A Collusion-Resistant Automation Scheme for Social Moderation Systems

Jing-Kai Lou[†‡], Kuan-Ta Chen[‡], and Chin-Laung Lei[†]
[†]Department of Electrical Engineering, National Taiwan University
[‡]Institute of Information Science, Academia Sinica
{kaeaura, ktchen}@iis.sinica.edu.tw, lei@cc.ee.ntu.edu.tw

***Abstract*—For current Web 2.0 services, manual examination of user uploaded content is normally required to ensure its legitimacy and appropriateness, which is a substantial burden to service providers. To reduce labor costs and the delays caused by content censoring, social moderation has been proposed as a front-line mechanism, whereby user moderators are encouraged to examine content before system moderation is required. Given the immerse amount of new content added to the Web each day, there is a need for automation schemes to facilitate rear system moderation. This kind of mechanism is expected to automatically summarize reports from user moderators and ban misbehaving users or remove inappropriate content whenever possible. However, the accuracy of such schemes may be reduced by collusion attacks, where some work together to mislead the automatic summarization in order to obtain shared benefits.**

In this paper, we propose a collusion-resistant automation scheme for social moderation systems. Because some user moderators may collude and dishonestly claim that a user misbehaves, our scheme detects whether an accusation from a user moderator is fair or malicious based on the structure of mutual accusations of all users in the system. Through simulations we show that collusion attacks are likely to succeed if an intuitive count-based automation scheme is used. The proposed scheme, which is based on the community structure of the user accusation graph, achieves a decent performance in most scenarios.

***Index Terms*—Bad Mouthing, Collusion Detection, Fraud Detection, Social Moderation, Web 2.0**

I. Introduction

The emergence of Web 2.0 not only provides individuals with new ways to gain publicity by posting their own writings, photos, and videos on the Internet, but also promotes interpersonal communications over the Internet. With the advent of new-generation Web technologies, notably AJAX [5], users are making substantial contributions to numerous Web 2.0 sites, e.g., Wikipedia, the cosmic compendium of knowledge; YouTube, the million-channel video sharing site; and MySpace, the online metropolis. Among these examples, YouTube attracts about one billion page views everyday, which clearly manifests the popularity of Web 2.0 sites. However, there is a downside to this phenomenon. The core concept of Web 2.0 is that people have the freedom to participate and to share their content on the Internet. While most people behave responsibly, a minority of users may abuse the freedom and upload inappropriate content. For example, they may post pictures that violate copyright laws, upload splatter movies on a video sharing site, or use bots [2] to gain unfair advantages in an online game. Hence, proper *content censorship* is essential for Web 2.0 services that allow users to upload their own content.

Currently, most Web 2.0 sites employ *system moderators*, i.e., they hire official moderators to verify the legitimacy of content uploaded by users. However, *system moderation is labor-intensive* because of the vast amount of user content such that it may become a heavy burden to the service providers. Take the album Flickr as an example. According to [1], it receives $4,320,000$ photos everyday on average. If each moderator can handle 100 photos per hour and works 8 hours per day, Flickr would need to hire $5,400$ moderators working to check all the uploaded pictures.

System moderation is not the only means of ensuring the appropriateness of users' content. *Social moderation has been proposed to solve the content censorship problem* [8]. Under this approach, all users, or only selected users, can report content they consider inappropriate so that system moderators can inspect and evaluate it. In this way, social moderation reduces service providers' labor costs and speeds up content censorship because popular sites normally have many more user moderators than system moderators. However, user-assisted moderation does not solve the problem completely, as system moderators may be overwhelmed by the sheer volume of user reports. If we assume that only 1% of photos uploaded to Flickr are problematic and reported by user moderators, the system moderators would still need to handle approximately $43,200$ reports each day. Clearly, such system moderation requirements would still place a huge burden on Web 2.0 service providers.

One way to further reduce the labor required for system moderation is to eliminate manual inspection by official moderators as much as possible. This is our motivation for proposing *social moderation automation, which automatically summarizes the reports submitted by users.*

We consider a subset of a social moderation system in which user moderation only contributes *one-way, negative votes*. In other words, users are allowed to label content as inappropriate based on their own judgments, but they cannot oppose other users' accusations against certain content. We call this system *an accusation-based social moderation system*. For brevity, we refer to it as a *social moderation system* hereafter. Although the system can be used to detect inappropriate content or misbehaving users, we assume that the system's goal is to detect misbehaving users.

A count-based scheme is probably the most intuitive way to automate an accusation-based social moderation system, as it identifies misbehaving users by considering *the number of accusations*. Under the scheme, a user is deemed to have misbehaved if the number of accusations he/she receives is greater than a certain threshold.

However, the above approach may be made ineffective by collusion attacks. *Collusion* is a secret agreement made between two or more people for fraudulent or deceitful purposes,

978-1-4244-2309-5/09/$25.00 ©2009 IEEE

where the parties accuse one or more innocent users in order to take revenge or to gain unfair advantages.

In this paper, *we propose an automation scheme to deal with collusion attacks in social moderation systems.* Our solution is a community-based scheme that analyzes the relations between the accusing users and accused users. Then, based on the derived information, the scheme infers whether the accusations are fair or malicious; that is, it distinguishes users that genuinely misbehave from victims of collusion attacks. We use simulations to evaluate the performance of our proposed scheme. The results show that, in terms of detecting users who misbehave, the count-based and community-based schemes can achieve accuracy rates higher than 90%. However, our scheme is much more robust to collusion attacks in that at least 90% collusion victims can be saved from collusion attacks. In contrast, more than 50% of collusion victims are incorrectly identified by count-based scheme in most scenarios.

II. Related Work

Collusion detection is one of the most widely studied security problems in trust-based systems, such as reputation systems and recommendation systems, which summarize the ratings of individuals and items respectively. In most scenarios, the ratings given by people are subjective and unverifiable; therefore the question *Are the raters honest?* is an important and challenging problem in this field, as discussed in [3, 7, 10, 11].

In trust-based systems, unfair raters may not cause any harm to the system justice if they act alone; however, they can easily cause harm (i.e., incorrect overall ratings) if they collude with other like-minded raters and compile unfair ratings together.

Cosley et al [3] considered the *shill* problem in e-commerce recommender systems. To resolve the shilling problem, Dellarocas [4] suggested using controlled anonymity and cluster filtering to eliminate the effects of unfair ratings and discrimination. O'Mahony et al [10] conducted empirical studies of the robustness of the KNN user-user algorithm by injecting shilling users into the system, and proved that the algorithm is robust to shilling attacks.

III. Community-based Scheme

In this section, we present our community-based scheme for automating social moderation. The motivation for this scheme is that, for collusion attacks, the number of accusations made against a specific user is no longer a reliable indicator of whether he/she has misbehaved, as some of the accusations may be incorrect intentionally or maliciously. Instead, we must first determine which accusations are *fair* which ones are *made by mistake,* and which ones are *motivated by malicious intent* before we can conclude that the targeted users have definitely misbehaved.

A. Accusing graph

Let A denote the set of identities of accusing users, and let B denote the set of identities of accused users. For every accusing user $a \in A$, and every accused user $b \in B$, exactly one of the following is true: 1) a has accused b; or 2) a has not accused b. We define an accusation relation $\mathcal{R}_\alpha$ as follows. Let $a \in A$ and $b \in B$. We say that $(a, b) \in \mathcal{R}_\alpha$ if a has accused b. By an accusation relation $\mathcal{R}_\alpha$, we mean a subset of the Cartesian product $A \times B$, which is the set of all ordered pairs (a, b), where $a \in A$ and $b \in B$.

In addition, we define an accusing graph to describe the accusation relations between all users in a social moderation system. Assuming a system comprises m users, denoted by $U = \{u_1, u_2, \ldots, u_m\}$, we define that each user has two identities. $A = \{a_1, a_2, \ldots, a_m\}$ is the accusing identity set of users, and $B = \{b_1, b_2, \ldots, b_m\}$ is the accused identity set of users. If a user u_1 accuses a user u_2, then $(a_1, b_2) \in \mathcal{R}_\alpha$; however if u_1 does not accuse u_3, then $(a_1, b_3) \notin \mathcal{R}_\alpha$. Therefore, an accusing graph $G(A + B, \mathcal{R}_\alpha)$ is a complete representation of the user accusation relations in a social moderation system.

Accusing Graph. *An accusing graph $G(A + B, \mathcal{R}_\alpha)$ is a graph in which the nodes can be divided into two disjoint sets A, the identities of accusing users, and B, the identities of accused users, such that every accusing edge connects a node in A to a node in B, and each edge (a, b) belongs to the accusation relation $\mathcal{R}_\alpha$.*

B. Accusing Community

In the social network area, a *community* is a subgraph in which the node-node interconnections are dense, and the edges between the nodes in different communities are much less dense [6]. We call the communities in an accusing graph *accusing communities*. By definition, users tend to accuse other users in the same community more intensively than users in other communities. In other words, users in the same community tend to share similar *accusing tendencies*, *which indicates that the targets of those users overlap significantly*. Thus, the accusing community can be seen as a way of categorizing users into different clusters according to their accusing tendencies.

C. Community Partitioning

To partition an accusing graph into different accusing communities (called *communities* hereafter), we adopt the Girvan-Newman algorithm [9]. The algorithm progressively removes the edges with high in-betweenness values. By doing so, the algorithm gradually divides the graph into several components to reveal the underlying community structure of the network. For example, in the accusing graph $G(X + Y, R_\alpha)$ shown on the left-hand side of Figure 1, $X = \{A, B, C, D, E, F, G, H, I, J, K\}, Y = \{a, b, c, d, e, f, i, j, k\}$, and $R_\alpha = \{(A, b), (D, b), (D, c), (F, c), (G, e), (H, c), (I, b), (I, g), (I, h), (J, g), (J, h), (K, g), (K, h)\}$. Through the Girvan-Newman algorithm, we can obtain three communities, namely G_1, G_2, and G_3, by removing the edge with the highest in-betweenness value, (I, b), as shown on the right-hand side of the figure.

D. Inter-community Edge

We classify users in a social moderation system into two categories: misbehaving users and innocent users. The latter can be further classified into three sub-categories, namely, *victims*, who are accused by purposeful colluders; *unfortunate users*, who are accused due to misjudgment or by accident; and *law-abiding users*, who are not accused by others at all.

Next, we present three properties of the edges between different accusing communities, which we call inter-community edges.

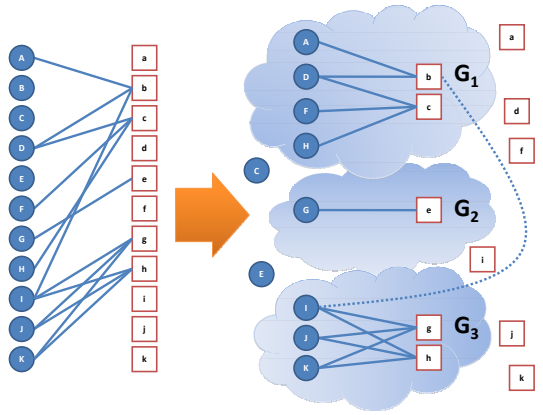

Fig. 1. After apply the Girvan-Newman algorithm, the accusing graph is partitioned into three communities.

Property 1. *It is unlikely that an inter-community edge is an accusing edge between a colluder and a victim.*

Colluders share similar malicious purposes to falsely accuse other users; thus a group of colluders should share similar accusing tendencies. As mentioned in Section III-B, users that share similar accusing tendencies tend to belong to the same community. Furthermore, colluders and their victims tend to be part of the same community because of the accusation relation between them. Therefore, it is unlikely that an inter-community edge is an accusing edge between a colluder and a victim. □

Property 2. *It is unlikely that an inter-community edge is an accusing edge between a careless accuser and an unfortunate user.*

Even though a user is honest, the user may accuse another user because of misjudgment or by accident. Due to the nature of error votes, careless accusers and unfortunate users should be uniformly distributed among all the system's users. As they are unlikely to form large connected components, they tend to form small communities that are isolated from other nodes in the accusing graph. In other words, careless accusers and unfortunates users tend to belong to the same community; therefore, it is unlikely that an inter-community edge is an accusing edge between a careless accuser and an unfortunate user. □

Property 3. *An inter-community edge is most likely an accusing edge between an honest accuser and a misbehaving user.*

An inter-community edge can be one of the following accusing edges: 1) between a colluder and a victim, 2) between a careless accuser and an unfortunate user, and 3) between an honest accuser and a misbehaving user. Based on Property 1 and Property 2, we deduce Property 3; An inter-community edge is most likely an accusing edge between an honest accuser and a misbehaved user. □

E. Identifying Misbehaving Users

According to Property 3, inter-community edges probably represent fair accusations; therefore, we use the edges to identify real misbehaving users. To do so, we define two features to characterize the possibility that a user has misbehaved based on the inter-community edges connected to that user. One is called *incoming accusations*, denoted by $\mathcal{IA}$, and the other is called *outgoing accusations*, denoted by $\mathcal{OA}$.

To explain $\mathcal{IA}$ and $\mathcal{OA}$, we first define *the community indegree* of an accused user, and *the community outdegree* of an accused user. After partitioning an accusing graph into several communities such as $G_1, G_2, \ldots, G_n$, each community always contains a number of accusers and a number of accused users. We let $a_{i,j}$ represent the j-th accuser and $b_{i,p}$ represent the p-th accused user in community G_i, where $i \in \{1, 2, \ldots, n\}$.

Community Indegree. *For an accused user $b_{o,p}$ in community G_o, the community indegree of $b_{o,p}$ is the total number of accusations made against $b_{o,p}$ by any user in community G_i where $i \neq o$. It is denoted as $in(b_{o,p})$.*

Community Outdegree. *For an accusing user $a_{i,j}$ in community G_i, the community outdegree of $a_{i,j}$ is the total number of accusations made against any user in community G_o where $o \neq i$. It is denoted as $out(a_{i,j})$.*

As for $\mathcal{IA}$, we denote the *incoming accusations* about $b_{o,p}$ as $\mathcal{IA}(b_{o,p})$, which means the number of accusations made by accusers in other communities about $b_{o,p}$; therefore, $\mathcal{IA}(b_{o,p})$ corresponds to the number of inter-community edges connected to $b_{o,p}$. We can rewrite $\mathcal{IA}(b_{o,p})$ as

$$\mathcal{IA}(b_{o,p}) = in(b_{o,p}).$$

We use the incoming accusation feature to approximate the number of honest accusations made against a user. Thus, a user u is more likely to have misbehaved if $\mathcal{IA}(u)$ is higher.

As for $\mathcal{OA}$, we use $W_{o,p}$ to represent users that accuse a user $b_{o,p}$ and stay in the community G_o. More precisely, $W_{o,p} = \{a_{o,j} \in G_o | (a_{o,j}, b_{o,p})\}$. Then, the *outgoing accusation* feature of the user $b_{o,p}$ is defined by

$$\mathcal{OA}(b_{o,p}) = \sum_{w \in W_{o,p}} out(w).$$

As inter-community edges probably represent fair accusations made by honest users, $\mathcal{OA}(b_{o,p})$ can be seen an indicator of the degree of honesty of accusers $W_{o,p}$, where a higher $\mathcal{OA}(b_{o,p})$ value indicates that the accusers in $W_{o,p}$ are more honest. $b_{o,p}$ has probably misbehaved if $\mathcal{OA}(b_{o,p})$ is large.

F. Scheme Overview

We summarize our community-based scheme:

1) Apply the Girvan-Newman algorithm to partition the accusing graph into a number of accusing communities.
2) Compute the feature pair $(\mathcal{IA}, \mathcal{OA})$ of each user based on the inter-community edges related to the user.
3) Apply the k-means algorithm to partition all users into two clusters based on their $(\mathcal{IA}, \mathcal{OA})$ pairs, and label users in the cluster with larger $(\mathcal{IA}, \mathcal{OA})$ as misbehaved users.

IV. Performance Evaluation

We use simulations to evaluate the performance of our scheme in detecting misbehaving users in a social moderation system. First, we describe the simulation setup, and then assess the performance of both a counter-based scheme and the proposed community-based scheme in different scenarios. In

TABLE 1
SIMULATION PARAMETERS

Variable	Definition
N	Number of users
R	Number of rounds
V	Number of victims
P	The probability of honest users accusing misbehaving users
P_c	The probability of colluders accusing victims
P_{error}	The probability of users making an accusation by mistake
T	Number of misbehaving users
C	Number of colluders

addition, we then study the effects of the following parameters on the results: the total number of users, the number of misbehaving users, the number of colluders, and different colluder formations.

A. Simulation Setup

In our simulation, we assume there are N users in the system, of which a random set of T users misbehave and another random set C are colluders. Note that $|C|$ is always greater than 3, and the number of victims, V, is always smaller than $|C|$.

In our simulation, the following rules apply to users.

1) An honest user only accuse misbehaving users.
2) A colluder accuses victims.
3) A user has a probability, P_{error}, of making an accusation by mistake. When this occurs, an honest user will accuse an innocent user, or a colluder will accuse a user other than the collusion victims specified.

Our simulation is round-based, and we apply the proposed automation moderation scheme after the simulation runs for R rounds. In each round, each honest user has a probability P of accusing a misbehaving user; similarly, a colluder has a probability P_c of accusing a victim. As colluders have a stronger motivation to participate in the social moderation scheme, we set $P = 10\%$, $P_c = 20\%$, and $P_{error} = 5\%$ as default values. When summarizing the accusation relationships, duplicate accusations made by a user about the same target are removed; hence, there is at most one edge between any two nodes in an accusing graph. The parameters used our simulations are listed in Table 1.

B. Implementation of the Count-based Scheme

We implement the count-based scheme as a baseline method. We also apply the k-means algorithm to partition all users into two clusters based on the number of accusations made against each user, and label users in the cluster with the higher average accusation count as misbehaving users.

C. Performance Metrics

We use two performance metrics to quantify the accuracy and robustness of the social moderation automation schemes, namely, the *correctness of summarization*, and the *collusion resistance*. The correctness of summarization (correctness for short) is defined as

$$correctness = \frac{|\ detected\ misbehaviors\ \cap\ actual\ misbehaviors\ |}{|\ detected\ misbehaviors\ \cup\ actual\ misbehaviors\ |}.$$

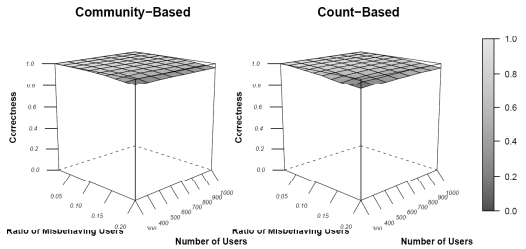

Fig. 2. Correctness with different number of users and ratio of misbehaving users

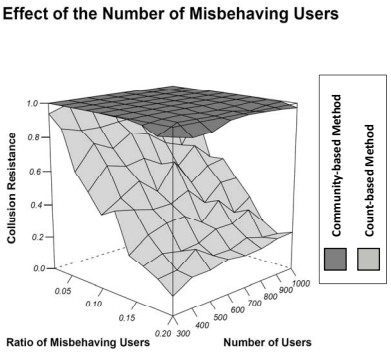

Fig. 3. Collusion resistance with different numbers of users and misbehaving users.

Meanwhile, the collusion resistance index measures whether an automation scheme can prevent collusion attacks. It is defined as follows:

$$collusion\ resistance = 1 - \frac{|\ misidentified\ victims\ |}{|\ all\ victims\ |}.$$

D. Effect of User Population

To assess the effects of the number of users, the number of misbehaving users, and the number of colluders in the compared automation schemes, we set the total user population between 300 and 1000, the ratio of misbehaving users between 2% and 20%, the ratio of colluders between 4% and 24%. We then observed the performance of the count-based and community-based schemes under each scenario.

As shown in Fig. 2, both schemes achieve higher than 90% accuracy regardless of the total number of users and the ratio of misbehaving users. However, the collusion resistance of the count-based scheme decreases dramatically to less than 63% when the ratio of misbehaving users is higher than 10%, as shown in Fig 3. This result implies that the count-based scheme is not robust to collusion attacks, especially when the number of misbehaving users is large. As the number of misbehaving users increases, the number of accusations made about each misbehaving user declines and becomes comparable to that made about collusion victims. Thus, the count-based scheme fails to distinguish between misbehaving users and victims in such scenarios.

Similarly, as the number of colluders increases, the number of accusations made about collusion victims grows and becomes comparable to that of actual misbehaving users. Therefore, it is difficult for the count-based scheme to distinguish between honest accusations and accusations based on

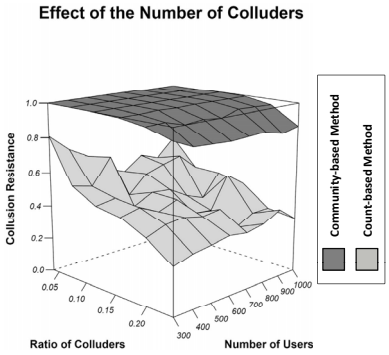

Fig. 4. Collusion resistance with different numbers of users and colluders.

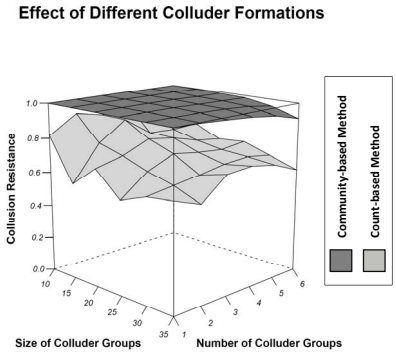

Fig. 5. Collusion resistance with different numbers of colluder groups of different size.

collusion. As shown in Figure 4, our scheme always achieves a collusion resistance rate of 84% or higher regardless of the ratio of misbehaving users and colluders. This supports our assertion that the inter-community edges in the accusing graph are useful for distinguishing honest accusations from accusations based on collusion.

We observe that the correctness of both schemes is always higher than 90% in all the above scenarios, even when the collusion resistance is low. This indicates that correctness is not a useful index for quantifying the robustness of automation schemes under collusion attacks. In the following evaluations, we only consider collusion resistance.

E. Effect of Different Colluder Formations

We also investigate the effect of different colluder formations on the performance of the count-based and community-based automation schemes. Here, we set the total number of users at 800, and the number of misbehaving users at 80. The number of colluder groups is gradually increased from 1 to 6 and the size of each group is increased from 10 to 35, as shown in Fig. 5.

We observe the collusion resistance of the count-based scheme decreases to around 70% when the size of each collusion group is greater than 25. At the same time, the collusion resistance of our community-based method still holds at around 90% in all the scenarios.

V. Conclusion

To resolve the collusion problem, we propose a community-based scheme that can determine whether an accusation is honest or malicious based on the community structure of an accusing graph. Through simulations, we show that our proposed scheme outperforms the naive count-based scheme. The evaluation results show that the collusion resistance of our scheme is around 90% irrespective of the population size, the number of misbehaving users, the number of colluders, and different colluder formations. In contrast, the count-based scheme fails to prevent collusion attacks in the same scenarios. We believe that collusion-resistant schemes like the one proposed in this paper will play an important role in the design of social moderation systems for Web 2.0 services

VI. Acknowledgement

This work was supported in part by Taiwan Information Security Center (TWISC), National Science Council under the grants NSC 97-2219-E-001-001 and NSC 97-2219-E-011-006 and by the iCAST project sponsored by the National Science Council under the grants NSC97-2745-P-001-001. It was also supported in part by Taiwan E-learning and Digital Archives Programs (TELDAP) sponsored by the National Science Council of Taiwan under the grants NSC 96-3113-H-001-010 and NSC 96-3113-H-001-012.

References

[1] Flickr. http://www.flickr.com/.

[2] K.-T. Chen, J.-W. Jiang, P. Huang, H.-H. Chu, C.-L. Lei, and W.-C. Chen. Identifying MMORPG bots: a traffic analysis approach. In *ACE '06: Proceedings of the 2006 ACM SIGCHI international conference on advances in computer entertainment technology*, 2006.

[3] D. Cosley, S. K. Lam, I. Albert, J. A. Konstan, and J. Riedl. Is seeing believing?: how recommender system interfaces affect users' opinions. In *CHI '03: Proceedings of the SIGCHI conference on human factors in computing systems*, pages 585–592, 2003.

[4] C. Dellarocas. Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In *EC '00: Proceedings of the 2nd ACM conference on electronic commerce*, pages 150–157, 2000.

[5] J. J. Garrett. Ajax: A new Approach to Web Applications, Adaptive Path Essay Archive, 2005.

[6] H. Ino, M. Kudo, and A. Nakamura. Partitioning of web graphs by community topology. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 661–669, 2005.

[7] S. K. Lam and J. Riedl. Shilling recommender systems for fun and profit. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 393–402, 2004.

[8] C. Lampe and P. Resnick. Slash(dot) and burn: distributed moderation in a large online conversation space. In *CHI '04: Proceedings of the SIGCHI conference on human factors in computing systems*, pages 543–550, 2004.

[9] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 103:8577, 2006.

[10] M. O'Mahony, N. Hurley, N. Kushmerick, and G. Silvestre. Collaborative recommendation: A robustness analysis. *ACM Transactions on Internet Technology*, 4(4):344–377, 2004.

[11] A. Whitby, A. Josang, and J. Indulska. Filtering out unfair ratings in bayesian reputation systems. In *Proceedings of the 7th international workshop on trust in agent societies*, 2004.