Friend Recommendation System based on Semantic

Pooja Desai[1], Rupali Bosale[2], Kaminee Bodakhe[3], Sonali Agwan[4]

[1] *Department Of Computer Engineering, SRES College Of Engineering Kopargaon.*
[2] *Department Of Computer Engineering, SRES College Of Engineering Kopargaon.*
[3] *Department Of Computer Engineering, SRES College Of Engineering Kopargaon.*
[4] *Department Of Computer Engineering, SRES College Of Engineering Kopargaon.*

Abstract

In Existing social networking services recommend friends to users based on their social graphs, which is not appropriate to reflect a user's preferences on friend selection in real life. We present a semantic-based friend recommendation system for social networks in which friends are recommended to user according to their life styles instead of social graphs. By using friend matching graph algorithm, a friend matching graph is being generated which shows/measures similarity of life styles between users, and recommends friends to users if their life styles have high similarity. By using text mining, we track user's daily life as activity documents, from which his/her life styles are extracted by using the Latent Dirichlet Allocation algorithm. We also propose a similarity metric to measure the similarity of life styles between users, and rank the user based on their friend life style high similarity friend matching graph. After receiving a request, Friend recommendation system returns a list of people with highest recommendation scores to the query user. Finally, Friend Recommendation System integrates a feedback mechanism to further improve the recommendation accuracy.

Keywords- *Friend Recommendation system, Life extraction analysis system, feedback control, matching graph, Recommend Friend.*

---

1. INTRODUCTION

Existing social networking services recommend friends to users based on their social graphs, which may not be the most appropriate to reflect a user's preferences on friend selection in real life. In this paper, we present Friend, a novel semantic-based friend recommendation system for social networks, which recommends friends to users based on their life styles instead of social graphs. By taking advantage of web based, Friend discovers life styles of users from user-centric sensor data, measures the similarity of life styles between users, and recommends friends to users if their life styles have high similarity. Inspired by text mining, we model a user's daily life as life documents, from which his/her life styles are extracted by using the Latent Deric let Allocation algorithm. We further propose a similarity metric to measure the similarity of life styles between users, and calculate users' impact in terms of life styles with a friend-matching graph. Upon receiving a request, Friend returns a list of people with highest recommendation scores to the query user. Finally, Friend integrates a feedback mechanism to further improve the recommendation accuracy. We have implemented Friend on the online web based system, and evaluated its performance on both small-scale experiments and large-scale simulations. The results show that the recommendations accurately reflect the preferences of users in choosing friends.

1.1 EASE OF USE

A. Problem definition

A friend Recommendation System for Social Network, which recommends friend to user based on their life styles and rank the friends according to their daily activities performed. System also identifies the fake accounts and notify the friend accordingly. In case of wrong user identification, system provides users to clarify himself/herself in the form of feedback control.

B. Scope

We can recommend a friend based on their life styles for social networking sites. Modules of the system will include life style extraction, friend matching graph ,query and friend recommendation and feedback control. System will update the friend list as per the user response. System also identifies fake account and notifies the friends

2. SYSTEM ARCHITECTURE

Following are different modules in the proposed system.

1. Life style extraction
2. Life style analysis
3. Generation of friend matching graph
4. Friend recommendation
5. Feedback control

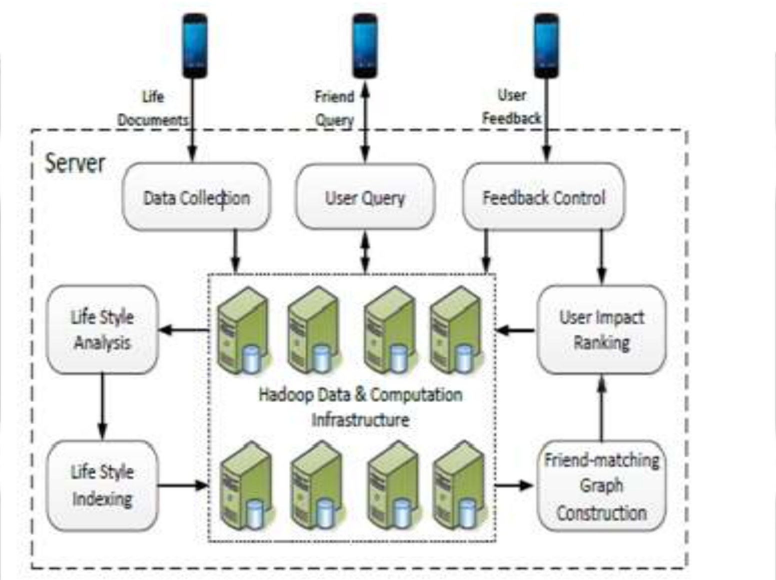

Fig-1: System architecture of Friend Recommendation System

C. Life style extraction

This module is intended for data collection from the different users. Extract the keywords from the data which can be used for life style analysis. This data can be collected from different aspects like life style activities etc. of the user.

D. Life style analysis

This module will analyse the data collected by previous module. The analysis can be in the form of counting the frequency of getting particular word in use by user, the users work activities etc. based on that we can connect the users with similar life style.

E. Generation of friend matching graph

To characterize the relations among users, system will generate the friend matching graph to represent the similarity between their life style and how they influence other people in the graph. In particular we will use the link weight between two users to represent the similarity of their life styles. Based on the friend matching graph, we can obtain a user's affinity reacting how likely this user will be chosen as another user's friend in the network.

F. Friend recommendation

This module will recommend friends to the users by analysing friend matching graph.

G. feedback control

To support performance optimization at runtime, we will implement a feedback control mechanism. Server generates a reply in response to a query; the Feedback mechanism allows us to measure the satisfaction of users, by providing a user interface that allows the user to rate the friend list**.**

2.1 LIFE STYLE EXTRACTION USING LDA

Given the life documents of all users, Eq. 1 can be further represented as a matrix decomposition problem.

$$p(w|d) = p(w|z)p(z|d) \qquad (1)$$

where $p(w|d) = [p(w|d_1),p(w|d_2),..,p(w|d_n)]$ is the activity-document matrix as shown in containing the probability of each activity over each life document, and $p(w|dk) = [p(w_1|d_k), p(w_2|d_k),\ldots,p(w_W|d_k)]^{\mathbf{T}}$ is the kth column in the activity-document matrix representing the probabilities of activities over the life document d_k of user k;

$p(w|z) = [p(w|z_1), p(w|z_2),.., p(w|z_Z)]$ is the activity topic matrix as shown in Figure 5 representing the probability of each activity over each life style (topic), and $p(w|zk) =[p(w_1|z_k),p(w_2|z_k),..,p(w_W|z_k)]^T$ is the kth column in the Activity-topic matrix representing the probabilities of activities over the life style z_k; $p(z|d) = [p(z|d_1); p(z|d_2),.., p(z|dn)]$ is the topic document matrix as shown in Figure 5 containing the probability of each topic over each life document, and $p(z|d_k)= [p(z_1|d_k),p(z_2|d_k),.., p(z_Z|d_k)]^T$ is the kth column in the topic-document matrix representing the probabilities of life styles over the life document d_k of user k.

The above matrix decomposition problem is actually the Latent Dirichlet Allocation (LDA) model. We use the Expectation Maximization (EM) method to solve the LDA decomposition, where the E-step is used to estimate the free variational Dirichlet parameter and γ parameter ϕ in the standard LDA model and the M-step is used to maximize the log likelihood of the activities under these parameters.

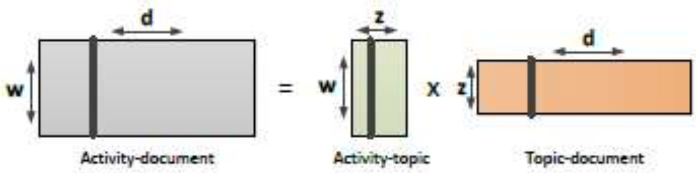

Fig. 5: Matrix decomposition for life styles analysis. (Redrawn from [19])

After the EM algorithm converges, we are able to calculate the decomposed activity-topic matrix. Readers are referred to for more details of the LDA algorithm and alternative decomposition approaches. It is worth noting that since our system uses unsupervised learning algorithms to recognize activities and the topic model to discover life styles, the derived "activities" (Multinomial or cluster centers from the K-means algorithm) or "topics" do not carry physical meanings. As mentioned in, such meaning can be estimated via the additional step of comparing the topic activations to the actual structure of the subject's day and then identifying topics that correspond to possible daily routines. In Friend, since we are to only compare "similarity" in activities or topic patterns, there is no need to infer the physical meaning of each cluster center or topic. On the other hand, not revealing the actual physical meaning of activities and topics also has advantages from the perspective of preserving privacy.

2.2 SIMILARITY METRIC

We define a new similarity metric to measure the similarity between two life style vectors. Let Li = $[p(z_1 | d_i), p(z_2 | d_i), .., p(z_Z | d_i)]$ and $L_j = [p(z_1 | d_j), p(z_2 | d_j), .., p(z_Z | d_j)]$ denote the life style vectors of user i and user j, respectively. We argue that the similarity is not only affected by their life style vectors as a whole, but also by the most important life styles, i.e., the elements within the vector with larger probability values, also known as the dominant life styles. We also argue that two users do not share much similarity if majority of their life styles are totally different. Therefore, the similarity of life styles between user i and user j, denoted by S(i,,j), is defined as follows:

$$S(i, j) = S_c(i, j) \, . \, S_d(i, j)$$

where $S_c(i, j)$ is used to measure the similarity of the life style vectors of users as a whole, $S_d(i, j)$ is used to emphasize the similarity of users on their dominant life styles. We adopt the commonly used cosine similarity metric for Sc(i, j), that is,

$$Sc(i, j) = \cos(Li, L_j)$$

In order to calculate $S_d(i, j)$, we first define the set of dominant life styles of a user.

Algorithm 1 Computing users' impact ranking

Input: The friend-matching graph G.
Output: Impact ranking vector $\mathbf{r}$ for all users.

for $i = 1$ to n **do**
  $\mathbf{r}_0(i) = \frac{1}{n}$
end for
$\delta = \infty$
$\epsilon = e^{-9}$
while $\delta > \epsilon$ **do**
  for $i = 1$ to n **do**
    $\mathbf{r}_{k+1}(i) = \sum_j \frac{1-\varphi}{n}\mathbf{r}_k(j) + \varphi \frac{\sum_j \omega(i,j) \cdot \mathbf{r}_k(j)}{\sum_j \omega(i,j)}$
  end for
  $\delta = \sum_{i=1}^{n} |\mathbf{r}_{k+1}(i) - \mathbf{r}_k(i)|$
end while
return $\mathbf{r}$

Algorithm 2: Friend recommendation

Input: The query user i, the recommendation coefficient β

and the required number of recommended friends from the system p.
Output: Friend list Fi.
1. Fi←Ø, Q←Ø.
2. extracts i's life style vector Li using the LDA algorithm.
3. **for** each life style zk the probability of which in Li
is not zero **do**
4. put users in the entry of zk into Q
5. **end for**
6. **for** each user j ≠ Q **do**
7. S(i, j) ← 0
8. **end for**
9. **for** each user j in the database **do**
10. $R_i(j) = \beta S(i, j) + (1-\beta) r_j k$
11. **end for**
12. Sort all users in decreasing order according to $R_i(j)$
put the top p users in the sorted list to F_i

2.3. FEEDBACK CONTROL

To support performance optimization at runtime, we also integrate a feedback control mechanism into Friend. After the server generates a reply in response to a query, the feedback mechanism allows us to measure the satisfaction of users, by providing a user interface that allows the user to rate the friend list. Let ŕ denote the impact ranking vector calculated from the feedback of users. Here, ŕ = [_ŕ (1), ŕ(2),.., ŕ(n)]^T where n is the number of current users of the system. Let ŕ(i, j) denote the score that user j rates user i. Then we have: _r(i; j) =(r(i; j) if user j rates user i. r(i) where the second equation in means that the feedback score is equal to the original score if user j does not rate user i. This may commonly occur when the user does not know the persons being recommended especially when our system becomes very large. Based on we define the impact ranking of user i influenced by the feedback of users as follows:

ŕ (i) = ŕ(i, j)/n

which takes the feedback of all users into consideration. Finally, the original impact ranking vector R calculated

from the friend-matching graph is updated as follows:

r = ɖr + (1 -ɖ) ŕ

Where ɖ is named as confidence factor and. The final impact ranking vector considers both the influence of friend-matching graph and the feedback from users. When ɖ > 0:5, the friend-matching graph dominates the impact ranking, however, when ɖ < 0:5, users' feedback will significantly affect the impact ranking. In this way, the system takes users' feedback into consideration to improve the accuracy of future recommendations.

3. CONCLUSION

This System presents the design and implementation of Friend Recommendation System based on Semantic. Different techniques are proposed which are not used previously, so this system extracts life styles from user-centric data collected from the user's activities and recommendation potential friends to users if they share similar life styles. The results will prove that the recommendation accurately react the preferences of user in choosing friends.

4. ACKNOWLEDGMENT

We gratefully acknowledge H.O.D of computer engineering department of our college for their kind support for this project. We also thank our project guide and co-guide for highlighting our path and their gracious guidance. In last we like to thank all the friends who had given some valuable contribution for this system.

5. REFERENCES

[1] Zhibo Wang, Jilong Liao, Qing Cao,Hairong Qi and Zhi, "Friendbook: A Semantic- based Friend Recommendation System for Social Networks", IEEE Transaction on Mobile Computing,2014.

[2] Amazon. http://www.amazon.com/(obseravation of ratings).

[3] Net_x. https://signup.netix.com/(obseravation of ratings).

[4]Rottentomatoes.http://www.rottentomatoes.com/(obseravation of ratings).

[5] K. Farrahi and D. Gatica-Perez, "Probabilistic mining of sociogeo graphic routines from mobile phone data",Selected Topics in Signal Processing, IEEE Journal of, 4(4):746-755, 2010.

[6] W. H. Hsu, A. King, M. Paradesi, T. Pydimarri, and T. Weninger ,"Collaborative and structural recommendation of friends using weblog-based social network analysis",Proc. of AAAI Spring Symposium Series, 2006.

[7] T. Huynh, M. Fritz, and B. Schiel,"Discovery of Activity Patterns using Topic Models" ,Proc. of UbiComp, 2008.

[8] E. Miluzzo, C. T. Cornelius, A. Ramaswamy, T. Choudhury, Z. Liu, and A. T. Campbell,"Darwin Phones: the Evolution of Sensing and Inference on Mobile Phones", Proc. of MobiSys, pages 5-20, 2010.

[9] E. Miluzzo, N. D. Lane, S. B. Eisenman, and A. T. Campbell,"Cenceme-Injecting Sensing Presence into Social Networking Applications",Proc. of EuroSSC, pages 1-28, October 2007.

[10] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava,"Using Mobile Phones to Determine TransportationModes",ACM Transactions on Sensor Networks (TOSN), 6(2):13, 2010.

[11] L. Bian and H. Holtzman,"Online friend recommendation through personality

matching and collaborative _ltering", Proc. of UBICOMM, pages 230-235, 2011.

[12] J. Kwon and S. Kim, "Friend recommendation method using physical and social context" ,international Journal of Computer Science and Network Security, 10(11):116-120, 2010.