Analysis of Regular Patterns in Un-Weighted Directed Graphs

Anand Gupta, Netaji Subhas University of Technology, India

Hardeo Kumar Thakur, Netaji Subhas University of Technology, India

<https://orcid.org/0000-0002-2954-1308>

Anmoll Kumar Jain, Netaji Subhas University of Technology, India

Prakhar Rustagi, Netaji Subhas University of Technology, India

ABSTRACT

Time evolving networks tend to have an element of regularity. This regularity is characterized by the existence of repetitive patterns in the data sequences of the graph metrics. As per the research, the relevance of such regular patterns to the network has not been adequately explored. Such patterns in certain data sequences are indicative of properties like popularity, activeness, etc., which are of vital significance for any network. These properties are closely indicated by data sequences of graph metrics – degree prestige, degree centrality, and occurrence. In this paper, (a) an improved mining algorithm has been used to extract regular patterns in these sequences, and (b) a methodology has been proposed to quantitatively analyse the behavior of the obtained patterns. To analyze this behavior, a quantification measure coined as “Sumscore” has been defined to compare the relative significance of such patterns. The patterns are ranked according to their Sumscores, and insights are then drawn upon it. The efficacy of this method is demonstrated by experiments on two real-world datasets.

KEYWORDS

Degree Centrality, Degree Prestige, Dynamic Networks, Occurrence Sequence, Regular Patterns, Sumscore

I. INTRODUCTION

Real life networks are increasingly being modeled as graphs. Graphs provide an excellent representation of interconnections amongst the nodes of a network. Apart from the information depicted by these connections, some information lays unapparent amongst its properties. Thus, it is imperative to develop methods to process graphs to mine such information. A large amount of work has been done in the field of mining this data from such interconnections.

These connections have traditionally been modeled as static networks (Elseidy et al., 2014; Guimei et al., 2009; Inokuchi et al., 2000; Inokuchi et al., 2003; Kuramochi & Karypis, 2001; Kuramochi & Karypis, 2004; Yan & Han, 2002), which do not change over time. However, real life networks are dynamic in nature, where the nodes and their connections evolve with time. Thus, nowadays, networks are increasingly being modeled in a time-series representation (Borgwardt et al., 2006; Desikan & Srivastava, 2004; Duan et al., 2009; Gupta & Thakur, 2013; Gupta, Thakur, & Goel, 2014; Gupta & Thakur, 2013; Gupta & Thakur, 2015; Gupta, Thakur, & Gundherva, 2014; Gupta et al., in press;

DOI: 10.4018/IJIRR.289571

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

Gupta, Thakur, & Kishore, 2014; Halder et al., 2013; Holder & Cook, 2009; Lahiri & Berger-Wolf, 2010; Lin et al., 2008; Obulesu et al., 2014; Rasheed et al., 2011; Yang et al., 2014) where it is represented as a series of graphs. Each subgraph is a snapshot of the network at successive intervals within the duration of observation (Yang et al., 2014). For such representations, data in the graph is seen as data sequence of occurrence, indegree, outdegree etc. for each node and edge. The length of the sequence is equal to the number of snapshots, and each element in the sequence corresponds to the status of the node, or edge, for that snapshot.

Such dynamic graphs usually exhibit recurring patterns in their data-sequences. These recurring patterns find highly lucrative possibilities in domains of email, social networks, transportation networks, stock markets, and many more. They can be used to predict information such as share-price trends, road rush hours, vacation hotspots, airline traffic etc. Extensive research has been carried out for discovery of such patterns (Borgwardt et al., 2006; Gupta & Thakur, 2013; Gupta, Thakur, & Goel, 2014; Gupta & Thakur, 2013; Gupta & Thakur, 2015; Gupta, Thakur, & Gundherva, 2014; Gupta et al., in press; Gupta, Thakur, & Kishore, 2014; Halder et al., 2013; Holder & Cook, 2009; Lahiri & Berger-Wolf, 2010; Obulesu et al., 2014; Rasheed et al., 2011) within evolving network domains. But much attention has not been given to the behavioral analysis of such patterns, i.e. the application and interpretation of the patterns for the said domains.

In 2004, Desikan et al. (2004) highlighted the significance of studying the evolving nature of graphs and put forward three generic approaches that may apply for analyzing such graphs. To the best of our knowledge, no further work has been done in applying said approaches for regular patterns to analyze a dynamic graph for the extraction of behavioral information. This information can be the statistics of a mobile user's activity over the network, the popularity of a person among his peers, or the distribution of air-traffic on a route through the year, etc. This paper extends on the 'single node analysis' proposed by Desikan et al. (2004), and proposes a new method to find such statistics. In addition, an improved and efficient approach to derive the regular patterns of in-degree and out-degree sequences has been presented. The proposed method uses a new metric called "Sum-Score" for comparing the relative significance of different entities within a network. Three data sequences of –occurrence, degree centrality, and degree prestige have been chosen as the best suited for the purpose of behavioral analysis by the authors. These sequences are first mined for regular patterns for different entities such as nodes and edges. These patterns are then assigned the proposed metric called 'Sumscore' to compare the significance of information the patterns exhibit in their particular sequence. A ranking scheme is then used for enumerating the entities in their order of importance in accordance with their Sumscores.

Let us present an example to further explain the need for such analysis and the results that could be derived thereof.

In a network of airplanes, one can consider a graph-like analogy where airports represent nodes, and flights are analogous to edges. Mining the regular patterns for occurrence sequences, one can estimate the traffic on a given route. The proposed method of classification, gives an index to regularly used routes based on the percentage of the active time-slices (a time-slice is said to be active if the edge exists in that time-slice) in their occurrence sequence pattern. This can be used to compare the activeness of a route. This information can also be used to predict the busiest as well as the least used routes and subsequently regulate air traffic.

In-degree patterns of such a network give a measure of the inflow of flights at a particular airport. This information can be used to estimate and compare the popularity of frequently visited tourist destinations. Air fares can be revised using this estimate which can help generate more revenue for the company, as well as help in forming promotional schemes for less popular destinations for developmental purposes.

The degree centrality characterizes the outflow of information from a particular node. Such out-degree patterns can help in measuring the export of cargo from different regions using cargo flight

patterns. This information can be used to contrast export from various ports and hence can be used in the calculation of a comparative index between different regular exporter regions.

The remainder of the paper is organized into the following sections. Section II presents the basic notations and definitions used. Section III outlines the procedure followed. Next, results of the experiment conducted have been tabulated and analysed in Section IV. Finally, Section V draws out the conclusion of the paper.

II. NOTATIONS AND DEFINITIONS

A graph G here denotes a dynamic directed graph $G = (V, E, F)$, where V denotes the node set, $E = \{ \langle u, v \rangle \mid u, v \in V \}$ denotes the edge set where $\langle u, v \rangle$ is an ordered pair of nodes, and F is the mapping, assigning direction to edges as defined in (Gupta, Thakur, & Kishore, 2014). In this paper, occurrence for the ordered pair of nodes $\{ \langle u, v \rangle \in E \}$ representing a directed edge is represented such that '1' corresponds to the presence of the edge in the time-slice under consideration, and a '0' represents its absence.

A directed edge needs to be defined by two nodes, its originating node and terminating node. We distribute these functions as following roles.

Definition 1: Actor and subject – Let E be the edge between $\langle v_1, v_2 \rangle$ with the direction from v_1 to v_2 , then node v_1 is denoted as the actor of E and v_2 as the subject.

To determine the existence of an edge within the time-slices of a graph, an occurrence sequence of that edge is sought.

Definition 2: Occurrence sequence- As defined in (Gupta, Thakur, & Kishore, 2014), the occurrence sequence of an edge 'e' is a sequence of 1s and 0s with a length T such that if 'e' appears at time-slice 't', the 'tth' position in its occurrence sequence is 1, otherwise it is 0.

To characterize the connectivity in vertices along with edges, we look for patterns in their degree centrality and degree prestige sequences.

Definition 3: Degree Centrality- Degree centrality of a node is a sequence of its outdegree at all time-slices where **outdegree** is the number of edges directed away from the node in a directed graph.

Definition 4: Degree Prestige- Degree prestige of a node is a sequence of its indegree at all time-slices where **indegree** is the number of edges directed into the node in a directed graph.

To mine the requisite sequences, we need to create a cumulative summary graph consisting of all time-slices in the data set for all the active nodes and edges.

Definition 5: Summary graph - Summary graph of a dynamic directed graph $G = \langle G_1, G_2, \dots, G_T \rangle$, as put forward in (Gupta, Thakur, & Kishore, 2014), is defined as sets of $G_s = \langle V_s, E_s, L_s \rangle$, where $V_s = \langle V_1, V_2, \dots, V_T \rangle$, $E_s = \langle E_1, E_2, \dots, E_T \rangle$, and L_s is the set of labels of E_s , which maps in each set of G_s , each edge e to an occurrence sequence of length T . This definition can be extended to nodes, where in each set of G_s , each node v is mapped to an indegree and outdegree sequence of length T .

From the summary graph, we look for regular patterns in the degree prestige and centrality sequences for nodes and in occurrence sequences for edges.

Definition 6: Regular Pattern: A pattern is said to be regular if it is seen to be repeated at least a threshold number of times in the sequence uninterruptedly. Multiple patterns may exist in a given sequence which may be repeating. The longest pattern is considered to be the regular pattern as it gives the most information about the trend of the sequence.

Definition 7: Regular Edge: As defined in (Gupta, Thakur, & Goel, 2014), an Edge $e = \langle v_1, v_2 \rangle$ is said to be regular if it is seen to occur in at least threshold number of consecutive time-slices.

The computed sequences are subjected to sum score calculations to inculcate a factor for length of regular patterns along with its contribution.

Definition 8: Sum-score: It is a measure obtained by multiplying the sum of the elements (values) of a regular pattern by the length of the pattern. For example: for a random regular pattern '50050':
sum-score = $(5+0+0+5+0) \times (5) = (10) \times (5) = 50$.

Definition 9: Z - Score: It is the signed number of standard deviations an observation or datum is above or below the mean. It is calculated as the Sumscore, minus the mean of the Sumscores, divided by its standard deviation i.e. $z = (X - \mu) / \sigma$.

The next section continues onto the procedure of scoring the nodes and edges of a graph-based network to rank them in their order of importance.

III. PROCEDURE FOR EXTRACTION AND ANALYSIS OF PATTERNS

Given is a set of edges with respective time stamps in the form of edge lists constituting a dynamic graph. This set is first converted into a time-series of graphs. A time period is chosen to mark one time-slice and the edges in the list are separated into different edge lists according to their respective time-stamps. The edge lists for each time-slice represent a multi-edged directed sub-graph. The procedure for ranking the edges in the graph over all time-slices is followed as below.

Step 1: Extraction of the Occurrence Sequence:

To ascertain the occurrence of an edge within a graph, a forward and backward directed edge between the same nodes pair is treated the same. Therefore, all node pairs for each time-slice are arranged in ascending order. This means that node pairs such as '7-2' are arranged as '2-7'. Self-loops such as '3-3' are removed to avoid redundancies. Multiple instances of the same node pair are deleted. A matrix Z containing all the unique edges over all time-slices is then constructed. The edges of matrix Z are then compared with the edge-lists from all time-slices to form the summary graph. This graph contains the connections between all unique node pairs over the complete time period of observation. The occurrence sequence can then be extracted from this graph. The algorithm for occurrence extraction is as follows:

Algorithm 1: Extraction of occurrence sequences

Input: A ListL of time-slices of edge lists of network N having columns of actor and subject
Step 1: For (each time-slice T in L)
Arrange all node pairs in ascending order
Eliminate multiple instances of the node pair
Step 2: Create a cumulative matrix Z which contains a list of all unique edges present over all the time-slices.
Step 3: For (each edge E in Z)
If (E is found in time-slice t)
Set Z [E] [T+2] = 1
Else
Set Z[E][T+2]= 0
Output: The matrix Z containing occurrence sequences for all unique edges.

Step 2: Extraction of the Sequences of Degree Prestige And Degree Centrality-

The edge lists consist of the unique edges signifying the connection from actors in the first column to the subjects in the second. The number of occurrences of a node in the actor's column in one time-

slice depicts the frequency of outgoing connections for that node, i.e. the degree centrality of the particular node. Similarly the same in the subject column signify its incoming, i.e. degree prestige. Summary graphs are now constructed in form of a matrix, containing degree sequences of actors and subjects respectively, from all the time-slices. The algorithm for this is as follows:

Algorithm 2: Extraction of sequences of degree centrality and prestige.

Input: A List L of time-slices of unique edge lists of network N having columns of actor and subject.
Step1.Creating an empty matrix $M[\text{total nodes}][\text{Number of time slices}] = \{0\}$
Step 2: For (each item I in list L) do
Bind a column of 1's with length(I) to Item I.
/*use aggregate function to sum each occurrence of desired column*/
Temp = Aggregate (I, col(subject), sum)
/*subject column will give indegree of nodes, while actor column will give its outdegree*/
Step 3: /*Extract the subject column as "NODE" and Degree column as "Degree" in a new data frame*/
DegreeMatrix = data.frame(Node=Temp[,subject column],Degree=Temp[,degree column])
Step 4: For(Element E in column of subject)
Update the value $M[E][I] = \text{DegreeMatrix}[E][\text{Degree}]$
Output – Matrix M with the required degree sequence over entire period of observation

Step 3: Extracting Relevant Patterns

The data sequences obtained are now processed for finding regular patterns. A sequence exhibiting recurrence at least a threshold number of times consecutively is considered a regular sequence. Also, among the many possible patterns in that sequence, the pattern with the greatest length is selected as the regular pattern, because it provides the most information of the status of connections over time. For e.g. consider the sequence- "1,2,1,2,1,1,1" with threshold set to two, the pattern "1,2" repeating twice is the major pattern of the sequence instead of "1" repeating thrice. Also, if the pattern is a sequence of zeroes, the next pattern in order of length is considered, as the sequence of zeroes does not give any useful measure. For e.g. in the sequence- "1,1,1,0,0,0,0,0", although the longest pattern consists of all zero elements- "0,0,0" repeating twice, the pattern '1' repeating thrice is chosen.

To extract regular patterns, the sequences are to be converted into strings. Until now, methods to mine patterns (Gupta, Thakur, & Gundherva, 2014; Gupta et al., in press; Gupta, Thakur, & Kishore, 2014) in time dynamic graphs rely on the use of symbols for this conversion. Such a method results in loss of precision, in case the size of the set of elements in the sequence is greater than the size of the symbol set. We modify the method to work with numerical values instead of symbols, resulting in the elimination of precision loss. The strings are then passed through an extraction algorithm using the "regexpr" search function in R, built upon the algorithm presented in (Gupta, Thakur, & Kishore, 2014). The algorithm has been modified for our needs to ignore those patterns containing only zeros. The modified algorithm has been presented below.

Once the regular patterns have been mined, we now proceed onto the proposed method to carry out the behavioral analysis of the mined patterns.

Algorithm 3: Extracting regular patterns

Input: Matrix M containing numerical data sequences for all thenodes (edges in case of occurrence sequence)
Step1. Create an empty data frame D with initial columns of all nodes (or edges) present M
Step2. Bind a column of empty String to all rows in D
Step3. For (each row I in Matrix S)
For (for each column j (excluding node pairs) in Matrix S)
{
concatenate M[i][j] to D[i](Kuramochi & Karypis, 2001) along with a separator '/'.
/*to convert numerals into strings and differ between consecutive values like "2" and "22"*/
}
Step3: D is matrix with last column as string sequences having separator '/'.
Step4: Define PatternExtractor(String S)
{
Numchar = nchar(S)
Len = Numchar/threshold
For(Integer k in Len to 1)
{
Form a regular expression RE to search for repeating sequences in S.
R<- regexpr(RE,S)
/*regexpr finds the repeating sequence in the string S matching the RE along with various attributes*/
L<- attr(R,"capture.length")
If (L>0) /*A pattern of length >0 has been found*/
{
Pattern <- Substring(S, R, R+L-1) /*Extract the pattern from string*/
/*Now To eliminate patterns consisting of zeroes*/
Vec = Convert(Pattern)
If (Mean(Vec)>0)
Break /*A definite non zero pattern of largest length has been found*/
}
Set R=0
}
If (R >0)
Extract Pattern P
Else
Set P = NA
Return P
}
Step 5: Create Matrix E with all columns except last from D.
/*Find and extract patterns amongst last column and returns result as list Pat of patterns*/
Step 6: Pat = Lapply (D[Last Column],PatternExtractor)
Bind Pat to Matrix E.
Output: A Matrix E consisting of nodes (edges in case of occurrence sequence) and their requisite regular patterns.

Step 4: Ranking the Nodes According to The Regular Pattern Length and Strength of Nodal Connections.

We rank nodes and edges on the basis of their **strength of connections** and **the factor of regularity**. The strength of a connection is directly related to the number of times the connection exists over all time-slices; and regularity is determined by the existence of a recurring pattern (with repetition over a minimum threshold) of interactions between the nodes. To quantify strength, we take the sum of all elements of the pattern. For e.g. a pattern with “100, 0” having strength of (100+0) = (100) is stronger than the pattern “25, 25” with a strength of (25+25) = (50). On the other hand, to quantify the factor of regularity, we consider the length of the repetitive pattern into our calculations (e.g. 1, 1, 0 is more regular than 1, 1).

To consolidate both the factors together, we propose the metric of ‘**Sumscore**’, used for each pattern, which is obtained by taking the sum of the elements of the pattern for each edge/node and then multiply it with the length of the pattern. The length of the patterns provides weight to the impact of the strength of the pattern.

Then we apply a scoring system, called z-scores (Tigger, n.d.), on the calculated scores to **normalize** the distribution of Sumscores. Further, we use the **variance** of the patterns to break the ties in z scores of clashing edges/nodes to assign unique rankings to them.

$$z_i = \frac{X_i - \bar{X}}{\sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n}}} \quad (1)$$

Algorithm 4: Calculating z-scores

Input: Extracted set of patterns in a matrix P.
Step 1 Extract the column vector C of patterns from Matrix P.
Step 2 Define Convert(p)/pattern string p*/
{
2.1 List L = Store indexes of all sentinels ‘/’ in the pattern P.
2.2 For (each i in 1 to length(L)-1)
2.2.1 S = Find and extract substring between L[i+1] and L[i]
2.2.2 Convert S to integer and Add S to vector V
2.4 Return vector V;
}
Step 3.For (i in 1 to length(C)) do
Vec= Convert(p[i]th pattern) into a vector.
Var[i] = variance of the converted vectorVec.
Sumscore[i] =length(p[i]) * sum(Vec)
zscore[i] = calculating zcore by passing the Sumscore vector and applying method in Equation (1)
Step 4. Bind Var, Sumscoreandzscore columns to P.
Step 5. Output [Orderby (zscore[i] followed by variance of P[i])]
Step 6. Bind a column of “Rank” of Length(C) to P
Output: Set of ranked patterns in matrix P ordered by their zscores followed by variance

Z-score is a useful measure to determine how far away a particular score is from the mean of the data, in terms of multiples of standard deviations. This provides a method to analyze relative performance of patterns within the dataset. This methodology of analysis is tested on two real world networks. The following section shows the experimental results of the methodology proposed. The results are analysed with context to the domain of the network and possible inferences are drawn.

IV. EXPERIMENTAL ANALYSIS

Datasets

We exemplify the proposed approach by putting two real world network datasets through analysis. In coherence with the aforesaid example, we choose flight statistics that will exhibit all the relevant parameters that fit the example. Second set is an Email dataset from ENRON corp.

1. **T-100 Domestic Market (US Carriers Only)** -: This data set was taken from Bureau of Transportation statistics (BTS) - US Department of transportation. The dataset can be downloaded from https://www.transtats.bts.gov/Tables.asp?DB_ID=110. This table contains domestic market data reported by U.S. air carriers, including carrier, origin, destination, and service class for enplaned passengers, freight and mail when both origin and destination airports are located within the boundaries of the United States and its territories.

This data set has been taken for year 1990, having 151,832 flights. It is converted into series of 12 datasets, one for each month. The frequency of each distinct flight is cumulated for every month.

2. **Enron Email Network** - This dataset was collected and prepared by the CALO Project (A Cognitive Assistant that learns and Organizes). The dataset can be downloaded from

<http://www.cs.cmu.edu/~enron/>. It contains data from about 150 users, mostly senior management of Enron, organized into folders. Weighted directed graphs are constructed from communications among these users from December 1999 to March 2002. The corpus contains a total of about 0.5M messages. We have chosen one month as a time-slice and there are total 28 time-slices. Thus there are 2,359 vertices and 73,592 edges on average. For the edges which have multiple existences within the time-slice of 1 month, net weight and direction in the time-slice are calculated.

The two data sets have been analyzed with respect to three properties of graphs – degree prestige, degree centrality and occurrence sequences. We use the histogram plots (Deng & Wickham, 2011) to analyze the distribution of data with around 200 bins to bring the bin width nearly equal to 1. We also employ quartile distribution to draw inferences in a more effective way. The plots have been built with respect to the Sumscores of patterns of all three properties for both the data sets. Results of the analysis are tabularized and described below.

1. Occurrence Sequences

Table 1 shows the characteristics of regular patterns found for occurrence sequences in both the datasets. The T-100 dataset has total of 61144 unique flight paths, out of which 6043 flight paths have flights in a regular pattern. Here, Sumscore measures relative significance of a flight route based on the frequency and number of the flights on it, with respect to its peers. Amongst such repeatedly taken flights, the analysis puts up flight path '10140'-'10279' as the most regularly used in that year. The path '14893-16440' (Sacramento - Austin) is borne out to be the least significant amongst regular flight paths. As can be seen from Table-1 and Fig 1(i), more than 25% of the flight

Table 1. Occurrence Sequence found after applying the algorithm to the datasets with Threshold =2

	T-100	Enron
Direction Patterns found-	6043	8434
Range of the patterns length- (max and min)	1-6	1 – 14
Most regularly significant edge	10140-10279	3-116
Least regular edge	14893-16440	10-1303
Most significant pattern-	'1,1,1,1,1,1'	'1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1'
Least significant non zero pattern	1	'1'
Z-score(Sumscore) of most significant edge	1.024(36)	12.099(196)
Z-score(Sumscore) of least significant edge	-1.32(1)	-0.77(1)
Expected value of Sumscore	20.75	12.76
Quartile distribution of Sumscore values (25%, 50%, 75%)	4,18,36	5,8,12

Figure 1a. The distributions for occurrence sequence Sumscores of (i) T-100D flights and (ii) Enron emails

paths have the same Sumscore of 36. This is the maximum possible Sumscore because the dataset comprises of 12 time-slices with the threshold of repetition set to 2. This highlights the flight paths which have regular flight deployment and form an essential cog in the system of smooth flight travel across various destinations. Airport authorities may find this analysis useful for prioritizing their resources for important flights.

Considering the Enron dataset, it has a total of 73592 unique edges representing direct communication between employees. Out of those, 8434 edges are found to have regular patterns in their sequence. The length of the patterns in the sequence varies from 1 to 14. The z score for the busiest connection, given by the edge '3-116', of 12.099 signifies its variation from the mean score of the dataset. The busiest connection thus encounters the traffic and information which is nearly **12 times** the standard deviation from the mean of scores. Higher the Z-score of a connection, greater is its significance in the network. On the same lines, the edge '10-1303' has seen to have the least amount of

Figure 1b. The distributions for occurrence sequence Sumscores of (i) T-100D flights and (ii) Enron emails

communication. This analysis can be put to use by the management of a company to analyse the level of communication between different employees. Thus teams can be better formed for coordinating work which require high levels of cooperation, leading to soaring of productivity and profit.

2. Degree Centrality sequences

Table 2. Out-Degree (Degree Centrality) Metrics found after applying the algorithm to the datasets with Threshold =2

	T-100	Enron
Out-degree Patterns found-	237	210
Range of the patterns length- (max and min)	1-5	1-11
Most regularly significant node	11056	189
Least regular node	14811	143
Most significant pattern-	'95,96'	'13,13,13,0,0'
Least significant non-zero pattern	1	1
Z-score(Sumscore) of most significant node	6.70(382)	6.277(195)
Z-score(Sumscore) of Least significance	-0.76(1)	-0.73(1)
Expected value of Sumscore	40.07	21.4
Quartile distribution values of Sumscores (25%,50%,75%)	10,24,50	4.25,10.5,25

Table 2 shows the properties of evolutionary patterns in Degree Centrality sequences of both the data sets. The z-score here measures the relative regular **outflow** of information between the nodes.

For the T-100 dataset, amongst the patterns found for 237 airport terminals, the length of patterns varies between 1 and 5. This signifies that in a period of 12 months, the usual continuous departure flights range from a span of **2** to even **10** months. Airport 11056 is found to be the **busiest** airport in terms of regularity of flight departures as well as number of departures, with a high Sumscore of

Figure 2a. The distributions for outdegree Sumscores of (i) T-100D flights and (ii) Enron emails

Figure 2b. The distributions for outdegree Sumscores of (i) T-100D flights and (ii) Enron emails

382 which lies 6 standard deviations away from the mean expected score. In T-100 dataset of market carriers, this port can be said to be the most significant exporter of regularly used freight goods. Whereas, the airport represented by 14811 is the **least busy** one, in terms of regularly used flights with a negative score of -0.73. The histogram plot in Fig 2(i) shows distribution of Sumscores in this dataset. This estimation can be used to prepare relative footfall estimates, preparing sizes of departure gates at airports to accommodate planes docking and preparing basic amenities and help kiosks. The quartile distribution shows that top 25 percentile have cumulative score of 5871 while rest 75

percentile sum up to 3626. This estimate shows that the top 25 percent airports are more important in providing regular business than the rest combined.

For the dataset of Enron, we can say that node 189 is most significant in terms of sending viable amount of information as well as the regularity with which the information is being sent by it across the network. It has a score of **195**, which lies 6 standard deviations away from the mean score. This score can be considered as a measure of the relative reach of a node, taking into account the regularity of its outgoing connections and the amount of such connections. The node 143 is least active in these terms, with a z-score of -0.73. Such information is useful in comparing the performance of interaction capabilities of an employee with others. It can also signify how well they communicate with their subordinates and peers. The pattern length ranges from 1 to 10 in 28 time slices, giving a repetition ranging from 2 to 20 time slices. The quartile distribution shows that 25 percentile gives a cumulative score of 3071 while the rest sum upto 1423. The Fig 2(ii) illustrates the distribution of Sumscores of outdegree sequences of the Enron dataset.

3. Degree Prestige Sequences

Table 3 shows results of the analysis on evolutionary patterns in Degree prestige sequence. Here, the Sumscores represent the relative significance amongst the nodes with respect to regular **inflow** of information.

Table 3. In-Degree metrics found after applying the algorithm to the datasets with Threshold =2

	T-100	Enron
Indegree Patterns found-	237	779
Range of the patterns length- (min and max)	1-5	1 – 10
Most regularly significant node	11258	132
Least regular node	14811	2349
Most significant pattern-	'31,31,31'	'65,54'
Least significant non zero pattern	'1'	'1'
Z-score/Sumscore of most significant node	6.94(279)	11.21(236)
Z-score/Sumscore of Least significance	-0.96(1)	-0.63(1)
Expected value of Sumscore	34.84	13.65
Quartile distribution values (25%, 50%, 75%)	11,24,46	3,8,16

For the T-100 dataset, a high Sumscore signifies most regularly visited destination. The pattern length ranges from 1 to 5, giving a repetitions ranging on a 2 to 10 month scale. Airport represented by node 1259 receives the maximum inflow of air-traffic on regular basis, whereas airport 4812 encounters the least regularly used flights. Figure 3(i) shows the density distribution for the Sumscores of T-100D dataset. It shows that the top 25 percentile have a cumulative score of 4885, while the rest 75 percentile add only up to 3374. This reinforces that the top locations indicated by this analysis are indeed the power houses for air transportation, and the authorities would do good for prioritizing their resources for these locations. It can also be used to estimate the baggage capacity required, immigration desks, basic amenities, car parking facilities etc.

Figure 3. The distributions for indegree Sumscores of (i) T-100D flights and (ii) Enron emails

Figure 3b. The distributions for indegree Sumscores of (i) T-100D flights and (ii) Enron emails

In Enron dataset, Node 132 of the Enron dataset has the highest Sumscore rating, which signifies it to be most important node, receiving the highest amount of information regularly in the network. Similarly node 2349 falls to the bottom of the scale with a rating of -0.63. This is a direct measure of regular incoming information to a user. This information is useful in gauging the popularity of a person amongst his peers. The range of patterns here lies between 1 and 10. The quartile distribution shows that top 25 percentile accumulates a Sumscore of 7072 while rest have sum of 3563. Fig 3(ii) shows this distribution. Corporations may find this useful for quantizing ‘in how much esteem, a person is held by his colleagues’. It would help show the extent to which managers are reached out to for guidance and help, and can aid in evaluating corporate quotient.

V. CONCLUSION

In this paper, we have analysed the relation between regular patterns and behavioral properties like popularity, activeness etc. of a node. These patterns have been mined in data sequences of three specific graph metrics. To quantitatively analyze this relation, we have defined a new metric - 'Sumscore' and then ranked the regular patterns on the basis of this score. Experiments conducted on two real world networks reveal the popularity and activeness of the nodes of the network. Insights have then been drawn upon these results with context to the network. The results and plots demonstrate that the proposed metric of Sumscore is successfully able to segregate and rank nodes and edges in accordance to their significance to the network.

These obtained regular patterns consists of numerical sequences which repeat themselves at least a threshold number of times. The numerical patterns are useful in modeling the behavior with high precision. These sequences must repeat in exactly the same fashion(numbers) to be considered a pattern. However, need may arise for sequences having close but different values to be considered similar. Such a tradeoff is possible between precision and degree of similarity between repeating sequences, depending upon the requirement.

REFERENCES

- Borgwardt, K. M., Kriegel, H. P., & Wackersreuther, P. (2006). Pattern mining in frequent dynamic subgraphs. *Proceedings of IEEE International Conference on Data Mining*, 818-822.
- Deng, H., & Wickham, H. (2011). Density estimation in R. Electronic Publication.
- Desikan, P., & Srivastava, J. (2004). Mining temporally evolving graphs. *Proceedings of 6th International Workshop on Knowledge Discovery on the Web, WebKDD*, 13-22.
- Duan, D., Li, Y., Jin, Y., & Lu, Z. (2009). Community mining on dynamic weighted directed graphs. *Proceeding of the 1st ACM international workshop on Complex networks meet information & knowledge management (CNKM)*, 11-18. doi:10.1145/1651274.1651278
- Elseidy, M., Abdelhamid, E., Skiadopoulos, S., & Kalnis, P. (2014). Grami: Frequent subgraph and pattern mining in a single large graph. *PVLDB*, 7(7), 517-528. doi:10.14778/2732286.2732289
- Guimei, L., Wong, L., & Chua, H. N. (2009). Complex discovery from weighted PPI networks. *Journal of Bioinformatics*, 25(15), 1891-1897. doi:10.1093/bioinformatics/btp311 PMID:19435747
- Gupta & Thakur. (2013). A novel method to determine regular pattern in edge labelled dynamic graphs. *Proceeding of the 7th International Conference on Data Mining and Warehousing (ICDMW '13)*, 39-47.
- Gupta, Thakur, & Goel. (2014). Evolution of regular directed patterns in dynamic social networks. *Proceedings of 3rd IEEE International Conference on Advances in Computing Communications and Informatics (ICACCI'14)*, 1466-1471.
- Gupta, A., & Thakur, H. K. (2013). Method in Determining Regular Pattern in Edge Labelled Dynamic Graph and the concept of Partial Regular. *International Journal of Information Processing*, 7, 55-65.
- Gupta, A., & Thakur, H. K. (2015). Framework for Mining Regular Pattern in Dynamic Networks. *International Journal of Knowledge Engineering and Data Mining*, 3(3-4), 299-314.
- Gupta, A., Thakur, H. K., & Gundherva, N. (2014). Mining regular pattern in Edge Labeled Dynamic Graph. *Proceedings of 9th IEEE International Conference on Industrial and Information Systems (ICIIS'14)*, 1-6. doi:10.1109/ICIINFSS.2014.7036608
- Gupta, A., Thakur, H. K., Gupta, T., & Yadav, S. (in press). Regular Pattern Mining (with Jitter) on Weighted Directed Dynamic Graphs. *Journal of Engineering Science and Technology*, 12(4).
- Gupta, A., Thakur, H. K., & Kishore, P. (2014). Mining Regular Patterns in Weighted Directed Networks. *Proceedings of 13th IEEE International Conference of Information Technology (ICIT'14)*, 215-220. doi:10.1109/ICIT.2014.44
- Halder, S., Han, Y., & Lee, Y. (2013). Discovering Periodic Patterns using Supergraph in Dynamic Networks. *Research Notes in Information Science*, 14, 148-151.
- Holder, L. B., & Cook, D. J. (2009, June). Learning patterns in the dynamics of biological networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 977-986). ACM.
- Inokuchi, A., Washio, T., & Motoda, H. (2000). An apriori-based algorithm for mining frequent substructures from graph data. *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery, PKDD '00*, 13-23. doi:10.1007/3-540-45372-5_2
- Inokuchi, T., Washio, T., & Motoda, H. (2003). Complete mining of frequent patterns from graphs: Mining graph data. *Journal of Machine Learning*, 50(3), 321-354. doi:10.1023/A:1021726221443
- Kuramochi, M., & Karypis, G. (2001). Frequent subgraph discovery. *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 313-320.
- Kuramochi, M., & Karypis, G. (2004). An efficient algorithm for discovering frequent subgraphs. *Journal of IEEE TKDE*, 16(9), 1038-1051. doi:10.1109/TKDE.2004.33
- Lahiri, M., & Berger-Wolf, T. Y. (2010). Periodic subgraph mining in dynamic networks. *Journal of Knowledge and information systems*, 24(3), 467-497. doi:10.1007/s10115-009-0253-8

Lin, Y. R., Chi, Y., Zhu, S., Sundaram, H., & Tseng, B. L. (2008). Facetnet: a framework for analyzing communities and their evolutions in dynamic networks. *Proceedings of the 17th International Conference on the World Wide Web (WWW08)*, 685-694. doi:10.1145/1367497.1367590

Obulesu, O., Reddy, A. R. M., & Reddy, M. T. (2014). Finding Frequent and Maximal Periodic Patterns in Spatiotemporal Databases towards Big Data. *International Journal of Emerging Technology and Advanced Engineering*, 4(1), 266–265.

Rasheed, F., Alshalalfa, M., & Alhajj, R. (2011). Efficient periodic pattern mining in time series databases Using Suffix Trees. *IEEE Transactions on Knowledge and Data Engineering*, 23(1), 79-94.

Tigger. (n.d.). http://tigger.uic.edu/classes/chsc/chsc432/restricted/exercises/ranking_scoring/index_percentile_rescaling.htm

Yan, X., & Han, J. (2002). gspan: Graph-based substructure pattern mining. *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 721–724.

Yang, Y., Yu, J. X., Gao, H., Pei, J., & Li, J. (2014). Mining most frequently changing component in evolving graphs. *Journal of World Wide Web*, 17(3), 351-376. doi:10.1007/s11280-013-0204-x

Anand Gupta is an Assistant Professor in Netaji Subhas Institute of Technology (NSIT). NSIT is affiliated to Delhi University, India. His research interests include Data mining, database system, image processing and information retrieval.

Hardeo Kumar Thakur is a Teaching cum research Fellow in Netaji Subhas Institute of Technology (NSIT). NSIT is affiliated to Delhi University, India. He is pursuing his Ph.D. in the area of dynamic graph mining. Final year student at Netaji Subhas Institute of Technology, University of Delhi, Dept. of ECE.