Predict Psychiatric Disorders via Graph Classification on Brain Networks

Ann Huang
Montreal, Canada
zixiang.huang@mail.mcgill.ca

Jenny Cao
Montreal, Canada
yuanhui.cao@mail.mcgill.ca

Alex Liu
Montreal, Canada
alex.liu@mail.mcgill.ca

ABSTRACT

Empirical findings from the neuroscience literature suggest that psychiatric disorders are often associated with altered connectivity patterns between different regions of the brain. At the intersection of neuroscience and network science, it is a growing line of research to exploit the network structure of the brain, and apply graph mining algorithms to extract brain network patterns that differ between healthy individuals and psychiatric patients. Here we aim to examine possible improvements over a previously proposed, highly interpretable graph classification algorithm called "Contrast Subgraph", in an effort to increase performance and robustness in discriminating psychiatric patients from healthy individuals. We first propose a direct improvement on contrast subgraphs by instead extracting the top K contrast subgraphs to satisfy the need for complex classification tasks. We also propose a completely different, less interpretable embedding-based approach using state-of-the-art algorithms, graph2vec [10] and sub2vec [1].

Author Keywords

graph classification, psychiatric disorder, brain network, functional connectivity, contrast subgraph

INTRODUCTION

Neurological or severe psychiatric disorders can often be reflected by the impairments in the structural or functional connectivity of the brain network [12]. The development of brain imaging techniques such as the functional Magnetic Resonance Imaging (fMRI) provides us with powerful tools to construct connectivity patterns between brain regions. Such techniques paves the way to the study of "connectome" in the human brain, which is a comprehensive mapping of the interactions and connections between different brain regions. It inspires us to regard the brain as an interconnected network where each individual functional modality or brain area is the node, and the connectivity between areas serves as the edge. Hence, if the connectivity between brain areas is perturbed in patients with neurological or psychiatric disorders, the graph structure of the patient's brain is highly likely to differ from that of a healthy individual. The problem of diagnosing psychiatric diseases can therefore be framed as, given the graph structure of the brain network obtained from fMRI scan, how can we faithfully discriminate a psychiatric patients from healthy individuals? From a network analysis's perspective, a binary classification of brain networks into "typically developed" class and "functionally impaired" class via graph classification techniques can be used to solve this problem.

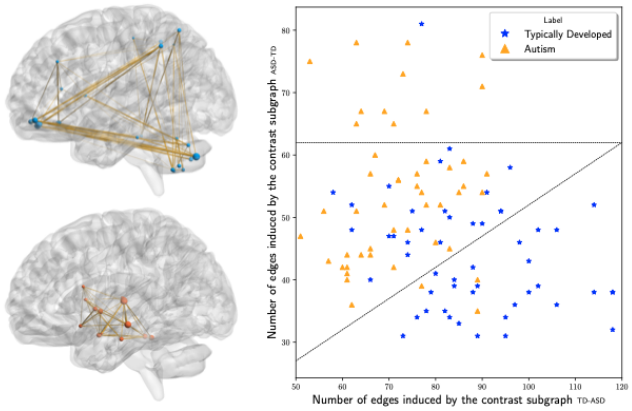

Figure 1. Example of real-world contrast subgraphs extracted from a brain-network dataset consisting of 49 children affected by Autism Spectrum Disorder (class ASD) and 52 Typically Developed (class TD) children. Top-left: the contrast subgraph TD-ASD. Bottom-left: the contrast subgraph ASD-TD. On the right, scatter plot showing for each individual the number of edges present in the subgraph induced by the contrast subgraph TD-ASD (x-axis) and by the contrast subgraph ASD-TD (y-axis). [6]

In this paper, we will first focus on improving a highly interpretable graph feature extraction algorithm called "Contrast Subgraph"[6], in an effort to extend its originally proposed computational framework, improve the resulting classification accuracy, and map our experimentation findings back to neuroscience literature. A contrast subgraph is a set of vertices whose induced subgraph is densely connected in one classification group but sparsely connected in another group, which can be used to find the differences between two graphs. Figure 1 provides more concrete examples of this concept.

On the top-left figure, we can see the contrast subgraph that maximizes the difference between the edges in the class TD with respect to that in the class ASD (the TD-ASD subgraph). On the bottom-left figure, we see the ASD-TD contrast subgraph. To differentiate between the TD-ASD and ASD-TD

CHI'20, April 25–30, 2020, Honolulu, HI, USA

ISBN 978-1-4503-6708-0/20/04…$15.00

DOI: https://doi.org/10.1145/3313831.XXXXXXX

subgraph, the TD-ASD subgraph highlights the denser edges in the TD class, and the ASD-TD highlight those in the ASD class.

Results using contrast subgraphs can be easily analyzed, just as the scatter plot on the right. The x-axis represents the number of edges in the contrast subgraph TD-ASD, and the y-axis represents the number of edges in the contrast subgraph ASD-TD. Each instance in the scatter plot represents an individual's brain imaging data. This allows us to make the following observation: If the number of edges induced by the contrast subgraph ASD-TD is smaller than half of the number of edges induced by the contrast subgraph TD-ASD, then there are high chances that the individual is not affected by ASD. This example shows that results from contrast subgraphs are highly explainable in such a way that we can access which brain regions have the largest change in connectivity induced by the psychiatric disorder, and at which point the progression in connectivity change signifies the development of psychiatric disorder. In the following sections, we will explore and extend this algorithm and test using classification tasks.

In order to find contrast subgraphs, we will first calculate the difference graphs that are obtained by transforming the graphs into matrices, calculate the differences between the matrices, and use the results to establish difference graphs. By extracting the subgraph(s) with highest contrasts (i.e highest connectivity difference) from the difference graph, we will successfully find the contrast graphs. More details will be discussed in the Problem Definition and Methodology section.

Furthermore, we will also explore an embedding-based approach for feature extraction. To that effect, we will be vectorizing the graph representation of our dataset using state-of-the-art techniques graph2vec [10] and sub2vec[1]. As will be seen later, these methods are considerably less computationally hungry than our top-k method, but they instead suffer in the area of interpretability. We are interested in seeing if this avenue will offer a beneficial trade-off between interpretability and classification accuracy.

We will then conduct binary classification to evaluate the quality of these feature extraction methods. The results and findings will be discussed in later on sections.

RELATED WORKS

Brain Imaging Data Analysis

Previous research has proposed to use simple machine learning techniques, such as Support Vector Machine (SVM)[7], Linear Discriminant Classifier (LDC)[5], and logistic regression [11] to separate brain imaging data from psychiatric patients from typically developed individuals. However, these methods completely disregard the network structure of the brain, causing the loss of important information which may affect the classification accuracy.

Later on, the development of deep learning methods has drawn people's attention to its potential application to neuroimaging data analysis. There are successful cases where Deep Neural Networks (DNN) are applied to classify brain imaging data [9]. However, this model tends to have an abundance of parameters and is therefore prone to overfitting. In practice, the number of parameters in a DNN often far exceeds the number of neuroimaging samples we have, making the model hard to generalize to unseen data, in such a way that it becomes no longer meaningful to seek diagnosis advice on new patients from such model. In addition, using deep learning methods on plain brain imaging data also ignores the underlying network structure of the brain.

In another line of research, the researchers have also proposed to use graph embedding methods followed by a Convolutional Neural Network (CNN) model for binary classification on brain imaging data [8]. Despite an effort in exploiting the network structure of the brain and in reducing the number of model parameters, the classification results from this model lacks interpretability and explainability, which is of uttermost importance in diagnosing any type of disorders or diseases.

Lanciano et al. [6] designed a highly interpretable algorithm that extracts contrast subgraphs and classify brain networks based on the contrast subgraph. The details of this approach will be stated in the Problem Definition section. This approach simplifies the vast number of features related to the brain networks into only two scalar features for linear classification. It is shown to outperform several state-of-the-art graph embedding algorithms.

Graph Embedding Methods

We will be using two state-of-the-art graph embedding algorithms that allow graph classification on low-dimensional embeddings: graph2vec and sub2vec.

Graph2vec is an unsupervised representation learning technique that can learn fixed-length vector representations of arbitrary sized graphs [10] It views an entire graph as a document and subgraphs that rooted at each node as words that compose the document. Given a set of graphs, it considers the set of all rooted subgraphs around every node as its vocabulary. It uses document embedding techniques for graph embedding. After training, graphs with similar vocabularies will be embedded close to each other in the embedding space. Graph2vec has achieved significant improvements in graph classification accuracy over other representation learning techniques.

Sub2vec is an unsupervised feature learning techniques for subgraphs [1]. It was originally proposed to accommodate tasks such as community detection which are intuitively dependent on subgraphs. It learns the vector representations of subgraphs to preserve the local connectivity information or overall structure of the subgraphs. Here we adopt this technique to learn vector representations on whole graphs by treating them as the subgraphs of the union of all the graphs.

MOTIVATION

Despite outperforming other baseline methods, the classification accuracy of Contrast Graph drops significantly as the heterogeneity of the dataset gets larger, from 83% to nearly 60% on a heterogeneous dataset as shown by the original paper [6]. This suggests that the original approach which extracts only the subgraph with highest connectivity difference between two groups (i.e the top contrast subgraph) for linear

classification in a 2D space may not suffice for more complex classification tasks.

Therefore, in an attempt to improve the model's performance in face of data heterogeneity and task complexity, we propose to extend the original framework by extracting k subgraphs with highest connectivity contrasts between two groups (i.e the top k contrast graphs), and find a hyperplane to separate two groups of data in the high-dimensional feature space (we will limit k to preserve the interpretability of the classification outcomes). This way, we can strength the robustness of this approach, improve its overall performance, while preserving its explainability, thus providing a highly transparent and effective way of predicting psychiatric disorders from brain imaging data.

There is, however, also the possibility that a focus on interpretability might be hampering classification accuracy. To that effect, we will also be examining the embedding-based approaches mentioned above. These methods have promise of better performance while completely dropping any hint of interpretability.

OUR PROPOSAL AND CONTRIBUTION

In this project, we propose three objectives:

1. On the algorithmic side, we want to extend the original framework to extract top-k contrast subgraphs, and combine information from multiple contrast subgraphs for binary classification on brain networks, aiming to improve the classification accuracy.

2. On the application side, we want to first apply the top-k contrast subgraph method to the ABIDE dataset used in the original paper to classify Autism Spectrum Disorder patients and healthy individuals. Our aim here is to interpret the contrast subgraph extraction results to find the top-k brain regions that are most affected by Autism Spectrum Disorder. We also want to map our contrast subgraph extraction findings back to neuroscientific and psychiatric literature, to validate our model's ability to capture meaningful information about the underlying functional impairments of the brain under various psychiatric disorders.

3. We would like to also examine the performance of proven state-of-the-art graph embedding methods with respect to the task at hand, to see if they can manage a positive trade-off between interpretability and performance.

PROBLEM DEFINITION

Finding the single top contrast graph

Here we focus on extracting one contrast subgraph on brain networks $G = (V, E)$, which requires us to find the set of vertices whose induced subgraph has the densest connections in one class and the sparsest connections in another. The results will be used to classify brain networks and draw further conclusions, but that will not be the focus here.

Consider two sets of graphs $A = \{G_1^A, ..., G_{r_A}^A\}$ and $B = \{G_1^B, ..., G_{r_B}^B\}$, which in this context represents the typically developed group and the functionally impaired group. For these datasets, a vertex can represent a neuron, a functional modality, or a brain area, and an edge represents connectivity between two neurological units. The vertices will normally be the same or similar functional units measured across different patients, which allows us to draw general conclusions regarding human brain functionalities.

To summarize the high-level connectivity information about group A and group B, we introduce two *summary graphs* $G^A = (V, w^A)$ and $G^B = (V, w^B)$. The summary graphs are defined over the same vertices as the original group of graphs, while w^A, w^B are weight functions that assigns a real value to each pair of nodes representing the average connectivity density between the two nodes in group A and group B, respectively. In particular, for undirected unweighted graphs, w^A is defined to be the fraction of graphs $G_i^A \in A$ in which u and v is connected by an edge:

$$w^A(u,v) = \frac{1}{r_A} |G_i^A \in A \quad \text{s.t.} \quad (u,v) \in E_i^A|$$

The algorithm aims to find the subset of vertices $S^* \subseteq V$ that maximizes the contrast-subgraph objective, which is the absolute value of the difference in average-degree density between the two groups:

$$\delta(S) = \sum_{u,v \in S} (|w^A(u,v) - w^B(u,v)| - \alpha)$$

$$= |e^A(S) - e^B(S)| - \alpha \binom{|S|}{2}$$

where e^A is the sum of edge weights (or equivalently, the number of edges in unweighted graphs) in the subgraph of G^A induced by the set of vertices in S. The penalty term α controls the size of the subgraph to extract.

$$e^A(S) = \sum_{u,v \in S} w^A(u,V)$$

Finding the k-top contrast graphs

To improve performance, we want to find the k subgraphs with maximum aggregated difference in average degree density between two groups, while these k subgraphs cannot overlap with each other too much. This is quantified by the pairwise Jaccard coefficient between the sets of nodes of the k subgraphs. Therefore, we enforce an upper bound on the pairwise Jaccard coefficients between the k subgraphs.

The precise computational problem can be stated as:

$$S = (S_1, ..., S_k) = \arg\max_{S_1,...,S_k} \sum_{i=1}^{k} \delta(S_i)$$

$$\text{subject to} \quad \frac{|S_i \cap S_j|}{|S_i \cup S_j|} \leq \alpha \quad \forall S_i, S_j \subset S$$

$$\text{where} \quad \delta(S_i) = \sum_{u,v \in S} (|w^A(u,v) - w^B(u,v)| - \alpha)$$

is the penalized difference in average degree density between two classes for one single subgraph, as shown in the previous subsection of finding the single top contrast graph.

Previous work by Balulau et al.[2] proposed an efficient algorithm (MinAndRemove) and a simplified version of this algorithm (FastDSLO) for finding the k densest subgraphs within a graph. In the context of brain network classification, the problem of finding the top k contrast subgraphs between the two groups of graphs (typically developed VS functionally impaired) is mathematically equivalent to finding the k densest subgraphs of the difference graph $|w^A(u,v) - w^B(u,v)|$. Therefore, we adopt the algorithm proposed by Balulau et al. to solve our problem.

METHODOLOGY

Graph Preprocessing

In order to be used as input to our densest subgraph extraction algorithm, the unweighted undirected brain networks need to be processed to construct the difference graph. The difference graph is defined as

$$D = |w^A(u,v) - w^B(u,v)|$$

where w^A, $w^B$is the summary graph of the group of typically developed brain networks and the graph of functionally impaired brain networks, respectively. We calculate the edge weight of the summary graph of one graph as the fraction of times this edge appears in brain networks within this group, as demonstrated by the following equation:

$$w^A(u,v) = \frac{1}{r_A}|G_i^A \in A \quad \text{s.t.} \quad (u,v) \in E_i^A|$$

where $i = 1, 2, ..., r_A$ and r_A is the total number of brain networks within the group A. We then use the summary graphs of the two groups to construct the difference graph D, and extract the k-densest subgraphs on the difference graph D for further analysis.

Extract k-top contrast graphs

In order to extract top-k densest sub-graphs from the data, we first look at the FastDSLO algorithm.

The FastDSLO algorithm uses the linear programming (LP)-based algorithm for the densest sub-graph developed by Charikar[3] as the basis. The Charikar algorithm finds the current least connected node by the number of edges it has, and stores this node and the current average degree of the current graph. After all the nodes have been ranked, we have the order of connectivity and the current average degree of the subgraph for all the nodes. This means that the densest subgraph will be the subgraph that has the highest average degrees, and we can obtain this subgraph by removing all the nodes that have a lower connectivity rank. This enables us to find a 2-approximation of the densest subgraph from the current graph.

In FastDSLO, we set an α to represent the portion of least connected nodes to remove from the extracted subgraphs. Therefore, when α is 0, the entire subgraph calculated from the Charikar algorithm is the densest subgraph. When $0 < \alpha < 1$, we will remove α * m nodes from the densest subgraph calculated by the Charikar algorithm, with m being the total number of nodes in the Charikar densest subgraph. The reduced densest subgraph will be a more minimalized densest subgraph, which is more efficient in this case. In order to obtain the next densest subgraph, we will remove the currently found densest subgraph from the data, and then use the Charikar algorithm on the remaining graph again.

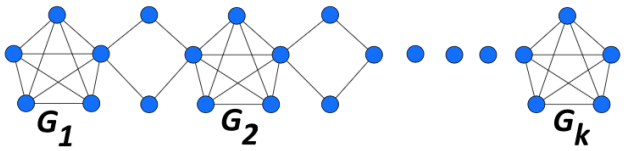

Figure 2. Worst case scenario using the Naive Charikar algorithm to find the densest subgraph.

FastDSLO

Algorithm 1 FastDSLO(G, k, α)

Input: A graph $G = (V, E)$, an integer $k > 0$, $\alpha \in [0, 1]$
Output: A list L of at most k subgraphs of G, $G_i = (V_i, E_i)$, s.t. the constraint on the pairwise Jaccard coefficient on the V_i's is not violated.
$L := \emptyset$
while $< k$ subgraphs are found and G is not empty **do**
  Find a 2-approximation solution $G_i = (V_i, E_i)$ to the densest subgraph problem by running the greedy algorithm in [9].
  $L := L \cup \{G_i\}$
  For each node v in V_i, let $\Delta_G(v)$ be the set of neighbors of v in G.
  Remove the $\lceil (1-\alpha)|V_i| \rceil$ nodes with minimum value $|\Delta_H(v) \setminus V_i|$ and all their edges from G.
end while
return L

Nevertheless, the Charikar algorithm doesn't guarantee a minimal solution. In the example in Figure 2[2], the graph has an average degree of 2, since there are 10k + 4(k - 1) edges and 5k + 2(k - 1) nodes, with k being the number of pentacles in the graph. If we run the Charikar algorithm on this graph, the result for the densest subgraph will be the entire graph. However, the minimal densest subgraph is the disjoint set of pentacles in the graph. Since we are trying to minimize the overlap of our top K densest subgraphs, this shows the potential for a poor output. Therefore, we need a more robust way to extract the densest subgraphs.

To that effect, we also make use of the MinAndRemove algorithm. This is essentially the same as FastDSLO with a minimizing process added. It achieves this by first running the Charikar algorithm[3] on the input graph to obtain a 2-approximation G', same as FastDSLO. However, in this case, it is only to speed up the following steps. As such, the algorithm then removes from G' all nodes with degrees less than its average density, while recording the densest resulting subgraph G''. Once that is done, in order to further minimize G'', the algorithm picks a node uniformly at random and, using the same linear programming method as Charikar[3], attempts to find a densest subgraph G''' of G'' which doesn't contain that node. If it manages to do so, a random node is picked from G''' and the process is repeated. Otherwise, there is then

at least one densest subgraph which contains that node. Using the same LP technique, we find every such subgraph and return the one with the smallest cardinality. This effectively minimizes the 2-approximation[2].

It should however be noted that while MinAndRemove outputs more optimal solutions compared to FastDSLO, the latter still holds merit in that it is substantially faster, and therefore is more appropriate for very large datasets. We will then be examining the efficacy of both.

MinAndRemove

Algorithm 2 MINANDREMOVE(G, k, α)

Input: A graph $G = (V, E)$, an integer $k > 0$, $\alpha \in [0, 1]$
Output: A list L of at most k subgraphs of G, $G_i = (V_i, E_i)$, s.t. the constraint on the pairwise Jaccard coefficient on the V_i's is not violated.
$L := \emptyset$
while $< k$ subgraphs are found and G is not empty **do**
  Find a minimal densest subgraph $G_i = (V_i, E_i)$ of G by running Algorithm 3
  $L := L \cup \{G_i\}$
  For each node v in V_i, let $\Delta_G(v)$ be the set of neighbors of v in G.
  Remove the $\lceil (1-\alpha)|V_i| \rceil$ nodes with minimum value $|\Delta_H(v) \setminus V_i|$ and all their edges from G.
end while
return L

Feature vectors

Our previous step finds the top-k densest subgraphs in the difference graph, which means that we have determined the top-k regions that show the highest connectivity change between the typically developed group and the functionally impaired group. Now we want to assess whether such local changes in connectivity can help us predict the functional state of the brain network. Therefore, we try to classify individual brain networks based on the connectivity information derived from the contrast subgraphs. Our aim is to evaluate the informativeness and the effectiveness of the contrast subgraphs extracted. We map the contrast subgraphs extracted from the difference graph to the nodes in individual brain networks. That is, we find the same set of nodes present in the individual network as in the contrast subgraphs we extracted. For each individual's network, we count the number of edges in each subgraph and use it as a feature for classification. We propose that from each contrast subgraph S we can constructed two features:

$$||G_i[S] - G^{[TD]}[S]||_1$$

$$||G_i[S] - G^{[ASD]}[S]||_1$$

where $G_i[S]$ represents the subgraph formed by nodes in the subgraph S in the i^{th} individual brain network, $G^{[TD]}[S]$ is the projection of the subgraph S in the summary graph of the typically developed group. Similarly for $G^{[ASD]}[S]$. Therefore, the number of features equals twice the number of subgraphs we want to extract.

EXPERIMENT SETUP

To test the effectiveness of our algorithm, we apply our K-top Contrast Subgraph framework to the ABIDE dataset, and compare its classification accuracy to several state-of-the-art graph classification algorithms such as graph2vec and sub2vec. We use graph classification accuracy as an evaluation metric for the informativeness and effectiveness of our K-top Contrast Subgraph approach and other graph embedding frameworks. Our rationale is that if these techniques could generate feature vectors that effectively preserve the structure and connectivity of the input graphs, then a difference in the original graphs could be projected to a difference in their feature vectors in the embedded space.

Dataset

We use neuroimaging data from Autism Brain Imaging Data Exchange (ABIDE) project [4] for experimentation. It contains open-source neuroimaging data from 573 typically developed (TD) individuals and 539 patients who suffered from ASD. The original dataset was further divided into 4 nonexclusive subgroups: Children, Adolescents, EyesClosed, and Male. The number of TD subjects and number of ASD subjects were balanced in each group.

The raw dataset was preprocessed first by parcellating the brain into 116 Regions of Interest (ROIs). The time series data of each ROI reflecting its functional activity patterns was extract and the pairwise Pearson correlation coefficient was computed between any pair of ROIs, producing a 116x116 correlation matrix. This correlation matrix is binarized with a threshold equal to 80-th percentile of the distribution of all correlation coefficients, which is a common practice in literature [7]. After this step, every pair of nodes (u, v) with value 1 in the correlation matrix is considered as connected by an edge in the brain network, and every pair of nodes with value 0 in the correlation matrix is considered as disconnected. The brain network of each human subject is therefore converted to an undirected unweighted graph with 116 nodes. The thresholded correlation matrix is equivalent to the adjacency matrix of the graph.

Table 1. Datasets used in the experiments

Dataset	Description	TD	ASD
Children	Age ≤ 9	52	49
Adolescents	Age in [15, 20]	121	116
EyesClosed	Eyes closed during scanning	158	136
Male	Male individuals	418	420

Tasks

We will run FastDSLO, MinAndRemove, graph2vec and sub2vec algorithms to generate feature vectors for each brain network. These feature which are then used as input to the SVM and the Decision Tree classifier, which will be used to help classify brain imaging data into TD or ASD. This allows us to evaluate the accuracy of feature vectors generated by top-K contrast subgraphs.

Algorithm Application

In our experiment, we will focus on four algorithms and compare their performance.

The MinAndRemove algorithm and FastDSLO algorithm were implemented in Java originally.[2] To make use of the extensive open-source resources implemented in Python, we first convert the original algorithms to python, and apply it on the ABIDE dataset.

For the graph2vec method, we used a Python implementation of the algorithms which takes the edge list and node degrees of each graph as input. The original ABIDE dataset represents each brain network in an adjacency matrix. Here we first construct a set of *.json* file that each stores the edge list and and node degrees of a brain network, and use the file folder as the input to the graph2vec algorithm. The table below shows the hyperparameters we use to generate embeddings for the ABIDE dataset using the graph2vec algorithm.

The sub2vec approach also takes the edge list of each graph as the input. As required by the algorithm, we convert the original dataset to a set of edge lists saved in *.txt* files and use as input to the algorithm.

We also tried to run the original Contrast Subgraph approach from Lanciano et al. (2020) to compare the performance of our K-top Contrast Subgraphs to the original framework of Contrast Subgraph. However, the code they provided requires setting up external fast linear programming solver CVX, which we weren't able to implement due to hardware restraints. Moreover, the authors implemented their algorithms in depracated softwares such as Python 2.7, MATLAB 2019a, and Networkx 1.9.1, while all three members of our team are using Python version newer than Python 3.6, MATLAB 2021a, and Networkx 2.6.1. The differences in Python syntax and Networkx function usage made it hard to debug the original code, and running MATLAB from MAC OS terminals faced severe obstacles due to system admin permission restraints. Therefore, we will compare the performance our our K-top Contrast Subgraph approach directly to the classification accuracy the authors reported in their paper [6]

The tables below shows the hyperparameters we use to generate graph embeddings using the FastDSLO algorithm, the graph2vec approach and the sub2vec approach, respectively.

Table 2. Hyperparameters of FastDSLO used in the experiment

Hyperparameter	Description	Value
k	Number of contrast subgraphs	5
α	Overlap between subgraphs	0.8

Table 3. Hyperparameters of MinAndRemove used in the experiment

Hyperparameter	Description	Value
k	Number of contrast subgraphs	5
α	Overlap between subgraphs	0.8

Table 4. Hyperparameters of graph2vec used in the experiment

Hyperparameter	Description	Value
Dimensions	Number of dimensions	128
Min-count	Minimal feature count to keep	1
Learning-rate	Initial learning rae	0.025

Table 5. Hyperparameters of sub2vec used in the experiment

Hyperparameter	Description	Value
Dimensions	Number of dimensions	128
Walk length	Length of random walk	150000
Iter	Training iterations	20

Evaluation via graph classification

We will use binary classification methods such as Support Vector Machine (SVM) and Decision Tree to evaluate the results from the previous algorithms. We choose these two classifiers because they have relatively few parameters and are more interpretable compared to other classification methods. In addition, for SVM, it finds an optimal boundary that best separates two groups of data and the weight parameter can be directly interpreted as the relative importance of individual contract graphs in terms of predicting the functional state of the brain. Similarly, for decision tree, the decision criteria at each decision point represents the determinants for the functional state of the brain.

We use design matrix from regression analysis as the input into the classification algorithms, where each row is one brain network sample, and each column corresponds to one of the feature vectors for classification. The number of features equals the number of contrast subgraphs extracted. For each contrast subgraph, we find their corresponding nodes in the individual brain networks. We then count the number of edges in the individual brain networks induced by this set of nodes as the value in the corresponding row and column, or in other words, the corresponding brain network and contrast subgraph. In general, we classify individual brain networks based on their connectivity density in each contrast subgraph.

For both approach, we randomly split all brain networks into 80/20 training/test sets. In the SVM approach, there is an additional step of hyperparameter tuning where we use 5-fold cross validation on the training set to select the best set of hyperparameters for the classifier. We then apply the trained classifiers to the test set and report the confusion matrix of the classification.

RESULTS

Contrast Subgraph Extraction

We first applied the FastDSLO algorithm and extracted the top 3 densest subgraphs of the difference graph. The result is shown in figure 3. As a proof of concept, the FastDSLO algorithm extracts k-top subgraphs ranked by the average degree.

These subgraphs correspond to the brain regions where the functional connectivity showed greatest changes. We want to

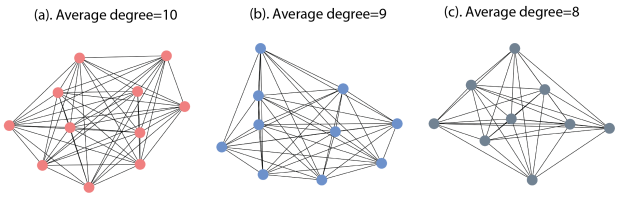

Figure 3. Densest subgraphs extracted from the difference graph. (a) the densest subgraph has an average degree of 10 (b) the second densest subgraph has an average degree of 9 (c) the third densest subgraph has an average degree of 8.

map these nodes back to the brain regions and compare our findings to the neuroscience literature. The original dataset was preprocessed and converted to unweighted undirected graphs by Lanciano et al (2020) [6]. However, the authors did not provide the labels of nodes in the graphs, so for now we have no access to the correspondence between node index and brain regions. We have emailed the authors of this paper but have not received a reply yet.

Performance Comparison

To evaluate the performance of graph classification algorithms, we can compare their accuracy and efficiency. In this experiment, we only compared the accuracy of each algorithm. We will not compare the efficiency here because our top-K contrast subgraph algorithm can be greatly improved with helper engines such as CVX. Due to hardware restraints, we weren't able to successfully implement these helper algorithms. More details will be discussed in the future works section.

The following table shows the accuracy of using each algorithm in the ABIDE dataset. Since we were unable to run the original contrast subgraph algorithm, the accuracy of the original Contrast Subgraph is obtained from [6], and this data is only used for reference here. [6] only provides accuracy for SVM classification, such no reference is provided for the Decision Tree classifier.

Table 6. Graph classification accuracy using SVM

	Children	Adolescents	Male	Eyesclosed
FastDSLO	0.55±0.05	0.51±0.04	0.54±0.03	0.51±0.07
MinAndRemove	0.62±0.04	0.65±0.03	0.67±0.02	0.70±0.03
*Contrast Subgraph	**0.86±0.07**	0.72±0.07	0.72±0.08	0.65±0.03
graph2vec	0.65±0.08	**0.97±0.03**	**1.0±0.0**	**1.0±0.0**
sub2vec	0.51±0.02	0.55±0.03	0.82±0.02	0.58±0.05

Table 7. Graph classification accuracy using Decision Tree

	Children	Adolescents	Male	Eyesclosed
FastDSLO	0.50±0.1	0.55±0.07	0.49±0.04	0.51±0.03
MinAndRemove	0.58±0.03	0.57±0.04	0.59±0.03	0.62±0.04
graph2vec	**0.58±0.1**	**0.93±0.03**	**1.0±0.0**	**0.97±0.01**
sub2vec	0.47±0.09	0.46±0.04	0.54 ±0.05	0.46 ±0.06

EVALUATION

This work gives us several interesting insights into brain network classification problem.

Firstly, although our implementation of K-top Contrast Subgraph fails to outperform the original framework on the children, adolescent and male datasets, it did manage to do it through MinAndRemove+SVM on the eyesclosed dataset. In fact, we can see that as the datasets get more heterogeneous with respect to the people they involve, the original framework's performance decreases while MinAndRemove's steadily increases to the point of outperforming the former. This suggests that our proposed method is more preferable for data with higher heterogeneity, when interpretability is important. However, its absolute performance is still quite low and more work is therefore required in that regard.

Secondly, across almost all combinations of datasets and classifiers, graph2vec provides the feature vectors that consistently achieved highest performance in downstream graph classification. It yielded a classification accuracy of >93% on the Adolescents, Male, and Eyesclosed dataset, and outperforms the other graph mining techniques on the Children dataset. All accuracy and standard deviation were obtained by running the algorithm 30 times. This demonstrates that the graph2vec is a highly robust and competent graph embedding framework that is well-suited for downstream graph classification tasks when. It should be noted that this comes at the cost of interpretability. On the other hand, the sub2vec approach only achieved chance level accuracy for the Children, Adolescents, and Eyesclosed dataset, while showed a significant improvement in accuracy on the Male dataset when using the SVM classifier (86%). The relatively poor performance of sub2vec may be due to that this framework was introduced to learn features for subgraphs, while we apply it on entire graphs and treat them as the "subgraphs" of the joint of the set of whole graphs. Our experiment suggests that sub2vec is less suited for brain network classification task compared to graph2vec.

Thirdly, it is surprising to find that the SVM have consistently higher classification accuracy across all subcategories of the dataset and no matter which technique we use to extract the feature vectors. SVM classifier assumes linear separability of the input classes, which Decision Tree relaxes such constraint and performs well for nonlinear separable inputs. One possible explanation is that the decision boundary for the ABIDE dataset can be well parameterized by a hyperplane in SVM, while a sequence of binary decisions in the Decision Tree is hard to capture the global structure of the decision boundary and may reaches the maximum tree depth before classifying the input data into homogeneous sets.

Lastly, the graph embedding approaches (graph2vec and sub2vec) achieves high performance for larger datasets. It is demonstrated by the improved classification accuracy on the Adolescents, Male, Eyesclosed dataset compared to the Children dataset. It it highly likely that larger dataset provides more training data to the algorithm, so that graph2vec and sub2vec could learn embeddings that better preserve the graph structure of the input graphs. Another possible explanation is that the functional connectivity of a child's brain network ex-

hibits extensive differences to an adult's brain network. Also, due to less exposure to the sensory stimuli and motor demands in a social setting, the functional connections between some of the brain regions may still yet to be established. Because we are using unweighted graphs in our experiment, two brain regions are either linked by an edge, or not. Suppose ASD reduces the connectivity between certain brain regions, then it would be intrinsically hard to detect a difference in connectivity if these regions showed no connectivity in the first place. Further work could be done using weighted graphs of brain networks.

DISCUSSION

Next steps & Future works Much work can be done to refine our current study. Above all, our MinAndRemove algorithm solves large linear programming problems that involves thousands of variables. As a consequence, it usually takes several minutes to extract each contrast subgraph. To tackle this problem, it is ideal to set up an external fast linear programming solver, such as CVX or Gurobi. After this step, we would be able to compare the runtime efficiency and scalability of the methods we compared in our experiment. Also, we could re-preprocess the raw fMRI data of the ABIDE dataset into undirected weighted graphs or directed weighed graphs, and compare the classification accuracy between these three types of graphs. Besides, it's worth further exploring the structure and connectivity of the subgraphs we extracted using the K-top Contrast Subgraph algorithm, in order to understand its low accuracy in brain network classification tasks. Furthermore, in order to establish an explainable brain network classification approach, we need to look into the weight parameters of the plane orthogonal to the fitted separating plane of the SVM classifier. These weight parameters can be directly translated to the relative importance of the multiple contrast subgraphs we extract. In this way, we would know which brain regions might be the most critical in predicting the autisum specturm disorder. However, at this moment, we still await a response from the authors of Lanciano et al. (2020) to provide us with the correspondence between the graph nodes and the brain regions.

CONCLUSIONS

Graph classification using features obtained from graph mining algorithms or graph embedding methods enables us to predict psychiatric disorders from the functional connectivity of the brain network. In Lanciano et al (2020), the authors proposed to employ contrast-subgraph based features to do binary classification on the brain networks of ASD patients and typically developed people. In this paper, we extended the framework of Contrast Subgraph by extracting k-top contrast subgraphs from the dataset, in addition to also applying an entirely different embedding-based approach using graph2vec and sub2vec. We then compared their classification accuracies to the original Contrast Subgraph approach.

Though our proposed MinAndRemove's performance is relatively low and leaves room for improvement, our results do suggest that it's actually more preferable for data with increased heterogeneity if interpretability is of value. Interestingly, sub2vec, as a state-of-the-art graph embedding method, showed little promise in our experimental setting. The most prominent finding of our experiment is that the graph embeddings generated by graph2vec consistently achieves the highest accuracy among all algorithms, at the cost of interpretability.

REFERENCES

[1] Bijaya Adhikari, Yao Zhang, Naren Ramakrishnan, and B Aditya Prakash. 2018. Sub2vec: Feature learning for subgraphs. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 170–182.

[2] Oana Denisa Balalau, Francesco Bonchi, TH Hubert Chan, Francesco Gullo, and Mauro Sozio. 2015. Finding subgraphs with maximum total density and limited overlap. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. 379–388.

[3] Moses Charikar. 2000. Greedy Approximation Algorithms for Finding Dense Components in a Graph. *Approximation Algorithms for Combinatorial Optimization* (2000).

[4] Cameron Craddock, Yassine Benhajali, Carlton Chu, Francois Chouinard, Alan Evans, András Jakab, Budhachandra Singh Khundrakpam, John David Lewis, Qingyang Li, Michael Milham, and others. 2013. The neuro bureau preprocessing initiative: open sharing of preprocessed neuroimaging data and derivatives. *Frontiers in Neuroinformatics* 7 (2013).

[5] Tobias Kaufmann, Kristina C Skåtun, Dag Alnæs, Nhat Trung Doan, Eugene P Duff, Siren Tønnesen, Evangelos Roussos, Torill Ueland, Sofie R Aminoff, Trine V Lagerberg, and others. 2015. Disintegration of sensorimotor brain networks in schizophrenia. *Schizophrenia bulletin* 41, 6 (2015), 1326–1335.

[6] Tommaso Lanciano, Francesco Bonchi, and Aristides Gionis. 2020. Explainable Classification of Brain Networks via Contrast Subgraphs. (2020).

[7] Anton Lord, Dorothea Horn, Michael Breakspear, and Martin Walter. 2012. Changes in community structure of resting state functional connectivity in unipolar depression. (2012).

[8] Lu Meng and Jing Xiang. 2018. Brain network analysis and classification based on convolutional neural network. *Frontiers in computational neuroscience* 12 (2018), 95.

[9] Muhammad Faiz Misman, Azurah A. Samah, Farah Aqilah Ezudin, Hairuddin Abu Majid, Zuraini A. Shah, Haslina Hashim, and Muhamad Farhin Harun. 2019. Classification of Adults with Autism Spectrum Disorder using Deep Neural Network. *2019 1st International Conference on Artificial Intelligence and Data Sciences (AiDAS)* (2019), 29–34.

[10] Annamalai Narayanan, Mahinthan Chandramohan, Rajasekar Venkatesan, Lihui Chen, Yang Liu, and Shantanu Jaiswal. 2017. graph2vec: Learning distributed representations of graphs. *arXiv preprint arXiv:1707.05005* (2017).

[11] Mehdi Rahim, Bertrand Thirion, Claude Comtat, and Gaël Varoquaux. 2016. Transmodal learning of functional networks for Alzheimer's disease prediction. *IEEE journal of selected topics in signal processing* 10, 7 (2016), 1204–1213.

[12] Neil D Woodward and Carissa J Cascio. 2015. Resting-state functional connectivity in psychiatric disorders. *JAMA psychiatry* 72, 8 (2015), 743–744.