Learning Methods for Similarity Handling in Phonebook-centric Social Networks

Péter Ekler

Department of Automation and Applied Informatics
Budapest University of Technology and Economics, Hungary
peter.ekler@aut.bme.hu

Tamás Lukovszki

Faculty of Informatics
Eötvös Loránd University, Hungary
lukovszki@inf.elte.hu

Abstract: The capabilities of mobile phones enable them to participate in popular social network applications. The phonebooks in the mobile devices represent social relationships, that can be integrated in the social networks. Following we refer to this solution as phonebook-centric social networks. Such a network allows synchronization between the phonebook of the mobile phone and the social network. By the synchronization the goal is to identify the persons listed in the phonebook and the network, it means to find similar entries and keep the data consistent. The detection of those similar members is based on the similarity of the personal data, e.g., similar name, same phone number, e-mail, etc… We implemented a phonebook-centric social network, called Phonebookmark. In this paper we present the semi-automatic similarity handling of Phonebookmark. The similarity detection algorithm collects potential similarities and assigns a score to them. The score expresses how likely a phonebook contact and another member of the network identify the same person. Then the user can decide whether to accept or ignore the proposed similarity. During the first 9 month period users have accepted more than 90% of the proposed similarities which encouraged us to improve the algorithm further. We extended the algorithm with two learning methods. The first extension enables to learn similar names like Katharine and Kate and later this knowledge is used to detect similar names more efficiently. The second extension learns from the similarity resolving decisions of the users and based on this it maintains conditional probabilities of accepting a similarity if certain attributes of the personal data are equal. The extended algorithm still detects at least the same amount similarities but with better accuracy.

Keywords: social networks, phonebook, similarities, mobile phones

1 Introduction

The popularity of social networks was noticeable in the last few years. Several social networks appeared and attracted thousands or even millions of users. The online social networks Facebook [3] and Myspace [4] are among the top ten visited websites on the Internet [1]. The basic idea behind these networks was that users can maintain social relationships on these networks. A social network is basically a social structure consisting of nodes that generally correspond to individuals or organizations. These nodes are connected by different type of relations. Users of social networks are able to share personal detail about themselves, talk in forums, share photos or entire galleries, play games, etc. Basically it is an environment created by the people who are using it.

Mobile phones and mobile applications are another hot topic nowadays. Both hardware and software capabilities of mobile phones have been evolving in the last decades. Yet support of mobile devices is generally marginal in most social networks, it is limited to photo and video upload capabilities and access to the social network using the mobile web browser. However, if we consider the phonebook in our mobile phone, we realize that basically it is a small part of a social network because every contact in our phonebook has some kind of relationship to us. Given an implementation that allows us to upload as well as download our contacts to and from the social networking application, we can completely keep our contacts synchronized so that we can also see all of our contacts on the mobile phone as well as on the web interface. In the rest of this paper we refer to this solution as a *phonebook-centric social network*.

One of the key features of phonebook-centric social networks is that the phonebook of the mobile phone is automatically updated with the latest information provided by the friends of its owner. This means also that the persons in ones phonebook also get the latest information about one automatically, so there is no need to notify them one by one if the phone number changes for example. In addition to that, the private contacts are also uploaded to the phonebook-centric social network. These contacts are not visible to other members of the site. Having all of the contacts in the system has the following benefits:

- The contacts can be managed (list, view, edit, call, etc.) from a browser.
- The service notifies the user if duplicate contacts are detected in its phonebook and warns about it.
- The contacts are safely backed up in case the phone gets lost.
- The contacts can be easily transferred to a new phone if the user replaces the old one.
- The phonebook can be shared between multiple phones, if one happen to use more than one phone.

- It is not necessary to explicitly search for the friends in the service, because it notices if there are members similar to the contacts in the phonebooks and warns about it.

Phonebookmark is a phonebook-centric social network implementation by Nokia Siemens Networks. We took part in the implementation and before public introduction it was available for a group of general users from April to December of 2008. It had 420 registered members with more than 72000 private contacts, which is a suitable number for testing the handling of similarities. During this period we have collected different type of data related to the social network.

However phonebook-centric social networks raises problems which have not been expected in general social networks before. One of the biggest challenge is detecting and handling similarities. Similarities appear in the system when one of the phonebook contacts and a member of the system have similar personal data, e.g. same name, phone number, e-mail, etc... In this paper we present an algorithm for detecting and resolving similarities on a user-friendly and efficient way.

The rest of the paper is organized as follows. Section 2 summarizes related work in the field of mobile phone enabled social networks. Section 3 defines the term of *phonebook-centric social network* and the related terms. Section 4 demonstrates the static similarity detecting and handling algorithm and Section 5 shows learning method extensions to the proposed algorithm. Finally Section 6 concludes the paper and summarizes the results.

2 Related Work

Nowadays, the number of social network users is increasing, thus the efficient implementation of these networks is an important research area. Newman et al. [5] describe some novel uniquely solvable models of the structure of social networks based on random graphs with arbitrary degree distributions. They give models both for simple unipartite networks such as acquaintance networks and bipartite networks such as affiliation networks. They compare the predictions of their models to data from a number of real-world social networks and find that in some cases, the models show high correlation with the data, whereas in others the correlation is lower, perhaps indicating the presence of additional social structure in the network that is not captured by the random graph.

Bakos et al. [2] have concluded that search engines generally lack the trust and level of personalization needed for recommendation systems to answer searches like: I need a reliable plumber close to my house. In order to achieve personalization, social relevance and an acceptable level of privacy, the search database itself needs to be personalized. One possible dimension of personalization is the social neighborhood of the searcher. In particular,

phonebook links represent a readily available infrastructure to create a peer-to-peer social network for socially relevant search. They have demonstrated their concept via a novel search engine algorithm for social networks that operates on S60 and uses SMS messages to communicate.

Nathan et al. [6] propose the Serendipity system that senses a social environment and cues informal interactions between nearby users who might know each other. Their system uses Bluetooth addresses to detect and identify proximate people and matches them from a database of user profiles. They show how inferred information from the mobile phone can augment existing profiles, and they present a novel architecture for investigating face-to-face interaction designed to meet various levels of privacy requirements.

In a social network nodes and links represent participants and their relationships, respectively. Tomiyasu et al. [8] have designed and implemented a query propagation mechanism and its applications to realize a social network composed by cellular phone users. In these applications, users can retrieve information on their friends or their friends' friends by propagating the query in the network. To propagate a query in a wide range and improve the query success ratio most users who receive the query must relay it to all their friends. However, this increases network traffic. In their paper they have proposed a query routing method to decrease the number of communication packets by using user profiles.

We investigated previously [7] how mobile devices can connect to a web-based social network. We outlined an architecture where mobile devices connect to a social network via web services. Additionally, we demonstrated this solution with a web-based social network application that offered all of its main functionality to a mobile client through web-service functions.

During the development period of *Phonebookmark*, we have observed other phonebook-centric social network solutions on the web. Zyb [11] and Plaxo [8] allow for synchronizing with mobile phones and managing the contacts using a web browser. Xing [10] has also mobile access, but focuses more on business relationships. Automatic similarity detection is missing from these systems though, thus there is no notification like in *Phonebookmark* when one of a user's phonebook contacts becomes (or already is) a user of the system.

The key difference between our current work and previous research is that the former social networking solutions do not allow mobile phones to become an integrated component in the social network. They do not fully exploit the fact that the phonebook of these devices is in itself an important part of the social network.

3 Phonebook-centric Social Networks

The functionality provided by the newer social networks is more and more interesting. In this investigation we focus on social networks which somehow involve mobile phones into their functionality. The reason is that the phonebook of the mobile phone represents some kind of social relationships between us and our contacts.

In a phonebook-centric social network (Figure 1) it is possible that one of our private contacts in our phonebook is similar to a member of the network, e.g. has the same name, phone number, etc… Following we will refer to this as *similarity*. A similarity detection algorithm enables more advanced functionality for social networks. Such an algorithm allows us to detect and resolve similarities in the network, and recommend possible relationships. In addition to that this algorithm enables also to recognize *duplications* in phonebooks.

Following we discuss first the simpler *phonebook-enabled social networks*, and then we derive phonebook-centric social networks from them. The difference is that phonebook-enabled social networks do not support similarity detection.

Definition 1. A *member* is a registered user of the social network. A member can log into the system, find and add acquaintances, upload and share information about themselves, write forum or blog entries, etc. They can upload their contact list to the social network and maintain a backup phonebook there. We denote the set of registered members by U_M.

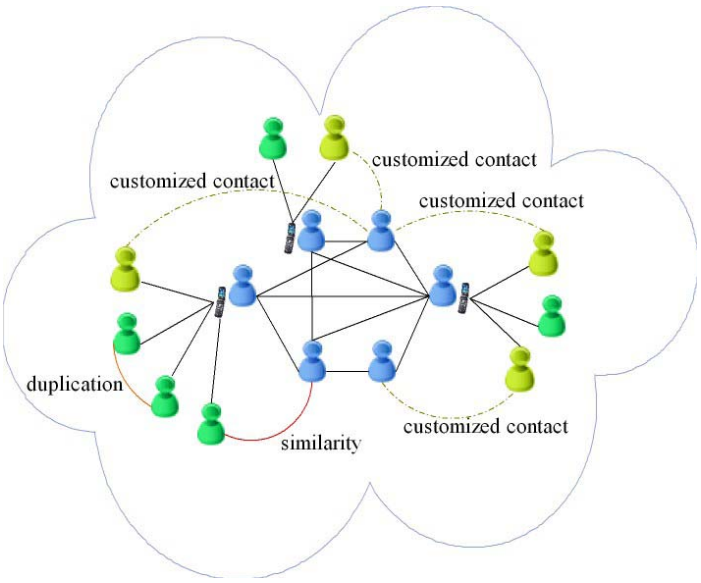

Figure 1
Structure of a phonebook-centric social network

Definition 2. A *private contact* corresponds to a phonebook entry of a member. Each member may have multiple private contacts. However, these private contacts are not shared between members. A private contact is transferred into the system when a member synchronizes his or her phonebook with the social network. We denote the set of private contacts in the phonebooks by U_{PC}.

In a phonebook-enabled social network the sets U_M and U_{PC} are disjoint sets. Relationships between members are represented by the edge set E_{MM} and relationships that a private contact belongs to a member are represented by the edge set E_{MPc}. Formally:

$$E_{MM} \subseteq \{(u_M, u'_M) : u_M, u'_M \in U_M, u_M \neq u'_M\}$$
$$E_{MPc} \subseteq \{(u_M, u_{PC}) : u_M \in U_M, u_{PC} \in U_{PC}\} \quad (1)$$

A phonebook-enabled social network is represented by a (directed) graph:

$$G_{PESN} = (U, E), U = U_M \cup U_{Pc}, E = E_{MM} \cup E_{MPc}, \quad (2)$$

In case of phonebook-centric social networks, where a similarity detection algorithm operates, if a member resolves a similarity between another member and a private contact a new *customized contact* appears.

Definition 3. A customized contact is created from a private contact when a member is similar to a private contact and the owner member of the private contact marks them as similar person. This way the owner can edit this contact in her or his phonebook but if the referred member changes her or his profile, the change will be propagated to the customized contact. However this propagation will take effect only if the owner member has not edited that specific profile detail yet. Following we will refer to this propagate mechanism as customization. The set of customized contacts is denoted by U_C.

The set E_S of edges indicate detected similarities between private contacts and members of the network and the set E_D of edges indicate (potential phonebook contact) duplications between private contacts of a member. Users can decide whether they accept or ignore the detected similarities and duplications. Formally:

$$E_S = \{(u_{Pc}, u_M) : u_M \in U_M, u_{Pc} \in U_{Pc}, (u_{Pc}, u_M) \notin E_{MPc}, \\ \exists (u_{Pc}, u'_M) \in E_{MPc}, \exists (u'_M, u_M) \in E_{MM}, u'_m \in U_M\}$$

$$E_D = \{(u_{Pc}, u'_{Pc}) : u_{Pc}, u'_{Pc} \in U_{Pc}, u_{Pc} \neq u'_{Pc}, \\ \exists ((u_{Pc}, u_M), (u'_{Pc}, u_M)) \in E_{MPc}, u_M \in U_M\} \quad (4)$$

If a user accepts a duplication one of the private contact is deleted. However if a user accepts a similarity between one of his or her private contact and a member, a new customized contact is created and the private contact is deleted. In addition to

that the E_S edge is deleted and a new E_{MC} edge is created. The the set of E_{MC} edges are connections between customized contacts and the referred members and the set of E_{MoC} edges represents connections between members and their customized contacts and. Formally:

$$\begin{aligned} E_{MC} &= \{(u_M, u_c) : u_M \in U_M, u_C \in U_c, \exists u'_M \in U_M, \\ & u'_M \neq u_M, (u_M, u'_M) \in E_{MM}, (u_M, u_C) \in E_{MoC}\} \\ E_{MoC} &= \{(u_{Mo}, u_c) : u_{Mo} \in U_M, u_C \in U_c, \exists u'_M \in U_M, \\ & u'_M \neq u_{Mo}, (u_{Mo}, u'_M) \in E_{MM}, (u_{Mo}, u_C) \in E_{MC}\} \end{aligned} \quad (5)$$

A phonebook-centric social network is represented by the following graph:

$$\begin{aligned} G_{PCSN} &= (U, E), U = U_M \cup U_{Pc} \cup U_C, \\ E &= E_{MPc} \cup E_{MoC} \cup E_{MC} \cup E_D \cup E_S \end{aligned} \quad (6)$$

The goal of the proposed similarity detecting and handling algorithm in this paper is to detect the relevant similarity edges (E_S) and allow users to select the right similarity easily if more than one member is similar to a private contact.

4 Similarity Handling Algorithm

In this section first we introduce the static version of the proposed similarity detecting algorithm which was used originally in Phonebookmark. After, we propose an extension to the algorithm with learning methods in Section 5.

Phonebookmark provides a semi-automatic similarity detecting and resolving mechanism. First it detects similarities and calculates a weight for them, which indicates how likely the corresponding phonebook contact and the member of the network identify the same person. Phonebookmark uses this weight to determine the proper order of multiple similarities (Figure 2).

The basic idea behind the static similarity detection algorithm is defining different types of matching criteria (e.g. last names are the same) and assigning weight values to these criteria. Based on these criteria we can define a limit above which we consider two persons similar or duplicated. Table 1 shows match terms and their weights, that we have used in the initial static algorithm. These values were formed by intuition after several measurements in Phonebookmark. The proper balance of these weights can be improved with learning methods, which will be discussed in the next section.

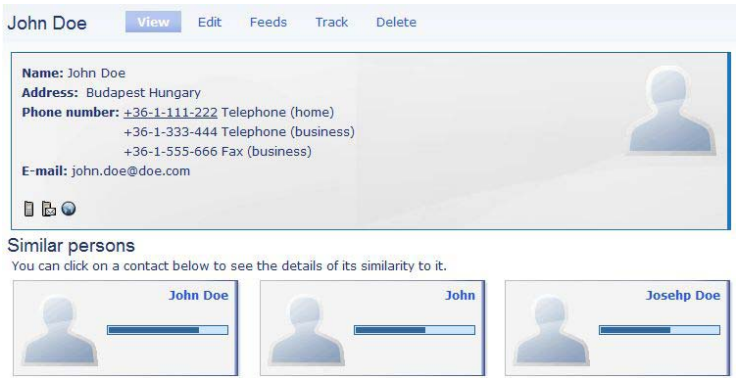

Figure 2
Multiple similarities

In Table 1 the *private phone number* stands for mobile phone number, home phone number, pager number, etc., while *public phone number* stands for fax number, workplace number, etc.

Furthermore the *different birthday* term has a negative value. It means that if two persons have different birthdays then the algorithm decreases the similarity value because it is a relevant difference.

Table 1
Matching terms and weights

Fields	Weight
Similar first and last name	30
Same private phone number	10
Same public phone number and first name	10
Same e-mail address	15
Different birthday	-40

After a detected similarity is being accepted by the user, Phonebookmark provides a user interface where the details of the two people can be merged. Here the user can choose whether to resolve or ignore the similarity, which is the base of the semi-automatic behavior (Figure 3).

During the operational period of Phonebookmark, the similarity detecting algorithm found about 1200 similarities and users have resolved more than 90 percent of these, which is an encouraging number.

Figure 3
Semi-automatic similarity resolution

5 Extending the Algorithm with Learning Methods

In this section we present two extensions of the similarity detection algorithm. The first extension enables to learn similar names like Katharine and Kate. This knowledge is used to detect similar names more efficiently. The second extension learns from the similarity resolving decisions of the users. It maintains conditional probabilities of accepting a similarity if certain attributes of a phonebook contact and a member are equal.

5.1 Learning Similar Names

The learning algorithm is able to detect similar names like Joe and Joseph which improves the probability of detecting similarities even if people have entered names in different ways. We have implemented the similar name handling by storing similar names in a database (see Table 2). When people use the system and accepts a similarity then the system checks whether the corresponding phonebook contact and the network member have the same first name. If not, an entry is stored in the similar names table indicating that the two names may be similar because they were resolved by the users manually.

Table 2
Similar names table

first name 1	first name 2	count
Joe	Joseph	10
Katharine	Kate	7
Samantha	Sam	2

The system also counts how many times these similar but different first names lead to an accepted similarity. When this reaches a certain limit then the similarity detection algorithm will handle these first names as the same first name.

5.2 Learning Weights to Match Terms

During the first operational period of Phonebookmark we have used the fixed weights introduced in Table 1. After this period we have studied the data and based on that we have extended the algorithm with a learning method which modifies the weights dynamically based on decisions of the users.

When we test a private contact and a member for proposing a similarity, we define the following events:

- M_1: they have the same first name and last name,
- M_2: they have the same private phone number,
- M_3: they have the same public phone number,
- M_4: they have the same e-mail address,
- M_5: they have the same birth day.

Let $M=\{M_1,M_2,M_3,M_4,M_5\}$. Further, we refer to the event that the private contact and the member lead to an accepted similarity as S.

Based on the frequency of the above events in the past, for each subset $H \subseteq M$, we maintain the conditional probability $\Pr[S|H]$, i.e. the conditional probability that the similarity becomes accepted if the set of events H occurs. Table 3 shows some of the conditional probabilities calculated from the Phonebookmark database after the operational period.

Table 3
Conditional probabilities calculated after the first operational period of Phonebookmark

H	$\Pr[S \mid H]$
$\{M_1,M_2,M_3,M_4,M_5\}$	1
$\{M_1,M_2,M_3,M_4\}$	0.9784
$\{M_1,M_3,M_4,M_5\}$	0.9897
$\{M_1,M_2,M_4,M_5\}$	0.9913
…	
$\{M_1,M_2,M_4\}$	0.9961
$\{M_1,M_2,M_5\}$	0.9090
…	
$\{M_2\}$	0.8730
$\{M_1\}$	0.8794

For each subset $H \subseteq M$, Pr[$S|H$] can be calculated as follows:

$$\Pr[S \mid H] = \frac{\Pr[S \wedge H]}{\Pr[H]} = \frac{\frac{n_{SH}}{(|U_M|-1)|U_{PC}|}}{\frac{n_H}{(|U_M|-1)|U_{PC}|}} = \frac{n_{SH}}{n_H} \tag{7}$$

In (7) n_{SH} is the number of cases where the events of H occur and the similarity has been accepted and n_H is the number of all cases where the events of H occur. $(|U_M|-1)U_{PC}$ is the number of private contact and member pairs that may define a similarity in the phonebook-centric social networks.

In our phonebook-centric social networks, for each $H \subseteq M$, the values n_{SH} and n_H can be initialized from the collected data during the operational period of Phonebookmark. After that, when the system is used actively these values, and thus the conditional probabilities can be dynamically maintained very efficiently based on the decisions, whether the users accept or reject the proposed similarities. We obtain the following theorem:

Theorem 1: After accepting or rejecting a proposed similarity, the overall time for maintaining the conditional probabilities is O(1).

Proof: If a user accepts a proposed similarity, where the events of H occur, then the values n_{SH} and n_H both increase by one. Therefore,

$$\Pr[S \mid H] = \frac{n_{SH}+1}{n_H+1}. \tag{8}$$

For all $H' \subseteq M$, $H' \neq H$, the values $n_{SH'}$ and $n_{H'}$ and the probability Pr[$S|H'$]do not change. Thus, we have to modify the value of two variables, which can be done in constant time.

If the user rejects the proposed similarity where the events of H occur, then n_H increases by one and n_{SH} does not change. Therefore,

$$\Pr[S \mid H] = \frac{n_{SH}}{n_H+1}. \tag{9}$$

For all $H' \subseteq M$, $H' \neq H$, the values $n_{SH'}$ and $n_{H'}$ and the probability Pr[$S|H'$] do not change. Thus, we have to modify the value of one variable, which can be done in constant time. □

Conclusions

Social networks that handle mobile devices have several interesting research implications. Considering the contacts in the phonebooks as social relationships leads to the concept of phonebook-centric social networks. Such a network

provides a synchronization mechanism between the phonebook of the mobile phone and the social network. By the synchronization the goal is to identify the persons listed in the phonebook and the network, it means to find similar entries and keep the data consistent. We have implemented such a network, called Phonebookmark. Phonebookmark is in internal use since April 2008. Currently, it has around 400 users with more than 70,000 private contacts.

In this paper we have focused on similarity handling and resolving. We have introduced a static, weight-based algorithm for detecting similarities and duplications along with a set of useful methods for similarity resolving and treating multiple similarities. Our measurements in Phonebookmark showed that using the introduced algorithm the rate of the detected false similarities is below 10 percent. These results encouraged us to develop the algorithm further with learning methods. We have introduced a method for learning similar names. Furthermore, we have presented a method, which learns from the similarity resolving decisions of the users. It efficiently maintains conditional probabilities of accepting a similarity if certain attributes of a phonebook contact and a member are equal. Using these probabilities for weighting the proposed similarities increase the accuracy of the similarity detection.

References

[1] Alexa top sites. http://www.alexa.com/site/ds/top_sites, September 2009

[2] B. Bakos, L. Farkas, J. K. Nurminen. Phonebook Search Engine for Mobile p2p Social Networks. Databases and Applications, pp. 210-215, 2005

[3] Facebook social networking application. http://www.facebook.com, September 2009

[4] Myspace social networking application. http://www.myspace.com, September 2009

[5] M. E. J. Newman, D. J. Watts, S. H. Strogatz. Random Graph Models of Social Networks. Proceedings of the National Academy of Sciences, 2002

[6] N. Eagle, A. Pentland. Social Serendipity: Mobilizing Social Software. IEEE Pervasive Computing, 2005

[7] P. Ekler, H. Charaf. Investigating the Role of Mobile Devices in Social Networks. Microcad International Conference, March 2008

[8] Plaxo social networking application. http://www.plaxo.com, December 2008

[9] T. H. H. Tomiyasu, T. Maekawa, S. Nishio. Profile-based Query Routing in a Mobile Social Network. Mobile Data Management, MDM 2006, May 2006

[10] Xing social networking application. http://www.xing.com, September 2009

[11] Zyb social networking application. http://www.zyb.com, December 2008