Extraction & Visualization Of Social Relations On Social Networking Services Using Association Rule Mining

Mayuri R. Tone, M. B. Gudadhe

Abstract: Social media and Social Network Analysis (SNA) acquired a huge popularity and represent one of the most important social and computer science phenomena of recent years. One of the most studied problems in this research area is influence and information propagation. The aim of this paper is to analyze the information diffusion process and predict the influence (represented by the rate of infected nodes at the end of the diffusion process) of an initial set of nodes in two networks: Facebook users contacts users commenting these posts. These networks are dissimilar in their structure (size, type, diameter, density, components), and the type of the Relationships (explicit relationship represented by the contacts links, and implicit relationship created by commenting on post), they are extracted using Node XL tool. Three models are used for modeling the dissemination process: Linear Threshold Model (LTM), Independent Cascade Model (ICM) and an extension of this last called Weighted Cascade Model (WCM). Networks metrics and visualization were manipulated By Node Xl. Experiments results show that the structure of the network affects the diffusion process directly. Unlike results given in the blog world networks, the information can spread farther through explicit connections than through implicit relations.

Keywords: Information diffusion, influence, social media, social network analysis.

Introduction:

Human social relationships were bounded according to time and space, but the evolution of information and communication technologies tools allowed people to inexpensively and reliably share information anytime and anywhere through social media (YouTube, Flicker, Twitter, Facebook, blogs, emails, etc). These tools are Helpful resources of information, opinions and behaviors regarding different areas of interest. Studying and measuring the social media have attracted considerable interest of many researchers in various domains and led them to create a new field called Social Network Analysis (SNA). SNA methods have been applied to a wide range of areas like business, healthcare, academia, politics and terrorism. In our daily life, there are innumerable situations in which we are influenced in our decision making by what others around us are doing. Simple examples of influence Are when academic researchers choose to work on a topic that is currently ″hot″, or when we listen to the same music that our friends listen to. The marketing strategies were enhanced with a word-of-mouth approach using probabilistic models of interactions to choose the best viral marketing plan. In this paper, we focus on analyzing the information propagation process for anticipating the capability of nodes in spreading the information throughout the network. We also aim to understand how the structure of the network and the type of its relationships can influence the propagation process. Likewise, this analysis is done on two different networks: an explicit network created from Facebook user's contacts, and an implicit network created from users' comments on Facebook chat. The discussion of networks treats them as static structures: we take a snapshot of the nodes and edges at a particular moment in time and then analyze their structure and the diffusion of information process. These networks were extracted using +Node XL tool.

- *Mayuri R. Tone, Final Year M. Tech. Computer Science & Engineering,*
- *M. B. Gudadhe, Assistant Professor*

The primary communication method of social networking services is posts. By writing posts and comments, people share their thoughts, opinions, and real time status. People tend to write messages and replies to people they have close friendships with or are arguing with. Also people write lots of comments one after another when they are having an in-depth conversation. Based on this tendency, we made a model on the latent social relationship among users by examining and analyzing the message threads. In this paper, we propose a method to extract the latent social relationship from a social networking service by analyzing the users' activities. The users' writing patterns are especially considered to examine the intensity of the conversation and the strength of the social relation. We applied our algorithms to a Facebook dataset and developed an evaluation system to appraise the proposed algorithms. The experimental result shows that the proposed method using a weighted harmonic rule with a root-included sliding window fits best for social relation extraction.

Literature Survey

The dramatic increase of popularity of social networks has attracted a lot of research. Factors like social interaction, knowledge exchange, knowledge discovery, ability to capture data about various types of social interactions at a very fine granularity with practically no reporting bias, and availability of data mining techniques for building descriptive and predictive models of social interactions have become key drivers for computer science research in SNA. Discovering knowledge from these networks is a challenging and primary research issue because of their size, reach ability and diversity. Several research studies have been conducted on social network analysis and two main approaches have been studied: Extraction of social relation on social networking services can be analysed based on users' activities in social services. Several method to extract social relation on social networking services such as link mining, crawling process, extracting key phrases, collective context, frequent set mining algorithm. The different sampling algorithms that have been implemented to search or examine the social network graph Facebook that consist of countless friend-friend relationships. Out of the two sampling techniques, the visiting technique, BFS

is known to deliver biasness in the scenario of incomplete traversal [1]. The ontology-aware classification approach is used for interest and link prediction in social network. Analysis of friendship networks has identified ways in which graph features can be used for prediction of link existence and persistence [2]. Clustering similar entity pairs according to their collective context in web document is to discover a relevant key phrase that relates entities. Cluster & label selection process is used to extract social relation [3]. To extract labels that are useful for describing relations in social networks is to analyse surrounding local context in which entities of interest co-occur on web & to seek clues to describe relation [4]. Although use of all these techniques focus on the links among the user such as citation, trackback, and comment. In this project, to solve this kind of problem, a new method is used i.e. association rule mining to extract social relation. In this method, the one-way relation from a comment writer to a post writer is assumed. However, often users make arguments or comments within a comment thread and the main post writer may not be included in the context of the current comment thread. In other words, there are complex relationships between the comment writers and the post writer or among the comment writers, which are not limited to a one-way connection. In order to alleviate this problem, the focus is on the latent relationship within the message thread. The primary communication method of social networking services is posts. By writing posts and comments, people share their thoughts, opinions, and real time status. People tend to write messages and replies to people they have close friendships with or are arguing with [5]. Based on this tendency, we made a model on the latent social relationship among users by examining and analyzing the message threads. In this project, several methods are used to extract the latent social relationship from a social networking service by analyzing the users' activities. The users' writing patterns are especially considered to examine the intensity of the conversation and the strength of the social relation [5]. Discovering knowledge from these networks is a challenging and primary research issue because of their size, reach ability and diversity. Several research studies have been conducted on social network analysis and two main approaches have been studied: one is influential user discovery and the other is social network construction. The aim of the first approach is finding most influential users in communities by analyzing their relationships and activities and the second approach concentrates on discovering social network of users. This section presents the work conducted so far in the latter approach. Data from different sources like Web, e-mail communication logs, instant messenger logs, blogs, etc. has been used either individually or in combination for the purpose of social network extraction.

Sr. no.	Record name	Author name	Research work	Year
1.	International Journal of Advanced Research in Artificial Intelligence Vol. 4	Dr. Mamta Madan, Meenu Chopra	The different sampling algorithms that have been implemented to search or examine the social network graph Facebook that consist of countless friend-friend relationships. Out of the two sampling techniques, the visiting technique, BFS is known to deliver biasness in the scenario of incomplete traversal	Nov 4, 2015.
2.	AAAI	Waleed Aljandal, Vikas Bahirwani, Doina Caragea, William H. Hsu	The ontology-aware classification approach is used for interest and link prediction in social network. Analysis of friendship networks has identified ways in which graph features can be used for prediction of link existence and persistence	2009
3.	IJCAI	Junichiro Mori, Mitsuru Ishizuka, Yutaka Matsuo	Clustering similar entity pairs according to their collective context in web document is to discover a relevant key phrase that relates entities. Cluster & label selection process is used to extract social relation	2007
4.		Junichiro Mori, Takumi Tsujishita, Yutaka Matsuo, and Mitsuru Ishizuka	To extract labels that are useful for describing relations in social networks is to analyse surrounding local context in which entities of interest co-occur on web & to seek clues to describe relation	

5.	The Twenty-Fourth AAAI Conference on Artificial Intelligence	Meesun Song, Wonkyu Lee, Junghwan Kim	Several methods are used to extract the latent social relationship from a social networking service by analyzing the users' activities. The users' writing patterns are especially considered to examine the intensity of the conversation and the strength of the social relation	2010
6.	International Journal of Computing and Business Research (IJCBR) ISSN	Tamanna Jain	Physical analysis of these huge amount of information stored in modern databases Is very difficult. Data mining provides tools to reveal unknown information in large databases Which are stored already. A well-known data mining technique is association rule mining in which it Provides an effective algorithm is apriori algorithm. The large quantity of information collected through the set of association rules using algorithm can be used not only for illustrating the relationships in the database, but also used for differentiating between different kinds of classes in a database.	Jan 2013
7.	ICWSM	Agrawal, R. and Srikant	Finding all such rules is valuable for cross-marketing and attached mailing applications. Other applications include catalogue design, add-on sales, store layout, and customer segmentation based on buying patterns. The databases involved in these applications are very large. It is imperative, therefore, to have fast algorithms for this task.	1994
8.	ICWSM	Furukawa, T., Matsue, Y., and Ohmukai	The user log data are used to identify readership relations among bloggers. After analysis of more than 50,000 users for about two years, we reveal some interactions between social relations and readership relations. We first show that bloggers read other weblogs on a regular basis	2007

PROPOSED WORK

1.1 Overview/Introduction

The publication of social network data entails a privacy threat for their users. Sensitive information about users of the social networks should be protected. The challenge is to devise methods to publish social network data in a form that affords utility without compromising privacy. Previous research has proposed various privacy models with the corresponding protection mechanisms that prevent both inadvertent private information leakage and attacks by malicious adversaries. These early privacy models are mostly concerned with identity and link disclosure. The social networks are modeled as graphs in which users are nodes and social connections are edges. The threat definitions and protection mechanisms leverage structural properties of the graph. This paper is motivated by the recognition of the need for a finer grain and more personalized privacy.

Algorithm Description

- Input: A social network SN, with number of user as node n and user features like images posts etc.
- Output: A social Network SN with user accessing feature of other user.

1) Create number of number of nodes (user) n = 1, N having personal information.
2) Each node (user) will have facility add other nodes (user) as a friend by send friend request.
3) Each node (user) will have facility to accept other node (user) friend request and connect to it as a friend.
4) Each node (user) can have n = 1, N numbers of friends.
5) In normal condition every other node (user) can write post on other user (friends) wall, read friends post, share friends post, share image.
6) The algorithm starts out with group formation, during which all nodes that have not yet been grouped are taken into consideration, in clustering-like node.
7) Here each user (node) will form two group of its connecting friends, one close friend and other friends, this will depend upon user (node) to whom to select as close friend or friend which will be based on maximum similarity both user (nodes) have.
9) When a person post more than three messages and other person send replies to all messages then they will be close friends.

1.2 Working of Algorithm

1) The project is divided into two part ,first is Web Pages were JSP and html pages are kept which are user interface of this project, and second is Source Package were Java code is kept.
2) The first page that we see when we run the project is LoginForm.jsp, here in html page is design and it contains username and password field to check username and password are correct or not, when button is click, it goes to java class “LoginCheck”. This class is in Source Package folder inside org.Servlet.
3) In this class we get data from username and password filed, then get database connection from DBConnectionClass class and then check that username and password exists in database or not.
4) Then we again go to LoginFrom.jsp which is in Web Pages folder, were if new user is there, we go to registration in SignUp.jsp.
5) In Signup.jsp registration page is deigns using html and when button is click, we go to “SignUpServlet” .This class is in Source Package folder inside org.Servlet.
6) In this class we take data from Signup.jsp like username, email, password, date of birth, photo etc., then get database connection from DBConnectionClass class and then store whole record in table “sign_up_tbl” .At the same time here we create node for each user by creating table for each user and giving them name as “n1”,”n2”.(This is our first module i.e. Data Collection)

Show All or Limit 0 50 Refresh

	id	fname	last_name	email	password	birthday	sex	photo
	1	sachin	deshmukh	sachin@gmail.com	test			Chrysanthemum.jpg
	2	amit	deshmukh	amit@gmail.com	test			Desert.jpg
	3	kanchan	deshmukh	kanchan@gmail.com	test			Hydrangeas.jpg
	4	somil	deshmukh	somil@gmail.com	test			Jellyfish.jpg
*	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

7) If we successfully get login, then we MainPage.jsp which is Web Pages folder, in this page is design in html, to show login user page and link to add friends, send request of friends, set attribute, add images etc.

8) In MainPage.jsp when we click “Friends” link we go to “FrndPage.jsp”.In FrndPage.jsp we can see all the friends, In this page we call GetFrndList.class, This class is in Source Package inside org.DBPacge .IN this class we get database connection from DBConnectionClass class and then get all friends list of specific user from sign_up_tbl table .and then List of friends are shown in “FrndPage.jsp”.
9) In MainPage.jsp when we click “Friend Request” link we go to “AcceptFrnd.jsp”.In AcceptFrnd.jsp we can see all the user which are not friends, In this page we call MutualFriend.class, This class is in Source Package inside org.DBPacge .IN this class we get database connection from DBConnectionClass class and then get all non friends list of specific user from sign_up_tbl table .and then List of non friends are shown in “AcceptFrnd.jsp“.
10) In AcceptFrnd.jsp, when we click “Accept Request” button, it go to “AcceptFrnd”, this class is in Source Package folder inside org.Servlet.
In this class we get data i.e. from were request has come and to whom friends has to be added, then get database connection from DBConnection class, and then update specific node table with frndReq values i.e. If n1 is table, and n2 is his friend ,then in n1 table there will be row with frndReq column as 2. . (This is Reduce Node Degree Module)
11) In MainPage.jsp when we click “Send Request” link we go to “showUsers.jsp”. In showUsers.jsp.IN this page we get database connection from DBConnectionClass class and then get all friends list of specific user from sign_up_tbl table .and then show list of all user in “showUsers.jsp“.
12) IN showUsers.jsp when we click “Send Request “ button ,we go to “sendRequst.jsp”,IN sendRequst.jsp we get data i.e. from which node(user) request has come, and to which node (user) request has to receive and then insert row in specific node table for friend request.
I.e. if request is send from node n1, to n2, then in n1 table one row will be added fro friend request to n2. . (This is Reduce Node Degree Module)
13) In MainPage.jsp when we click “Set Post Attributes” link we go to “goupAutherity.jsp”.In goupAutherity.jsp.IN this page we first show all the friends of user who have accepted request from above code. That information is there in node specific table i.en1, n2.

Then we have add name of close friend in text area ,and when we click submit button ,we go to CloseFrndGrp.jsp, in this jsp we get we get database connection from DBConnectionClass class and then get names of users enter in text area as close friend and the update specific node table with close friend relation . In n1 is node table ,and it has enter n4 as close friend ,then in n1 table against n4 columns “ftype “ will be updated as close friend.

	flist	frndReq	ftype
	4	(NULL)	closefreind
*	(NULL)	(NULL)	no_Type

(This is Add Node Degree module)

14) Then we have add name of normal friend in below text area ,and when we click submit button ,we go to FriendGroup.jsp, in this jsp we get we get database connection from DBConnectionClass class and then get

names of users enter in text area as friend and the update specific node table with close friend relation . In 4 is node table ,and it has enter n2 as friend ,then in n4 table against n2 columns “ftype “ will be updated as friend.

	flist	frndReq	ftype
	3	(NULL)	closefreind
	1	(NULL)	closefreind
	2	(NULL)	freind
*	(NULL)	(NULL)	no_Type

(This is Add Node Degree module)

15) Then in next portion we provide authority to group .i.e. whether close friend group and friend group should see post, send post and see image of specific user. When you click check box and click submit button ,we go to setShowAttribute.jsp,in this jsp we get database connection and then get specific assigned authority and then update sign_up_tbl with authority as “yes” or “no”

postAuthFrnd	postAuthClose	showAuthFrnd	showAuthClose	image_auth_frnd	image_auth_clos
no	no	no	no	no	no
no	no	no	no	no	no
yes	no	no	yes	no	yes
no	yes	no	yes	no	yes
no	no	no	no	no	no

Here in FrienGroup.jsp and groupAutjority.jsp our GINN algorithm code is written i.e. The algorithm starts out with group formation, during which all nodes that have not yet been grouped are taken into consideration, in clustering-like node. In the first run, two nodes with the maximum similarity of their neighborhood labels are close friend. Their neighbor labels are modified to be the same immediately so that nodes in one group always have the same neighbor labels. Larger value indicates larger similarity of the two close friends. Then nodes having the maximum similarity with any node in the group are clustered into the group till the group has ` nodes with different sensitive labels. Thereafter, the algorithm proceeds to create the friend group. If fewer than ` nodes are left after the last group’s formation, these remainder nodes are clustered into friend groups according to the similarities between nodes and groups. (This is Add Node Degree module)

15) In Main Page .jsp when we click “Post on frnd wall” link, we go to “FreindsPage.jsp”. In FreindsPage.jsp, we first get list of all the friends to specific node table, and then from sign_up_tbl table check whether that specific node have post authority, if it has authority ,then it can post data on that user(node) . (This is Add Noise Node module)

15) In Main Page .jsp when we click “share image” link, we go to “shareImages.jsp”. In shareImages.jsp, we first browser to get image path and then go to Share Image class which is inside Source Package folder inside org.ServletPacage, in this class we get image path from ServletFileUpload class and the get database connection from DBConnectionClass(),and then insert row in image_tbl, with image name and who has posted image.

	image	poster
	Penguins.jpg	somil@gmail.com
	Lighthouse.jpg	somil@gmail.com
*	(NULL)	(NULL)

(This is Add Noise Node module)

16) In Main Page .jsp when we click “Show Share Image” link, we go to “ShowImages.jsp”. In ShowImages.jsp, in this jsp we get database connection and then from image_tbl table get image name and who has posted that image and, the show that image on page. (This is Add Noise Node module)

17) In Main Page .jsp when we click “Show Friend Share Image” link, we go to “ViewFrndImages.jsp”. In ViewFrndImages.jsp, we first get list of all the friends to specific node table, and then from sign_up_tbl table check whether that specific node have show image authority, if it has authority ,then it can see image of that user(node) . (This is Add Noise Node module)

18) To see node graphs we have create class in Source Package folder ,inside NodeCreationpackage,the class name is CraeteNode.java,in this class we get database connection from DBConnectionClass.java, then from sign_up_tbl table ,get list of all user and then from node table get friends and close friend list. We use org.neo4j.graphdb.Node class we create nodes in graphs form, from org.neo4j.graphdb.Relationship class we create relationship between each node, from org.neo4j.graphdb.RelationshipType class we create which type of relation is having i.e. close or normal friend and then from org.neo4j.graphdb.factory.GraphDatabaseFactory class we show that in graph format in neo4j URL.

METHODOLOGY AND IMPLEMENTATION

2.1 Introduction

The social networks are modeled as graphs in which users are nodes and features (images, post) are labels. Labels are denoted either as sensitive or as non-sensitive. We treat node labels both as background knowledge an adversary may possess, and as sensitive information that has to be protected. We present extraction of social relations. To this aim, the algorithms transform the original graph into a graph in which nodes are sufficiently indistinguishable. The algorithms are designed to do so while losing as little information and while preserving as much utility as possible. We evaluate empirically the extent to which the algorithms preserve the original graph's structure and properties.

2.2 Block diagram

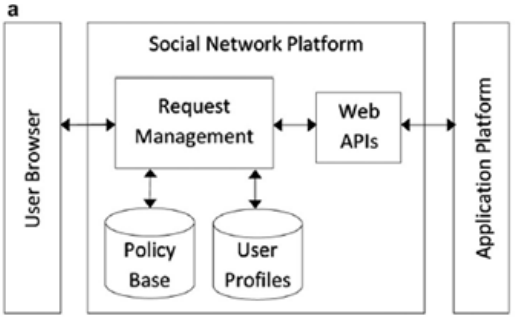

Fig.1 *Social Networking Block Diagram*

2.3 Implementation

Applications require permission to access user's profile data to provide a service customized to the user's profile data. In this section we present our approach to enable fine grain access control for third party applications, to limit applications' access only to relevant user's profile data. We first provide some preliminary definitions related to applications and API set, and then we discuss our proposed fine grain access control framework for API based applications. Our social network web sites will release APIs (Application Programming) that allow developers to leverage and aggregate information stored in user profiles and provide extended social network services. The exposed APIs are basically a set of web services that provide a limited and controlled view for the application to interface with the social network site. The social network application architecture includes three interacting parties namely the user, social network server, and the third party application server. Fig. 1, shows the different blocks used in the social networks architecture. Note that the application server is able to connect to social network through the exported web APIs. Furthermore, these requests are filtered through the request management module. Social networks provide mechanisms for users to customize their profiles and to add applications developed by external developers. The application provides the customized services by accessing the exported APIs. Fig. 2(b) depicts the interaction stages between the user browser, social network and the third party application. The interaction starts when a user requests an application APP (Steps 1e2). The application server interacts with the social network server by instantiating API calls (Step 3). Upon receiving the responses of the API calls, the application server compiles and sends a response to the social network which is forwarded to the requesting user (Steps 4e5).

Experimental Results

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement. For our project we have taken Facebook data from various sites like http://snap.stanford.edu/data/egonets-Facebook.html, http://konect.uni-koblenz.de/networks/facebook-wosn-links http://socialnetworks.mpi-sws.org/data-wosn2009.html. We have downloaded the data and added that to the excel sheet graph, to provide comparison between various techniques available for privacy protection in social networking based on Scalability, Time and Security.

Conclusion

In this paper we have investigated social relations.We assume that user will divides its nodes into two group close friend and friend such that a model for attaining privacy while publishing the data, in which node labels are both part of adversaries' background knowledge and sensitive information that has to be protected. We accompany our model with algorithms that transform a node network graph before publication, so as to limit adversaries' confidence about sensitive label data. Our experiments on both real and synthetic data sets confirm the effectiveness, efficiency and scalability of our approach in maintaining critical graph properties while providing a comprehensible privacy guarantee.

REFERENCES:

[1] Dr. Mamta Madan, Meenu Chopra, "Using Mining Predict Relationships on the Social Media Network: Facebook (FB)" in (IJARAI) International Journal of Advanced Research in Artificial Intelligence, Vol. 4, Nov 4, 2015.

[2] Waleed Aljandal, Vikas Bahirwani, Doina Caragea, William H. Hsu, "Ontology-Aware Classification and Association Rule Mining for Interest and Link Prediction in Social Networks".

[3] Junichiro Mori, Mitsuru Ishizuka, Yutaka Matsuo, "Extracting Keyphrases to Represent Relations in Social Networks from Web".

[4] Junichiro Mori, Takumi Tsujishita, Yutaka Matsuo, and Mitsuru Ishizuka, "Extracting Relations in Social Networks from the Web using Similarity between Collective Contexts".

[5] Meesun Song, Wonkyu Lee, Junghwan Kim, "Extraction and Visualization of Implicit Social Relations on Social Networking Services" in Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10).

[6] Tamanna Jain, "An Efficacious Analysis On Usage And Rule Mining Associated With Social Network Portal", in International Journal of Computing and Business Research (IJCBR) ISSN (Online) : 2229-6166 Volume 4 Issue 1 January 2013.

[7] Agrawal, R. and Srikant, R. 1994. Fast Algorithms for Mining Association Rules. In Proceedings of the 20th [8] Furukawa, T., Matsue, Y., and Ohmukai, I. 2007. Social Networks and Reading Behavior in the Blogosphere. In Proceedings of ICWSM'07, 51-58.