FRIEND MATCHING USING PROBABILISTIC TOPIC MODEL

Vidya R. and Nishada S. G.
Department of Computer Science and Engineering, Mohandas College of Engineering, Kerala, India

ABSTRACT
Recommender Systems provide suggestions for users to guide in various decision-making processes. The recommender systems can be defined by the purpose of recommendation, mechanism and data gathering. Recommendation system for social networks are different since the items recommended are rational human beings. The paper focuses on designing a friend matching system by analyzing user lifestyles as common criteria. Large amount of data collected from various users create high dimensional data. In order to resolve this, probabilistic topic modeling is used. Content based machine learning approaches are used to find out suspicious users in the recommendation system. The results are evaluated based on the datasets created from the real world users.

Keywords: friend-of-ofriend, probabilistic topic modeling, latent dirichlet allocation.

INTRODUCTION
Online Social network services have thrived in great popularity recent years. Social networking sites have attracted tremendous numbers of users and play an important role in online interaction. By connecting users with similar common interests, online social networks open up a new channel for information sharing and social networking. The open ended nature of their applications motivates rich user-generated content, including tags, text document, multimedia, and so on.

One fundamental phenomenon in social network services is friendship formation. Members make friends with each other through social interactions and information exchange. A member in online social network may be frustrated to find new friends from a tremendous number of irrelevant users. Suggesting relevant users with common interests to each individual can help improve user experience. Most social network websites match members based on the number of mutual friends. This method suffers the drawback of interest mismatch and it is useless to expand the circle of the members. According to these studies the rules to group people together include: 1) habits or life style; 2) attitudes; 3) tastes; 4) moral standards; 5) economic level and 6) people they already know. Apparently, rule #3 and rule #6 are the mainstream factors considered by existing recommendation systems. Grouping friends by means of their habits were not adopted widely. This is due to the difficulty in analyzing user habits in real time. Rather, life styles are usually closely correlated with daily routines and activities. Therefore, if it is possible to obtain the daily routines of users then we can recommend new friends based on the common habits.

The remainder of this paper is organized as follows:

Section II provides an overview on related works in friend recommendation system. Section III provides system description of the proposed work. It explains the model for capturing user habits, how to identify lifestyles and generate lifestyle vector. This also explains an efficient friend recommendation algorithm. Section IV deals with datasets used in the work. Section V presents the results of applying the proposed method and Section VI concludes.

RELATED WORK
Xiao Yu, Ang Pan, Lu-An Tang [1] proposed a friend recommendation mechanism by identifying geographically related friends. This friend recommendation approach considered the users current geography. Similarity among user interests were not included which lacks the user’s preference on friend selection in real world. Alvin Chin, Bin Xu, Hao Wang [2] proposed a friend recommendation based on physical context. Here physical context is based on meetings and encounters. The method uses the intuition that people who meet for a conference can be recommended as friends. Jeff Narachitparames, Mehmet Hadi Gunen and Sushil J. Louis [3] proposed a friend recommendation based on network topology and genetic algorithm. This approach also ensures the likelihood of a person pursuing a friendship of someone they know than someone they do not know. Zhing Wang [4] proposed Friendbook which extracts user habits based on sensors like accelerometer and gyroscope. The user activities captured were limited to indoor activities. The proposed method captures user habits with the help of smartphones. Inspired by advancements on probabilistic topic modeling; the study proposes a generative model based on LDA (Latent Dirichlet Allocation) for dimensionality reduction. Thus topic model can generate relevant interests of a user. The proposed method provides a modified friend recommendation algorithm that enhances accuracy compared to traditional algorithms.

THE PROPOSED SCHEME
Friend Matcher is an android application that generates new friendship recommendations to the users. Sensors in Smartphone are used to capture user’s daily habits. The user habits are captured and analyzed by means of probabilistic topic modeling. The habits captured can be divided based on regular activities and user behavior. Regular activities comprises of lifestyles like traveler, teacher, student and so on. The parameters we
consider for lifestyle analysis are SMS, search history, application installed and detection of suspicious activity. The previous works considered various indoor sensor readings to detect user habits. The flowchart below depicts the model used for capturing user life habits.

Data Collection
Interest of the user is analyzed by location based services, search history, SMS and applications installed in the smartphone. The system updates the server about GPS coordinates and the services accessed by the user. Further by the application of text mining interests can be captured from SMS and search history. For this an unsupervised learning approach called probabilistic topic modeling is used. The same word can appear in different topics due to its ambiguity.

Lifestyle Analyzer
Topic modeling uses a non deterministic algorithm called as LDA [11]. This will identify lifestyles of the user. In LDA [11], features of each document can be divided into two parts: features relevant to user lifestyles and irrelevant features like stopwords. LDA outputs set of meaningful words and corresponding ranks. The algorithms for user lifestyle generation are as follows:

Table-1. Identify user lifestyles using topic model.

| Step 1: | For each label  
| For each user document in the collector  
| Train the LDA model for k topics  

Obtain top M words for topic K by posterior probability

Step 2: Merge all top M words for all documents  
Generate the lifestyle vector.

Each user’s collector module consists of several lifestyles. System represents each user in terms of a lifestyle vector. Lifestyle vectors are represented as probabilities of several lifestyles over collector. Collector module logs all user routines. Let the lifestyles of user1 be lifestyle1, lifestyle2. Then we write:

$$L1 = [p(lifestyle1|collector1), p(lifestyle2|collector1)]$$  

The topics or the lifestyles extracted are updated in the database. Based on the parameters we consider for lifestyle analysis, the equation for lifestyle vector of a user is written as:

$$\text{LF}_u = \text{LF}_{GPS} + \text{LF}_{URL} + \text{LF}_{SMS} + \text{LF}_{APP} + \text{LF}_{SP}$$  

Thus lifestyle vector of user depends on values obtained from GPS, visited URLs, messages, applications in the mobile and detection of suspicious activity. The similarity between users is calculated based on the cosine similarity. A threshold value is set and only those values above the threshold are considered for friendship calculation. Based on the values obtained, a graph is plotted with users as nodes and similarity value as edges. For friendship recommendation, a friend up to level3 is considered from the friendship graph. This is to ensure that recommendations are not obtained from complete strangers.

In order to check the suspicious activity a spam classifier is used. The system is trained with set of spam and non-spam messages. Based on this training set, system can identify whether a lifestyle belongs to spam category or not. A suspicious user thus identified is not eliminated from the system. Instead a negative score is assigned to such a user. Figure-2 shows the modules used in the system.
The similarity value calculated between the users provides the friendship score. These score values are sorted in the descending order. Corresponding users with the top five scores are recommended as new friends. The users can express the satisfaction on received recommendations using the feedback module. An algorithm for friend matching is proposed which will select top five friends based on the recommendation score. Modules used in the system are:

**Similarity Calculation**

In order to find interest similarity between users we use cosine similarity as the metric. Let $L_1 := \{p(z_1|d_1), p(z_2|d_1), p(z_3|d_1) \ldots p(z_t|d_1)\}; L_2 := \{p(z_1|d_2), p(z_2|d_2), p(z_3|d_2) \ldots p(z_t|d_2)\}$ where $z$ represents the user lifestyles and $d$ represents the document in collector. Therefore, the similarity of habits between user1 and user2 is denoted by:

$$\text{Sim}(u_1, u_2):= \cos(L_1, L_2)$$

**Algorithm 1: Find Similarity between users**

- **Step 1:** For each lifestyle $z_k$, the probability is not zero
  - If $z_k = \text{sms}$ then Check suspicion (); End if
  
  $\text{Sim}_{i,j} = \text{vector}(i) \times \text{vector}(j)$
  
  $k = k+1$

- **Step 2:** For each user with no match in lifestyle do
  - Initialize the similarity value, $S(i,j) = 0$
  - Endfor

**Friend Graph Creation**

Using similarity metric, we find the interest similarity between users. Then a friend matching graph is created. It is a weighted graph $G(V,E,W)$ where users in the system are represented as vertices and the similarity score as edges. Only if there is similarity in lifestyles an edge will be created. The similarity score obtained should be greater than the threshold. Such edges are only inserted. Thus weight of an edge $W(i,j)$ is $S(i,j)$. Here we use friend of friend (FoF) approach so as to avoid receiving recommendations from complete strangers. For friendship recommendation, friends up to level3 are considered.

**Friend Suggestion**

A user when submits a query for friend suggestion, pool creation and rank calculation takes place. For each user with similar interests the system calculates friendship score. Suppose the system finds ‘$j$’ similar users for user1. Then, rank calculation is given by the equation $F_{\text{score}}(u_1, u_2)$:

$$F_{\text{score}}_{u_1, u_2} = \text{Sim}_{u_{best}} + \text{Sim}_{u_{next}} + \text{Sim}_{u_{avg}} + \text{Sim}_{u_{avg}}$$

The system first creates a pool of users. Among the users, new friends will be suggested. This pool of users are created from the friend graph where the system starts capturing details of friends of friends and navigate deep down the tree till level 3. We want to limit recommendations from strangers, hence we set the threshold as 3. Algorithm is illustrated below:

**Algorithm 2: Create list for recommendation**

- **Step 1:** alreadyfriends := getFriends ();
  - alreadyfriends_level:= 0
  - temp ← alreadyfriends

- **Step 2:** while (temp≠ null)
  - For each user ‘x’ from temp
  - current ← temp (0) //first entry in temp;
  - remove.temp (0);

- **Step 3:** if current. Level < 3 then
  - Frndlist:= getfrndsof_current();
  - For each user ‘y’ in the frndlist
  - y.level:= current. Level + 1
  - temp.add (y.)
  - End for

  End if end for

**End while**

The system selects friends for recommendation from the pool created. The friend suggestion method is explained with the help of algorithm.

**Algorithm 3: Friend Suggestion Algorithm**

- **Step 1** Extract user $U_i$, lifestyle vector using topic modeling.
- **Step 2** Compute lifestyle vector for each
- **Step 3** get recommend_list ()
- **Step 4** For each user in recommend_list,
  - Getsimilarity ()
  - Draw the friendship graph();
  - Fscore:= $\text{Sum}_{u_{best}} + \text{Sim}_{u_{next}} + \text{Sim}_{u_{avg}} + \text{Sim}_{u_{avg}}$
  - Endfor

**Endfor**

- **Step 5** sort all users in decreasing order according to score
- **Step 6** Output the top five users from the sorted list

**Identify Suspicious Activity**

In order to check the suspicious activity a spam classifier is used. Using topic modeling, the user SMS is used for monitoring this activity. Initially the system is trained with set of spam and non-spam messages. Based on this training set, system can identify whether a lifestyle belongs to spam category or not. We find out the score for user sms based on the rank calculation. Rank is expressed in terms of probability distribution. Difference between the ranks is calculated to identify if the user is suspicious. A suspicious user thus identified is not eliminated from the system. Instead a negative score is assigned to such a user. The server maintains a list of suspicious users.
Algorithm 4: Find malicious users

Check suspicion ()

Step 1: Train the system with set of spam and non spam messages

Step 2: For each user in the friend list

- Check the distribution of spam and non spam

  \[ \text{Score} = \text{rank (no spam)} - \text{rank (spam)} \]

Step 3: If \((\text{score} < 0)\) then

  Raise suspicion

  \[ L_{F_{\text{sms}}} = - (L_{F_{1}}) \]

End if End for

EXPERIMENTAL DATA

We rely on datasets directly collected from the user’s Smartphone. Based on the GPS coordinates, the services accessed by the users are obtained. Similarly the user browsed URLs are captured from the mobile device in real time. The SMS database used in the study contains one set of SMS messages in English of 5,574 messages, tagged according being ‘not spam’ and ‘spam’. Table-3 provides some statistics of SMS dataset.

Table-2. SMS dataset details.

<table>
<thead>
<tr>
<th></th>
<th>No spam</th>
<th>spam</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total SMSes</td>
<td>4827</td>
<td>747</td>
</tr>
<tr>
<td>Average number of words</td>
<td>0.133</td>
<td>0.167</td>
</tr>
<tr>
<td>Average Presence of urls</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

Following are the observations obtained from analyzing SMS. Numbers of special characters were high for no spam category as the messages were informal. Presence of URL was found more in spam category. Figures below represent tag cloud generated from analyzing SMS messages using topic modeling. The system is trained with set of spam and non spam messages together. Suspicion is raised whenever data collected from the user contains threat beyond a threshold value.

EXPERIMENTAL RESULTS

The results were evaluated with the help of eight Smartphone users. Each user used a Smartphone with Friend Matcher installed. The lifestyles considered were office, student, researcher, lecturer and business. Using topic modeling lifestyles of the users were classified. Classification result depends on the value set as number of topics. It is observed that as the number of topics is set to 100, classification results are better. Figure-4 shows the lifestyle distribution for a particular user.

Figure-4. Lifestyle distribution.

The lifestyle distribution of each user can be graphically represented based on the probability distribution. For evaluation purpose six lifestyles are considered. The bar chart indicates the relevance of each lifestyle to a user. Thus the lifestyle business contributes to 0.8, computers to 0.7 and student to 0.4. Thus lifestyle vector is denoted by \( L_1 = \{0.8, 0.7, 0.4 \ldots\} \). The final recommendation scores obtained is shown below.

Figure-5. Recommendation scores.

The interesting factor in our work is the consideration of more realistic parameters in the system. The parameters considered depend on the lifestyle values. The lifestyle values affect the friendship scores. This in turn influences the efficiency of friend recommendation system. Here we plot two graphs with user size as x-axis and lifestyle weights as y-axis. The Friend Graph is plotted with lifestyle weights obtained by considering parameters like location based services and visited URLs. In this case lifestyle weight is obtained by:

Figure-3. Nonsparse and spam words identified by topic modeling.
S&P Friend graph considers the modified parameters like SMS, applications installed, music and suspicion detection also. The modified parameters change the lifestyle weights. Thus we write:

\[
LF_{ij} = LF_{esp} + LF_{URL} \tag{4}
\]

We observe difference in lifestyle weights between the two graphs due to the modified parameters considered. This difference in lifestyle weight is the error we have identified using our proposed method. This justifies that our work enhances accuracy in recommending friends.

We define the difference in lifestyle weights between the two graphs as the error we have identified using our proposed method. This justifies that our work enhances accuracy in recommending friends.

\[
LF_{ij} = LF_{esp} + LF_{URL} + LF_{sms} + LF_{music} + LF_{susp} \tag{5}
\]

CONCLUSIONS

The work proposes an approach quite different from the existing friend recommendation mechanisms. The latter relies on collaborative method of recommending friends. Whereas the proposed work extracts user habits by tracking daily activities of the user. It recommends potential friends to users if they share similar life styles. The result shows that the recommendations accurately reflect the preferences of users in choosing friends.

As the system extracts the user interests, this can be used to recommend several interesting features to a particular user. Thus the user receives meaningful information according to his tastes. The friendship parameters used in the system helps to increase user’s credibility in system. As a means to offer privacy, the complete information of the user is not provided while recommendation. System just displays the relevant score calculated.

ACKNOWLEDGEMENTS

I would like to thank all the participants who participated for data collection. I extend my gratitude to Prof. Nishada S.G for providing guidance to this work.

REFERENCES


VINCI, page 15, 2011.


