IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 10, Issue 8, August 2020, ||Series -I|| PP 34-40

Friend Recommendation System for Social Network a Semantic Approach

[1]NAKKA TEJALATHA, [2]KUCHIPUDI. RAJKAMAL,
[3]Dr. GOPISETTI GURU KESAVA DASU

[1] M.Tech Scholar, Department of Computer Science Engineering,
[2]Assist Professor Eluru College of engineering and technology, ap, india.
[3]Professor and Hod Department of Cse, Eluru College of Engineering and Technology, Ap, India.
Received 12 August 2020; Accepted 28 August 2020

Abstract: 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 material life.In this report, we present Friendbook, a novel semantic-based friend recommendation system for social networks, which recommends friends to users based on their lifestyles instead of social graphs. By advantage of sensor-rich smartphones, Friendbook 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 lifestyles have high similarity. Inspired by text mining, we model a user's daily life as life documents, from which his/her life styles are taken out by using the Latent Dirichlet Allocation algorithm. We further propose a similarity metric to evaluate the similarity of life styles between users, and calculate users' impact in terms of lifestyles with a friend-matching graph. Upon getting a request, Friendbook returns a list of people with highest recommendation scores to the query user.Finally, Friendbook integrates a feedback mechanism to further amend the recommendation accuracy. We have implemented Friend book on the Android-based smartphones, and assessed its performance on both small-scale experiments and large-scale simulations. The outcomes indicate that the recommendations accurately reflect the preferences of users in choosing allies.
Index Terms—Friend recommendation, mobile sensing, social nets, life style

I. INTRODUCTION

Twenty years ago, people typically made friends with others who live or go near to themselves, such as neighbors or fellow workers.We visit friends made through this traditional fashion as G-friends, which stands for Ge- graphical location-based friends because they are influ- enced by the geographical distances between each other.With the rapid improvements in social networks, services such as Facebook, Twitter and Google+ have provided us revolutionary ways of gaining friends. According to Facebook statistics, a user receives an average of 130 friends, perhaps larger than any other time in history [2].

One challenge with existing social networking services is how to commend a dependable friend to a user. For example, Facebook relies on a social link analysis, among those who already share common friends and recommends symmetrical users as possible friends. Unluckily, this approach may not be the most appropriate based on recent sociological findings [16], [27], [29], [30].Agreeing to these studies, the conventions to group people together include: 1) habits or lifestyle; 2) attitudes; 3) tastes; 4) moral standards; 5) economic level; and 6) people they already experience. Apparently, rule #3 and rule #6 are the mainstream factors considered by existing recommendation systems. Rule #1, although probably the most intuitive, is not widely used because users' life styles are difficult, if not impossible, to capture through web actions. Rather, life styles are usually closely correlated with daily routines and activities. So, if we could collect information on users' daily routines and activities, we can exploit rule #1 and recommend friends to people based on their similar life styles. This recommendation mechanism can be deployed as a standalone app on smartphones or as an add-on to existing social network frameworks. In both cases, Friendbook can help mobile phone users find friends either among strangers or within a certain group aslongastheysharesimilarlifestyles.

In our daily lives, we may experience hundreds of natural processes, which form meaningful sequences that form our lives. In this paper, we use the word activity to specify- equally refer to the actions taken in the order of seconds, such as "sitting", "walking", or "typing", while we use the phrase life style to refer to higher-level abstractions of daily lives, such as "office work" or "shopping". For illustration, the "shopping" life style mostly consists of the "walking" activity, but may likewise curb the "standing" or the "sitting" activities.

To model daily lives properly, we take out an analogy between people's daily lives and documents, as depicted in Figure 1. Previous research on probabilistic topic models in text mining has treated documents as mixturesof topics, and topics as mixtures of words [10].

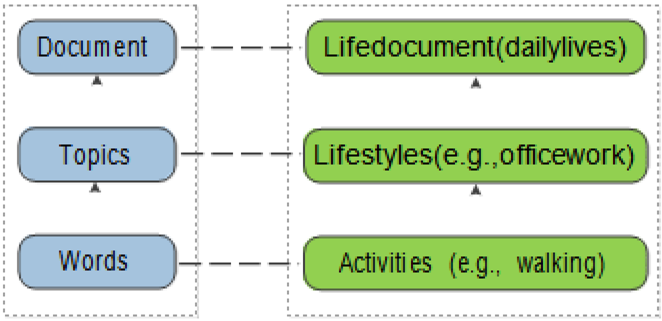


Fig. 1: An analogy between word documents and people's daily lives.

Prompted by this, similarly, we can handle our day-to-day lives (or life documents) as a mixture of life styles (or topics), and each lifestyle as a miscellany of actions (or row). Take note here, fundamentally, we represent daily lives with "life documents", whose semantic meanings are reflected through their issues, which are lifestyles in our field.Just like words serve as the basis of documents, people's activities naturally serve as the primitive *vocabulary* of these lifedocuments.

Our proposed resolution is also moved by the recent advances in smartphones, which have become more and more popular in people's lives. These smartphones (e.g., iPhone or Android-based smartphones) are fitted with a deep set of embedded sensors, such as GPS, Ac- accelerometer, microphone, gyroscope, and a camera. Therefore, a smartphone is no longer just a communication device, but also a powerful and environmental reality sensing platform from which we can extract rich context and capacity-aware data. From this perspective, smartphones serve as the ideal platform for sensing daily routines from which people's lifestyles could be found.

In malice of the powerful sensing capabilities of smart- phones, there are still multiple challenges for extracting users' life styles and recommending potential friends based on their similarities. First, how to automatically and accurately discover life styles from noisy and heterogeneous sensor data? Second, how to measure the similarity of users in terms of life styles? Third, who should be recommended to the user among all the friend candidates? To address these challenges, in this paper, we present Friendbook, a semantic-based friend recommendation system based on sensor-rich smartphones. The contributions of this work are summarized as follows:

· To the best of our knowledge, Friendbook is the first friend recommendation system exploiting a user's life style information discovered from smartphone sensors.

Inspired by achievements in the study of text mining, we model the daily lives of users as life documents and apply the probabilistic topic model to extract life style information about users.

We offer a unique similarity metric to character- is the similarity of users in terms of lifestyles and then construct a friend-matching graph to recom- mend friends to users based on their lifestyles.

We incorporate a linear feedback mechanism that ex- pilots the user's feedback to improve recommender- tonal accuracy.

We conduct both small-scale experiments and large- scale simulations to assess the functioning of our organization. Experimental results show the strength of our organization.

II. RELATED WORK

Recommendation systems that attempt to suggest items (e.g., music, movie, and books) to users have become more and more popular in recent years. For example, Amazon [1] recommends items to a user based on items the user previously visited, and items that other users are looking at. Netflix [3] and Rotten Tomatoes [4] recommend movies to a user based on the user's previous ratings and viewing habits. Loosely talking, existing friend recommendation in social networking systems, e.g., Facebook, LinkedIn and Twitter, recommend friends to users if, according to their social relations, they share mutual friends.

Meanwhile, another recommendation mechanism has also been suggested by researchers. For example, Bian andHoltzman [8] presented MatchMaker, a Collabora- tive filtering friend recommendation system based on personality matching. Kwon and Kim [20] suggested a friend recommendation method using physical and societal setting. [32] recommended geographically related friends in social networking by combining GPS data and social network structure. Hsu et al. [18].

Activity recognition serves as the basis for extract- in high-level daily routines (in tight correlation with life styles) from low-level sensor data, which has been widely studied using several types of wearable sensors. Lester et al. [21] useddata from wearable sensors to recognize activities based on the Hidden Markov Model (HMM). Li et al. [22] recognized static postures and dynamic transitions by using accelerometers and gyros.

The rise of smartphones enables activity recognition using the deep set of sensors on the smart phones. Reddy et al. [26] used the built-in GPS and the accelerometer on the smartphones to detect the transportation modality of an person.CenceMe [24] used multiple sensors on the smartphone to capture user's activities, state, habits and surroundings. SoundSense [23] used the mike on the smartphone to recognize general sound types (e.g., music, voice) and discover user specific sound effects.EasyTracker [7] used GPS traces collected from smart- phones that are installed on transit vehicles to determine routesserved,locatestops,andinferschedules.

III. SYSTEMOVERVIEW

In this segment, we present a high-level overview of the Friendbook system. Form 2 depicts the system designer- Tour of Friendbook which adopts a customer-server mode where each node is a smartphone carried by a user and the servers are data centers or clouds.

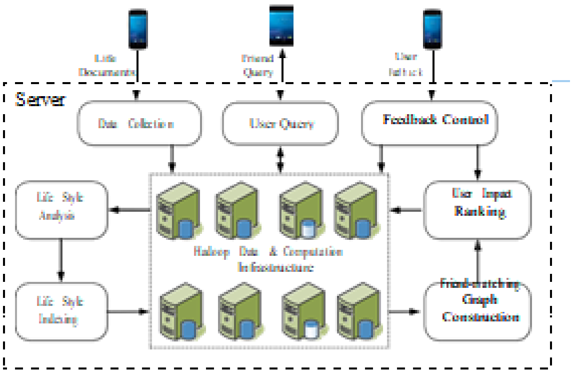


Fig. 2: System architecture of Friendbook.

On the customer side, each smartphone can record data about its user, perform real-time activity recognition and report the generated life documents to the waiters. It is worth mentioning that an offline data collection and training phase is required to make an appropriate activity classifier for real-time activity recognition on smartphones. We dropped three months on collecting raw data of 8 volunteers for building a large training data set. As each user typically generates around 50MB of raw data each day, we choose MySQL as our low grade data warehousing platform and HadoopMapReduce as our computational infrastructure. As a user con- tinually uses Friendbook, he/she will accumulate more and more body processes in his/her life documents, based on which, we can discover his/her life styles using the probabilistic topic model.

IV. QUERY AND FRIEND RECOMMENDATION

Before a user initiates a request, he/she should have accumulated enough activities in his/her life documents for efficient life styles analysis. The period for collecting data usually takes at least one day. Longer time would be expected if the user wants to get more satisfied friend recommendation results. After receiving a user's request (e.g., life documents), the server would extract the user's life style vector, and based on which recommend friends

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 $\mathbf{r}$

The recommendation results are exceedingly dependent on users' preference. Some users may prefer the arrangement to recommend users with high impact, while some users may want to know users with the most similar life styles. It is as well possible that some users want the organization to recommend users who hold high impact and also similar lifestyles to them. To better characterize this require- ment, we suggest the following metric to facilitate the good word,

$$R_i(j)=\beta S(i,j)+(1-\beta)r_j\kappa(1)$$

where $R_i(j)$ is the recommendation score of user j for the query user i, $S(i, j)$ is the similarity between user i and user j, and r_j is the impact of user j. β [0, 1] istherecommendationcoefficientcharacterizingusers' preference. *κ is introduced to make S (I, j) and rug in the same order of magnitude, which can be more or less set to n/10, where n is the number of users in the organization.* When β = 1, the recommendation is solely founded on the similarity; when β = 0, the recommendation is solely based on the impact ranking.
With the metric in Eq. 15, our recommendation mecha- nism for finding the most appropriate friends to a query user is identified as follows. For a query user I, the host calculates the recommendation scores for all the users in the system and sorts them in the descending order according to their recommendation scores. The top p users will be turned back to the query user myself. The parameter p is an integer and can be determined by the querying user.The complexity of our recommendation mechanism is O (n) since it matches all users in the organization, where n is the overall number of users in the organization.

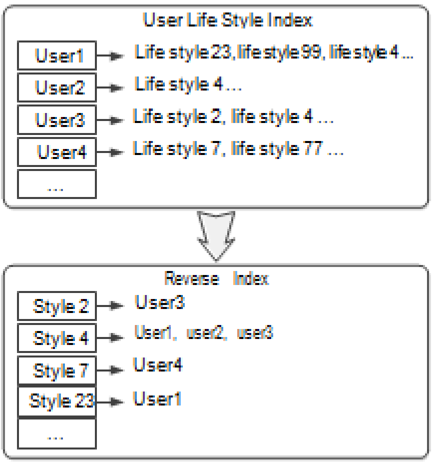


Fig. 2: Illustration of the reverse index table.

We need the reverse index table, before calculating a recommendation score for each user, the server first picks up all the users having overlapping life styles with the query user and sets the similarities of rest users to the query user to 0. The host then checks all the users to calculate their recommendation scores. Although the complexity is still O (n), we can note that the reverse index table reduces the computation overhead, the advantage of which is considerable when the organization is in large-scale.

The pseudo code of the friend recommendation mechanism is shown in Algorithm 2.

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 F_i.

F_i ,Q $\emptyset$ $\leftarrow \emptyset$
extracts i's life style vector $\mathbf{L}_i$ using the LDA algo- rithm.
for each life style z_k the probability of which in $\mathbf{L}_i$ is not zero **do**
  put users in the entry of z_k into Q
end for
for each user j / Q **do**
  $S(i, j) \leftarrow 0$
end for
for each user j in the database **do**
  $R_i(j) = \beta S(i, j) + (1 - \beta)r_j\kappa$
end for
sort all users in decreasing order according to $R_i(j)$

Privacy is really important particularly for users who are sensitive to information leakage. In our design of Friendbook, we as well studied the privacy issue and the existing system can provide two levels of secrecy protection. First, Friendbook protects users' privacy at the information layer. Instead of uploading raw data to the servers, Friendbook processes raw data and sorts them into activities in real-time. The recognized active- ities are labeled by integers. Second, Friendbook protects users' privacy at the life pattern level. Instead of telling the similar animation styles of users, Friendbook only shows the recommendation scores of the recommended friends with the users. With the recommendation score, it is nearly impossible to infer the life styles of recommended friends.

V. EVALUATION

In this part, we demonstrate the performance evaluation ofFriendbook on both small-scale field experiments and large-scale simulations.

5.1 Evaluation using RealData

We first evaluate the performance of Friendbook on small-scale experiments. Eight volunteers help con- tribute data and evaluate our system. Table 1 demon- strates the profession of these users. Most of them are students, while the rest includes a businessman, an office worker, and a waitress. Each volunteer carries a Nexus S smartphone with Friendbook application installed in forward motion.

They are needed to initiate the application after they awaken up and wrench it away before they get to bed. Apart from this, we do not visit any extra demand on the exercise of the smartphone. For instance, we do not want them to take the smartphone all the time during the day or attach the smartphone to some special roles of the physical structure.

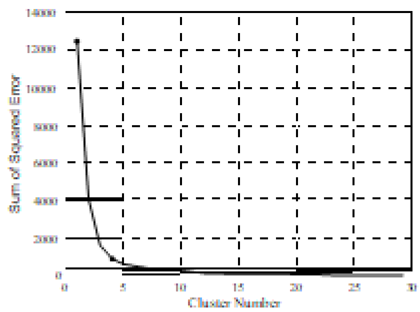
It is worth mentioning that some of the eight users are already friends before experiments, but some of them are not. In fact, some strangers within the group become friends later. However, strangers living far away from each other do not become friends, although they choose each other as a friend at the friend recommended- turn phase.

TABLE 1: Profession of users

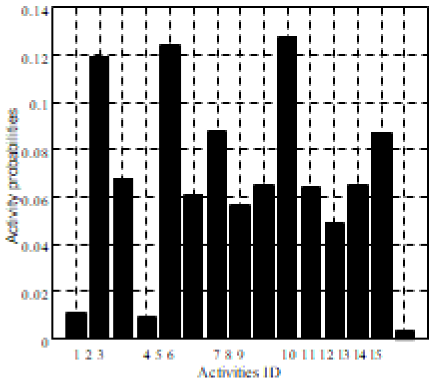
User ID	1	2	3	4	5	6	7	8
Student	√	√	√		√		√	
Waitress				√				
Office Worker						√		
Businessman								√

5.1.1 ActivityClassification

Image 3 shows the classification results using the K- means clustering algorithm on the data gathered from the 3 users for a period of three months. Feature vectors instead of raw data are used for classified- tion. Each feature vector consists of 7 attributes, f = [Tc, accx, Accy, arcs, gyrx, gray, gyrz] (see Section 4.2), and we extract feature vectors every 60 minutes. As depicted in Figure 8 (a), the sum of squared error drops quickly when the cluster number K increases from 1 to 10



(a) Classificationresults



(a) Activitydistribution

Fig. 3: Classification performance using the K-means clustering.

And so does not switch too much after 15. This means that the daily lives of the volunteers are roughly composed of 15 different types of actions. Although we can use additional activities to characterize their daily lives at a finer scale, we find 15 an appropriate compromise as the most important natural processes are already taken. Other activities occur rarely and cannot considerably affect the friend recommendation results. Thus, we use K = 15 as the number of activities in Friendbook.

Form 3 (b) shows the distribution of 15 active- ities. Most of activities have the probability of about 6%, and three activities have the probability of larger than 10%, and the remaining three activities have the probability of less than 2%. Corresponding to each ac- tivity, a centroid feature vector is calculated by using the K-means clustering algorithm. These 15 centroids are distributed to each smartphone so that it can perform real-time on-board activityrecognition.

VI. CONCLUSION

In this report, we presented the design and implementation of Friendbook, a semantic-based friend recommend- dating scheme for social networks. Different from the friend recommendation mechanisms relying on social graphs in existing social networking services, Friend- book extracted life styles from user-centric data collected from sensors on the smartphone and recommended potential friends to users if they share similar life styles.

REFERENCES

[1]. Amazon.http://www.amazon.com/.
[2]. Facebook statistics. http://www.digitalbuzzblog.com/facebook-statistics-stats-facts-2011/.
[3]. Netfix.https://signup.netflix.com/.
[4]. Rotten tomatoes.http://www.rottentomatoes.com/.
[5]. G.R.Arce.NonlinearSignalProcessing:AStatisticalApproach.
[6]. *John Wiley & Sons*, 2005.
[7]. B. Bahmani, A. Chowdhury, and A. Goel. Fast incremental and personalized pagerank. *Proc. of VLDB Endowment*, volume 4, pages 173-184,2010.
[8]. J. Biagioni, T. Gerlich, T. Merrifield, and J. Eriksson. EasyTracker: Automatic Transit Tracking, Mapping, and Arrival Time Prediction UsingSmartphones.*Proc.ofSenSys*,pages68-81,2011.
[9]. L. Bian and H. Holtzman. Online friend recommendation through personality matching and collaborative filtering. *Proc. of UBI- COMM*, pages 230-235,2011.
[10]. C. M. Bishop. Pattern recognition and machine learning. *Springer New York*,2006.
[11]. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent DirichletAllocation.
[12]. *Journal of Machine Learning Research*, 3:993-1022, 2003.
[13]. P. Desikan, N. Pathak, J. Srivastava, and V. Kumar. Incremental page rank computation on evolving graphs. *Proc. of WWW*, pages 1094-1095,2005.
[14]. N.EagleandA.S.Pentland.RealityMining:SensingComplexCo- cial Systems. *Personal Ubiquitous Computing*, 10(4):255-268, March 2006.
[15]. K. Farrahi and D. Gatica-Perez. Probabilistic mining of socio- geographic routines from mobile phone data. *Selected Topics in SignalProcessing,IEEEJournalof*,4(4):746-755,2010.
[16]. K. Farrahi and D. Gatica-Perez. Discovering Routines from Largescale Human Locations using Probabilistic Topic Models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(1), 2011.
[17]. B. A. Frigyik, A. Kapila, and M. R. Gupta. Introduction to the dirichlet distribution and related processes.

Department of Electrical Engineering, University of Washignton, UWEETR-2010-0006,2010.

Authors

NAKKA TEJALATHAis Pursuing M.tech(CSE) from ELURU COLLEGE OF ENGINEERING AND TECHNOLOGY,WEST GODAVARI,AP, INDIA

KUCHIPUDI. RAJKAMALis working asAssist PROFESSOR in ELURU COLLEGE OF ENGINEERING AND TECHNOLOGY, AP, INDIA.

Dr. GOPISETTI GURU KESAVA DASUBE(CSE), ME(CSE), PHD(CSE) PROFESSOR and HOD in the DEPARTMENT OF CSE, ELURU COLLEGE OF ENGINEERING AND TECHNOLOGY, AP, INDIA

NAKKA TEJALATHA, et. al. "Friend Recommendation System for Social Network a Semantic Approach." *IOSR Journal of Engineering (IOSRJEN)*, 10(8), 2020, pp. 34-40.