GRAPH CONNECTIVITY AND VECTOR DEGREE IN MOTIF-BASED GRAPHS

BY

QIHAO WANG

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2020

Urbana, Illinois

Adviser:

Professor Kevin Chen-Chuan Chang

ABSTRACT

Graph theory is frequently used to model and encode relationships in social networks, the internet, or biology. Due to the fundamental limitation that an edge can only connect two nodes, a standard graph is only capable of describing pairwise relationships. Therefore, some new models have been introduced to capture the relationships among more than two nodes by replacing standard edges with novel representations, like motif-edge or hyperedge. By replacing standard edges with motif-edges, the graph succeeds in describing the higher-order complex networks. However, a problem of connectivity in motif-based graphs also arises because generated motif-based graphs can be disconnected even though the standard graphs are connected. Here we first survey existing works on connectivity metrics in standard graphs, then propose a new measurement of degree, called *vector degree*, to extend the definition of degree from standard graphs to motif-based graphs. Some properties of vector degree will be compared with standard degree, and some connectivity metrics will be extended to motif-based graphs as well.

ACKNOWLEDGMENTS

The author thanks Professor Kevin Chen-Chuan Chang and Hongtai Cao for their comments and suggestions on this thesis. The University of Illinois at Urbana-Champaign provided necessary educational and experimental environments for the development of this thesis. ECE Editorial Services at the University of Illinois at Urbana-Champaign gave helpful suggestions on the format and grammar for this thesis.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

A graph is a tuple (E, V), which can be used to describe a pairwise relationship. V is the set of vertices in the graph, and E is the set of edges used to connect vertices. Let n be the number of vertices in the graph, and m be the number of edges. Edges can be either directed or undirected. In a social network, the relationsips may not necessarily be pairwise. In order to capture more general relationships among more than two nodes, a motif-based graph [1] G_M is generated from the normal graph. A motif M is a small subgraph pattern that occurs frequently in the original graph. The size of a motif is the number of nodes in the motif. For example, triangle is a frequently used motif in undirected graphs, and it has a size of three. Then in a motif-based graph, all edges are replaced by a motif-edge: if two vertices are in the same motif instance, then a motif-edge will connect these two vertices. Since a motif typically involves more than two nodes, a motif-edge can connect more than two vertices at the same time. As a result, a motif-based graph is able to capture the relationships among more than two vertices. Constructing a motif-based graph from a standard graph requires three steps.

1. All motifs in the standard graph should be found using subgraph isomorphism algorithms.
2. For each found motif instance, each pair of nodes in this motif instance will be connected by a standard edge. If there has already been an edge between two nodes, then the weight of this edge will be increased by one.
3. After all motif instances are processed, an undirected, weighted graph will be constructed. The adjacency matrix for this graph will be called *motif-adjacency matrix*.

For instance, if the graph on the left in Figure 1.2 is the standard graph and M_9 in Figure 1.1 is used to generate a motif-based graph, then the graph

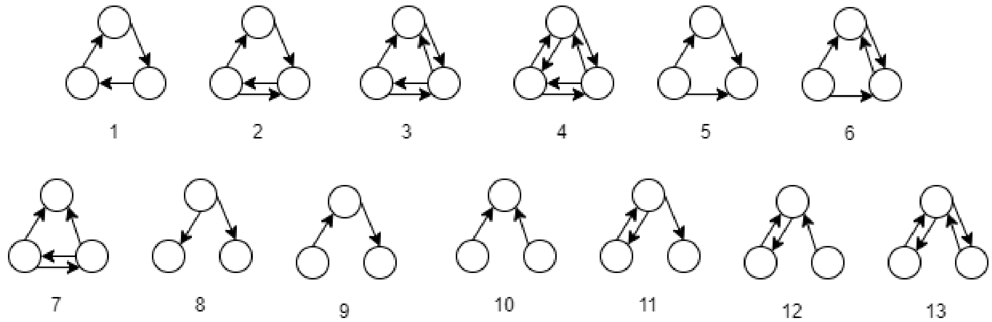

Figure 1.1: All 13 motifs of size 3 [1]

on the right in Figure 1.2 will be expected output. The number on the edges will be weights, and the adjacency matrix for this graph will be the corresponding motif-based adjacency matrix.

After constructing the motif-adjacency matrix, some algorithms in standard graphs can also be applied to solve relevant problems, like a partitioning problem [1]. In standard graph settings, normalized Laplacian matrix is used to accomplish this task. The second smallest eigenvalue of Laplacian matrix and its corresponding eigenvector can be used to partition the graph. However, the second smallest eigenvalue will be zero if the graph is disconnected. As a result, the graph should be connected if a Laplacian matrix is used to partition its vertices. Unfortunately, it is possible that a motif-based graph could be disconnected even though the original graph is connected. Therefore, graph connectivity is also an important topic in the area of motif-based graphs. In this thesis, some metrics used to measure standard graph connectivity and centrality in previous work are first introduced in Chapter 2. Since degree is required to calculate some metrics, a novel definition of degree in motif-based graphs, called *vector degree*, is proposed and analyzed in Chapter 3. In Chapter 4, some metrics related to degree will be extended to motif-based graphs using vector degree. In Chapter 5, related work will be discussed.

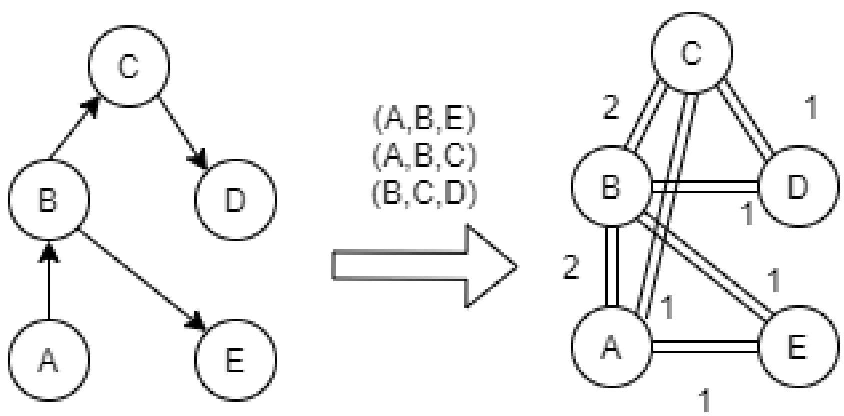

Figure 1.2: Generating motif-based graph from standard graph

CHAPTER 2

CONNECTIVITY AND CENTRALITY METRICS

In this chapter, several graph connectivity and centrality metrics will be introduced. Both connectivity and centrality are essential in analyzing social networks, and standard degree is one of the most frequently used centrality metrics. More connectivity metrics related to degree will be analyzed in Chapter 4 along with vector degree to extend their scope to motif-based graphs.

2.1 Local Clustering Coefficient

2.1.1 Definition

Local clustering coefficient was first introduced in 1998 by Watts and Strogatz [2] to describe the clusters in the graph. Each node will be associated with a coefficient to describe how far away it and its neighbors are from forming a complete graph. Let n_u be the number of neighbors of a node u, and m_u be the number of edges between the neighbors of node u. Then the local clustering coefficient of node u will be $C(u) = \frac{2n_u}{m_u(m_u-1)}$ for an undirected graph and $C(u) = \frac{n_u}{m_u(m_u-1)}$

2.1.2 Application

Local clustering coefficient was usually used as a metric to evaluate the connectivity in social networks. Similar nodes in a social network are believed to cluster together, so some algorithms try to discovery some relationships to make the original graph more heavily connected. For instance, Tian et al. [3] try to construct a heavily connected social network by recovering missing relationships, and local clustering coefficient is the metric used to evaluate the algorithm in this thesis.

2.2 Eigenvector Centrality

2.2.1 Definition

Eigenvector centrality, proposed by Newman [4], is one of the most frequently used centrality metrics in undirected and unweighted graphs. Since the calculation of this centrality is the same as solving an eigenvector problem, this centrality is called *eigenvector centrality*. Each node u will be assigned a score which is proportional to the sum of the scores of its neighbors, as shown in Equation 2.1.

$$C_u = \frac{1}{\lambda}\Sigma_{(u,v)\in E}C_v \tag{2.1}$$

After the rearrangement of Equation 2.1 and replacement of edges with adjacency matrix, Equation 2.1 will become Equation 2.2, where A is the adjacency matrix of the graph and C is the vector of the score of each node.

$$\lambda C = AC \tag{2.2}$$

The solution to Equation 2.2 is the same as solving the eigenvector value problem of an adjacency matrix.

2.2.2 Application

Eigenvector centrality is most frequently used as a metric to find influential users in a social network. For instance, Maharani et al. [5] use eigenvector centrality to find influential users in the Twitter network, then some marketing strategy can be applied.

2.3 Betweenness Centrality

2.3.1 Definition

Betweenness centrality was first introduced by Freeman [6] in 1977. For each node u, its betweenness centrality is related to the total number of shortest paths that use node u as an intermediate node. Let s and t be all destinations and terminals that are different from node u in the graph G. Since it is possible that there are multiple shortest paths from s to t, the total number of shortest paths from s to t is denoted by g_{st}. Let $g_{st}(u)$ be the total number of shortest paths from s to t that use node u as an intermediate node; then the betweenness centrality of node u is $B(u) = \Sigma_{u \neq s \neq t} \frac{g_{st}(u)}{g_{st}}$.

2.3.2 Application

Betweenness centrality is frequently used to find the influential users in a social network. It is capable of identifying some essential nodes in a network which may be ignored by other metrics. For instance, Hoppe and Reinelt [7] try to analyze the critical individual in a social network of leaders. An edge within a cluster is defined as a bond and an edge between different clusters is defined as a bridge. If other metrics like eigenvector centrality are used, the nodes connected to bridges may not receive significant attention because those nodes may not be heavily connected. Since there are relatively few bridges compared to bonds, nodes connected to bridges could be frequently used as intermediate nodes when the shortest paths between two clusters are computed. Therefore, betweenness is able to assign higher score to nodes which have relatively small degree.

2.4 Principal Component Centrality

2.4.1 Definition

Principal component centrality was first introduced by Ilyas and Radha [8] due to eigenvector centrality's failure to assign informative scores to all nodes. Eigenvector centrality tends to assign a high score to only a small fraction of nodes in the graph, so it cannot be used to analyze all nodes in the graph. Principal component centrality only cares about p eigenvectors of p highest eigenvalue. An $n * p$ matrix X_{n*p} will be formed such that each column of this matrix is an eigenvector of an adjacency matrix. After arranging these p eigenvectors into the $n * p$ matrix X_{n*p}, the principal component centrality will be solved as $C_p = \sqrt{(A_{n*n}X_{n*p}) \circ (A_{n*n}X_{n*p})\mathbf{1}_{n*1}}$, where $\mathbf{1}_{n*1}$ is a vector of all ones, and $\circ$ is the Hadamard product operation.

2.4.2 Application

Ilyas and Radha [9] leveraged principal component centrality later to find the influential nodes in a social network. Its performance is compared with eigenvector centrality to show the advantage of principal component centrality.

CHAPTER 3

VECTOR DEGREE

In Chapter 2, the calculation of some metrics requires the knowledge of degree. However, degree is not well defined in a motif-based graph. In an undirected graph, a scalar degree is used to count the number of incident edges. In a directed graph, two scalar degrees, an in-degree and an out-degree, are used to account for the direction of edges. In order to capture the connectivity in a motif-based graph, vector degree is proposed in Section 3.1. Then some properties of vector degree are analyzed in Section 3.2 to show that vector degree is a generalization of standard degree.

3.1 Definition of Vector Degree

Let $G_t = (V_t, E_t)$ be a motif of size k. Let f be a bijective function $f : V_t \mapsto \{1, 2, \ldots, k\}$. Then for each node u, its degree vector is calculated as follows: 1. Initialize its vector degree as a zero vector of length k. 2. For each motif instance that contains node u, if node u is matched to node v in the motif t, then the value of the vector degree at $f(v)^{th}$ position will be increased by 1. If more than one motif is used to construct the motif-adjacency matrix, then the vector degrees from different motifs can be concentrated to form the vector degree for all motifs. Supposing the graph has n nodes, then it will be possible to store the vector degrees of all nodes in an n by k table, which will be the degree table for the motif-based graph. For instance, Table 3.1 is the vector degree of the example given in Figure 1.2

Table 3.1: Vector Degree of Figure 1.2

Node	First Degree	Second Degree	Third Degree
A	2	0	0
B	1	2	0
C	0	1	1
D	0	0	1
E	0	0	1

3.2 Properties of Vector Degree

In order to compare the properties of vector degree with scalar degree, scalar degree will be converted to vector degree. Then the degree table will be an n by 2 table because an edge is a motif of size 2. In the directed graph case, each row of the degree table can be written as (in-degree, out-degree). In the undirected graph case, each row of the degree table can be written as (scalar-degree, scalar degree). Note that the number of motif instances will be twice the number of edges in the undirected graph because for each edge in the original graph, there are 2 possible mappings by changing the order of the matched nodes. For instance, an edge uv will exist in two edge motif instances: (u, v) and (v, u).

3.2.1 Degree Sum Formula

In standard graphs, the degree sum formula states the property of the summation over degrees of all nodes. The degree sum formula for undirected graph and directed graph is shown in Equation 3.1 and 3.2, respectively.

$$\Sigma_v deg_v = 2|E| \tag{3.1}$$

$$\Sigma_v InDegree_v = \Sigma_v OutDegree_v = |E| \tag{3.2}$$

A generalization of this property will hold when vector degrees are used to replace standard degrees. Let M be the number of motif instances found in the standard graph. Then for a vector degree of a motif of size k, the following properties also hold:

$$\Sigma_i DegreeTable[i][j] = M, \forall j \tag{3.3}$$

$$\Sigma_j \Sigma_i DegreeTable[i][j] = kM \tag{3.4}$$

Proof. Each column of the degree table represents a node in G_t. This column will sum up to M because M motif instances require the corresponding node in G_t to be mapped for M times. Since there are k columns, it will sum up to kM. □

If $k = 2$, then the properties in Equation 3.1 and Equation 3.2 are special cases of Equation 3.4 and Equation 3.3, respectively.

3.2.2 Maximum Value

In a standard graph, the degree of each node has an upper limit as shown in Equation 3.5 because a node will have a maximum degree value if it is connected to all remaining nodes in the graph. Note that Equation 3.5 holds for degrees in undirected graph, in-degree, and out-degree.

$$max(deg_v) \leq n - 1 \tag{3.5}$$

Each dimension of a vector degree also has an upper limit as shown in Equation 3.6.

$$max(VectorDegree_u[i]) \leq \Pi_{j=1}^{k-1}(n - j), \forall i, j \tag{3.6}$$

Proof. For a fixed node u and a motif of size k, Equation 3.6 can be proved by induction.

Base Case: When $k = 2$, the maximum value for each dimension is the same as Equation 3.5.

Inductive Hypothesis: $\forall l \leq k$, the maximum value for each dimension $\leq \Pi_{j=1}^{l-1}(n - j)$.

When $l = k + 1$, an additional node needs to be mapped in the original graph. Since k nodes have already been chosen in the previous k matchings, there will be $n - k$ candidate nodes. Therefore, the maximum value of each dimension in vector $\leq \Pi_{j=1}^{k-1}(n - j) * (n - k) = \Pi_{j=1}^{k}(n - j)$. □

Table 3.2: Dataset Used for Evaluation

Dataset	n	m
Facebook	4039	88234
P2P-Gnutella30	36682	88328
Web-NotreDame	325729	1497134

When $k = 2$, Equation 3.6 is the same as Equation 3.5.

In addition to Equation 3.6, vector degree also has upper limits for different norms, which will be used in Chapter 4 to extend some metrics in standard graphs to motif-based graphs. The upper bounds of L_1, L_2, and L_{inf} norm are shown in Equation 3.7, 3.8, and 3.9 respectively.

$$L_1 \leq k\Pi_{j=1}^{k-1}(n-j) \tag{3.7}$$

$$L_2 \leq \sqrt{k}\Pi_{j=1}^{k-1}(n-j) \tag{3.8}$$

$$L_{\text{inf}} \leq \Pi_{j=1}^{k-1}(n-j) \tag{3.9}$$

3.2.3 Vector Degree Distribution

In standard graphs, degree can have different kinds of distribution like normal distribution or power law distribution. Some datasets are evaluated to show that vector degree also has similar distributions. The algorithms used to find the motifs in the dataset is RI [10]. All datasets are collected from SNAP [11]. Table 3.2 summarizes the information of three datasets used in this section. The motifs used to generate the motif-based graph will be motifs 1 and 9 from Figure 1.1.

The Facebook dataset is a heavily connected dataset since it has a similar number of edges as the P2P dataset, although it has fewer nodes. Figure 3.1 shows that the vector degree in a heavily connected graph has a power law distribution.

The distribution of vector degree in the P2P dataset is also very similar to a power law distribution. Since the original dataset is not heavily connected, the P2P dataset has many fewer motif instances than the Facebook dataset. Therefore, the distribution in Figure 3.2 is not as clear as the distribution in Figure 3.1 since the sample size becomes smaller. However, even in a

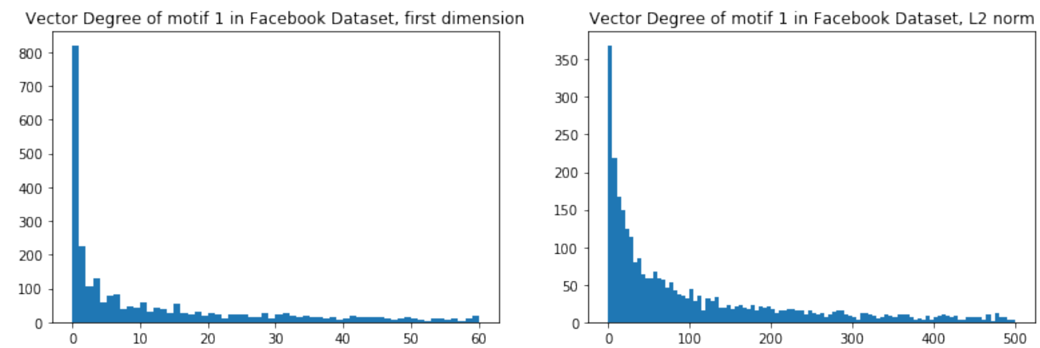

Figure 3.1: Distribution of vector degree in Facebook dataset

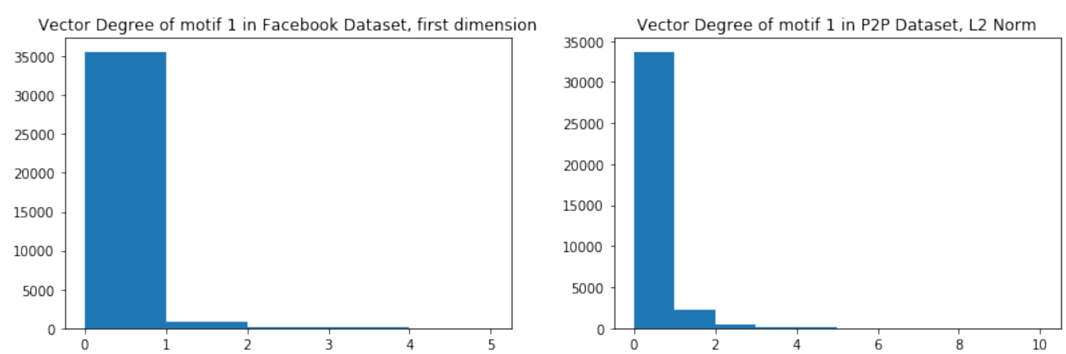

Figure 3.2: Distribution of vector degree in P2P dataset

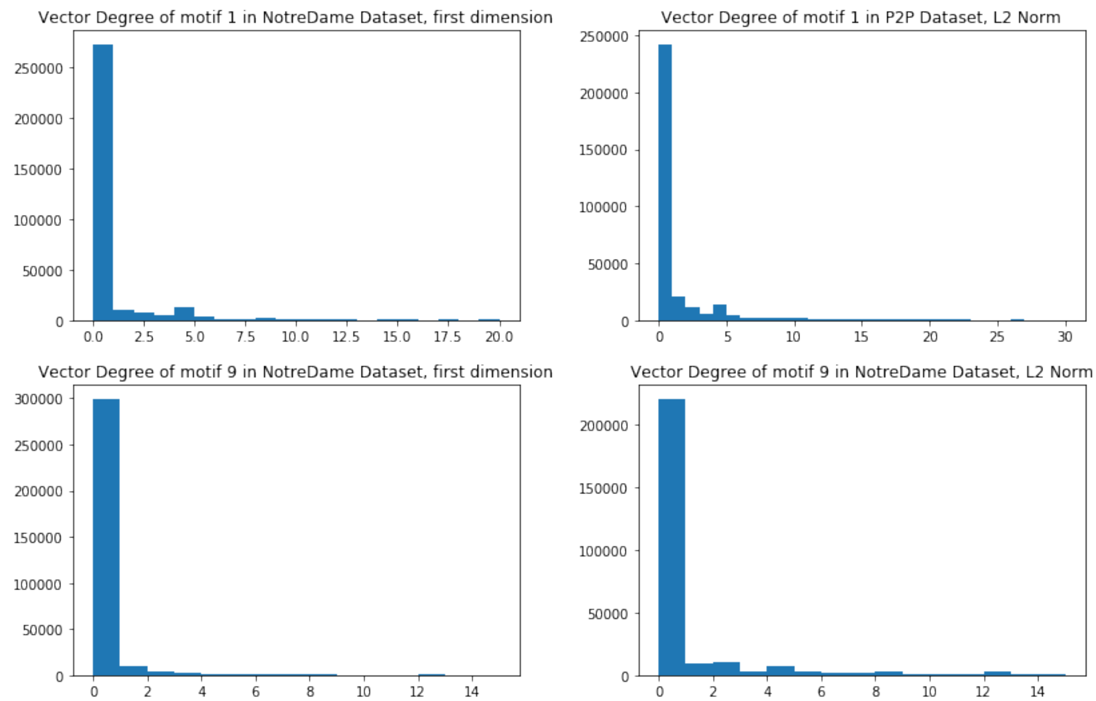

Figure 3.3: Distribution of vector degree in NotreDame dataset

weakly connected dataset, the distribution of vector degree still has a similar distribution to that of standard degree.

Finally, even in a much larger dataset, the distribution of vector degree is still a power law distribution, as shown in Figure 3.3. In addition to motif-1 in Figure 1.1, motif-9 is also used to make sure that this distribution also holds for other motifs.

CHAPTER 4

VECTOR DEGREE AND CONNECTIVITY METRICS

By defining vector degree, some metrics related to degree in standard graphs can be extended to motif-based graphs. L_1 norm will be used frequently in this section because L_1 norm equals the total number of motif instances that contain the target node.

4.1 Graph Density

In a standard graph, graph density is the number of edges divided by the maximum possible number of edges in the graph. The densities of directed and undirected graphs are shown in Equation 4.1 and 4.2 respectively. Density can take value of $[0, 1]$. A higher value of density means that the graph is more heavily connected.

$$Density(Directed) = \frac{m}{n(n-1)} \tag{4.1}$$

$$Density(Undirected) = \frac{2m}{n(n-1)} \tag{4.2}$$

By extending this definition to higher-order complex networks, the motif-based graph density is shown in Equation 4.3.

$$\frac{\Sigma_i \Sigma_j DegreeTable[i][j]}{kn\Pi_{j=1}^{k-1}(n-j)} \tag{4.3}$$

The numerator of Equation 4.3 will be kM according to Equation 3.4. Indeed, the numerator is the summation of the L_1 norm of all vector degrees in the graph. The denominator of Equation 4.3 is the summation of the maximum value of L_1 norm from Equation 3.7, which will be the maximum possible number of motif instances in the graph multiplied by k. Therefore, Equation 4.3 is an extension of graph density to motif-based graphs. If $k = 2$, Equation 4.3 becomes Equation 4.1 and 4.2. The additional factor of 2 in the undirected graph case comes from each edge being actually two motif instances, as discussed in Section 3.2.

4.2 Assortativity

Assortativity is a measure of the likelihood that two nodes of similar degree are connected. It can takes a value of [-1,1]. A positive value means that a node tends to connect to nodes with similar degrees. A negative value indicates that a node with high degree tends to connect with nodes with low degrees. The extension of assortativity is derived from the definition of assortativity in directed graphs [12]. In a directed graph, let $\alpha, \beta \in \{In, Out\}$. Then in a directed graph, for each directed edge $i = (source, sink)$, let j_i^α be the α-degree of the source, and k_i^β be the β-degree of the sink. The averages of j^α and k^β over all nodes are denoted by $\bar{j^\alpha}$ and $\bar{k^\beta}$ respectively. Then the assortativity of a directed graph is calculated using Equation 4.4.

$$r(\alpha, \beta) = \frac{\Sigma_i (j_i^\alpha - \bar{j^\alpha})(k_i^\beta - \bar{k^\beta})}{\Sigma_i \sqrt{(j_i^\alpha - \bar{j^\alpha})^2} \sqrt{(k_i^\beta - \bar{k^\beta})^2}} \tag{4.4}$$

In order to extend assortativity to a motif-based graph, the domain of α and β will be changed such that $\alpha, \beta \in \{1, 2, 3, \ldots, k\}$. Then the i^{th} degree will be valued at the i^{th} position of the vector degree. Assortativity in a motif-based graph has a meaning similar to that in a standard graph. For instance, if $\alpha, \beta = 3$ in a motif-based graph generated from motif 9 in Figure 1.1, then the assortativity indicates how often two nodes that are frequently used as terminals of paths are also connected by a path.

4.3 Motif-edge Connectivity

In a standard graph, edge connectivity is the number of edges required to be removed from the graph such that the original graph will be disconnected. Edge connectivity is related to degree because degree gives an upper bound of edge connectivity as shown in Equation 4.5.

$$EdgeConnectivity \leq min_u deg(u) \tag{4.5}$$

Assuming node v is the node with smallest degree, then the graph will be disconnected if all edges incident to node v is removed. After the removal of these edges, node v is isolated.

Motif-edge connectivity is the same as edge connectivity except that the motif-edges will be removed instead of the standard edges. The same inequality in Equation 4.5 also holds true if edge connectivity is replaced by motif-edge connectivity, and scalar degree is replaced by L_1 norm of vector degree.

$$MotifConnectivity \leq min_u \Sigma_{i=1}^{k} DegreeTable[u][i] \tag{4.6}$$

For each node u, its L_1 norm of vector degree is the total number of motif instances that contain node u. If this number of motif-edges are removed from the graph, then node u will be isolated. Therefore, vector degree also sets an upper bound for edge connectivity, which serves the same function as standard degree.

There is a similar connectivity metric related to vertex called *vertex connectivity*. Vertex connectivity is the number of vertices along with their incident edges needed to be removed from the graph such that the graph is disconnected. In both standard and motif-based graphs, Equation 4.7 shows the connection between motif/edge connectivity and vertex connectivity.

$$VertexConnectivity \leq Edge/MotifConnectivity \tag{4.7}$$

Proof. Let S be the set of edges/motifs needed to be removed from the graph such that the graph will be disconnected. Then $\forall e \in S$, the removal of a node that e is incident to will also remove e. Then by removing $|S|$ such vertices, the graph will be disconnected. □

4.4 Rich-club Coefficient

In standard graphs, the rich-club coefficient [13] is derived from the idea of rich-club. In a network, the rich-club is the set of vertices that are "rich" in a particular metric. For instance, a rich-club can be a set of vertices that have a degree higher than some constant γ. Then the rich-club coefficient is the same as density except that only vertices in the rich-club will be considered. The formula for calculating rich-club coefficient $C_{\Gamma,\gamma}$ for a metric Γ and constant γ is shown as Equation 4.8, where $V_{\Gamma\geq\gamma}$ and $E_{\Gamma\geq\gamma}$ are the set of vertices and the set of edges connecting these vertices that satisfy $\Gamma \geq \gamma$ respectively.

$$C_{\Gamma,\gamma} = \frac{2|E_{\Gamma\geq\gamma}|}{|V_{\Gamma\geq\gamma}|(|V_{\Gamma\geq\gamma}| - 1)} \tag{4.8}$$

In order to extend the rich-club coefficient to motif-based graphs, a metric Γ and a constant γ should be chosen. Γ can be a particular dimension of vector degree, or one of the norms of vector degree. Then let $V_{\Gamma\geq\gamma}$ have the same definition as for a standard graph. Define $E_{\Gamma\geq\gamma}$ as the set of motif-edges such that if each motif-edge e is treated as a set of vertices that are connected by e, then $\forall e \in E_{\Gamma\geq\gamma}, e \subset V_{\Gamma\geq\gamma}$ Let $|V_{\Gamma\geq\gamma}| = n_\Gamma$ and $|E_{\Gamma\geq\gamma}| = m_\Gamma$. Then the rich-club coefficient for motif-based will be calculated using Equation 4.9. If $k = 2$, Equation 4.8 will be the same as Equation 4.8.

$$C_{\Gamma,\gamma} = \frac{m_\Gamma}{n_\Gamma \Pi_{j=1}^{k-1}(n_\Gamma - j)} \tag{4.9}$$

CHAPTER 5

RELATED WORK

The idea of motif-based graph and motif-adjacency matrix was introduced by Benson et al. [1]. In order to construct a motif-based graph, many algorithms have been developed, like RI [10], VF3 [14], and LAD [15]. Motif-based graphs are frequently compared with hypergraphs, and the two are sometimes interchangeable. Zhou et al. [16], Benson et al. [1], and Rodríguez [17] related the idea of Laplacian matrix to higher-order complex graphs, and the problem of disconnection arises in the usage of a Laplacian matrix. Milencović and Pržulj [18] introduced the idea of using vectors to represent graphlet orbits, but their definition of graphlet signature only applies to the unique graphlet orbits. Therefore, the graphlet signature has no properties similar to those of standard degree discussed in Chapters 3 and 4. In addition, the graphlet signature is computed using graphlets, while vector degree in this thesis is computed using motifs.

Different centrality metrics are used in various papers to evaluate algorithms. These metrics include standard degree, local clustering coefficient [2], Eigenvector centrality [4], betweenness centrality [6], and principal component centrality [8]. These metrics are essential in analyzing social networks and evaluating network algorithms.

CHAPTER 6

CONCLUSION AND FUTURE RESEARCH

When constructing a motif-based graph, the problem of disconnection arises and it becomes necessary to measure the connectivity in motif-based graphs. A survey of the metrics of graph connectivity and centrality has been conducted, and degree plays an important role in some metrics. In order to extend these metrics to motif-base graphs, vector degree is proposed to replace the standard degree in higher-order complex networks. Some properties of vector degree are compared with those of standard degrees to show that vector degree is a generalization of standard degree in complex networks. Then some metrics are extended to motif-based graphs by replacing standard degrees with vector degrees.

For future research, the semantic meaning of vector degree will be further exploited. In addition, more properties should be analyzed to prepare for the applications in fields like clustering and classification. More metrics related to standard degree should be adapted to vector degree as well.

REFERENCES

[1] A. R. Benson, D. F. Gleich, and J. Leskovec, “Higher-order organization of complex networks,” *Science*, 2016.

[2] D. J. Watts and S. H. Strogatz, “Collective dynamics of Àsmall-world networks,” *Nature*, 1998.

[3] Y. Tian, Q. He, Q. Zhao, and X. Liu, “Boosting social network connectivity with link revival,” *International Conference on Information and Knowledge Management*, 2010.

[4] M. E. J. Newman, “The mathematics of networks,” *The New Palgrave Encyclopedia of Economics*, 2008.

[5] W. Maharani, Adiwijaya, and A. A. Gozali, “Degree centrality and eigenvector centrality in twitter,” *Proceedings of 2014 8th International Conference on Telecommunication Systems Services and Applications (TSSA)*, 2014.

[6] L. C. Freeman, “A set of measures of centrality based on betweenness,” *Sociometry*, 1977.

[7] B. Hoppe and C. Reinelt, “Social network analysis and the evaluation of leadership networks,” *The Leadership Quarterly*, 2010.

[8] M. U. Ilyas and H. Radha, “A KLT-inspired node centrality for identifying influential neighborhoods in graphs,” *Proceedings of the 44th International Conference on Information Sciences and Systems*, 2010.

[9] M. U. Ilyas and H. Radha, “Identifying influential nodes in online social networks using principal component centrality,” *Proceedings of IEEE International Conference on Communications*, 2011.

[10] V. Bonnici, R. Giugno, A. Pulvirenti, D. Shasha, and A. Ferro, “A subgraph isomorphism algorithm and its application to biochemical data,” *BMC Bioinformatics*, 2013.

[11] J. Leskovec and A. Krevl, “SNAP Datasets: Stanford large network dataset collection,” http://snap.stanford.edu/data, June 2014.

[12] J. G. Foster, D. V. Foster, P. Grassberger, and M. Paczuski, "Edge direction and the structure of networks," *Proceedings of the National Academy of Sciences*, 2010.

[13] S. Zhou and R. J. Mondragon, "The rich-club phenomenon in the internet topology," *IEEE Communications Letters*, 2004.

[14] V. Carletti, P. Foggia, A. Saggese, and M. Vento, "Introducing VF3: A new algorithm for subgraph isomorphism," *Graph-Based Representations in Pattern Recognition*, 2017.

[15] C. Solnon, "AllDifferent-based filtering for subgraph isomorphism," *Artificial Intelligence*, 2010.

[16] D. Zhou, J. Huang, and B. Schlkopf, "Learning with hypergraphs: Clustering, classification, and embedding," *Advances in Neural Information Processing Systems*, 2006.

[17] J. Rodríguez, "On the Laplacian eigenvalues and metric parameters of hypergraphs," *Linear Multilinear Algebra*, 2002.

[18] T. Milencović and N. Pržulj, "Uncovering biological network function via graphlet degree signatures," *Cancer Informatics*, 2008.