Social Influence Analysis

Enroll. No.	-14103133 , 14103188 , 14103209
Name of Student	-Shantam , Avneesh, Prakhar
Name of supervisor(s)	-Sakshi Aggarwal

December-2017

Submitted in partial fulfillment of the Degree of

Bachelor of Technology in

Computer Science Engineering

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING & INFORMATION TECHNOLOGY

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

(I)

TABLE OF CONTENTS

DECLARATION

I/We hereby declare that this submission is my/our own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Place: Signature:

Date: Name:

Enrollment No

CERTIFICATE

This is to certify that the work titled "**Social Influence Analysis**" submitted by "**Shantam Wadhwa, Avneesh Singhal & Prakhar Srivastava**." of **B.tech** of Jaypee Institute of Information Technology University, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of any other degree or diploma.

Signature of Supervisor ....

Name of Supervisor ....

Designation ....

Date ....

ACKNOWLEDGEMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

We are highly indebted to **Sakshi Aggarwal** for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We would like to express my gratitude towards my parents & panel members of for their kind co-operation and encouragement which help me in completion of this project.

Our thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

Signature of the Student …………………….

Name of Student …………………….

Enrollment Number …………………….

Date …………………….

SUMMARY

In large social networks, users are influenced by others for various reasons. For example, the colleagues have strong influence on one's work, while the friends have strong influence on one's daily life.

Our approach is to differentiate the social influences from different angles (topics) or small-scale networks and to measure the strength of those social influences. In this work, we focus on coming up with an approach for measuring the strength of social influence with a generalized outlook towards the existing social networks.

The existing methods mainly focus on qualitative identification of the existence of influence, but do not provide a quantitative measure of the influential strength. Much effort has been made for social network analysis and a large number of work has been done. For example, methods are proposed for identifying cohesive subgraphs within a network where cohesive subgraphs are defined as "subsets of actors among whom there are relatively strong, direct, intense, frequent, or positive ties" . Quite a few metrics have been defined to characterize a social network, such as betweenness, closeness, centrality, centralization, etc.

The approach used in the project focuses on Topic-based learning which refers to prediction of expertise on basis of the follower network of a person in context to the topic being discussed. Two approaches are used in this paper, Basic TAP, and Distributed TAP. Distributed TAP is based on basic TAP only, but is used for parallel processing of large datasets using multiple systems.

LIST OF FIGURES

(VIII)

LIST OF SYMBOLS & ACRONYMS

SYMBOL	DESCRIPTION
N	number of nodes in the social network
M	number of edges in the social network
T	number of topics
V	the set of nodes in the social network
E	the set of edges
v_i	a single node
y_i^z	node-v_i's representative on topic z
$\mathbf{y}_i$	the hidden vector of representatives for all topics on node v_i
θ_i^z	the probability for topic z to be generated by the node v_i
e_{st}	an edge connecting node v_s and node v_t
w_{st}^z	the similarity weight of the edge e_{st} w.r.t. topic z
μ_{st}^z	the social influence of node v_s on node v_t w.r.t. topic z

Chapter 1 : INTRODUCTION

1.1 General Introduction

- Social Media's presence is ubiquitous today. Its extent can be realized with Facebook's 1.55 billion monthly active users. As Social Media grew from its embryonic state in 2004, it gradually transformed into a communication platform on which governance, news, protests and friendships originated and sustained. Around the world, social media consistently plays a major role from managing natural disasters to winning election campaigns.
- In large social networks, users are influenced by others for various reasons. For example, the colleagues have strong influence on one's work, while the friends have strong influence on one's daily life. Our approach is to differentiate the social influences from different angles (topics) or small-scale networks and to measure the strength of those social influences. In this work, we focus on coming up with an approach for measuring the strength of social influence with a generalized outlook towards the existing social networks.

1.2 Relevant current/open problems

- A central problem for social influence is to understand the interplay between similarity and social ties. The main problems faced are in identifying of practice through conformity, compliance and obedience, social influence in virtual worlds.
    - There exist several challenging problems in terms of differentiating the social influences from different angles (topics). The key questions arising in this context are
        - (a) How to quantify the strength of those social influences?
        - (b) How to construct a model and estimate the model parameters for real large networks?
    - The problem of learning the influence degree from historical user actions.
    - To study the problem of friendships drifting over time, and their effect on mutual social influence.

1.3 Integrated Summary of Literature Reviewed

Integrated Summary of Referred Papers

Paper I

- Through the paper, the author discusses in intricate detail the meaning of Social Influence in the modern community, and its effects on the social life of the people connected. The author also sets a base for Social Network related approach in the form of different parameters of a graph, for ex. clustering coefficient, in-out degree of nodes etc. and how the measure of these parameters can be used to interpret the behavior of nodes(or users) when put in specific scenarios, for ex. how likely it is that a node befriends or follows someone who is not related to the node's direct circles.
- It also differentiates how change of focus from quantity(no. of nodes related) to quality(closeness of the related nodes), could bring unprecedented changes in a user's decisions for what he does on the internet.

Paper II

- The existing methods mainly focus on qualitative identification of the existence of influence, but do not provide a quantitative measure of the influential strength.
- The approach used in the paper focuses on Topic-based learning which refers to prediction of expertise on basis of the follower network of a person in context to the topic being discussed. Two approaches are used in this paper, Basic TAP, and Distributed TAP. Distributed TAP is based on basic TAP only, but is used for parallel processing of large datasets using multiple systems.

Paper III

- The ways to measure the strength of social ties people have cannot yet be broken into solid parameters. This paper proposes a model, focusing on variables which can be used to check the strength of mutual bonds among people, for example broader classification matrix, and autocorrelation improvement. Datasets Examined are undirected (FB and LinkedIn) to redeem maximum possible connections.

Paper IV(7)

- In this paper, there is an analysis of social network in order to understand the interplay between similarity and social ties. People are similar to their neighbors in a social network for two distinct reasons : Firstly, they grow to resemble their current friends due to social influence. Secondly, they tend to form new links to others who are already like them, this is selection by sociologists. Social Influence can push systems towards uniformity of behavior while Selection can lead to fragmentation. So, they are developing techniques for interaction between Social Influence and Selection using data from online communities where both Social Interaction and changes in behavior over time can be measured.

- According to the authors , it is important to isolate the respective effect of Social Influence and Selection as, Social Influence produce network-wide uniformity i.e new behavior spreads along the links. Example- Viral marketing. Whereas Selection tends to drive the network towards smaller clusters of like-minded people. Example-Recommender systems. Developed a framework using data from Wikipedia. Basically, they focused on two central questions: First, Can we characterize how Social Interaction affects interests and vice-versa? Second, Can we characterize the relative degree to which interests and Social Interactions affect what people do?
- Wikipedia focuses on two types of recorded actions i.e. editing articles and editing the discussion page associated with particular user. Editing articles are the indicators of interest as each article is on a particular topic whereas Editing the discussion page is the indicator of social ties between the users who are communicating. Considering the pairs of Wikipedia editors who have communicated with one another- after aggregating over all such pairs , they founded that there is a sharp increase in the similarities between two editors before they interact. Therefore, similarity leads to interaction but not vice-versa. Then they compared their model with Holme-Newman Model which is too simple to produce the effects we see in Wikipedia. In order to measure how Selection and Social influence operate , they tracked a time-evolving vector representing each person's activities and they also studied how the vector of two people are changing directly around the moment when they first interact . Mathematical expressions used: cosine metric and Jaccard similarity.

Paper V(8)

- There has been a long standing goal of social analysts to characterize the relationship that exists between person's social group and his/her personal behavior. In this paper, the author applied various data mining techniques to study this relation of over 10 million people, by turning to online sources of data. The people who chat with each other are more likely to share interests according to the analysis.
- For the characteristics of each person in a network, they turned to two sources: First, is the demographic data of instant messaging(IM) users such as person's age, gender

and geographic location. Second is based on personal behavior, for this they turned to web search behavior. For obtaining the relationship between users, conditional probability is used. Two datasets have been used in this paper- MSN messenger instant messaging network and keyword searches made by various users on Microsoft Web search engine along with the information about user's personal characteristics such as zip codes, age and gender. Then there are single-valued attributes and multi-valued attributes. Then this relationship is shown by various histograms. Then the conditioning is done on personal attributes. The results shows that people who talk to each other on the messenger network are more likely to be similar than a random pair of users, where similarity is measured in terms of matching on attributes such as queries issued, query categories, age, zip and gender.

Paper VI(9)

- Conformity is a type of social influence involving a change in opinion or behavior in order to fit within a group. This paper is about how the effect of conformity plays a role in changing user's online behavior. Then a Confluence model is proposed in this paper to formalize the effects of social conformity into a probabilistic model.
- Confluence can distinguish and quantify the effects of the different types of conformities. There are certain experimental results on four different types of large social networks, i.e., Flicker, Gowalla, Weibo and Co-Author, verify the existence of the conformity phenomena. Action prediction accuracy (AUC) of different methods by considering the effect of conformity in four networks are also given through histograms. The goal is to study how a user's behavior conforms to her peer friends and the communities (groups) that she belongs to.
- Three levels of conformities respectively from the aspect of individual, peer relationship, and group are also defined. Then Conformity-aware Factor Graph Model is explained in detail. They simply considered four attribute features, i.e., the number of friends, the number of new friends in the recent three time stamps, the number of total groups that the user joined, and the number of groups the user joined in recent three time stamps. Then there is an algorithm about Distributed Model Learning in order to handle large networks. On all the four data sets, they used the historic users' actions as the training data in different methods and used the learned model to predict users' action in the next time stamp- Prediction performance analysis.

1.4 Problem Statement

To develop Generic set of features /method to quantify social influence in various applications and comparative study of its existing approaches.

1.5 Overview of proposed solution approach and Novelty/benefits

To address these fundamental questions, we propose Topical Affinity Propagation (TAP) to model the topic-level social influence on large networks. In particular, TAP can take results of any topic modeling and the existing network structure to perform topic-level influence propagation. With the help of the influence analysis, we present several important applications on real data sets such as 1) what are the representative nodes on a given topic? 2) how to identify the social influences of neighboring nodes on a particular node? To scale to real large networks, TAP is designed with efficient distributed learning algorithms that is implemented and tested under the Map-Reduce framework.

We further present the common characteristics of distributed learning algorithms for Map-Reduce. Finally, we demonstrate the effectiveness and efficiency of TAP on real large data sets.

Chapter 2: Analysis, Design and Modeling

2.1 Requirements Specifications

Social network analysis often focus on macro-level models such as degree distributions, diameter, clustering coefficient, communities, small world effect, preferential attachment, etc we focus on measuring the strength of topic-level social influence quantitatively. With the proposed social influence analysis, many important questions can be answered such as 1) what are the representative nodes on a given topic? 2) how to identify topic-level experts and their social influence to a particular node? 3) how to quickly connect to a particular node through strong social ties?

2.2 Functional and Non Functional requirements

Topic distribution: In social networks, a user usually has interests on multiple topics. Formally, each node v 2 V is associated with a vector µv 2 RT of T-dimensional topic distribution (P z µvz = 1). Each element µvz is the probability(importance) of

the node on topic z.

Topic-based social influences: Social influence from node s to t denoted as ¹st is a numerical weight associated with the edge . In most cases, the social influence score is asymmetric. Furthermore, the social influence from node s to t will vary on different topics.

Thus based on the above concepts, we can define the tasks of topic-based social influence analysis. Given a social network G = (V;E) and a topic distribution for each node, the goal is to find the topic-level influence scores on each edge.

2.3 Design Documentation

2.3.1 Class diagrams

1) Class Diagram of Node Feature :

: **Node Feature**
+ a single node (Vi)
+node representation on topic y_i^z
+ topic z

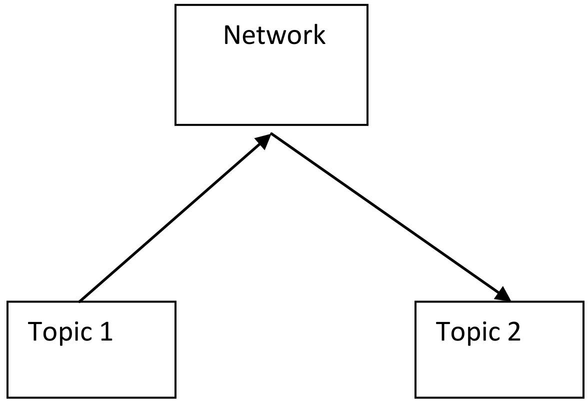
2) Class Diagram of Edge Feature :

: **Edge Feature**
+ node representation on topic y_i^z
+ the hidden vector of representatives for all topics on node y_i
+ topic z

2.3.2 Activity diagrams

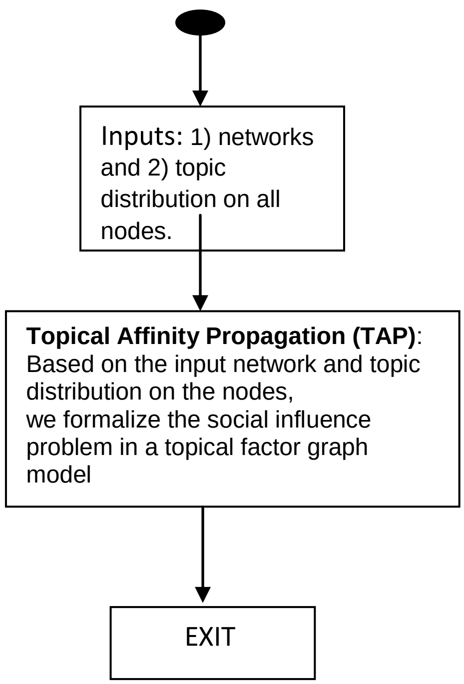
1. Activity Diagram for Topic Modeling through network analysis.

Fig-1



2. Activity Diagram for input of data & the algorithm formalization using TAP algorithm.

Fig-2



.

2.3.3 Algorithms

The goal of topic-based social influence analysis is to capture the following information: nodes' topic distributions, similarity between nodes, and network structure. In addition, the approach has to be able to scale up to a large scale network. Following this thread, we first propose a Topical Factor Graph (TFG) model to incorporate all the information into a unified probabilistic model. Second, we propose Topical Affinity Propagation (TAP) for model learning. Third, we discuss how to do distributed learning in the Map-Reduce framework. Finally, we illustrate several applications based on the results of social influence analysis.

New TAP Learning Algorithm

Fig-3

Input: $G = (V, E)$ and topic distributions $\{\theta_v\}_{v \in V}$
Output: topic-level social influence graphs $\{G_z = (V_z, E_z)\}_{z=1}^{T}$

Calculate the node feature function $g(v_i, y_i, z)$;
Calculate b_{ij}^z according to Eq. 8;
Initialize all $\{r_{ij}^z\} \leftarrow 0$;
repeat
    foreach *edge-topic pair* (e_{ij}, z) **do**
        Update r_{ij}^z according to Eq. 5;
    end
    foreach *node-topic pair* (v_j, z) **do**
        Update a_{jj}^z according to Eq. 6;
    end
    foreach *edge-topic pair* (e_{ij}, z) **do**
        Update a_{ij}^z according to Eq. 7;
    end
until *convergence*;
foreach *node* v_t **do**
    foreach *neighboring node* $s \in NB(t) \cup \{t\}$ **do**
        Compute μ_{st}^z according to Eq. 9;
    end
end
Generate $G_z = (V_z, E_z)$ for every topic z according to $\{\mu_{st}^z\}$;

Fig-4

In this work, we define the node feature function g as:

$$g(v_i, y_i, z) = \begin{cases} \dfrac{w_{iy_i^z}^z}{\sum_{j \in NB(i)} (w_{ij}^z + w_{ji}^z)} & y_i^z \neq i \\ \dfrac{\sum_{j \in NB(i)} w_{ji}^z}{\sum_{j \in NB(i)} (w_{ij}^z + w_{ji}^z)} & y_i^z = i \end{cases} \qquad (1)$$

Chapter-3 Implementation and Testing

3.1 Implementation details and issues

As a social network may contain millions of users and hundreds of millions of social ties between users, it is impractical to learn a TFG from such a huge data using a single machine. To address this challenge, we deploy the learning task on a distributed system under the map-reduce programming model.

Map-Reduce is a programming model for distributed processing of large data sets. In the map stage, each machine (called a process node) receives a subset of data as input and produces a set of intermediate key/value pairs. In the reduce stage, each process node merges all intermediate values associated with the same intermediate key and outputs the final computation results. Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key.

Code Snippets:

```
def swap(u,v):
                    """
                    exchange the values of u and v
                    """
                    return v,u
```

```
def _calculate_g(self):
                                  """eq. 1"""
                           for i in self.G.nodes():
                           n = self.G.neighbors(i)
                           self.g[i] = np.zeros((len(n)+1, self.T))
                           sumin = np.zeros((self.T))
                           sumout = np.zeros((self.T))

                          for t, attr in self.G[i].iteritems():
                          this = int(t) - 1
                          for k in xrange(self.T):
                           w = float(attr['weight'])
                           sumout[k] = sumout[k] + w * self.theta[this,k]
                           for t, attr in self.G[i].iteritems():
                            for k in xrange(self.T):
                          w = float(attr['weight'])
                          this = int(i) - 1
                          sumin[k] = sumin[k] + w * self.theta[this,k]
                          self.g[i][len(n),k] = sumin[k] / (sumin[k] +
                          sumout[k])
                          j = 0
                          for t,attr in self.G[i].iteritems():
                          for k in xrange(self.T):
                          w = float(attr['weight'])
                          this = int(t) - 1
                          self.g[i][j,k] = w * self.theta[this,k] / (sumin[k]
                          + sumout[k])
                          j+=1




                          def _calculate_b(self):
                          """eq. 8"""
                          for i in self.G.nodes():
```

```
n = self.G.neighbors(i)
self.b[i] = np.zeros((len(n)+1, self.T))
self.r[i] = np.zeros((len(n)+1, self.T))
self.a[i] = np.zeros((len(n)+1, self.T))
sum_ = np.zeros((self.T))
for j in xrange(len(n)+1):   # +1 to include self.
for k in xrange(self.T):
sum_[k] += self.g[i][j,k]
for j in xrange(len(n)+1):
for k in xrange(self.T):
self.b[i][j,k] = np.log(self.g[i][j,k] / sum_[k])


def _calculate_mu(self):
self.MU = {}
# Export
for k in xrange(self.T):
subg = nx.DiGraph()
# Influence
for i in self.G.nodes():
n = self.G.neighbors(i)
for j in self.G.nodes():
if j in n:
j_ = n.index(j)
i_ = self.G.neighbors(j).index(i)
# Equation 9.
j_i = 1./ (1. + np.exp(-1. * (self.r[i][j_,k] +
self.a[i][j_,k])))
i_j = 1./ (1. + np.exp(-1. * (self.r[j][i_,k] +
self.a[j][i_,k])))
if j_i > i_j:   # Add only strongest edge.
subg.add_edge(j, i, weight=float(j_i))
else:
subg.add_edge(i, j, weight=float(i_j))
```

```
                        # Theta
                        for i in self.G.nodes():
                        G.node[i]['theta'] = self.theta[i, k]
                        self.MU[k] = subg


 def graph(self, k):
                     return self.MU[k]
                     def build(self):
                     nc = 0
                     self.iteration = 0.
                     cont = True
                     while cont:
                     self.iteration += 1
                     self._update_r()
                     self._update_a()
                     nc,cont = self._check_convergence(nc)
                     self._calculate_mu()
                     self.write('./output/')
                     edgepath = './sample/graph-16.edge'
                     distpath = './sample/distribution.txt'
                     G = nx.Graph()

theta = np.random.rand(234,10)
                           print 'first model'
                           model = TAPModel(G, theta)
                           model.build()
                           alt_r, alt_a, alt_G = model.r, model.a,
                           model.G
                           print 'second model'
                           model2 = TAPModel(G, theta)
                           model2.prime(alt_r, alt_a, alt_G)
                           model2.build()
```

3.2 Testing of the implemented modules

Introduction: - In a Software development project, errors can be injected at any stage during development. In Testing phase, the program to be tested is executed with a set of test cases and the output of the program for the test cases is evaluated to determine if the program is performing as expected. Testing is the process of executing a program with the intent of finding errors.
There are 2 basic approaches for testing: functional and structural. In functional testing the structure of the program is not considered. Test cases are decided solely on the basis of the requirements specification of the program or module and internals are not considered for selection of test cases. Due to its nature, functional testing is often called Black Box testing.
For testing purpose the test cases are designed in the following format.
Condition: -xxxxxxxxx
Input Criteria: -xxxxxx
Input: - xxxxxxxxxxxx
Result: - xxxxxxxxxxx

The condition describes the rule of the system.Input criteria specify the valid input boundaries.Input defines the sample input.Result gives the output of the sample input.

Black Box Testing

Introduction: -Black Box testing doesn't consider the structure of the program; hence the test cases that used for Black box testing are,
Test case for 'Node should connect whenever the edge(other client) is available on the network'.
Condition: "The Server/client must be under running state when a client get Connected".
Input Criteria: Any client can connect to Server, which is not running.
Input: Topic 1 wants to connect.
Result: All clients in the same topic(null by default) in the network will be shown. Topic can be changed to jump to your own group.

Test case for 'A node should be disconnected from the edge, even the node is terminated abnormally'.

Condition: " A node has to be disconnected when it is terminated abnormally".
Input Criteria: Take any node.
Input: Terminate the node program in any way (normally or abnormally).
Result: The particular node is disconnected from the edge.

Chapter-4 Findings & Conclusion

4.1 Findings

I . Citation Network

Citation network: a data set consists of paper and citation relationship chosen from ArnetMiner. It contains 10 topics:

Topic 0: Data Mining / Association Rules,
Topic 1: Web Services,
Topic 2: Bayesian Networks / Belief function,
Topic 3: Web Mining / Information Fusion,
Topic 4: Semantic Web / Description Logics,
Topic 5: Machine Learning,
Topic 6: Database Systems / XML Data,
Topic 7: Information Retrieval,
Topic 8: Pattern recognition / Image analysis,
Topic 9: Natural Language System / Statistical Machine Translation.

The dataset consists of 10 topics, e.g., graph-16.net indicates the data file is for Topic 16. For each topic, there is citation network.

Each data file, e.g., graph-T16_sub0.net, consists of two sections: *Vertices and *Edges.

Citation Network Figure : Fig-5

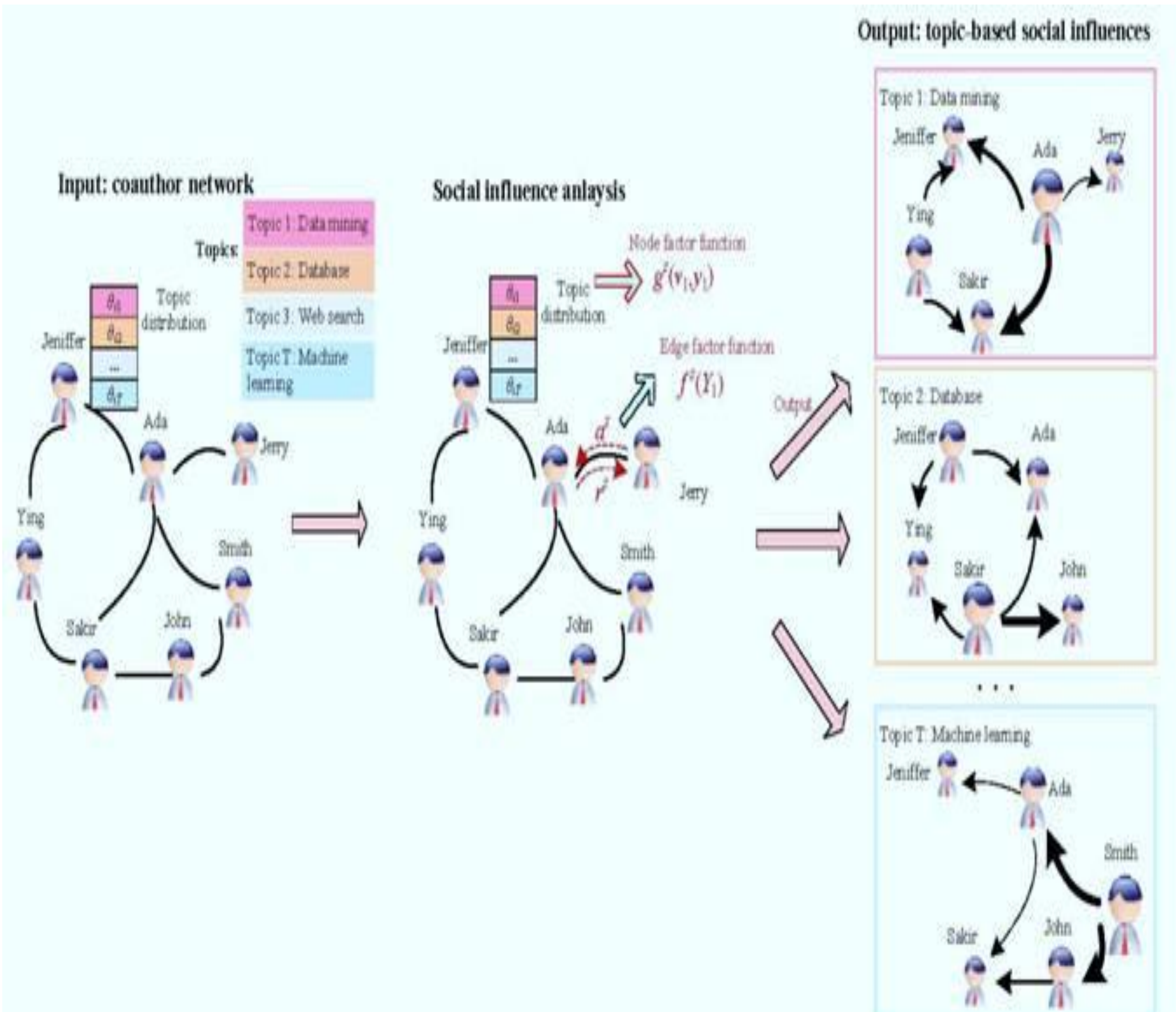


Fig 6

Results:

Number of Weakly Connected Components: 37
Number of Strongly Connected Components: 231

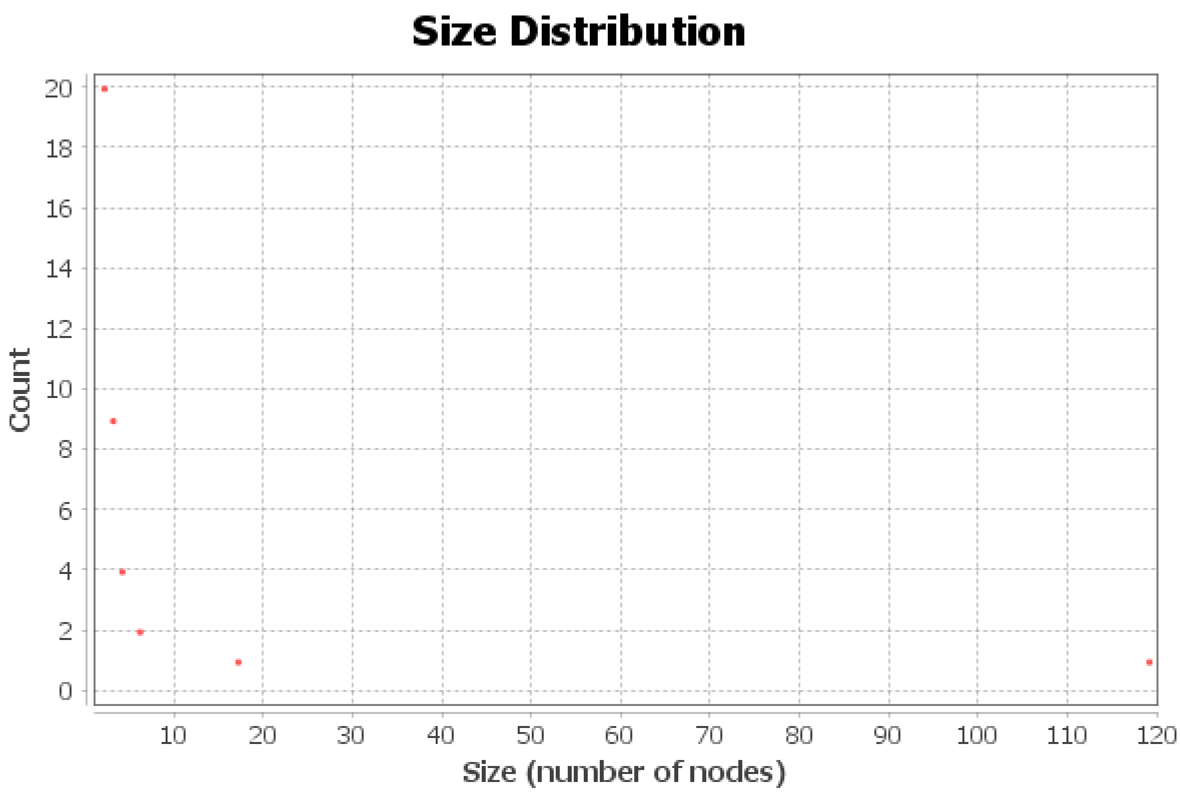


Fig-7

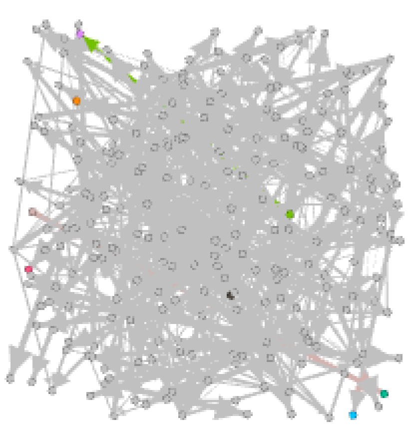


Fig 8

Weighted Degree Report

Results:

Average Weighted Degree: 0.766

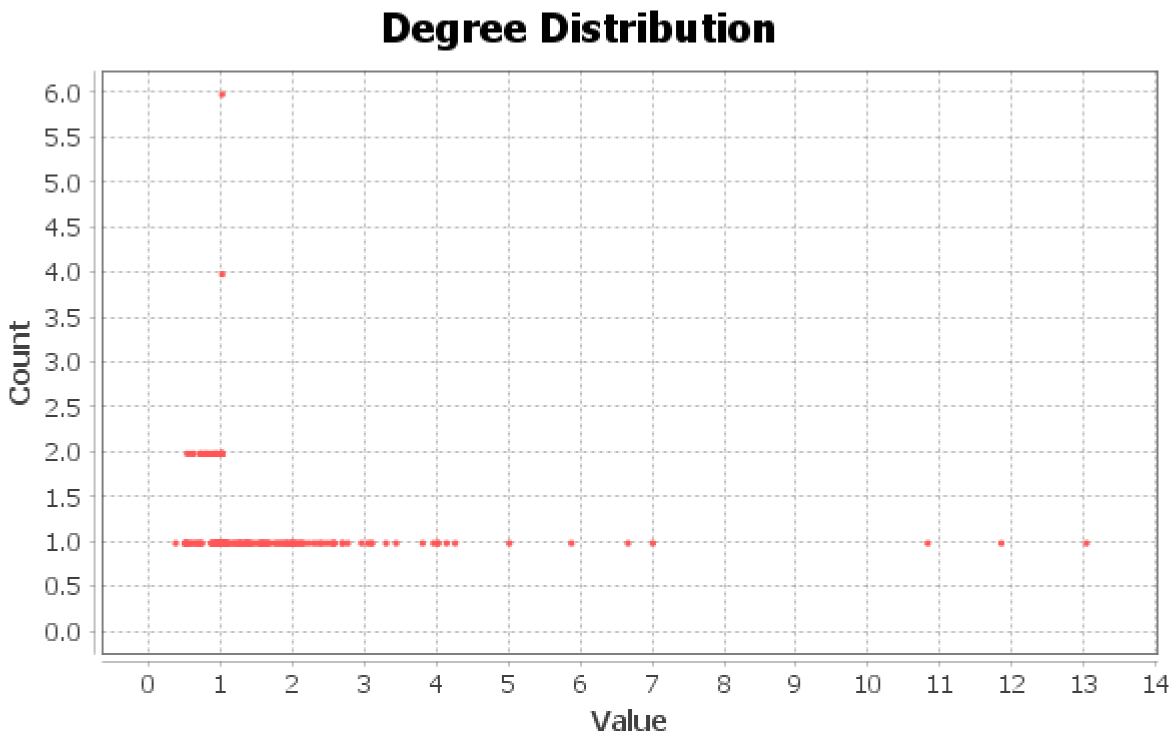
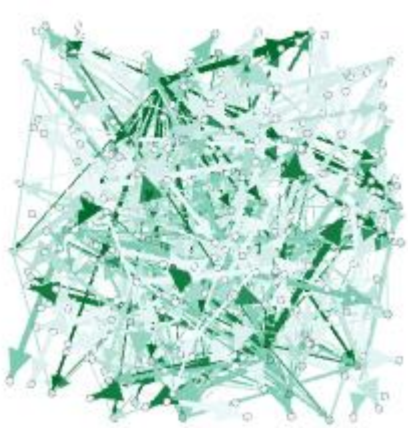


Fig 9



Analysis Results

Analysis of Different datasets provided and run on gephi tool.

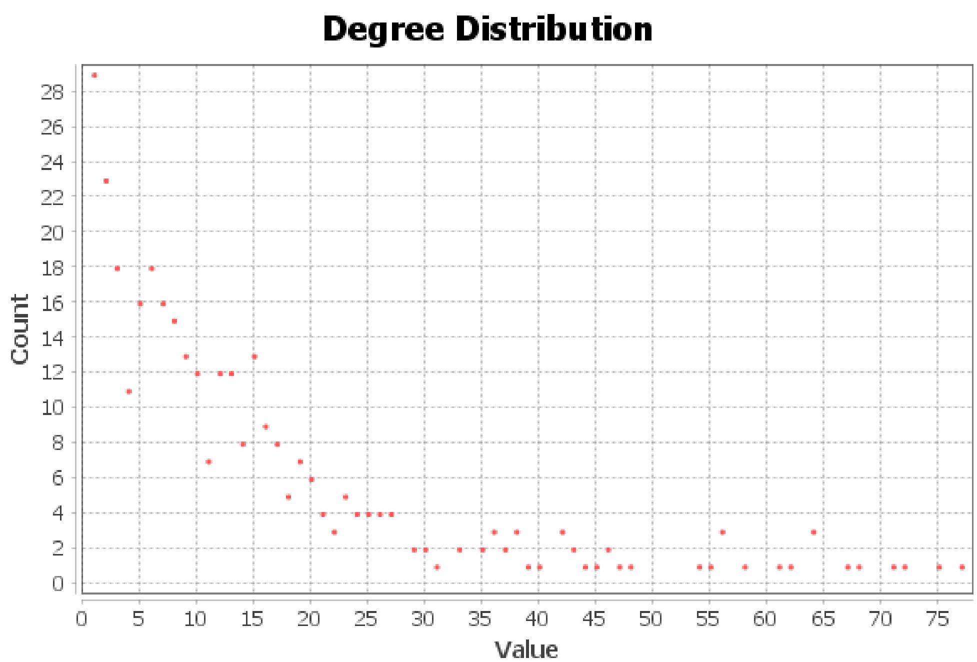
Facebook

Dataset Detail: Undirected graph, anonymized,

Nodes 4039,Edges 88234

Degree Report

Results: Average Degree: 15.129



Weighted Degree Report

Results:

Average Weighted Degree: 30.258

Graph Distance Report

Parameters:

Network Interpretation: undirected

Results:

Diameter: 11
Radius: 1
Average Path length: 3.7524459221891004

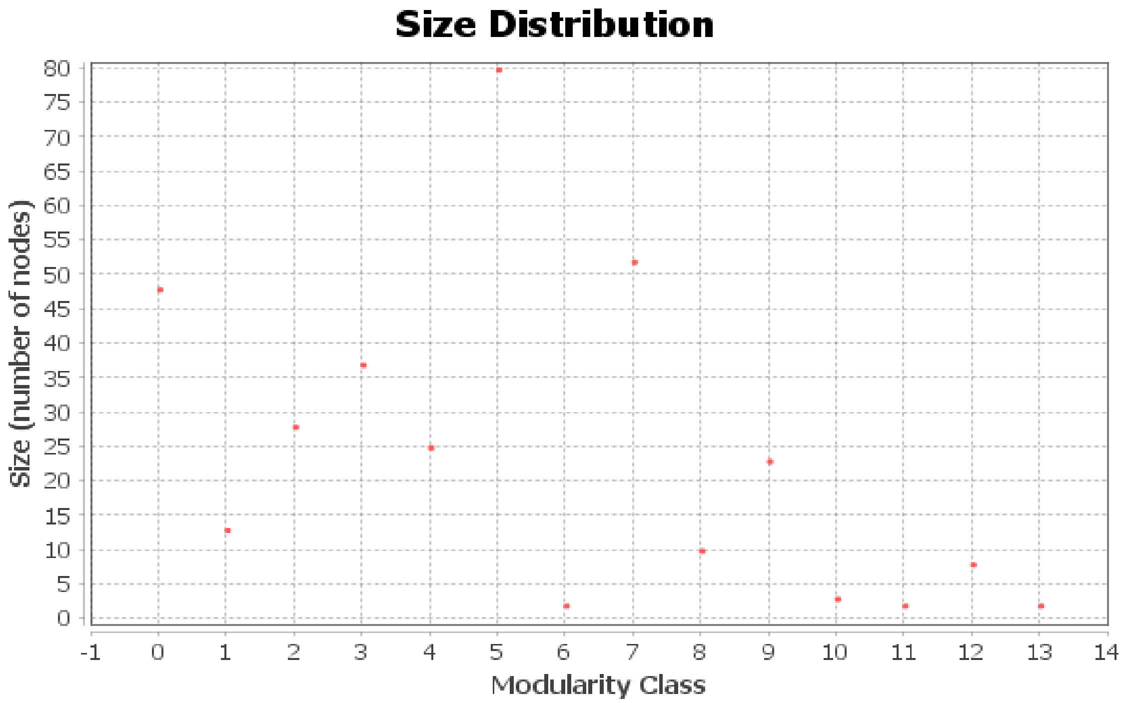
Modularity Report

Parameters:

Randomize: On
Use edge weights: On
Resolution: 1.0

Results:

Modularity: 0.458
Modularity with resolution: 0.458
Number of Communities: 14



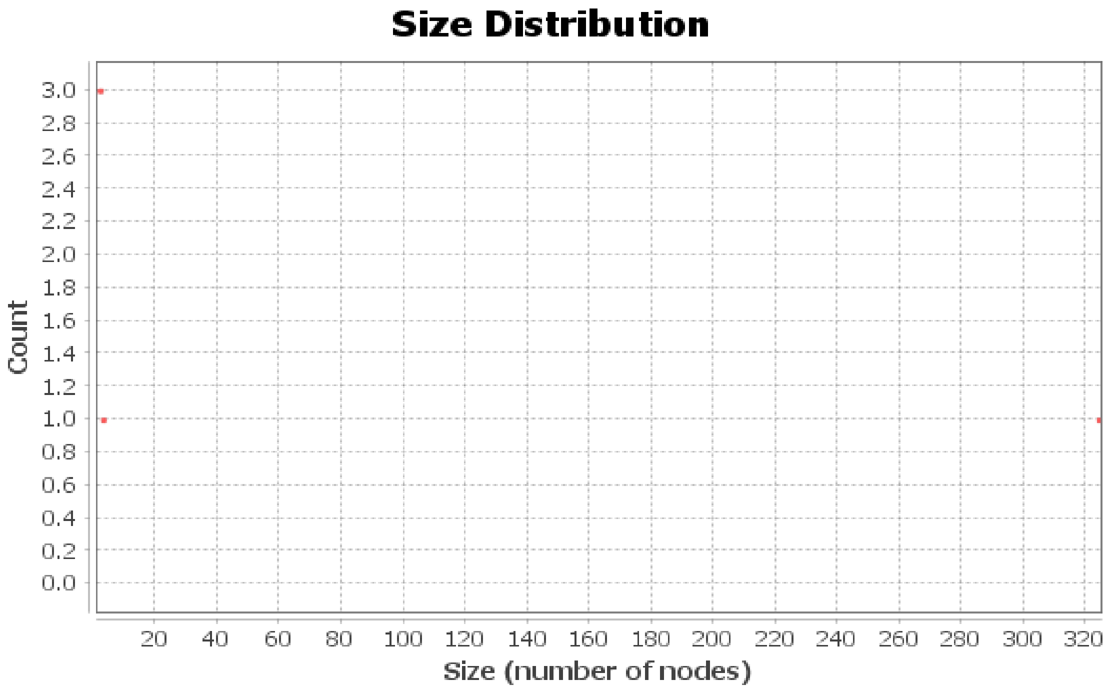
Connected Components Report

Parameters:

Network Interpretation: undirected

Results:

Number of Weakly Connected Components: 5



Clustering Coefficient Metric Report

Parameters:

Network Interpretation: undirected

Results:

Average Clustering Coefficient: 0.557
Total triangles: 10740
The Average Clustering Coefficient is the mean value of individual coefficients.

4.2 Conclusion

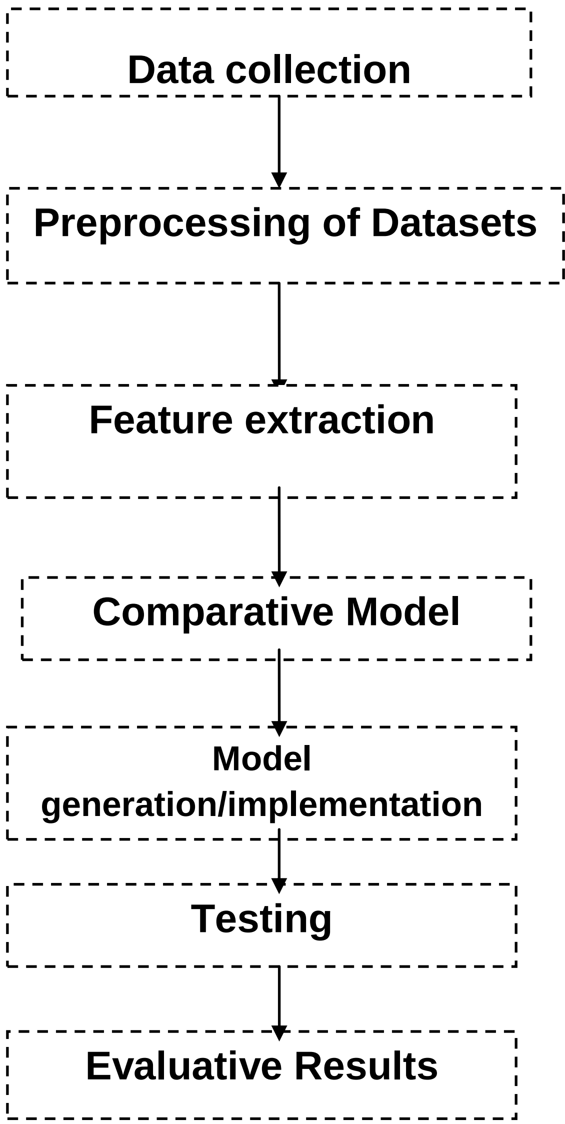
We propose a Topical Affinity Propagation (TAP) approach to describe the problem using a graphical probabilistic model. To deal with the efficient problem, we present a new algorithm for training the TFG model. A distributed learning algorithm has been implemented under the Map-reduce programming model.
Experimental results on three different types of data sets demonstrate that the proposed approach can effectively discover the topic based social influences. The distributed learning algorithm also has a good scalability performance. We apply the proposed approach to expert finding. Experiments show that the discovered topic-based influences by the proposed approach can improve the performance of expert finding.

4.3 Future Work

The general problem of network influence analysis represents an new and interesting research direction in social network mining. There are many potential future directions of this work. One interesting issue is to extend the TFG model so that it can learn topic distributions and social influences together.
Another issue is to design the TAP approach for (semi-)supervised learning. Users may provide feedbacks to the analysis system. How to make use of the useful supervised information to improve the analysis quality is an interesting problem. Another potential issue is to apply the proposed approach to other applications (e.g., community discovery) to further validate its effectiveness.

Appendix

Work Break down Structure



Gephi tool

Gephi is an open-source software for network visualization and analysis. It helps data analysts to intuitively reveal patterns and trends, highlight outliers and tells stories with their data. It **uses** a 3D render engine to display large graphs in real-time and to speed up the exploration.

References

[1] A SURVEY OF MODELS AND ALGORITHMS FOR SOCIAL INFLUENCE ANALYSIS Jimeng Sun Jie Tang

[2] Social Influence Analysis in Large-scale Networks Jie Tang Jimeng Sun Chi Wang and Zi Yang

[3] Modeling Relationship Strength in Online Social Networks .Rongjing Xiang Jennifer Neville Monica Rogati

[4] Feedback Effects between Similarity and Social Influence in Online Communities David Crandall Dan Cosley Daniel Huttenlocher Jon Kleinberg Siddharth Suri

[5] Yes, There is a Correlation - From Social Networks to Personal Behavior on the Web Parag Singla Matthew Richardson

[6] Confluence: Conformity Influence in Large Social Networks Jie Tang Sen Wu Jimeng

[7] A. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization: Scalable online collaborative filtering. In Proceeding of the 16th international conference on World Wide Web (WWW'07), 2007.

[8] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. In Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation (OSDI'04), pages 10–10, 2004.

[9] Y. Dourisboure, F. Geraci, and M. Pellegrini. Extraction and classification of dense communities in the web. In WWW'07, pages 461–470, 2007.

[10] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In SIGCOMM, pages 251–262, New York, NY, USA, 1999. ACM.

[11] G. W. Flake, S. Lawrence, and C. L. Giles. Efficient identification of web communities. In Proceedings of the