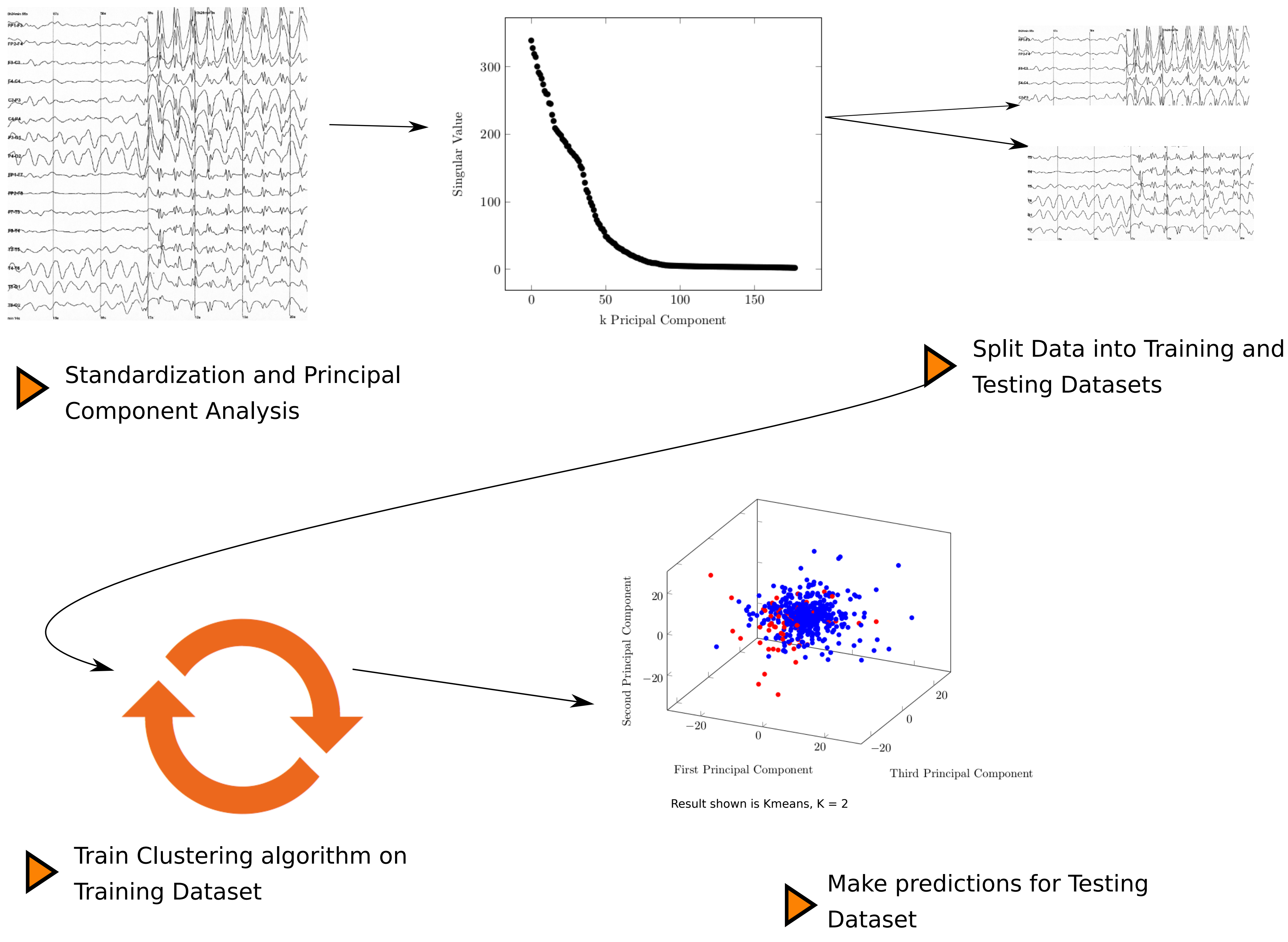
- For both Kmeans and EM, the tolerance is set to 0.005.
- For Kmeans, if the sum of the change of the centroids of each cluster is below the tolerance, then the algorithm converges.
- For EM, if the sum of the change of the centroids of each cluster or the sum of the change of log likelihoods is below the tolerance, then the algorithm converges.

References

[1] Andrzejak RG, Lehnertz K, Rieke C, Mormann F, David P, Elger CE (2001) Indications of nonlinear deterministic and finite dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state, Phys. Rev. E, 64, 061907

[2] Arthur, David & Vassilvitskii, Sergei. (2007). K-Means++: The Advantages of Careful Seeding. Proc. of the Annu. ACM-SIAM Symp. on Discrete Algorithms. 8. 1027-1035. 10.1145/1283383.1283494.

Process


Singular Value
k Pricipal Component
300
200
100
0
50
100
150
Second Principal Component
First Principal Component
Third Principal Component
20
0
−20

Standardization and Principal Component Analysis

Split Data into Training and Testing Datasets

Train Clustering algorithm on Training Dataset

Make predictions for Testing Dataset

Result shown is Kmeans, K = 2

Results

Clustering Metrics

- Kmeans Dunn index and number of iterations for convergence on training dataset

	k=2	k=3
Min Inter	9.94	0.74
Max Intra	92.77	91.71
Dunn	0.0963	0.0080
# Iterations	20	27

- EM Silhouette index and number of iterations for convergence on training dataset

k	k=2	k=3
Silhouette	0.3789	0.0580
# Iterations	13	14

KMeans, K = 2

3D Scatter Plot of Predictions for Testing Dataset shown above

Prediction outcome

Actual value

	n	p	Total
n′	6829	238	7067
p′	72	1661	1733
Total	6901	1899	

- Confusion Matrix for Training Dataset, 82.61% Accuracy, 11.89% Sensitivity, 98.56% Precision, 99.96% Specificity

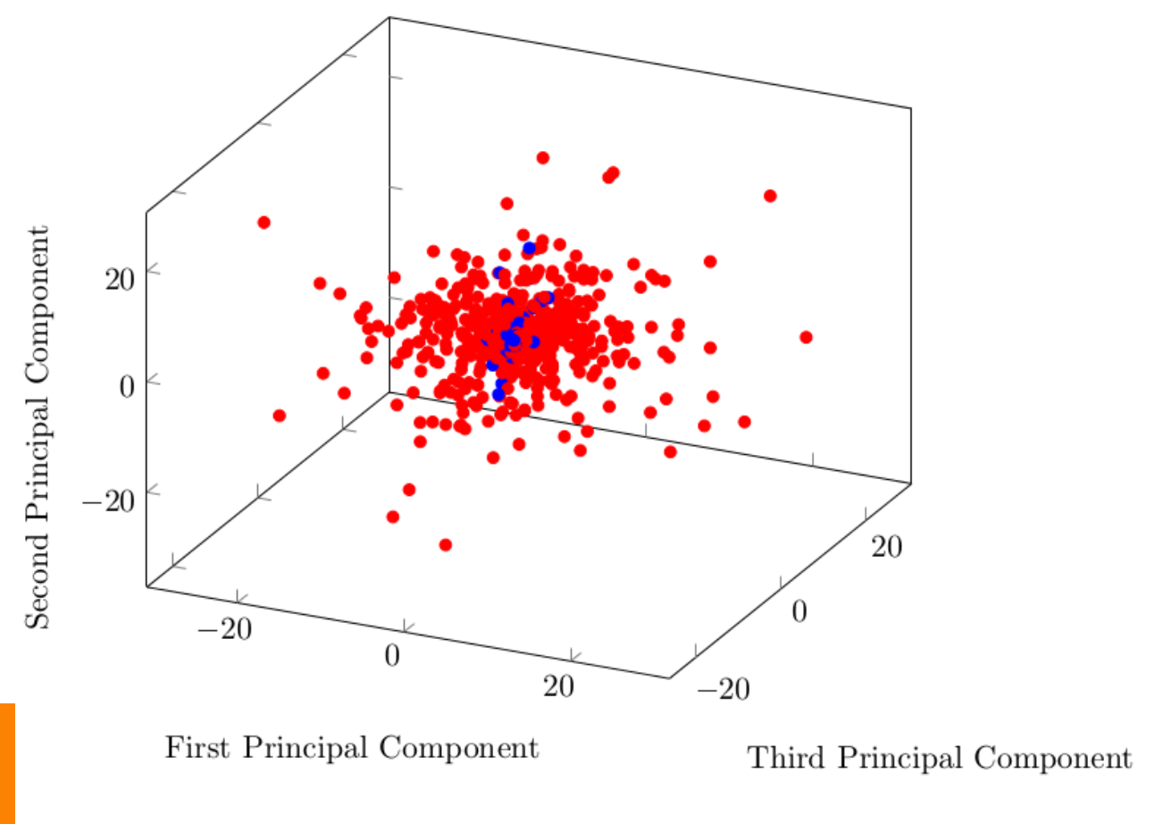
Prediction outcome

Actual value

	n	p	Total
n′	1895	239	2134
p′	107	458	565
Total	2002	697	

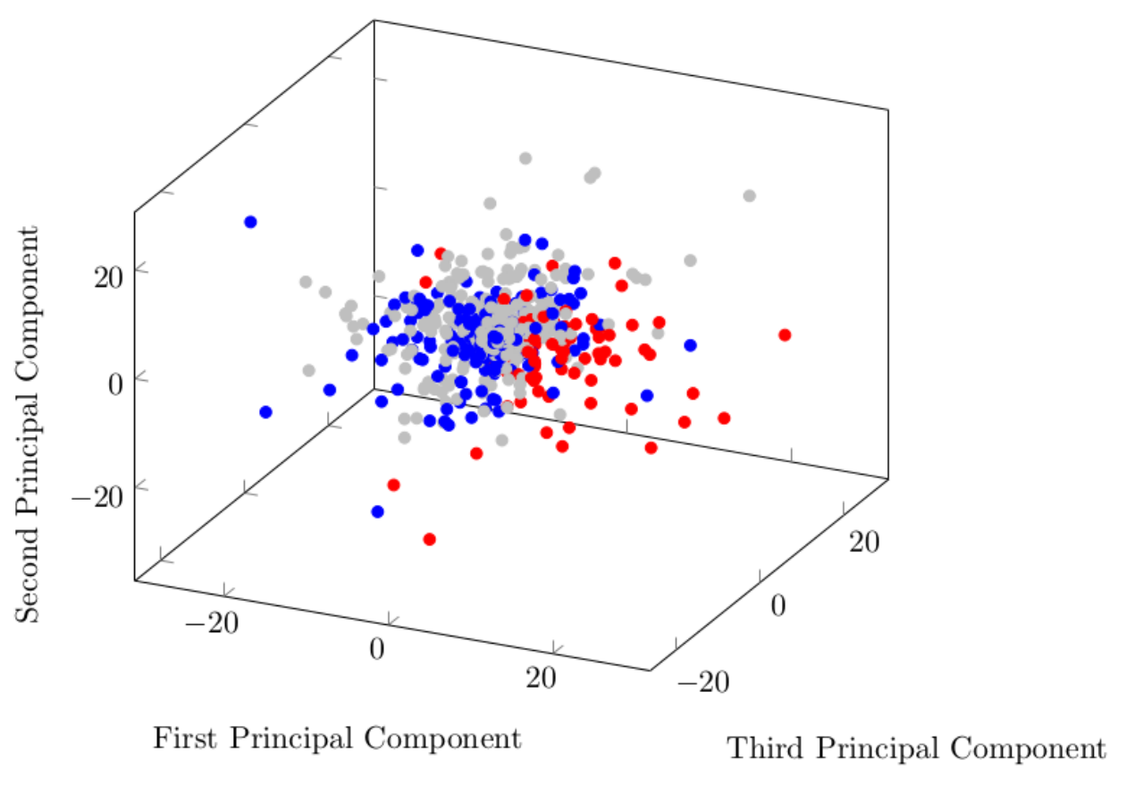
- Confusion Matrix for Testing Dataset, 81.44% Accuracy, 11.66% Sensitivity, 98.51% Precision, 99.95% Specificity

EM, K = 3



- Confusion Matrix for Training Dataset, 67.32% Accuracy, 100% Sensitivity, 37.60% Precision, 59.30% Specificity
- Confusion Matrix for Testing Dataset, 66.25% Accuracy, 92.23% Sensitivity, 37.58% Precision, 59.35% Specificity

KMeans, K = 3



Prediction outcome

Actual value

	n	p	Total
n′	4191	2876	7067
p′	0	1733	1733
Total	4191	4609	

- Confusion Matrix for Training Dataset, 30.40% Accuracy, 14.54% Sensitivity, 5.15% Precision, 34.29% Specificity

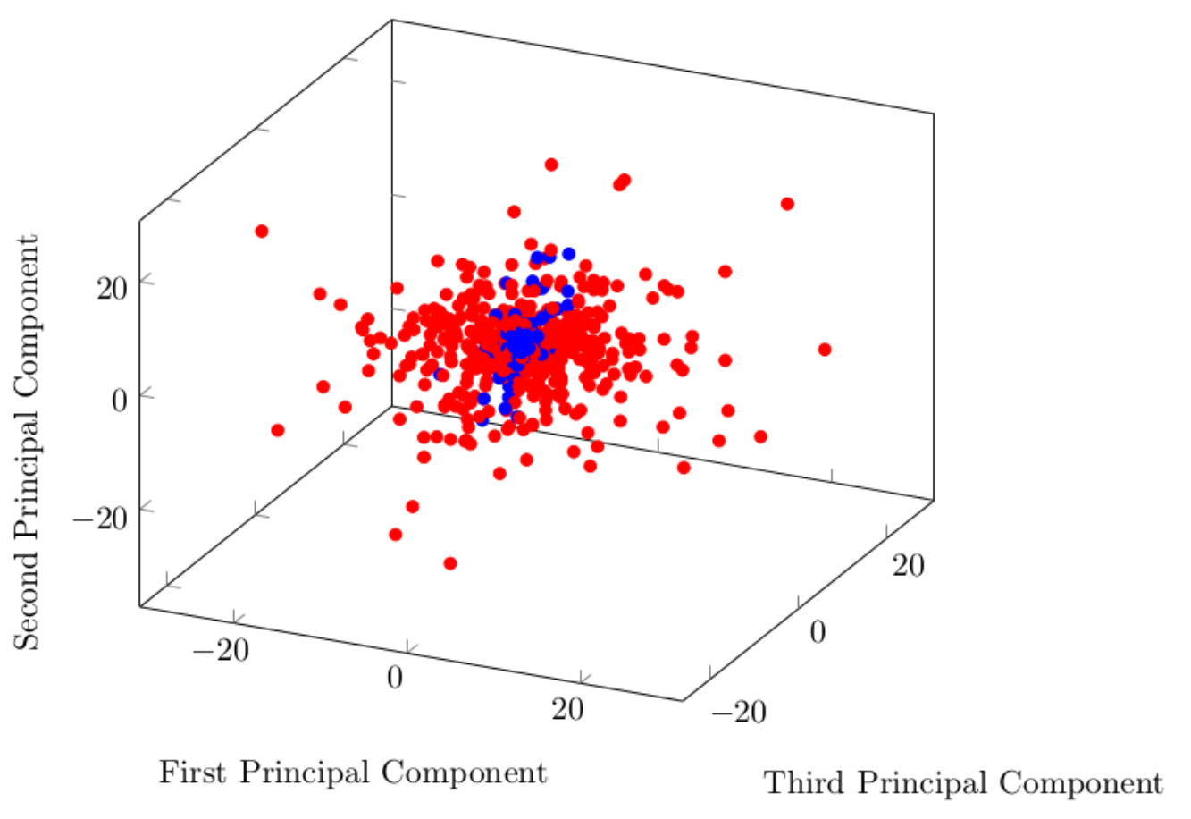
Prediction outcome

Actual value

	n	p	Total
n′	1266	868	2134
p′	44	521	565
Total	1310	1389	

- Confusion Matrix for Testing Dataset, 28.79% Accuracy, 13.45% Sensitivity, 5.04% Precision, 32.85% Specificity

EM, K = 2



Prediction outcome

Actual value

	n	p	Total
n′	6829	238	7067
p′	72	1661	1733
Total	6901	1899	

- Confusion Matrix for Training Dataset, 96.48% Accuracy, 95.85% Sensitivity, 87.47% Precision, 96.63% Specificity

Prediction outcome

Actual value

	n	p	Total
n′	1895	239	2134
p′	107	458	565
Total	2002	697	

- Confusion Matrix for Testing Dataset, 87.22% Accuracy, 81.10% Sensitivity, 65.85% Precision, 88.84% Specificity

Discussion/Future Goals

Implement parallelized Spectral Clustering Algorithms

- From looking at the actual grouping in the 3D scatter plot, it looks like the center sphereoid may represent no seizure activity, and the outside may represent seizure activity.
- Spectral Clustering Algorithms are better adept in clustering layered hyper-spheroidal clusters.

Clustering with the EM algorithm K = 3 yields two clusters with higher sensitivity and less accuracy. This is due to the maximum likelihood of the third class being least on all accounts.

More Cross Validation is needed

- The algorithms are trained and tested, once per K. There should be more variations of the training and validation datasets for proper comparison of accuracy metrics.
- K-fold Cross-Validation is a good validation technique that is easily parallelizable.