ISLAMIC UNIVERSITY OF TECHNOLOGY

Predicting signs and directions of links in online Social Networks

By:

Shadman Sakib Chowdhury (094433)

Md. Safayet Khan (094414)

Supervised by:

Md. Mohiuddin Khan

Assistant Professor

Department of Computer Science and Engineering

Co-supervised by:

Mahmud Hasan

Assistant Professor

Department of Computer Science and Engineering

A thesis submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering

Academic Year: 2012-2013

Department of Computer Science and Engineering

Islamic University of Technology.

A Subsidiary Organ of the Organization of Islamic Cooperation.

Dhaka, Bangladesh.

October 2013

Declaration of Authorship

We,

Shadman Sakib Chowdhury (094433)
Md. Safayet Khan (094414), declare that this thesis titled,'Predicting signs and directions of links in online Social Networks' and the work presented in it are our own. We confirm that:

- This work was done wholly while in candidature for a Bachelor degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.

Submitted By:

Shadman Sakib Chowdhury (094433)

Md. Safayet Khan (094414)

Predicting signs and directions of links in online Social Networks

Approved By:

Prof. Dr. M.A. Mottalib
Head of the Department,
Department of Computer Science and Engineering,
Islamic University of Technology.

Md. Mohiuddin Khan
Thesis Supervisor,
Assistant Professor,
Department of Computer Science Of Technology,
Islamic University of Technology.

Abstract

Studying online social networks gives us a new way to visualize the network data. The relations between the users in the social networks can be defined as positive and negative where positive means friendship and negative relations means antagonism. Signs in the social networks are important because the attitude of one user toward another user can be estimated from evidence provided by their common relationships and also other members surrounded in the specific social network. We studied how the relationships can be denoted by the signs. For that purpose we need to predict the sign (relationship type) between two users where the users can be represented by the nodes and their relationship can be represented by signed edges. Again implementing triad and a new theme quad has been implemented for better demonstration to predict the signs and directions between actors of social networks.

Acknowledgements

Special thanks to:

Md. Kamrul Hasan, PhD

Assistant Professor

Department of Computer Science and Engineering

Contents

List of Figures

Dedicated to our parents. . . .

Chapter 1

Introduction

1.1 introduction

A social network is a set of actors (or points, or nodes, or agents) that may have relationships (or edges, or ties) with one another. Networks can have few or many actors, and one or more kinds of relations between pairs of actors. To build a useful understanding of a social network, a complete and rigorous description of a pattern of social relationships is a necessary starting point for analysis. That is, ideally we will know about all of the relationships between each pair of actors in the population. The amount of information that we need to describe even small social networks can be quite great. Managing these data, and manipulating them so that we can see patterns of social structure can be tedious and complicated. All of the tasks of social network methods are made easier by using tools from mathematics. For the manipulation of network data, and the calculation of indexes describing networks, it is most useful to record information as matrices. For visualizing patterns, graphs are often useful.

1.2 Motivation

Social relationships and networking are key components of human life, and they have been historically bound according to time and space limitations; these restrictions have been partially removed because of the Internet evolution and its use diffusion. In particular, the emergence of Web technologies and their evolution towards the Web 2.0 enables people to organize themselves into Virtual SocialNetworks in the same manner they organize themselves in social networks in thereal world.The difference between both earlier social networks and Virtual Social Networksmainly consists of the mechanism used by the members to communicate each other. If the link prediction can be done accurately then the users of virtual social network can be notified about their future. In one sense the analyst groups of social networking sites are doing the job of analysis of the general users to connect with other people. It would motivate the people more to join the virtual social networks and make the networking more effective.

1.3 State of the art

Link prediction problem has been in the scene after the sudden overwhelming popularity of the virtual social networks earlier this millennium. The more the number of users got bigger the more the relationships increased and it became necessary to find a feature helping the users to find their possible connections automatically. Since then several algorithms has been introduced so far. But the problem remains regarding the huge set of the user data which are required to be analyzed in order to do the prediction. So the prediction is both time and resource consuming. And no prediction has been proved to be meeting the threshold of the actual linking.

1.4 Goal of the thesis

The first goal of our thesis is to define a collection of features based on different types of approaches. For that purpose we have used three types of approach. They are seven vector, sixteen vector and twenty three vector approach. To create Quad feature that is taking consideration four nodes and predicting signs and directions of links in an effective way. Also optimizing time complexity is also a major concern.

Chapter 2

Basics of Social Network Analysis and Link Prediction

2.1 Social network

A social network is a social structure made up of a set of social actors (such as individuals or organizations) and a complex set of the dynamic ties between these actors. The social network perspective provides a clear way of analyzing the structure of whole social entities.These networks are groups of individuals who share a commonality. Their common bond of social networks may be the community in which members live, their religion, subdivision, career interest, social interests and common friends or shared beliefs. While traditionally, social networks were made up of people who might gather face-to-face, today's social networks are predominately online. Examples of today's social networks include social networking sites, such as LinkedIn, Facebook, Wikipedia, MySpace, Slashdot and others. With increasing mobile phone popularity and enhanced cell phone technology, even Twitter might be considered a social network of sorts where you can let your network of friends know what you are doing or thinking at any given moment. Today's social networks are predominately free to join, which makes them extremely popular. The motivations behind joining a social network are: exchange of information, friendship, social aspect, recreation.

2.2 Social network analysis

Social network analysis is the study of the network to identify local and global patterns, locate influential entities and examine network dynamics. The analysis of these social networks emerged from social psychology, statistics, psychology and graph theory. George Simmel authored early structural theories in sociology emphasizing the dynamics of triads and web of group affiliations. And the credit of developing the first "sociograms" goes to Jacob Moreno. In general, social networks are self-organizing, emergent, and complex, such that a globally coherent pattern appears from the local interaction of the elements that make up the system. These patterns become more apparent as network size increases. However, a global network analysis of, for example, all interpersonal relationships in the world is not feasible and is likely to contain so much information as to be uninformative.

Social network
Blog
Charity
School
Friends
Family
Work
Club
Business
Events
Business
Forum

Figure 2.1: Different aspects of a social network

Practical limitations of computing power, ethics and participant recruitment and payment also limit the scope of a social network analysis. The nuances of a local system may be lost in a large network analysis. Hence the quality of information may be more important than its scale for understanding network properties. Thus, social networks are analyzed at the scale relevant to the researcher's theoretical question.

2.3 Representations of social networking

Social network analysts use two kinds of tools from mathematics to represent information about patterns of ties among social actors: graphs and matrices. We have studied the graph part. There are lots of different kinds of "graphs."

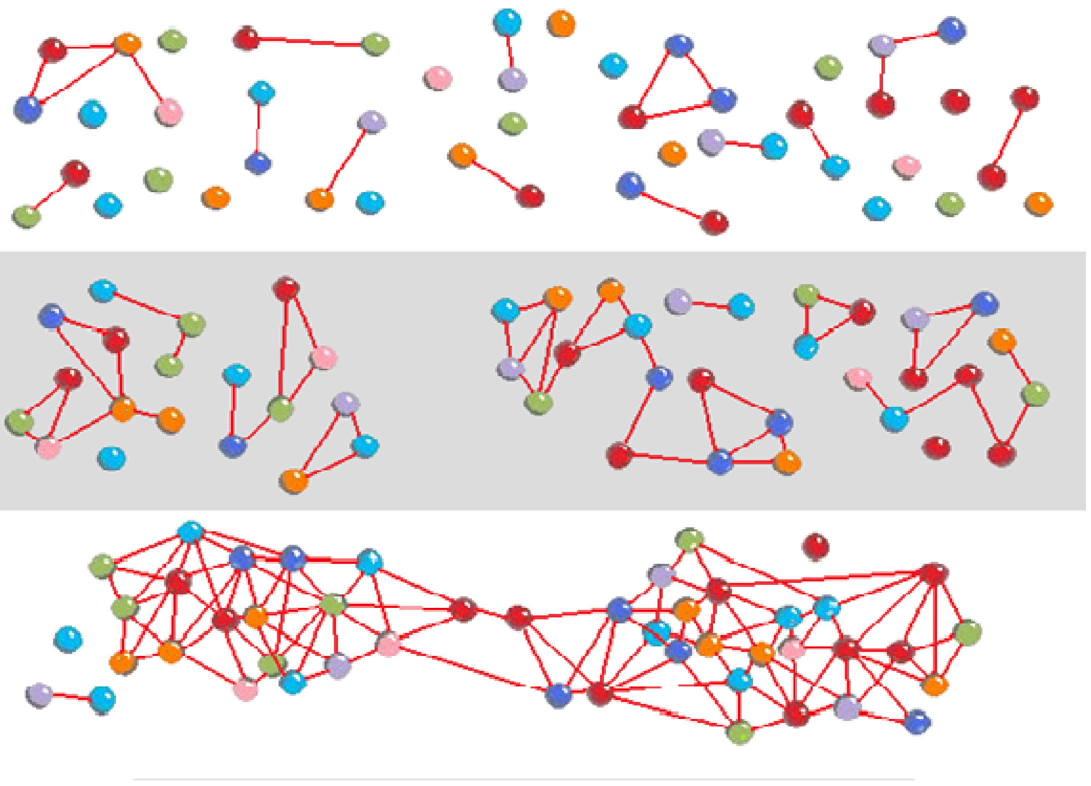


Figure 2.2: Self organization of a network based on Nagier, Levina and Timme(2011)

Bar-charts, pie-charts, line and trend charts, and many other things are called graphs and/or graphics. Network analysis uses (primarily) one kind of graphic display that consists of points (or nodes) to represent actors and lines (or edges) to represent ties or relations. When sociologists borrowed this way of graphing things from the mathematicians, they re-named their graphics "socio-grams." The kind of graphic displays consist of "directed graphs" "signed graphs" or simply "graphs." Figure 3 below shows a graph with four labeled nodes, but no connections.

Figure 4 implies a directed graph of friendship ties.

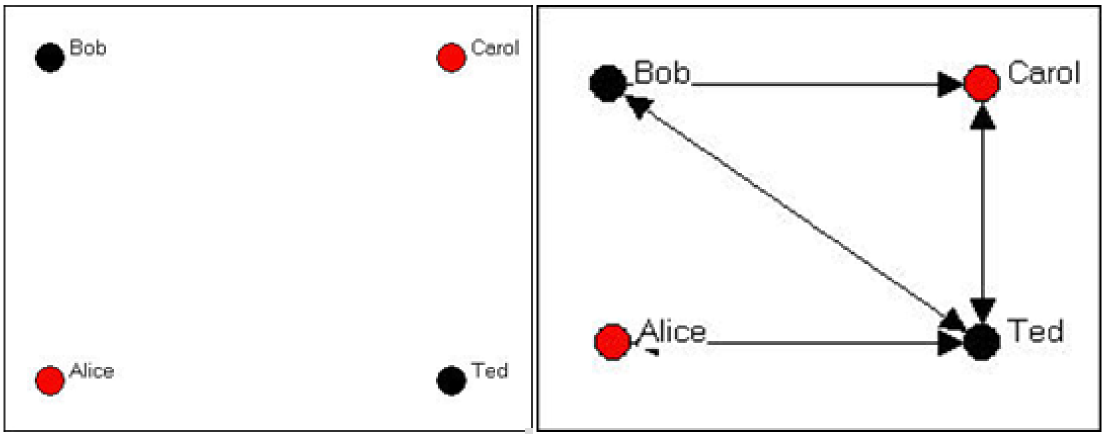


Figure 2.3: Labeled nodes with no edges and Labeled nodes with directed edges

2.4 Types of graphs

There can be two kinds of graphs:
1) Directed:
The edges connecting the nodes don't have any directions.

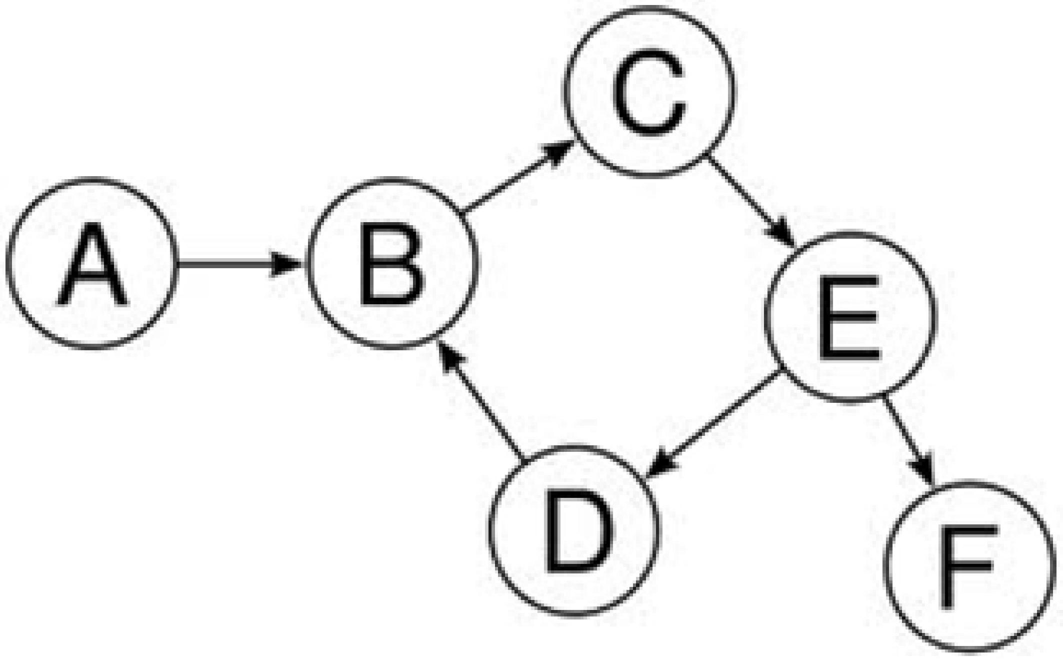


Figure 2.4: Directed graph

2) Undirected:
The edges connecting the nodes have directions indicating the relationship type of the connected nodes.

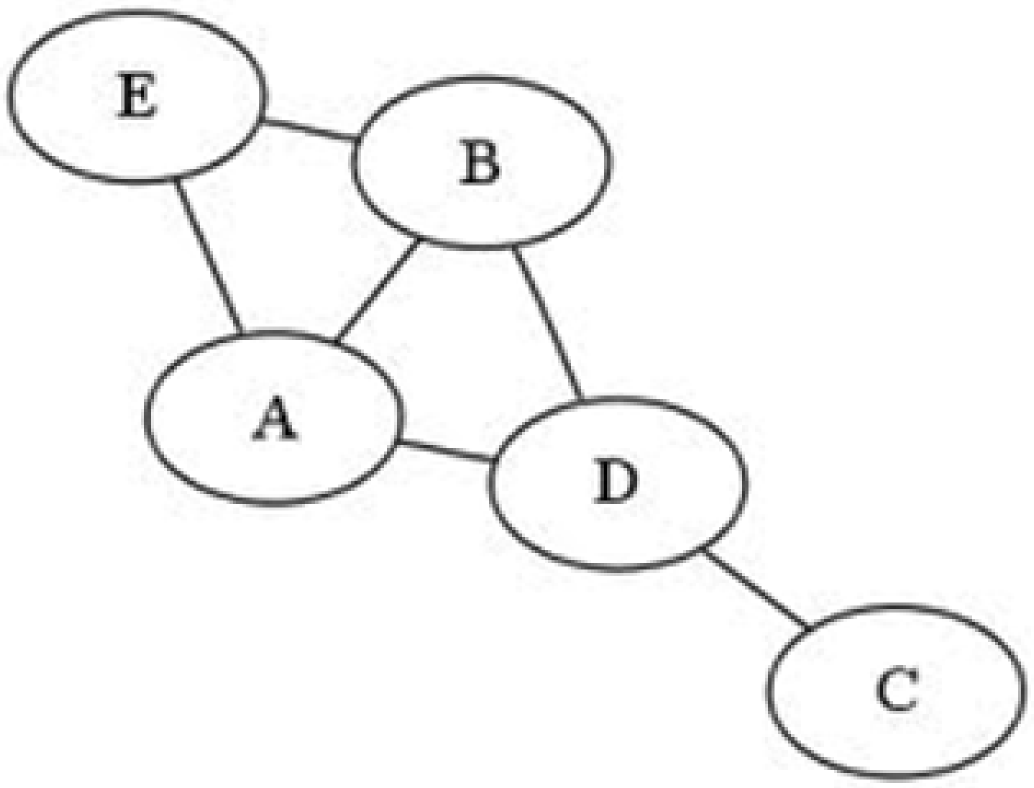


FIGURE 2.5: Undirected graph

The edges connecting the nodes can be singed or unsigned. The signed edges can be positive or negative. Positive signs represent positive relationship and negative signs mean negative relationship. Such positive and negative and negative relationships are considered in social networks like "Slashdot".

2.5 Theories of signed networks

The analysis of the online social networks can be based on two social theories.
1) Structural balance theory and
2) Status theory
The structural balance theory was first organized in social psychology in the mid-20s. According to this theory, two friends with a common enemy are more likely to be friends with each other. On the other hand the phenomenon of two enemies

with a common friend is unlikely to happen. According to this theory "friend of my friend is my friend" and "enemy of my friend is my enemy". The theory is more likely to be viewed as a model of likes and dislikes and it can be represented by the undirected signed graphs.

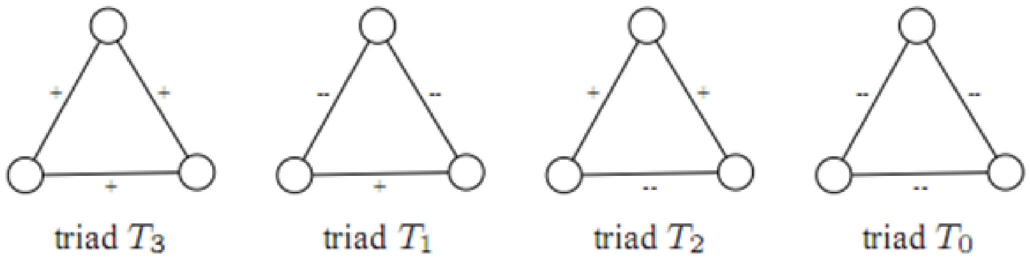


FIGURE 2.6: Balance theory

On the other hand the status theory states that a signed link from suppose Vi to Vl can have more than one interpretation, depending on vi's intention in creating the link. In particular a positive link from Vi to Vl may mean "Vi is my friend", but it also means "Vl has higher status than Vi do". Similarly, a negative link from Vi to Vlmay mean "Vl is Vi's enemy" or "Vi thinks Vl has lower status than Vi".

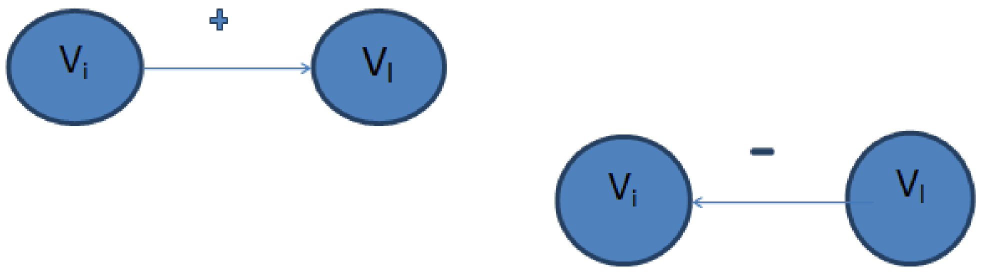


FIGURE 2.7: Status theory

2.6 Link prediction

As we are discussing about a social network or just a part of it to analyze the relations we have to take a fair amount of nodes to analyze the information regarding the nodes. There are links in the networks and the links represent the existing relationships among the nodes. By analyzing the already present links the prediction of the absent links among nodes to be or not to be can be predicted. And this prediction using the information regarding the nodes is the link prediction. Suppose in case of the folowing figure There isn't any relationship between vj and vl at time t. We have to predict the link between them after time t+1.

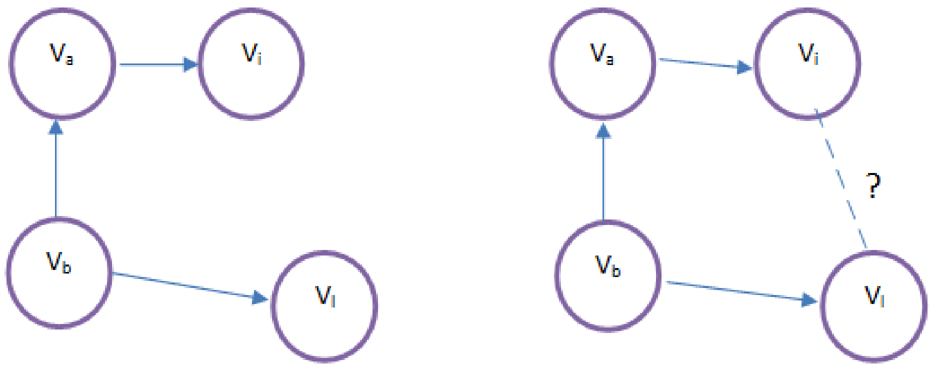


FIGURE 2.8: edges at time t and edges at time t+1

Social networks are highly dynamic, sparse and have collective structure, that's why its outcome is difficult to predict. Moreover, because of early stage popularity, it is possible to estimate the popularity at later stage. The task of accurate prediction the presence of links or connections in a domain is on the one hand important task and on the other hand is very challenging. Since the links from the network, their maintenance and quality, reflect social behaviors of individuals, the research on them can be helpful at the quantitative and qualitative assessment of human relationships in the age of information society. Moreover, link prediction is applicable to a wide variety of areas like bibliographic domain, molecular biology, criminal investigations and recommender systems. If we want to evaluate the sign of a link created from A to B in the context of A and B's relations to additional nodes X with whom they have links. (For example,

the node X in our example who jointly endorses A and B) Thus, we de?ne a contextualized link (more brie?y, ac-link) to be a triple (A, B and X) with the property that a link forms from A to B after each of A and B already has alink either to or from X. Overall there are sixteen differenttypes of c-links, as the edge between X and A can go in either direction and have either sign yielding four possibilities, and similarly for the edge between X and B, for a total of4*4 = 16. For each of these types of c-links we are interested in the frequencies of positive

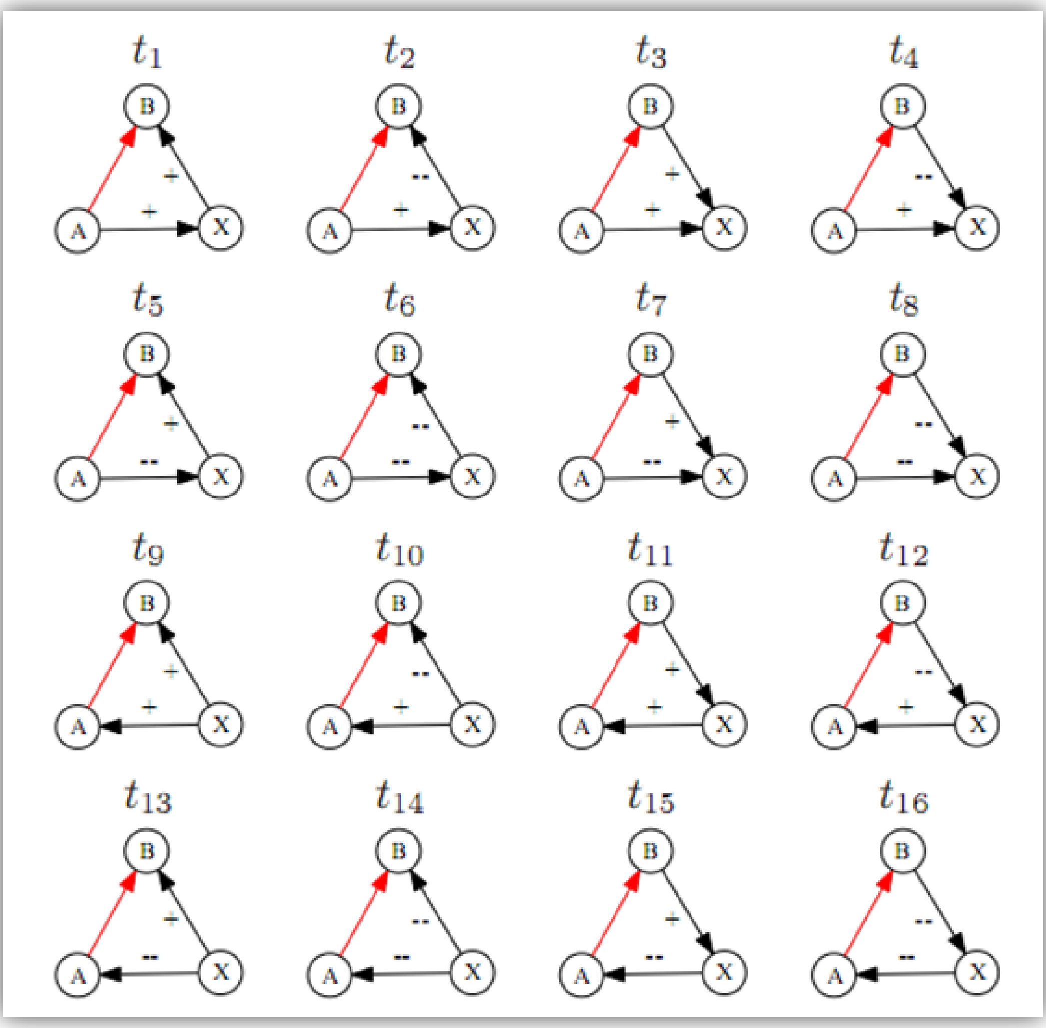


FIGURE 2.9: Triad feature with 16 factors

versus negative labels for the edge from A to B. The figure shows all the possible types of c-links, labeled t1-t16.

2.7 Datasets considered

The datasets that we are following are the publications by Jure Leskovec et al. in which they followed the datasets of Wikipedia, Slashdot and Epinions. In these three large virtual social networks each link is explicitly labeled as positive or negative. A brief description of each of the virtual networks is given below.

2.7.1 Epinions

It is a product review website where a very active community of users is connected into a network of trust and distrust, which is then combined with review ratings to determine which reviews are most authoritative.

2.7.2 Slashdot

It's a technology related news website. In 2002 Slashdot Zoo was introduced which Allows users to tag each other as 'friends' or 'foes'. The relationship as a friend indicate that a user likes another user's comment, while a foe relation means that a user finds another user's comment uninteresting.

2.7.3 Wikipedia

It is an encyclopedia which is authored collectively by its users and the user community is very active. The studied network corresponds to votes cast by users in elections for promoting individuals to role of admin. The voting may be positive or negative. Positive vote indicates a supporting vote and negative an opposing vote. In all networks the background proportion of positive edges isabout the same, with 80 percent of the edges having a positive sign.

Chapter 3

Related works

3.1 Predicting edge sign

Considering the problem of predicting the sign of individualedges in the datasets, the set-up for this problem follows the framework of Guha et al.given a full network with all but one of the edge signs visible, and we are interested in predicting the sign of this single edge whose sign has been suppressed. This can be viewed as leave-one-out cross-validation in the present context, where we learn using the rest of the network and aim to predict the missing sign of a single edge.

3.2 A machine learning approach

Given a directed graph G = (V,E) with a sign (positive or negative) on each edge, we let s(x, y) denote the sign of the edge (x, y)from x to y. That is, s(x, y) = 1 when the sign of (x, y) is positive, +1 when the sign is negative, and 0 when there is no directed edge from x to y. Sometimes we will also be interested in the sign of a directed edge connecting x and y, regardless of its direction; thus, we write s'(x, y) = 1 when there is a positive edge in one of thetwo directions (x, y) or (y, x), and either a positive edge or no edgein the other direction. We write s'(x, y) = ?1 analogously whenthere is a negative edge in one of these directions, and either a negative edge or no edge in the other direction. We write s'(x, y) = 0in all other cases (including when there are edges (x, y) and (y, x)with opposite signs, though this is in fact rare in our datasets). Fordifferent formulations of our task, we will suppose that for a particular edge (u, v), the sign s(u, v) or s'(u, v) is hidden and that weare trying to infer it.

3.3 Features

The definition is begun by a collection of features for our initial machine-learning approach to this problem. The features aredivided into two classes. The first class is based on the (signed)degrees of the nodes, which essentially record the aggregate localrelations of a node to the rest of the world. The second class isbased on the principle from social psychology that we can understand the relationship between individuals u and v through theirjoint relationships with third parties w: for example, is there someone who has a positive relationship toward both u and v, a negativerelationship toward both u and v, or a positive relationship towardone and a negative relationship toward the other. Thus, features ofthis second class are based on two-step paths involving u and v. The ?rst class of features based on degreeareas follows.As we are interested in predicting the sign of the edge from u tov Through their joining relationship with a third party w, we consider outgoing edges from u and incoming edges to v. Specically d+in(v) and d?in(v) is used to denote the number of incoming positive and negative edges to v, respectively. Similarly weuse d+out(u) and d?out(u) to denote the number of outgoing positiveand negative edges from u, respectively. C(u, v) is used to denotethe total number of common neighbors of u and v in an undirectedsense -that is, the number of nodes w such that w is linked by anedge in either direction with both u and v. We will also refer to thisquantity C(u, v) as the embeddedness of the edge (u, v). Our sevendegree features are the ?ve quantities d+in(u), d-in(v), d+out, d-outand C(u, v), together with the total out-degree of u and the totalin-degree of v, which are d+out(u) + d-out(u) and d+in(v) + d+in(v) respectively. For the second class of feature we consider each triad involvingthe edge (u, v), consisting of a node w such that w has an edgeeither to or from u and also an edge either to or from v. There are16 distinct types of triads involving (u, v): the edge between w andu can be in either direction and of either sign, and the edge betweenw and v can also be in either direction and of either sign; this leads to 2*2*2*2 = 16 possibilities. Each of these 16 triad types mayprovide different evidence about the sign of the edge from u to v,some favoring a negative sign and some favoring a positive sign.We encode this information in a 16-dimensional vector specifyingthe number of triads of each type that (u, v) is involved in.

Chapter 4

Implementation procedure

4.1 Dataset description

For implementing the related works we have studied before we have seen that there is 7 vector, 16 vector and 23 vector approach. These approaches are based upon the features that we generated.So our objective would be implementing the 7 vector and 16 vector approaches then combining the two we would get the features. Dataset is available at http://snap.stanford.edu and is also appended at the end of the report. The dataset used in this implementation is from Slashdot which is a technology related news website. Slashdot allowed its users to tag each other as "friends " or "foes" . We used this as our sample data using a friends link to be positive (i.e. having a value of 1) and a foe (i.e. having a value of -1). We sampled this data and randomly assigned some of the links to be the test data. From this test data , we evaluate the effectiveness of our schemes.[6]

4.2 Algorithms used

- Start with a graph G(V, E) with some the edges having value s (u , v) =1 for positive edges and s (u , v) = -1 for negative edges.[1]

- From the particular node we have calculated d+out(i.e. the number of positive edges that are going out), d-out (i.e. the number of negative edges that are going out) ,d+in (i.e. the number of positive edges that are incoming to the node), d-in (i.e. the number of negative edges that are incoming to the node).[1]

- From a node u if there is a signed edge to another node w and again from w if there is a signed edge to another node v . From u to w there could be 2*2 = 4 types of possibilities (i.e. direction wise and sign wise) and from w to v there could be 2*2 =4 types of possibilities (i.e. direction wise and sign wise). So 2*2*2*2 = 16 types of triads.[1][2]

- From a node u if there is a signed edge to another node w and again from w if there is a signed edge to another node v and from v if there is a signed edge to another node x. From u to w there could be 2*2 = 4 types of possibilities (i.e. direction wise and sign wise) and from w to v there could be 2*2 =4 types of possibilities (i.e. direction wise and sign wise). From v to x there could be 2*2 = 4 types of possibilities (i.e. direction wise and sign wise) So 2*2*2*2*2*2 = 64 types of quads.[1][2]

- From a particular node to another node we calculate the number of triads it forms and then from the existing dataset we check out all the combination of the triads and thus we evaluate.

- As balance theory states that odd number of positive edges form a balance set. So enhancing that theory we can say that for the quad even number of positive edges can form a balance set.

4.3 Choosing framework

To implement the features that we are looking for are generated by:
-Writing code in Java We at first took the data set as our input data. Then from the input data set we have created the features.

4.4 Feature creation

To create different features we simply took the data set and then implement some algorithms.The features are listed below:

- A particular node's total number of positive edges both going in and out and also total number of negative edges both going in and out
- From a particular node to every other node in the dataset what is the class feature are listed

The class features are defined as follows:

From	To	Edge	Feature(class)
U	V	1	1
U	V	-1	-1
V	U	1	2
V	U	-1	-2

From	To	Edge	From	To	Edge	Feature(class)
U	V	1	V	U	1	3
U	V	1	V	U	-1	4
U	V	1	V	U	1	-3
U	V	1	V	U	-1	-4

From table 2 we have seen that it is indicating every node having a bidirectional edge from a node to other node. For this type of edges we have defined different feature class as mentioned in the table.

- Triad feature involving two particular nodes and the other node could be any of the nodes from the whole dataset.This is where comes the prediction . Because from the two nodes connecting with some other node gives us different results resulting triads are different in this case. The configuration of the triads are considered as it is mentioned in the theory.[1]

- Counting all the triads of a particular node. Later on it will be used to predict the signs of the links.

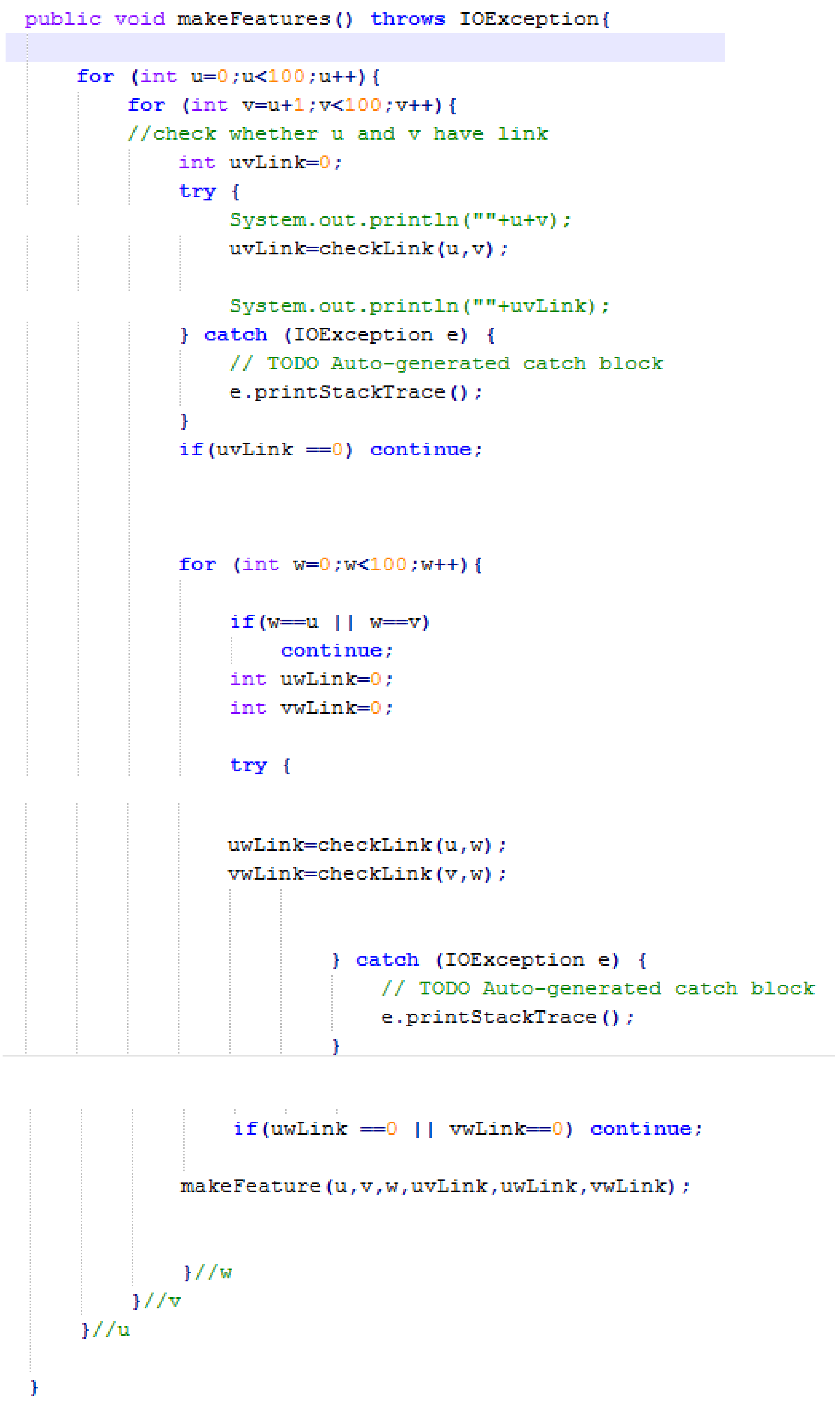
- From the existing dataset we calculate all the possibilities of links between every two nodes. Then we list down all the nodes and their adjacent nodes with their present relationship types and named them as t1, t2, t3t16. The naming was done by comparing the relationships with the 16 type of triads.

- For the purpose of evaluating the current dataset we simply check out all the given relationships among neighboring nodes. Then we evaluate it with the triad predictions that have been extracted by us.

- Also to implement quad we have considered four nodes that will create the quad. We have not taken the diagonal relationship among nodes. Only we have taken consideration the adjacent nodes relationships. There are 64 possible quads for the nodes. It can be demonstrated like 2*2*2*2*2*2 = 64. Because we have taken consideration of both sign and direction. The relationship is shown to u -¿v -¿ w -¿ x. All of these quads have been named q1, q2, q3..q64.

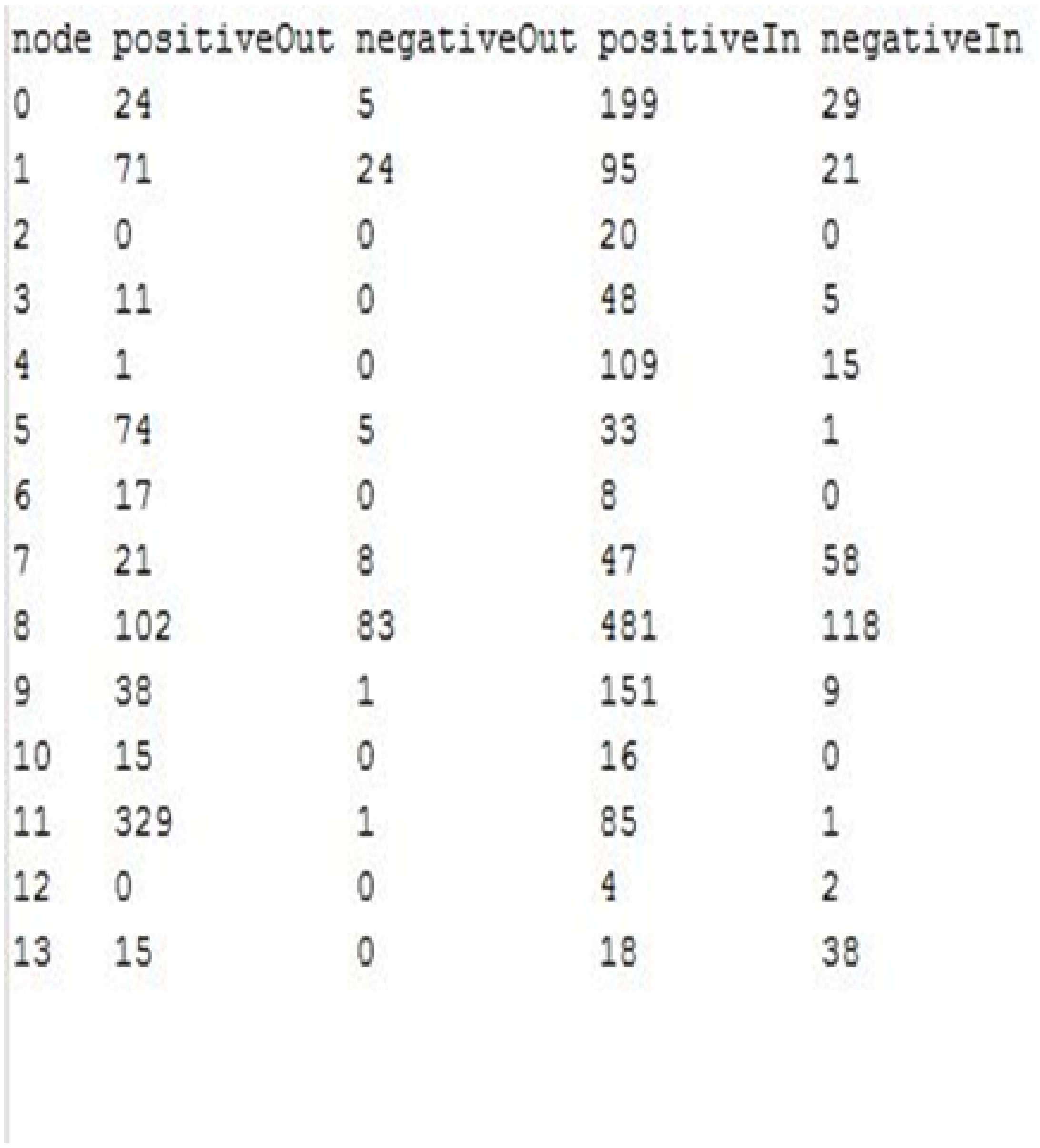
4.4.1 First feature

The pseudo code and the generated feature's snapshot for the first feature is given below:



```
public void makeFeatures() throws IOException{

    for (int u=0;u<100;u++){
        for (int v=u+1;v<100;v++){
        //check whether u and v have link
            int uvLink=0;
            try {
                System.out.println(""+u+v);
                uvLink=checkLink(u,v);

                System.out.println(""+uvLink);
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
            if(uvLink ==0) continue;



            for (int w=0;w<100;w++){

                if(w==u || w==v)
                    continue;
                int uwLink=0;
                int vwLink=0;

                try {


                uwLink=checkLink(u,w);
                vwLink=checkLink(v,w);


                        } catch (IOException e) {
                            // TODO Auto-generated catch block
                            e.printStackTrace();
                        }


                if(uwLink ==0 || vwLink==0) continue;

            makeFeature(u,v,w,uvLink,uwLink,vwLink);


            }//w
        }//v
    }//u

}
```

FIGURE 4.1: Pseudo code for feature 1

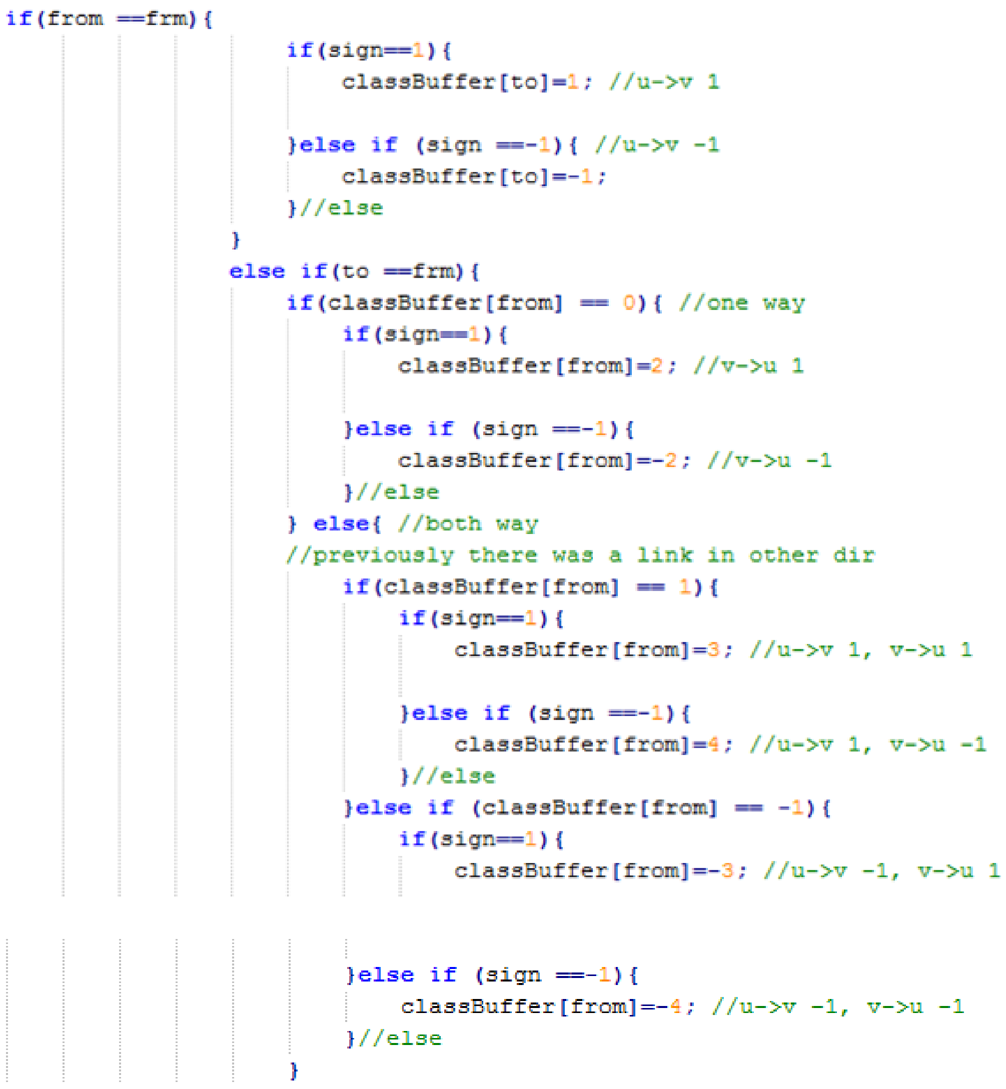


node	positiveOut	negativeOut	positiveIn	negativeIn
0	24	5	199	29
1	71	24	95	21
2	0	0	20	0
3	11	0	48	5
4	1	0	109	15
5	74	5	33	1
6	17	0	8	0
7	21	8	47	58
8	102	83	481	118
9	38	1	151	9
10	15	0	16	0
11	329	1	85	1
12	0	0	4	2
13	15	0	18	38

FIGURE 4.2: 1st feature

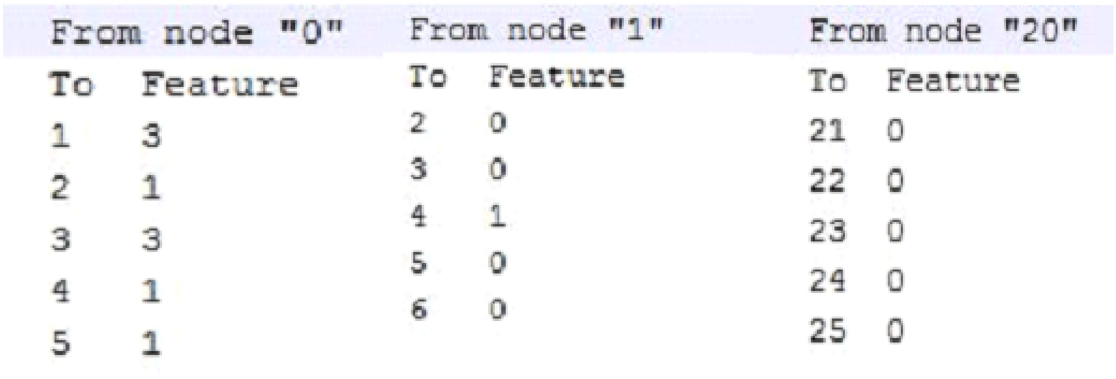
4.4.2 Second feature

The pseudo code and the generated feature's snapshot for the second feature is given below:



```
if(from ==frm){
                    if(sign==1){
                        classBuffer[to]=1; //u->v 1

                    }else if (sign ==-1){ //u->v -1
                        classBuffer[to]=-1;
                    }//else
                }
                else if(to ==frm){
                    if(classBuffer[from] == 0){ //one way
                        if(sign==1){
                            classBuffer[from]=2; //v->u 1

                        }else if (sign ==-1){
                            classBuffer[from]=-2; //v->u -1
                        }//else
                    } else{ //both way
                    //previously there was a link in other dir
                        if(classBuffer[from] == 1){
                            if(sign==1){
                                classBuffer[from]=3; //u->v 1, v->u 1

                            }else if (sign ==-1){
                                classBuffer[from]=4; //u->v 1, v->u -1
                            }//else
                        }else if (classBuffer[from] == -1){
                            if(sign==1){
                                classBuffer[from]=-3; //u->v -1, v->u 1


                        }else if (sign ==-1){
                            classBuffer[from]=-4; //u->v -1, v->u -1
                        }//else
                    }
```

FIGURE 4.3: Pseudo code for feature 2

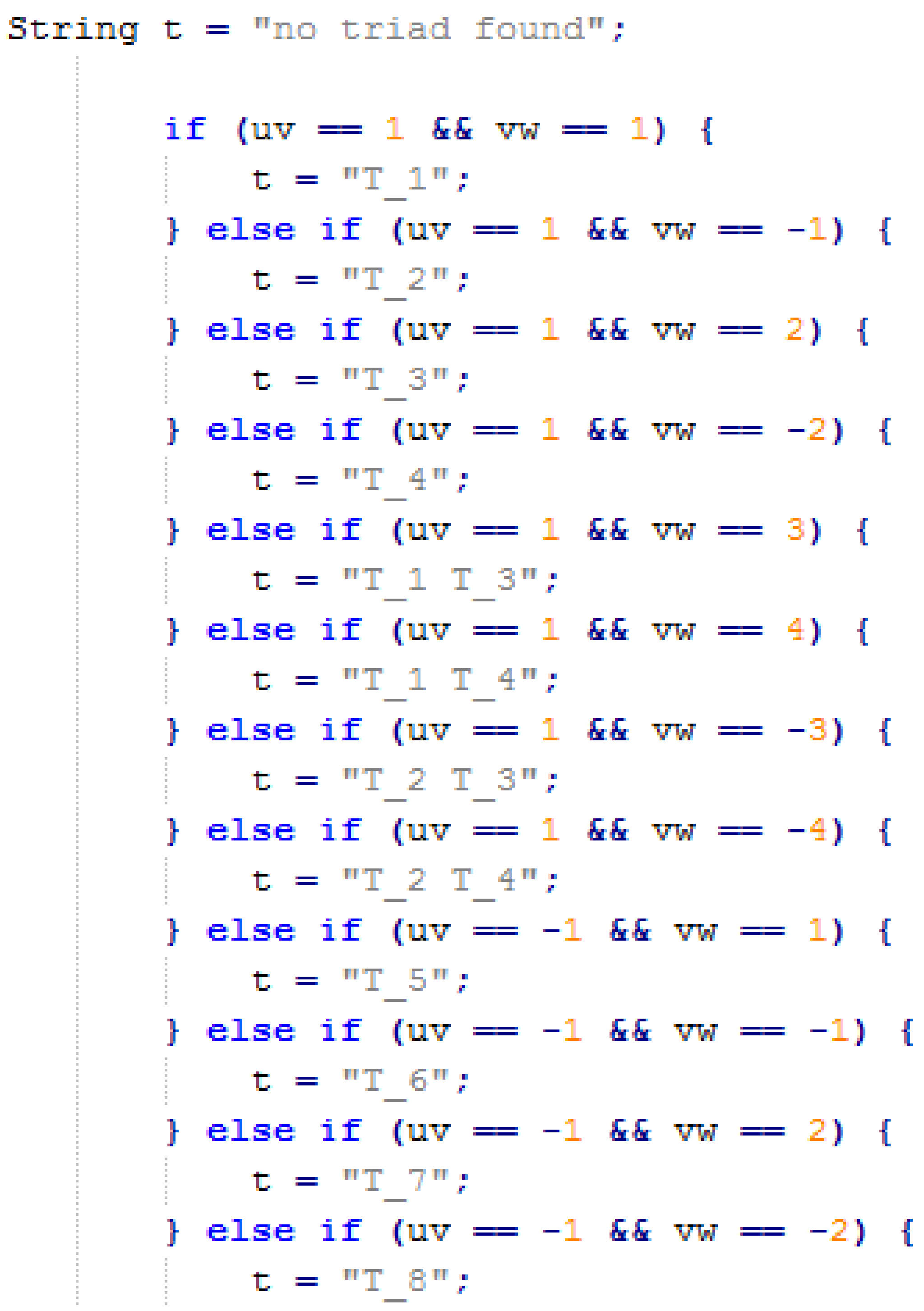


```
From node "0"      From node "1"      From node "20"
To  Feature        To  Feature        To  Feature
1   3              2   0              21  0
2   1              3   0              22  0
3   3              4   1              23  0
4   1              5   0              24  0
5   1              6   0              25  0
```

FIGURE 4.4: 2nd feature

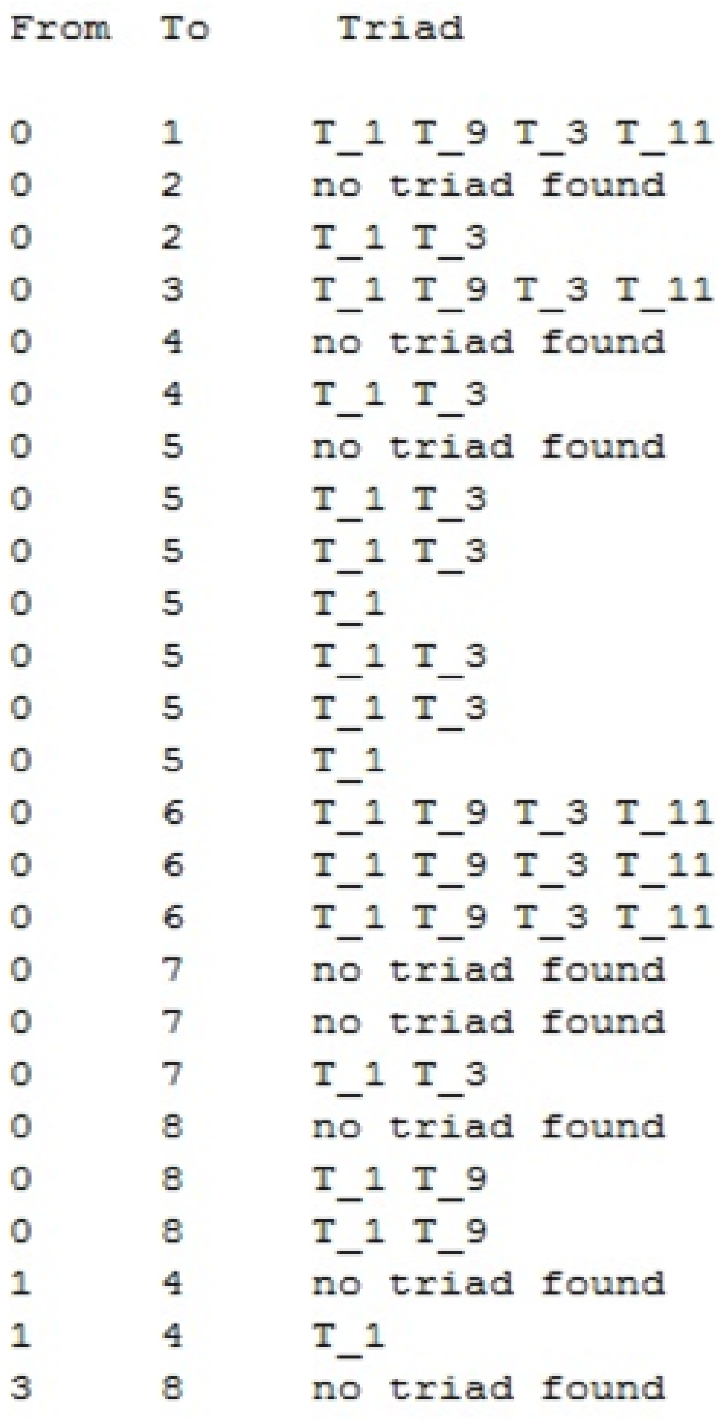
4.4.3 Third feature

The pseudo code and the generated feature's snapshot for the third feature is given below:



```
String t = "no triad found";

        if (uv == 1 && vw == 1) {
            t = "T_1";
        } else if (uv == 1 && vw == -1) {
            t = "T_2";
        } else if (uv == 1 && vw == 2) {
            t = "T_3";
        } else if (uv == 1 && vw == -2) {
            t = "T_4";
        } else if (uv == 1 && vw == 3) {
            t = "T_1 T_3";
        } else if (uv == 1 && vw == 4) {
            t = "T_1 T_4";
        } else if (uv == 1 && vw == -3) {
            t = "T_2 T_3";
        } else if (uv == 1 && vw == -4) {
            t = "T_2 T_4";
        } else if (uv == -1 && vw == 1) {
            t = "T_5";
        } else if (uv == -1 && vw == -1) {
            t = "T_6";
        } else if (uv == -1 && vw == 2) {
            t = "T_7";
        } else if (uv == -1 && vw == -2) {
            t = "T_8";
```

Figure 4.5: Pseudo code for feature 3

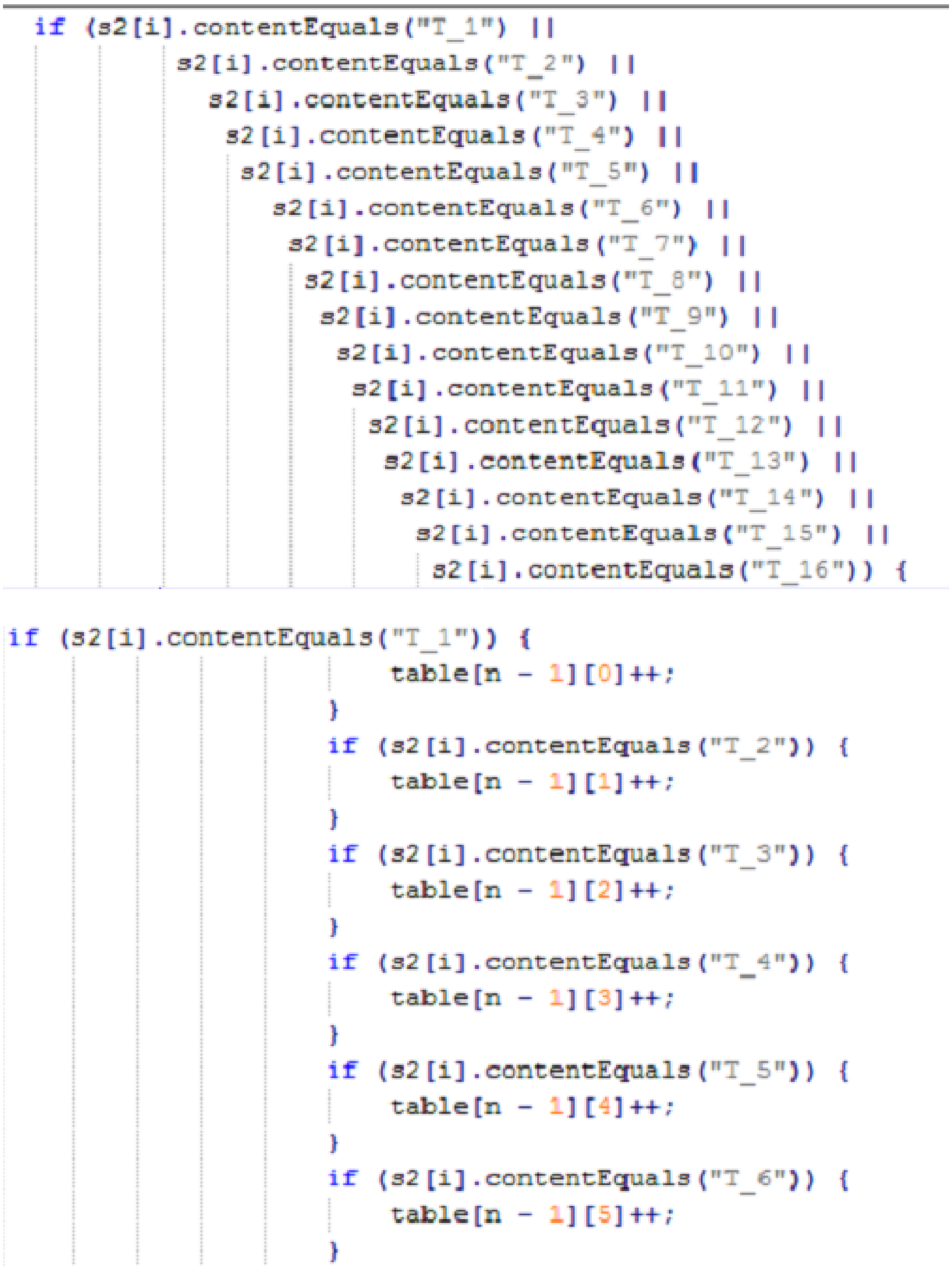


```
From  To     Triad

0     1     T_1 T_9 T_3 T_11
0     2     no triad found
0     2     T_1 T_3
0     3     T_1 T_9 T_3 T_11
0     4     no triad found
0     4     T_1 T_3
0     5     no triad found
0     5     T_1 T_3
0     5     T_1 T_3
0     5     T_1
0     5     T_1 T_3
0     5     T_1 T_3
0     5     T_1
0     6     T_1 T_9 T_3 T_11
0     6     T_1 T_9 T_3 T_11
0     6     T_1 T_9 T_3 T_11
0     7     no triad found
0     7     no triad found
0     7     T_1 T_3
0     8     no triad found
0     8     T_1 T_9
0     8     T_1 T_9
1     4     no triad found
1     4     T_1
3     8     no triad found
```

Figure 4.6: 3rd feature

4.4.4 Fourth feature

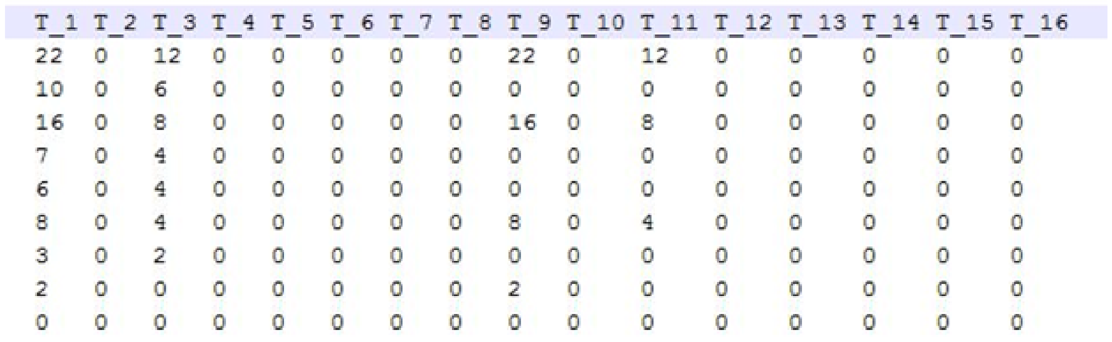
The pseudo code and the generated feature's snapshot for the fourth feature is given below:



```
if (s2[i].contentEquals("T_1") ||
            s2[i].contentEquals("T_2") ||
              s2[i].contentEquals("T_3") ||
               s2[i].contentEquals("T_4") ||
                s2[i].contentEquals("T_5") ||
                  s2[i].contentEquals("T_6") ||
                   s2[i].contentEquals("T_7") ||
                    s2[i].contentEquals("T_8") ||
                     s2[i].contentEquals("T_9") ||
                      s2[i].contentEquals("T_10") ||
                       s2[i].contentEquals("T_11") ||
                        s2[i].contentEquals("T_12") ||
                         s2[i].contentEquals("T_13") ||
                          s2[i].contentEquals("T_14") ||
                           s2[i].contentEquals("T_15") ||
                            s2[i].contentEquals("T_16")) {
```

```
if (s2[i].contentEquals("T_1")) {
                        table[n - 1][0]++;
                    }
                    if (s2[i].contentEquals("T_2")) {
                        table[n - 1][1]++;
                    }
                    if (s2[i].contentEquals("T_3")) {
                        table[n - 1][2]++;
                    }
                    if (s2[i].contentEquals("T_4")) {
                        table[n - 1][3]++;
                    }
                    if (s2[i].contentEquals("T_5")) {
                        table[n - 1][4]++;
                    }
                    if (s2[i].contentEquals("T_6")) {
                        table[n - 1][5]++;
                    }
```

FIGURE 4.7: Pseudo code for feature 4

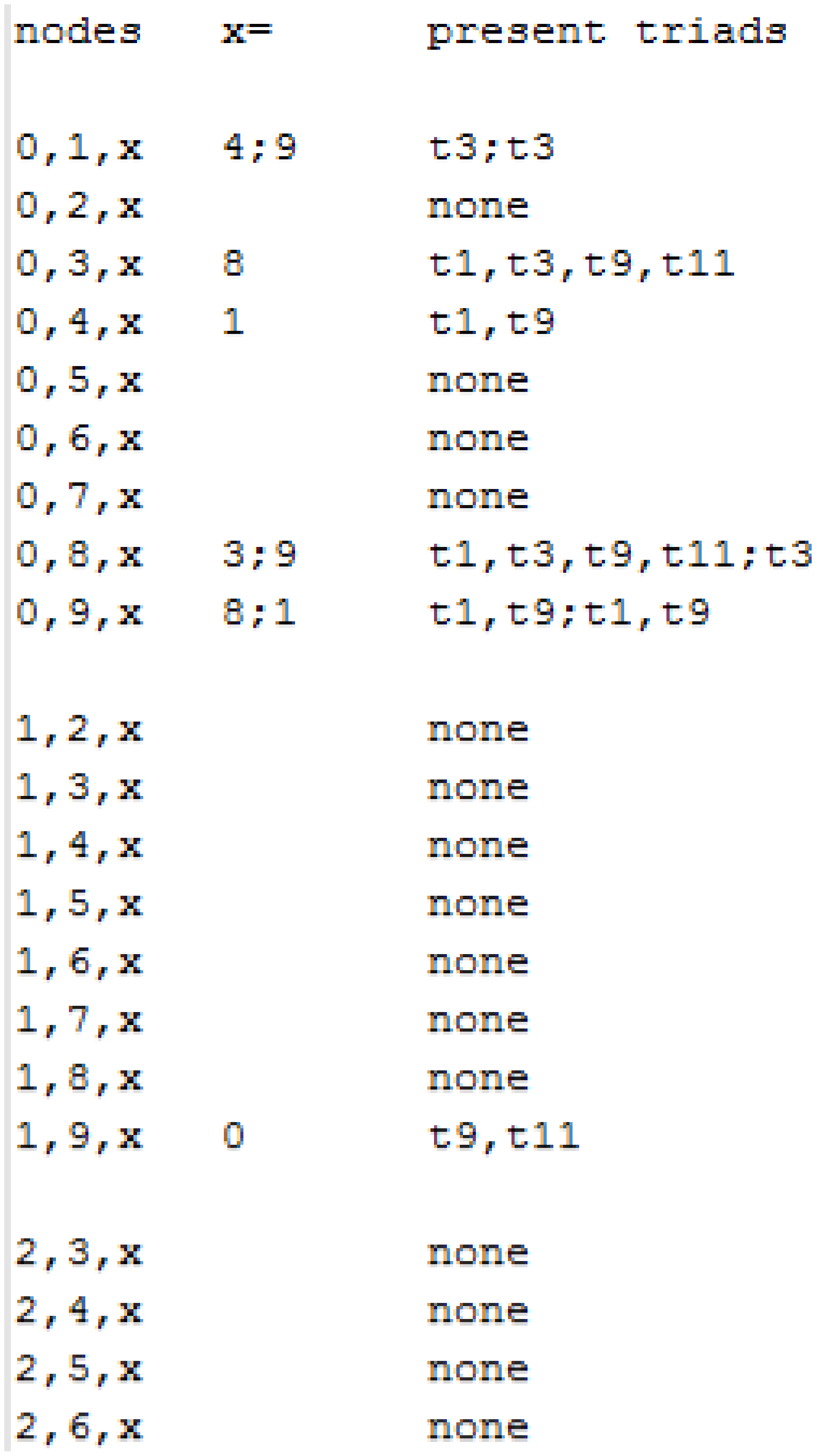


T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_10	T_11	T_12	T_13	T_14	T_15	T_16
22	0	12	0	0	0	0	0	22	0	12	0	0	0	0	0
10	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	8	0	0	0	0	0	16	0	8	0	0	0	0	0
7	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	4	0	0	0	0	0	8	0	4	0	0	0	0	0
3	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4.8: 4th feature

4.4.5 Fifth feature

The generated feature's snapshot for the fifth feature is given below:

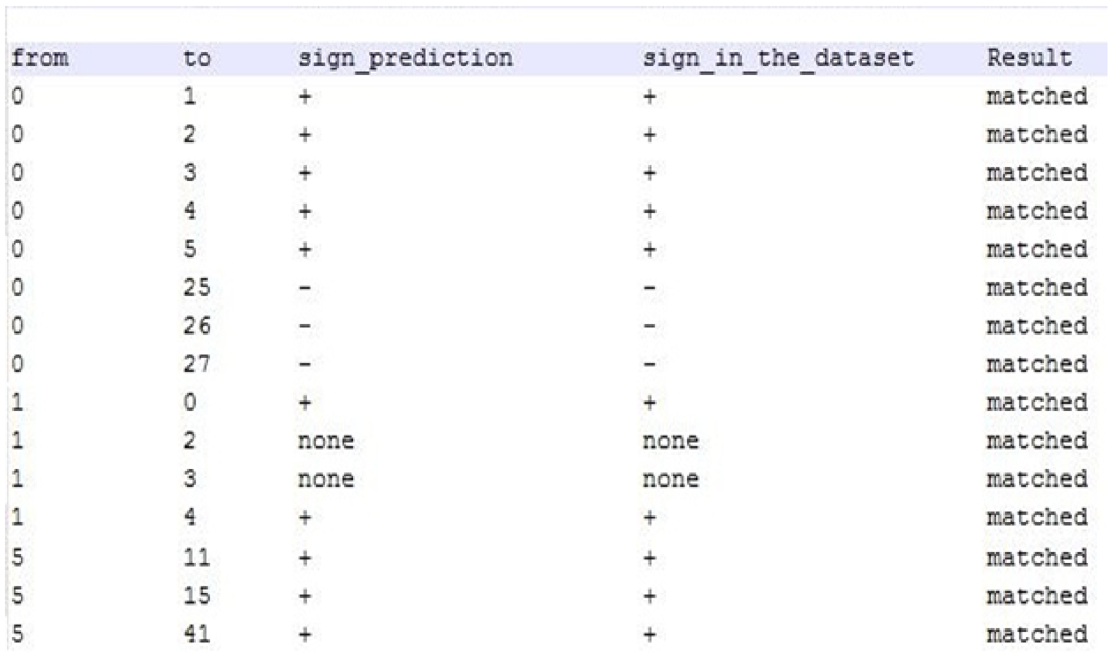


```
nodes    x=        present triads

0,1,x    4;9       t3;t3
0,2,x              none
0,3,x    8         t1,t3,t9,t11
0,4,x    1         t1,t9
0,5,x              none
0,6,x              none
0,7,x              none
0,8,x    3;9       t1,t3,t9,t11;t3
0,9,x    8;1       t1,t9;t1,t9

1,2,x              none
1,3,x              none
1,4,x              none
1,5,x              none
1,6,x              none
1,7,x              none
1,8,x              none
1,9,x    0         t9,t11

2,3,x              none
2,4,x              none
2,5,x              none
2,6,x              none
```

FIGURE 4.9: 5th feature

4.4.6 Sixth feature

The generated feature's snapshot for the sixth feature is given below:

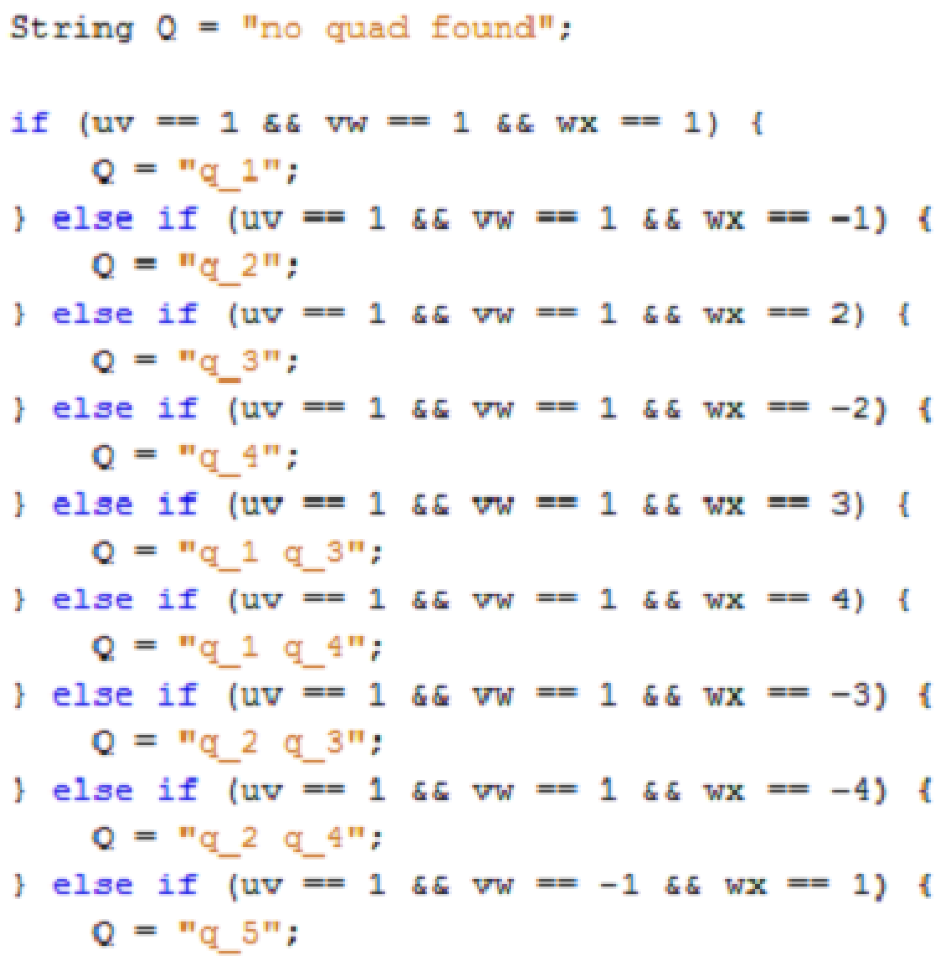


from	to	sign_prediction	sign_in_the_dataset	Result
0	1	+	+	matched
0	2	+	+	matched
0	3	+	+	matched
0	4	+	+	matched
0	5	+	+	matched
0	25	-	-	matched
0	26	-	-	matched
0	27	-	-	matched
1	0	+	+	matched
1	2	none	none	matched
1	3	none	none	matched
1	4	+	+	matched
5	11	+	+	matched
5	15	+	+	matched
5	41	+	+	matched

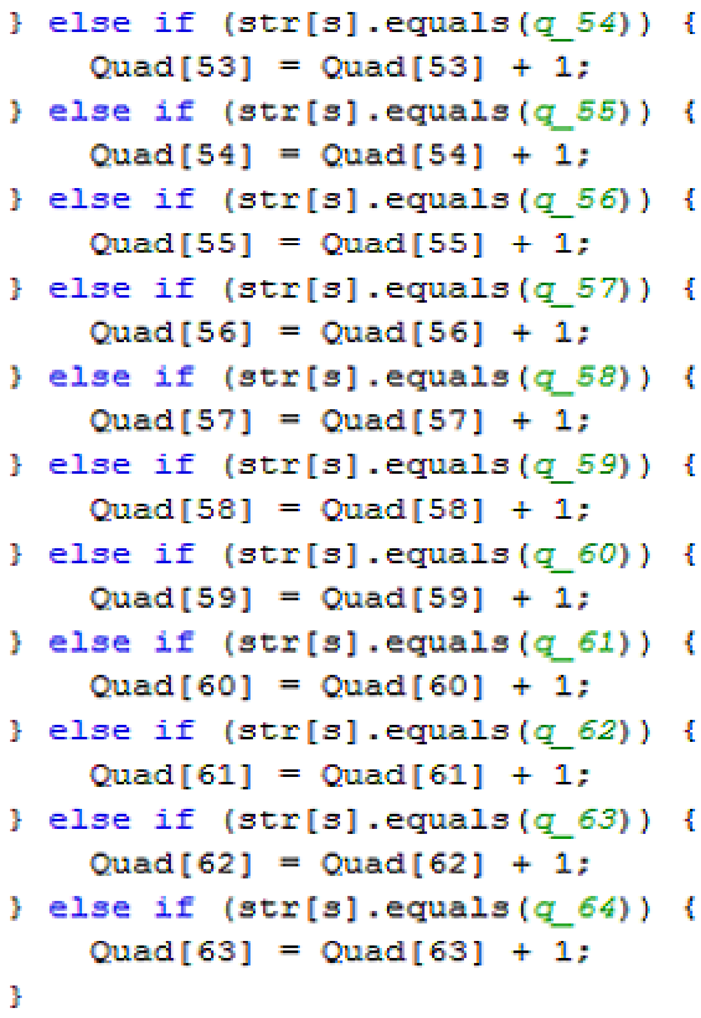
FIGURE 4.10: 6th feature

4.4.7 Seventh feature

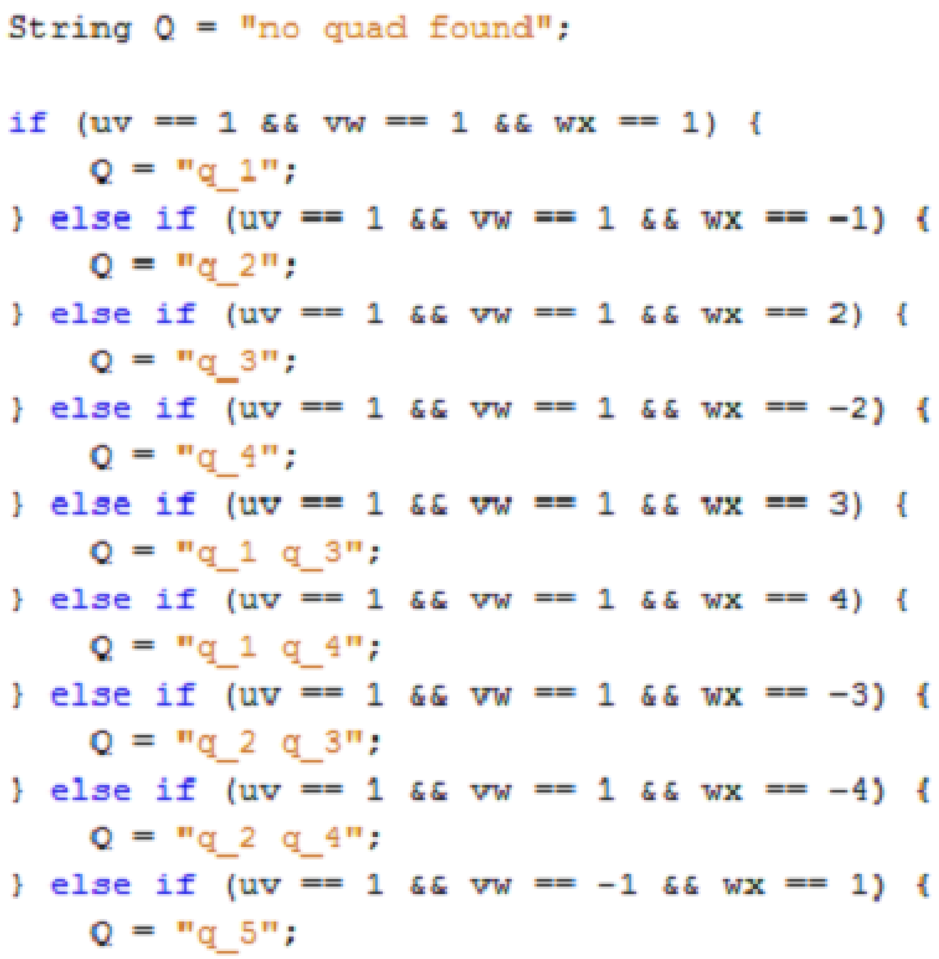
The pseudo code and the generated feature's snapshot for the seventh feature is given below:



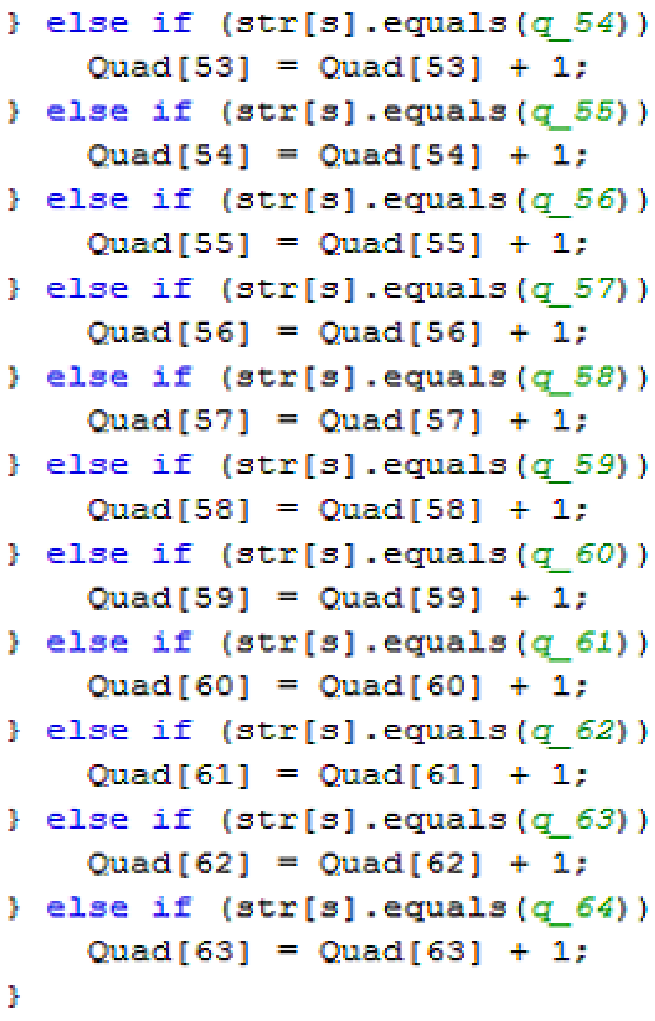
```
String Q = "no quad found";

if (uv == 1 && vw == 1 && wx == 1) {
     Q = "q_1";
} else if (uv == 1 && vw == 1 && wx == -1) {
     Q = "q_2";
} else if (uv == 1 && vw == 1 && wx == 2) {
     Q = "q_3";
} else if (uv == 1 && vw == 1 && wx == -2) {
     Q = "q_4";
} else if (uv == 1 && vw == 1 && wx == 3) {
     Q = "q_1 q_3";
} else if (uv == 1 && vw == 1 && wx == 4) {
     Q = "q_1 q_4";
} else if (uv == 1 && vw == 1 && wx == -3) {
     Q = "q_2 q_3";
} else if (uv == 1 && vw == 1 && wx == -4) {
     Q = "q_2 q_4";
} else if (uv == 1 && vw == -1 && wx == 1) {
     Q = "q_5";
```



```
} else if (str[s].equals(q_54)) {
     Quad[53] = Quad[53] + 1;
} else if (str[s].equals(q_55)) {
     Quad[54] = Quad[54] + 1;
} else if (str[s].equals(q_56)) {
     Quad[55] = Quad[55] + 1;
} else if (str[s].equals(q_57)) {
     Quad[56] = Quad[56] + 1;
} else if (str[s].equals(q_58)) {
     Quad[57] = Quad[57] + 1;
} else if (str[s].equals(q_59)) {
     Quad[58] = Quad[58] + 1;
} else if (str[s].equals(q_60)) {
     Quad[59] = Quad[59] + 1;
} else if (str[s].equals(q_61)) {
     Quad[60] = Quad[60] + 1;
} else if (str[s].equals(q_62)) {
     Quad[61] = Quad[61] + 1;
} else if (str[s].equals(q_63)) {
     Quad[62] = Quad[62] + 1;
} else if (str[s].equals(q_64)) {
     Quad[63] = Quad[63] + 1;
}
```



```
String Q = "no quad found";

if (uv == 1 && vw == 1 && wx == 1) {
    Q = "q_1";
} else if (uv == 1 && vw == 1 && wx == -1) {
    Q = "q_2";
} else if (uv == 1 && vw == 1 && wx == 2) {
    Q = "q_3";
} else if (uv == 1 && vw == 1 && wx == -2) {
    Q = "q_4";
} else if (uv == 1 && vw == 1 && wx == 3) {
    Q = "q_1 q_3";
} else if (uv == 1 && vw == 1 && wx == 4) {
    Q = "q_1 q_4";
} else if (uv == 1 && vw == 1 && wx == -3) {
    Q = "q_2 q_3";
} else if (uv == 1 && vw == 1 && wx == -4) {
    Q = "q_2 q_4";
} else if (uv == 1 && vw == -1 && wx == 1) {
    Q = "q_5";
```



```
} else if (str[s].equals(q_54)) {
    Quad[53] = Quad[53] + 1;
} else if (str[s].equals(q_55)) {
    Quad[54] = Quad[54] + 1;
} else if (str[s].equals(q_56)) {
    Quad[55] = Quad[55] + 1;
} else if (str[s].equals(q_57)) {
    Quad[56] = Quad[56] + 1;
} else if (str[s].equals(q_58)) {
    Quad[57] = Quad[57] + 1;
} else if (str[s].equals(q_59)) {
    Quad[58] = Quad[58] + 1;
} else if (str[s].equals(q_60)) {
    Quad[59] = Quad[59] + 1;
} else if (str[s].equals(q_61)) {
    Quad[60] = Quad[60] + 1;
} else if (str[s].equals(q_62)) {
    Quad[61] = Quad[61] + 1;
} else if (str[s].equals(q_63)) {
    Quad[62] = Quad[62] + 1;
} else if (str[s].equals(q_64)) {
    Quad[63] = Quad[63] + 1;
}
```

Chapter 5

Link prediction based on Triad relationship

5.1 Process of prediction

From the existing dataset of Slashdot among every 3 neighboring interconnected nodes we have defined them as 16 types of triads. These triads are formed if they are interconnected with each other in such a way that, suppose we are working with nodes A, B and C. If node A and C are connected and B and C are connected, they form a triad. And from this triad we can predict the relationship between node A and B. The prediction of the relationship consists of the sign and the direction prediction between the two target nodes. The prediction can be shown in the following way-

From the above figure we can see that if two nodes form a triad with another node(the 3rd node is not our concern) we can evaluate the relationship based on that.In the first one node A is connected with node B and as it forms triad no.1 and if the two nodes form the other triads like triad no.2 to triad no.16.Then the relationship can be determined such like that.
We have used both balance and the status theories for doing the prediction based on the interrelationship among three nodes. The balance theory determines the sign of the predicted link and status theory determines the direction of the predicted link.

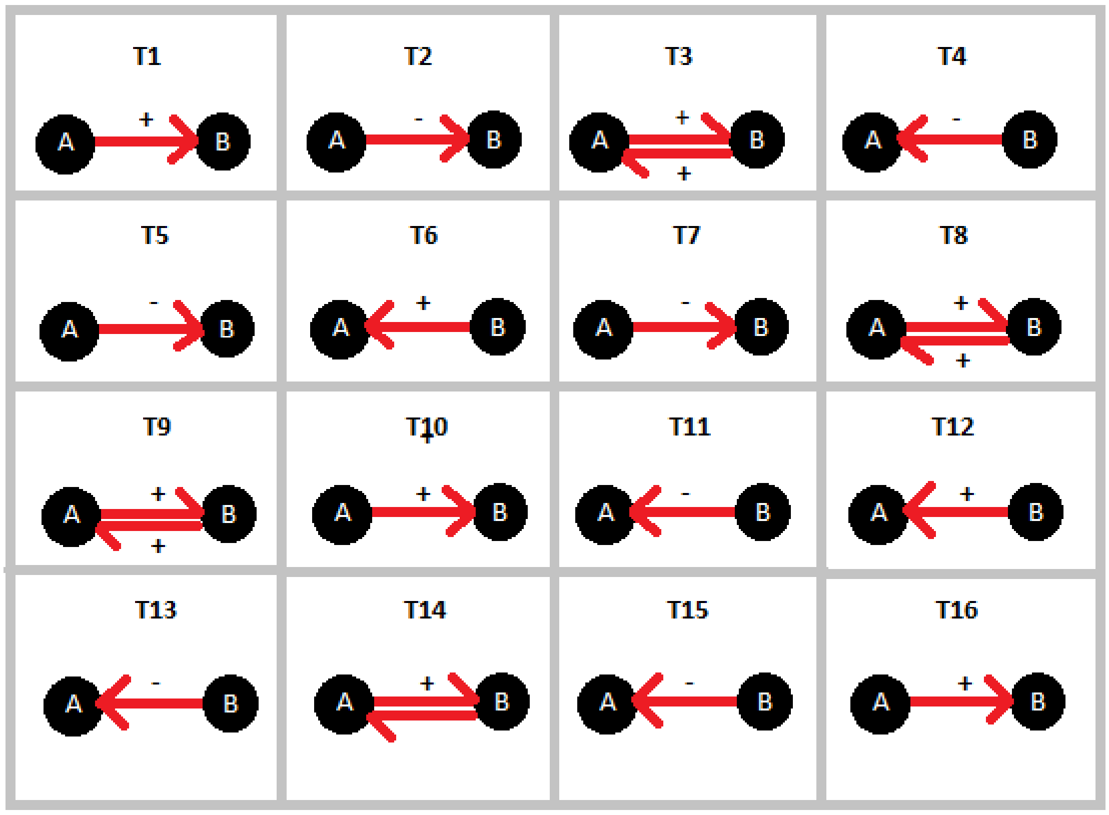
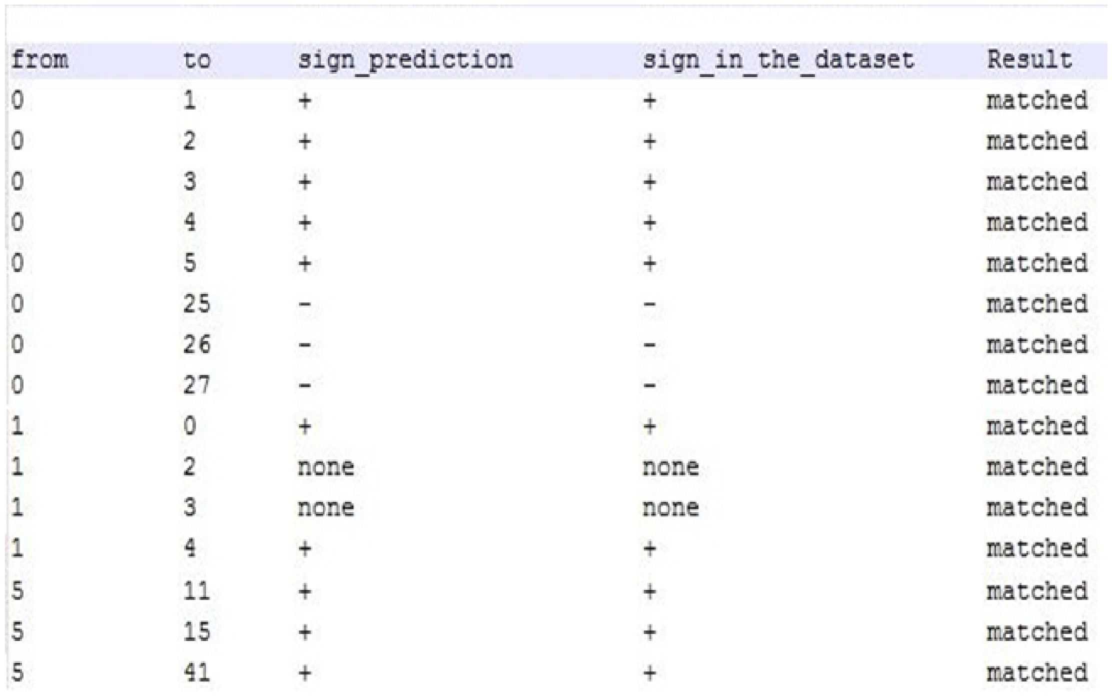


FIGURE 5.1: Link prediction

5.2 Evaluation of the prediction

The evaluation process can be done by two different approaches: - Using different datasets of different times to train and evaluate - Using a single dataset to train and evaluate As we have used only 1 dataset of Slashdot which is from a specific time, we had to use it for both testing and evaluating. We have worked with the nodes whose relationships are already present in the dataset. We have taken every 3 nodes adjacent to each other, considered their relationship bearing in mind that one of the links are missing among 3 and predict the sign and the direction of the link considering the relation type between the rest two nodes. The prediction gives us the desired signs and directions of the hypothetical missing links which are not actually missing but we assumed them to be missing. Now by comparing the predicted relationships with original dataset we found out that most of the prediction was correct with an accuracy of near about 100 percent.

By the evaluation it can be concluded that the prediction done using the neighboring nodes were correct. As the evaluation was done comparing them with the present links among the nodes, the prediction of the missing links among the nodes



from	to	sign_prediction	sign_in_the_dataset	Result
0	1	+	+	matched
0	2	+	+	matched
0	3	+	+	matched
0	4	+	+	matched
0	5	+	+	matched
0	25	-	-	matched
0	26	-	-	matched
0	27	-	-	matched
1	0	+	+	matched
1	2	none	none	matched
1	3	none	none	matched
1	4	+	+	matched
5	11	+	+	matched
5	15	+	+	matched
5	41	+	+	matched

Figure 5.2: Evaluation of link prediction

can also be predicted and they can be evaluated if the dataset from a later time can be gathered.

Chapter 6

Quad

6.1 Implementing Quad

Previously we worked with triads, where 3 nodes and their interrelations were used. Now to extend the size of our basic structure we have tried to implement ”quad”, where 4 nodes and their interrelations are used to make a basic structure.

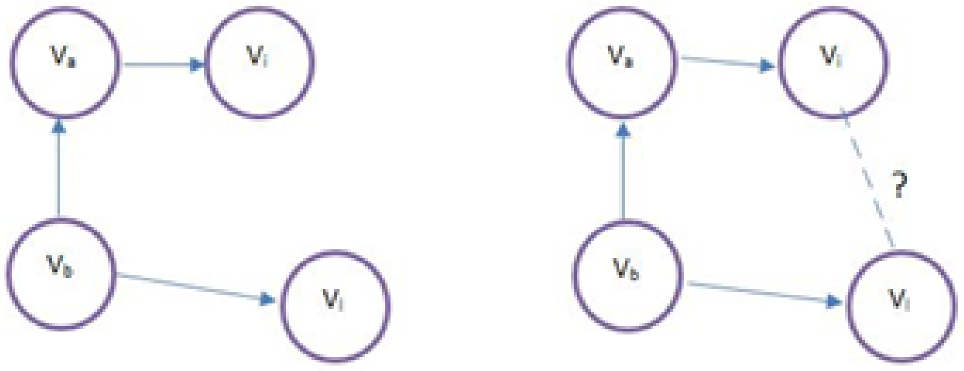


FIGURE 6.1: quad

According to the figure above Vl is connected to Vb, Vb is connected to Va and Va is connected to Vi. They all form a quad if Vi is connected with Vl. The goal of creating this quad is to predict the relationship between Vi and Vl. The relationships among nodes in the quads are denoted by signed directed straight lines.

The above configurations consist of 1-16 and 17-32 configurations of quads, where 3 nodes are interconnected and the other to be predicted.

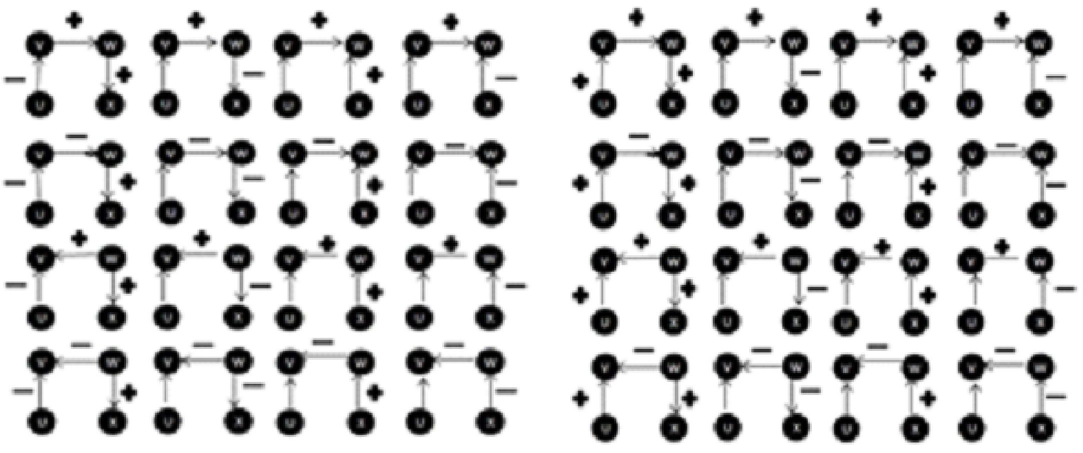


FIGURE 6.2: Quad configuration (1-32)

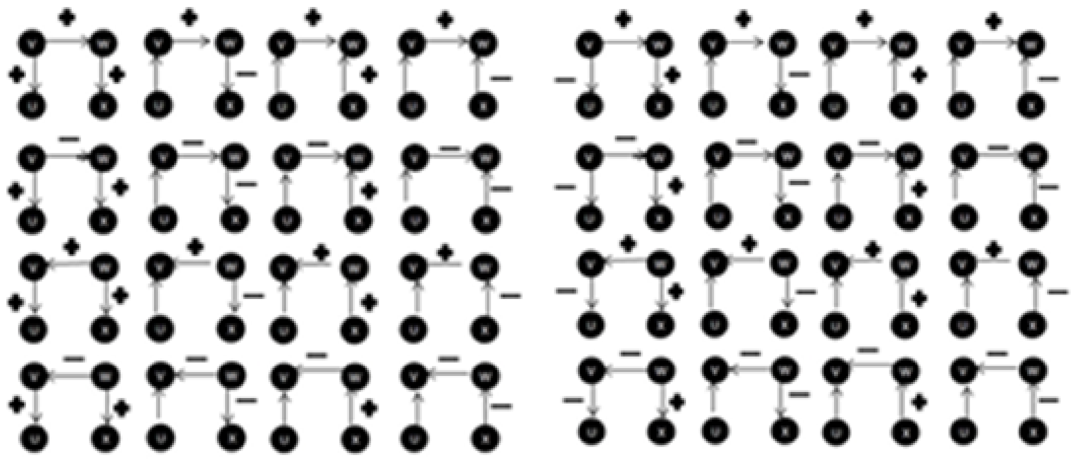


FIGURE 6.3: Quad configuration (33-64)

The above configurations consist of 33-48 and 49-64 configurations of quads, where 3 nodes are interconnected and the other to be predicted.

As all the links have both signs and directions, these signs and directions have 64 different configurations, which we calculated as (2*2)*(2*2)*(2*2) = 64. Here (2*2) denotes the possible configurations between two nodes and we have to use the relationship of 3 such possible combinations, which add up to 64.

6.2 Balance theory for Quad

We have already dealt with balance theory for triads. Now we are trying to implement balance theory for quad. According to our assumption based on the original balance theory implemented on triads, we can denote that a quad is balanced if it has even number of positive edges. For odd number of positive edges the quads are not assumed to be balanced.

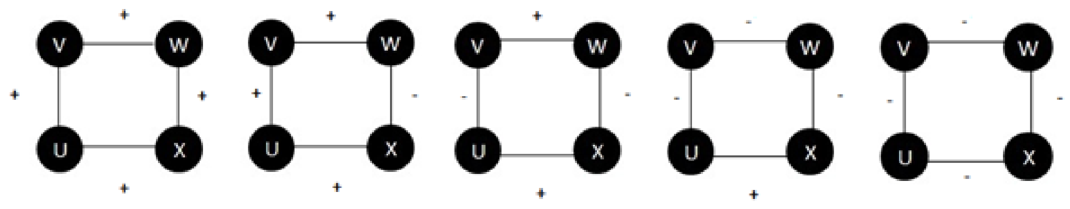


Figure 6.4: Balance theory assumption for quad

In the figure above the 1st configuration denotes all the edges to be of positive links. That means all them are friends to each other. And this configuration is balanced according to our assumed theory, as it has 4 positive edges, which is positive.
The 2nd configuration denotes 3 positive edges and one negative edge. It is not balanced as there is an odd number of positive edges.
The 3rd configuration denotes 2 positive and 2 negative edges. That means "the enemy of my friend is the friend of my enemy". As it is logical and it has even number of positive edges this configuration is balanced.
The 4th configuration denotes 1 positive and 3 negative edges. This configuration is not balanced.
The 5th configuration is not balanced either.

Chapter 7

Summary and future work

7.1 Summary

Social network is a highly dynamic object. It changes over time. So when it comes to social networking analysis it becomes a challenge because it is a vast field with lots of actors (i.e. users).The relationship between the users or actors can be represented by edges or links and each individual user can be represented by a node or vertex. The edges that we are considering are signed edges having a value of +1 , -1 or 0. From the signs of the edges it is desirable to identify the sign between two nodes where the edge is hidden,which is referred as link prediction. Our task is to predict the sign of the edge in between nodes and if there is no edge we have also taken care of that. For that purpose we have taken "Slashdot zoo" as our social network. Slashdot is a news related website. In 2002 they have introduced Slashdot zoo which allows its users to tag others as friends or foes. The friendship is represented by 1 and on the other hand for or enmity is represented by -1 and also if there is no relationship between the users then it is 0.From the above structure we have defined some of the features. In our first feature a particular node's total number of positive edges both going in and out and also total number of negative edges both going in and out are calculated. In the second feature from a particular node to all other node what is the type of the edge that exits in between them based on some criteria. In our third feature we have generated the number of triads that can be formed via a third node which is not our concerning factor. In the fourth feature we have calculated the total number of triads.We also evaluate the triads with the existing dataset. Creating quad based on the interrelationship among the nodes of the online users.

7.2 Future work

We have implemented the triads based on balance theory. We have implemented Quad using four nodes. That means from a particular node to another node if we want to predict the sign in a more effective way we have considered that first node (u) to be linked with a node (v) and again node (w) to be linked with a node (x) and thus if node (u) is linked with node (x) then we have tried to predict the sign of the edge in between the node u to node x .But in the future we will try to use more number of nodes to make a larger feature to check out the relationship among the nodes more emphasisly.Using unstructured dataset would be a key point and also minimizing the time complexity would be a keen interest for ourselves.

Bibliography

[1] ”Data mining Concepts and techniques” by Jiawei Han and MichelineKamber

[2] ”Computational social network analysis” by Ajith Abraham, Aboul-Ella Hassanien, Vaclav Snasel

[3] Teng, Chun-Yuen, Yu-Ru Lin, and Lada A. Adamic. ”Recipe recommendation using ingredient networks.” Proceedings of the 3rd Annual ACM Web Science Conference. ACM, 2012

[4] L, Linyuan, and Tao Zhou. ”Link prediction in complex networks: A survey.”Physica A: Statistical

[5] Bollen, Johan, et al. ”Happiness is assortative in online social networks.”Artificial life 17.3 (2011): 237-251

[6] Backstrom, Lars, and Jure Leskovec. ”Supervised random walks: predicting and recommending links in social networks.” Proceedings of the fourth ACM international conference on Web search and data mining. ACM, 2011

[7] Asur, Sitaram, and Bernardo A. Huberman. ”Predicting the future with social media.” Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on. Vol. 1. IEEE, 2010

[8] Liben?Nowell, David, and Jon Kleinberg. ”The link?prediction problem for social networks.” Journal of the American society for information science and technology 58.7 (2007): 1019-1031

[9] Gubichev, Andrey, et al. ”Fast and accurate estimation of shortest paths in large graphs.” Proceedings of the 19th ACM international conference on Information and knowledge management. ACM, 2010

[10] Menon, Aditya Krishna, and Charles Elkan. ”Link prediction via matrix factorization.” Machine Learning and Knowledge Discovery in Databases. Springer Berlin Heidelberg, 2011. 437-452

[11] Fortunato, Santo. ”Community detection in graphs.” Physics Reports 486.3 (2010): 75-174

[12] Leskovec, Jure, Daniel Huttenlocher, and Jon Kleinberg. ”Signed networks in social media.” Proceedings of the 28th international conference on Human factors in computing systems. ACM, 2010

[13] Leskovec, Jure, Daniel Huttenlocher, and Jon Kleinberg. ”Predicting positive and negative links in online social networks.” Proceedings of the 19th international conference on World wide web. ACM, 2010