An Anti-Turing Test: Social Network Friends' Recommendations

Iaakov Exman and Alex Krepch

Software Engineering Department, The Jerusalem College of Engineering – Azrieli,
POB 3566, Jerusalem, 91035, Israel

Abstract. A routine activity of social networks' servers is to recommend possible friends that one may know and stimulate addition of these people to one's contacts. An intriguing issue is how these recommendation lists are composed. This work investigates the main factors involved in the recommendation activity, in order to reproduce these lists including its time dependent characteristics. After a preliminary analysis of actual data collected from social networks, we propose relevant algorithms. Besides conventional approaches, such as friend-of-a-friend, two techniques of importance have not been emphasized in previous works: randomization and direct use of *interestingness* criteria. An automatic software tool to implement these techniques is proposed. Its architecture and implementation is discussed.

1 Introduction

Social networks are very prominent to what is meant nowadays by software, being one of the most visible content transformers. Among other aspects, social networks continuously offer lists of suggested friend candidates. This is one of the central activities of such networks.

In this work we analyze data retrieved from recommendation lists of large public social networks, to obtain important factors influencing the generation of such lists and respective algorithms of relevance. In particular we pay attention to the order of suggested candidates within a given list and the variability of the lists' composition along the time.

Recommendations lists are very intriguing. One can hardly believe how some of the candidates appear there. In this work we do *not* focus on candidates known to the recommendation receiver. Our attention is directed to previously unknown candidates, and how they are possibly selected from a large social network database.

We coin these lists a sort of anti-Turing test, since we ask ourselves to what extent we are stereotyped by a reduced number of variables. We rather look like people artificially disguised as computers, than real complex people in such a game.

The ultimate test for the validity of our analysis is the ability to reproduce in a time-dependent fashion the general characteristics of the recommendation lists one receives. To this end a software tool was proposed and is being continuously developed.

Exman I. and Krepch A..
An Anti-Turing Test: Social Network Friends' Recommendations.
DOI: 10.5220/0004641600550061
In *Proceedings of the 4th International Workshop on Software Knowledge* (SKY-2013), pages 55-61
ISBN: 978-989-8565-76-1

1.1 Related Work

Here we present a concise review of related work. Chen et al. [1] studied four recommendation algorithms:

1. *Content Matching* – closely related to finding documents of similar content;
2. *Content*-Plus-Link – adds to the content matching, the existence of a social link between the candidate and the recommendation receiver;
3. *Friend*-of-Friend – considers only social network structure;
4. *SONAR* – aggregates social relationship information from different public data sources (within IBM).

Their conclusion was twofold: algorithms based on social network information produced better-received recommendations and found more known contacts for users; algorithms using content similarity were stronger in discovering new friends.

Roth et al. [8] describe an *implicit social graph* and use it together with interaction-based affinity in suggesting friends. Huberman et al. [5] specifically analyze Twitter. They conclude that what really matters is a sparse and *hidden network of connections* underlying the declared set of friends and followers.

Golbeck and Hendler [3] investigated how trust information can be inferred from social network members not directly connected and integrated into applications, such as TrustMail, an email client. Tang et al. [9] deal with automatic labeling the "intensity" of social relationships, say "colleagues" or "friends".

Konstas and collaborators [7] deal with collaborative recommendation in social networks, using for instance, linear filtering.

Tools for various actions on social networks include the Referral Web by Kautz et al. [6]. Gross and Acquisti [4] refer to the problem of privacy in online social networks, in particular Facebook.

In the remaining of the paper we introduce recommendation factors of importance (section 2), describe a preliminary analysis of social networks data (section 3) provide kinds of relevant algorithms (section 4), overview the software architecture and implementation of our tool (section 5), and conclude with a discussion (section 6).

2 Recommendation Factors

Recommendation factors can be roughly classified into *contents* of the given person – either the candidate or the receiver of the recommendation – and *interactions* among two or more members of the social network, directly or indirectly.

2.1 Unary Content Variables

Unary content variables include among others:

- *Profession* – acquired in a certain institution; degree obtained;
- *Education institutions* – say high-school or universities;
- *Employer* – company, public service or other organizations;
- *Occupation* – may differ from the profession; rank in the organization;
- *Specific skills* – within the profession and/or occupation;

- *Languages spoken* –
- *Hobbies* – leisure activities;
- *Geographical location* – of residence and work; country, state, county, city;

Note that one could look at each one of such unary variables as sub-sets or ranges. For example, universities could be classified into categories – say ivy league.

2.2 Interaction Variables

Interaction can be direct, among people who know each other, such as:

- *Joint Publications* – co-authors of the same paper or book;
- *Exchanged Messages* – email, phone conversation, if data is available from providers;

Indirect interactions are possible also among people who are not mutual acquaintances:

- *Common Friends* – the person receiving the recommendation and the friend candidate have common friends; this is the widely discussed issue of *transitivity*;
- *Common Search topics* – again if data is available from providers of search engines.

3 Preliminary Analysis of Social Networks: Data Collected

We have collected data from pages of members[1] of large social networks, to make a preliminary analysis, pointing to novel recommendation approaches.

For each member page and time stamp, we collected samples containing the first 50 recommendations, with the values of the available variables. Conclusions were inferred from the analysis of values within and among samples along time, for given social networks.

Collected data are presented under the rubric of the conclusions inferred, in order to provide support for the conclusions.

3.1 Degree of Randomness

Our observations show three possible behaviors of variables within a sample:

- *Uniform* – very regular throughout the whole range;
- *Random* – difficult to recognize any obvious regularity;
- *Recognizable trend* – it is neither uniform nor random; one can recognize definite trends, to be discussed below and in the next sub-section.

For LinkedIN, variables with uniform behavior include first and foremost the network degree, and in some samples, the number of shared connections and the Boolean

[1] Data was collected with the consent of the respective page owners: the paper authors and their friends.

variable telling whether the candidate is known to the recommendation receiver. The network degree – the distance between the candidate and a friend of the receiver – is almost always "2nd", i.e. the receiver has a friend directly connected to the candidate, implying a systematic "friend-of-a-friend" policy. The number of shared connections in some samples may be almost constant with few exceptions. In certain samples the candidates are almost always "unknown" to the receiver, again with few exceptions.

Still for LinkedIN, a localized recognizable trend seems to be that the first candidates – say the first two candidates – differ from the uniformity of the above mentioned variables. The most important characteristic is that the candidates may be "known" to the receiver. This is a sort of stimulus to accept the suggestions. The other variables for these candidates may be quite random. Thus, the degree may be different from "2nd" and the number of shared connections may be any value. Besides the first candidates, other randomly placed candidates – in varying fractions of the distribution – may also be "known".

For Facebook, a recognizable trend for shared connections is a unimodal distribution, whose peak is at low values. A plot of the histogram of shared connections for a given sample is seen in Fig. 1.

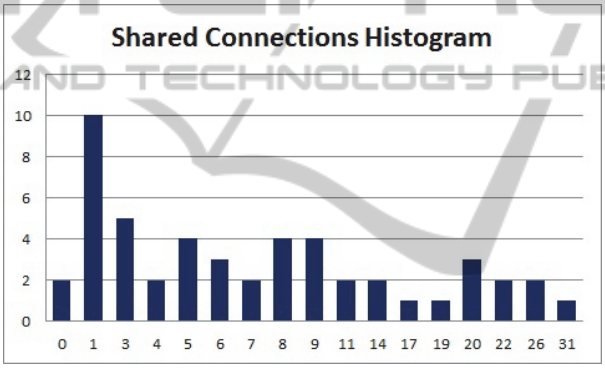

Fig. 1. Shared connections histogram – these are shown for a Facebook sample of suggested friend candidates. One can see the relative diversity of shared connection values in the horizontal axis. The unimodal peak is at the value 1 shared connection, with 10 candidates displaying this value.

3.2 Unexpectedness

There are variables in which there is a non-localized recognizable trend, which demands some deeper explanation.

For LinkedIN, one such variable is the geographical location of the workplace. If one takes the locations' distribution provided by the social network for the friends of the given member (the recommendation receiver), one finds that it is a very different distribution from that of the candidates in the recommendation list.

For instance, for a given sample the member friends are located in 6 countries, with the big majority in the country of the member himself. The candidates are located in 4 countries, with the majority still in the country of the member. But the

second country in terms of candidates has a very significant increase relatively disproportional to the member's friends – say about 30% instead of less than 10%.

A tentative explanation is as follows. The *unexpected* increase of this variable is due to the potential *interestingness* of this second country. On the one hand, it is not a negligible country in terms of the member friends, hinting to a potential increase. On the other hand it is *interesting*, i.e. its contribution to the distribution is unexpected given some of its characteristics, say country size, distance from the member's country, or international relations. The issue of *interestingness* will be further described in the next section.

3.3 Time Dependence

Recommendation lists change along time for each additional contact, but seem to change also when no contacts were added since the previous visit to the member's page in the network.

The changes of recommendation lists within a short time (say a few hours) may be very dramatic. Thus the important variable in this respect is not absolute time, but the fact that it is a new visit of the member's page.

4 Kinds of Algorithms

Algorithms for recommendation list generation may involve semantic considerations – e.g. friend-of-a-friend – as well as abstract mathematical operations involving linear or non-linear filtering. Coefficients in linearly weighted sums express the relative importance of factors involved and should usually be normalized. Non-linear expressions may impart different orders of magnitudes to the factors' importance.

4.1 Interestingness

Interestingness according to Exman [2] is a function of both relevance and unexpectedness. The latter quantities can be themselves expressed in various forms and also be combined in more than one way. The simplest way is just a multiplication:

$$\textbf{Interestingness} = \textbf{Relevance} \quad \textbf{Unexpectedness} \tag{1}$$

One form to express relevance is by calculating the match of a candidate to contents defining a domain. The respective unexpectedness is calculated by a measure of mismatch to the domain. Together, with a normalization factor NormF, this is written as:

$$\textbf{Interest} = \textbf{Match} * \textbf{Mismatch} / \textbf{NormF} \tag{2}$$

We advance the idea that interestingness for friend candidates, in which the domain is given by the recommendation receiver contents, is similar to that of any other content retrieval.

5 Software Architecture

A tool called *RECOMM* is being gradually developed to test the hypotheses that we propose. Here its architecture and implementation are concisely described.

Fig. 2 shows the following internal modules:

- *Randomize Inputs* – to select random candidates to add to the previous recommendation list; to determine their order in the recommendation list; to assign values to chosen variables;
- *Interestingness* – to increase the chances of candidates in detriment of others, based on their potential interest to the recommendation receiver;
- *Calculate Recommendation* – combining the above algorithms with the friend-of-a-friend alrgorithm;
- *Sorting & Threshold* – according to recommendation grades;
- *Decoration* – add a picture of the candidate and some possible additional features for display.

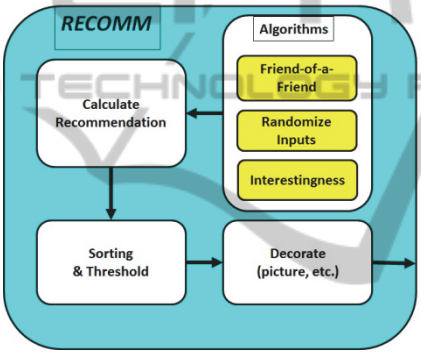

Fig. 2. *RECOMM* Architecture –modules displayed as upper states in the system statechart.

RECOMM is being implemented in C#.

6 Discussion

A framework was proposed for generating recommendations of friendship candidates in a given social network. The framework contains the important variables for given social networks, recommendation algorithms and a set of controls to output a recommendation list.

6.1 Validation

Validation of *RECOMM* output is performed against actual recommendation lists of specific social networks, e.g. LinkedIN or Facebook, for data obtained from members of these networks.

Preliminary collection of data has been performed and analyzed. *RECOMM* output of recommendation lists will be statistically characterized for similarity to the collected data.

6.2 Future Work

After full development of the *RECOMM* tool, it will be extensively used to test the hypotheses advanced about the relative importance of the above mentioned variables and algorithms for specific social networks.

An interesting issue is the degree of generality of the chosen variables and algorithms for diverse social networks, i.e. to what extent the tool will need fine tuning to apply it to each different network.

6.3 Main Contribution

The main contribution of this work is the recognition of the importance of randomization and interestingness to generate recommendation lists.

References

1. Chen, J., Geyer, W., Dugan, C., Muller, M. and Guy, I.: "Make New Friends, but Keep the Old" – Recommending People on Social Networking Sites, in Proc. CHI 2009, pp. 201-210, (2009).
2. Exman, I.: Interestingness - A Unifying Paradigm - Bipolar Function Composition, in Proc. KDIR'2009 Int. Conf. on Knowledge Discovery and Information Retrieval, pp. 196-201, (2009).
3. Golbeck, J. and Hendler, J.: Inferring Trust Relationships in Web-based Social Networks, ACM Trans. on Internet Technology (TOIT), Vol. 6, pp. 497-529, (2006).
4. Gross, R. and Acquisti, A.: Information Revelation and Privacy in Online Social Networks (The Facebook Case), in ACM Workshop on Privacy in the Electronic Society (WPES), (2005).
5. Huberman, B.A., Romero, D.M. and Wu, F.: Social Networks that Matter: Twitter under the Microscope, arXiv:0812.1045v1, December (2008).
6. Kautz, H. Selman, B. and Shah, M.: Combining Social Networks and Collaborative Filtering, Comm. ACM, Vol. 40, 63-65, (1997).
7. Konstas, I., Stathopoulos, V. and Jose, J.M.: On Social Networks and Collaborative Recommendation, in Proc. SIGIR'09 32nd Int. ACM SIGIR Conf. Information Retrieval, pp. 123-124, (2009).
8. Roth, M., Ben-David, A., Deutscher, D., Flysher, G., Horn, I., Leichtberg, A., Leiser, N., Matias, Y. and Merom, R.: Suggesting Friends Using the Implicit Social Graph, in Proc. ACM Conf. KDD'10, (2010).
9. Tang, W., Zhuang, H. and Tang, J.: Learning to Infer Social Ties in Large Networks, in Machine Learning and Knowledge Discovery in Databases, LNCS vol. 6913, pp. 381-397, (2011).