Bring Order to Online Social Networks

Ruichuan Chen, Eng Keong Lua, Zhuhua Cai

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

In this paper, we propose a social reputation model to guide users to browse the 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 the similar interest, we further exploit the inherent friend relationships to perform the reliable social enhancements of vote history extension and efficient reputation estimation. Our social reputation model provides a strong incentive for user cooperation, and moreover, our model can handle the practical problems of inactive users, unpopular content and Sybil attacks effectively and efficiently. Our evaluation on a large-scale realistic 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 around 94%.

Categories and Subject Descriptors
C.2.4 [Computer-Communication Networks]: Distributed Systems—Distributed applications

General Terms
Design, Security

Keywords
Online social networks, social reputation model

1. INTRODUCTION
Online social networking sites such as YouTube, Flickr, MySpace and Facebook are among the most popular sites on the Internet [1], 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 by spam, and then visit the associated spammers’ sites. Moreover, since each participating user has a unique interest, there exists an additionally massive amount of irrelevant (not spam but non-preferred) content for each particular user.

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. Recently, some commercial applications have been deployed into online social networking systems, e.g., the Marketplace application in Facebook, therefore, undesirable content (here, commodities) may even waste participating users’ money directly. As a result, undesirable content has the detrimental effects of decreasing the confidence of participating users, and ultimately, leading users to abandon the online social networking systems.

In general, traditional reputation models [10, 20, 21] and recommender systems [3, 11, 16] could be used to help users distinguish between desirable (i.e., preferred according to their own interests) and undesirable 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 full consideration of past popularity votes about content. Moreover, in realistic online social networking systems, most existing schemes encounter several practical problems: a) inactive users: many users browse and/or vote only a few (content) items, and b)
unpopular content: much content is browsed and/or voted by only a few users. These problems make existing schemes not have sufficient overlapping votes to help inactive users identify the desirable content, and to help participating users identify whether an unpopular (content) item is desirable. Lastly, most existing schemes are vulnerable to Sybil [6] attacks where malicious users create a large number of virtual identities to perform malicious behaviors.

In this paper, we propose a social reputation model to guide surfing users to browse the desirable content effectively and efficiently, avoiding the problems in existing schemes. Generally, in an online social networking system, there is a centralized service provider (or a set of centralized service providers) who manages/maintains the whole system and knows all participating 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 surfing user’s potential next-click items, based on the statistical correlation between the surfing user and those associated users who have voted this item. This reputation score can be used to help the surfing user 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 the similar interest and give similar votes on specific items: besides, the friends are usually more trustworthy than other common users. As in the real-world, a surfing user also can identify the desirable content depending largely on her friends’ past experiences; thus, we further exploit the friend relationships to socially enhance the effectiveness and efficiency of our basic reputation model. Specifically, the service provider uses a surfing user’s friends’ vote histories to reliably extend the surfing user’s own vote history directly or indirectly. 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 a strong incentive for participating users to give votes frequently and accurately, and it can help users identify the desirable content without aggravating the overhead of current online social networking systems significantly. Moreover, our model can solve the problems of inactive users, unpopular content and Sybil attacks effectively and efficiently, based on our proposed social enhancement. We have implemented a prototype system, and evaluated its performance on a massive-scale testbed with realistic network traces. The evaluation results illustrate that our social reputation model works well in various different scenarios, and could be deployed in practical online social networking systems.

The rest of this paper is organized as follows. We specify the system model in section 2. The details of our proposed social reputation model are elaborated in section 3. We then present the experimental design and analyze the experimental results in section 4. Section 5 discusses several possible design choices, followed by an overview of related work in section 6. Finally, we conclude this paper in section 7.

2. SYSTEM MODEL

Current online social networking systems are usually run by powerful service providers, and are accessible via the Web. Generally, an online social network is composed of users and content. To elaborate our design clearly, we introduce the “user” and “content” in the following two sections, respectively.

2.1 User Model

In our design, each user has exactly one unique identity. Even if a person creates multiple identities and the associated inter-identity links, we consider each of these identities as a separate user.

In online social networking systems, a user may invite another user (e.g., her real-world acquaintance, online acquaintance or like-minded contact) to be her friend. If the invitee user accepts the friend invitation, a symmetric friend link is created between them. Though online social networks have a high fraction of symmetric friend links [13], some systems (e.g., Flickr and LiveJournal) allow users to link to any other users without consent from the link target, thus there may also exist many asymmetric friend links. Usually, a user and her friends share a similar interest and have a similar opinion on a specific item; moreover, the friends are usually more trustworthy than other common users. As a result, the friend links are often utilized by users as a shortcut to others, and many users surf on the online social networks following such friend links [13].

2.2 Content Model

The online social networking systems have been transforming the way content is created and distributed. In such systems, the participating users construct online social networks for sharing, organizing and locating interesting content. Here, a (content) item may be a video, photo, blog, commodity, contact, etc.

Specifically, some online social networking systems (e.g., Flickr and YouTube) allow users even nonparticipating people to browse other users’ shared content by default. However, in some other systems (e.g., Facebook, MySpace and LinkedIn), participating users willingly share personal information about themselves, thus there are significantly higher privacy-related concerns and the privacy protection becomes vital; here, a user’s shared content is usually only accessible to her close friends, or as an alternative, there exists various fine-
grained privacy settings for the shared content.

3. DESIGN

In this section, we first describe the threat model in online social networking systems; afterwards, we elaborate and analyze our basic reputation model and its social enhancement, respectively.

3.1 Threat Model

Current online social networking systems have widely been employed as ideal platforms to spread spam. Here, the spammers manufacture spam content with attractive titles and/or popular tags but fake data; then, they share such spam content in the system for attracting unsuspecting users to visit their sites. Actually, for various users with different interests, there are an additionally massive amount of irrelevant (not spam but non-preferred) content, e.g., a rock-and-roll lover may consider the classical music as irrelevant, and vice versa.

The recent measurement study in [13] reported that about 80% browse actions in online social networking systems result from following friend links to visit friends, friends-of-friends, and further. Naturally, when a user surfs on the online social network, she may be attracted by numerous attractive titles and/or popular tags of undesirable (i.e., spam or irrelevant) content with relatively high probability, and then choose to browse the undesirable content even the associated sites.

Besides, the remaining 20% browse actions were reported to be the result of either using search facilities or following the links indicated by external Web sites or emails [13]. Here, for instance, when a user searches for a tag, the service provider should return the descriptor information of the requested tag’s associated (content) items back to the user. Difficult to distinguish desirable (i.e., preferred according to her own interest) items from undesirable items merely via the descriptor information, the user may unfortunately select an undesirable item to browse.

Undesirable content could potentially be reached by hundreds of millions of users, thus severely wasting one of the most valuable resources in the information age: human attention. Moreover, undesirable content has the detrimental effects of decreasing the confidence of participating users, and ultimately, causing users to abandon the online social networking systems.

In this work, our objective is to bring order to the online social networks. We will propose a social reputation model to guide users to browse the desirable content without the influence of undesirable content, and meanwhile, to overcome the drawbacks of existing schemes.

3.2 Basic Reputation Model

In this section, we specify our basic reputation model, and then in section 3.3, we use the social information to enhance its effectiveness and efficiency.

3.2.1 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 [13]. Usually, a user gives votes on her browsed items from her own perspective. Due to the fact that there is generally a centralized service provider (or a set of centralized service providers) managing/maintaining the whole online social networking system, these votes cast by participating users can be incrementally uploaded to the centralized service provider in a piggyback way, 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 “which score has been given by a specific user on a specific item?”. Specifically, 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 vote scores according to the user’s interest. An example of the centralized vote history database is presented in Table 1, where the “∅” symbol means that the users have not voted the corresponding items.

<table>
<thead>
<tr>
<th></th>
<th>Braveheart</th>
<th>Titanic</th>
<th>E.T.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>1</td>
<td>∅</td>
<td>0.6</td>
</tr>
<tr>
<td>Bob</td>
<td>0.8</td>
<td>−1</td>
<td>0.9</td>
</tr>
<tr>
<td>Cindy</td>
<td>∅</td>
<td>−0.3</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Table 1: A Fragment of a Vote History Database
systems, the service provider first extracts each potential next-click item \( C_i \)'s associated voter list \( V_{L_i} \) from the centralized vote history database. Then, the service provider traverses \( V_{L_i} \) and obtains each associated voter \( V_{ij} \) (indicated by \( V_{L_j} \))'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 the surfing 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 final reputation score.

Without loss of generality, we suppose that there are \( m \) items on which both the surfing 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:

\[
sim(U, V_{ij}) = \cos_{\text{norm}}(VH_U, VH_{V_{ij}}) = \frac{\sum_{k=1}^{m} (a'_k \times b'_k)}{\sqrt{\sum_{k=1}^{m} (a'_k)^2} \times \sqrt{\sum_{k=1}^{m} (b'_k)^2}} \tag{1}
\]

Here, the \( \cos_{\text{norm}} \) is a function of computing the normalized cosine similarity of two vectors; \( VH_U \) and \( VH_{V_{ij}} \) 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 the surfing 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 uncorrelatedly. That is, the \( \sim(U, V_{ij}) \) actually reflects whether the surfing user \( U \) and the associated voter \( V_{ij} \) have the 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 the surfing user \( U \). This reputation score can be used to help the surfing user 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). The weight coefficient \( \sim(U, V_{ij}) \) and the associated voter \( V_{ij} \) are computed as follows:

\[
R(C_i, U) = \frac{\sum_{j=1}^{|V_{L_i}|} (v_{ij} \times \sim(U, V_{ij}))}{\sum_{j=1}^{|V_{L_i}|} |\sim(U, V_{ij})|} \in [-1, 1] \tag{2}
\]

Here, \( |V_{L_i}| \) denotes the size of \( C_i \)'s positively correlative associated voter list \( V_{L_i} \); moreover, if there are no positively correlative voters associated with the potential next-click item \( C_i \) (i.e., \( |V_{L_i}| = 0 \)), 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, the surfing user should be inclined to browse the item with a higher reputation score.

**Privacy Issue.** As mentioned in section 2.2, some current online social networking systems (e.g., Flickr and YouTube) allow a participating 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 surfing user would like to browse an item with high reputation score but privately protected, the surfing user should 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 surfing 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.

### 3.2.3 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 the associated voters’ past votes from the surfing user’s perspective. In our design, the reputation computation relies on the surfing user’s own vote history, thus the reputation score of the same item is distinct for different users with different interests.
Threat-resistant. Our basic reputation model presented before 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 the surfing user’s own vote history, our proposed reputation model is relatively resistant to various malicious voting behaviors performed by malicious users.

Sparsity-resistant. In large-scale networked systems, the votes given by participating users 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 surfing user and the item’s associated voters have the similar interest to some extent; moreover, in such systems, most users browse content via following friend links [13], thus the surfing 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 surfing 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 influence the performance of our basic reputation model significantly.

Incentive. Since many participating users in current networked systems including the online social networking systems are rational in seeking to maximize their individual utilities, the current reputation models are greatly penalized by the lack of accurate votes given by users. In our design, the dependence on a surfing user’s own vote history provides a strong incentive for the user to give votes on her browsed items more often and accurately. Via giving a sufficient number of accurate votes, a surfing 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 surfing user cannot obtain the reliable reputation score to help identify the desirable content.

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

Inactive user problem. In order to identify desirable content reliably, a user has to vote a sufficient number of items to make the service provider really understand the user’s interest. However, in a realistic online social networking system, 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 the newly incoming users (i.e., newcomers without any vote history), because they cannot benefit from our basic reputation model at all.

Unpopular content problem. In current online social networking systems, there may be a number of unpopular (or even new) items with only very limited votes, which 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 both the surfing user and the associated voters may also be relatively sparse even if our basic reputation model is generally sparsity-resistant. Taking into account 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.

Bearing these challenges in mind, we utilize the inherent friend relationships existing in online social networking systems to socially enhance the effectiveness and efficiency of our basic reputation model, as elaborated in the next section.

3.3 Social Enhancement

In the 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 participants. A user and her friends often share the similar interest and may give similar votes on a specific item; moreover, the 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.

3.3.1 Vote Extension

In an online social networking system, a surfing user may have only a few past votes. This would make it relatively 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 surfing user’s vote history reliably, we can additionally consider her friends’ vote histories before performing the reputation computation.

Proxy-based (indirect) Extension. Due to the fact that a user shares the similar interest with her
friends, we could naturally let each of these friends act as a proxy to perform an independent basic 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.

Without loss of generality, we assume that the surfing 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 \( R_j(C, F_j) \) for each potential next-click item \( C_i \), as described in section 3.2.2. Finally, the service provider integrates these \( f \) reputation scores \( \{R_j(C, F_j)\}_{j=1}^f \) with the surfing user \( U \)’s own computed reputation score \( R_{C, U} \) to generate the proxy-based reputation score \( R_{C, U}^\prime \) of each potential next-click item \( C_i \).

\[
R_{C, U}^\prime = \frac{R_{C, U} + \sum_{j=1}^{f} R_j(C, F_j)}{1 + f} \tag{3}
\]

The key idea of solving the problems described in section 3.2.3 is to extend a surfing 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 surfing user’s vote history indirectly. However, due to the fact that 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 surfing user’s vote history directly. Similarly, we assume that a surfing user \( U \) with vote history \( V H_U \) has \( f \) friends in the system, denoted by \( \{F_j\}_{j=1}^f \); moreover, each friend \( F_j \) has the vote history \( V H_{F_j} \). In this direct extension scheme, the service provider first extracts the vote histories of the user \( U \)’s friends from the centralized vote history database, then we can perform an averaging on these friends’ vote histories as well as the user’s own vote history to compute \( U \)’s extended vote history \( V H_{U}^\prime \), as follows.

\[
V H_{U}^\prime = \text{avg} \left( V H_U, \{V H_{F_j}\}_{j=1}^f \right) \tag{4}
\]

Here, each vote history is treated as a vector, and the \text{avg} function is defined to be a function of computing the averages of nonempty (i.e., non-\( \emptyset \)) values at each position in these vectors.

To illustrate the computation process intuitively, we give an example. Suppose a surfing user \( U \) with local vote history \( V H_U = \{0.8, \emptyset, \emptyset, \emptyset, -0.3, 0\} \) has two friends \( F_1 \) and \( F_2 \) in the system. So, the service provider is able to extract the two friends’ vote histories \( V H_{F_1} \) and \( V H_{F_2} \) from the centralized vote history database.

\[
\begin{align*}
V H_{F_1} &= \{ \emptyset, -1, 0.6, \emptyset, -0.2, \emptyset \} \\
V H_{F_2} &= \{0.8, \emptyset, 0.5, \emptyset, -0.3, \emptyset\}
\end{align*}
\]

Thus, the final extended vote history \( V H_{U}^\prime \) is

\[
V H_{U}^\prime = \{0.8 + 0.8, -1 - 1, 0.6 + 0.5, \emptyset, -0.3 - (-0.2) + (-0.3), 0\} \\
\approx \{0.8, -1, 0.55, \emptyset, -0.27, 0\}
\]

Due to the fact that the friends are usually trustworthy and share the similar interest with the surfing user, the service provider can apply the above direct vote history extension scheme to enrich the surfing user’s vote history reliably. Based on such extended vote history \( V H_{U}^\prime \), the service provider can perform the reputation computation as described in section 3.2.2 to compute the final reputation score of each potential next-click item normally. Note that, the surfing user could virtually experience much more actually 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.

3.3.2 Efficient Estimation

From the previous description of vote history extension, each time the service provider executes the reputation computation, she should rely on the surfing user’s friends’ vote histories indirectly or directly. Note that, a user and her friends usually share the similar interest, and moreover, most user browse actions result from following friend links. These two observations imply that the items a surfing user tends to browse may have been browsed and voted by her friends, with relatively high probability.

Considering the above implication, in our design, before performing the actual reputation computation, if many friends of the surfing 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 surfing user again; as an alternative, the service provider can estimate the reputation score.

Without loss of generality, we assume that the service provider wants to compute the reputation score of a potential next-click item \( C_i \) for a surfing 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.

\( ^1 \)In practice, the vote history can be stored in a more compact way.
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 \( \delta \) 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 (\( f' \geq 4 \) in our design) of friends have voted the item identically (\( \delta < 0.1 \) in our design), the service provider can reliably utilize the associated votes to efficiently estimate the reputation score \( R'_{(C_{i}, U)} \) of the item \( C_{i} \).

\[
R'_{(C_{i}, U)} = \frac{\sum_{j=1}^{f'} v_{ij}}{f'} \quad \text{if} \quad \begin{cases} f' \geq 4 \\ \delta = \frac{\sum_{j=1}^{f'} |v_{ij} - R_{(C_{i}, U)}|}{f'} \leq 0.1 \end{cases}
\]

(5)

Here, the \( R'_{(C_{i}, U)} \) can be treated as the final reputation score \( R_{(C_{i}, U)} \) of the item \( C_{i} \), from the surfing user \( U \)’s perspective.

Specifically, with low probability, a malicious user may masquerade as like-minded user to become the surfing user’s friend, and a friend may also be compromised. To address this “malicious friend” problem, the service provider should first compute the correlation coefficient \( \text{sim}_{(U, F_{j})} \) between the surfing user \( U \) and each associated friend \( F_{j} \). If \( \text{sim}_{(U, F_{j})} < 0.5 \), the friend may be malicious or uncorrelated, so we choose not to take this friend’s vote into account.

3.3.3 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, as follows.

“New”-resistant. The key idea to solve the general “new” problem, including the inactive (or new) user problem and the unpopular (or new) content problem described in section 3.2.3, is to extend the surfing user’s vote history reliably. Since friends are usually trustworthy and have the similar interest, in our social reputation model, the service provider utilizes a surfing user’s friends’ vote histories to reliably extend the user’s own vote history indirectly or directly.

For inactive users or unpopular content, the number of items co-voted by both the (inactive) surfing user and the (unpopular content) item’s associated voters should be much larger through taking into account the surfing 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, the surfing user has to resort to an ad-hoc or experiential reputation estimation, as used in current online social networking systems. Such case is unavoidable because there are no associated votes.

Sybil-resistant. In online social networks, Sybil [6] users could create a large number of identities but few friend relationships with genuine users [23, 24]; 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 the surfing user’s and her friends’ vote histories; therefore, our social reputation model is Sybil-resistant. Complementarily, we could further utilize friend links to construct SybilGuard-style [24] “random routes” to defend against Sybil attacks more specifically and effectively.

Therefore, our social reputation model substantially enhances the performance of our previously proposed basic reputation model. Note that, if a surfing 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.

3.4 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.

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

Storage Overhead. On the other hand, considering the vital privacy issues, the service provider should maintain each participating user’s vote history in a centralized vote history database\(^2\) for supporting our proposed reputation model. Similarly, compared with the massive volume of originally maintained content (e.g., videos, photos, etc.), the storage cost of maintaining these vote histories will

\(^2\)Alternatively, this vote history database could be maintained by a set of centralized service providers in a DHT [17] way for load balancing.
also not aggravate the system’s storage overhead significantly.

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

During the reputation computation process, computing the similarity between the surfing user and each of these potential next-click items’ associated voters is relatively expensive (see expression 1). Fortunately, most users browse content by following friend links, so that the surfing 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 participating 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 surfing 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 the similar interest, and moreover, most users browse content via following friend links, so that an item a surfing 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.

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

## 4. EVALUATION

In this section, we first describe the experimental setup, and then we present the key performance metric. Finally, we comparably evaluate the performance of our proposed social reputation model in the online social networking system.

### 4.1 Experimental Setup

The goal of our social reputation model is to guide users to browse the desirable content in the 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 participating 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 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.

**User Model:** We use the Flickr trace (1,846,198 users and 22,613,981 friend links, Jan 2007) to generate the social/friend network. Specifically, since this trace does not indicate each participating user’s genuineness and interest, we have to perform an appropriate processing on the generated social network.

According to the small-world property of online social networks [2, 13], we generate 50 genuine user groups for the social network as follows. We first randomly select 50 bootstrapping users as the seeds of these genuine user groups, and then use the breadth-first search algorithm to extend these genuine user groups to guarantee that each group contains a different number of unique users (Zipf distribution with its parameter $\alpha = 1$). Note that, if we cannot generate a required number of genuine users from a bootstrapping user, we randomly select another bootstrapping user to replace this one. Except these users contained by genuine user groups, the other users are spammers whose task is to raise the reputation of spam content and abuse genuine users’ attention.

Moreover, we further assign various interests to different users. In a macroscopic view, each user in the online social networking system has her own unique interest; while in a microscopic view, a user usually shares the similar interest with her direct friends who also share the similar interest with their own direct friends, and so forth. With the growth of the length of friend links, the interest difference between the newly reached users and the original user increases gradually. Concretely, in our experiments, we first randomly generate 50 real numbers in $[-1, 1]$ as the *interest scores* of the aforementioned 50 bootstrapping users. Then, when we execute the breadth-first search algorithm, we assign each newly reached user an interest score in $[-1, 1]$ according to the normal distribution with a mean of her friends’ assigned interest scores and a variance of 0.01. This process assigns each genuine user an interest score, and
finally, we assign each unreached user (i.e., spammer) a random number in $[-1, 1]$ as her interest score. So now, each user corresponds to an interest score, and the difference between two users’ interest scores reflects the diversity of the two users’ interests.

**Content Model:** In large-scale online social networking systems, there exists a massive number of (content) items with various different tags. In our experiments, there are $10^8$ unique items, $60\%$ of which are spam items; moreover, there are $10^5$ unique tags, each of which corresponds to a number of items following Zipf distribution with $\alpha = 0.8$ (similar to the parameter in current content distribution networks).

To model a highly malicious environment, each genuine user shares 50 authentic (i.e., not spam but maybe desirable or irrelevant to other genuine users) items, and each spammer shares 400 spam items. Further, each content is randomly assigned an interest score in $[-1, 1]$, and the probability of a user with interest score $s_u$ sharing an item with interest score $s_c$ is inversely proportional to $|s_u - s_c|$.

**Execution Model:** Surfing actions are initiated at randomly distributed users following Zipf distribution with $\alpha = 0.5$. Specifically, an experiment is composed of 50 (experimental) days, and each day is divided into $5 \times 10^6$ surfing cycles. In each surfing cycle, a user may browse content via following friend links, using search facilities or even following the links indicated by external sources [13].

Suppose a genuine user with interest score $s_u$ browsed an item with interest score $s_c$. If the browsed item is spam, the genuine user gives a vote of $-1$ on this item; otherwise, the vote given by this genuine user should follow the normal distribution with a mean of $1 - |s_u - s_c|$ and a variance of 0.01. This actually reflects the interest difference between the genuine user and the browsed item. In particular, a genuine user votes on her browsed items with certain probability, and we further add noise to model users’ mistakes through making genuine users vote correctly with only $90\%$ probability and randomly otherwise. On the other hand, for a spammer, she gives a malicious (i.e., opposite) vote on each browsed item also with only $90\%$ probability and randomly otherwise.

### 4.2 Performance Metric

Usually, when a genuine user surfs on an online social networking site, she may merely try to browse several top reputed items listed on each visited page. In our experiments, we characterize the system performance using the **precision of the top 10 reputed items**, which is defined as the fraction of desirable items existing in these top 10 reputed items. Here, “desirable” means that the interest difference between the surfing user and the reputed item is below 0.2. Specifically, this metric is computed at the end of each (experimental) day.

### 4.3 Evaluation Results

In this section, we evaluate the performance of our proposed social reputation model. Specifically, in our experiments, a surfing user browses content via either following friend links or using search facilities. Unless otherwise stated, the total fraction of spammers is set to 0.2, and a genuine user gives a vote on her browsed item with a probability of 0.9.

#### 4.3.1 Browse via Following Friend Links

In this browse mode, each participating user first browses one of her friends’ content, then the friend-of-friend’s content, and so forth. In our experiments, the length of such browse path is distributed randomly in the range of $[3, 20]$.

**Comparison.** We first comparably evaluate the performance of the following four models with the default parameters.

- **Social Reputation Model-D:** Our basic reputation model with direct vote extension plus efficient estimation.
- **Social Reputation Model-P:** Our basic reputation model with proxy-based vote extension plus efficient estimation.
- **Basic Reputation Model:** Our basic reputation model without any social enhancement.
- **Baseline Model:** In such model, the reputation score of an item is the unweighted average of all the associated votes given on this item.

As shown in Figure 1a, the “Social Reputation Model-D” outperforms all other models with around 94% of these top 10 reputed items being desirable. Moreover, the two social reputation models converge faster than the “Basic Reputation Model”. This indicates that, though each user has only a small number of past votes at the startup, the user in socially-enhanced systems could additionally consider her friends’ vote histories, directly or indirectly, to help identify desirable content more accurately.

Somewhat interestingly, the “Baseline Model” works well at the startup, but its performance decreases rapidly. This is because that, at the startup, an item which tends to be browsed by a user may merely be browsed and voted by the user’s close friends with high probability, and naturally these associated votes should be able to reflect the user’s own interest, so that the unweighted-averaging-based reputation computation employed by

---

[3] Actually, an extra small part of browse actions result from following links indicated by external sources; however, due to the uncertain feature, we are not able to model such browse actions in our experiments.
the “Baseline Model” could compute the accurate personalized reputation score; however, as time goes by, an item may be browsed and voted by many spammers and users with different interests, then these associated votes should become increasingly less reliable, thus influencing the performance of “Baseline Model” significantly.

Furthermore, we repeated the above experiment based on YouTube trace (1, 157, 827 users and 4, 945, 382 friend links, Jan 2007). We get the extremely similar experimental result, and find that our two social reputation models in the network with YouTube trace converge a little slower than in the network with Flickr trace. The reason is that each user in YouTube has 4.27 friend links on average which is smaller than 12.25 in Flickr. This phenomenon also implies that, in order to achieve a good performance, a user actually does not need too many friends. We will revisit this phenomenon later.

Specifically, since our “Social Reputation Model-D” always outperforms the three other models in the following experiments, we hereafter choose to merely present the results of “Social Reputation Model-D”.

Impact of Spammers. In this experiment, we investigate the influence incurred by different fractions of spammers. Interestingly, as shown in Figure 1b, the different fractions of spammers have, however, no significant impact on the performance of our social reputation model. Our analysis takes two factors into consideration. First, in this browse mode, a user browses other users’ content via following friends links, so that an item tends to be browsed by a user may be browsed and voted by her friends before, thus the service provider is able to perform an accurate reputation computation for the user; second, the spammers cannot easily build friend relationships with genuine users, thus a genuine user could browse spammers’ content via following friend links with low probability. Besides, the experimental result further implies that our social reputation model can help users identify desirable content even under Sybil [6] attacks where a large number of participating users being controlled by Sybil users.

Impact of Friends. In an online social networking system, a user may have many friends, and considering all friends’ vote histories may incur much computation overhead on the system. In this experiment, we limit the maximum number of considered friends during the process of reputation computation. Figure 1c shows that, in this browse mode, the service provider actually does not need to consider all of a surfing user’s friends, and only a small number of friends (e.g., 4) are enough to compute the accurate personalized reputation score.

Impact of Voting. Many current reputation models are penalized by the lack of users’ voting behaviors. Figure 1d shows that our social reputation model
can work well with various different voting probabilities; moreover, even if genuine users are very inactive to vote their browsed items (e.g., voting probability = 0.2), our proposed model can still work well. This benefits from our vote history extension scheme which could extend a user’s vote history by considering her friends’ vote histories. The experimental result validates that our social reputation model is sparsity-resistant. Specifically, in current online social networking systems, only a few users may vote their browsed items actively, so this experimental result further indicates that our social reputation model can indeed be deployed in practice. Note that, if voting probability = 0, our social reputation model would fall back to an ad-hoc or experiential selection scheme; in fact, any existing reputation models cannot work with such voting probability.

**Inactive Users and Unpopular Content.** Current reputation models could hardly have the sufficient capacity of handling the problems of inactive users and unpopular content. In the following experiments, we first classify the users into three categories: *active users* (i.e., the number of whose browse actions ranks top 20%), *inactive users* (i.e., the number of whose browse actions ranks bottom 20%) and *common users*; then, we similarly classify the content into three categories: *popular content* (i.e., whose browse count ranks top 20%), *unpopular content* (i.e., whose browse count ranks bottom 20%) and *common content*. Figure 1e indicates that, no matter active or inactive users, our social reputation model can provide them with the similar capacity of identifying desirable content. Moreover, we can see from the Figure 1f that our model can also effectively help users identify the interests of content even these unpopular content.

### 4.3.2 Browse via Search Facilities

In this browse mode, the selection of a specific item to browse is done by first inputting a tag\(^4\) according to the popularity of the tag (i.e., proportional to the number of the tag’s associated items) into the search facility of online social networking system, and then choosing an associated item from the returned search result to browse based on our proposed reputation models.

**Comparison.** In this mode of browsing via search facilities, we comparably evaluate the performance of the aforementioned four reputation models. Figure 2a illustrates that “Social Reputation Model-D” significantly outperforms all other three models; the two social reputation models can converge faster than “Basic Reputation Model”, which validates the effective-

\(^4\)Though we merely specify the tag-based search in this paper, we are also able to support some other kinds of search, e.g., keyword-based search and title-based search, in a similar way.
ness of our proposed social enhancements; moreover, the “Baseline Model” cannot work well. Since our “Social Reputation Model-D” always outperforms the three other models in the following experiments, we choose to merely present the experimental results of “Social Reputation Model-D” hereafter.

**Impact of Spammers.** We vary the fraction of spammers, and investigate the influence incurred by spammers. Figure 2b shows that the performance of our social reputation model decreases slightly with the growth of the fraction of spammers, and our model can help users identify the desirable content even in a highly malicious environment with 30% of users being spammers. In real-world online social networking systems, some spammers may perform Sybil [6] attacks to create a large number of virtual spammers, and the above experimental result validates the analysis described in section 3.3.3 that our social reputation model is Sybil-resistant. Of course, we are able to utilize friend links to construct the SybilGuard-style [24] “random routes” structure here to defend against Sybil attacks more specifically and effectively.

**Impact of Friends.** Now, we limit the maximum number of considered friends during the process of reputation computation. Figure 2c shows that such limitation has no significant impact on our performance, so that the service provider merely needs to consider a small part of a surfing user’s friends to compute the accurate personalized reputation score. Complementary with the analysis in section 3.4, the experimental results shown in Figures 1c and 2c together validate that our social reputation model will not significantly aggravate the overhead of the online social networking systems.

**Impact of Voting.** As shown in Figure 2d, we evaluate the performance of our social reputation model with the voting probability varying in steps of 0.2. The experimental result illustrates that, in the mode of browsing via search facilities, the voting probability could influence our performance especially when the voting probability is very low (e.g., 0.2). Fortunately, as analyzed in section 3.2.3, our reputation model is able to provide a strong incentive for users to give votes on their browsed items frequently and accurately, so we expect that the voting probability would not be too low in realistic online social networking systems with the deployment of our social reputation model.

**Inactive Users and Unpopular Content.** Finally, we evaluate whether our social reputation model can handle the problems of inactive users and unpopular content. Figure 2e indicates that our social reputation model is indeed able to help inactive users identify desirable content, and moreover, Figure 2f indicates that our model can help users identify the interests of all kinds of content including the unpopular content. Therefore, our social reputation model can effectively handle the problems of inactive users and unpopular content, in the mode of browsing via search facilities.

5. **DISCUSSION**

We now discuss several possible design choices, as follows.

**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 the 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. Specifically, the users in each interest group may have a specific interest, and moreover, these users need not necessarily link to each other in the online social network; thus, considering a surfing 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.

6. **RELATED WORK**

6.1 **Online Social Networks**

Online social networking systems are experiencing explosive growth, both in terms of involved communities and overall participating users. Despite the different purposes of various online social networking systems, the underlying social networks have been shown to exhibit the similar power-law, small-world and scale-free properties [2, 13].

Currently, some proposals focused on improving the performance of existing online social networking systems. For instance, Guha et al. in [9] utilized the pseudorandom substitution technique to preserve user privacy while allowing users to advertise the private information to their extended social network; and in [18], Tootoonchian et al. decoupled social networking information from content sharing, and proposed an access control scheme, Lockr, to eliminate the need for users to manage many system-specific social networks.

On the other hand, due to the fact that the social networking technique has significantly impacted how Inter-
net users make use of the today’s Internet, some proposals have been made to improve existing distributed applications by exploiting the inherent properties of social networks. PGP [25] is one of the early social networking applications, in which the participants create a “web of trust” to authenticate public keys based on their acquaintances’ opinions in a fully self-organized manner. This “web of trust” model utilized the friend-of-friend trust structure, and then Garriss et al. in [8] adopted the similar structure to develop the “Reliable Email (Rx)” , an automated whitelisting-based email acceptance system, to mitigate spam emails. Re: exploits social relationships among email correspondents, and moreover, it does not incur false positives among socially connected users. In [12], Mislove et al. tried to combine the information contained in both hyperlinks and social links, so they merged the social networking technique into the Web search engine to optimize the ranking results by considering various interests of different users.

To defend against Sybil attacks [6], two promising decentralized protocols, SybilGuard [24] and SybilLimit [23], have been proposed. They utilize the social networks among user identities based on the fact that Sybil users could create many identities but few trust relationships; finally, the two protocols have the capacity of allowing only very limited Sybil users to be accepted even in a large-scale network. Furthermore in [14], based on the similar fact that it is difficult for a user to create arbitrarily many trust relationships, the Ostra system explores the use of existing social links to impose a cost on the information senders, thus preventing the adversary from sending excessive unwanted communication. Recently, Ramachandran and Feamster in [15] proposed a framework called Authenticatr to establish authenticated out-of-band communication channels between applications by utilizing social links existing in various online social networking systems; and in [22], Yardi et al. presented Lineup, a lightweight CAPTCHA-like [19] authentication mechanism that challenges a user with her tagged friend photos to authenticate that user’s identity or membership.

6.2 Reputation Models

So far, many reputation models have been proposed to identify the spam content in networked systems. In general, these reputation models can be grouped into three main categories: peer-based models, object-based models and hybrid models.

In peer-based reputation models, e.g., EigenTrust [10] and PeerTrust [21], to reflect the level of honesty, each participating user is assigned a reputation score by considering her past behaviors in pairwise transactions. According to the reputation score, genuine users could collectively identify content spammers, and then isolate these spammers from the system. However, the studies in [7, 20] evaluated the potential impact of peer-based reputation models, and implied that these models are sensitive to the dynamic changing of peer behaviors as well as many other factors unrelated to malicious behaviors.

Among current object-based reputation models, Credence [20] is the typical representative of them. In Credence, genuine users determine the object authenticity through secure tabulation and management of endorsements from other users. This model designs a decentralized flow-based trust computation to discover trustworthy users. Compared with our proposed social reputation model, current object-based reputation models target only spam content but not the irrelevant content from each user’s perspective; moreover, they are vulnerable to various aforementioned problems.

Aiming at combining the benefits of both peer-based and object-based models, several hybrid reputation models, e.g., XRep [5] and X2Rep [4], have been further presented. Nevertheless, due to the fact that most of the participating users in existing networked systems are rational in seeking to maximize their individual utilities, current reputation models are penalized by the lack of reliable user cooperation. Comparably, our social reputation model is able to provide a strong incentive for users to give votes on their browsed items more often and accurately, thus enhancing the system performance significantly.

7. CONCLUSION

We have described, analyzed and evaluated our social reputation model, which distinguishes different users’ interests and exploits the inherent friend relationships to make participating users be armed with the knowledge of how to browse desirable content in online social networking systems. Our social reputation model provides a strong incentive for participating users to give votes frequently and accurately, and it will not aggravate the system overhead significantly. Moreover, our proposed model is able to address the problems of inactive users, unpopular content and Sybil attacks. The analysis and prototype-based evaluation indicate that our social reputation model can indeed be deployed in practical online social networking systems.

Actually, many techniques in our social reputation model are not specific to the online social networking systems, but are applicable to numerous other content sharing systems and e-commerce systems. The detailed checking of such applicability will be left to our future work.

8. REFERENCES


