

Recommender system for community in social network

Apoorva S P¹ , Lekhashree MG¹ , Dr. K. Raghuv eer²

¹Department of Information Science and Engineering, The National Institute of Engineering, Mysore, Karnataka

²Head of the Department, Information Science and Engineering, The National Institute of Engineering, Mysore, Karnataka

Abstract - Social networking service is a platform to build relations among people who share interests, activities, backgrounds or real-life connections. Communities in a social network are the gathering places for the people with common interest. Social network analysis is in high demand nowadays for the increasing number of users. They involve themselves into different communities. They share post, their views, what they like etc in communities. So it is important for them to find suitable communities where they have common factors like friends, followers and their activities etc. In this paper, we propose a technique for recommending a community in social network based on strong friends from a user's friend list using clique and quasi-clique concepts introduced in graph mining and also using users area of interest, we recommend suitable communities for a user in a social network.

Index Terms— social network analysis, strong friends, user's interest, social media, quasi-clique.

1. INTRODUCTION

The rapid growth and exponential use of social digital media has led to an increase in popularity of social networks and the emergence of social computing. In general social networks are structures made of social entities that are linked by some specific types of interdependency (e.g. friendship, common interest, beliefs, or financial exchange). Recommendation system like social graph generation & forecasting using social network mining, finding strong groups of friends, finding popular friends in social network are well known. In this paper we develop a recommendation system for user's communities. Day by day community is increasing with the rapid growth of social users. Nowadays we see the vast activity of social life. All those community can't be suitable for a particular social users. Communities in a social network are the gathering places for the people with common interest. So it is important for them to find suitable communities where they have common factors like friends, followers and their activities etc. They involve themselves into different communities. We then find those friends who are connected with a suitable community based on finding

strong friends from a user's friend list and using quasi-clique technique.

2. PROBLEM STATEMENT

Recommendation of suitable communities for a user in a social network based on strong friends and user interest is a key factor for building a better relationships among people.

Current social networking applications such as facebook, twitter, linked in etc allows the users to create the community or group by themselves. There is no such suggestions for community creations or recommendation of existing communities for the users. Lack of this feature is a major drawback of existing social networking applications. Recommending or suggesting the communities or groups for the users plays a vital role in building the good relationships between the people

Limitations - No Suggestions for community creations, No Recommendations of Communities, Lack of user satisfaction, Less reliable and Less Efficient, Lack of customized services for the users.

3. PROPOSED SYSTEM

Proposed system is a new social networking application similar to facebook, twitter etc. Proposed system provides the services according to the users area of interest. Proposed system recommends suitable communities for a user in a social network. Proposed system establishes better relationships among users/people. Proposed system recommends the communities for the users based on users common factors such as friends, followers, posts and activities. Proposed system makes use of a technique called as "quasi clique" for community recommendation. Proposed system satisfy the users to a better extent.

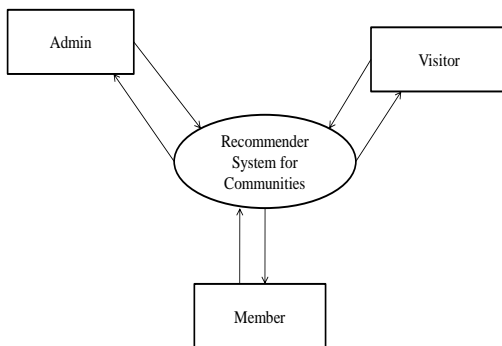
4. ANALYSIS AND DESIGN

To determine what operations will be performed, in what order, and under what circumstances, but not what kinds of data will be input to and output from the system, nor where the data will come from and go to, nor where the data will be stored (all of which are shown on a DFD). A data flow diagram (DFD) is a graphical representation of the "flow" of

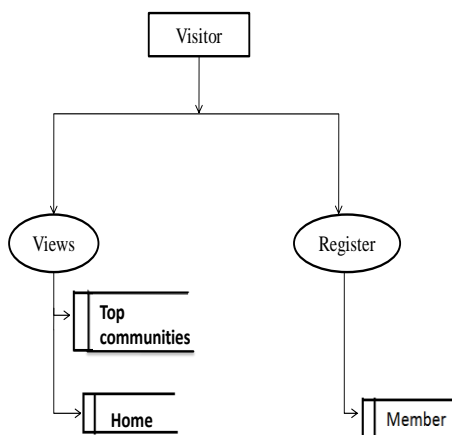
data through an information system. DFDs can also be used for the visualization of data processing (structured design).

It is common practice to draw a context-level data flow diagram first, which shows the interaction between the system and external agents which act as data sources and data sinks. On the context diagram (also known as the 'Level 0 DFD') the system's interactions with the outside world are modelled purely in terms of data flows across the *system boundary*. The context diagram shows the entire system as a single process, and gives no clues as to its internal organization.

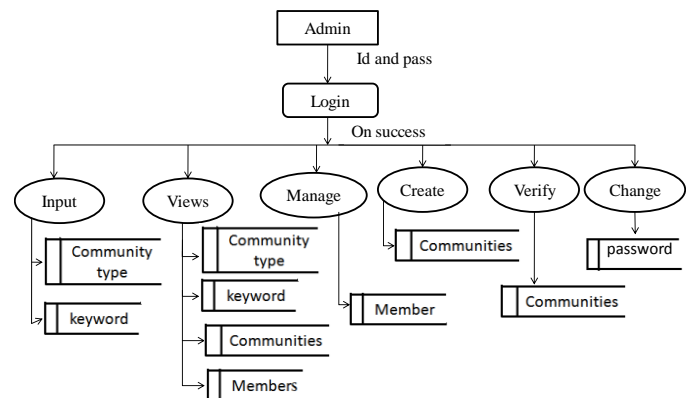
Context Flow Diagram



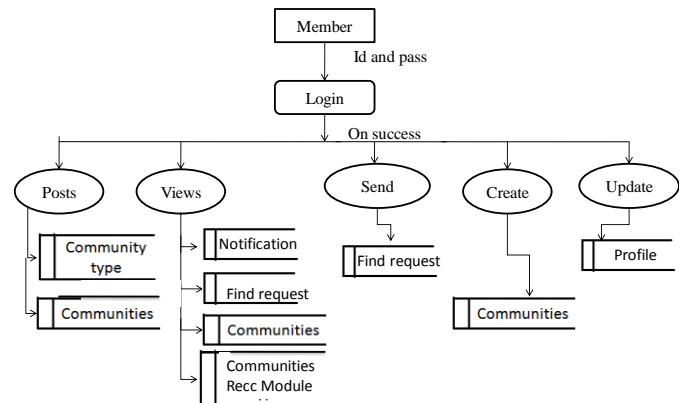
Data Flow Diagram (Visitors)



Data Flow Diagram (Admin)



Data Flow Diagram (Members)



5. SYSTEM ANALYSIS AND DESIGN

System design and analysis contains Logical Design & Physical Designing, logical designing describes the structure & characteristics or features, like output, input, files, database & procedures. The physical design, which follows the logical design, actual software & a working system. There will be constraints like Hardware, Software, Cost, and Time & Interfaces.

User Requirements -

Application mainly contains 3 users, namely -

Administrator - Administrator is one who maintains the entire application. Admin is owner of the application.

Visitor - Visitor is a person who visits to the application. Visitor is a new comer to the application. Visitor has only limited accessibility.

Member - Member is person who receives the services from the application. Member is a service receiver. Community recommendation is a key service provided by the proposed system.

Modules

Administrator has the following modules :

Login Module - In this module, administrator gets login to the application by giving login id and password. Whole application contains only one administrator. The admin id and password is set in the database once the application is deployed in the server.

Add Community Types - In this modules, administrator of the website adds the new community types, updates or delete the existing community types. Adding the community types gives the suggestions for the users for community creations.

Set Keywords for Community Type - Here administrator sets all possible keywords for each community type. Setting the keywords classifies the shared information into a particular type. This is one time updation.

Create Communities - In this module, administrator creates communities such as friends, relatives, college, school etc. Here administrator creates the communities based on the type.

View Registered Users [Members]- Here administrator can view the members profile.

Visitor has the following modules :

View Basic Information [home page, login page, registration page]

Displaying the Top Existing Communities

Registration Module - Here new users can get registered to the application by giving their details such as user id, password, name, Date of Birth etc. During the registration process the user gets an id and password, using this id and password users can get login to the application and can make use of application services.

Member has the following modules :

Login Module - In this module users can get login to the application by specifying their credentials such as user id and password and can make use of application services.

Profile Updating - In this module users can update their profile details such as general profile details, contact details, educational details, personal details, and professional details etc.

Notifications - Here users can view the notifications send by other users of the website such as friend request, events, friend request approval, new communities etc.

Find Friends - Here users can search for friends based on school name, college, profession, city wise, area wise, user name, Email Id and can send the friend request, once the request is approved users can share the information such as text messages, pictures, videos etc with each other.

Information Sharing Module - The main objective of the application is to share the information. Here website users can share the information such as text messages, pictures and videos with others.

Post Comments - Here users of the website can post the comments for the postings in their walls and can share their opinions regarding the postings such as text messages, pictures, and videos

Community Recommendation Module - This is the key service given by the application. In this module, system recommends suitable communities for a user in a social

network based on user interest in order to establish better relationships among users/people. System recommends the communities for the users based on users common factors such as friends, followers, posts and activities

6. DEFINITIONS

Input - Users common factors such as friends, followers, posts and activities.

Output - Recommendation of Communities [based on user's interest and strong friends]

Normalized Interaction Strength - We calculate the normalization value of interactive strength for every friend by dividing the interaction strength with the total number of interaction to his all friends.

Cumulative Normalized Interaction Strength - $C_n = C_{n-1} + B_n$

Minimum Strength - Minimum strength (minstr), minimum strength value (minstr) is the threshold or base value for recognizing strong and weak friends.

Strong Friends - Strong Friends (sui), we define $S_{ui} = \{s_1, s_2, \dots, s_n\}$ as a set in descending order based on nis_{ui}, f_j . When the cumulative normalized interaction strength (nis_{ui}, f_j) exceeds the minimum strength value (minstr) we consider up to those friends as a strong friend set. The set indicates that the user (ui) is most likely going to interact with these friends.

7. ALGORITHMS

7.1 Identification of User's Interests

1. Extract the user posts from the server (Mi)
2. Tokenization [keyword extraction method]
3. Classify the user posts
4. Identify the user's area of interest

7.2 Strong Friends Prediction

1. Scan the Friend database and community database
2. Normalization [normalized interaction strength](nis_{ui}, f_j)
3. Predict the cumulative normalized interaction strength [$C_n = C_{n-1} + B_n$]
4. Set Minimum strength (minstr)
5. Recognizing sui and wui - Check if($C_{NIS} < \text{minstr}$)
6. Add to strong friends
7. Only consider those communities(C_{ui}) [based on user's area of interest] which contains strong friends.
8. Calculate the normalized interaction strength(nis_{ui}, f_j) based on the community.
9. Recommendation - display the communities(C_{ui}) in the descending order.
10. If no communities which contains strong friends then
11. calculate the normalized interaction strength(nis_{ui}, f_j) for all communities
12. Recommendation - display the communities(C_{ui}) in the descending order.

8. IMPLEMENTATION

Methodology – The waterfall model is a popular version of the systems development life cycle model for software engineering. Often considered the classic approach to the systems development life cycle, the waterfall model describes a development method that is linear and sequential. Waterfall development has distinct goals for each phase of development. Imagine a waterfall on the cliff of a steep mountain. Once the water has flowed over the edge of the cliff and has begun its journey down the side of the mountain, it cannot turn back. It is the same with waterfall development. Once a phase of development is completed, the development proceeds to the next phase and there is no turning back.

Technology

.Net framework - The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet. The .NET Framework is designed to fulfill the following objectives: To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.

C# - (pronounced as 'C Sharp') is a new computer-programming language developed by Microsoft Corporation, USA. C# is a fully *object-oriented* language like Java and is the first Component-oriented language. It has been designed to support the key features of .NET Framework, the new development platform of Microsoft for building component-based software solutions. It is a simple, efficient, productive and type-safe language derived from the popular C and C++ languages. Although it belongs to the family of C/C++, it is a purely object-oriented, modern language suitable for developing Web-based applications.

3 Tier Architecture - Three tier architecture consists of three layers. They are :

The Data Layer: The key component to most applications is the data. The data has to be served to the presentation layer somehow. The data layer is a separate component (often setup as a separate single or group of projects in a .NET solution), whose sole purpose is to serve up the data from the database and return it to the caller.

Business Layer: Though a web site could talk to the data access layer directly, it usually goes through another layer called the business layer. The business layer is vital in that it validates the input conditions before calling a method from the data layer.

Presentation Layer: The ASP.NET web site or windows forms application (the UI for the project) is called the presentation layer. The presentation layer is the most important layer simply because it's the one that everyone sees and uses. Even with a well structured business and data layer, if the presentation layer is designed poorly, this gives the users a poor view of the system.

9. CONCLUSION

In this paper we work through two main processes. Those are selecting strong friends and user's interest and then applying quasi-clique technique to recommend a suitable community for a particular user. Usually when we work on some huge database it will take more time to check as well as to compare. But if we represent the database in a graph and work on that graph then it will be time consuming and easily understandable. As in our calculation a portion of a user's friends is accessed at a time so it is faster and we can visualize easily. Our future plan is to simplify our algorithm and add experimental results on real social network data.

10. REFERENCES

- [1] Marjaneh Safaei, Merve Sahan and Mustafa Ilkan, "Social Graph Generation & Forecasting using Social Network Mining," 33rd Annual IEEE International Computer Software and Applications Conference (COMPSAC'09), 2009, pp. 31-35.
- [2] Kadge, Sanam and Bhatia, Gresha, "Graph Based Forecasting For Social Networking Site", Communication, Information & Computing Technology (ICCICT) International Conference , 2012, , pp. 1-6.
- [3] Cameron, Juan J and Leung, CK-S and Tanbeer, Syed Khairuzzaman, "Finding Strong Groups of Friends among Friends in Social Networks", Ninth IEEE International Conference, 2011, pp. 824-831.
- [4] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in Proc. ACM SIGMOD 1993, Volume 22, pp. 207-216.
- [5] Silva, Arlei and Meira Jr, Wagner and Zaki, Mohammed J, "Structural correlation pattern mining for large graphs", Proceedings of the Eighth Workshop on Mining And .Learning with Graphs, 2010.