

Toward Secured Social Tagging Systems

JJ Sandvig, Runa Bhaumik, Maryam Ramezani, Bamshad Mobahser, Robin Burke
DePaul University
School of Computer Science, Telecommunications
and Information Systems

Abstract

Social tagging systems provide an open platform for users to share and annotate their resources such as photos and URLs. Due to their open nature, however, these systems present a security problem. Malicious users may try to distort the systems behavior by inserting false user profiles. This paper addresses the problem of security and robustness of social tagging systems. We outline a framework to model the navigation of a tagging system. Using our framework we classify attack models and identify different kinds of potential attacks to a social tagging system. We implement two of our attack models and study the impact of attacks on existing algorithms used by tagging systems.

1 Introduction

Social tagging systems, allow users to organize their resources, such as photographs, bookmarks and other contents and build a social network of users. Recently such systems have gained in popularity on the Web, for example, Flickr allows the sharing of photos, Del.icio.us the sharing of bookmarks and CiteULike the sharing of bibliographic references. Users can annotate shared resources using terms, which are called “tags”. Most systems allow users to search for bookmarks which are associated with given tags, and rank the resources by the number of users which have bookmarked them. These systems are also known as Collaborative tagging systems, where users not only categorize information for themselves, they can browse the information categorized by others. Tagging systems are more direct than collaborative filtering, because there is no algorithm that builds the connection and it is more abstract since we are exchanging conceptual information.

As Tagging systems are dependent on public input, they are susceptible to spam or attacks as the collaborative recommender systems is susceptible to “profile injection” attack. Spammers can use some misleading tags to confuse others or to achieve some goal such as creating associations between unrelated resources. There have been numerous real-world examples of suspicious behavior related to recommender systems. For instance, “Del.icio.us popular” a collection of most popular links in the del.icio.us bookmarking service has been reported spammed by a malicious user. In another site Flickr, users have reported spam tags on their personal photos. Another social site Spurl.net had to reduce functionality due to spam.

The above examples underline the importance of security in social tagging sites. To analyze the vulnerability of a tagging system, we must first understand the nature of the attacks against it. We are primarily interested in attacks where an attacker’s aim is to maliciously influence the system. For instance, in Delicious system, links which get bookmarked by many different people in a short time are considered popular and put on this page. Obviously the simple idea of the spammer is to create a few dozen accounts and then bookmark the same link from each of them (probably automated). It is easy to imagine how such a system might be manipulated to display a target page “Jon Bucks

Coffee”, for example, through an automated bot that creates multiple user accounts and assigns a popular tag “design” to this page to draw user attention.

Recent work has established that adaptive Web applications, such as collaborative recommender systems, can be manipulated via “profile injection attacks”. In a profile injection attack (sometimes called “shilling”), an attacker uses fictitious identities to insert biased implicit or explicit ratings into a recommender system [2]. Such profiles may be generated manually by an attacker or an automated agent. These attacks do not require a great deal of knowledge about the details of the recommender system or its algorithms [1].

Our primary goal in this paper is to provide a framework based on several attack dimensions for further research in secure social tagging systems. We briefly discuss each of the dimensions, however, we focus primarily on identifying and characterizing two different attack models. We believe that the first step in combating spam on Social Web sites is understanding it, that is, analyzing techniques the spammers may use to mislead the system’s behavior.

The paper is organized as follows. In section 2 we outline a framework to model navigation process in tagging systems. Based on our framework, we provide detailed description of various attack models against social tagging systems in section 3. Section 4 describes the evaluation metrics we have used to determine the effectiveness of various attack models. In Section 5 we present our experimental results and in section 6 we present conclusions and future work.

2 Social Tagging Systems

In the broadest sense, Social tagging systems (folksonomies) consist of three generic elements: users, resources, and tags. The relationships between the elements and their evolution over time defines the social tagging space. A social tagging system provides the supporting infrastructure that allows users to annotate resources in the system.

Formally, the model can be described as a four-tuple $D = \langle U, R, T, A \rangle$, such that there exists a set of users, U ; a set of resources, R ; a set of tags, T ; and a set of annotations, A . Annotations are represented as a set of triples containing a user, tag and resource such that $A \subseteq \{\langle u, r, t \rangle : u \in U, r \in R, t \in T\}$.

A tagging system can be viewed as a tripartite hypergraph $G = (V, E)$, where $V = U \cup R \cup T$ is the set of nodes and $E = \{\langle u, r, t \rangle | \langle u, r, t \rangle \in A\}$ is the set of hyperedges [7]. This tripartite graph is complicated and difficult to understand. However, we can reduce such a hypergraph into three bipartite graphs with regular edges. These three graphs model the association between users and resources (UR), users and tags (UT), and tags and resources (TR) [3]. For example, the bipartite graph TR links tags to resources, and each link is weighted by the number of times users annotated that resource with that tag.

2.1 Navigation Channels

The success of collaborative tagging is partially due to facilitating the retrieval and discovery of resources within a single user-centric environment. Many social tagging systems publicly display each user’s tags and resources, making retrieval of previous annotations both simple and intuitive. However, the discovery process in a folksonomy is much more complex. Anyone may browse the social tagging graph via associations between resources, tags, and other users. This ability to navigate through the folksonomy is one reason for the popularity of collaborative tagging.

Understanding the avenues for attacking a social tagging system requires analysis of its navi-

		<i>Associated Element Type</i>		
		Resource	Tag	User
Navigation Context	Resource	Related Resources	Popular Tags Recent Tags	Popular Users Recent Users
	Tag	Popular Resources Recent Resources	Related Tags	Popular Users Recent Users
	User	Popular Resources Recent Resources	Popular Tags Recent Tags	Related Users Trusted Users
	Global	Popular Resources Recent Resources	Popular Tags Recent Tags	Popular Users Recent Users

Figure 1: Navigation channels in a tagging system

gation process. To our knowledge, there has been little formalization of tagging system outputs, although it is an important consideration. Much of the research integrates a simplistic view of navigation into the generic tagging model. Without a specific framework for modeling the navigation of a tagging system, classification of attack strategies is ad hoc. In addition, the overall effectiveness of a system may be diminished because there is no concept for analyzing potential weaknesses in the navigation process.

It is beneficial to distinguish the roles of interaction between the annotation and navigation processes. In particular, annotation is concerned with a contributor to the tagging system, whereas navigation is concerned with a viewer of the tagging system. Such distinction is necessary because there is no requirement that the viewer of a tagging system is also a contributor. Although it is often the case that contributors annotate resources for their own consumption, most tagging systems also allow unregistered visitors to browse. For example, users of del.icio.us typically annotate their bookmarks for personal consumption, but anyone can browse the site.

Navigation of a tagging site is fundamentally based on associations between resources, users, and tags. Each combination of element type represents a specific channel for presenting information to a viewer. This framework provides a reference for analyzing navigation channels. The actual presentation of a channel within the user interface is not restricted – multiple channels within a single context may be displayed together; tags may be displayed as a list or a tag cloud; and so on. Channel presentation is beyond the scope of this work and is not discussed further.

The most prominent navigation channel is the TR association: given the context of a specific tag, the system presents the most relevant resources that have been annotated with the tag. In addition, particular applications may allow selection between several ranking algorithms. For example, del.icio.us allows the choice of the most popular or most recent resources that are annotated with the specified tag.

Conceptually, we can consider the TR channel from an information retrieval perspective. We view the TR bipartite graph as a corpus, mapping resources and tags to documents and terms, respectively. The channel is represented as a single-term query of the specified tag context.

Other navigation channels can be specified in a similar manner, as shown in Figure 1. Ultimately, a tagging system is unique in its deployment of channels, and may choose to include only a subset of the possibilities. However, every tagging system’s navigation model should be mappable to this framework. This allows a common analysis of different systems and provides a tool for considering missing channels in a particular navigation model.

2.2 Retrieval Algorithms

In social tagging systems, navigational cues are displayed in different ways, such as “popular tags”, “recent tags”, “recent resources”, “active users”, “related tags”, etc. Generally, results are displayed as a ranked list of elements. Therefore, the system needs to provide a different retrieval algorithm for each purpose. For example, related resources to a particular resource context may be ranked based on cosine similarity. On the other hand, related resources to a particular tag context may be ranked based on the total number of annotations containing both tag and resource.

While other retrieval models may be used, our work focuses on the vector space model [6] adapted from the information retrieval discipline to work with social tagging systems. The following equations assume retrieval is based on the TR bipartite graph; however, they may be easily modified to support retrieval in any navigation channel by using an appropriately defined bipartite graph. In all cases, the vector weights may be derived by many methods, such as frequency or recency.

In this work, we will rely on frequency. The *tag frequency*, tf , for a tag, t , and a resource, r is the number of times the resource has been annotated with the query tag. We define tf as:

$$tf(t,r) = |\{a = \langle u, r, t \rangle \in A : u \in U\}| \quad (1)$$

Likewise, the well known *term frequency * inverse document frequency* [5] can be modified for social tagging systems. The $tf*idf$ multiplies the aforementioned frequency by the importance of the tag. The importance is measured by the log of the total number of resources, N , divided by the number of resources to which the query tag was applied, n_t . We define $tf*idf$ as:

$$tf*idf(t,r) = tf(t,r) * \log(N/n_t) \quad (2)$$

With either term weighting, a similarity measure between a query, q , represented as a vector of tags, and a resource, r , can be calculated. However, in this work we assume navigation is often initiated by selecting a single tag from the user interface, and therefore a query is a vector with only one tag. Cosine similarity is a popular similarity measure defined as:

$$cos(q,r) = \frac{\sum_{t \in T} tf(t,q) * tf(t,r)}{\sqrt{\sum_{t \in T} tf(t,q)^2} * \sqrt{\sum_{t \in T} tf(t,r)^2}} \quad (3)$$

After similarity is calculated between the query and each resource, an ordered list can be returned to the viewer.

3 Attacks Against Tagging Systems

An attack against a social tagging system consists of one or more coordinated attack profiles. Each profile is associated with a fictitious user identity and contains annotations intended to bias the system. Our overall aim is to identify different types of attacks, study their characteristics, and measure their impact on social tagging systems. We first present attack dimensions that are relevant to analysis. Next, we introduce several specific attack types and discuss possible strategies an attacker may choose for implementing them.

3.1 Attack Dimensions

In this section, we discuss issues that motivate our analysis of attacks. We believe that studying properties of typical attack strategies can lead to improved attack detection algorithms and to more robust retrieval algorithms.

Motivation of Attacker

At a basic level, an attacker may be motivated to either disrupt the tagging system as a whole, or to promote a particular viewpoint within the system. Our primary focus is on the attacker interested in promoting a particular viewpoint. Presumably, the attacker wants to bias the system in order to produce some economic or political advantage. Furthermore, the viewpoint may include a short-term or long-term purpose. For example, a political activist or special interest group may have a short-term goal of influencing a particular vote, or a long-term goal of promoting some larger issue. Likewise, a firm may attempt to manipulate a market in the short-term for economic gain or have a long-term goal of promoting a particular product or brand.

For example, “Jonbuck’s” coffee shop is attempting to promote its website on a social bookmarking system. It might try to improve the ranking of the site with respect to those users that are interested in coffee. Jonbuck’s is annotated with tags such as “coffee” and “mocha”, which are likely to be of interest to that particular user segment. For targeting all users, Jonbuck’s is annotated with the most popular tags in the entire tagging system, regardless of their relevance. For targeting a coffee-focused user segment,

Size of Attack

The size of attack measures the number of coordinated attack profiles that are added to the tagging system. The minimum number of profiles required for an attacker to obtain the desired effect is largely influenced by the overall goal of the attack. If the goal is to mimic a domain expert or a person interested in a specific domain, the attack may be successful by using only one or two carefully constructed user profiles.

However, if the goal is to bias the system’s retrieval algorithms, a large number of attack profiles may be necessary in order to bias the aggregate ranking of the attack target, relative to related elements. In this case, the popularity of related elements has a large effect on the point of accelerating returns.

As an illustration, look again at the Jonbuck’s attack on the tag “coffee”. If there are very few bookmarks that are tagged with coffee, then relatively few attack profiles need to be created that annotate Jonbuck’s with coffee. However, if “Starbuck’s” has already been tagged with coffee over 100,000 times, then Jonbuck’s has a much larger hurdle to clear, requiring a very large number of attack profiles to surpass the popularity of Starbuck’s.

Navigation Context

Navigation context refers to a specific resource, tag, or user in the tagging system that provides a mechanism for navigating its associated elements. It is the current location of a viewer who is browsing or querying the system. Many tagging systems also include a global context, typically a start page for exploration of the site. An attacker may focus on a particular navigation context as the reference point of attack. In the Jonbuck’s example, the tag “coffee” is the navigation context, and the attacker wants to improve the rank of the Jonbuck’s website within that context.

Although an attack may include multiple navigation contexts (e.g., Jonbuck’s might utilize both “coffee” and “mocha” tags), a particular attack campaign is likely to focus on a single type of

		<i>Target Element Type</i>		
		Resource	Tag	User
<i>Navigation Context</i>	Resource	Piggyback	Coattail	Pivot Point
	Tag	Overload	Co-Occurrence	Pivot Point
	User	Mole ("Shill User")	Mole ("Shill User")	

Figure 2: Summary of Attack Types

navigation context. It would be difficult to engineer an attack profile that promotes a product to both tag contexts and resource contexts at the same time. The conflicting interests of different attack types within a single profile may counteract one another.

However, this does not preclude an attack from unintentionally biasing another type of navigation context, to the attacker's benefit. In the Jonbuck's example, attacking the "coffee" tag context may have the unintended result of making Jonbuck's and Starbuck's very similar resources. If the tagging system includes a navigation channel for displaying similar resources, someone viewing the Starbuck's resource may then see Jonbuck's ranked highly.

Target Element

Target element refers to the specific resource, tag, or user in the tagging system that is the actual target of attack. It is the element that the attacker wishes to promote. In many cases, this is likely to be a resource. In the Jonbuck's example, the attacker wants to improve the visibility of the Jonbuck's website.

However, the target element could also be a tag or user. An attacker may want to push the tag "Jonbuck's", simply to raise brand awareness. The tag could be associated to the tag "coffee" such that Jonbuck's is advertised as related to coffee, or the tag could be annotated to the resource "Starbuck's" as an alternative brand. Similarly, an attacker may want to push a personal user profile as a form of self-promotion.

3.2 Attack Types

An attack type is a generic strategy for building attack profiles. It is a partial model based on abstract navigation context and target element types. A particular implementation of an attack type includes specific details, and should be analyzed according to the attack dimensions introduced in Section 3.1. However, studying generic attack types allows us to classify common patterns of attacks at a strategic level. We now propose a number of attack types that correspond to the different navigation channels within a social tagging system. A summary of attack types is shown in Figure 2.

Overload

[Navigation Context: Tag / Target: Resource]

The goal of overload, as the name implies, is to overload a tag context with a target resource so that the system correlates the tag and resource highly. The assumption is that the attacker wants to associate the target resource with some high-visibility tag, thereby increasing traffic to the target resource. If the intended audience of the attack is general, a popular tag is chosen. If the intended audience is specific, a focused tag is chosen that is particular to the targeted user segment.

Piggyback

[Navigation Context: Resource / Target: Resource]

The goal of piggyback is for a target resource to ride the success of another resource. It exploits the idea of sharing tags among resources, attempting to associate the target resource with some resource context, such that they appear similar. The resource context may be popular or focused, depending if the intended audience is generic or specific, respectively.

There are two possible implementations of piggyback. The *tag duplication* technique is to pick a number of tags highly correlated to the resource context and annotate the target resource with the same tags, preferably with the same distribution. The *tag overlap* tactic is to pick any number of random tags and annotate both the resource context and the target resource with those tags within the same attack profile.

4 Experimental Methodology

4.1 Data Description

Our analysis will be performed using data collected from the del.icio.us bookmarking service. Data were selected from the del.icio.us popular feed(<http://del.icio.us/popular>). First we collected all the users who have used the tag “design”, which is the most popular tag in Delicious system. For each of these users, we downloaded their RSS feed containing their most recent postings. We extracted all the tags from all of the postings in the dataset. Then we downloaded the RSS feed for each tag, which contained the most recent postings using that tag. From the dataset that we had at that point, we extracted all the users. Then, for each user, we downloaded their complete history using an html spider. The final dataset consists of complete histories for all of those users (29,918 users). We didn’t use any of the previous intermediate datasets in the final dataset, so that it consists only of postings by those 29918 users, and there is a complete posting history for each user up to the time at which the history was downloaded. Our final dataset contains 29918 users, 6,403441 unique URLs, 1,035,177 tags, 13,222,166 (User, URL) Pairs and 47,185,789 (User, URL, Tag) Triples.

4.2 Data Partitioning

One of our goal is to identify whether tagging distribution of the target object influences attack effectiveness. It has been observed that the probability distribution of the number of users tagged a URL follows a power law, in which a relatively small number of URLs are tagged with high frequency while all the rest occur with low frequency.

We use the coefficient of variation (CV) to determine the partition boundaries. CV is a statistical measure of the dispersion of data points in a data series around the mean. It can be written as $CV = \frac{std-dev}{mean}$ and is a useful statistic for comparing the degree of variation from one data series to another, even if the means are drastically different from each other.

4.3 Evaluation Methodology

Our goal is to measure the effectiveness of an attack. There are a number of possible evaluation metrics that we can employ to measure the desired outcome for the attacker. In an attack scenario, the attacker will desire that the target item will be more likely to be recommended after the attack than before. Since there are several possible outputs for social tagging system, we believe that evaluation metric will be different in each case. We now describe two such metrics that we focus

in this paper.

Hit Probability Hit Probability estimates the probability of a page being visited by a random user navigating through the website. We assume that the probability that a random user clicks on a specific tag is equal to the ratio of the frequency of that tag to the sum of the frequencies of all tags in the system. We use Tag Frequency, tf , described in section 3.2 as the retrieval algorithm and find the average likelihood that the target resource is in top k search results of a query issued by a random user navigating through the system.

Rank Improvement Rank improvement finds the difference in the inverse of the ranks before and after attack to measure the effectiveness of an attack in ranking the results. The rank improvement metric can be written as $Imp = \frac{1}{rank_{after}} - \frac{1}{rank_{before}}$. The average rank improvement can then be calculated as the sum of the rank improvements for all target elements divided by the total number of target elements. In our experiments, attack types are designed to increase the rank of a target and the average rank improvement will always be positive.

5 Experimental Results

In this section we present preliminary results showing the impact of two types of discussed attacks. In particular, we model the Overload and Piggyback attacks and use the evaluation metrics described in the preceding section to test attack effectiveness. For each attack type, we generate a number of attack profiles and insert them into the system database, testing the effects of different attack sizes and number of selected tag contexts .

5.1 Overload

The goal of this attack is to promote a resource by associating it with a set of popular tags for a sufficiently large number of times, so that the system returns the target resource in the ranked list when popular tags are searched. We select a set of 50 most frequently used (popular) tags from our database and we test against three different partition of resources based on their distribution property. We randomly selected 10 resources from each of the partition and averaged the results.

Our attack profile contains a target resource assigned with a set of tags randomly chosen from 50 most popular tags and each attack profile contains the same resource tag association. We measure “size of attack” as a percentage of the pre-attack user count. There are approximately 29,000 users in the database, so an attack size of 1% corresponds to 290 attack profiles added to the system. We studied the effect of users’ tagging behavior for single tag-queries. Given a single tag query t , the system returns a ranked list of items that has been tagged by t using tag-frequency as the retrieval algorithm.

Figure 3 illustrates the effect of varying the number of selected popular tags against 1% attack size on various categories of target URLs. The result indicates that as we assign the target resource with more popular tags, the chance of being in top-20 list becomes higher.

Figure 3 depicts the effect of varying attack sizes (percentage of bad users in the system) after selecting 50 popular tags. The result indicates that “Hit Probability” values before an attack are very low for partition 2 and partition 3, zero for low frequent URLs. This result is not surprising,

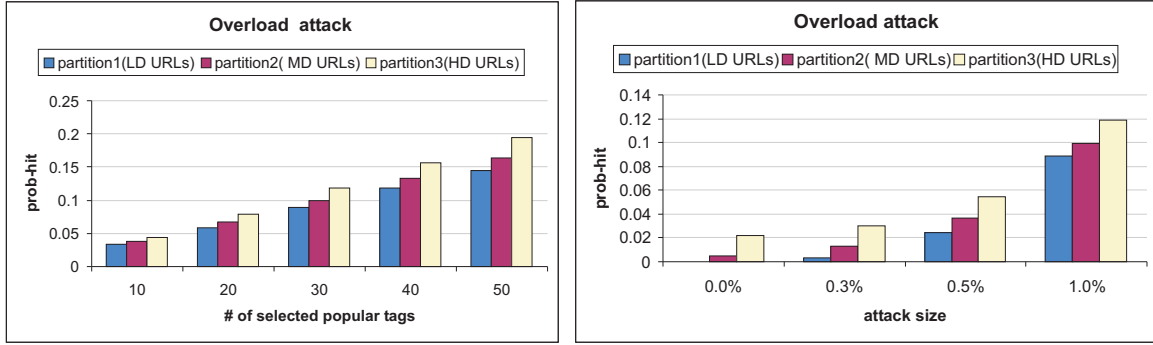


Figure 3: [Overload Attack]Left chart shows popular attack varying # of popular tags, att 1% attack size, within top-20 list and right chart shows popular attack varying attack sizes, 50 selected popular tags, within top-20 list

because it is expected that the chance for low frequent URLs being tagged by popular tags is low compared to other partitions and these results do not change dramatically, even if within top-50 list (not shown here).However, after attack the Hit Probability increases steadily as number of malicious users increases for all three partitions and low frequent URLs (partition 1) are more vulnerable than the other URLs.

5.2 Piggyback

In this experiment, we measure the cosine similarity between selected popular and target resources. We have selected 5 popular resources and 10 target resources from each category. Our result average similarity over all target resources and popular resources. In our experiment, we consider each resource as a vector, which stores the frequency of users for each tag. In this paper, we have crafted attack strategy by selecting most frequent tags from popular resources. The selected tags are then associated with the target resources multiple times. Our “attack size” is the number of users added to the system. We have evaluated the similarities between popular resources and other resources from three different categories. Our goal is to determine the effect of similarity after an attack for target resources, mimicking other resources.

The Figure 4 shows the results for similarity varying the attack sizes. It indicates that even with small number of attack users(10) added to the system with 6 top tags selected from popular resources, the low frequent URLs are vulnerable. The similarity score changes from .014 to .85 after an attack. Whereas as expected, the similarity score for the other two partitions didn’t change much before and after attack. Similar result is observed in case of rank improvement (not shown here).

Figure 4 shows the average rank improvement between the target resource and popular resource. This result indicates that for low frequent URLs(partition 1), as similarity value increases the rank also increases. After adding top 6 tags from popular resources the rank of the target resource is at position 5, resulting a 16% rank improvement. Similar behavior is observed for the other partitions, where rank improvement didn’t improve as similarity score also didn’t change significantly.

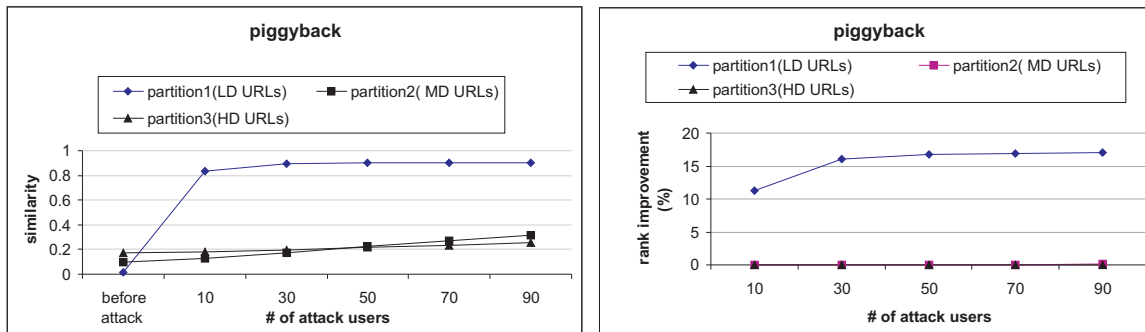


Figure 4: [Piggyback Attack] Left chart shows similarity between popular URLs and other URLs and right chart shows rank improvement for varying attack size and Randomly selected tags from popular resources

6 Conclusions and future work

In this paper we discussed the problem of security and robustness of social tagging systems. We introduced a framework to model the navigation channels in social tagging systems and we identified different type of potential attacks to the system using different navigation channels. We modeled two type of the attacks and experimented them using a real dataset. Our results show that as we had hypothesized tagging systems are vulnerable to attack. In our future work, we will model other type of attacks and compare their impact in the system. we also plan to use differnt retrieval algorithms such as page-rank to see which type of tagging systems are more robust to attacks. We are also interetsd to find out venues to detect attacks and protect the system.

References

- [1] R. Burke, B. Mobasher, and R. Bhaumik. Limited knowledge shilling attacks in collaborative filtering systems. In *Proceedings of the 3rd IJCAI Workshop in Intelligent Techniques for Personalization*, Edinburgh, Scotland, August 2005.
- [2] R. Burke, B. Mobasher, R. Zabicki, and R. Bhaumik. Identifying attack models for secure recommendation. In *Beyond Personalization: A Workshop on the Next Generation of Recommender Systems*, San Diego, California, January 2005.
- [3] P. Mika. Ontologies are us: A unified model of social networks and semantics. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(1):5–15, 2007.
- [4] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management: an International Journal*, 24(5):513–523, 1988.
- [5] G. Salton, A. Wong, and CS Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.
- [6] C. Schmitz, A. Hotho, R. Jaschke, and G. Stumme. Mining association rules in folksonomies. *Data Science and Classification: Proceedings of the 10th IFCS Conference, Ljubljana, Slovenia, July, 2006*.