A Co-classification Framework for Detecting Web Spam and Spammers in Social Media Web Sites

Feilong Chen, Pang-Ning Tan, and Anil K. Jain
Department of Computer Science & Engineering
Michigan State University
East Lansing, Michigan 48824
{chenfeil, ptan, jain}@cse.msu.edu

ABSTRACT

Social media are becoming increasingly popular and have attracted considerable attention from spammers. Using a sample of more than ninety thousand known spam Web sites, we found between 7% to 18% of their URLs are posted on two popular social media Web sites, `digg.com` and `delicious.com`. In this paper, we present a co-classification framework to detect Web spam and the spammers who are responsible for posting them on the social media Web sites. The rationale for our approach is that since both detection tasks are related, it would be advantageous to train them simultaneously to make use of the labeled examples in the Web spam and spammer training data. We have evaluated the effectiveness of our algorithm on the `delicious.com` data set. Our experimental results showed that the proposed co-classification algorithm significantly outperforms classifiers that learn each detection task independently.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering*

General Terms

Algorithms, Experimentation, Measurement, Security

Keywords

Social Media, Web Spam, Classification

1. INTRODUCTION

Web spamming refers to any deliberate activity to promote fraudulent Web pages in order to mislead Web users into believing they were viewing legitimate Web pages. Since search engines play a pivotal role in guiding users to find relevant information on the World Wide Web, much of the early spamming activities were directed toward misguiding search engines into ranking spam pages unjustifiably higher. Despite the extensive research in this area, Web spam detection remains a critically important but unsolved problem because spammers may adjust their strategies to adapt to the defense mechanisms employed against them.

Social media are becoming increasingly popular and have attracted considerable attention from spammers. From a list of 94,198 spam Web sites extracted from a benchmark email spam data [9], we found 6,420 ($\approx 7\%$) of them were posted at `digg.com` and 16,537 ($\approx 18\%$) of them were posted at `delicious.com`. While there has been considerable research to detect spam from hyperlinked Web pages to improve search engine performance, spam detection from social media is still in its infancy, with existing work focusing primarily on detecting spam in blogs [3, 7, 6] and online review forums [4]. Unlike spam detection from hyperlinked Web pages, there are richer amount of data available in social media that can be utilized for Web spam detection, such as the links between users, hyperlinks between the Web content, and links between users and the Web content. Users may also assign a set of keyword tags to annotate the Web content they have posted. Therefore, a key challenge is to systematically incorporate all the heterogeneous data in a unified framework to improve Web spam detection.

In addition to Web spam detection, it is useful to identify the spammers who are responsible for posting such links to prevent them from future spamming activities. This is a challenging task because some legitimate users may unknowingly post links to spam Web sites, while some spammers may deliberately add links to non-spam Web sites to avoid detection. This paper assumes that spammers tend to post considerably higher amount of Web spam compared to non-spammers. Since both Web spam and spammer detection tasks are related, it would be advantageous to train their classifiers simultaneously to make use of their labeled examples. To the best of our knowledge, there has not been any prior work on the joint detection of Web spam and spammers, an approach which we termed as *co-classification*.

This paper presents a co-classification framework for Web spam and spammer detection in social media based on the maximum margin principle. Specifically, we formalize the joint detection tasks as a constraint optimization problem, in which the relationships between users and their submitted Web content are represented as constraints in the form graph regularization. To ground the discussion of our framework, we use as an example the social bookmarking Web site `delicious.com`. While the concepts in this paper are presented for the social bookmarking domain [2], our proposed framework is applicable to other social media Web sites

CIKM'09, November 2–6, 2009, Hong Kong, China.

where the following data are available: (1) links between users, (2) links between users and their submitted Web content (social news, blogs, opinions, etc), and (3) tags or other content-based features derived from the Web content. We also show that our framework is applicable to both supervised and semi-supervised learning settings. Experimental results using the `delicious.com` data set showed that our co-classification framework significantly outperforms classifiers that learn each detection task independently.

2. PRELIMINARIES

We begin with a brief discussion of the terminology and notations used in this paper. While the terminology introduced here are based on the features available at `delicious.com`, they are also applicable to other social media Web sites.

- **User:** A registered visitor of the social bookmarking Web site. Let $\mathcal{U}$ be the set of all users.
- **Bookmark:** A shortcut to the URL of a Web page. Let $\mathcal{B}$ be the set of all bookmarks.
- **Tag:** A keyword or text description assigned by a user to a bookmark. Let $\mathcal{T}$ be the set of all tags.
- **Post:** A 4-tuple $(b,\ u,\ t,\ \tau)$, where $b \in \mathcal{B}$, $u \in \mathcal{U}$, $t \subseteq \mathcal{T}$, and τ is the timestamp at which the user posted the bookmark on the Web site. We denote the set of all posts, also known as the posting history, as Π. Furthermore, let $E_b = \{(u,b)|u \in \mathcal{U}, b \in \mathcal{B}, \exists t, \tau : (b,u,t,\tau) \in \Pi\}$.
- **Fan:** A directional link from one user to another. If a user u adds another user v to his/her network, then u becomes a fan of v. Let $E_u = \{(u,v)|u,v \in \mathcal{U}\}$ be the set of all pairs of users in which u is a fan of v.

The overall data can be represented as a graph $\mathcal{G} = (V, E)$, where $V = \mathcal{B} \cup \mathcal{U}$ is the set of nodes (bookmarks and users) and $E = E_b \cup E_u$ is the set of links. Our work is based on the following two assumptions: (1) If $(u,b) \in E_b$ and b is a spam bookmark, then u is more likely to be a spammer. (2) If $(u,v) \in E_u$ and v is a spammer, then u is also likely to be a spammer. As will be shown later in Section 3, these assumptions are enforced as graph regularization constraints in our proposed co-classification framework.

Throughout this paper, matrices are denoted by boldface capital letters like $\mathbf{X}$ and vectors are denoted by boldface small letters like $\mathbf{y}$ (for column vector) or $\mathbf{y}^T$ (for row vector), where T is the transpose operator. Elements of matrices and vectors are of the form X_{ij} and y_j, respectively.

3. METHODOLOGY

In this section, we first formalize the Web spam and spammer detection problems. We then present the derivation of our proposed co-classification framework.

3.1 Problem Formulation

Suppose we are given:

- a set of l_b labeled bookmarks, $\mathcal{V}_b = \{(\mathbf{x}_1^{(b)}, y_1^{(b)}), (\mathbf{x}_2^{(b)}, y_2^{(b)}), \cdots (\mathbf{x}_{l_b}^{(b)}, y_{l_b}^{(b)})\}$, where $\mathbf{x}_i^{(b)}$ is a d_b-dimensional feature vector for the i-th post and $y_i^{(b)}$ is its class label ($+1$ for a spam bookmark and -1 for non-spam).
- a set of $n_b - l_b$ unlabeled bookmarks, $\mathcal{U}_b = \{\mathbf{x}_{l_b+1}^{(b)}, \mathbf{x}_{l_b+2}^{(b)}, \cdots, \mathbf{x}_{n_b}^{(b)}\}$.
- a set of l_u labeled users, $\mathcal{V}_u = \{(\mathbf{x}_1^{(u)}, y_1^{(u)}), (\mathbf{x}_2^{(u)}, y_2^{(u)}), \cdots (\mathbf{x}_{l_u}^{(u)}, y_{l_u}^{(u)})\}$, where $\mathbf{x}_i^{(u)}$ is a d_u-dimensional feature vector for the i-th user and $y_i^{(u)}$ is its class label ($+1$ for a spammer and -1 for non-spammer).
- a set of $n_u - l_u$ unlabeled users, $\mathcal{U}_u = \{\mathbf{x}_{l_u+1}^{(u)}, \mathbf{x}_{l_u+2}^{(u)}, \cdots, \mathbf{x}_{n_u}^{(u)}\}$.
- a set of user-bookmark pairs, E_b, where $(u,b) \in E_b$ if user u posts a bookmark b.
- a set of user-user pairs, E_u, where $(u,v) \in E_u$ if user u is a fan of user v.

Furthermore, let $\mathbf{X}_b$ (or $\mathbf{X}_u$) be a matrix whose i^{th} row corresponds to the feature vector of bookmark (or user) i, i.e., $\mathbf{x}_i^{(b)T}$ (or $\mathbf{x}_i^{(u)T}$).

Given $\mathcal{V}_b$, $\mathcal{V}_u$, E_b, and E_u, the goal of Web spam and spammer detection tasks is to learn a pair of classifiers: (1) $f_b(\mathbf{x}^{(b)})$ that accurately maps a bookmark $\mathbf{x}^{(b)}$ to its class label $y^{(b)} \in \{-1, +1\}$ and (2) $f_u(\mathbf{x}^{(u)})$ that accurately maps a user $\mathbf{x}^{(u)}$ to its class label $y^{(u)} \in \{-1, +1\}$.

3.2 Maximum Margin Classifier

The classifiers used in this paper are extensions of the least-square support vector machine (LS-SVM), a variant of maximum margin classifier proposed by Suykens et al. [8]. Specifically, we may construct a LS-SVM classifier for bookmarks by minimizing the following objective function:

$$\begin{aligned}\mathcal{L}_b(\mathbf{w}_b, b_b, \mathbf{e}, \alpha) &= \frac{1}{2}\mathbf{w}_b^T\mathbf{w}_b + \gamma_1\frac{1}{2}\sum_{i=1}^{l_b} e_i^2 \\ &- \sum_{i=1}^{l_b}\alpha_i\left\{y_i^{(b)}\left[\mathbf{w}_b^T\mathbf{x}_i^{(b)} + b_b\right] - 1 + e_i\right\},\end{aligned}$$

Similarly, the corresponding objective function for classifying users can be expressed as follows:

$$\begin{aligned}\mathcal{L}_u(\mathbf{w}_u, b_u, \xi, \beta) &= \frac{1}{2}\mathbf{w}_u^T\mathbf{w}_u + \gamma_2\frac{1}{2}\sum_{i=1}^{l_u} \xi_i^2 \\ &- \sum_{i=1}^{l_u}\beta_i\left\{y_i^{(u)}\left[\mathbf{w}_u^T\mathbf{x}_i^{(u)} + b_u\right] - 1 + \xi_i\right\},\end{aligned}$$

However, by training them independently, the classifiers do not make use of the labeled examples in the Web spam and spammer training data as well as the link information between the users and their corresponding bookmarks.

3.3 Co-Classification of Bookmarks and Users

Instead of solving the optimization problems for classifying bookmarks and users independently, our co-classification framework utilizes additional information from the link structure between users and bookmarks in E_b to ensure that their solutions are consistent with each other.

The objective function for our co-classification framework of bookmarks and users can be written as follows:

$$
\begin{aligned}
\mathcal{L} &= \frac{1}{2}\mathbf{w}_b^T\mathbf{w}_b + \gamma_1\frac{1}{2}\sum_{i=1}^{l_b} e_i^2 + \frac{1}{2}\mathbf{w}_u^T\mathbf{w}_u + \gamma_2\frac{1}{2}\sum_{i=1}^{l_u}\xi_i^2 \\
&- \sum_{i=1}^{l_b}\alpha_i\left\{y_i^{(b)}\left[\mathbf{w}_b^T\mathbf{x}_i^{(b)} + b_b\right] - 1 + e_i\right\} \\
&- \sum_{i=1}^{l_u}\beta_i\left\{y_i^{(u)}\left[\mathbf{w}_u^T\mathbf{x}_i^{(u)} + b_u\right] - 1 + \xi_i\right\} \\
&- \gamma_3\sum_{(i,j)\in E_u}\delta_{ij}[\mathbf{w}_u^T\mathbf{x}_i^{(u)} - \mathbf{w}_u^T\mathbf{x}_j^{(u)}] \\
&- \gamma_4\sum_{(i,j)\in E_b}\eta_{ij}[\mathbf{w}_u^T\mathbf{x}_i^{(u)} + b_u - \mathbf{w}_b^T\mathbf{x}_j^{(b)} - b_b], \qquad (1)
\end{aligned}
$$

where ($\mathbf{w}_b$, $\mathbf{w}_u$, b_b, b_u, $\mathbf{e}$, ξ, α, β) are the model parameters to be estimated from training data and (γ_1, γ_2, γ_3, γ_4) are the user-specified parameters. For our experiments, we set $\gamma_1 = \gamma_2 = 1$ whereas γ_3 and γ_4 are estimated from the data via ten-fold cross validation. The term associated with γ_3 is used to enforce the constraint due to links between users. To understand the rationale for this term, note that the value of the objective function increases whenever $\mathbf{w}_u^T\mathbf{x}_i^{(u)} + b_u < \mathbf{w}_u^T\mathbf{x}_j^{(u)} + b_u$ (i.e., if a non-spammer is a fan of a spammer). Thus, the graph regularization term can be viewed as penalizing models that assign non-spammers as fans of spammers. This idea was inspired by prior works on using graph regularization methods to combine link structure and content information in Web pages [10, 1]. The parameter δ_{ij} is the weight of the directed edge $E_u(i,j)$. Instead of assigning a binary 0/1 weight to every pair of nodes (users), we normalize the weight based on the out-degree of the source node, i.e., $\delta_{ij} = 1/\sum_j E_u(i,j)$.

Similarly, the last term in the objective function is used to penalize models in which non-spammers are allowed to bookmark many spam pages:

$$\eta_{ij}[\mathbf{w}_u^T\mathbf{x}_i^{(u)} + b_u - \mathbf{w}_b^T\mathbf{x}_j^{(b)} - b_b], \quad \forall (i,j) \in E_b,$$

where $\eta_{i,j} = 1/\sum_j E_b(i,j)$ is the weight of $E_b(i,j)$.

The objective function given in (1) can be solved in a supervised or semi-supervised learning setting, depending on the nodes used to express the graph regularization constraints in E_b and E_u. In a supervised learning setting, E_b and E_u involve only bookmarks and users that are part of the labeled training data. For semi-supervised learning, the constraints due to E_b and E_u in (1) include unlabeled bookmarks and users as well. The solution of the objective function is obtained by taking the derivative of $\mathcal{L}$ with respect to each of the model parameters and setting them to zero. This reduces to a system of linear equations that can be expressed in matrix notation as follows:

$$
\begin{bmatrix}
\mathbf{I}_{d_u} & 0 & 0 & -\mathbf{Z}_u^T & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \mathbf{y}^{(u)T} & 0 & 0 & 0 & 0 \\
0 & 0 & \gamma_2\mathbf{I}_{l_u} & -\mathbf{I}_{l_u} & 0 & 0 & 0 & 0 \\
\mathbf{Z}_u & \mathbf{y}^{(u)} & \mathbf{I}_{l_u} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \mathbf{I}_{d_b} & 0 & 0 & -\mathbf{Z}_b^T \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{y}^{(b)T} \\
0 & 0 & 0 & 0 & 0 & 0 & \gamma_1\mathbf{I}_{l_b} & -\mathbf{I}_{l_b} \\
0 & 0 & 0 & 0 & \mathbf{Z}_b & \mathbf{y}^{(b)} & \mathbf{I}_{l_b} & 0
\end{bmatrix}
\begin{bmatrix}
\mathbf{w}_u \\ b_u \\ \xi \\ \beta \\ \mathbf{w}_b \\ b_b \\ \mathbf{e} \\ \alpha
\end{bmatrix}
=
\begin{bmatrix}
\mathbf{p}_1 \\ -q \\ \mathbf{0}_{l_u} \\ \mathbf{1}_{l_u} \\ \mathbf{p}_2 \\ q \\ \mathbf{0}_{l_b} \\ \mathbf{1}_{l_b}
\end{bmatrix}
$$

where $\mathbf{I}_d$ is a $d \times d$ identity matrix, $\mathbf{0}_d$ is a d-dimensional vector of 0s, $\mathbf{1}_d$ is a d-dimensional vector of 1s, and

$$
\begin{aligned}
\mathbf{Z}_u &= [\mathbf{x}_1^{(u)T}y_1^{(u)}; \ldots; \mathbf{x}_{l_u}^{(u)T}y_{l_u}^{(u)}] \\
\mathbf{Z}_b &= [\mathbf{x}_1^{(b)T}y_1^{(b)}; \ldots; \mathbf{x}_{l_b}^{(b)T}y_{l_b}^{(b)}] \\
\mathbf{p}_1 &= \gamma_3\sum_{(i,j)\in E_u}\delta_{ij}[\mathbf{x}_i^{(u)} - \mathbf{x}_j^{(u)}] \\
&\quad + \gamma_4\sum_{(i,j)\in E_b}\eta_{ij}\mathbf{x}_i^{(u)} \\
\mathbf{p}_2 &= -\gamma_4\sum_{(i,j)\in E_b}\eta_{ij}\mathbf{x}_j^{(b)} \\
q &= \gamma_4\sum_{(i,j)\in E_b}\eta_{ij}
\end{aligned}
$$

The block structure of the matrix equation suggests that the system of linear equations can be further decoupled into two subproblems, one for learning the parameters of the bookmark classifier and the other for user classifier.

$$
\begin{bmatrix}
\mathbf{Z}_u\mathbf{Z}_u^T + \gamma_2\mathbf{I}_{l_u} & \mathbf{y}^{(u)} \\
\mathbf{y}^{(u)T} & 0
\end{bmatrix}
\begin{bmatrix}
\beta \\ b_u
\end{bmatrix}
=
\begin{bmatrix}
1 - \mathbf{Z}_u\mathbf{p}_1 \\ -q
\end{bmatrix} \qquad (2)
$$

$$
\begin{bmatrix}
\mathbf{Z}_b\mathbf{Z}_b^T + \gamma_1\mathbf{I}_{l_b} & \mathbf{y}^{(b)} \\
\mathbf{y}^{(b)T} & 0
\end{bmatrix}
\begin{bmatrix}
\alpha \\ b_b
\end{bmatrix}
=
\begin{bmatrix}
1 - \mathbf{Z}_b\mathbf{p}_2 \\ q
\end{bmatrix} \qquad (3)
$$

Equations (2) and (3) can be solved for the model parameters $(\alpha, \beta, b_u, b_b)$. Furthermore, $\mathbf{w}_u$ and $\mathbf{w}_b$ can be expressed in terms of β, α, $\mathbf{Z}_u$ and $\mathbf{Z}_b$. As a result, a user $\mathbf{x}_{\text{test}}^{(u)}$ and a bookmark $\mathbf{x}_{\text{test}}^{(b)}$ can be classified as follows: $f_u(\mathbf{x}_{\text{test}}^{(u)}) = \mathbf{w}_u^T\mathbf{x}_{\text{test}}^{(u)} + b_u$ and $f_b(\mathbf{x}_{\text{test}}^{(b)}) = \mathbf{w}_b^T\mathbf{x}_{\text{test}}^{(b)} + b_b$. Finally, although the proposed co-classification framework assumes a linear classifier, it can be extended to non-linear models using the well-known *kernel trick*. However, we have considered only linear kernels in this study.

Algorithm 1 summarizes the high-level overview of our `Co-Class` algorithm. The `LinearSolver` function solves the system of linear equations given in (2) and (3). The `Classify` function takes as input the model parameters and unlabeled examples to generate their predicted class labels.

Algorithm 1 Co-Class Algorithm

Input: $\mathcal{V}_b, \mathcal{V}_u, \mathcal{U}_b, \mathcal{U}_u, E_b, E_u, \gamma$
Output: $\{y_i^{(u)} | i = l_u + 1 \cdots n_u\}, \{y_j^{(b)} | b = l_b + 1 \cdots n_b\}$.
Method:
1. $(\alpha, \beta, b_u, b_b) \leftarrow$ `LinearSolver`$(\mathcal{V}_b, \mathcal{V}_u, E_b, E_u, \gamma)$
2. $\mathbf{y}^{(u)} \leftarrow$ `Classify`$(\mathcal{U}_u, \mathcal{V}_u, \beta, b_u)$
3. $\mathbf{y}^{(b)} \leftarrow$ `Classify`$(\mathcal{U}_b, \mathcal{V}_b, \alpha, b_b)$

4. EXPERIMENTAL EVALUATION

This section presents our experiment results to demonstrate the effectiveness of our `Co-Class` algorithms. We use a real-world data set acquired from `delicious.com` to evaluate our algorithms. The dataset consists of posting history for nearly 3 million users, whose feature vectors are generated from about 2.5 million tags. In addition, the data also contains about 110,000 bookmarks, whose feature vectors include about 300,000 tags. The user-bookmark links, E_b and user-user links E_u are also obtained by preprocessing the user profile pages. URLs harnessed from the email

Table 1: Experimental results on the delicious.com data set.

Training data = 10%	Bookmark			User		
	Recall	Precision	F_1	Recall	Precision	F_1
linear SVM	0.4266 ± 0.0335	0.7587 ± 0.0150	0.5451 ± 0.0255	0.2537 ± 0.0397	0.6873 ± 0.0220	0.3686 ± 0.0408
SVM-rbf	0.4320 ± 0.0415	0.7591 ± 0.0136	0.5517 ± 0.0310	0.2541 ± 0.0276	0.6901 ± 0.0408	0.3720 ± 0.0541
Co-class	0.5101 ± 0.0251	0.7702 ± 0.0326	0.6136 ± 0.0430	0.3802 ± 0.0390	0.7476 ± 0.0275	0.5131 ± 0.0316

Training data = 30%	Bookmark			User		
	Recall	Precision	F_1	Recall	Precision	F_1
linear SVM	0.5783 ± 0.0177	0.8110 ± 0.0163	0.6749 ± 0.0103	0.3521 ± 0.0162	0.7097 ± 0.0206	0.4703 ± 0.0147
Co-class	0.6300 ± 0.0186	0.8203 ± 0.0200	0.7103 ± 0.0317	0.4502 ± 0.0319	0.7330 ± 0.0211	0.5647 ± 0.0280

spam benchmark data in [9] are used to label the spam bookmarks. A user is labeled as a spammer if he/she posted to at least one of those spam bookmarks. To test the performance of our algorithms, we randomly selected a sample of $20,000$ bookmarks and $20,000$ users. 20% of the data set were identified to be spam (or spammers) and the remaining 80% were non-spam (or non-spammers).

For comparison purposes, we use SVM-light [5] to build a pair of support vector machine classifiers that learn the classification of users and bookmarks independently using their respective training data. We reported the precision, recall, and F_1-measures after performing repeated 10-fold cross validation on the data. To make the problem more challenging, for each fold, we use 10% of the data for training while the remaining 90% are used as test data. We repeated the experiment 20 times by sampling another $20,000$ bookmarks and users from the original data. Table 1 shows the relative performance of linear and nonlinear SVM against the semi-supervised `Co-Class` algorithm. Clearly, the F_1-measure for `Co-Class` is significantly higher than that for SVM, both in terms of classifying bookmarks and classifying users. The F_1-measure for bookmark classification is higher than that for user classification but the margin of improvement using `Co-Class` is larger in terms of classifying users than classifying bookmarks. We suspect this is due to the additional information in the link structure of E_u, which helps to boost the performance of our co-class algorithm when classifying users. The results also showed that most of the improvement in `Co-Class` is due to its higher recall value. Since the data set is moderately skewed, SVM tends to classify many of the examples into the larger class, which explains its lower recall value. In contrast, graph regularization using the link structure in E_b and E_u allows our `Co-Class` algorithm to identify more spam and spammers from the data set without losing its precision. We also increased the percentage of training data during 10-fold cross validation from 10% to 30%. The results in Table 1 showed our algorithm significantly outperformed linear SVM, though the margin of improvement is less than that obtained using 10% training data.

5. CONCLUSION

This paper focuses on the problem of Web spam and spammer detection in social media Web sites. Unlike search engine spam, Web spam from social bookmarking and social news aggregator Web sites are potentially damaging because it may direct users to malicious Web sites that compromise browser security. To overcome this problem, we presented a co-classification framework that simultaneously trains classifiers for detecting Web spam and spammers. We demonstrated that such a strategy is more effective than learning each task independently. For future work, we plan to extend the methodology to incorporate data from multiple social media Web sites. For example, an interesting research direction is to investigate the co-classification of bookmarks and users from both `delicious.com` and `digg.com`. Furthermore, it would be useful to investigate methods for reducing the number of user parameters in our learning framework.

6. ACKNOWLEDGMENTS

This research was partially supported by ONR Grant Number N00014-09-1-0663 and the Army Research Office.

7. REFERENCES

[1] J. Abernethy, O. Chapelle, and C. Castillo. Web spam identification through content and hyperlinks. In *Proc. of the SIGIR Workshop on Adversarial Information Retrieval on the Web (AIRWEB'08)*, Beijing, China, April 2008.

[2] F. Chen, J. Scripps, and P. Tan. Link mining for a social bookmarking web site. In *Proc. of IEEE/WIC/ACM Int'l Conf. on Web Intelligence*, 2008.

[3] K. Ishida. Extracting spam blogs with co-citation clusters. In *Proc. of the 17th Int'l Conf. on World Wide Web*, pages 1043–1044, New York, NY, 2008.

[4] N. Jindal and B. Liu. Opinion spam and analysis. In *Proc. of Int'l Conf. on Web Search and Web Data Mining (WSDM 08)*, 2008.

[5] T. Joachims. http://svmlight.joachims.org/.

[6] G. Koutrika, F. A. Effendi, Z. Gyongyi, P. Heymann, and H. Garcia-Molina. Combating spam in tagging systems: An evaluation. 2(4):1–34, 2008.

[7] Y. Lin, H. Sundaram, Y. Chi, J. Tatemura, and B. L. Tseng. Detecting splogs via temporal dynamics using self-similarity analysis. *ACM Trans. Web*, 2(1):1–35, Feb. 2008.

[8] J. Suykens, T. Gestel, J. Brabanter, B. Moor, and J. Vandewalle. *Least Squares Support Vector Machines*. World Scientific Pub, Singapore, 2002.

[9] S. Webb, J. Caverlee, , and C. Pu. Introducing the Webb spam corpus: Using email spam to identify web spam automatically. In *Proc. of CEAS '06*, 2006.

[10] T. Zhang, A. Popescul, and B. Dom. Linear prediction models with graph regularization for web-page categorization. In *Proc. of ACM SIGKDD Int'l Conf on Data Mining*, pages 821–826, Philadelphia, PA, 2006.