Centralities in Simplicial Complexes

Anton Smerdov, Yury Biktairov, Konstantin Sobolev
Skolkovo Institue of Science and Technologies

21/3/19

Contents

Abstract

Graph can be seen as a subset of the more general object named simplicial complex, which consists of elements of higher-orders (i.e. triangles, tetrahedras, etc.), in addition to nodes and edges. Practically, this allows to encode higher-order interactions, while graphs only encode pairwise ones. Concept of node centrality of graphs can be generalized to higher-order elements of simplicial complexes. As a result, one could extract additional higher-order local information from a simplicial complex model, such as degree, closeness, betweenness, eigenvector, and subgraph centrality for simplicial complexes.

In the following work we implemented extraction of these high level graph features and conducted experiment on EEG binary classification problem in order to asses impact of new high level features. Our approach is primarily based on [1].

1 Introduction

In many of real-world machine learning problems, the input data is in the form of graphs, and the task is to predict either node and edge labels or graph labels.

For example, in the problem of exploration of protein-protein interactions each node of the graph is a protein, and the edges indicate whether two proteins interact. However, the study of the interaction of two proteins is expensive in terms of the cost of the required resource. Therefore, the problem is to predict the presence of edges among proteins with regard to some known edges, so biological researchers can study only the interaction of proteins, since their algorithm predicts the existence of an edge between them. This problem may also be considered as a problem of label classification, suggesting the presence of a complete graph among proteins and classifying each label as existing or not present.

Community detection, especially in social analysis, is very popular for understanding the structure of the network and how it grows. This problem is similar to the classification of each node to be a member of a community or not. However, the general problem is unsupervised, since neither the membership nor the number of communities are known in advance, but in some cases we already know the label of some nodes and want to predict the labels of the others.

Graph data description is the conventional way to describe the system with many interacting parts. This approach captures many of the complex structural and dynamical properties of the systems being studied. This method, however, fails to represent interactions of higher order than pairwise ones [2]. This drawback has lead to increased attention of many scientists to the area of simplicial complexes [3-5].

Let us introduce the main concepts of the framework we are going to use.

1.1 Simplices and simplicial complexes

First of all, let us introduce the generalisation of the graph's edge – a simplex.

Definition 1. Let V be a set of nodes or vertices. Then a $k-simplex$ is a set $v_0, v_1, ..., v_k$ such that $vi \in V$ and $v_i \neq v_j$ for all $i \neq j$. *A face of a k-simplex* is a $(k-1)-$simplex of the form $v_0, ..., v_{i1}, v_{i+1}, ..., v_k$ for $0 \leq i \leq k$.

Simplex is meant to represent a set of objects interacting as a whole. In general, however, such situation usually implies that any subset of this objects is also interacting as a whole. In order to comply with this sensible observation, we provide the following definition.

Definition 2. A *simplicial complex* C is a collection of simplices such that if a simplex S is a member of C then all faces of S are also members of C.

In other words, in simplicial complexes simplices are closed under operation of taking subsets. Such structures will be our main object of interest further on.

1.2 Adjacency

The main goal of the provided approach is to use usual graph centralities in context of simplices. In order to do it we should define the notion of their adjacency. We are only going to analyse adjacency of equally sized simplices, so it should be clear now that the final result is going to be a separate adjacency matrix for every size presented in a certain complex.

Firstly, we will introduce two additional definitions – lower- and upper-adjacency – and then construct the final one.

Definition 3. Let σ_j and σ_i be two k-simplices. Then, the two k-simplices are lower adjacent if they share a common face. That is, for two distinct k-simplices $\sigma_j = v_0, v_1, ..., v_k$ and $\sigma_i = w_0, w_1, ..., w_k$ then σ_j and σ_i

are lower adjacent if and only if there is a (k1)-simplex $\tau = x_0, x_1, ..., x_{k1}$ such that j and $\tau \in \sigma_i$. We denote lower adjacency by $\sigma_j \smile \sigma_i$.

Definition 4. Let σ_j and σ_i be two k-simplices. Then, the two k-simplices are upper adjacent if they are both faces of the same common (k + 1)-simplex. That is, for $\sigma_j = v_0, v_1, ..., v_k$ and $\sigma_i = w_0, w_1, ..., w_k$ then σ_j and σ_i are upper adjacent if and only if there is a(k+1)-simplex $\tau = x_0, x_1, ..., x_{k+1}$ such that $\sigma_j \in \tau$ and $\sigma_i \in \tau$. We denote the upper adjacency by $\sigma_j \frown \sigma_i$.

Clearly, if two simplices are upper-adjacent, they are necessarily lower-adjacent too.

Now it is time to speculate on which definition is preferable to be used for the analysis. The lower adjacency is not sensitive to the existence of the higher order simplices, so it is not sufficient on it's own. The upper adjacency only connects faces of existing simplices, so it is likely to lose important structural information. So, the simplest natural way to address both of this issues is to uses the combination of this definitions. Namely, use $A = A_L - A_U$ as adjacency matrix, where A_L is lower-adjacency matrix and A_U is upper-adjacency matrix.

Definition 5. Let i and j be two k-simplices in a simplicial complex. Then, for $k \geq 1$ the adjacency matrix A_k at the k-level in the simplicial complex has entries defined by

$$(A^k)_{ij} = \begin{cases} 1, & \text{if } \sigma_j \smile \sigma_i. \text{ and } \sigma_j \not\prec \sigma_i, \\ 0, & \text{if } i = j \text{ or } \sigma_j \not\succ \sigma_i \text{ or } \sigma_j \frown \sigma_i, \end{cases}$$

for k = 0 the adjacency matrix shall be given by the upper adjacency matrix.

1.3 Clique complex

The last essential question is how to obtain a meaningful simplicial complex, which will represent the system well enough. One way is based on the graph representation of the interactions within the system and called clique complex of a graph.

Definition 6. A *clique* of some graph is it's complete induced subgraph.

Clearly, any induced subgraph of a clique is also a clique, so cliques are closed under operation of taking any subset and, therefore, can be used as simplices of a proper complex.

Definition 7. A *clique complex* is a simplicial complex formed from a network as follows. The nodes of the network become the nodes of the simplicial complex. Let X be a clique of k nodes in the network. Then, X is a $(k-1)-$simplex in the clique complex.

Clique complexes is exactly what we used in our experiments.

2 Experiment

2.1 Data description

Dataset is an EEG data for 100 participants. Each sample consists of correlation matrix for signals from 117 sensors of EEG and binary label - has participant depression or not.

In order to convert convert data to a graph, we convert correlation matrix to adjacency matrix by binarizing elements using adjusted threshold h and removing self-connections of nodes (set diagonal elements to zero).

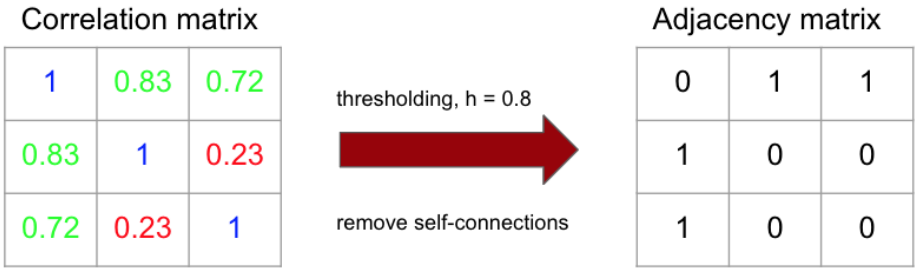

Figure 1: Data transformation scheme

2.2 Features description

We use global features describing a graph as features for baseline model.

- Number of edges
- Average global efficiency - average efficiency of all pairs of nodes
- Average local efficiency - the average of the local efficiencies of each node

As advanced features, we used medians of the following features from the paper:

Characteristics name	Formula	Description
Simplicial Closeness	$C(F) = \frac{1}{\sum_{Y \neq F} d(Y,F)}$	Reciprocal of simplicial farness, $d(Y,F)$ - shortest path from Y to F
Simplicial Betweenness	$g(F) = \sum_{S \neq F \neq T} \frac{\sigma_{ST}(F))}{\sigma_{ST}}$	σ_{ST} # of shortest paths between S and T, $\sigma_{ST}(F)$ # --//-- frough F
Simplicial Degree	$\delta_k = \sum_i A_{i\bullet} = \sum_j A_{\bullet j}$	number of other k-simplices to which s is adjacent
Clustering Coefficient	$c_u = \frac{2T(u)}{deg(u)(deg(u)-1)}$	where T(u) is the number of triangles through node u and deg(u) is the degree of u
Average Degree for Neighbours	$k_{nn,i} = \frac{1}{\lvert N(i)\rvert} \sum_{j \in N(i)} k_j$	N(i) are neighbours of node i and kj is the degree of node j which belongs to N(i)

Figure 2: Advanced features description

These features are computed for every simplex in the graph. However, for feature we select only median of each feature. This is because median is one of the most robust statistics which are easy to compute.

2.3 Evaluation

Since it's a balanced binary classification, we used ROC AUC as an evaluation metric. We used four different algorithms to test our new features:

1. Logistic regression.
2. KNearestNeighbours with $k = 3$.
3. Random forest with max depth 3 and number of estimators 100.
4. Support Vector Classifier with RBF kernel.

Scores for various algorithms are presented in Table 2. The second level interaction features seem to be more relevant.

We suppose that some of the features might be redundant and decided to selected the most important of them, in order to make models more stable and understand the relevance of the features. In order to perform feature

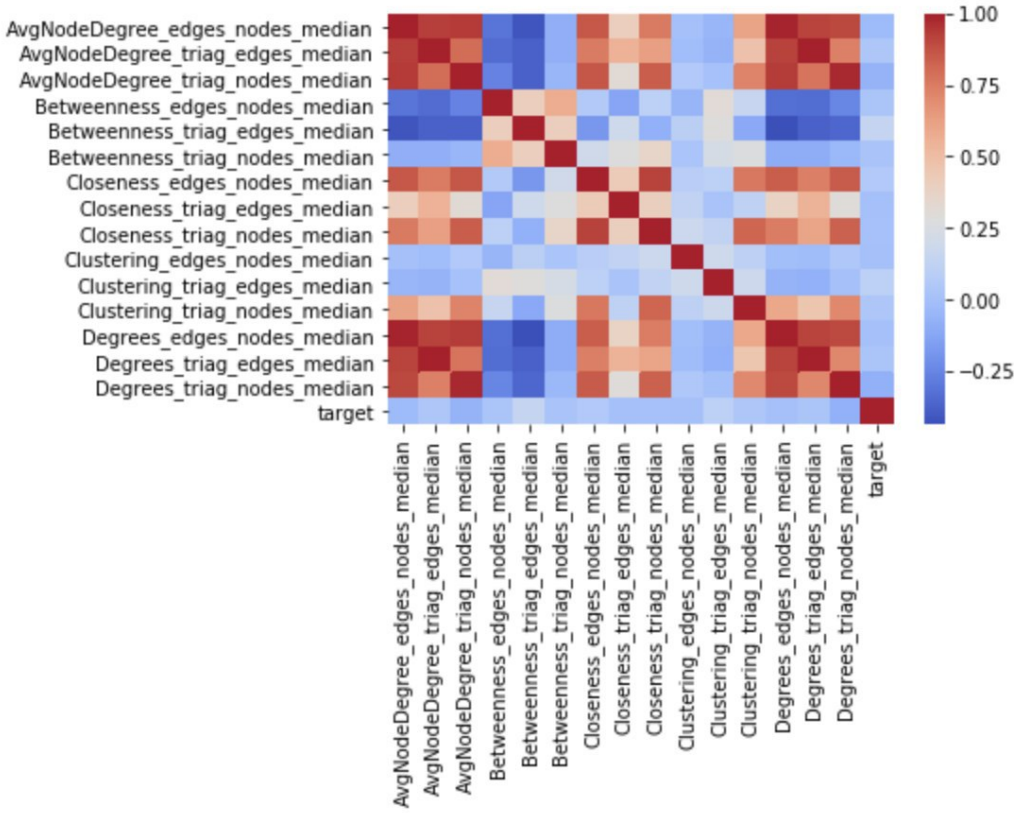

Figure 3: Features correlation matrix

Model name	1st level features	2nd level features	3rd level features
Logistic regression	0.543	**0.638**	0.572
Random Forest	0.511	0.591	**0.597**
SVC	0.552	0.540	0.570
KNN	**0.555**	0.465	0.505

Table 1: Scores for the first, second and third level features.

selection we used Lasso with coefficient 0.03. It selected 2,4 and 6 features for the 1st, 2nd and the 3rd levels respectively.

Model name	1st level features	2nd level features	3rd level features
Logistic regression	0.598	0.637	0.636
Random Forest	0.526	**0.659**	**0.703**
SVC	**0.599**	0.578	0.599
KNN	0.447	0.523	0.548

Table 2: Scores after feature selection

As a result, models based on the selected features perform better than on all the features. Scores after feature selection are presented in Table 2, the most important features are listed in 3. Interestingly, now the best quality is achieved for the 3rd level features. That proves that our algorithm is able to extract important high-order information from the data.

1st level features	2nd level features	3rd level features
Degrees 1 lvl	Betweenness 2 lvl	AvgNodeDegree 3.1 lvl
Global efficiency 1 lvl	Degrees 1 lvl	Closeness 3.2 lvl
	Local efficiency 2 lvl	Clustering 3.1 lvl
	Local efficiency 1 lvl	Clustering 3.2 lvl
		Global efficiency 3.1 lvl
		Local efficiency 3.1 lvl

Table 3: The most important features

3 Implementation

We have implemented a python code that performs graph data analysis and feature extraction. All experiments were conducted also in Python. Code for feature extraction, feature selection and experiments is available at link: repository link.

4 Conclusions

In this project we have implemented the algorithm proposed in the paper and extracted the high-order features from EEG data for patients with or without depression. The new method of feature extraction provides significantly better results than the old one and improves ROC AUC score from 0.6 to 0.7. Finally, we have figured out that new high level features are among the most important features of all classification models in our experiment and once again convinced that the method proposed in the article has a positive effect on classification score.

5 References

[1] E. Estrada, G. Ross, Centralities in Simplicial Complexes. Applications to Protein Interaction Networks, Journal of Theoretical Biology (2017)

[2] T.S. Evans, Clique graphs and overlapping communities. Journal of Statistical Mechanics: Theory and Experiment 2010(12), 12037 (2010)

[3] G. Bianconi, C. Rahmede, Network geometry with flavor: from complexity to quantum geometry. Physical Review E 93(3), 032315 (2016)

[4] Z. Wu, G. Menichetti, C. Rahmede, G. Bianconi, Emergent complex network geometry. Scientific reports 5 (2015)

[5] G. Bianconi, C. Rahmede, Emergent hyperbolic network geometry. Scientific Reports 7 (2017)