Bring Order to Online Social Networks

Ruichuan Chen
MPI-SWS, Germany
rchen@mpi-sws.org

Eng Keong Lua
Nanyang Polytechnic, Singapore
Carnegie Mellon University, USA
eklua@computer.org

Zhuhua Cai
Rice University, USA
zhuhua.cai@rice.edu

***Abstract*—Online social networking systems are rapidly becoming popular for users to share, organize and locate interesting content. However, these systems have increasingly been employed as platforms to spread spam and irrelevant content, abusing valuable human attention and service resource.**

In this paper, we propose a *social reputation model* to guide users to browse desirable content. First, we compute the statistical correlation between different users to distinguish various user interests; then, since a user's friends are usually trustworthy and share similar interest, we further exploit the inherent friend relationships to perform reliable social enhancements of vote history extension and efficient reputation estimation. Our social reputation model provides strong incentives for user cooperation, and moreover, our model can handle practical problems of inactive users, unpopular content and Sybil attacks effectively and efficiently. Our evaluation on a large-scale network validates our analysis, and shows that our social reputation model can help users find the desirable content in various scenarios with a precision of 94%.

I. Introduction

Online social networking sites such as YouTube, Flickr, MySpace and Facebook are among the most popular sites on the Internet, and continue to experience explosive growth both in terms of the number of communities and the overall population. In such systems/sites, participating users construct online social networks by declaring social links with their friends, e.g., real-world acquaintances, online acquaintances or like-minded contacts. The online social networks so constructed provide a powerful means for users to share, organize and locate interesting content.

Alongside with rapid popularization, current online social networking systems, however, have unfortunately been employed as ideal platforms to spread *spam*. Such spam generally employs attractive titles and/or popular tags but fake data, so that unsuspecting users without adequate experience and knowledge may be attracted, and then visit spammers' sites. Moreover, since each user has a unique interest, there exists an additionally massive amount of *irrelevant* (not spam but non-preferred) content for each user. Altogether, the *undesirable* (i.e., spam or irrelevant) content existing in online social networking systems could potentially attract hundreds of millions of users, thus severely abusing one of the most valuable resources in the information age: human attention.

In general, traditional reputation models [1], [2], [3] and recommender systems [4], [5], [6] could be used to help users identify desirable content. Nevertheless, current reputation models target only spam content but not the irrelevant content from each user's perspective; while, recommender systems usually identify a small number of users with similar interests to help make recommendation about new content, without the full consideration of past popularity votes about content.

In this paper, we propose a *social reputation model* to guide users to browse desirable content. Generally, in an online social networking system, there is a centralized service provider (or a set of centralized service providers) who maintains the whole system and knows all users' vote histories; therefore, in our basic reputation model, the service provider is able to utilize these maintained vote histories to compute a personalized reputation score for each of a user's potential next-click items, based on the statistical correlation between the user and those associated users who have voted this item. This reputation score can be used to help users make a more informed decision on whether to browse a particular item.

Moreover, in an online social networking system, a user may have a number of friends who share similar interest and give similar votes on specific items; besides, the friends are usually more trustworthy than other common users. As in real-world, a user also can identify the desirable content depending largely on her friends' past experiences; thus, we further exploit friend relationships to socially improve our basic reputation model. Specifically, the service provider utilizes a user's friends' vote histories to extend the user's own vote history. By considering these extended vote histories, the service provider is able to perform a more accurate and efficient reputation computation.

Our social reputation model provides strong incentives for users to give votes frequently and accurately, and it can help users identify desirable content without aggravating the system overhead significantly. Moreover, our model can solve the problems of inactive users, unpopular content and Sybil attacks. The evaluation results illustrate that our social reputation model works well in various scenarios, and could be deployed in practical online social networking systems.

II. Social Reputation Model

In this section, we elaborate our basic reputation model and its social enhancement, respectively.

A. Basic Reputation Model

1) Vote Generation and Maintenance: In current online social networking system, most users browse content through following friend links, and the others browse content via using search facilities or following the links indicated by external sources [7]. Each user can give votes on browsed items from her own perspective. Generally, there is a centralized service provider (or a set of centralized service providers) maintaining the whole system, these votes cast by users can be incrementally uploaded to the centralized service provider, so that the service provider has the *complete* capacity of knowing all users' vote histories including, for instance, "which users have voted a specific item?", "which items have been voted by a specific user?", and "what score has been given by a specific user on a specific item?". Hereafter, we name the users who have voted an item as the item's *associated voters*.

In our design, the value range of a vote varies between -1 (extremely undesirable) and 1 (extremely desirable). Here, a vote reflects the associated voter's own interest, e.g., even if an item is not spam, it may still be given a vote of -1 once the voter considers this item extremely irrelevant. That is, both the spam and irrelevant items will be given low scores according to the user's interest.

2) Reputation Computation: Generally, a *user* U who is browsing or searching on the online social networking site/system will find a number of items listed on each page. Among these items, some may be undesirable, and genuine users usually want to browse the remaining desirable items.

In current systems, due to the lack of reliable evidence, a user typically resorts to an ad-hoc or experiential selection mechanism for choosing the *next-click item*. In the following, we develop a reputation model to help users identify desirable items from each user's own perspective.

Vote Extraction. In our model, the service provider first extracts each *potential* next-click item C_i's associated voter list VL_i from the centralized vote history database. Then, the service provider traverses VL_i and obtains each associated voter V_{ij} (indicated by VL_i)'s past vote history VH_{ij}. Once these vote histories have been obtained, the service provider can easily extract each voter V_{ij}'s vote v_{ij} on the item C_i.

Similarity Computation. With the above vote extraction mechanism, the service provider can obtain each associated voter V_{ij}'s vote v_{ij} on the potential next-click item C_i. Based on these votes, the service provider is able to compute the reputation score of C_i for user U. The simplest way is to execute the unweighted averaging on these votes; however, this scheme cannot distinguish between different voters, e.g., both like-minded voters and conflict-minded voters are treated equally. Instead, in our design, we compute a *normalized cosine similarity* measure for weighing each vote, and execute the *weighted averaging* to compute the reputation score.

Assume that there are m items on which both user U and an associated voter V_{ij} have voted; moreover, U and V_{ij} have the vote histories of $VH_U = \{a_1, a_2, \cdots, a_k, \cdots, a_m\}$ and $VH_{V_{ij}} = \{b_1, b_2, \cdots, b_k, \cdots, b_m\}$ given on these m co-voted items, respectively. Then, the weight coefficient for the vote from V_{ij} can be computed as follows:

$$\begin{aligned} sim_{(U,V_{ij})} &= \cos_{norm}(\overrightarrow{VH_U}, \overrightarrow{VH_{V_{ij}}}) \\ &= \frac{\sum_{k=1}^{m}(a'_k \times b'_k)}{\sqrt{\sum_{k=1}^{m}(a'_k)^2} \times \sqrt{\sum_{k=1}^{m}(b'_k)^2}} \end{aligned} \tag{1}$$

Here, $\cos_{norm}$ is a function of computing the normalized cosine similarity of two vectors; $\overrightarrow{VH_U}$ and $\overrightarrow{VH_{V_{ij}}}$ are the vectorized VH_U and $VH_{V_{ij}}$; a'_k and b'_k are the normalized a_k and b_k, i.e., $a'_k = \frac{a_k}{\max(|a_k|,|b_k|)}$ and $b'_k = \frac{b_k}{\max(|a_k|,|b_k|)}$. In particular, we use several heuristics to address exceptional cases that arise in practice. Firstly, if there are no co-voted items (i.e., $m = 0$), then $sim_{(U,V_{ij})} = 0$; secondly, if both user U and the associated voter V_{ij} give a vote of zero on a co-voted item (i.e., $a_k = b_k = 0$), then $a'_k = b'_k = 1$; thirdly, if U gives the votes of zero on all these co-voted items and V_{ij} gives non-zero votes on all of them (i.e., $\forall k \in [1,m], a_k = 0$ and $b_k \neq 0$), then $sim_{(U,V_{ij})} = 0$; similarly, if U and V_{ij}, respectively, give non-zero and zero votes on all these co-voted items (i.e., $\forall k \in [1,m], a_k \neq 0$ and $b_k = 0$), then $sim_{(U,V_{ij})} = 0$ as well.

The weight coefficient $sim_{(U,V_{ij})}$ expresses the statistical correlation between the two users' vote histories, and captures whether they tend to vote correlatively or uncorrelatively. That is, $sim_{(U,V_{ij})}$ actually reflects whether user U and the associated voter V_{ij} have similar interest over time.

Weighted Averaging. Based on the above computed weight coefficient of each associated voter, the service provider performs the weighted averaging to compute the reputation score $R_{(C_i,U)}$ of each potential next-click item C_i for user U. This reputation score can be used to help users make a more informed decision on whether to browse an item. Specifically, we merely consider the positively correlative associated voters because the votes from negatively correlative associated voters may be unreliable (e.g., conflict, chaotic, or even malicious).

$$R_{(C_i,U)} = \frac{\sum_{j=1}^{|VL'_i|}\left(v_{ij} \times sim_{(U,V_{ij})}\right)}{\sum_{j=1}^{|VL'_i|}|sim_{(U,V_{ij})}|} \in [-1, 1] \tag{2}$$

Here, $|VL'_i|$ denotes the size of C_i's positively correlative associated voter list VL'_i; moreover, if there are no positively correlative voters associated with the potential next-click item C_i (i.e., $|VL'_i| = 0$), then $R_{(C_i,U)} = 0$. This weighted averaging scheme differentiates different voters, and gives more weight to votes from these like-minded voters, thus it can be used to assist in better distinguishing between desirable and undesirable items. Based on the computed reputation score of each potential next-click item, users should be inclined to browse the item with a higher reputation score.

Privacy Issue. Some online social networking systems (e.g., Flickr and YouTube) allow a user's shared content to be visible to other common users even nonparticipating people, by default. However, some other systems (e.g., Facebook, MySpace and LinkedIn) merely allow a user to visit her close friends, e.g., those direct or two-hop friends. Therefore, in such privacy-concerned systems, once a user would like to browse an item with high reputation score but privately protected, the user has to issue a friend (or item) request to solicit the item's publisher to be her friend (or directly share this item with her). Since the user and this desirable item's publisher may actually share a similar interest, this kind of requests could guide users to create many friend links between *potential* friends.

3) Analysis: Here, we first present four *advantages* of our proposed basic reputation model.

Personalized. We compute each potential next-click item's reputation score by weighing associated voters' past votes from each user's perspective. In our design, the reputation computation relies on a user's own vote history, thus the reputation score of the same item is distinct for different users with different interests.

Threat-resistent. Our basic reputation model has the capacity of distinguishing desirable content from not only the usual spam content but also the irrelevant content, from each user's own perspective. Moreover, due to the fact that our reputation computation is rooted from the evaluation based on a user's own vote history, our proposed reputation model is relatively resistent to various malicious voting behaviors.

Sparsity-resistent. In large-scale networked systems, users' votes may be very sparse, so called the *sparsity* problem. Fortunately, in online social networking systems, tending to browse a potential next-click item implies that the user and the item's associated voters have similar interest to some extent; moreover, in such systems, most users browse content via following friend links [7], thus the user and the potential next-click item's associated voters may be even within only a few friend-hops. These indicate that there should be a substantial number of items co-voted by both the user and these associated voters, so that the service provider is able to compute an accurate reputation score for each potential next-click item with high probability, and this sparsity problem will not affect the performance of our basic reputation model significantly.

Incentive. Since users are usually rational in seeking to maximize their individual utilities, existing reputation models are greatly penalized by the lack of accurate votes given by users. In our design, the dependence on a user's own vote history provides strong incentives for the user to give votes on her browsed items more often and accurately. Via giving a sufficient number of accurate votes, a user enables the service provider to compute the reliable personalized reputation score of each item for her; otherwise, due to the lack of accurate votes, the user cannot obtain the reliable reputation score to help identify the desirable content.

Though having the above advantages, our basic reputation model has to face a couple of practical *challenges*.

Inactive User Problem. To identify desirable content, a user has to vote a sufficient number of items to make the service provider really understand the user's interest. However, there are many users who participate into the system *inactively* (i.e., they browsed only a few items), so that they could merely give a small number of votes even if they are willing to vote. Moreover, this problem becomes much more serious for newly incoming users (i.e., newcomers without any vote history), and they cannot benefit from our basic reputation model.

Unpopular Content Problem. In current online social networking systems, there may be a number of *unpopular* (or new) items with only very limited votes. This indicates that there may exist only a few associated voters; further, these associated voters may have certain unusual interests, thus the items co-voted by a user and these associated voters may be relatively sparse even if our basic reputation model is generally sparsity-resistent. Considering these few voters' sparse associated votes may result in a biased evaluation of the unpopular items' reputation scores, so that we are not able to give accurate reputation scores to the unpopular content.

B. Social Enhancement

In real-world, people usually consult their friends in choosing the movies to watch, the things to buy, etc. Similarly, in online social networking systems, a user may also have a number of friends, e.g., her real-world acquaintances, online acquaintances, or like-minded contacts. These friends are very different from the great majority of other users. A user and her friends often share similar interest and may give similar votes on a specific item; moreover, friends are usually more trustworthy than other common users. To exploit the inherent information of friends, we provide two kinds of social enhancement for our basic reputation model: *vote extension* and *efficient estimation*.

1) Vote Extension: In an online social networking system, a user may have only a few past votes. This would make it difficult to accurately compute the correlation between each associated voter and the user herself, thus influencing the performance of our basic reputation model. To *extend* the user's vote history reliably, we additionally consider her friends' vote histories before performing the reputation computation.

Proxy-based (indirect) Extension. Since a user shares similar interest with her friends, we could let each friend act as a *proxy* to perform an independent reputation computation, and then integrate these computed reputation scores as well as the user's own computed reputation score to generate the final reputation score of each potential next-click item.

Assume that user U has f friends in the system, denoted by $\{F_j\}_{j=1}^{f}$, so that the service provider can rely on each F_j's vote history to compute an individual reputation score $R_{(C_i,F_j)}$ for each potential next-click item C_i, as described in section II-A2. Finally, the service provider integrates these f reputation scores $\{R_{(C_i,F_j)}\}_{j=1}^{f}$ with user U's own computed reputation score $R_{(C_i,U)}$ to generate the proxy-based reputation score $R^p_{(C_i,U)}$ of each potential next-click item C_i.

$$R^p_{(C_i,U)} = \frac{R_{(C_i,U)} + \sum_{j=1}^{f} R_{(C_i,F_j)}}{1+f} \tag{3}$$

The key idea of solving the problems described in section II-A3 is to extend a user's vote history reliably. In this proxy-based extension scheme, we utilize a user's friends acting as proxies to perform the individual reputation computations, i.e., we actually enrich a user's vote history *indirectly*. However, since *each* of these friends may also have only a few past votes, so they may generate biased/inaccurate reputation scores as well; moreover, the proxy-based extension scheme needs $f+1$ reputation computations which may incur much burden on the service provider; therefore, the applicability of this proxy-based extension scheme is questionable. In the following, we will elaborate another vote extension scheme which could conquer these drawbacks.

Direct Extension. In this scheme, we extend a user's vote history *directly*. Assume that user U with vote history VH_U has f friends in the system, denoted by $\{F_j\}_{j=1}^{f}$; moreover, each friend F_j has the vote history VH_{F_j}. In this direct extension scheme, the service provider first extracts the vote histories of user U's friends from the centralized vote history database, and then performs an averaging on these friends' vote histories as well as the user U's own vote history to compute U's extended vote history VH'_U, as follows.

$$VH'_U = avg\left(VH_U, \{VH_{F_j}\}_{j=1}^{f}\right) \tag{4}$$

Here, each vote history is treated as a vector, and the avg is defined to be a function of computing the averages of *nonempty* values at each position in these vectors.

Since friends are usually trustworthy and share similar interest, the service provider can apply the above direct vote history extension scheme to enrich a user's vote history reliably. Based on the extended vote history VH'_U, the service provider can perform the reputation computation as described in section II-A2 to compute the final reputation score of each potential next-click item. Note that, a user could *virtually* experience much more unbrowsed items and the reputation computation is executed only once, therefore, the service provider is able to compute the final reputation score more accurately and efficiently.

2) Efficient Estimation: Each time the service provider executes the reputation computation, she relies on a user's friends' vote histories indirectly or directly. Note that, a user and her friends usually share similar interest, and most users' browsing actions result from following friend links. These two observations imply that the items a user tends to browse may have been browsed and voted by her friends.

Considering the above implication, in our design if many friends of a user have already voted a potential next-click item, the service provider does not need to compute the reputation score of the item for the user again; as an alternative, the service provider utilizes friends' votes to efficiently *estimate* the reputation score.

Assume that the service provider wants to compute the reputation score of a potential next-click item C_i for user U, and moreover, U has f friends among whom there are f' friends $\{F_j\}_{j=1}^{f'}$ having already given the votes $\{v_{ij}\}_{j=1}^{f'}$ on the item C_i. Note that, all such information can be extracted from the centralized vote history database. Specifically, if only a few friends have voted the item (i.e., f' is small), or there are significant differences among these f' friends' associated votes (i.e., the average absolute deviation δ of $\{v_{ij}\}_{j=1}^{f'}$ is large), the associated votes may be biased, so now we have to return back to use the normal social reputation model as described before; otherwise, if a sufficient number (say, $f' \geq 4$) of friends have voted the item identically (say, $\delta \leq 0.1$), the service provider can reliably utilize the associated votes to efficiently estimate the reputation score $R'_{(C_i,U)}$ of the item C_i.

$$R'_{(C_i,U)} = \frac{\sum_{j=1}^{f'} v_{ij}}{f'} \quad \text{if} \begin{cases} f' \geq 4 \\ \delta = \frac{\sum_{j=1}^{f'} \left| v_{ij} - R'_{(C_i,U)} \right|}{f'} \leq 0.1 \end{cases} \tag{5}$$

Here, $R'_{(C_i,U)}$ can be treated as the final reputation score $R_{(C_i,U)}$ of the item C_i, from user U's perspective.

Specifically, a malicious user may masquerade as like-minded user to become a user's friend, and a friend may also be compromised. To address this "malicious friend" problem, the service provider should compute the correlation coefficient $sim_{(U,F_j)}$ between user U and each associated friend F_j. If $sim_{(U,F_j)} < 0.5$, the friend may be malicious or uncorrelated, so we choose not to take this friend's vote into account.

3) Analysis: Through integrating the basic reputation model with these social enhancements, we obtain a social reputation model which can not only inherit the advantages of basic reputation model but also have several new features.

"New"-resistent. The key idea to solve the *general* "new" problem, including the inactive (or new) user problem and the unpopular (or new) content problem as described in section II-A3, is to extend user's vote history reliably. Since friends are usually trustworthy and have similar interest, in our social reputation model, the service provider utilizes friends' vote histories to reliably extend a user's own vote history.

For inactive users or unpopular content, the number of items co-voted by both an (inactive) user and the (unpopular content) item's associated voters should be much larger through taking into account the user's friends' vote histories, thus the service provider is able to perform a more accurate/unbiased reputation computation. Secondly, for a new user without any vote history, once joining the online social networking system, the user builds up her friend links and relies on our vote extension scheme to initialize her vote history, so that the new user could also perform the reputation model normally. Finally, for new content, users have to resort to an ad-hoc or experiential reputation estimation, as used in existing systems. Such case is unavoidable because there are no associated votes.

Sybil-resistent. In online social networks, Sybil [8] users could create a large number of identities but few friend relationships with genuine users [9]; moreover, even if an item's associated voters are Sybil users, they still cannot significantly influence the performance of our system because our reputation computation is rooted from the evaluation based on a user's and her friends' vote histories; therefore, our social reputation model is Sybil-resistent. Complementarily, we could further utilize friend links to construct SybilLimit-style [9] "random routes" to defend against Sybil attacks.

To sum up, our social reputation model substantially enhances the performance of our basic reputation model. Note that, if a user does not have any friends in the system, our social reputation model falls back to the basic form. In some sense, the user should "pay the price" for having no friends.

C. System Overhead

An effective social reputation model should not incur much burden on current online social networking systems. In the following, we will discuss three main kinds of system overhead and some complementary countermeasures.

Computation Overhead. While a user is surfing on the online social network, the service provider guides the user to browse desirable content by computing the reputation score of each potential next-click item, from the user's own perspective.

Firstly, during the reputation computation process, computing the similarity between a user and each of these potential next-click items' associated voters is relatively expensive. Fortunately, most users browse content by following friend links, so that a user and these associated voters may be within only a few friend-hops with relatively high probability; therefore, in our system, the service provider could *periodically* compute the similarity between a user and each of her close friends (e.g., within two friend-hops) to avoid repeatedly computing those frequently needed similarity scores.

Secondly, if the number of a potential next-click item's associated voters is too large, the service provider should choose a subset of these voters (having the most vote overlap with the surfing user) to perform the reputation computation, both in order to control the computation overhead, and to ensure that the most useful associated voters are considered. Similarly, if the number of a user's friends is too large, the service provider should also choose a subset of these friends (having the most abundant vote history) to perform the vote extension in our social reputation model.

Lastly, a user and her friends usually share similar interest, and moreover, most users browse content via following friend links, so that an item a user tends to browse may have been browsed and voted by her friends. We propose an efficient reputation estimation scheme to further reduce the computation overhead of our social reputation model.

Communication Overhead. In our system, each user incrementally uploads her vote history to the service provider. Compared with the high throughput of current online social networking system, this kind of uploading will not aggravate the system's communication overhead significantly.

Storage Overhead. To support our social reputation model, the service provider should maintain each user's vote history in a centralized vote history database. Alternatively, this vote history database could be maintained by a set of centralized service providers, e.g., in a DHT manner. Compared with the massive volume of originally maintained content (e.g., videos, photos, etc.), maintaining these vote histories will not aggravate the system's storage overhead significantly.

According to the above discussion on three main kinds of system overhead, we conclude that our social reputation model will not incur significant overhead, and it is applicable to current online social networking systems.

III. Evaluation

The goal of our social reputation model is to guide users to browse the desirable content in online social networking systems. Ideally, we would solicit the support of the administrators of current online social networking systems, and then deploy our social reputation model in realistic systems, so that we could extract all users' vote histories and friend relationships to perform the planned experiments. Unfortunately, we were not allowed to perform such deployment due to the administrators' consideration of operation and privacy; we did, however, have the realistic massive-scale network traces [7] of current online social networking systems. Therefore, we chose to develop a prototype system implementing our proposed social reputation model with approximately 6120 lines of Java code, and evaluated its performance based on these realistic traces. The evaluation results show that our social reputation model can help users find the desirable content with a precision of around 94%, via either following friend links or using search facilities. [10] describes the evaluation in detail.

IV. Discussion

Below we further discuss some possible design choices:

Length of Friend Links. In our social reputation model, we have merely considered the inherent information of direct friends, but one might also consider using friend relationships of two hops or even longer. Considering these extra friends could further extend a user's vote history, and make the service provider have a better chance to perform the efficient reputation estimation. On the other hand, as the length of friend links increases, it becomes increasingly unclear whether the extra friends still share similar interest and are trustworthy. Therefore, there is a tradeoff between efficiency and reliability.

Interest Group. Many current online social networking systems allow users to create and join interest groups. Users in each interest group may have a specific interest, and moreover, these users do not necessarily link to each other in the online social network; thus, considering a user's group members could additionally utilize the information of these shared-interest non-friends. However, many interest groups are unrestricted by allowing any user to join, therefore, taking into account the interest group membership is not highly reliable.

References

[1] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *WWW*, 2003.

[2] L. Xiong and L. Liu, "Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 7, pp. 843–857, 2004.

[3] K. Walsh and E. G. Sirer, "Experience with an object reputation system for peer-to-peer filesharing," in *NSDI*, 2006.

[4] D. Billsus and M. J. Pazzani, "Learning collaborative information filters," in *ICML*, 1998.

[5] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *WWW*, 2001.

[6] M. R. McLaughlin and J. L. Herlocker, "A collaborative filtering algorithm and evaluation metric that accurately model the user experience," in *SIGIR*, 2004.

[7] A. Mislove, M. Marcon, K. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and analysis of online social networks," in *IMC*, 2007.

[8] J. R. Douceur, "The sybil attack," in *IPTPS*, 2002.

[9] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao, "Sybillimit: A near-optimal social network defense against sybil attacks," in *IEEE Symposium on Security and Privacy*, 2008.

[10] R. Chen, E. K. Lua, and Z. Cai, "Bring order to online social networks." [Online]. Available: http://www.mpi-sws.org/~rchen/papers/social-tr.pdf