Mining Brain Network of Alzheimer's Disease Patients

By
A'aeshah Alhakamy, Kumud Bhat, Khan Al Farabi, Jiachen Wang

Department of Computer and Information Science
Indiana University – Purdue University Indianapolis (IUPUI)

Data Mining
Prof. Yuni Xia

December 2015

Abstract

Efficient diagnosis of Alzheimer's disease (AD), the most regular kind of dementia in elderly patients, is of essential significance in medical research. Late studies have exhibited that AD is firmly identified with the structure change of the brain network, i.e., the connectivity among different brain regions. The brain region of interest (ROIs) and neural connections among them will provide useful pattern-based biomarkers to recognize healthy control (HC), patients with Mild Cognitive Impairment (MCI), and patients with AD. In this paper, we investigate multiple machine learning algorithms for identifying the connectivity among different brain regions. Our results demonstrate that PCA, LDA, K-mean, DBSCAN and SVM algorithms are valuable in revealing the brain region connectivity similarity and differences among these groups which could help diagnosing Alzheimer's disease.

Keywords
Data mining, Alzheimer's disease, k-Means, Density-based clustering (DBSCAN), Principle component analysis (PCA), Linear Discriminant Analysis (LDA), Support Vector Machine (SVM)

INTRODUCTION

Decreases in cognitive and motor capacities, together with numerous confirmations of neurological degeneration, turn out to be progressively likely as healthy people age. The truth of the matter is that everybody will experience altered brain functions, albeit some at a prior age or at a quicker rate than others. In that capacity, recognizing the motor and cognitive delays of normal aging from those because of pathological procedures and comprehension the individualized disease diagnostic and prognostic patterns are progressing research challenges. In this report, Alzheimer's disease (AD), a plainly understood neurodegenerative illness bringing about cognitive impairment, is amongst the best concentrated on ailments of the focal nervous system because of its staggering impact on patients and their family, and to the financial effect of advanced social orders. In spite of that, it stays incurable.

Consistently a vast number of new Alzheimer's disease (AD) cases are diagnosed. The outcome is dementia in elderly people; internment and costly medical care are a typical result. An almost demonstrative and visualization can enhance the patient personal satisfaction, minimizing the requirement for internment and expensive medical care, decreasing the patient and the family's agony and reducing the social, financial consequences for the general public. In this setting, seeing whether and when a patient will advance from Mild Cognitive Impairment (MCI) to AD is of a noteworthy essential to the timely administration of pharmaceutics and therapeutic interventions. Moreover, it can permit medical specialists to improve periodicity of medical consults.

Mild Cognitive Impairment is right now thought to be an early phase of a Neurodegenerative Disease, especially AD. Patients diagnosed to have MCI are respected with uncommon consideration since they are accepted to have a higher risk to develop to dementia as usually AD. Under these suspicions, the factual finding of MCI conditions and a possible evaluation of its prescient worth for the transformation to AD are along these lines of major significance. Still and all, the meaning of MCI and its analysis criteria are not yet consensual; the pathologic and molecular substrate of individuals diagnosed to have MCI is not entrenched. Furthermore, individuals considered to be experiencing pre-MCI that is individuals having

cognitive complaints yet not satisfying the criteria for MCI, have as of late been appeared to have a high risk of progression to MCI and AD. This makes the diagnosis of MCI a tough task in itself and subsequently changes the expectation of MCI to AD transformations into a significantly more entangled task.

Neuropsychological tests have been utilized by medical specialists for the most part because they are less expensive and speedier than PET Scans and biomarkers look. Besides, innovation, for example, PET Scans and biomarkers are not all around accessible. The neuropsychological tests include straightforward Tasks, for instance, those concerning orientation, memory, attention and language to assess the mental condition of the patient.

This paper intends to utilize this data to anticipate the transformation of HC (Healthy People) to MCI to AD. The utilization of data mining Algorithms will permit to the extraction of knowledge or guidelines from the data in what respects the expectation of HC to MCI to AD. In this work, we handle the issue of distinguishing healthy people from patients with MCI, from those effectively experiencing AD, utilizing neuropsychological data. To achieve this goal we use state of the art machine learning techniques. Moreover, to find difference and similarity between each group we visualized our dataset first using VTK. Then, we used GNU Octave and MATLAB to implement K-means and DBSCAN for clustering, PCA and LDA algorithms to reduce the data dimensionality which have a suitable environment for performing data mining and visualization aspects, for classification purpose we implement SVM which provide a model to predict if a patient have MCI or AD.

BACKGROUND

It has been widely known that human brain features a very complex neural network. There are approximately 1010 neurons linked by 1014 synaptic connections. However, single neuron does not have a significant role until connections are made between them. The dynamic interactions among neurons support information processing and cognitive behaviors. The availability of modern brain mapping approaches, such as MRI and DIT, makes it possible to collect large-scale neuroanatomical connectivity data. Given the complexity of brain network, how to map and analyze brain network and thus to further link brain structure to its function is a significant computational challenge we are facing today.

2.1 Alzheimer Disease (AD)

Alzheimer's disease (AD) is a dynamically neurodegenerative disease. It is the most widely recognized kind of dementia in elderly patients. Presently, around 5 million individuals (around 10% of the populace more than 60) in the U.S. are burdened by AD. The directly assessed cost to care the patients is over $100 billion every year. As the population ages throughout the following decades, the AD cases and the related expenses are relied upon to go up significantly. AD researchers have in this manner heightened their endeavors to investigate approaches to defer, cure, or prevent the onset and progression of AD.
Alzheimer's sickness is an irreversible, progressive brain disease that gradually obliterates memory and

2.2 Mild Cognitive Impaired (MCI)

Mild cognitive impairment (MCI) is a transitional stage between the expected cognitive decay of ordinary aging and the most genuine decrease of dementia. It can include issues with memory, language, speculation and judgment that are further prominent than ordinary age-related changes. On the off chance that you have cognitive impairment, you may know that your memory or mental capacity has "slipped." Your family and dear companions likewise may see a change. Be that as it may, these progressions aren't sufficiently serious to interfere with your everyday life and common activities.
Mild cognitive impairment could rise your risk of later progressing to dementia, caused by Alzheimer's disease or other neurological circumstances. But some people with mild cognitive impairment at no time get worse, and a few ultimately get better.

METHOD

3.1 Data Acquisition and Preprocessing

The original datasets were provided by radiology department, IU School of Medicine. There are three files of original data: AD (Alzheimer’s disease patients); MCI (mild cognitive impaired); HC (healthy people).

The data were collected from a total of 104 subjects: 43 for HC, 42 for MCI and 19 for AD. For each subject, 234 ROIs were segmented. For each person, the brain network data is presented in 234 X 234 matrix. The brain is divided into 234 regions of interests (ROI), and each number shows the weighted connection between two ROIs. For example, [2] [3], row 2 and column 3, tells how strong the connection between ROI2 and ROI 3 is. Some are 0, which means there is no talk between those two regions. For example, in AD file, there are 4446 X 234, that's because we listed the matrices from all the 19 patients together, one by one. Each network is 234 X 234, and 234 X 19 = 4446, so total is 4446 X 234. By extracting the data, we mean extracting certain parts of the matrix. These features show the biggest differences among AD, MCI, and HC.

For each subject, the connectome edges and their weights among ROIs were presented by a matrix.
First of all, we would like to use visualization tools to visualize the training data and then to extract the features, which can better present the difference between patients and healthy people. We are currently working on this step, and we've seen some features that we are interested in. We are working on extracting the features. Secondly, we will further study the features with some date mining algorithms. We will test the efficiency of our model through analyzing testing data to compare AD patients and healthy control.

3.2 Data Visualization and Feature Extraction

Datasets and visualization tools: The program was implemented with c++ using visualization toolkit (VTK). VTK is widely used for 3D computer graphics, image processing and visualization. In this study, brain connectome data were collected for 104 subjects in 3 categories: 43 subjects for HC; 42 subjects for MCI and 19 subject for AD. For each subject, 234 ROIs were segmented and registered. Edges and corresponding weights were also computed. For each subject, the weighted network can be presented as a matrix, with rows and columns correspond to nodes and elements of the matrix stand for weights. The data was read in all together as StructuredPoints to create a volume object. Each layer corresponds to one subject matrix.

Volume rendering: We applied volume rendering techniques to study our volume datasets. These functions create a semi-transparent cloud of the datasets and allows all layers of surfaces to be observed at once. Two major transfer functions, vtkPiecewiseFunction and vtkColorTransferFunction, were used for opacity transfer and color transfer respectively. These two functions allow us to define color and opacity of every voxel. Then these two functions both went into vtkVolumeProperty which represented common properties for rendering. Later we applied vtkVolumeMapper to provide a mapper for rendering volume. Both vtkVolumeMapper and vtkVolumeProperty went into vtkVolume which was used to represent the volume in a rendering scene. Because our original brain network data is very sparse, we applied vtkGaussianSplatter function to "splat" each point to nearby voxels. Gaussian distribution function allowed each voxel to form a "snow ball". In this way, each individual voxel had a better chance to connect with nearby voxels to form a volume.

Feature Extraction: The part was implemented with QT. To extract the features, we integrated VTK into QT through QVTKWidget. The features we extracted are small matrices cropped from the original matrices. QT allows us to interactively define the parts we are interested in. According to the coordinates of the points we defined, we were able to extract the features.

3.3 Dimensionality Reduction Algorithms

Like clustering techniques, dimensionality reduction look for the characteristic structure in the data, however for this situation in an unsupervised way or request to compress or depict data utilizing less information. This can be valuable to visualize dimensional data or to simplify data which can then be utilized as a part of a supervised learning technique.
In our project we use PCA and LDA, via PCA, we are projecting the entire set of data (without class labels) onto a different subspace, and in LDA, we are trying to determine a suitable subspace to distinguish between patterns that belong to different classes. Or, roughly speaking in PCA we are trying to find the axes with

maximum variances where the data is most spread (within a class, since PCA treats the whole data set as one class), and in LDA we are additionally maximizing the spread between classes.

3.3.1 Principal Component Analysis (PCA)

PCA is a method for distinguishing patterns in data and communicating the data so as to highlight their similitude and contrast. Since patterns in data can be elusive in data of high dimension, where the advantage of the graphical representation is not accessible, PCA is an effective tool for analyzing data. The other fundament point of preference of PCA is that once you have discovered these patterns in data, and you compress the data, i.e., By reducing the number of dimensions, without much loss of data. This procedure utilized as a part of image compression also. Here are the main steps to accomplish this principle:

1. Take the whole dataset consisting of d-dimensional samples ignoring the class labels

```
fid = fopen('C:\Users\khan\Desktop\ADDDDDD\MCI_2.csv', 'r');
data=csvread(fid);
y = data(:,1);
X = data(:,2:end);
```

2. Compute the d-dimensional mean vector (i.e., the means for every dimension of the whole dataset)

```
mu = mean(X);
Xm = bsxfun(@minus, X ,mu);
```

3. Compute the scatter matrix (alternatively, the covariance matrix) of the whole data set

```
C = cov(Xm);
```

4. Compute eigenvectors ($e_1,e_2,...,e_d$) and corresponding eigenvalues ($\lambda_1,\lambda_2,...,\lambda_d$)

```
[W_pca,D] = eig(C);
  [D, i] = sort(diag(D), 'descend');
```

5. Sort the eigenvectors by decreasing eigenvalues and choose k eigenvectors with the largest eigenvalues to form a $d \times k$ dimensional matrix W(where every column represents an eigenvector)

```
  [D, i] = sort(diag(D), 'descend');
```

6. Use this d×k eigenvector matrix to transform the samples onto the new subspace. This can be summarized by the mathematical equation: y=WT×x (where x is a d×1-dimensional vector representing one sample, and y is the transformed k×1-dimensional sample in the new subspace.)

```
Xproj1 = project(Xm, W_lda(:,1:2));
coverlda=cumsum(W_lda) / sum(W_lda);
adald1=Xproj1(find(y==0.0),:);
adald2=Xproj1(find(y>0.0),:);
```

For the first features, for all three data sets (AD,HC,MCI)the original dimension is 520x51 matrix . After applying PCA it is reduced to 51x 51.

For the second features, for all three data sets (AD,HC,MCI)the original dimension is 190x17 matrix . After applying PCA it is reduced to 17x 17.

3.3.2 Linear Discriminant Analysis (LDA)

Practically, LDA for dimensionality reduction would be just another preprocessing step for a general machine learning or pattern classification task.

Listed below are the 5 general steps for performing a linear discriminant analysis; we will explore them in more detail in the following sections.

1. Compute the *d*-dimensional mean vectors for the different classes from the dataset.

```
dimension = columns(X);
```

2. Compute the scatter matrices (between-class and within-class scatter matrix).

```
Sw = zeros(dimension,dimension);
Sb = zeros(dimension,dimension);
```

3. Compute the eigenvectors ($\mathbf{e_1, e_2, ..., e_d}$) and corresponding eigenvalues ($\boldsymbol{\lambda_1, \lambda_2, ..., \lambda_d}$) for the scatter matrices.

```
[W_lda, D] = eig(Sw\Sb);
```

4. Sort the eigenvectors by decreasing eigenvalues and choose **k** eigenvectors with the largest eigenvalues to form a $d \times k$-dimensional matrix **W** (where every column represents an eigenvector).

```
[D, i] = sort(diag(D), 'descend');
```

5. Use this $d \times k$ eigenvector matrix to transform the samples onto the new subspace. This can be summarized by the equation $\boldsymbol{Y} = \boldsymbol{X} \times \boldsymbol{W}$ (where $\boldsymbol{X}$ is an $n \times d$-dimensional matrix; the ith row represents the ith sample, and $\boldsymbol{Y}$ is the transformed $n \times k$-dimensional matrix with the n samples projected into the new subspace).

```
proj1 = project(Xm, W_lda(:,1:2));
coverlda=cumsum(W_lda) / sum(W_lda);
%for i=1:520
adald1=Xproj1(find(y==0.0),:);
adald2=Xproj1(find(y>0.0),:);
```

For the First feature Y is 520x1 and for the second feature Y is 190x1 in LDA.

3.4 Clustering Algorithms

Clustering strategies are regularly composed by the modelling methodologies, for example, centroid-based and hierarchal. All systems are concerned with utilizing the innate structures as a part of the data to best arrange the data into gatherings of most extreme shared feature. For our project we used k-means and DBSCAN and we also use 19 subjects of each group to apply those algorithms on the datasets.
For both k-means and DBSCAN, in the first feature the input matrix of AD, HC, MCI is 520x52 and for the second feature is 190x18.

3.4.1 k-Means

K-means is one of the least difficult unsupervised learning algorithms that take care of the well-known clustering problem. The methodology takes a fundamental and simple approach to group a given information set to a certain number of clusters. Here are the main steps to accomplish this algorithm:
1. Randomly select '*c*' cluster centers.

```
centroid = rand(nbCluster, data_dim);
centroid = centroid .* repmat(data_diff, nbCluster, 1) + repmat(data_min, nbCluster, 1);
```

2. Calculate the distance between each data point and cluster centers.

```
dists = [dists d];
```

3. Assign the data point to the cluster center whose distance from the cluster center is a minimum of all the cluster centers.

```
[a, assignment] = min(dists');
 assignment = assignment';
```

4. Recalculate the new cluster center using: $v' = \left(\frac{1}{ci}\right) \sum_{j=1}^{ci} xi$
where, 'c_i' represents the number of data points in i^{th} cluster.

```
centroid(c,:) = sum(d,1);
 pointsInCluster(c,1) = size(d,1);
```

5. Recalculate the distance between each data point and newly obtained cluster centers.

```
centroid( c , : ) = centroid( c, : ) / pointsInCluster(c, 1);
centroid( c , : ) = (rand( 1, data_dim) .* data_diff) + data_min;
```

6. If no data point was reassigned then stop, otherwise repeat from step 3).

3.4.2 Density-based clustering (DBSCAN)

DBSCAN is a density constructed algorithm which determines clusters with arbitrary shape and with insignificant number of input parameters. The input parameters prerequisite for this algorithm is the radius of the cluster (Eps) and minimum points essential inside the cluster (Minpts).
The elementary indication behind this DBSCAN algorithm is as follows,

The Eps neighborhood of a point p, indicated by NEps(p) is distinct by

$$N_{Eps}(p) = \{p \in D | dist(p,q) \leq Eps\}$$

There are two types of points in the cluster, the points which is inside the cluster (core points), and points on the border of the cluster (border points). The points which are neither core points nor border points are called noise points.

DBSCAN Density Based Spatial Clustering of Applications with noise is deliberate to determine the spatial data clusters with noise. The algoritm steps are as follows,

1. Select an arbitrary point p

```
ptsC  = zeros(Npts,1);
```

2. Retrieve all points density-reachable from p w.r.t. Eps and Minpts.

```
function [C, ptsC, centres,neighbourPts] = dbscan(P, E, minPts)
```

3. If p is a core point, a cluster is formed.

```
ptsC(n) = Nc;
```

4. If p is a border point, no points are density reachable from p and DBSCAN visits the next point of the database.

```
  if length(neighbourPtsP) >= minPts
       neighbourPts = [neighbourPts  neighbourPtsP];
```

Continue the process until all the points have been processed.

3.5 Classification Algorithms

Classification entails of predicting a certain result based on a given input. First, the algorithm processes a training set holding a set of attributes and the corresponding outcome, which called goal or prediction attribute. Then, we tries to discover relationships between these attributes which make prediction the oucome possible. After that, we give the algorithm a data set which never knwon, refer to as prediction set, which holds the same set of attributes, except for the prediction attribute – not yet known. The algorithm analyses the input and yields a prediction. The prediction accuracy defines the algorithm effeciency.

3.5.1 Support Vector Machine (SVM)

In nowadays machine learning applications, support vector machines (SVM) are viewed as an unquestionable requirement attempt it offers a standout amongst the most powerful and precise strategies among all surely understood algorithms. It has a sound hypothetical establishment, requires just a dozen samples for training, and is insensitive to the dimensions number. Furthermore, effective techniques for training SVM are likewise being developed at a quick pace. In a two-class learning undertaking, the purpose of SVM is to locate the best grouping capacity to recognize individuals from the two classes in the training data. The metric for the idea of the "best" classification function can be acknowledged geometrically.

For a linearly separable dataset, a linear classification function compares to an isolating hyperplane f (x) that goes through the center of the two classes, separating the two. When this function is resolved, new data case xn can be classified by basically testing the indication of the function f (xn); xn fits in with the positive class if f (xn) > 0. Since there are numerous such linear hyperplanes, what SVM further guarantee is that the best such function is found by boosting the margin between the two classes. Instinctively, the margin is characterized as the measure of space, or separation between the two classes as characterized by the hyperplane.

Geometrically, the margin compares to the most limited separation between the nearest data points to a point on the hyperplane. Having this geometric definition authorities us to investigate how to expand the margin, so that despite the fact that there are a boundless number of hyperplanes, just a couple qualify as the solution for SVM. The motivation behind why SVM demands discovering the greatest margin hyperplanes is that it offers the best generalization capability. It permits not just the best classification fulfillment (e.g., accuracy) on the training data, additionally leaves much space for the right classification without bounds data. To guarantee that the most extreme margin hyperplanes are really found, a SVM classifier endeavors to amplify the accompanying function as for $\vec{w}$ and b:

$$lp = \frac{1}{2}\,||\vec{w}|| - \sum_{i=1}^{t} \alpha_i\, y_i\,(\ \vec{w}\ .\ \vec{x_i} + b) + \sum_{i=1}^{t} \alpha_i$$

where t is the number of training samples, and αi,i = 1,..., t, are non-negative numbers such that the derivatives of lp with respect to αi are zero. αi are the Lagrange multipliers and lp is called the Lagrangian. In this equation, the vectors $\vec{w}$ and constant b define the hyperplane.

Matlab Implementation

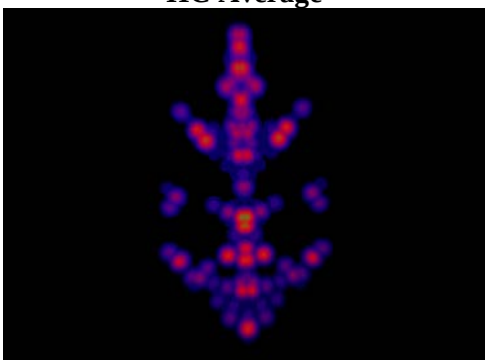
```
SVMMODEL=fitcsvm(tdata,labs_training,'Standardize',true,'KernelFunction','RBF','KernelScale','auto');
classOrder=SVMMODEL.ClassNames;
sv=SVMMODEL.SupportVectors;
CVSVMModel = crossval(SVMMODEL);
classLoss = kfoldLoss(CVSVMMODEL);
result = predict(SVMMODEL,testdata);
```

RESULTS A DISCUSSIONS

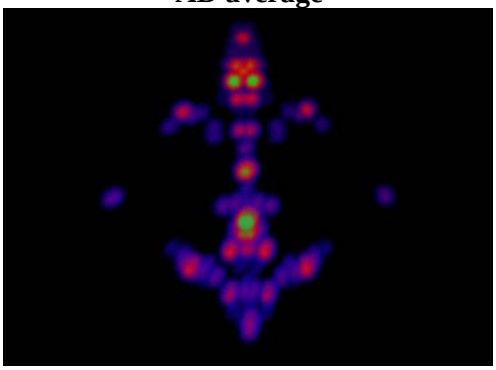
Data Visualization:

Visualization of average data

HC Average



AD average



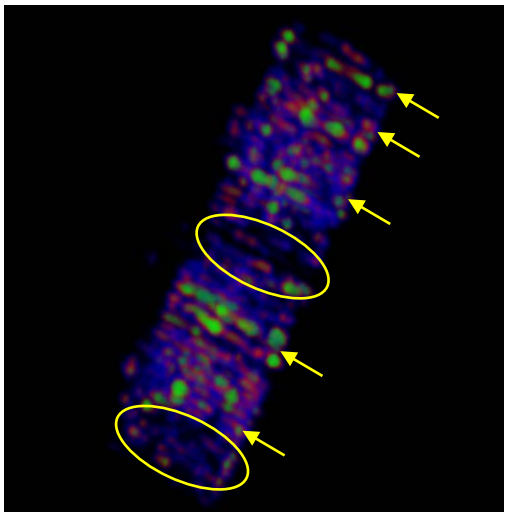
Here we calculated the average number of each connection and then visualized the average brain connection network of HC and AD. Color change (blue-red-green) indicates the increase of the numbers (weight of the connections). Comparing the visualization results of HC and AD, there are some parts of the brain network of AD patients experience abnormal activities. Those parts can serve as potential features for our future study. To define features, we not only need to find the areas that show a difference with average data, we also need to see consistency of all the subjects among the same group. For this reason, we need to put together each subject matrix from each group and exam them as a whole object.

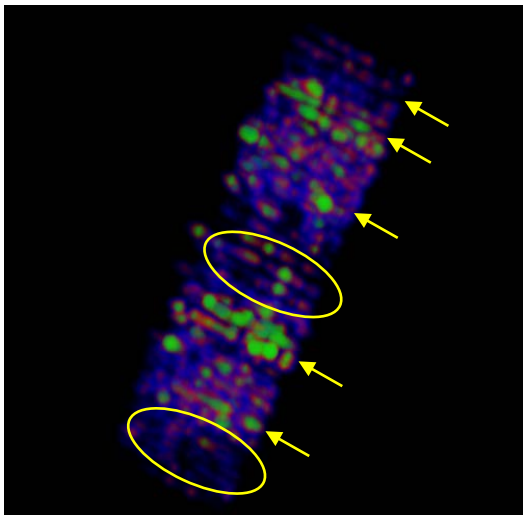
Visualization of all the subjects together for each group:

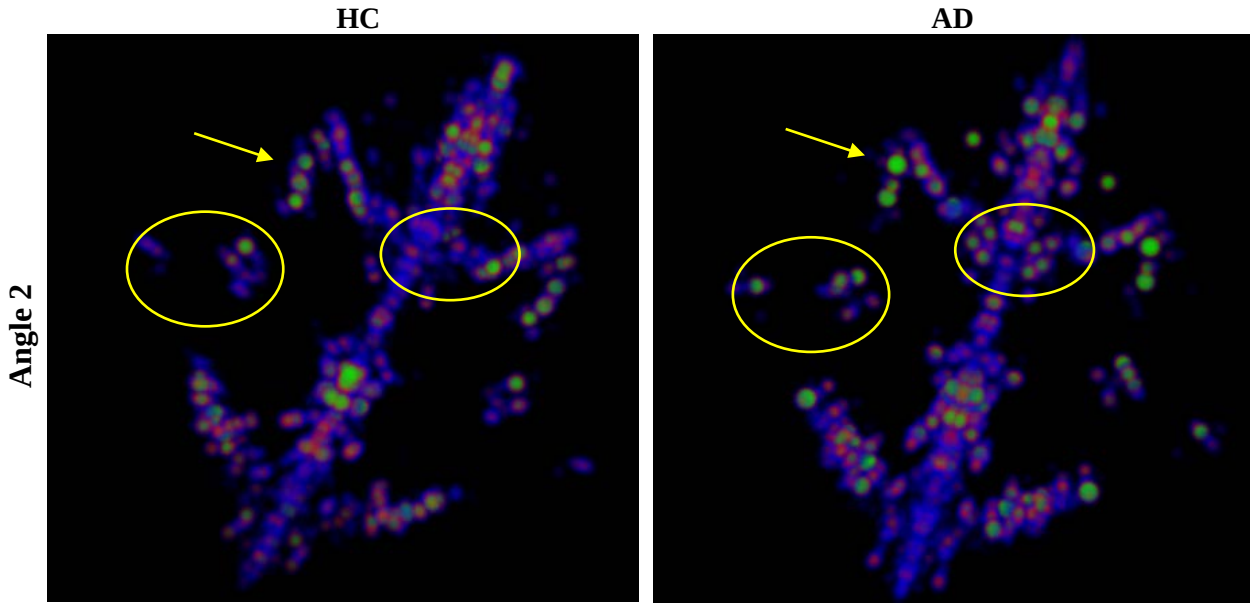
HC

AD

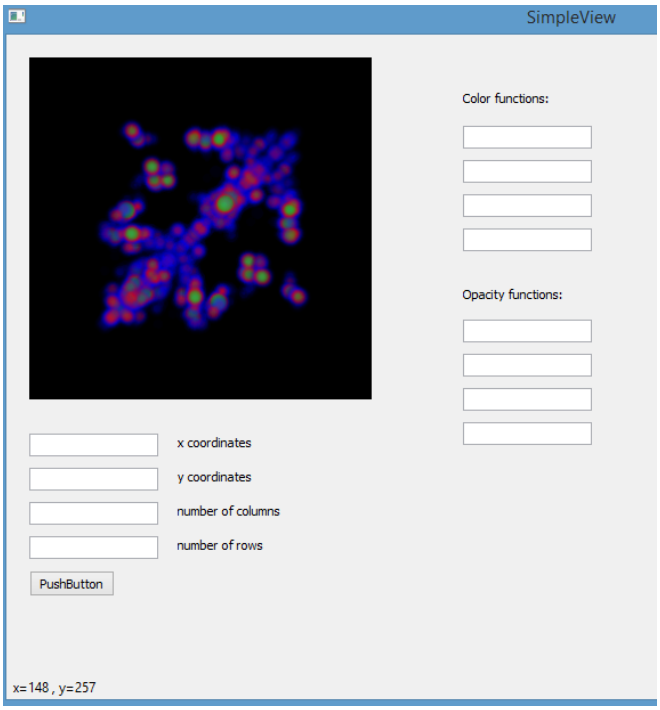
Angle 1







Here I marked a few regions that show different activities of brain network in comparing AD group and HC group. Those parts of the brain network can serve as potential features, especially those that are consistent among all subjects of the group. The main characteristic of Alzheimer's disease is neuronal loss in the brain.



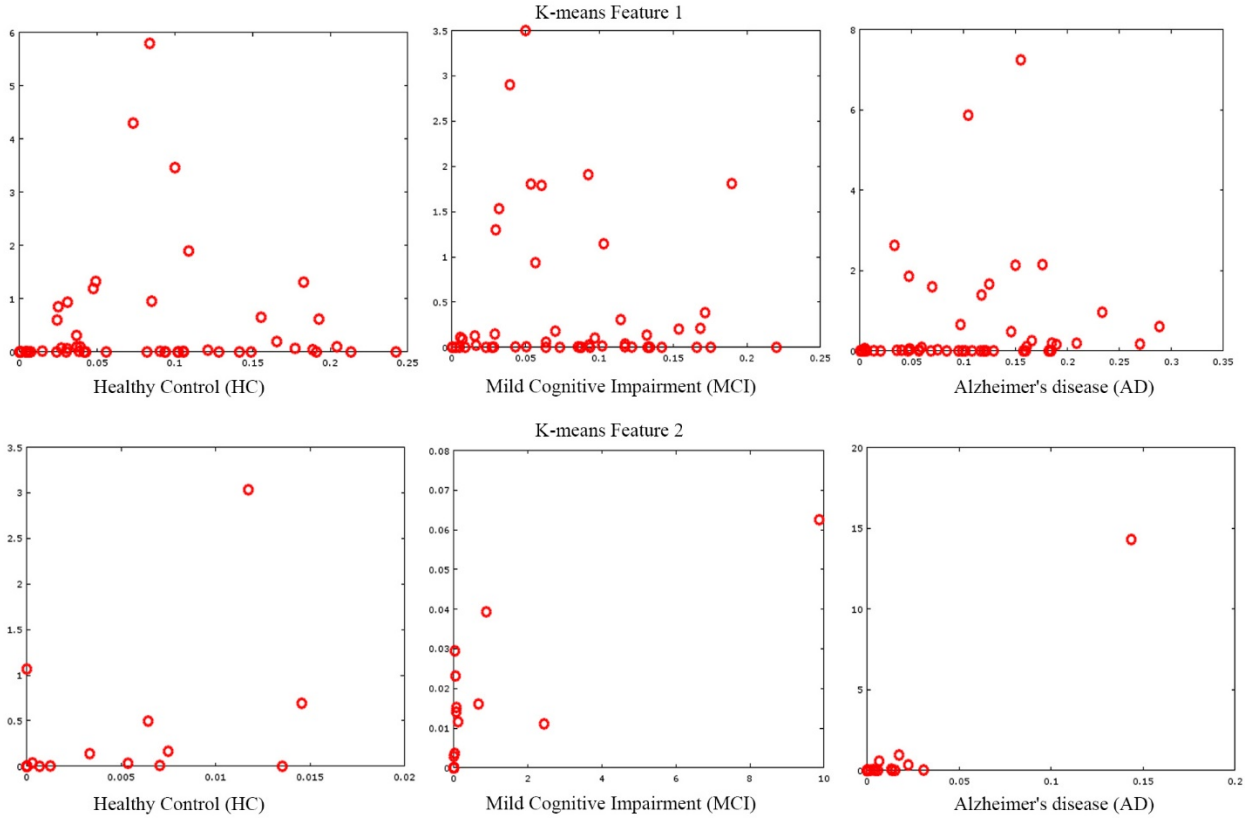
Some of the features indeed show a loss of connections in AD patient's brain, comparing to healthy people. Interestingly, there are also a few features show over-activities for AD brain network. Those different features indicate a complex mechanism for Alzheimer's disease. We are interested in extracting those features and apply different algorithms to study them and eventually we want to generate a model that can help in diagnosing Alzheimer's disease.

Feature Extraction:

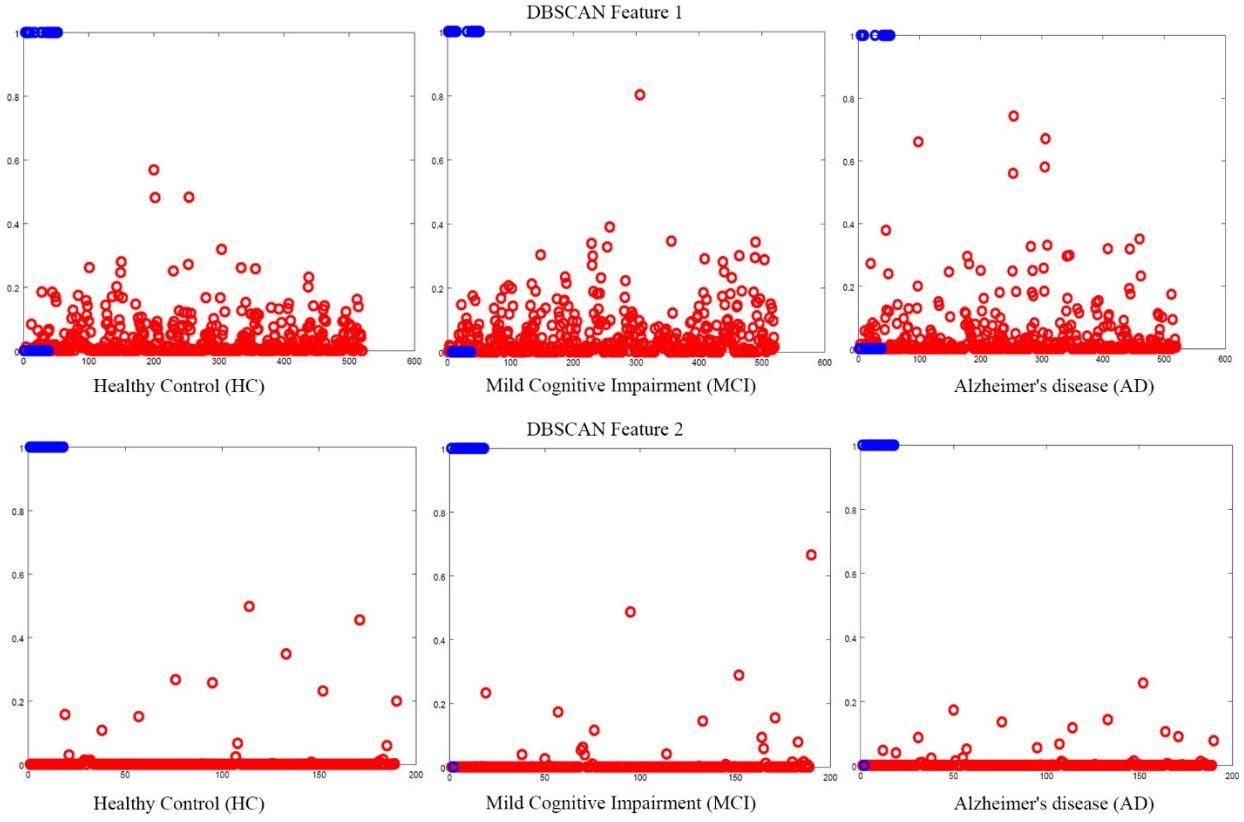
Feature extraction is done with user interface generated by QT. With this user interface, we can individually assign different colors and opacities for different range of data. Also, we can define the feature area by assigning the coordinates, row number and column number. By clicking the button, the feature matrices will be generated and exported as csv files.

Data Mining Algorithms

We have found that in order to visualize the centroids of the cluster k-means is better as we can find the centroids directly from k-means. Therefore, we have implemented k-means in octave and drawn the centroids of clusters. It looks better. The purity of the first feature in each group is 97% while in the second feature is 99%. Furthermore, the k-means help to find the outliers of our datasets.

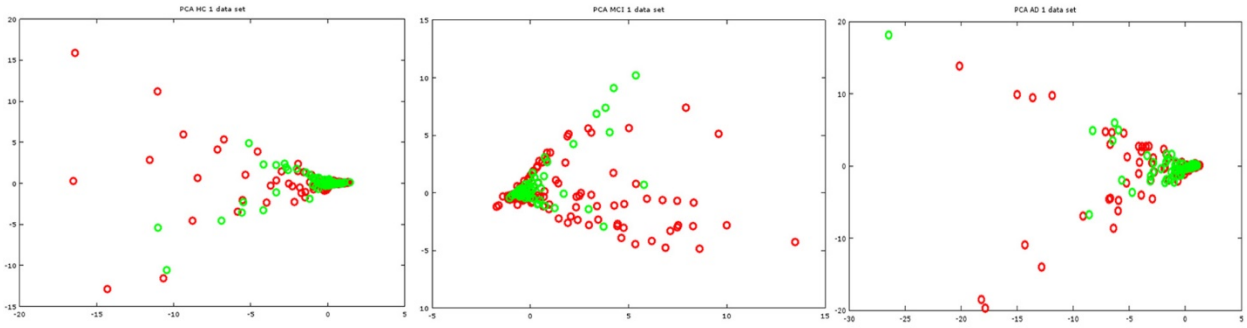


In DBSCAN we have used 10 cm as the radius of the cluster and the minimum threshold point is 10 to be considered for Core points. In this implementation we have shown the Core points of the extracted features of AD, HC and MCI. Also we have shown the Noise and the Border points. Here Noise points are zero and the border points are one.



Furthermore, We have implemented PCA and LDA for the features slightly different way. As we have observed that we use distinguish the features into 3 separate clusters; the third cluster has very less number of points and also in overlapping state with other cluster. For this reason, we have used 2 cluster format to make it well-visualized. In this state of my observation we can conclude that, the second features outperforms the first features with respect to coverage rate.

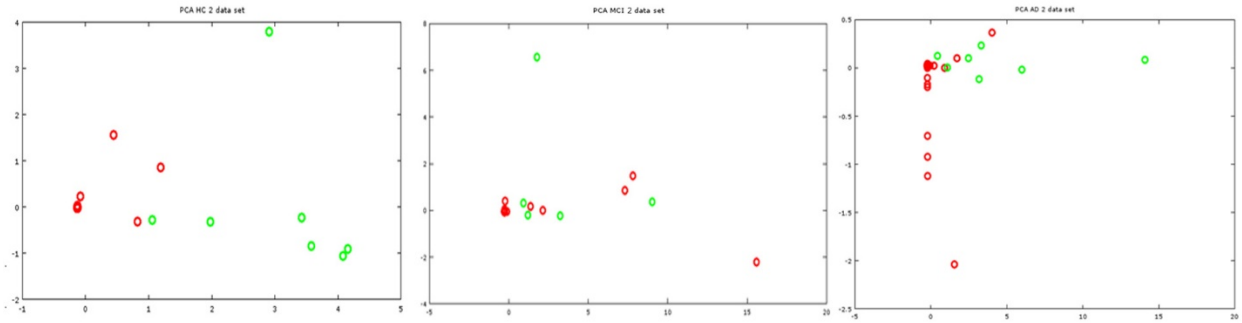
PCA Feature 1



Healthy Control (HC) Mild Cognitive Impairment (MCI) Alzheimer's disease (AD)

For the second feature, the first PCA component of AD dataset covers 92.37% of the total variance, the first PCA component of HC dataset covers 73.89% of the total variance and the first PCA component of MCI dataset covers 85.49% of the total variance. On the other hand, for the first features the first PCA component of AD dataset covers only 23.57% of the total variance, the first PCA component of HC dataset covers 22.10% of total variance and the first PCA component of MCI dataset covers 17.17% of the total variance.

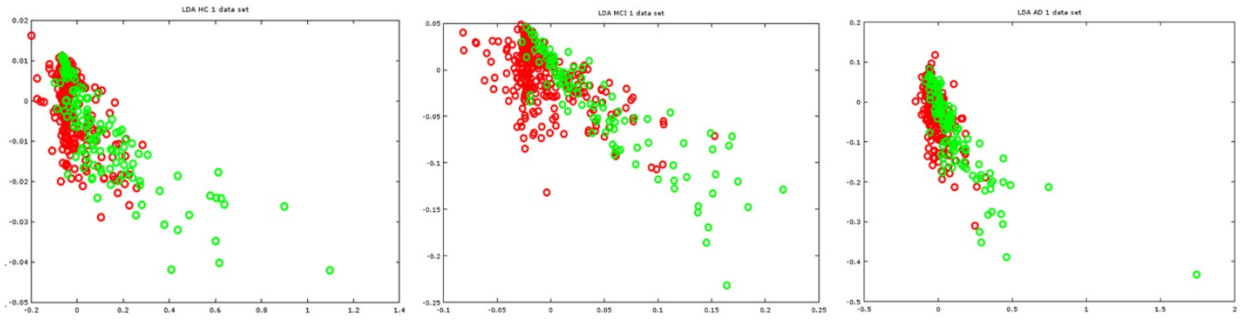
PCA Feature 2



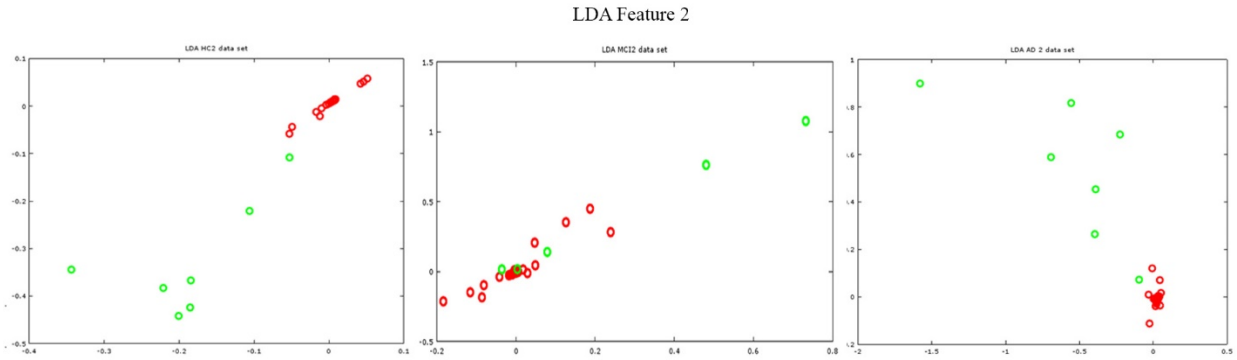
Healthy Control (HC) Mild Cognitive Impairment (MCI) Alzheimer's disease (AD)

Therefore, we have to consider more than one PCA component for the first features to make it in acceptable rate. Moreover, the normalization of the data does not show any significant change in the overall performance. We have use z-score normalization process. We have observe that, most of our data are very close to zeros and after calculating their mean and standard deviation, we see that, the values are less than 0.01 which very close zero actually. That is why z-score normalization is not effective in this case.

LDA Feature 1



Healthy Control (HC) Mild Cognitive Impairment (MCI) Alzheimer's disease (AD)



LDA compared to PCA is better algorithm in case of well separated cluster visualization. However, in our case we are getting sub optimal clustering.

We have implemented SVM, a supervised classification algorithm using MATLAB. In the implementation of SVM we start by reading the training data. Since our original data did not have labels, we created the corresponding labels file for training data. In our implementation we train SVM model using standard fitcsvm function. fitcsvm(training data, labels) returns the support vector machine classifier SVM model trained using the sample training data. The resulting SVM model is used to predict the test data using predict function.

We tested our SVM classification model on three cases. In the first case SVM classification model was applied on the whole dataset of AD, HC without any dimensionality reduction. The resulting accuracy was only 52%. Then we tested our SVM model on the dataset of AD, HC obtained after applying PCA. The resulting accuracy was 96.1%.

In the second case SVM classification model was applied on the whole dataset of HC, MCI without any dimensionalty reduction. The resulting accuracy was again 52%. Then we tested our SVM model on the dataset of HC, MCI obtained after applying PCA. The resulting accuracy was 96.1%.
Since our SVM model was giving high accuracy for the data set obtained after applying PCA, in the third case SVM classification model was tested on the MCI, AD dataset obtained after applying PCA. The resulting accuracy was 95.53%.

In this way our SVM classification model was successfully able to distinguish/predict 1) AD,HC 2)HC, MCI and 3) MCI,AD dataset features.

SVM Model without PCA

Datasets	**Misclassification Error**	**Accuracy**
HC, AD	0.48	0.52
HC, MCI	0.48	0.52
MCI, AD	-	-

SVM Model with PCA

Datasets	**Misclassification Error**	**Accuracy**
HC, AD	0.039	0.961
HC, MCI	0.039	0.961
MCI, AD	0.044	0.953

CONCLUSION AND FUTURE DIRECTION

This paper focus on the discrimination of patients with different cognitive impairments: MCI and AD. This is an important problem due to the inexistent consensus on the diagnostic frontier between these two impairments. In this sense, we use neuropsychological data to perform the discrimination. To tackle this problem, we apply state-of-art data mining techniques, we implemented K-means and DBSCAN algorithms for clustering purposes, also we used PCA, LDA to reduce the dimensionality of the huge brain datasets in each group: HC, MCI, AD, then for classification we applied SVM algorithm. We used these machine learning algorithms to discover the connectivity among different brain regions for AD study. Our results showed that the LDA leads to better discriminative results. Therefore, as long as the SVM accuracy increases, the higher prediction capability of the models was noticed.

In the future we plan to use LDA result with SVM and compare the results that we got from PCA and see the difference of each algorithm accuracy. For simplified reading our datasets in the model we could also use Support Vector Clustering (SVC) instead of SVM which not required labeling. For further clarification of our results we might try to extract more features and apply the algorithms on them.

References

[1] Lemos, L. *A data mining approach to predict conversion from mild cognitive impairment to Alzheimer's Disease*. Diss. Master's thesis, IST, 2012.

[2] Smith, Lindsay I. "A tutorial on principal components analysis." *Cornell University, USA* 51 (2002): 52.

[3] de Lemos, L. J. M., et al. "Discriminating Alzheimer? s disease from mild cognitive impairment using neuropsychological data." *Age (M±SD)* 70.8.4 (2012): 73-3.

[4] Bhagya Shree, S. R., H. S. Sheshadri, and Sandhya Joshi. "A Review on the Method of Diagnosing Alzheimer's Disease using Data Mining." *International Journal of Engineering Research and Technology*. Vol. 3. No. 3 (March-2014). ESRSA Publications, 2014.

[5] Sun, Liang, et al. "Mining brain region connectivity for Alzheimer's disease study via sparse inverse covariance estimation." *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009.

[6] "Mild Cognitive Impairment (MCI)." Mild Cognitive Impairment. Web. 13 Nov. 2015. <http://www.mayoclinic.org/diseases-conditions/mild-cognitive-impairment/basics/definition/con-20026392>.

[7] "GNU Octave." *GNU Octave*. Web. 13 Nov. 2015. <http://www.gnu.org/software/octave/>.

[8] "Linear Discriminant Analysis Bit by Bit." Linear Discriminant Analysis Bit by Bit. Web. 22 Nov. 2015. <http://sebastianraschka.com/Articles/2014_python_lda.html#preparing-the-sample-data-set>.

[9] "K-means Clustering Algorithm - Data Clustering Algorithms." K-means Clustering Algorithm - Data Clustering Algorithms. Web. 22 Nov. 2015. <https://sites.google.com/site/dataclusteringalgorithms/k-means-clustering-algorithm>.

[10] "A Tour of Machine Learning Algorithms." Machine Learning Mastery. 25 Nov. 2013. Web. 25 Nov. 2015. <http://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/>.

[11] Wu, Xindong, et al. "Top 10 algorithms in data mining." Knowledge and Information Systems 14.1 (2008): 1-37.

[11] Parimala, M., Daphne Lopez, and N. C. Senthilkumar. "A survey on density based clustering algorithms for mining large spatial databases." International Journal of Advanced Science and Technology 31.1 (2011)