STRUCTURAL BRAIN CONNECTOME EMBEDDING FOR ALZHEIMER'S DISEASE

by

Gurur Gamgam

B.S., Electrical and Electronics Engineering, Boğaziçi University, 2017

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronics Engineering
Boğaziçi University
2019

ACKNOWLEDGEMENTS

First and foremost, I would like to express sincere gratitude to my advisor Prof. Burak Acar for his encouragement, motivation and immense knowledge. His support during my master's degree was very important for me to complete the thesis and research. He improved my scientific perspective with his wonderful teaching. Without his guidance and help, this thesis would not have been possible.

Secondly, I would like to thank all my labmates at the VAVlab, specially Göktekin Durusoy and Demet Yüksel Dal, for creating a peaceful and productive working environment.

I am also grateful to Elif Gürbüz and Kemal Nennioğlu for accompanying me on this journey. Their presence was very encouraging for me. I want to thank my other friends who always support and motivate me.

Lastly, I would like to show my deepest gratitude to my family : my mothe, my father and my lovely sister. I feel lucky to have such a family who has always supported me throughout my life. I appreciate their eternal love and faith which always motivates me.

ABSTRACT

STRUCTURAL BRAIN CONNECTOME EMBEDDING FOR ALZHEIMER'S DISEASE

Neurodegenerative diseases are known to alter brain connectivity. Alzheimer's Disease (AD) is the most common one among these diseases. Although, many researches have been made to understand AD, there are still more to explore about the complicated nature of AD. To solve these mysteries, features extracted from connectomes are widely used. Following the poor specificity of global connectome features, more recently focus has been shifted towards substructures as potential biomarkers. A new model, inspired by the Deepwalk, is proposed to represent these substructures in this thesis. The model treats each individual connectome as a unique graph and learns nodal embeddings per connectome by means of a random walk and a neural network approach. The learned nodal embeddings are used as latent representations of local connectivity and their discriminative power is assessed in SVM based leave-one-out experiments over a cohort of 91 individuals. Promising results were obtained for AD-SCI / AD-MCI / MCI-SCI / AD-MCI-SCI classification tasks. Apart from classification, such latent representations of local connectivity may serve as an appropriate space to define the continuum of neurodegenerative disease progression temporally and spatially which means nodal embeddings can be utilized for monitoring disease progression

ÖZET

ALZHEİMER HASTALIĞI İÇİN YAPISAL BEYİN HARİTALARI GÖMÜLERİ

Nörodejeneratif hastalıklar beyin bağlantılarını değiştirir. Alzheimer hastalığı, nörodejeneratif hastalıklar arasında en yaygınıdır. Alzherimer'ı anlamak için yapılan çalışmalara rağmen, hastalığın karmaşık yapısıyla ilgili hala keşfedilicek şeyler bulunuyor. Bu yapıyı anlamlandırmak için, beyin haritasından çıkarılan genel yapıyı yansıtan özellikler yaygın bir şekilde kullanıldı. Bu özelliklerin düşük özgüllüğe sahip olmasından dolayı, son zamanlarda beyindeki belirli yapılar gizil bir işaretleyici olarak kullanılmaya başlandı. Bu tezde, bu yapıları göstermesi için Deepwalk'tan esinlenen yeni bir model sunuluyor. Bu model her bir kişisel beyin haritalarını eşsiz bir çizge olarak sayar ve rastgele yürüşlerleri sinir ağıyla birlikte kullanarak her bir beyin hartasi için düğümsel gömüler öğrenir. Öğrenilen düğümsel gömüler yerel bağlantıların gizli gösterimleri olarak kullanılır ve bu gömülerin ayırıcı gücü SVM temelli birini dışarı bırak deneyleriyle hesaplanır. Bu deneylerde AD-SCI / AD-MCI / MCI-SCI / AD-MCI-SCI sınıflandırma görevleriyle ilgili umut verici sonuçlar alındı. Sınıflandırma dışında, bu yerel bağlantıların gizli gösterimleri nöredejeneratif hastalıkların konumsal ve zamansal ilerleme sürecini tanımlamak için uygun bir uzayda bulunuyor olabilirler yani düğümsel gömülerden hastalığın ilerlemesini gözlemek için yararlanılabilir

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

k	Number of negative samples
w	Window length
K	Walks per node
n^t	Target node
n_k^c	Context node k
C	Corpus
$\underline{\underline{\mathbf{C}}}$	Context Embeddings Matrix
$\underline{\underline{\mathbf{W}}}$	Word Embeddings Matrix
$\underline{\mathbf{h}}$	Hidden Layer Vector
$\underline{\mathbf{n}}^t$	One-hot vector of target word
$\underline{\mathbf{n}}^c$	One-hot vector of context word
$\underline{\mathbf{q}}$	Score vector
q_i	ith entry of vector $\underline{\mathbf{q}}$
n_k	kth entry of one-hot vector
J	Cost function
E	Edges
e_{ij}	Edge between node i and node j
W_{ij}	Volume normalized weighted connectivity between i and node j
L	Random walk length
V	Vocabulary Size or Node Number
$P(n^t\|\{n_k^c\})$	Probability of being a target word when context words are given
$P(\{n_k^c\})\|n^t$	Probability of being context words target word is given
π_{ij}	Transition probability from node i to node j

LIST OF ACRONYMS/ABBREVIATIONS

AAL	Automated Anatomical Labeling
Aβ	Beta-amyloid
AD	Alzherimer's Disease
BOLD	Blood-oxygen-level-dependent
CBOW	Continuous-Bag-of-Words
CDR	Clinical Dementia Rating
CI	Cue Index
CNN	Convolutional Neural Network
CSF	Cerebrospinal Fluid
DGNR	Deep Neural Network for Graph Representation
DWI	Diffusion Weighted Magnetic Resonance Imaging
FA	Franctional anisotropy
FCSRT	Free and Cued Selective Reminding Test
fMRI	Functional Magnetic Resonance Imaging
fNET	Functional Network
GCN	Graph Convolutional Network
ISE	Integrated Squared Error
LLE	Locally Linear Embeddings
MCI	Mild Cognitive Impairment
MRI	Magnetic Resonance Imaging
NCE	Noise Contrastive Estimation
NN	Neural Network
PGSE	Pulse Gradients Spin Echo
SCI	Subjective Cognitive Impairment
SDNE	Structural Deep Network Embedding
sNET	Structural Network
SVD	Singular Value Decomposition
SVM	Support Vector Machine

T1w	T1 weighted Magnetic Resonance Imaging
TFE	Turbo Field Echo
TFR	Total Free Recall

1. INTRODUCTION

Aging gives rise to deterioration of cognitive functions. Possible causes of age-related cognitive problems are degeneration of hippocampus, region of brain that performs formation and retrieval of memories, decreased blood flow to the brain at older age and decline in hormones and proteins that guard and restore brain cells [1]. When there is no underlying condition apart from aging process, loss of cognitive functions is called as age-associated memory impairment. However in the presence of neurodegenerative diseases , loss in cognitive abilities becomes more severe. Heavy memory loss, confusion doing familiar tasks, difficulty with language, and personality changes are symptoms of such diseases and these group of symptoms is described as dementia. Dementia often occurs in older age and is different from normal-aging. Alzheimer's disease, dementia with Lewy bodies, Parkinson's disease dementia and Huntington's disease are neuro-degenerative diseases that cause dementia. Aside from neuro-degenerative diseases , vascular disorders, long term drug and alcohol addiction and depression may result in dementia.

Alzheimer's disease (AD) is the most common cause of dementia and accounts for 60-80 percent of dementia cases [2]. Beta-amyloid (Aβ) and tau proteins reaches abnormal levels in the brain of someone with AD and forms plaques and tangles to disrupt neuronal activities. Proteins transfer freely between brain and Cerebrospinal Fluid (CSF) , so levels of Aβ and tau protein in CSF sample obtained by lumbar puncture can be used for accurate diagnosis of early AD [3]. Early diagnosis for AD is crucial because progression have already occurred before AD shows its symptoms. However invasive biomarkers are relatively expensive and has potential side effects [4]. Improvement of magnetic resonance imaging (MRI) techniques gives rise to non-invasive biomarkers. Non-invasive biomarkers are inexpensive and easy to perform. For example, atrophy in brain measured by MRI is already used as biomarker for early diagnosis of AD. However volume decline in brain is also part of normal-aging thus distinguishing early AD by only looking at volume change is challenging.

Aside from measurement of atrophy in brain, many studies have been made to find relevant non-invasive biomarkers in recent years. These studies have focused on leveraging brain' network-like properties since it is well known that structural and functional organization of a brain can be modeled as a network. Thus a number of network based biomarkers are proposed and investigated for the purposes of early diagnosis and staging of AD. Network-based biomarkers uses different connectivity networks of brain, also named as connectomes , obtained by structural and functional MRI.

Connectome is a simplified description of the elements and connections organizing human brain, hence it is a network model [5]. In order to assemble a comprehensive network model , two main aspects of connectome must be emphasized : network elements (nodes) and connections between elements (edges). Choosing network elements at microscale, neurons and synapses, is not feasible since there are approximately 10^{10} neurons in human brain. A human connectome must provide realistic model to represent information about brain. Defining network elements at macroscale, areas of cerebral cortex or brain regions, is more practical and in this setup, number of elements varies in the range of 90 to 1000. At macroscale, brain is generally parcellated by predefined anatomical templates which can be volume-based or surface-based. Automated Anatomical Labeling (AAL) atlas is the most common volume-based template when Desikan-Killiany atlas and Destriux atlas are the most popular among surface-based templates [6]. With the tools of graph theory, it has been shown that connectomes at macroscale have non-random network properties such as existence of clusters of brain regions and small-worldness . Absence of a universally accepted parcellation scheme is a major drawback of macroscale connectomes.

Connectivities can be defined in two way : functional or structural connectivity hence resulting functional networks (fNETs) and structural networks (sNETs). These networks models can successfully represent functional and structural organization of the human brain.

fNETs are defined based on correlation of blood-oxygen-level-dependent (BOLD) MRI signals acquired by functional MRI (fMRI). Measured BOLD signal is a sign of activity at any given time. To be more precise, BOLD signals are changes in magnetic susceptibility and MRI tissue contrast that are indirectly indicative of underlying changes in spontaneous or experimentally controlled brain activation [7]. After the acquisition of BOLD signals , correlation of BOLD signals forms fNETs. Different measures for correlations can be applied to construct fNETs such as partial correlation or cross correlation.

sNETs are defined based on fibers reconstructed using tractography applied to diffusion weighted MRI (DWI) data. Cerebral white matter is considered as a marker for structural connectivity because myelinated fibers (axons) are found in white matter. Fiber tracts forms pathways between brain regions and DWI can represent information about the spatial orientation of fiber tracts. However current imaging methods are limited to detect fiber orientation where multiple fiber tracts are intersected or crossed. To overcome this problem, tractography, a computational algorithm, is proposed. Tractography can be utilized to trace complicated fiber tracts and results derived from tractography are consistent with known brain pathways [8]. There are two different approaches used for tractography : deterministic and probabilistic approaches. Deterministic approach assumes single orientation at each voxel (three dimensional pixel) , while probabilistic approach assumes a distribution of orientations.

Connectionist approaches have gained increasing popularity in parallel with the developments in MRI and through the incorporation of the well-developed graph theory into brain research. Initially fNETs and later sNETs have been being proposed and evaluated to study neurodegenerative diseases, specially AD [9]. Connectionist approaches performs network analysis however different studies use different analysis methods which can be grouped into three main categories : data centric approaches, network based local and global features, subnetwork based approaches.

Data centric approaches treat connectivity values as features. Since the dimension of resulting feature is too high a preprocessing method is required . Statistical

tests or support vector machine (SVM) based feature elimination algorithms are used as a preprocessing method to reduce dimensionality as well as selecting discriminative features. In this sense, Chen *et al.* [10] ranked fNET features in order to find discriminating features and performed classification on healthy and AD patients. Dai *et al.* [11] combined sNET and fNET features, including BOLD spectra to distinguish AD patients from healthy subjects. However direct usage of connectivity values fails to leverage network properties of brain, thus the information extracted from only significant edges may provide less insight into AD. In a different data centric study which accounts network structure of brain, Dipasquale *et al.* [12] used independent component analysis (ICA) on fMRI signals and observes disconnection within default mode network , a set of brain regions including medial frontal and posterior cingulate areas of the cortex , and functional connectivity damage. This work doesn't include a connectome analysis since fMRI data is directly used without construction of any fNET. Networks are decomposed into distinct networks by the use of high dimensional ICA. Decomposed networks are correlated in their fluctuations but they are maximally independent in spatial domain.

Extracting local and global features from network is another application of connectome analysis. Motivation behind using local and global features is that healthy brains have optimal balance between segregation and integration [13]. Segregation local features that measure the degree of nodes which the graph can be decomposed while integration is global features that measure global efficiency of all nodes [7]. It is found that the modularity is significantly reduced in an AD brain and it results in loss of small-world networks [8]. Small-world networks are networks with high clustering and short path length properties. Loss in small-world networks proves disruption in segregation and integration. Khazaee *et al.* [14] extracted a number of local (eg. clustering coefficient) and global (eg. characteristic path length) features from thresholded fNET and selected the most discriminative features by Fisher score algorithm to train an SVM classifier. Wee *et al.* [15] used local clustering coefficients from multiple sNETs after selecting statistically significant ones to diagnose Mild Cognitive Impairment (MCI). In [16] , Wee *et al.* combined fNET and SNET to derive nodal clustering coefficients and classified MCI and control subjects via a SVM classifier. Parasad *et*

al. used normalized fiber counts as features of sNET to distinguish AD patients from healthy patients [17]. Local and global features utilize network properties however they commonly suffer from low specificity [18].

More recently, there has been a paradigm shift with the recognition of multi-factorial nature of neurodegenerative diseases and the human brain's multi-subnetwork structure, each performs a different cognitive function [19]. Thus subnetwork based approaches come into prominence. These approaches mainly perform community detection using spectral algorithm [20]. Chen *et al.* [21] performed community detection using the spectral algorithm on fNET by finding group level connectivity networks and detected insula module lost its symmetric functional connections. Dai *et al.* [22] defined seed region of interests to employ community detection on a reduced network and found that highly connected hub regions are damaged. Sun *et al.* [23] also performed the same algorithm and observed that abnormal changes in the modularity of fNET, representing the reorganization and separation of subcortical regions.

In studies of AD, network analysis methods are limited by approaches outlined above. Despite the increasing popularity of machine learning methods on various areas, there is no remarkable study using machine learning for connectome analysis . However machine learning is already successfully applied to networks (graphs) to analyse social networks and protein networks. Machine learning tools may capture additional discriminative information and provide a better understanding of the progression and diagnosis of AD.

This thesis proposes to use a random walk based graph embedding method, a machine learning based method applied to networks, to represent sNETs for the purposes of diagnosis and staging of AD. More specifically, a corpus of node sequences are created by the means of random walk approach. Different sNETs result in different rules for generating corpus hence each patient models a different language. However, each sNET node is treated as a word, thus each patient sharing a common vocabulary. These corpora is fed into neural probabilistic language model and nodal embeddings per sNET are learned. Each nodal embeddings is used as features of a patient to train

and test a SVM classifier to discriminate AD, MCI and SCI (Subjective Cognitive Impairment) cases in a cohort of 91 individuals. Different architectures and objective functions are experimented to find optimal embeddings. It is expected to nodal embeddings of patients with same clinical label have similar embeddings, resulting with discriminative local connectivity patterns.

Remainder of this thesis is organized as follows. Chapter 2 reviews state of art methods for graph embeddings. Chapter 3 describes the methodology, the algorithm used to obtain graph embeddings, and theoretical background of the method. Chapter 4 proposes classification results from different experiments. Chapter 5 discusses results and also combines manifold learning with graph embeddings. Chapter 6 concludes thesis.

2. STATE OF ART

Graphs have been utilized in various areas including biology, social networks and linguistics. Connectivity between elements can be modeled as a graph and it allows researchers to understand the underlying structure and function of complex networks as well as achieving different tasks such as link prediction , node classification and clustering [24]. Advertisement and friendship recommendations are applications of link prediction on social networks [25]. Labeling documents in citation networks [26] and finding disease proteins [27] are examples of node classification task. Clustering is grouping similar nodes into the same subset and it can be used in image segmentation [28].

An increasing body of research has been made to analyse complex networks. Traditional network measures such as node degree, clustering coefficient, modularity and path length can describe graph topology and extract structural information from graphs but they are not applicable to complex networks which have millions of nodes and they can't be adapted to a learning problem, hence network measures are usually inefficient and inflexible. Recently, there has been an increasing attention on graph embedding methods that aim to find latent representation of nodes that encodes graph structure. Graph embedding, or representation learning, methods assume that nodes are embedded in a low-dimensional vector space and geometric relationship in this embedding space reflects underlying structure of original graph. Many graph embedding algorithms have been proposed to deal with large scale graphs and they can be categorized into three main areas : factorization based methods, random walk based methods and deep learning based methods.

Factorization based methods perform matrix factorization to obtain embeddings. Laplacian Eigenmaps [29] and Locally Linear Embeddings (LLE) [30] are basic algorithms based on factorization. These algorithms are often referred to as nonlinear dimensionality reduction techniques. They construct a similarity graph from M dimensional feature vectors and find m dimensional embeddings of nodes belonging to

constructed graph, where $m << M$. Laplacian Eigenmaps seek to find closer embeddings of nodes when nodes' local connectivities are similar. LLE follows similar concept but it assumes embedding of a node is a linear combination of the embeddings of neighboring nodes. Both algorithms try to solve constrained optimization problem and performs eigendecomposition. Optimization problem is defined in terms of the connections of constructed graph and the resulting embeddings. Several possible matrices can be used to represent connections of graph such as node adjacency matrix, Laplacian matrix and node transition probability matrix. Laplacian Eigenmaps and LLE use node adjacency matrix in objective function. Complexity of these methods is in the order of square of node number, thus scalability is the main drawback for these algorithms.

Apart from Laplacian Eigenmaps and LLE , many different approaches have been considered for factorization based graph embeddings. While Laplacian Eigenmaps and LLE only concern with finding similar embeddings for connected nodes, other algorithms take into account preserving graph structure. Laplacian Eigenmaps and LLE do it indirectly by penalizing dissimilarity between embeddings of connected nodes however Graph Factorization [31], HOPE [32] and GRAREP [33] assume that inner product of embeddings must reflect the structure of graph. Main difference of these algorithms is the use of distinct matrices to represent the structure of graph. Graph Factorization finds embeddings of which inner product represents node adjacency matrix as close as possible. It also introduces norm of embeddings as a regularization term at the objective function. Complexity of Graph Factorization is in the order of node number so it is more scalable to complex systems compared to Laplacian Eigenmaps and LLE. However node adjacency matrix is not always positive semidefinite and it may cause problems in factorization. GRAREP algorithm factorizes powers of node transition probability matrix , hence embeddings capture higher order proximity. Despite the additional structural information that embeddings have, the order of complexity is same as Laplacian Eigenmaps algorithm which results in scalability issue. HOPE algorithm aims to preserve higher order proximity and considers various matrices such as Katz Index and Common Neighbors. HOPE points out that direct factorization of higher order proximity matrices is expensive, and proposes a novel algorithm to find

embeddings. Firstly authors assume that most of higher order proximity matrices are equivalent to multiplication of two sparse matrices. Then two sparse matrices are factorized with modified singular value decomposition (SVD) to find optimal embeddings. This algorithm also captures higher order proximity but unlike GRAREP, HOPE is scalable, its complexity is linear with the number of edges in the graph.

Random walk based methods rely on the assumption that nodes co-occurring in short random walks must have similar embeddings. Deepwalk [34] and node2vec [35] are examples of random walk based methods. Their approach to embedding problem is the same : generating random walks and training a neural network language model to get embeddings. Random walks are created starting from each node until a fixed length is reached. This procedure is also repeated which means there are multiple random walks starting from same node. Resulting random walks form a corpus. The corpus is fed into a neural network and nodal embeddings are extracted. In this model, nodes are treated as words and random walks are treated as sentences. Neural network basically tries to maximize cooccurrence of words that are close to each other. Main difference between random walk based methods and factorization based methods is that the latter inputs a deterministic measure of node similarity. By using a stochastic measure of node similarity, random walk based methods offer better performance [36].

Despite the main intuition behind Deepwalk and node2vec is very similar, they differ in corpus creation and objective function used in training. For generating random walks, Deepwalk uses unbiased random walks, i.e. next node in a random walk only depends on connections of current node. However node2vec manipulates random walks and creates biased random walks by defining new parameters to control probability of returning to same node and probability of walking to a node that is connected to a previously visited node. Additional parameters introduced in node2vec may change the captured information in resulting embeddings. These parameters can be adjusted to obtain embeddings that have more local or global information. In addition, Deepwalk uses hierarchical softmax to reduce the complexity of training while node2vec uses negative sampling to achieve the same. Despite algorithmic differences, Deepwalk and node2vec do not show any notable differences on experiments. Among them, Deepwalk

is the most common random walk based method and have already been successfully applied in social networks.

Factorization and random walk based methods can be thought as encoders. These methods find a mapping for nodes. However parameters of the mapping are not shared between nodes. Parameter sharing can reduce the order of complexity which is the main issue for factorization based methods. Another disadvantage of these methods is their lack of leveraging node attributes which can be highly informative in some applications.

Finally, deep learning based methods try to overcome issues summarized above , unlike factorization and random walk based methods, parameters are shared in the encoder and node attributes can be leveraged. Structural Deep Network Embeddings (SDNE) [37] and Deep Neural Network for Graph Representation (DGNR) [38] use deep autoencoders. Autoencoders are neural network models which learns compressed form of the input data in unsupervised fashion and widely used in image processing for dimensionality reduction and image denoising. SDNE proposes semi-supervised model which has a supervised and an unsupervised part. Unsupervised part is designed to preserve second order proximity, i.e. measurement of two nodes whether they share common neighborhood node, by finding an embedding that can reconstruct its neighborhood structure. Supervised part uses Laplacian Eigenmaps to penalize dissimilarities of embeddings of connected nodes. Joint optimization of two objectives leads to embeddings that preserve network structure. Parameters are shared in the unsupervised part of the model. DGNR combines random walks with deep autoencoders. Cooccurrence of nodes in short random walks is encoded to positive point-wise mutual information matrix (PPMI) [39] which is already a common tool in representing words [40]. PPMI matrix is fed into a stacked denoising autoencoder to obtain embeddings.

Autoencoder based methods preserve graphs structure and can be applied to large scale graphs. However they do not benefit from node attributes. Graph Convolutional Networks (GCNs) handle with this issue by performing convolution operation on graph

signals. GCNs are highly influenced by Convolutional Neural Networks (CNNs). CNNs have already been successfully applied in many areas such as image classification [41] and video processing [42]. Image and video signals are defined on the Euclidean domain. Convolution, filtering and pooling operations are well defined operations on this domain. However all graphs are not in the regular Euclidean domain, thus convolution and filtering operations must be redefined for such cases. This problem is overcome by spectral graph theory [43]. Spectral graph theory uses basic relationship that the convolution operation is equivalent to the multiplication in frequency domain. It is proven that basis functions of Graph Fourier Transform are eigenvectors of the Graph Laplacian matrix. After constructing theoretical background for Graph Fourier Transform, filtering is defined in frequency domain. Therefore the convolution operation in graph is indirectly identified and is linked to the Graph Laplacian matrix [44]. Graph Laplacian matrix represents the underlying structure of graph. Also Kth order spectral filters have got localization property, i.e. embedding of a node is affected by nodes that are maximum K connection away [45]. This ensures that GCNs capture local information like CNNs as well as preserving the graph structure. With the identification of convolution and filtering operation on graphs, GCNs are used at node classification tasks [26]. Embeddings of nodes are encoded in GCN layers. GCNs are scalable to complex graphs, their order of complexity is linear with respect to the number of edges.

A human connectome is a network model, hence it is a graph. Therefore, graph embedding methods can be applied to human connectomes. Embeddings that preserve the network structure can be utilized to analyse connectomes such as finding structural and functional disconnections between brain regions. Apart from disconnections, changes in connectivities up to higher order can be also detected by embeddings, since embeddings are capable of encoding higher order similarities. However higher order similarity information would not be useful for fNETs, since fNETs already consist of correlation values between BOLD signals. Thus embeddings extracted by the means of random walks or matrix factorization is not informative for fNETs. An example use of graph embeddings for fNETs is constructing a population graph of which nodes are subjects and node attributes are formed from fNET connectivity values of subjects, then training a GCN to perform node classification task [46]. However this approach

is only useful for classifying subjects, it does not contribute to answer the questions such as which subnetworks AD targets at and how AD evolves. Graph embeddings of sNETs may be capable of answering these questions, since nodal embeddings of a sNET can represent local connectivity patterns by leveraging higher order similarities. In this sense, random walks based methods can be utilized to observe changes in local connections. Since AD can be characterized as a disconnection syndrome, these observations may help to diagnose and monitor disease. It is also mentioned previously that random walk based methods are the most common and successful ones among graph embeddings.

3. METHODOLOGY

In this thesis, a modified version of Deepwalk algorithm is followed [34]. As it previously described in random walk based embeddings, Deepwalk creates a corpus from unbiased random walks and learns embeddings via a neural network. Thus the algorithm has two parts : generating a corpus by random walks and neural network based embedding learning.

The original Deepwalk uses binary graphs of which its adjacency matrix entries are either 0 or 1. Hence transition from one node to other node follows a uniform distribution. However in sNETs, weights of adjacency matrix are highly informative about the structure of brain. Thus a slight modification is made in generating random walks by using weighted adjacency matrices throughout this thesis.

Random walks start from each node and are last until a fixed length is reached. The number of random walks starting from each node is fixed. There are three parameters in creating the corpus : walks per node (K), random walk length (L) and node number (V). More specifically a corpus consists of $V \times K$, L length random walks.

Constructed corpus is fed into a special neural network to obtain nodal embeddings. This neural network was originally built to find word embeddings in language modeling. To learn embeddings, a center word (target word) and a set of words that appear with center word within a fixed window size (context words) are taken as a training pair. Depending on architecture, either the target word is predicted from context words (Continuous Bag-of-words, CBOW) or the context words are predicted from the target word (Skip-Gram). In this case, random walks are analogous to sentences and nodes are analogous to words. In this thesis, both architectures will be experimented and compared in terms of classification performance.

Deepwalk uses hierarchical softmax for faster training performance. Hierarchical softmax creates a binary tree to reduce the complexity of calculating the objective

function. However negative sampling is used throughout this thesis. Negative sampling randomly samples k words in training, for the purpose of distinguishing observed data and artificially generated data. Different k values are selected and experimented to observe the impact of negative sample number.

To sum up, methodology to obtain embeddings is the following: constructing a connectome as input, creating a corpus from connectome and learning embeddings from the corpus. Algorithm steps are shown in 3.1.

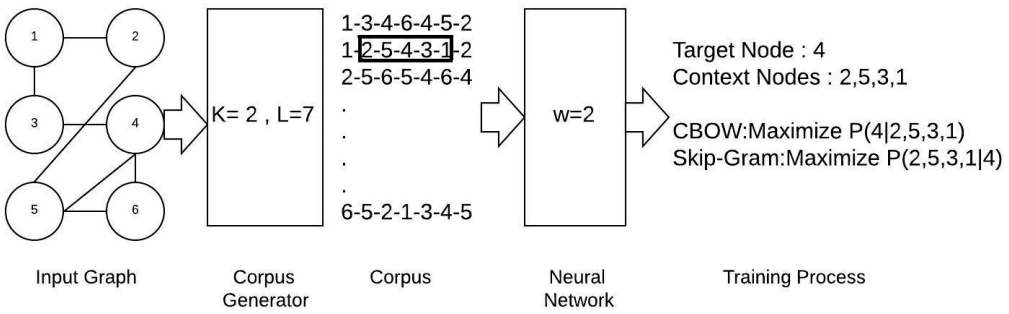

Figure 3.1. Deepwalk Algorithm

3.1. Connectome Construction

Let $\mathcal{G} = \{N, E\}$ represent an sNET where the nodes (N) correspond to cortical segments/parcels and the edges ($E = \{e_{ij}\}$) defined between pairs of nodes represent the strength of their connectivity as ascribed by DWI based tractography. FSL [47], FreeSurfer [48] and Tortoise [49] are used to preprocess, co-register and parcellate the T1 weighted MRI and DWI volumes (at $1.5mm$ isotropic sampling) using the Destrieux atlas . 4^{th} order Runge-Kutta integration based deterministic principal diffusion direction (PDD) tractography was applied with $30seed/voxel$, $0.7mm$ stepsize, 35° curvature and 0.15 fractional anisotropy (FA) thresholds as stopping criteria [50]. Minimum fiber length was set to $20mm$. Seed selection minimum FA criteria was 0.15. The connectivity e_{ij} between parcels n_i and n_j is defined using a volume normalized

weighted connectivity as,

$$W_{ik} = \sum_{\underline{\mathbf{r}} \in n_i} \tilde{\mathcal{N}}(\underline{\mathbf{r}}; \underline{\mathbf{e}}_k, \sigma) : \text{Association of fiber } f_k \text{ with parcel } n_i \quad (3.1)$$

$$e_{ij} = \frac{2}{v_i + v_j} \sum_{f_k} W_{ik} W_{jk} : \text{Connectivity between parcels } n_i \text{ and } n_j \quad (3.2)$$

where $\tilde{\mathcal{N}}(.; \underline{\mathbf{e}}_k, \sigma)$ is the truncated 3D isotropic Gaussian kernel[1] centered at $\underline{\mathbf{e}}_k$ (the closest end point of the fiber f_k to node n_i) with standard deviation σ. $\underline{\mathbf{r}}$ is a voxel position, v_i is n_i's volume. We have set $\sigma = 0.155mm$ using the Integrated Squared Error (ISE) [51].

3.2. Corpus Generation

A fixed length (L) random walk is run on each sNET separately by initiating the walk multiple times (K) from each node. The probability π_{ij}, to move from n_i to n_j, is defined as,

$$\pi_{ij} = \frac{e_{ij}}{\sum_j e_{ij}}, i \neq j \quad (3.3)$$

The resultant $V \times K$ L-length node sequences form the corpus from which a $D-$dimensional embedding is learned for each node and each sNET. $V = 148$ as the Destrieux atlas defines 148 parcels while K and L have set empirically .

3.3. Embedding Learning

Embeddings are learned via eural network language models. These language models are used in language modeling to get word embeddings to achieve syntactic and semantic tasks. Many different neural network architectures are proposed for obtaining better word representations [52]. Among them, CBOW and Skip-gram architectures

[1] $\tilde{\mathcal{N}}(\underline{\mathbf{r}}; \underline{\mathbf{e}}_k, \sigma) = \frac{1}{0.74} \mathcal{N}(\underline{\mathbf{r}} - \underline{\mathbf{e}}_k; \underline{\mathbf{0}}, \sigma) \mathbb{1}(|\underline{\mathbf{r}} - \underline{\mathbf{e}}_k| < 2\sigma)$, where $\mathcal{N}(\underline{\mathbf{r}} - \underline{\mathbf{e}}_k; \underline{\mathbf{0}}, \sigma)$ is 3D isotropic Gaussian with zero mean and σ standard deviation, 0.74 is the normalization factor for truncation at 2σ, $\mathbb{1}(.)$ is the indicator function.

become highly popular recently and draw increasing attention. Compared to previously proposed techniques, CBOW and Skip-gram outperform them in terms of syntactic and semantic tasks for language modeling [53].

CBOW and Skip-gram architectures have one hidden layer. CBOW tries to predict the target word from the context words while Skip-gram tries to predict the context words from the target word. Detailed model architectures taken from [53] can be seen in Figure 3.2.

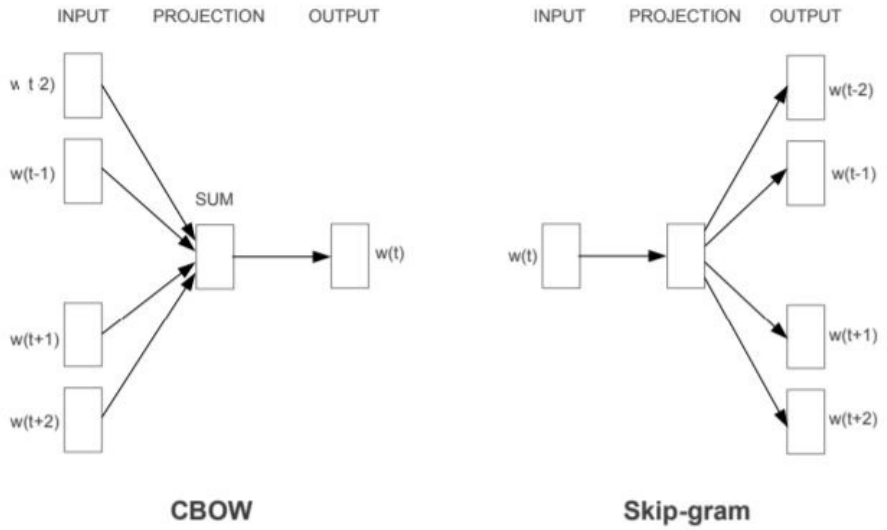

Figure 3.2. Neural network architectures for CBOW and Skip-Gram

3.3.1. CBOW and Skip-Gram

CBOW takes one target node and a number of context nodes, nodes that appear with the center node within a fixed window size, w. To provide better insight into formulation lets assume $w = 5$. Thus, subsequences of 11 nodes,one target node and ten context nodes, are extracted from the L-length node sequences that make up a corpus and used as a training sample. The target (center) node (n^t) and ten context nodes ($n_k^c, k \in [1, 10]$), are all encoded as a one-hot vectors, $\underline{\mathbf{n}}^t, \underline{\mathbf{n}}_k^c \in \mathbb{R}^V$.

Definition 1. One-hot Vector. A V dimensional one-hot vector representation of pth node is a binary vector that its p^{th} entry is 1 and other entries are 0s.

Input to hidden layer weight matrix $\underline{\underline{\mathbf{C}}} \in \mathbb{R}^{D \times V}$ stores D-dimensional context node embeddings in columns. The hidden layer, $\underline{\mathbf{h}} \in \mathbb{R}^{D}$, is the mean embedding of ten context nodes ($\{n_k^c\}$).

$$\underline{\mathbf{h}} = \frac{1}{10} \sum_{k=1}^{10} \underline{\underline{\mathbf{C}}}\ \underline{\mathbf{n}}_k^c \tag{3.4}$$

Hidden to output layer weight matrix $\underline{\underline{\mathbf{W}}} \in \mathbb{R}^{D \times V}$ contains target node embeddings that are subsequently used as nodal embeddings and utilized in classification. Hence output layer, $\underline{\mathbf{q}} \in \mathbb{R}^{V}$ represents scores. q_i is n_i's score of being the target word for the given context ($\{n_k^c\}$). Higher score means higher likelihood for n_i.

$$\underline{\mathbf{q}} = \underline{\underline{\mathbf{W}}}^T\ \underline{\mathbf{h}} \tag{3.5}$$

Skip-gram also extracts a target word and ten context words. However target word is input to Skip-gram model and ten context words are tried to be predicted. This results in each training sample consists of one target word and one of context words, thus ten training samples are produced from one pair of a target word context word set.

Input to hidden layer weight matrix $\underline{\underline{\mathbf{W}}} \in \mathbb{R}^{D \times V}$ stores target node embeddings. The hidden layer, $\underline{\mathbf{h}} \in \mathbb{R}^{D}$ is the embedding of target node ($\{n^t\}$).

$$\underline{\mathbf{h}} = \underline{\underline{\mathbf{W}}}\ \underline{\mathbf{n}}^t \tag{3.6}$$

Hidden to output layer weight matrix $\underline{\underline{\mathbf{C}}} \in \mathbb{R}^{D \times V}$ stores node embeddings . Hence output layer, $\underline{\mathbf{q}} \in \mathbb{R}^{V}$ represents the scores. q_i is n_i's score being a context word for the given target ($\{n^t\}$).

$$\underline{\mathbf{q}} = \underline{\underline{\mathbf{C}}}^T\ \underline{\mathbf{h}} \tag{3.7}$$

Traditional approach to use scores is to consider them as probabilities coming from a probability distribution. Thus CBOW computes probability of every word in vocabulary being the target word when context words are given, i.e. $P(n^t|\{n_k^c\})$. Similarly Skip-gram calculates probability of every word in vocabulary being one of context words when target word is given, i.e. $P(\{n_k^c\}|n^t)$. The probability values coming from a distribution must be normalized and non-negative. However these scores are not normalized and not necessarily non-negative, thus violate conditions of probability values. To convert scores into proper probabilities that come from a probability distribution, an additional layer is needed. The softmax layer after the output layer ensures that this condition is satisfied.

Definition 2. Softmax. Let $s(.)$ donate the softmax operation. $s(.)$ is defined on vectors as :

$$s(\underline{\mathbf{q}}) = \frac{\underline{\mathbf{q}}}{\sum_{k=1}^{V} \exp(q_k)} \tag{3.8}$$

An objective function can be defined based on softmax values and one-hot vector output. A simple approach is using $\underline{\mathbf{n}}^t - s(\underline{\mathbf{q}})$ for CBOW or $\underline{\mathbf{n}}_k^c - s(\underline{\mathbf{q}})$ for Skip-Gram. However computing gradients with respect to weight matrices in this loss function requires excessive computations. A cross-entorpy loss is introduced to compute the gradients of error much easier.

Definition 3. Cross-entropy Loss. Let $CE(.)$ donate the cross-entropy loss. $CE(\underline{\mathbf{q}}, \underline{\mathbf{n}}^t)$ is defined on vectors as :

$$CE(\underline{\mathbf{q}}, \underline{\mathbf{n}}^t) = -\sum_{k=1}^{V} n_k \log(q_k) \tag{3.9}$$

for CBOW while $\underline{\mathbf{n}}^t$ is replaced with $\underline{\mathbf{n}}_k^c$ for Skip-Gram and n_k corresponds to kth entry of input one-hot vector.

Cross-entropy loss can be written as $CE(\underline{\mathbf{q}}, \underline{\mathbf{n}}^t) = -\log(q_p)$ for target node p, since only the p^{th} entry of one-hot vector is non-zero, and equals to 1. When the cross-

entropy loss is used, objective of the CBOW and Skip-Gram are to maximize the log probabilities given below :

$$J_{cbow}(C) = \sum_{i=1}^{|C|} \log(P(n_i^t|n^c)) \tag{3.10}$$

$$J_{sg}(C) = \sum_{i=1}^{|C|} \sum_{-w\leq j\leq w, j\neq 0} \log(P(n_j^c|n_i^t)) \tag{3.11}$$

where C defines whole corpus and $|C|$ defines number of elements in the corpus.

With the softmax layer, embedding learning complexity is in the order of V. To speed up training process, two different objective functions are proposed : hierarchical softmax and negative sampling [54]. Hierarchical softmax is used in the original Deepwalk algorithm. Details of hierarchical softmax can be found in Appendix A.

3.3.2. Negative Sampling

Traditional softmax approach on score values can be formulated as

$$P(n^t|\{n_k^c\}) = \frac{\exp(q_{n^t})}{Z} \tag{3.12}$$

where q_{n^t} is the score of n^t, Z is the normalization factor. Hierarchical softmax avoids computing normalization factor by creating a binary tree. However structure of the constructed tree highly effects performance and creating an optimal binary tree is challenging. Another alternative approach proposed is negative sampling which will be used for all experiments in this thesis.

Negative sampling is highly influenced by the noise contrastive estimation (NCE) [55], proposed for training unnormalized probabilistic models. NCE also points out direct computation of normalization factor is expensive. It treats normalization factor as

another parameter to be learned in the model. NCE learns parameters while trying to discriminate observed samples and artificially generated noise samples by maximization of objective function based on log-likelihood of parameters.

Adapting NCE to language models, the learning problem of CBOW can posed as classification of n^t's as those drawn from $P(n^t|\{n_k^c\})$ (*positive samples*) and those that are drawn from a known noise distribution $P_0(n^t)$ (*negative samples*). It is also assumed that negative samples are α times more probable than positive samples, hence n^t comes from a joint distribution,$\frac{1}{\alpha+1}P(n^t|\{n_k^c\}) + \frac{\alpha}{\alpha+1}P_0(n^t)$. Mnih *et al.* [56] also reports that assuming $Z = 1$ in Equation 3.12 does not affect the performance, which they explain with the high degree-of-freedom of the neural model used. Further, assuming uniform distribution for $P_0(n^t)$ and setting the number of negative samples drawn per target node as $\alpha \in \mathbb{I}$, we get $\alpha P_0(n^t) = 1$. Adopting these simplifications, we have posterior probabilities for positive and negative samples as

$$P(I = 1|n^t, \{n_k^c\}) = \frac{\exp(q_{n^t})}{\exp(q_{n^t}) + \alpha P_0(n^t)} = \frac{1}{1 + \exp(-q_{n^t})} \tag{3.13}$$

$$P(I = 0|n^t, \{n_k^c\}) = \frac{\alpha P_0(n^t)}{\exp(q_{n^t}) + \alpha P_0(n^t)} = \frac{1}{1 + \exp(q_{n^t})} \tag{3.14}$$

where $I \in \{1, 0\}$ is a binary variable representing n^t's being a positive/negative sample. Hence, the objective function for training CBOW is defined as,

$$J(\underline{\underline{\mathbf{C}}}, \underline{\underline{\mathbf{W}}}) = \sum \Big[\log(P(I = 1|n^t, \{n_k^c\}) + \sum_{j=1}^{\alpha} P(n_j^-) \log(P(I = 0|n_j^-, \{n_k^c\})) \Big] \tag{3.15}$$

where the outer summation is over all $(n^t, \{n_k^c\})$ m-tuples ($m = \alpha + 1$) in the training set, n_j^- is the j^{th} negative sample associated with a given context, $P(n_j^-)$ is the probability of drawing that negative sample. We used modified unigram distribution for the prior $P(n_j^-)$, which is defined as

$$P(n_j^-) = \frac{\#(n_j)^\gamma}{\sum_{i=1}^{V} \#(n_i)^\gamma} \tag{3.16}$$

where $\#(n_i)$ is the number of occurrences of node n_i in the corpus and γ (negative sample exponential) is set to 0.75 in [54], where authors reported slightly better performance compared to unigram distribution. In this thesis different γ values will be experimented.

4. EXPERIMENTS AND RESULTS

The MRI data was acquired from 91 volunteers (46 male, 45 female, age=62 ± 10, 17 AD, 48 MCI, 26 SCI) with written consent in a single session using the Philips Achieva 3 T MRI system (Netherlands) with a 32-channel head coil. DWI volumes were acquired with $FOV = 200 \times 236mm^2$ at $2.27mm$ isotropic voxel size. 120 volumes were acquired at 6 shells in q-space using a single-shot, pulse-gradient spin echo (PGSE) EPI sequence with $TE/TR = 92ms/9032ms$. T1 weighted MRI (T1w) volumes were acquired via the 3D FFE (Fast Field Echo) pulse sequence with multi-shot TFE (Turbo Field Echo) imaging mode with $FOV = 220 \times 240mm^2$, $1.0mm^3$ isotropic voxels. AD was diagnosed if the subject had multiple cognitive deficits with functional impairment and a clinical dementia rating (CDR) scale score of at least 0.5. MCI was diagnosed if the subject scored a total free recall (TFR) < 28 or a cue index (CI) < 0.68 in Free and Cued Selective Reminding Test (FCSRT) and had a CDR score of 0.5. SCI was diagnosed if the subject scored > 27 in FTR-FCSRT or > 0.67 in CI-FCSRT and had a CDR score of 0.

Columns of $\underline{\underline{\mathbf{W}}}$, namely $\underline{\mathbf{w}}_i \in \mathbb{R}^D$, are nodal embeddings learned for each sNET (individual) independently and used as latent representations of local structure. Discriminative power of each nodal embeddings is assessed separately for AD/MCI, AD/SCI, MCI/SCI and AD/MCI/SCI classification tasks by following steps :

(i) Nodal embeddings are learned for each subject, resulting with 91 D-by-148 embedding matrices.
(ii) D dimensional embedding of a node i is taken from all subjects to form D-by-91 data matrix.
(iii) Resulting data is trained and tested by a Kernel-SVM using leave-one-out cross validation (Radial basis function is used as kernel function).
(iv) Accuracy is measured for node i.
(v) Above steps are repeated until accuracy for all 148 nodes is measured.

Pseudo-code for experiment pipeline can be seen at Fig 4.1. Kernel-SVM is used as classifier for all experiments. Kernel-SVM is a special type of SVM. To provide better insight into kernel-SVM, main idea behind SVM will be briefly introduced from scratch.

Input $\underline{\underline{\mathbf{W}}}^j$ for j=1,2,..91
$\underline{\underline{\mathbf{W}}}^j$ is embedding matrix of subject j
for $i = 1$ to 148 **do**
  Data matrix, $\underline{\underline{\mathbf{D}}} \in \mathbb{R}^{D\times 91}$ $\underline{\mathbf{d}}_j = \underline{\mathbf{w}}_i^j$
  RBF-SVM training and testing by leave-one-out cross validation
  Calculate accuracy for node i
end for

Figure 4.1. Experiment Pipeline.

SVM finds a separating hyperplane which maximizes margins. Margin is the distance between separating hyperplane (decision boundary) and the points that lie closest to separating hyperplane. These points are also called as support vectors and most difficult to classify. Since the decision boundary is determined by only the support vectors, solving does not require hard computations.

SVM is linear with respect to its weight vector when data is linearly separable and it is in the form of :

$$f(\underline{\underline{\mathbf{X}}}) = \underline{\mathbf{c}}^T \underline{\underline{\mathbf{X}}} + b \tag{4.1}$$

where $\underline{\mathbf{c}}$ weight vector, $\underline{\underline{\mathbf{X}}}$ is data matrix and $\underline{\mathbf{b}}$ is the bias vector. Weight vector $\underline{\mathbf{c}}$, can be found by solving the following optimization problem :

$$\begin{aligned} \underset{\underline{\mathbf{c}}}{\text{minimize}} \quad & \frac{1}{2}||\underline{\mathbf{c}}||^2 \\ \text{subject to} \quad & y_i(\underline{\mathbf{c}}^T \underline{\mathbf{x}}_i + \underline{\mathbf{b}}) \geq 1,\ i = 1, 2 \ldots, n. \end{aligned} \tag{4.2}$$

where $\underline{\mathbf{x}}_i$ i^{th} data point (assuming each vector from data matrix is a point in a space) and y_i label of this data point. This formulation assumes a binary classification task where the class labels are 1 or -1. The constraint given above is derived from these equations below :

$$\underline{\mathbf{c}}^T \underline{\mathbf{x}}_i + \underline{\mathbf{b}} \geq 1 \text{ when } y_i = 1 \tag{4.3}$$

$$\underline{\mathbf{c}}^T \underline{\mathbf{x}}_i + \underline{\mathbf{b}} \leq 1 \text{ when } y_i = -1 \tag{4.4}$$

Above equations are consequences of margin maximization. However Eq. 4.2 only tries to find max-margin classifier that perfectly separates data. When most data points are linearly separable but there are few points that can not be classified correctly, the optimization problem can be redefined as :

$$\begin{aligned} \underset{\underline{\mathbf{c}}}{\text{minimize}} \quad & \frac{1}{2}||\underline{\mathbf{c}}||^2 + R\sum_{i=1}^{n} \xi_i \\ \text{subject to} \quad & y_i(\underline{\mathbf{c}}^T \underline{\mathbf{x}}_i + \underline{\mathbf{b}}) \geq 1 - \xi_i,\ i = 1, 2 \ldots, n. \end{aligned} \tag{4.5}$$

ξ_i is the distance between the separating plane and the misclassified point. It softens the margin constraint by allowing some data points to be misclassified. R parameter controls the trade off between accuracy and margin size. This constrained optimization problem can be solved by Lagrange,

$$\begin{aligned} L(w, b) = \quad & \frac{1}{2}||\underline{\mathbf{c}}||^2 + R\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} a_i y_i(\underline{\mathbf{c}}^T \underline{\mathbf{x}}_i + \underline{\mathbf{b}}) + \sum_{i=1}^{n}(1 - \xi_i)a_i \\ & \frac{\partial L}{\partial c} = c - \sum_{i=1}^{n} a_i y_i \underline{\mathbf{x}}_i \\ & \frac{\partial L}{\partial b} = \sum_{i=1}^{n} a_i y_i \end{aligned} \tag{4.6}$$

Setting partial derivatives equal to zero, here is the solutions for the classifier parameters :

$$c = \sum_{i=1}^{n} a_i y_i \underline{\mathbf{x}}_i \tag{4.7}$$

$$\sum_{i=1}^{n} a_i y_i = 0 \tag{4.8}$$

rewriting Eq. 4.5 in terms of a_i and replacing c with Eq. 4.7 :

$$\begin{aligned} \underset{a_i}{\text{minimize}} \quad & \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_i \sum_j a_i a_j y_i y_j (\underline{\mathbf{x}}_i^T \underline{\mathbf{x}}_j) \\ \text{subject to} \quad & 0 \leq a_i \leq R \\ & \sum_{i=1}^{n} a_i y_i = 0 \end{aligned} \tag{4.9}$$

when $(\underline{\mathbf{x}}_i^T \underline{\mathbf{x}}_j)$ term is the inner product of data points. Dependence on w and b is removed and this problem can be solved by only using inner products.

When data is not linearly separable, introducing a slack variable ξ is not enough. Kernel-SVM addresses this issue and defines a feature map to project data into higher dimensional space so that the projected data points can be separated linearly. Kernel-SVM can be formulated as following :

$$f(\underline{\underline{\mathbf{X}}}) = \underline{\mathbf{c}}^T \phi(\underline{\underline{\mathbf{X}}}) + b \tag{4.10}$$

where $\phi(x)$ defines a feature map into higher dimensional space. Rewriting Eq. 4.9 now we need to solve :

$$\begin{aligned} \underset{a_i}{\text{minimize}} \quad & \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_i \sum_j a_i a_j y_i y_j (\phi(\underline{\mathbf{x}}_i)^T \phi(\underline{\mathbf{x}}_j)) \\ \text{subject to} \quad & 0 \leq a_i \leq R \\ & \sum_{i=1}^{n} a_i y_i = 0 \end{aligned} \tag{4.11}$$

Although a feature map is used in this formulation, it does not have to be known explicitly because it appears as in the form of inner product. Thus, calculating a kernel function is enough to solve problem.

Definition 4. Kernel function. Let $\phi(\underline{\mathbf{x}})$ donate a feature map. $k(\underline{\mathbf{x}}_i, \underline{\mathbf{x}}_j)$ is defined on vectors as:

$$k(x_i, x_j) = \phi(\underline{\mathbf{x}}_i)^T \phi(\underline{\mathbf{x}}_j) \tag{4.12}$$

There are many kernel functions such as linear kernel, polynomial kernel,sigmoid kernel etc.. Among them, radial basis function (Gaussian kernel) is used in this thesis.

Definition 5. Radial basis function. Let $k_{rbf}(\underline{\mathbf{x}}_i, \underline{\mathbf{x}}_j)$ donate a radial basis function. $k_{rbf}(\underline{\mathbf{x}}_i, \underline{\mathbf{x}}_j)$ is defined on vectors and performs :

$$k_{rbf}(\underline{\mathbf{x}}_i, \underline{\mathbf{x}}_j) = \exp(\gamma ||\underline{\mathbf{x}}_i - \underline{\mathbf{x}}_j||^2) \tag{4.13}$$

A kernel function simply computes the inner product of two projected vectors. Radial basis function implies an infinite dimensional feature map which can be proven by using Taylor expansion of $exp(x)$.

SVM classifier is capable of finding complex decision boundaries by using kernel trick. A trained neural network can also deal with the hard classification problems however considering classification is performed for each 148 node with different model parameters, SVM is advantageous over NN in terms of the training time.

RBF-SVMs are trained and tested by leave-one-out cross validation in this thesis. In cross validation, data is separated into subsets then, the classifier trained on the remaining subsets is tested in each subset and accuracy is measured. A simple k-fold cross validation splits data into k groups. Leave-one-out cross validation is a specific version of k-fold cross validation when k is equal to the number of data samples.

Since there are limited samples in our dataset, leave-one-out cross validation is more preferable then k-fold cross validation and a simple train/test split. Leave-one-out cross validation also results in less biased estimation of how well the classifier is performing compared to train/test split method.

Accuracy is considered as performance metric throughout all experiments. In this thesis, different parameters are experimented such as network architecture choice, number of walks starting at each node, negative sampling exponential etc.. Details of each experiments can be seen at Table 4.1. Tables including top performing nodes and circular graphs that shows all nodes' performance are given at Appendix B.

Table 4.1. Parameters of experiments.

Experiment	**Network Architecture**	**L**	**K**	**w**	γ	**k**
TEST 1	CBOW	40	7	5	0.75	2
TEST 2	CBOW	40	7	5	0	2
TEST 3	CBOW	40	7	5	1	2
TEST 4	CBOW	40	5	5	0.75	2
TEST 5	CBOW	40	5	5	0	2
TEST 6	CBOW	40	5	5	1	2
TEST 7	CBOW	40	10	5	0.75	2
TEST 8	CBOW	40	10	5	0	2
TEST 9	CBOW	40	10	5	1	2
TEST 10	CBOW	40	7	5	0.75	5
TEST 11	CBOW	40	7	10	0.75	2
TEST 12	CBOW	40	20	5	0.75	2
TEST 13	CBOW	40	50	5	0.75	2
TEST 14	Skip-Gram	40	7	5	0.75	2
TEST 15	Skip-Gram	40	7	5	0.75	5
TEST 16	Skip-Gram	40	7	10	0.75	2
TEST 17	Skip-Gram	40	5	5	0.75	2
TEST 18	Skip-Gram	40	10	5	0.75	2
TEST 19	Skip-Gram	40	20	5	0.75	2
TEST 20	Skip-Gram	40	50	5	0.75	2

Among these experiments, TEST 1 is chosen as the best performing experiment in terms of classification accuracy. Top performing nodes for TEST 1 are given at

Table 4.2. In addition, a circular graph is used to visualize accuracy of each node for TEST 1 at Figure 4.2.

Table 4.2. Classification accuracies of top performing nodes in TEST 1.

#	R/L	Node Name	AD-SCI	AD-MCI	MCI-SCI	All
95	R	Lateral occipito-temporal gyrus	**0.81**	0.66	0.60	0.41
66	L	Pericallosal sulcus	**0.77**	0.46	0.66	0.43
133	R	Ant. occipital sulcus & preoccipital notch	**0.77**	**0.75**	0.53	0.40
76	R	Inferior occipital gyrus (O3) and sulcus	**0.93**	**0.75**	0.60	**0.48**
106	R	Subcallosal area, subcallosal gyrus	**0.77**	**0.82**	0.47	0.39
146	R	Inferior temporal sulcus	0.60	**0.80**	0.50	0.46
124	R	Anterior transverse collateral sulcus	0.74	**0.77**	0.55	**0.51**
13	L	Orbital part of the inferior frontal gyrus	0.28	0.54	**0.72**	0.44
2	L	Inferior occipital gyrus (O3) and sulcus	0.51	0.62	**0.70**	0.42
44	L	Calcarine sulcus	0.56	0.57	**0.70**	0.39
39	L	Horiz. ramus of ant. seg. of lateral sulcus	0.58	0.48	**0.70**	0.39
121	R	Ant. seg. of the circular sulcus of insula	0.72	0.65	**0.69**	**0.52**
88	R	Triangular part of the inferior frontal gyrus	0.63	0.71	0.622	**0.51**
56	L	Intraparietal sulcus	0.2	0.62	0.58	**0.51**
		Mean±Std (Top 5 nodes)	**0.81±0.07**	**0.78±0.03**	**0.70±0.01**	**0.51±0.01**

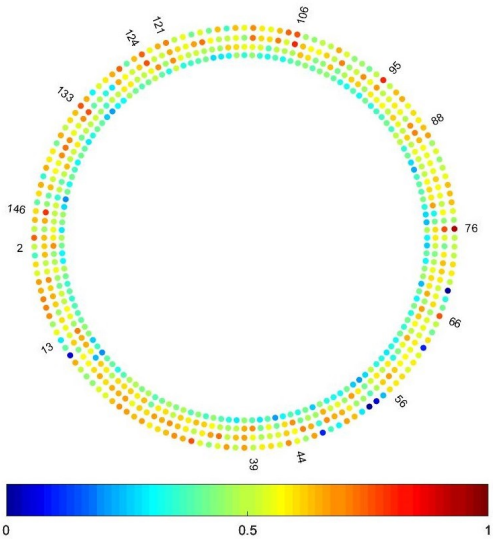

Figure 4.2. Classification accuracies for all nodes (from outer to inner : AD/SCI-AD/MCI-MCI/SCI-AD/MCI/SCI)

To evaluate the performance of nodal embeddings, a comparison with different methods would be useful. One alternative is taking the concatenation of some nodal embeddings instead embedding of a single node. To choose which nodes to concatenate, subnetworks of brain can be used. In this theses, nodes that belong to same subnetwork are concatenated. Parameters of concatenation experiment are the same with TEST 1. In concatenation experiment, RBF-SVMSs are trained ad tested by leave-one-out cross validation. Results of this experiment can be seen at Table 4.3.

Table 4.3. Concatenation Experiment.

Subnetwork Name	**AD-SCI**	**AD-MCI**	**MCI-SCI**	**All**
Visual	0.74	0.63	0.55	0.37
Somatosensory and Auditory	0.37	0.57	0.54	0.32
Dorsal attention	0.58	0.65	0.69	0.42
Salience	0.66	0.65	0.43	0.35
Limbic	0.70	0.63	0.66	0.45
Fronto-parietal	0.74	0.70	0.54	0.46
Default mode	0.53	0.66	0.55	0.44

Another comparison can be done between nodal embeddings and direct use of node adjacency matrix. Node adjacency matrix only captures information about first order proximity while nodal embeddings contain additional information since embedding learning maximizes the cooccurence of the target node with the surrounding context nodes hence leverages higher order proximity. It is expected that prosed method in this thesis learns the classification problems better compared to direct use of node adjacency matrix. In this experiment, RBF-SVMSs are trained ad tested by leave-one-out cross validation. Nodal features are the rows of node adjacency matrix. Top performing nodes for this experiment are given at Table 4.4. Also, a circular graph is used to visualize accuracies.

Table 4.4. Classification accuracies of top performing adjacency matrix nodes.

#	R/L	Node Name	AD-SCI	AD-MCI	MCI-SCI	All
53	L	Middle frontal sulcus	**0.72**	**0.79**	0.52	0.05
8	L	Mid.-post. part of the cingulate gyrus and sulcus	**0.72**	0.79	**0.70**	0.48
84	R	Posterior-ventral part of the cingulate gyrus	**0.70**	**0.80**	**0.69**	**0.58**
80	R	Anterior part of the cingulate gyrus and sulcus	**0.70**	**0.79**	0.62	0.53
49	L	Superior segment of the circular sulcus of the insula	**0.70**	**0.80**	0.68	0.06
48	L	Inferior segment of the circular sulcus of the insula	0.63	**0.79**	0.61	0.52
90	R	Superior frontal gyrus	0.49	0.62	**0.72**	**0.57**
120	R	Marginal branch of the cingulate sulcus	0.00	0.00	**0.70**	0.03
105	R	Straight gyrus	0.56	0.69	**0.70**	0.50
96	R	Lingual gyrus	0.63	0.71	0.61	**0.60**
82	R	Middle-posterior part of the cingulate gyrus and sulcus	0.00	0.75	0.65	**0.58**
136	R	Lateral orbital sulcus	0.00	0.72	0.01	**0.57**
		Mean±Std (Top 5 nodes)	**0.71±0.01**	**0.79±0.01**	**0.70±0.01**	**0.58±0.01**

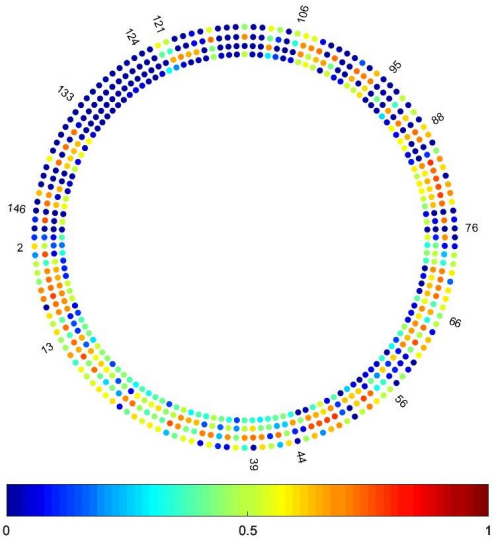

Figure 4.3. Classification accuracies for all nodes (from outer to inner : AD/SCI-AD/MCI-MCI/SCI-AD/MCI/SCI).

5. DISCUSSION

Neural network language models learn the relation between target words and words that occurs in their vicinity, context words. sNET nodes are analogous to words, hence nodal embeddings learn the relation between target node and context nodes i.e. encoding local connectivity patterns. Changes in local connectivity patterns for different sNETs can be captured by nodal embeddings since nodes with similar local connections are mapped to similar embeddings.

Experiments show that nodal embeddings may also capture information about disease progression since the discriminative power is higher for AD/SCI than AD/MCI and MCI/SCI. Distinguishing MCI patients is harder compared to those with AD and SCI, since MCI might be considered as a stage between AD and SCI. However not all MCIs evolve to AD. Also there is a shift in the top performing nodes as the disease progresses, which can be an indicator of spatial progression of the disease. Furthermore, no clear symmetry across the cortical regions/nodes in terms of their power is observed, in agreement with asymmetric changes in brain previously reported in AD literature.

By looking nodes' membership in subnetworks, it is possible to make basic comments on which cognitive functions are linked to top performing nodes and which subnetworks AD targets at. It is believed that human brain consists of subnetworks which all perform a specific cognitive function. A popular approach categorizes brain into seven subnetworks : visual, default mode, fronto-parietal, dorsal attention, somatosensory and auditory (Som Aud), limbic and salience subnetworks. In TEST 1, three of the most discriminative nodes for the AD/SCI classification (n_{95}, n_{76}, n_{133}) belong to visual network which was observed as an early feature of cognitive impairments in AD [57]. Further analysis on top performing nodes requires specialist opinions.

Nodal embeddings are directly related to parameters of the proposed method. Changing the parameters may result in different classification performance of different nodes. Parameters of the proposed method are either random walk related or neural

network related parameters. Random walk related parameters determine corpus size while neural network related parameters affect embedding learning process. Many parameters are experimented independently, i.e. corpus size changed while neural network parameters are kept same or neural network parameters are changed while the corpus is the same. It is observed that proposed method is more sensitive to corpus size and neural network architecture compared to other parameters.

Although CBOW architecture outperforms Skip-gram architecture in our classification tasks, Skip-gram architecture is reported as more successful at syntactic and semantic tasks for word embeddings. Mikolov *et al.* [54] does not give any clear justification of why Skip-gram is more successful than CBOW. A possible explanation for this outcome is that CBOW takes the mean of the context words, i.e. smooths the distributional information while Skip-gram treats each context word as an output to be predicted. Thus rare words are represented better in Skip-gram when a large vocabulary is available since Skip-gram takes more training samples than CBOW. However when the vocabulary size is limited which is the case for this thesis ($V = 148$), smoothing context words would be more helpful .

Corpus size is only dependent on L and K since vocabulary size is fixed. To experiment different corpus size, only the number of walks starting from each node (K) is changed when random walk length (L) is kept same. With different experiments, it is observed that increasing corpus size degrades the performance. In CBOW, classification accuracies do not decrease for classification tasks including MCI patients with the increasing corpus size, however sensitivity decreases dramatically. In addition, most of the nodes give nearly same classification performance. It is caused by unbalanced class ratio of MCI (48 of 91 patients are labeled as MCI). Classifier simply decides most subjects as MCI, even though class weight ratios are introduced in RBF-SVM classifier. This indicates that the model is not learning properly. In Skip-Gram, both classification accuracies and sensitivity are decreased. This performance drop is caused by limited vocabulary size. Finer cortical parcellations, i.e. larger vocabulary, may result in better classification performance.

Other parameters such as window length (w), negative sample exponent (γ) and number of negative samples (k) are experimented through different tests. Their impact were lower compared to K and network architecture. w parameter determines the degree of the local connectivities is captured in nodal embeddings. To exemplify, setting it equal to 1 results in nodal embeddings that encode node-adjacency matrix. To represent subnetworks around each node, w must be set greater than 1. In most experiments, $w = 5$ is used. Bigger w values also decrease performance since changes in local connectivity patterns become harder to be observed.

γ parameter is set empirically. Mikolov *et al.* [54], suggests that setting γ to 0.75 slightly outperforms compared to $\gamma = 1$. The latter corresponds to unigram distribution, which is the word frequency. Values lower than 1 sample rare words more frequently which improves quality of negative samples.

Different k values result in similar accuracies. Choice of $k = 2$ is better than $k = 5$ by a small margin, since a small set of negative samples are enough to distinguish target word from noise samples.

D, embedding dimension is closely related to the data size (91). Although it is not reported in this thesis, increasing D would improve RBF-SVM training performance however test accuracy would decrease which indicates poorer generalizability.

In the direct use of node adjacency rows, top accuracies for classification tasks including MCI patients seem better than nodal embedding accuracies. However the use of node adjacency rows lacks specificity, they simply do not learn the classification problem instead they label most of patients to numerically dominant group. It means that node adjacency rows do not contain any relevant information about these classification tasks. In the concatenation experiment, fronto-parietal, visual and limbic subnetworks gives meaningful results, however nodal embeddings outperforms concatenation of nodal embeddings. Since a subnetwork consists of around 20 nodes, resulting feature dimension is too big compared to data size and this is the cause of performance decrease.

Most discriminative nodes can be used to classify patients, i.e. they may be useful for diagnosis. Apart from diagnosis, nodal embeddings might be helpful for monitoring disease progression. To visualize relation between nodal embeddings and the disease progression, Laplacian Eigenmaps is used to learn a manifold where the disease progression can be represented in embedding space. Laplacian Eigenmaps is also used to reduce dimensionality of 8 dimensional nodal embeddings.

Laplacian Eigenmaps is a non-linear dimensionality reduction technique that aims to find mapping from a weighted graph formed by feature vectors $(\underline{\mathbf{x}}^i, \underline{\mathbf{x}}^2, ...\underline{\mathbf{x}}^n) \in R^M$ to embeddings $(\underline{\mathbf{y}}^1, \underline{\mathbf{y}}^2, ...\underline{\mathbf{y}}^n) \in R^m$ $(m \leq M)$ lying on a manifold embedded in R^M. Objective function of this mapping is the following:

$$\underset{y}{\text{minimize}} \quad \sum_{i,j} (\underline{\mathbf{y}}^i - \underline{\mathbf{y}}^j)^2 W_{ij} \tag{5.1}$$

As mentioned in Chapter 2, Laplacian Eigenmaps finds similar embeddings when nodes are heavily connected. There are possible approaches to create node adjacency matrix. One possible approach is finding closer points in terms of euclidean distance i.e. ith node and$j$th node is connected when $||\underline{\mathbf{x}}^i - \underline{\mathbf{x}}^j||^2 < \epsilon$ and choosing weights as either output of a function, $W_{ij} = g(\underline{\mathbf{x}}^i, \underline{\mathbf{x}}^j)$ or simple binary choice, $W_{ij} = 1$ if there is connection and $W_{ij} = 0$ if there is no connection. In this thesis, it is considered that all nodes are connected and weights are calculated as following :

$$W_{ij} = \exp(\frac{||\underline{\mathbf{x}}^i - \underline{\mathbf{x}}^j||^2}{8}) \tag{5.2}$$

Eq. 5.1 can be manipulated as :

$$\sum_{i,j} (\underline{\mathbf{y}}^i - \underline{\mathbf{y}}^j)^2 W_{ij} = \underline{\mathbf{y}}^T \underline{\underline{\mathcal{L}}}\, \underline{\mathbf{y}} \tag{5.3}$$

where $\underline{\underline{\mathcal{L}}}$ is Laplacian matrix of a graph, $\underline{\underline{\mathcal{L}}} = \underline{\underline{\mathcal{D}}} - \underline{\underline{\mathcal{W}}}$ and $\underline{\underline{\mathcal{D}}} = \sum_j W_{ij}$. Thus minimization problem turns into :

$$\begin{aligned} \underset{y}{\text{minimize}} \quad & \sum_{i,j} (\underline{\mathbf{y}}^i - \underline{\mathbf{y}}^j)^2 W_{ij} \\ \text{subject to} \quad & \underline{\mathbf{y}}^T \underline{\underline{\mathcal{D}}}\, \underline{\mathbf{y}} = 1 \end{aligned} \tag{5.4}$$

where the constraint is given to remove scaling factor for embeddings. Solution to this objective function is the minimum of generalized eigenvalue problem :

$$\underline{\underline{\mathcal{L}}}\, \underline{\mathbf{y}} = \lambda \underline{\underline{\mathcal{D}}}\, \underline{\mathbf{y}} \tag{5.5}$$

however minimum eigenvalue is 0 for Laplacian matrix which corresponds to constant eigenvector that takes 1 for every node. By eliminating this trivial solution, m dimensional embeddings are in the form of :

$$\underline{\mathbf{y}}^i = (v_i^1, v_i^2, ...v_i^m) \tag{5.6}$$

where v_i^m is ith entry of mth smallest eigenvector that is a solution to Eq. 5.5.

Two dimensional embeddings from Laplacian Eigenamps are extracted to learn a manifold where disease progression can be observed. Initial results are not satisfactory since embeddings of MCI and SCI subjects are overlapping too much. However embeddings of some AD subjects can be distinguished from those with MCI and SCI as shown in Fig 5.1.

Two dimensional embeddings in Fig 5.1 are extracted from nodal embeddings of node 124 (Right, Anterior transverse collateral sulcus), since the ratio of between interclass distances to intraclass distances is maximum for node 124. Although clusters are not clearly distinguished in this case, another mapping for nodal embeddings to

lower dimensions may help monitoring disease progression.

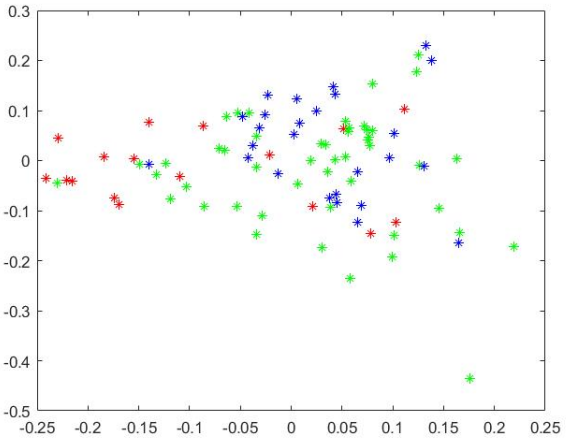

Figure 5.1. Two dimensional Laplacian Eigenmaps for Node 124 (Red-AD, Green-MCI, Blue-SCI).

6. CONCLUSION

Latent representations of nodes learned by graph embeddings preserve local network structure. Changes in local network structure may explain the nature of AD. Analysing subnetworks associated with the most discriminative nodes shows which cognitive functions are targeted at respectively, thus helps monitoring disease. It also allows to identify early risk factors for AD. In addition a node with high accuracy and sensitivity can be directly used for diagnosing AD. To sum up, this thesis offers a novel approach to understand, diagnose and monitor AD.

The preliminary results are promising. To yield better results, further improvements can be made. As in all machine learning applications, larger cohorts would provide benefit to proposed method. Increasing cortical parcellation resolution would give more accurate results due to finer anatomical localizations and the fact that neural network language models perform better with larger vocabulary size.

To create larger cohorts, more subjects are needed. Change in cortical parcellation requires a new anatomical atlas and recreating sNETs from the beginning. Collecting more data and processing them to obtain new networks are not dependent to proposed method. Improvements related to proposed method may include changes in creating random walks, using fNETs of subjects and forming new features from nodal embeddings.

Changes in creating random walk can be done by creating biased random walks as it mentioned in node2vec [35]. By introducing new variables, random walks can be manipulated to capture additional information of local structure.

fNETs of subjects capture information about correlation of BOLD signals. Since most of fNETs are dense matrices, nodal embeddings coming from fNETs is not able to capture any information about local structure by means of random walks. However functional similarities along with structural connections would be utilized to create a

new corpus in order to improve results.

Nodal embeddings can be used to form new features which may show progression of disease. In Chapter 5, Laplacian Eigenmaps is proposed to learn manifold where the disease progression can be monitored. However results are not successful enough, so an alternative mapping function must be investigated. To find such a function, protein levels, age and sex of a subject can be used apart from clinical label.

REFERENCES

1. Adams, J., *Cambridge National Level 1/2 Health and Social Care*, Hodder Education, 2018.

2. Alzheimer's Association, "2018 Alzheimers disease facts and figures", *Alzheimers & Dementia*, Vol. 14, No. 3, p. 367–429, 2018.

3. Reitz, C. and R. Mayeux, "Alzheimer disease: Epidemiology, diagnostic criteria, risk factors and biomarkers", *Biochemical Pharmacology*, Vol. 88, No. 4, p. 640–651, 2014.

4. Humpel, C., "Identifying and validating biomarkers for Alzheimers disease", *Trends in Biotechnology*, Vol. 29, No. 1, p. 26–32, 2011.

5. Sporns, O., "The human connectome: a complex network", *Annals of the New York Academy of Sciences*, Vol. 1224, No. 1, p. 109–125, 2011.

6. Qi, S., S. Meesters, K. Nicolay, B. M. T. H. Romeny and P. Ossenblok, "The influence of construction methodology on structural brain network measures: A review", *Journal of Neuroscience Methods*, Vol. 253, p. 170–182, 2015.

7. Bullmore, E. and O. Sporns, "Erratum: Complex brain networks: graph theoretical analysis of structural and functional systems", *Nature Reviews Neuroscience*, Vol. 10, No. 4, p. 312–312, 2009.

8. Sporns, O., G. Tononi and R. Kötter, "The Human Connectome: A Structural Description of the Human Brain", *PLoS Computational Biology*, Vol. 1, No. 4, 2005.

9. Fei, F., B. Jie and D. Zhang, "Frequent and Discriminative Subnetwork Mining for Mild Cognitive Impairment Classification", *Brain Connectivity*, Vol. 4, No. 5,

p. 347–360, 2014.

10. Chen, G., B. D. Ward, C. Xie, W. Li, Z. Wu, J. L. Jones, M. Franczak, P. Antuono and S.-J. Li, "Classification of Alzheimer Disease, Mild Cognitive Impairment, and Normal Cognitive Status with Large-Scale Network Analysis Based on Resting-State Functional MR Imaging", *Radiology*, Vol. 259, No. 1, p. 213–221, 2011.

11. Dai, Z., C. Yan, Z. Wang, J. Wang, M. Xia, K. Li and Y. He, "Discriminative analysis of early Alzheimers disease using multi-modal imaging and multi-level characterization with multi-classifier (M3)", *NeuroImage*, Vol. 59, No. 3, p. 2187–2195, 2012.

12. Dipasquale, O., L. Griffanti, M. Clerici, R. Nemni, G. Baselli and F. Baglio, "High-Dimensional ICA Analysis Detects Within-Network Functional Connectivity Damage of Default-Mode and Sensory-Motor Networks in Alzheimer's Disease", *Frontiers in Human Neuroscience*, Vol. 9, 2015.

13. Supekar, K., V. Menon, D. Rubin, M. Musen and M. D. Greicius, "Network Analysis of Intrinsic Functional Brain Connectivity in Alzheimers Disease", *PLoS Computational Biology*, Vol. 4, No. 6, 2008.

14. Khazaee, A., A. Ebrahimzadeh and A. Babajani-Feremi, "Identifying patients with Alzheimer's disease using resting-state fMRI and graph theory", *Clinical Neurophysiology*, Vol. 126, No. 11, p. 2132–2141, 2015.

15. Wee, C.-Y., P.-T. Yap, J. N. Brownyke, G. G. Potter, D. C. Steffens, K. Welsh-Bohmer, L. Wang and D. Shen, "Accurate Identification of MCI Patients via Enriched White-Matter Connectivity Network", *Machine Learning in Medical Imaging Lecture Notes in Computer Science*, p. 140–147, 2010.

16. Wee, C.-Y., P.-T. Yap, D. Zhang, K. Denny, J. N. Browndyke, G. G. Potter, K. A. Welsh-Bohmer, L. Wang and D. Shen, "Identification of MCI individuals using structural and functional connectivity networks", *NeuroImage*, Vol. 59, No. 3, p.

2045–2056, 2012.

17. Prasad, G., S. H. Joshi, T. M. Nir, A. W. Toga and P. M. Thompson, "Brain connectivity and novel network measures for Alzheimers disease classification", *Neurobiology of Aging*, Vol. 36, 2015.

18. Dragomir, A., A. G. Vrahatis and A. Bezerianos, "A Network-Based Perspective in Alzheimers Disease: Current State and an Integrative Framework", *IEEE Journal of Biomedical and Health Informatics*, Vol. 23, No. 1, p. 14–25, 2019.

19. Power, J. D., A. L. Cohen, S. M. Nelson, G. S. Wig, K. A. Barnes, J. A. Church, A. C. Vogel, T. O. Laumann, F. M. Miezin, B. L. Schlaggar and et al., "Functional Network Organization of the Human Brain", *Neuron*, Vol. 72, No. 4, p. 665–678, 2011.

20. Newman, M. E. J. and M. Girvan, "Finding and evaluating community structure in networks", *Physical Review E*, Vol. 69, No. 2, 2004.

21. Chen, G., H.-Y. Zhang, C. Xie, G. Chen, Z.-J. Zhang, G.-J. Teng and S.-J. Li, "Modular reorganization of brain resting state networks and its independent validation in Alzheimers disease patients", *Frontiers in Human Neuroscience*, Vol. 7, 2013.

22. Dai, Z., C. Yan, K. Li, Z. Wang, J. Wang, M. Cao, Q. Lin, N. Shu, M. Xia, Y. Bi and et al., "Identifying and Mapping Connectivity Patterns of Brain Network Hubs in Alzheimers Disease", *Cerebral Cortex*, Vol. 25, No. 10, p. 3723–3742, 2014.

23. Sun, Y., Q. Yin, R. Fang, X. Yan, Y. Wang, A. Bezerianos, H. Tang, F. Miao and J. Sun, "Disrupted Functional Brain Connectivity and Its Association to Structural Connectivity in Amnestic Mild Cognitive Impairment and Alzheimer's Disease", *PLoS ONE*, Vol. 9, No. 5, 2014.

24. Goyal, P. and E. Ferrara, "Graph embedding techniques, applications, and perfor-

mance: A survey", *Knowledge-Based Systems*, Vol. 151, p. 78–94, 2018.

25. Liben-Nowell, D. and J. Kleinberg, "The link prediction problem for social networks", *Proceedings of the twelfth international conference on Information and knowledge management - CIKM 03*, 2003.

26. Kipf, T. N. and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks", *International Conference on Learning Representations*, ICLR 2017, 2017, `https://openreview.net/pdf?id=SJU4ayYgl`.

27. Agrawal, M., M. Zitnik and J. Leskovec, "Large-Scale Analysis of Disease Pathways in the Human Interactome", *Pacific Symposium on Biocomputing 2018Pacific Symposium on Biocomputing 2018*, 2018.

28. Ding, C., X. He, H. Zha, M. Gu and H. Simon, "A min-max cut algorithm for graph partitioning and data clustering", *Proceedings 2001 IEEE International Conference on Data Mining*.

29. Belkin, M. and P. Niyogi, "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation", *Neural Computation*, Vol. 15, No. 6, p. 1373–1396, 2003.

30. Roweis, S. T., "Nonlinear Dimensionality Reduction by Locally Linear Embedding", *Science*, Vol. 290, No. 5500, p. 2323–2326, 2000.

31. Ahmed, A., N. Shervashidze, S. Narayanamurthy, V. Josifovski and A. J. Smola, "Distributed large-scale natural graph factorization", *Proceedings of the 22nd international conference on World Wide Web - WWW 13*, 2013.

32. Ou, M., P. Cui, J. Pei, Z. Zhang and W. Zhu, "Asymmetric Transitivity Preserving Graph Embedding", *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 16*, 2016.

33. Cao, S., W. Lu and Q. Xu, "GraRep", *Proceedings of the 24th ACM International*

on Conference on Information and Knowledge Management - CIKM 15, 2015.

34. Perozzi, B., R. Al-Rfou and S. Skiena, "DeepWalk", *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 14*, 2014.

35. Grover, A. and J. Leskovec, "node2vec", *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 16*, 2016.

36. L. Hamilton, W., R. Ying and J. Leskovec, "Representation learning on graphs:methods and applications", *Bulletin of the Technical Committee on Data Engineering*, Vol. 40, No. 3.

37. Wang, D., P. Cui and W. Zhu, "Structural Deep Network Embedding", *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 16*, 2016.

38. Cao, S., W. Lu and Q. Xu, "Deep Neural Networks for Learning Graph Representations", *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pp. 1145–1152, AAAI Press, 2016, `http://dl.acm.org/citation.cfm?id=3015812.3015982`.

39. Church, K. W. and P. Hanks, "Word association norms, mutual information, and lexicography", *Proceedings of the 27th annual meeting on Association for Computational Linguistics* -, 1989.

40. Bullinaria, J. A. and J. P. Levy, "Extracting semantic representations from word co-occurrence statistics: A computational study", *Behavior Research Methods*, Vol. 39, No. 3, p. 510–526, 2007.

41. Krizhevsky, A., I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks", *Communications of the ACM*, Vol. 60, No. 6, p.

84–90, 2017.

42. Le, Q. V., W. Y. Zou, S. Y. Yeung and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis", *Cvpr 2011*, 2011.

43. Chung, F. R. K., *Spectral graph theory*, Published for the Conference Board of the mathematical sciences by the American Mathematical Society, 2009.

44. Shuman, D. I., S. K. Narang, P. Frossard, A. Ortega and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains", *IEEE Signal Processing Magazine*, Vol. 30, No. 3, p. 83–98, 2013.

45. Defferrard, M., X. Bresson and P. Vandergheynst, "Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering", *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 3844–3852, Curran Associates Inc., USA, 2016, `http://dl.acm.org/citation.cfm?id=3157382.3157527`.

46. Parisot, S., S. I. Ktena, E. Ferrante, M. Lee, R. Guerrero, B. Glocker and D. Rueckert, "Disease prediction using graph convolutional networks: Application to Autism Spectrum Disorder and Alzheimer's disease", *Medical Image Analysis*, Vol. 48, p. 117–130, 2018.

47. Analysis Group, *FMRIB Software Library 6.0*, `https://fsl.fmrib.ox.ac.uk`, accessed at April 2019.

48. Laboratory for Computational Neuroimaging at the Athinoula A. Martinos Center for Biomedical Imaging, *Free Surfer*, `https://surfer.nmr.mgh.harvard.edu`, accessed at April 2019.

49. C. Pierpaoli, L. Walker, M. O. Irfanoglu, A. Barnett, P. Basser, L-C.

Chang, C. Koay, S. Pajevic, G. Rohde, J. Sarlls, and M. Wu, *TORTOISE: an integrated software package for processing of diffusion MRI data*, `https://tortoise.nibib.nih.gov`, accessed at April 2019.

50. Tench, C., P. Morgan, M. Wilson and L. Blumhardt, "White matter mapping using diffusion tensor MRI", *Magnetic Resonance in Medicine*, Vol. 47, No. 5, p. 967–972, 2002.

51. Moyer, D., B. A. Gutman, J. Faskowitz, N. Jahanshad and P. M. Thompson, "Continuous representations of brain connectivity using spatial point processes", *Medical Image Analysis*, Vol. 41, p. 32–39, 2017.

52. Bengio, Y., R. Ducharme, P. Vincent and C. Janvin, "A Neural Probabilistic Language Model", *J. Mach. Learn. Res.*, Vol. 3, pp. 1137–1155, Mar. 2003, `http://dl.acm.org/citation.cfm?id=944919.944966`.

53. Mikolov, T., K. Chen, G. Corrado and J. Dean, "Efficient Estimation of Word Representations in Vector Space", *Workshop Proceedings of International Conference of Learning Representation - ICLR 2013*, 2013.

54. Mikolov, T., I. Sutskever, K. Chen, G. Corrado and J. Dean, "Distributed Representations of Words and Phrases and Their Compositionality", *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pp. 3111–3119, Curran Associates Inc., USA, 2013, `http://dl.acm.org/citation.cfm?id=2999792.2999959`.

55. Gutmann, M. U. and A. Hyvarinen, "Noise-contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics", *J. Mach. Learn. Res.*, Vol. 13, No. 1, pp. 307–361, Feb. 2012, `http://dl.acm.org/citation.cfm?id=2503308.2188396`.

56. Mnih, A. and Y. W. Teh, "A Fast and Simple Algorithm for Training Neural Probabilistic Language Models", *Proceedings of the 29th International Coference*

on International Conference on Machine Learning, ICML'12, pp. 419–426, Omnipress, USA, 2012, http://dl.acm.org/citation.cfm?id=3042573.3042630.

57. Lavallée, M. M., D. Gandini, I. Rouleau, G. T. Vallet, M. Joannette, M.-J. Kergoat, T. Busigny, B. Rossion and S. Joubert, "A Qualitative Impairment in Face Perception in Alzheimer's Disease: Evidence from a Reduced Face Inversion Effect", *Journal of Alzheimers Disease*, Vol. 51, No. 4, p. 1225–1236, 2016.

APPENDIX A: HIERARCHICAL SOFTMAX

Hierarchical softmax creates a binary tree to factor out probabilities and assigns probabilities to path of the binary tree. In Deepwalk, Huffman coding is used to generate a binary tree. Huffman coding assigns short paths to frequent words. This method reduces the complexity around order of $\log(V)$. Formulation of hierarchical softmax for Skip-Gram is :

$$P(n_k^c|n^t) = -\sum_{i=1}^{L(n_k^c)-1} \sigma(I(node(n_k^c, i+1), child(node(n_k^c, i))\underline{\mathbf{v}}^T h) \tag{A.1}$$

$$I(x,y) = \begin{cases} 1, & \text{if x=y.} \\ -1, & \text{otherwise.} \end{cases} \tag{A.2}$$

$$h = \underline{\underline{\mathbf{W}}}\ \underline{\mathbf{n}}^t \tag{A.3}$$

where $node(n_k^c, i)$ is the ith node on the path from the root to node n_k^c and $L(n_k^c)$ is the length of the path going the node n_k^c. Lets assume $node(n_k^c, i)$ corresponds to pth node , so $child(node(n_k^c, i))$ is the two children of node p , $\underline{\mathbf{v}}$ is the pth column of $\underline{\underline{\mathbf{C}}}$ and $\sigma()$ is sigmoid function.

Definition 6. Sigmoid function. Let $\sigma()$ donate the sigmoid function. $\sigma(x)$ is defined on scalars and performs :

$$\sigma(x) = \frac{1}{1+\exp(-x)} \tag{A.4}$$

APPENDIX B: EXPERIMENTAL RESULTS

TEST 2 Parameters : i) CBOW ii) K=40 iii) L=7 iv) w=5 v) k=2 vi) $\alpha = 0$

Table B.1. Classification accuracies of top performing nodes in TEST 2.

#	R/L	Node Name	AD-SCI	AD-MCI	MCI-SCI	All
133	R	Ant. occipital sulcus & preoccipital notch	**0.77**	**0.80**	0.43	0.31
110	R	Temporal plane of the superior temporal gyrus	**0.77**	0.52	0.55	0.36
124	R	Anterior transverse collateral sulcus	**0.74**	**0.74**	0.49	0.41
76	R	Inferior occipital gyrus (O3) and sulcus	**0.74**	0.65	0.58	0.30
23	L	Parahippocampal gyrus	**0.74**	0.63	0.60	0.40
146	R	Inferior temporal sulcus	0.51	**0.82**	0.35	0.34
16	L	Superior frontal gyrus (F1)	0.63	**0.75**	0.55	0.36
147	R	Superior temporal sulcus	0.63	**0.74**	0.35	0.37
148	R	Transverse temporal sulcus	0.54	0.40	**0.69**	0.34
141	R	Postcentral sulcus	0.61	0.63	**0.69**	0.35
128	R	Superior frontal sulcus	0.63	0.70	**0.69**	0.43
121	R	Ant. seg. of the circular sulcus of insula	0.65	0.62	**0.68**	**0.48**
25	L	Angular gyrus	0.54	0.59	**0.68**	0.40
138	R	Orbital sulci	0.56	0.70	0.55	**0.53**
91	R	Long insular gyrus and central sulcus of the insula	0.72	0.66	0.62	**0.51**
27	L	Superior parietal lobule	0.61	0.71	0.61	**0.47**
108	R	Lateral aspect of the superior temporal gyrus	0.72	0.70	0.53	**0.46**
		Mean±Std (Top 5 nodes)	**0.75±0.02**	**0.77±0.04**	**0.69±0.01**	**0.49±0.03**

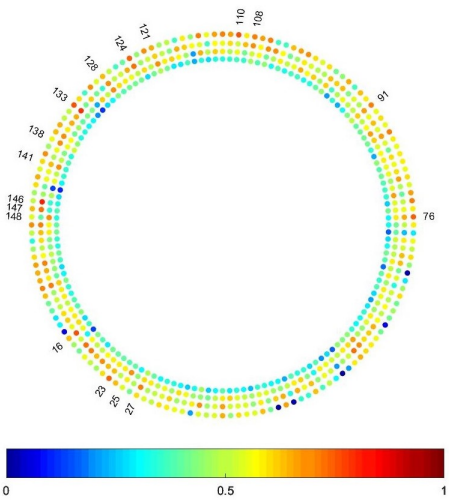

Figure B.1. Classification accuracies for all nodes (from outer to inner : AD/SCI-AD/MCI-MCI/SCI-AD/MCI/SCI)

TEST 3 Parameters: i) CBOW ii) K=40 iii) L=7 iv) w=5 v) k=2 vi) $\alpha = 1$

Table B.2. Classification accuracies of top performing nodes in TEST 3.

#	R/L	Node Name	AD-SCI	AD-MCI	MCI-SCI	All
124	R	Anterior transverse collateral sulcus	**0.81**	**0.72**	0.54	**0.47**
22	L	Lingual gyrus	**0.77**	0.63	0.55	0.47
146	R	Inferior temporal sulcus	**0.74**	**0.83**	0.42	0.35
107	R	Anterior transverse temporal gyrus	**0.74**	0.46	0.45	0.37
95	R	Lateral occipito-temporal gyrus	**0.72**	0.60	0.53	0.34
134	R	Lateral occipito-temporal sulcus	0.63	**0.74**	0.40	0.40
89	R	Middle frontal gyrus	0.67	**0.72**	0.58	0.44
57	L	Middle occipital sulcus and lunatus sulcus	0.51	**0.72**	0.57	0.41
31	L	Straight gyrus	0.63	0.63	**0.73**	0.46
56	L	Intraparietal sulcus	0.60	0.62	**0.70**	**0.56**
38	L	Middle temporal gyrus	0.67	0.65	**0.69**	**0.51**
127	R	Middle frontal sulcus	0.70	0.46	**0.68**	0.31
104	R	Precuneus	0.49	0.54	**0.68**	0.40
27	L	Superior parietal lobule	0.65	0.66	0.68	**0.53**
125	R	Posterior transverse collateral sulcus	0.58	0.62	0.51	**0.50**
		Mean±Std (Top 5 nodes)	**0.76±0.04**	**0.75±0.05**	**0.70±0.02**	**0.51±0.03**

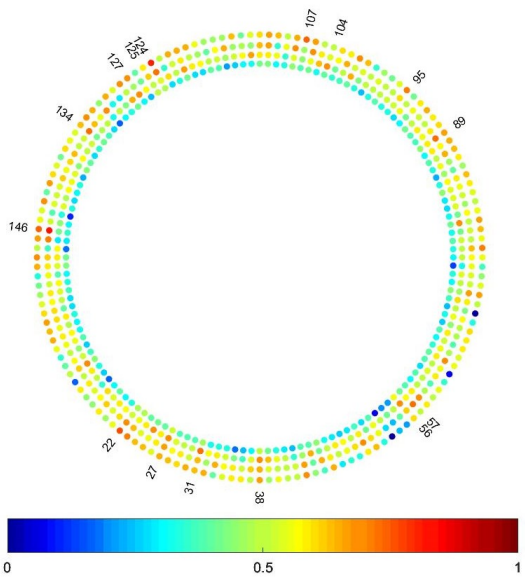

Figure B.2. Classification accuracies for all nodes (from outer to inner : AD/SCI-AD/MCI-MCI/SCI-AD/MCI/SCI)

TEST 4 Parameters: i) CBOW ii) K=40 iii) L=5 iv) w=5 v) k=2 vi) $\alpha = 0.75$

Table B.3. Classification accuracies of top performing nodes in TEST 4.

#	R/L	Node Name	AD-SCI	AD-MCI	MCI-SCI	All
124	R	Anterior transverse collateral sulcus	**0.84**	0.71	0.42	0.35
101	R	Superior parietal lobule	**0.81**	0.54	0.64	0.41
25	L	Angular gyrus	**0.79**	0.55	0.53	0.40
23	L	Parahippocampal gyrus	**0.79**	0.68	0.57	0.45
59	L	Subcallosal area, subcallosal gyrus	**0.77**	0.49	0.47	0.28
35	L	Anterior transverse collateral sulcus	0.58	**0.79**	0.32	0.34
29	L	Precentral gyrus	0.65	**0.79**	0.60	0.48
76	R	Inferior occipital gyrus (O3) and sulcus	0.65	**0.77**	0.45	0.37
68	L	Inferior part of the precentral sulcus	0.42	**0.77**	0.61	0.43
45	L	Central sulcus	0.51	**0.75**	0.49	0.35
113	R	Horiz. ramus of ant. seg. of lateral sulcus	0.72	0.51	**0.72**	0.45
56	L	Intraparietal sulcus	0.63	0.51	**0.72**	0.50
19	L	Middle occipital gyrus	0.65	0.69	**0.72**	**0.59**
128	R	Superior frontal sulcus	0.70	0.57	**0.71**	0.47
58	L	Transverse occipital sulcus	0.70	0.60	**0.69**	**0.51**
39	L	Horiz. ramus of ant. seg. of lateral sulcus	0.56	0.54	0.67	**0.53**
89	R	Middle frontal gyrus	0.32	0.72	0.61	**0.52**
57	L	Middle occipital sulcus and lunatus sulcus	0.72	0.68	0.66	**0.52**
		Mean±Std (Top 5 nodes)	**0.80±0.03**	**0.77±0.02**	**0.71±0.01**	**0.53±0.03**

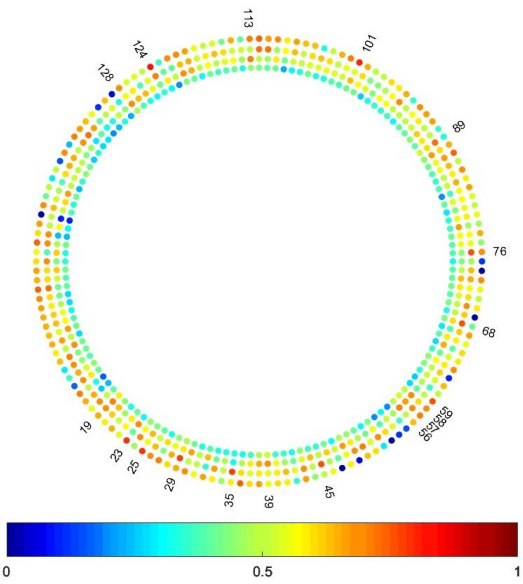

Figure B.3. Classification accuracies for all nodes (from outer to inner : AD/SCI-AD/MCI-MCI/SCI-AD/MCI/SCI)

TEST 5 Parameters: i) CBOW ii) K=40 iii) L=5 iv) w=5 v) k=2 vi) $\alpha = 0$

Table B.4. Classification accuracies of top performing nodes in TEST 5.

#	R/L	Node Name	AD-SCI	AD-MCI	MCI-SCI	All
23	L	Parahippocampal gyrus	**0.84**	0.50	**0.72**	0.39
124	R	Anterior transverse collateral sulcus	**0.79**	**0.74**	0.45	0.36
59	L	Ant. occipital sulcus & preoccipital notch	**0.77**	0.54	0.47	0.36
95	R	Lateral occipito-temporal gyrus	**0.74**	**0.75**	0.38	0.32
147	R	Superior temporal sulcus	**0.72**	0.39	0.66	0.34
35	L	Planum polare of the superior temporal gyru	0.54	**0.77**	0.28	0.30
1	L	Fronto-marginal gyrus and sulcus	0.63	**0.77**	0.32	0.24
18	L	Short insular gyri	0.35	**0.75**	0.46	0.33
56	L	Intraparietal sulcus	0.72	0.52	**0.80**	**0.54**
38	L	Middle temporal gyrus	0.72	0.64	**0.74**	**0.53**
58	L	Transverse occipital sulcus	0.67	0.65	**0.73**	**0.53**
113	R	Horiz. ramus of ant. seg. of lateral sulcus	0.65	0.59	**0.70**	0.45
89	R	Middle frontal gyrus	0.44	0.66	0.69	**0.57**
19	L	Middle occipital gyrus	0.63	0.71	0.70	**0.55**
		Mean±Std (Top 5 nodes)	**0.77±0.05**	**0.76±0.01**	**0.74±0.04**	**0.54±0.02**

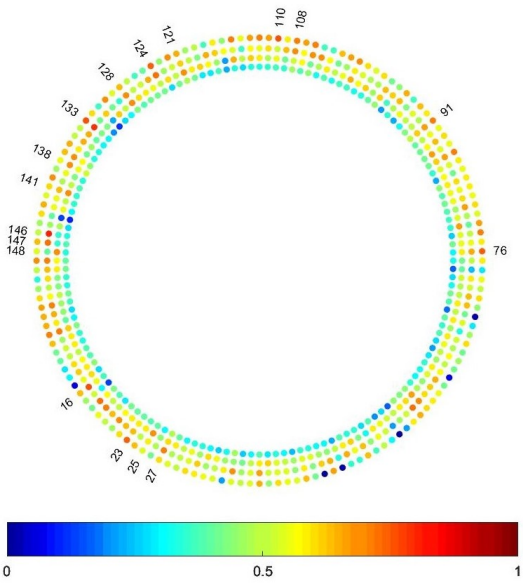

Figure B.4. Classification accuracies for all nodes (from outer to inner : AD/SCI-AD/MCI-MCI/SCI-AD/MCI/SCI)

TEST 6 Parameters: i) CBOW ii) K=40 iii) L=5 iv) w=5 v) k=2 vi) $\alpha = 1$

Table B.5. Classification accuracies of top performing nodes in TEST 6.

#	R/L	Node Name	AD-SCI	AD-MCI	MCI-SCI	All
124	R	Anterior transverse collateral sulcus	**0.81**	**0.72**	0.49	0.41
111	R	Inferior temporal gyrus	**0.79**	0.69	0.54	0.39
102	R	Inferior temporal gyrus	**0.77**	0.59	0.54	0.33
76	R	Inferior occipital gyrus (O3) and sulcus	**0.77**	**0.79**	0.43	0.40
38	L	Middle temporal gyrus	**0.74**	0.55	0.58	0.43
146	R	Inferior temporal sulcus	0.63	**0.74**	0.42	0.32
12	L	Opercular part of the inferior frontal gyrus	0.67	**0.74**	0.52	0.48
68	L	Inferior part of the precentral sulcus	0.54	**0.72**	0.65	**0.48**
56	L	Intraparietal sulcus	0.65	0.46	**0.74**	**0.50**
65	L	Ant. seg. of the circular sulcus of insula	0.49	0.32	**0.72**	0.36
44	L	Calcarine sulcus	0.51	0.45	**0.70**	0.40
113	R	Horiz. ramus of ant. seg. of lateral sulcus	0.65	0.52	**0.69**	0.44
29	L	Precentral gyrus	0.63	0.72	**0.69**	**0.53**
52	L	Inferior frontal sulcus	0.44	0.70	0.63	**0.51**
89	R	Middle frontal gyrus	0.44	0.66	0.58	**0.50**
		Mean±Std (Top 5 nodes)	**0.78±0.03**	**0.74±0.03**	**0.71±0.02**	**0.50±0.02**

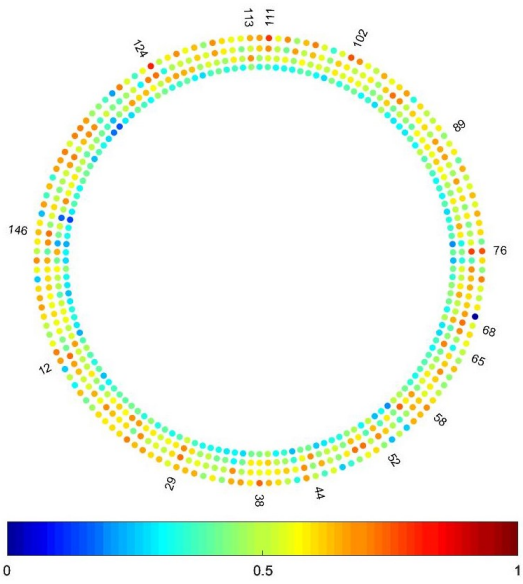

Figure B.5. Classification accuracies for all nodes (from outer to inner : AD/SCI-AD/MCI-MCI/SCI-AD/MCI/SCI)

TEST 7 Parameters: i) CBOW ii) K=40 iii) L=10 iv) w=5 v) k=2 vi) $\alpha = 0.75$

Table B.6. Classification accuracies of top performing nodes in TEST 7.

#	R/L	Node Name	AD-SCI	AD-MCI	MCI-SCI	All
124	R	Anterior transverse collateral sulcus	**0.79**	0.66	0.46	0.37
111	R	Inferior temporal gyrus	**0.79**	0.65	0.30	0.28
39	L	Horiz. ramus of ant. seg. of lateral sulcus	**0.79**	0.63	0.64	0.51
147	R	Superior temporal sulcus	**0.72**	**0.75**	0.27	0.30
134	R	Lateral occipito-temporal sulcus	**0.72**	0.69	0.50	0.34
146	R	Inferior temporal sulcus	0.63	**0.83**	0.27	0.35
133	R	Ant. occipital sulcus & preoccipital notch	0.67	**0.77**	0.53	0.45
96	R	Calcarine sulcus	0.70	**0.74**	0.62	0.48
14	L	Triangular part of the inferior frontal gyrus	0.47	**0.74**	0.55	**0.51**
38	L	Middle temporal gyrus	0.65	0.63	**0.73**	**0.51**
88	L	Triangular part of the inferior frontal gyrus	0.65	0.59	**0.70**	**0.51**
56	L	Intraparietal sulcus	0.51	0.63	**0.69**	0.48
90	R	Superior frontal gyrus (F1)	0.72	0.65	**0.68**	0.40
87	R	Orbital part of the inferior frontal gyrus	0.58	0.69	**0.68**	0.48
137	R	Medial orbital sulcus	0.65	0.57	0.64	**0.51**
25	L	Angular gyrus	0.49	0.63	0.66	**0.51**
		Mean±Std (Top 5 nodes)	**0.76±0.04**	**0.77±0.04**	**0.70±0.02**	**0.51±0.00**

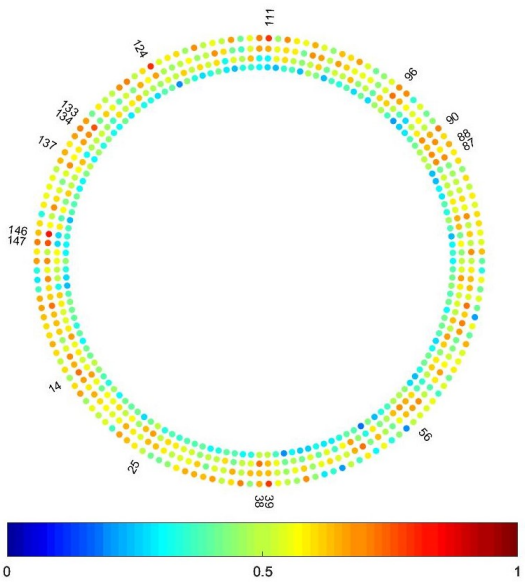

Figure B.6. Classification accuracies for all nodes (from outer to inner : AD/SCI-AD/MCI-MCI/SCI-AD/MCI/SCI)

TEST 8 Parameters: i) CBOW ii) K=40 iii) L=10 iv) w=5 v) k=2 vi) $\alpha = 0$

Table B.7. Classification accuracies of top performing nodes in TEST 8.

#	R/L	Node Name	AD-SCI	AD-MCI	MCI-SCI	All
124	R	Anterior transverse collateral sulcus	**0.81**	0.66	0.45	0.37
16	L	Superior frontal gyrus (F1)	**0.79**	0.68	0.47	0.37
96	R	Ant. occipital sulcus & preoccipital notch	**0.77**	0.67	0.46	0.41
147	R	Superior temporal sulcus	**0.74**	0.70	0.37	0.34
141	R	Postcentral sulcus	**0.74**	0.49	0.62	0.36
146	R	Inferior temporal sulcus	0.65	**0.80**	0.39	0.37
133	R	Ant. occipital sulcus & preoccipital notch	0.65	**0.77**	0.54	0.46
108	R	Lateral aspect of the superior temporal gyrus	0.74	**0.74**	0.42	0.33
72	L	Calcarine sulcus	0.47	**0.72**	0.60	0.44
111	R	Inferior temporal gyrus	0.72	**0.75**	0.35	0.30
121	R	Ant. seg. of the circular sulcus of insula	0.51	0.47	**0.74**	0.44
128	R	Superior frontal sulcus	0.74	0.63	**0.68**	**0.53**
56	L	Intraparietal sulcus	0.70	0.55	**0.68**	0.46
27	L	Superior parietal lobule	0.51	0.55	**0.68**	0.43
145	R	Subparietal sulcus	0.63	0.60	**0.66**	**0.47**
70	L	Suborbital sulcus	0.60	0.68	0.57	**0.50**
52	L	Inferior frontal sulcus	0.44	0.57	0.65	**0.50**
5	L	Transverse frontopolar gyri and sulci	0.67	0.66	0.66	**0.48**
		Mean±Std (Top 5 nodes)	**0.77±0.03**	**0.76±0.03**	**0.69±0.03**	**0.50±0.02**

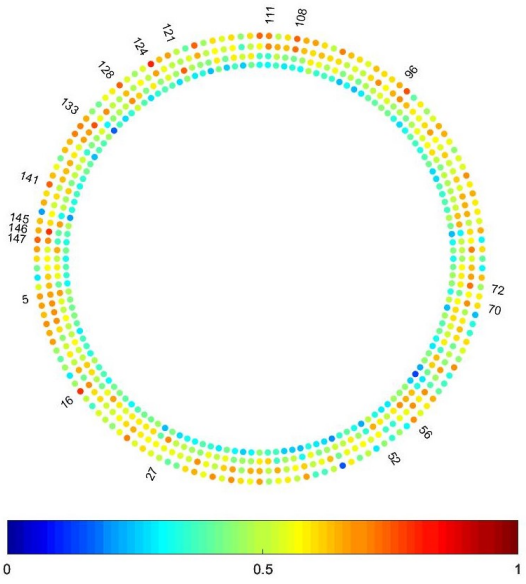

Figure B.7. Classification accuracies for all nodes (from outer to inner : AD/SCI-AD/MCI-MCI/SCI-AD/MCI/SCI)

TEST 9 Parameters: i) CBOW ii) K=40 iii) L=10 iv) w=5 v) k=2 vi) $\alpha = 1$

Table B.8. Classification accuracies of top performing nodes in TEST 9.

#	R/L	Node Name	AD-SCI	AD-MCI	MCI-SCI	All
124	R	Anterior transverse collateral sulcus	**0.81**	0.52	0.57	0.48
76	R	Inferior occipital gyrus (O3) and sulcus	**0.79**	0.69	0.47	0.42
31	L	Straight gyrus	**0.79**	0.69	0.57	**0.52**
134	R	Lateral occipito-temporal sulcus	**0.77**	0.68	0.31	0.30
115	R	Posterior ramus of the lateral sulcus	**0.77**	0.59	0.61	0.37
133	R	Ant. occipital sulcus & preoccipital notch	0.72	**0.83**	0.34	0.31
70	L	Suborbital sulcus	0.65	**0.77**	0.66	**0.57**
35	L	Planum polare of the superior temporal gyrus	0.70	**0.74**	0.35	0.32
17	L	Horiz. ramus of ant. seg. of lateral sulcus	0.58	**0.74**	0.30	0.28
146	R	Inferior temporal sulcus	0.72	**0.72**	0.32	0.36
25	L	Angular gyrus	0.58	0.65	**0.76**	**0.51**
56	L	Intraparietal sulcus	0.49	0.52	**0.72**	0.45
127	R	Middle frontal sulcus	0.74	0.50	**0.70**	0.42
52	L	Inferior frontal sulcus	0.63	0.55	**0.69**	**0.53**
38	L	Middle temporal gyrus	0.72	0.57	**0.69**	0.45
14	L	Triangular part of the inferior frontal gyrus	0.49	0.69	0.63	**0.52**
		Mean±Std (Top 5 nodes)	**0.79±0.02**	**0.76±0.04**	**0.71±0.03**	**0.53±0.02**

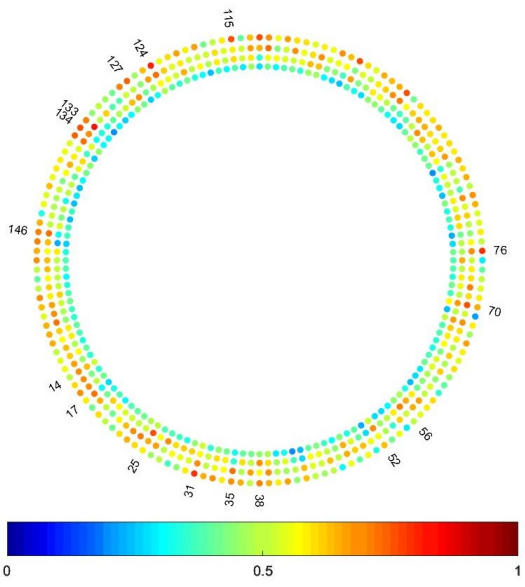

Figure B.8. Classification accuracies for all nodes (from outer to inner : AD/SCI-AD/MCI-MCI/SCI-AD/MCI/SCI)

TEST 10 Parameters: i) CBOW ii) K=40 iii) L=7 iv) w=5 v) k=5 vi) $\alpha = 0.75$

Table B.9. Classification accuracies of top performing nodes in TEST 10.

#	R/L	Node Name	AD-SCI	AD-MCI	MCI-SCI	All
141	R	Postcentral sulcus	**0.84**	0.54	**0.66**	0.45
124	R	Anterior transverse collateral sulcus	**0.84**	**0.75**	0.58	**0.53**
107	R	Anterior transverse temporal gyrus	**0.81**	0.59	0.54	0.40
146	R	Inferior temporal sulcus	**0.79**	0.71	0.47	0.44
133	R	Ant. occipital sulcus & preoccipital notch	**0.77**	**0.74**	0.47	0.40
96	R	Anterior transverse collateral sulcus	0.42	**0.77**	0.53	0.36
22	L	Lingual gyrus	0.65	**0.77**	0.53	0.45
117	R	Temporal pole	0.65	**0.72**	0.46	0.39
27	L	Superior parietal lobule	0.70	0.60	**0.70**	**0.54**
25	L	Angular gyrus	0.61	0.68	**0.69**	0.48
56	L	Intraparietal sulcus	0.54	0.59	**0.66**	0.43
39	L	Horiz. ramus of ant. seg. of lateral sulcus	0.51	0.51	**0.66**	0.35
14	L	Triangular part of the inferior frontal gyrus	0.54	0.65	0.65	**0.51**
103	R	Precentral gyrus	0.77	0.71	0.51	**0.50**
12	L	Opercular part of the inferior frontal gyrus	0.51	0.62	0.62	**0.50**
		Mean±Std (Top 5 nodes)	**0.81±0.03**	**0.75±0.02**	**0.67±0.02**	**0.52±0.02**

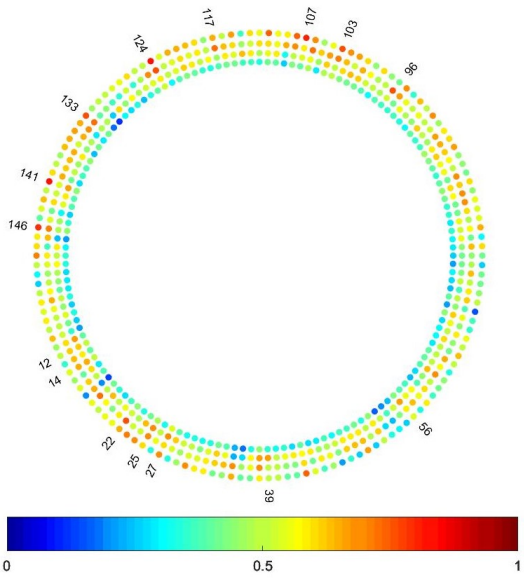

Figure B.9. Classification accuracies for all nodes (from outer to inner : AD/SCI-AD/MCI-MCI/SCI-AD/MCI/SCI)

TEST 11 Parameters: i) CBOW ii) K=40 iii) L=7 iv) w=10 v) k=2 vi) $\alpha = 0.75$

Table B.10. Classification accuracies of top performing nodes in TEST 11.

#	R/L	Node Name	AD-SCI	AD-MCI	MCI-SCI	All
107	R	Anterior transverse temporal gyrus	**0.79**	0.66	0.53	0.43
106	R	Subcallosal area, subcallosal gyrus	**0.77**	0.65	0.58	0.43
146	R	Inferior temporal sulcus	**0.74**	**0.77**	0.46	0.46
147	R	Superior temporal sulcus	**0.72**	0.66	0.26	0.24
140	R	Pericallosal sulcus	**0.72**	0.51	0.51	0.39
117	R	Temporal pole	0.65	**0.78**	0.51	**0.47**
124	R	Anterior transverse collateral sulcus	0.70	**0.77**	0.53	**0.47**
48	L	Calcarine sulcus	0.44	**0.74**	0.61	0.42
12	L	Opercular part of the inferior frontal gyrus	0.51	**0.74**	0.53	**0.48**
22	L	Lingual gyrus	0.56	0.59	**0.69**	0.45
44	L	Calcarine sulcus	0.65	0.57	**0.66**	0.45
108	R	Lateral aspect of the superior temporal gyrus	0.63	0.72	**0.65**	**0.51**
101	R	Superior parietal lobule	0.65	0.43	**0.65**	0.40
56	L	Intraparietal sulcus	0.54	0.50	**0.65**	0.40
27	L	Superior parietal lobule	0.65	0.51	0.64	**0.48**
		Mean±Std (Top 5 nodes)	**0.75±0.03**	**0.76±0.02**	**0.66±0.02**	**0.48±0.02**

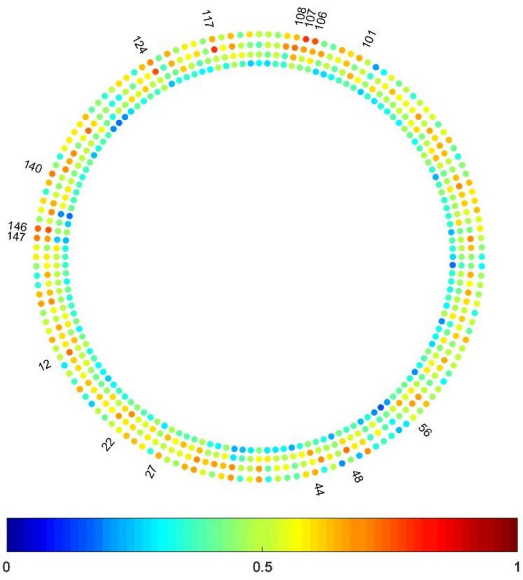

Figure B.10. Classification accuracies for all nodes (from outer to inner : AD/SCI-AD/MCI-MCI/SCI-AD/MCI/SCI)

TEST 12 Parameters: i) CBOW ii) K=40 iii) L=20 iv) w=5 v) k=2 vi) $\alpha = 0.75$

Table B.11. Classification accuracies of top performing nodes in TEST 12.

#	R/L	Node Name	AD-SCI	AD-MCI	MCI-SCI	All
107	R	Anterior transverse temporal gyrus	**0.77**	0.62	0.68	0.46
124	R	Anterior transverse collateral sulcus	**0.74**	**0.75**	0.46	0.43
106	R	Subcallosal area, subcallosal gyrus	**0.74**	0.70	0.54	0.46
128	R	Superior frontal sulcus	**0.70**	0.71	0.57	0.44
102	R	Inferior temporal gyrus	**0.70**	0.54	0.51	0.32
147	R	Superior temporal sulcus	0.56	**0.79**	0.46	0.39
133	R	Ant. occipital sulcus & preoccipital notch	0.63	**0.75**	0.43	0.35
146	R	Inferior temporal sulcus	0.65	**0.72**	0.35	0.30
137	R	Medial orbital sulcus	0.54	**0.72**	0.62	**0.50**
52	L	Inferior frontal sulcus	0.32	0.61	**0.77**	0.44
19	L	Middle occipital gyrus	0.56	0.66	**0.73**	**0.55**
2	L	Inferior occipital gyrus (O3) and sulcus	0.35	0.70	**0.71**	0.48
80	R	Ant. part of the cingulate gyrus and sulcus	0.51	0.50	**0.70**	**0.51**
25	L	Angular gyrus	0.54	0.60	**0.70**	**0.51**
115	R	Posterior ramus of the lateral sulcus	0.54	0.62	0.68	**0.50**
		Mean±Std (Top 5 nodes)	**0.74±0.03**	**0.75±0.03**	**0.72±0.03**	**0.51±0.02**

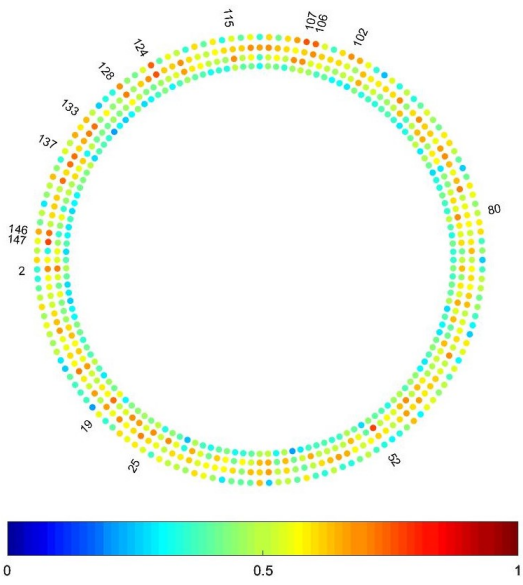

Figure B.11. Classification accuracies for all nodes (from outer to inner : AD/SCI-AD/MCI-MCI/SCI-AD/MCI/SCI)

TEST 13 Parameters: i) CBOW ii) K=40 iii) L=50 iv) w=5 v) k=2 vi) $\alpha = 0.75$

Table B.12. Classification accuracies of top performing nodes in TEST 13.

#	R/L	Node Name	AD-SCI	AD-MCI	MCI-SCI	All
98	R	Lateral occipito-temporal gyrus	**0.74**	**0.75**	0.57	0.46
124	R	Anterior transverse collateral sulcus	**0.72**	0.72	0.65	0.52
78	R	Subcentral gyrus and sulci	**0.70**	**0.77**	0.60	0.50
31	L	Straight gyrus	**0.70**	0.66	0.52	0.40
24	L	Orbital gyri	**0.70**	0.66	0.50	0.40
23	L	Parahippocampal gyrus	0.65	**0.79**	0.65	0.54
66	L	Calcarine sulcus	0.61	**0.77**	0.65	0.55
41	L	Posterior ramus of the lateral sulcus	0.56	**0.77**	0.64	0.48
122	R	Inferior segment of the circular sulcus of the insula	0.49	0.72	**0.70**	**0.56**
44	L	Calcarine sulcus	0.58	0.75	**0.70**	**0.58**
10	L	Posterior-ventral part of the cingulate gyrus	0.70	0.74	**0.70**	**0.57**
102	R	Postcentral gyrus	0.58	0.74	**0.69**	0.54
83	R	Posterior-dorsal part of the cingulate gyrus	0.65	0.71	**0.69**	0.51
4	L	Subcentral gyrus and sulci	0.56	0.74	0.69	**0.57**
68	L	Inferior part of the precentral sulcus	0.35	0.69	0.69	**0.56**
		Mean±Std (Top 5 nodes)	**0.71±0.02**	**0.77±0.01**	**0.70±0.01**	**0.57±0.01**

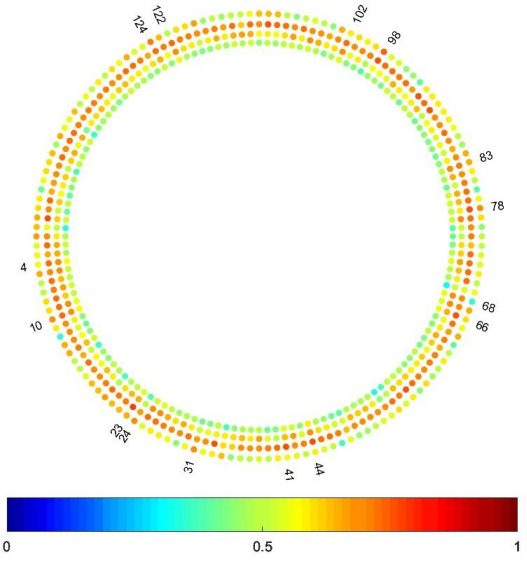

Figure B.12. Classification accuracies for all nodes (from outer to inner : AD/SCI-AD/MCI-MCI/SCI-AD/MCI/SCI)

TEST 14 Parameters: i) Skip-Gram ii) K=40 iii) L=7 iv) w=5 v) k=2 vi) $\alpha = 0.75$

Table B.13. Classification accuracies of top performing nodes in TEST 14.

#	R/L	Node Name	AD-SCI	AD-MCI	MCI-SCI	All
124	R	Anterior transverse collateral sulcus	**0.77**	**0.74**	0.55	**0.51**
143	R	Superior part of the precentral sulcus	**0.72**	0.50	0.51	**0.52**
125	R	Posterior transverse collateral sulcus	**0.72**	0.63	0.55	0.44
22	L	Lingual gyrus	**0.72**	0.60	0.55	0.40
135	R	Subcallosal area, subcallosal gyrus	**0.70**	0.55	0.60	0.46
108	R	Lateral aspect of the superior temporal gyrus	0.70	**0.74**	0.60	**0.51**
49	L	Sup. seg. of the circular sulcus of the insula	0.35	**0.74**	0.49	0.37
146	R	Inferior temporal sulcus	0.65	**0.72**	0.47	0.36
100	R	Supramarginal gyrus	0.54	**0.72**	0.55	**0.51**
148	R	Transverse temporal sulcus	0.65	0.59	**0.68**	0.45
115	R	Posterior ramus of the lateral sulcus	0.67	0.66	**0.66**	**0.52**
87	R	Orbital part of the inferior frontal gyrus	0.44	0.70	**0.66**	0.46
69	L	Superior part of the precentral sulcus	0.28	0.51	**0.66**	0.45
121	R	Ant. seg. of the circular sulcus of insula	0.49	0.49	**0.64**	0.38
		Mean±Std (Top 5 nodes)	**0.73±0.03**	**0.73±0.01**	**0.66±0.01**	**0.51±0.01**

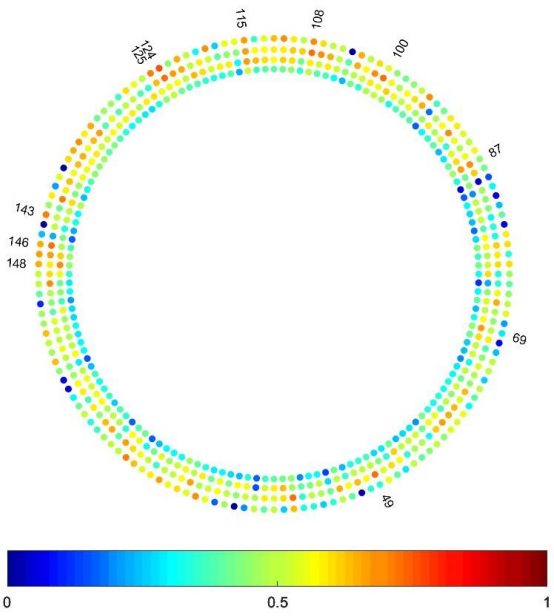

Figure B.13. Classification accuracies for all nodes (from outer to inner : AD/SCI-AD/MCI-MCI/SCI-AD/MCI/SCI)

TEST 15 Parameters: i) Skip-Gram ii) K=40 iii) L=7 iv) w=5 v) k=5 vi) $\alpha = 0.75$

Table B.14. Classification accuracies of top performing nodes in TEST 15.

#	R/L	Node Name	AD-SCI	AD-MCI	MCI-SCI	All
23	L	Parahippocampal gyrus	**0.77**	0.57	0.51	0.36
115	R	Posterior ramus of the lateral sulcus	**0.74**	0.66	0.54	**0.42**
140	R	Pericallosal sulcus	**0.72**	0.48	0.50	0.34
141	R	Postcentral sulcus	**0.70**	0.63	0.54	0.40
124	R	Anterior transverse collateral sulcus	**0.70**	0.57	0.43	0.32
92	R	Anterior transverse collateral sulcus	0.42	**0.77**	0.35	0.34
131	R	Calcarine sulcus	0.40	**0.74**	0.47	0.37
125	R	Posterior transverse collateral sulcus	0.58	**0.74**	0.53	0.41
108	R	Lateral aspect of the superior temporal gyrus	0.58	**0.72**	0.47	0.40
89	R	Middle frontal gyrus	0.63	**0.72**	0.46	0.42
52	L	Inferior frontal sulcus	0.58	0.43	**0.69**	0.42
25	L	Angular gyrus	0.67	0.48	**0.66**	0.39
130	R	Intraparietal sulcus	0.54	0.57	**0.64**	0.37
14	L	Triangular part of the inferior frontal gyrus	0.40	0.60	**0.64**	**0.43**
101	R	Superior parietal lobule	0.65	0.54	**0.62**	0.37
121	R	Ant. seg. of the circular sulcus of insula	0.56	0.65	0.60	**0.45**
113	R	Horiz. ramus of ant. seg. of lateral sulcus	0.51	0.62	0.61	**0.43**
13	L	Orbital part of the inferior frontal gyrus	0.47	0.68	0.61	**0.43**
		Mean±Std (Top 5 nodes)	**0.73±0.03**	**0.74±0.02**	**0.65±0.03**	**0.43±0.01**

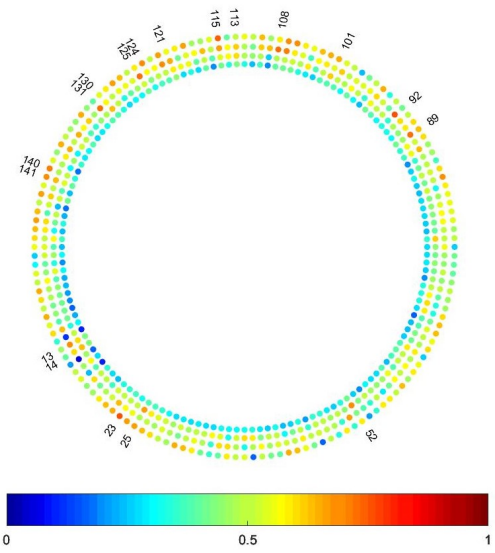

Figure B.14. Classification accuracies for all nodes (from outer to inner : AD/SCI-AD/MCI-MCI/SCI-AD/MCI/SCI)

TEST 16 Parameters: i) Skip-Gram ii) K=40 iii) L=7 iv) w=10 v) k=2 vi) $\alpha = 0.75$

Table B.15. Classification accuracies of top performing nodes in TEST 16.

#	R/L	Node Name	AD-SCI	AD-MCI	MCI-SCI	All
69	L	Lateral occipito-temporal gyrus	**0.72**	0.52	0.50	0.33
30	L	Precuneus	**0.67**	0.46	0.60	0.36
74	L	Transverse temporal sulcus	**0.65**	0.51	0.24	0.19
6	L	Inferior occipital gyrus (O3) and sulcus	**0.63**	0.52	0.35	0.18
114	R	Subcallosal area, subcallosal gyrus	**0.61**	0.60	0.54	0.42
147	R	Superior temporal sulcus	0.49	**0.74**	0.31	0.25
138	R	Orbital sulci	0.54	**0.72**	0.49	0.39
126	R	Anterior transverse collateral sulcus	0.58	**0.72**	**0.64**	**0.55**
51	L	Posterior transverse collateral sulcus	0.42	**0.71**	0.26	0.26
88	R	Triangular part of the inferior frontal gyrus	0.54	**0.70**	0.64	**0.53**
123	R	Sup. seg. of the circular sulcus of the insula	0.26	0.55	**0.68**	0.43
136	R	Lateral orbital sulcus	0.58	0.65	**0.66**	**0.43**
113	R	Horiz. ramus of ant. seg. of lateral sulcus	0.47	0.46	**0.66**	0.41
14	L	Triangular part of the inferior frontal gyrus	0.54	0.42	**0.66**	0.36
142	R	Inferior part of the precentral sulcus	0.42	0.66	0.55	**0.44**
115	R	Posterior ramus of the lateral sulcus	0.44	0.68	0.55	**0.44**
		Mean±Std (Top 5 nodes)	**0.66±0.04**	**0.72±0.01**	**0.66±0.01**	**0.48±0.06**

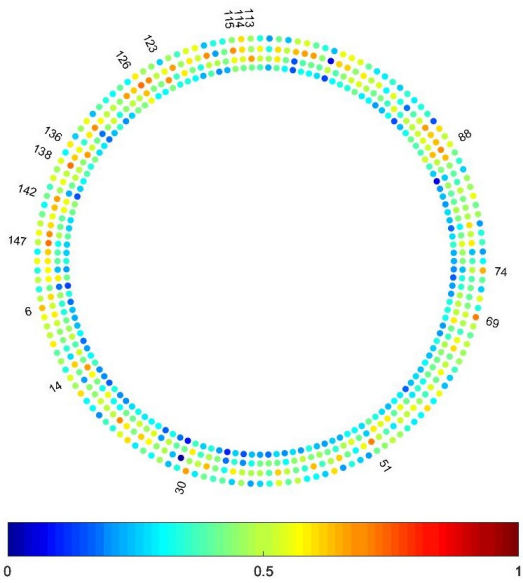

Figure B.15. Classification accuracies for all nodes (from outer to inner : AD/SCI-AD/MCI-MCI/SCI-AD/MCI/SCI)

TEST 17 Parameters: i) Skip-Gram ii) K=40 iii) L=5 iv) w=5 v) k=2 vi) $\alpha = 0.75$

Table B.16. Classification accuracies of top performing nodes in TEST 17.

#	R/L	Node Name	AD-SCI	AD-MCI	MCI-SCI	All
125	R	Posterior transverse collateral sulcus	**0.79**	0.60	0.54	0.43
133	R	Ant. occipital sulcus & preoccipital notch	**0.77**	0.39	0.31	0.34
124	R	Anterior transverse collateral sulcus	**0.74**	0.59	0.41	0.39
102	R	Postcentral gyrus	**0.74**	0.45	0.55	0.33
63	L	Subcallosal area, subcallosal gyrus	**0.74**	**0.75**	0.42	0.33
138	R	Orbital sulci	0.51	**0.77**	0.49	0.35
17	L	Calcarine sulcus	0.58	**0.77**	0.40	0.35
68	L	Inferior part of the precentral sulcus	0.11	**0.74**	0.57	0.42
12	L	Opercular part of the inferior frontal gyrus	0.37	**0.72**	0.47	0.36
142	R	Ant. seg. of the circular sulcus of insula	0.54	0.35	**0.74**	**0.46**
89	R	Middle frontal gyrus	0.54	0.20	**0.72**	0.44
56	L	Intraparietal sulcus	0.70	0.55	**0.70**	0.41
19	L	Middle occipital gyrus	0.63	0.62	**0.70**	**0.53**
126	R	Inferior frontal sulcus	0.56	0.43	**0.69**	0.42
121	R	Ant. seg. of the circular sulcus of insula	0.58	0.55	0.66	**0.48**
91	R	Long insular gyrus and central sulcus of the insula	0.54	0.65	0.64	**0.48**
46	L	Marginal branch of the cingulate sulcus	0.35	0.52	0.67	**0.46**
		Mean±Std (Top 5 nodes)	**0.76±0.02**	**0.75±0.02**	**0.71±0.02**	**0.48±0.03**

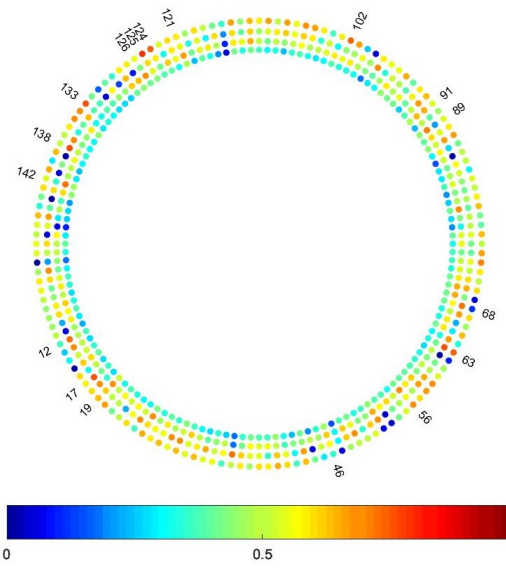

Figure B.16. Classification accuracies for all nodes (from outer to inner : AD/SCI-AD/MCI-MCI/SCI-AD/MCI/SCI)

TEST 18 Parameters: i) Skip-Gram ii) K=40 iii) L=10 iv) w=5 v) k=2 vi) $\alpha = 0.75$

Table B.17. Classification accuracies of top performing nodes in TEST 18.

#	R/L	Node Name	AD-SCI	AD-MCI	MCI-SCI	All
136	R	Lateral occipito-temporal gyrus	**0.77**	0.66	0.62	0.43
10	L	Posterior-ventral part of the cingulate gyrus	**0.74**	0.54	0.54	036
148	R	Transverse temporal sulcus	**0.72**	0.46	0.49	0.26
129	R	Inferior occipital gyrus (O3) and sulcus	**0.72**	**0.68**	0.41	0.31
128	R	Superior frontal sulcus	**0.72**	0.63	0.65	**0.45**
49	L	Anterior transverse collateral sulcus	0.61	**0.71**	0.41	0.33
70	L	Suborbital sulcus	0.54	**0.69**	0.53	**0.45**
55	L	Sulcus intermedius primus	0.49	**0.69**	0.61	0.44
13	L	Orbital part of the inferior frontal gyrus	0.37	**0.69**	0.41	0.37
56	L	Intraparietal sulcus	0.63	0.45	**0.78**	**0.46**
75	R	Fronto-marginal gyrus and sulcus	0.61	0.48	**0.70**	0.43
27	L	Superior parietal lobule	0.35	0.46	**0.70**	0.37
25	L	Angular gyrus	0.35	0.29	**0.69**	0.31
80	R	Ant. part of the cingulate gyrus and sulcus	0.61	0.62	**0.69**	0.45
88	R	Triangular part of the inferior frontal gyrus	0.54	0.65	0.64	**0.52**
105	R	Straight gyrus	0.67	0.62	0.65	**0.45**
		Mean±Std (Top 5 nodes)	**0.73±0.02**	**0.69±0.01**	**0.71±0.04**	**0.47±0.03**

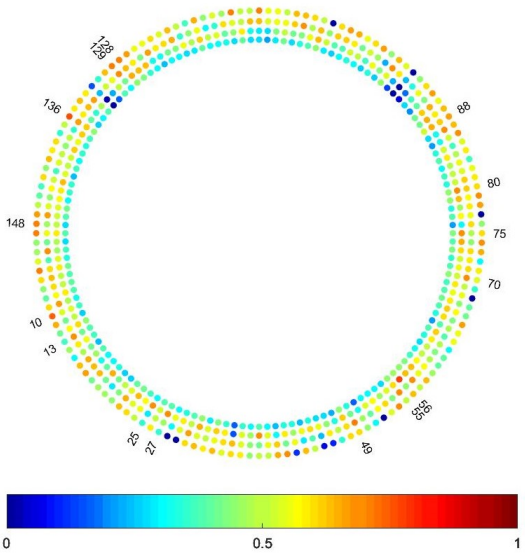

Figure B.17. Classification accuracies for all nodes (from outer to inner : AD/SCI-AD/MCI-MCI/SCI-AD/MCI/SCI)

TEST 19 Parameters: i) Skip-Gram ii) K=40 iii) L=20 iv) w=5 v) k=2 vi) $\alpha = 0.75$

Table B.18. Classification accuracies of top performing nodes in TEST 19.

#	R/L	Node Name	AD-SCI	AD-MCI	MCI-SCI	All
112	R	Lateral occipito-temporal gyrus	**0.74**	0.65	0.58	0.43
108	R	Lateral aspect of the superior temporal gyrus	**0.72**	0.68	0.52	0.41
147	R	Superior temporal sulcus	**0.70**	**0.70**	0.54	0.44
122	R	Inferior segment of the circular sulcus of the insula	**0.70**	0.55	0.61	0.37
124	R	Anterior transverse collateral sulcus	**0.67**	**0.70**	0.50	0.43
106	R	Subcallosal area, subcallosal gyrus	0.67	**0.77**	0.53	**0.51**
70	L	Suborbital sulcus	0.42	**0.72**	0.42	0.29
14	L	Triangular part of the inferior frontal gyrus	0.33	**0.71**	0.57	0.42
23	L	Parahippocampal gyrus	0.67	0.39	**0.69**	0.40
120	R	Marginal branch of the cingulate sulcus	0.47	0.59	**0.65**	**0.48**
22	L	Lingual gyrus	0.56	0.66	**0.65**	0.40
10	L	Posterior-ventral part of the cingulate gyrus	0.60	0.43	**0.65**	0.31
118	R	Calcarine sulcus	0.16	0.46	**0.65**	0.35
145	R	Subparietal sulcus	0.37	0.62	0.61	**0.48**
101	R	Superior parietal lobule	0.67	0.63	0.58	**0.47**
96	R	Lingual gyrus	0.61	0.68	0.51	**0.47**
		Mean±Std (Top 5 nodes)	**0.71±0.02**	**0.72±0.03**	**0.66±0.02**	**0.48±0.02**

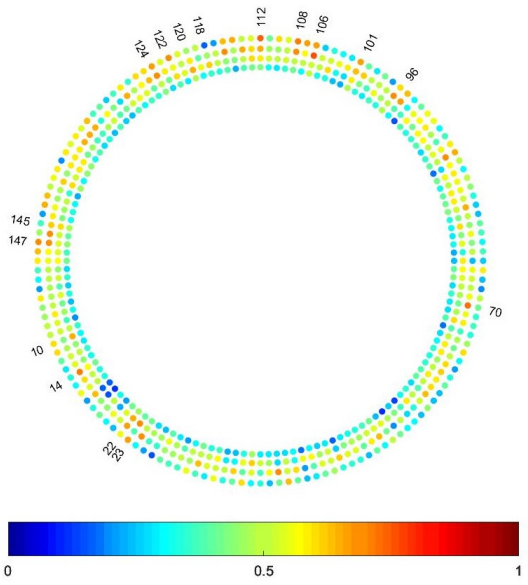

Figure B.18. Classification accuracies for all nodes (from outer to inner : AD/SCI-AD/MCI-MCI/SCI-AD/MCI/SCI)

TEST 20 Parameters: i) Skip-Gram ii) K=40 iii) L=50 iv) w=5 v) k=2 vi) $\alpha = 0.75$

Table B.19. Classification accuracies of top performing nodes in TEST 20.

#	R/L	Node Name	AD-SCI	AD-MCI	MCI-SCI	All
125	R	Posterior transverse collateral sulcus	**0.74**	0.69	0.50	0.41
88	R	Triangular part of the inferior frontal gyrus	**0.70**	0.57	0.65	**0.47**
25	L	Angular gyrus	**0.70**	0.39	**0.73**	0.42
24	L	Inferior occipital gyrus (O3) and sulcus	**0.70**	0.63	0.58	**0.47**
106	R	Subcallosal area, subcallosal gyrus	**0.67**	**0.72**	0.51	0.46
49	L	Anterior transverse collateral sulcus	0.32	**0.75**	0.54	0.39
146	R	Inferior temporal sulcus	0.54	**0.72**	0.49	0.36
124	R	Anterior transverse collateral sulcus	0.56	**0.71**	0.26	0.31
73	L	Superior temporal sulcus	0.61	**0.71**	0.53	0.46
27	L	Superior parietal lobule	0.63	0.48	**0.73**	0.45
123	R	Ant. seg. of the circular sulcus of insula	0.47	0.51	**0.68**	0.44
23	L	Parahippocampal gyrus	0.63	0.51	**0.68**	**0.47**
1	L	Fronto-marginal gyrus and sulcus	0.63	0.57	**0.68**	0.42
128	R	Superior frontal sulcus	0.61	0.54	0.66	**0.47**
108	R	Lateral aspect of the superior temporal gyrus	0.54	0.55	0.64	**0.47**
		Mean±Std (Top 5 nodes)	**0.70±0.02**	**0.72±0.02**	**0.70±0.03**	**0.47±0.00**

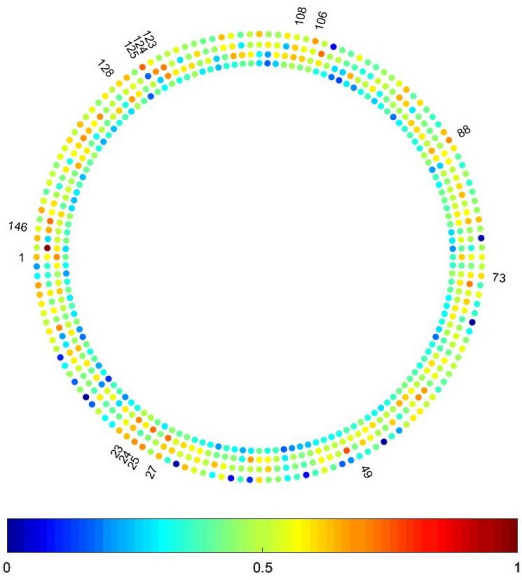

Figure B.19. Classification accuracies for all nodes (from outer to inner : AD/SCI-AD/MCI-MCI/SCI-AD/MCI/SCI)