IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 12, 2017, accepted December 13, 2017, date of publication December 18, 2017,
date of current version February 14, 2018.

*Digital Object Identifier 10.1109/ACCESS.2017.2784370*

# Detecting Spammer Groups From Product Reviews: A Partially Supervised Learning Model

**LU ZHANG, ZHIANG WU, (Member, IEEE), AND JIE CAO**
Jiangsu Provincial Key Laboratory of E-Business, Nanjing University of Finance and Economics, Nanjing 210023, China

Corresponding author: Jie Cao (jie.cao@njue.edu.cn)

This work was supported in part by the National Key Technologies Research and Development Program of China under Grant 2017YFD0401002, in part by the National Natural Science Foundation of China under Grant 91646204 and Grant 71571093, in part by the National Center for International Joint Research on E-Business Information Processing under Grant 2013B01035, in part by the Industry Projects in Jiangsu S&T Pillar Program under Grant BE2014141, and in part by the Surface Projects of Natural Science Research in Jiangsu Provincial Colleges and Universities under Grant 15KJB520012 and Grant 15KJB520011.

**ABSTRACT** Nowadays, online product reviews play a crucial role in the purchase decision of consumers. A high proportion of positive reviews will bring substantial sales growth, while negative reviews will cause sales loss. Driven by the immense financial profits, many spammers try to promote their products or demote their competitors' products by posting fake and biased online reviews. By registering a number of accounts or releasing tasks in crowdsourcing platforms, many individual spammers could be organized as spammer groups to manipulate the product reviews together and can be more damaging. Existing works on spammer group detection extract spammer group candidates from review data and identify the real spammer groups using unsupervised spamicity ranking methods. Actually, according to the previous research, labeling a small number of spammer groups is easier than one assumes, however, few methods try to make good use of these important labeled data. In this paper, we propose a partially supervised learning model (PSGD) to detect spammer groups. By labeling some spammer groups as positive instances, PSGD applies positive unlabeled learning (PU-Learning) to study a classifier as spammer group detector from positive instances (labeled spammer groups) and unlabeled instances (unlabeled groups). Specifically, we extract reliable negative set in terms of the positive instances and the distinctive features. By combining the positive instances, extracted negative instances and unlabeled instances, we convert the PU-Learning problem into the well-known semi-supervised learning problem, and then use a Naive Bayesian model and an EM algorithm to train a classifier for spammer group detection. Experiments on real-life Amazon.cn data set show that the proposed PSGD is effective and outperforms the state-of-the-art spammer group detection methods.

**INDEX TERMS** Spammer group detection, partially supervised learning, positive unlabeled learning, reliable negative set extraction, Naive Bayesian model, EM algorithm.

## I. INTRODUCTION

In e-commerce platforms, online product reviews become more and more important as the purchase decisions of the customers are strongly influenced by these reviews [1]. Due to the financial incentives, many imposters try to game the systems and consumers by posting biased ratings and reviews to promote their products or demote their competitors' products [2]. These imposters, also called *Review Spammers* or *Opinion Spammers*, become more and more damager as they could be organized by crowdsourcing tasks. As there are many accounts, the organized spammers, called *Spammer Group*, could take total control of the sentiment on their target products with little abnormal behavior.

Although many efforts have been done for review spam and individual spammer detection [3]–[11], limited attention has been received at the spammer group detection [12]–[14]. Generally, as there are usually no labeled instances (groups), most existing work find spammer group candidates first, and then use unsupervised ranking methods to identify real spammer groups from these candidates. However, according to the research in [12], we could easily label some groups manually to obtain some labeled instances (i.e., labeled spammer groups or non-spam groups). It is obvious that combining these labeled instances and other unlabeled groups will significantly improve the accuracy of spammer group detection.

Simultaneously utilizing labeled and unlabeled data is a typical problem of partially supervised learning. Strictly speaking, there are two types of partially supervised learning according to the constitution of labeled data [15]. One type is that the labeled data contains the instances of all classes (e.g., containing both spammer and non-spammer groups in this paper), which is commonly known as semi-supervised learning. The second type is that the labeled data only contains positive instances (e.g., spammer groups) and we need to learn from the positive and unlabeled instances. In this paper, we call the second type of partially supervised learning as Positive Unlabeled Learning (PU-Learning for short, where P and U stand for positive and unlabeled instances, respectively). Since labeling spammer groups is much easier than labeling non-spammer groups [12], we adopt PU-Learning as the main technique to detect spammer groups without labeling any non-spammer groups.

In this paper, we propose a Partially Supervised learning based Spammer Group Detection (PSGD) model. Like most existing spammer group detection methods, we use frequent item mining (FIM) to extract spammer group candidates, and then apply PU-Learning to detect real spammer groups from these candidates. Specifically, we manually label some spammer groups from the found group candidates as positive instances. Then, supervised by these positive instances, we design an algorithm to automatically extract reliable negative set (*RN*) which consists of only non-spammer groups. Combine the positive and negative instances, the PU-Learning problem could be converted into the well-known semi-supervised learning problem, then the Naive Bayesian model and Expectation Maximization (EM) algorithm are used to train a classifier to detect spammer groups. Our previous work [16] has proposed a semi-supervised learning method to detect spammer groups using Naive Bayesian model and EM algorithm, which requires both labeled spammer groups and labeled non-spammer groups. Differ from the previous work, we focus on the detection with only labeled spammer groups in this paper. Our main contributions are summarized as follows.

1) We propose PSGD, a partially supervised learning model to detect review spammer groups. Specifically, we only label some spammer groups as positive instances and learn a classifier from the positive and unlabeled instances. To the best of our knowledge, this is the first time PU-Learning is applied to spammer group detection.
2) We design a reliable negative set (*RN*) extraction algorithm which defines a feature strength function to measure the discriminative power of group features, and then iteratively removes instances containing high discriminative features from the unlabeled instances set to obtain *RN*. By combining the positive instances and the extracted negative instances, the PU-Learning problem can be converted into the well-known semi-supervised learning problem, thus many mature methods such as Naive Bayesian model and EM algorithm can be applied to construct the classifier.
3) We conduct extensive experiments on a real-life dataset collected from Amazon.cn. We propose two new group features and verify their effect for improving the performance of detection. Given the overall performance of PSGD, we also analyze the impact of the weighting factor of unlabeled data and evaluate the effectiveness of our proposed *RN* extraction algorithm. The experimental results demonstrate that PSGD can effectively detect spammer groups and outperforms the state-of-the-art spammer group detection methods.

The remainder of this paper is organized as follows. Section II discusses the related work about review spam, spammer and spammer group detection. Section III presents the overview of our PSGD model. In Section IV, we give the details of PSGD, including reliable negative set extraction and semi-supervised learning. Experimental results are reported in Section V. We conclude the paper and present the future work in Section VI.

## II. RELATED WORK

Ever since Jindal and Liu proposed the problem of review spam detection [3], a variety of methods and techniques have been proposed in this area, which can be summarized for detection of three targets [17]: review spam, spammer and spammer group. Among these, review spam detection and spammer detection have received dominant research attention. As the spammers tend to copy the texts of existing reviews for the target products, early methods find duplicate or near duplicate reviews to detect the review spam, where the similarity of reviews are mainly calculated by n-gram based review content comparison [3] or probabilistic language model [18]. In many works, the review spam detection is deemed as a binary classification or ranking problem. Many content features and metadata of reviews, such as parts-of-speech (POS), term frequency and n-gram features, are used for classification or ranking [19]–[23]. Spammer detection [5], [6], [24]–[27] has the similar principle with review spam detection, however, the features for classification or ranking are mainly constructed from user behavior such as review/rating posting time, rating deviation, burst review ratio, reviewer burstiness and ratio of verified purchase (only in Amazon). According to the above-mentioned content and behavior features, most works build supervised learning classification models [3], [4] or HITS-like unsupervised ranking algorithms [7], [28] to distinguish review spam or spammers from normal ones. Besides, with only a small portion of labeled reviews or reviewers, researchers applied semi-supervised learning on review spam or spammer detection, and some works have proposed to learn classifiers from positive and unlabeled data [29]–[32]. These research achievements have proved that the methods involving both labeled and unlabeled data outperform traditional methods using only supervised or unsupervised learning.

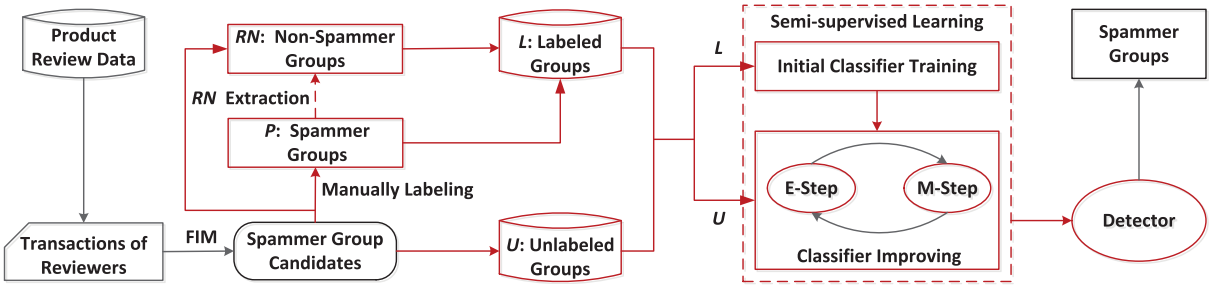


**FIGURE 1.** Overview of the PSGD model.

Comparing with the review spam or spammer detection, limited efforts have been done for spammer group detection. Existing work usually use frequent itemset mining (FIM) to discover group candidates first, and then identify the candidates as spammer or non-spammer groups using unsupervised ranking methods [12]–[14]. Since labeling spammer groups is much easier than labeling individual spammers [12], learning a classifier from these positive instances and other unlabeled instances is a straightforward way to improve the accuracy of spammer group detection. In this paper, we propose a partially supervised learning model to detect spammer groups, in which PU-Learning is applied to extract reliable negative set and train a semi-supervised learning classifier as detector based on Naive Bayesian model and EM algorithm.

## III. PSGD: MODEL OVERVIEW

A spammer group consists of a set of reviewers who co-reviews a set of common products. Thus, the data mining technique *Frequent Itemset Mining* (FIM) could be utilized to extract the groups [12], [13]. However, since many users may be coincidentally grouped because of the similar interest, the groups extracted by FIM are only the spammer group candidates and need to be further checked to identify the real spammer groups. Therefore, the detection of spammer groups usually contains two phases: (i) Discover spammer group candidates, (ii) Identify the real spammer groups from the candidates. Our proposed PSGD model is also along this line.

Fig. 1 illustrates the diagram of the PSGD model. In the context of spammer group detection, the reviewers are seen as the items and the reviewers who have co-reviewed a particular product are regarded as a transaction. By mining frequent itemsets, we find groups of reviewers who have co-reviewed multiple products together as spammer group candidates. Among the extracted candidates, some spammer groups are manually labeled to construct positive instances set, denoted as $P$. Then, by automatically extracting some groups whose features are significantly different with instances in $P$, the reliable negative set (denoted as $RN$) consisting of only non-spammer groups will be constructed. Combine $P$ and $RN$, we will obtain a labeled data set (denoted as $L$) containing both positive and negative instances, and the remainder spammer group candidates with unknown classes will construct an unlabeled data set (denoted as $U$). Based on $L$ and $U$, a semi-supervised learning classifier is trained to identify real spammer groups, which initials a Naive Bayes classifier on $L$ and improves it on $U$ using an Expectation Maximization (EM) algorithm. The following section will provide the technical details within the PSGD model.

## IV. PSGD: TECHNICAL DETAILS

In this section, we provide the technical details of PSGD model, including reliable negative set extraction and semi-supervised learning.

### A. RELIABLE NEGATIVE SET EXTRACTION

The PU-Learning first singles out a set of instances from $U$ that are significantly different with instances in $P$ to construct a reliable negative set ($RN$) [33]. Here, the "reliable" means we do not emphasize to extract ample negative instances, however, we need to make sure that the extracted instances are indeed non-spammer groups, i.e., belong to the opposite class of $P$. In other words, given the features of the instances, the discriminative power of these features between instances in $P$ and $RN$ must be maximized. Thus we propose the following objective function

$$O_1 : \max_{f \in F} D_f(P \cup RN) \tag{1}$$

where $F$ is the set of features belonging to the instances in $P$ and $RN$, and $D_f$ is the feature strength function that measures the discriminative power of features. Obviously, maximizing Eq. (1) is a combinatorial optimization which is a NP-hard problem. As an alternative, we present a greedy $RN$ extraction heuristic that defines a proper $D_f$ function to sort all features and maximizes $D_f(P \cup RN)$ for every feature.

Empirically, the high discrimination of a feature $f$ could be regarded as that $f$ frequently appears in $P$, and meanwhile, rarely appears in $P + U$. In text classification, this characteristic could be formalized as high support count (SC) and high inverse document frequency (IDF) of the feature in $P$ and $P + U$, respectively [34]. This idea could also be used to define $D_f$ in spammer group detection. However, since the feature values are numerical, we need to discretize them first

to calculate SC and IDF. If the possible values of a feature $f$ in both $P$ and $U$ form a sorted list $S$, to discretize $f$ into $k$ categories, $k-1$ cut points need to be found in $S$. In this paper, we employ the *Minimal Weighted Average Variance* [35] to determine the cut points. Assume a cut point value $c$ divides the list $S$ into two parts: $S_1^c$ and $S_2^c$, the weighted average variance (WAV) of $c$ on $S$ could be defined as

$$WAV_S^c = \frac{|S_1^c|}{|S|}Var(S_1^c) + \frac{|S_2^c|}{|S|}Var(S_2^c) \tag{2}$$

where $|S|$, $|S_1^c|$ and $|S_2^c|$ are the numbers of feature values in list $S$, $S_1^c$ and $S_2^c$, respectively, and $Var(S_1^c)$ and $Var(S_2^c)$ are the variances of all values in list $S_1^c$ and $S_2^c$, respectively. Thus, the ''best'' cut point is the value that could maximize the value of

$$\delta_c = Var(S) - WAV_S^c. \tag{3}$$

Then, to divide $S$ into $k$ parts, we employ the *Bisecting V-Clustering algorithm* to obtain $k-1$ cut points in a binary-recursive way. This algorithm first divides the list $S$ into two parts using the ''best'' cut points, and then repeatedly selects one part with the largest range and divides it into two parts again, until the number of parts arrives to $k$.

Suppose there are $K$ group features, a group could be represented as a $K$ dimension vector $g \in \mathbb{R}^K$ consisting of $K$ entries. After discretizing every feature value list into $k$ parts, we could obtain a new $K * k$ dimension feature space with each new feature $f_l^d$, $1 \le l \le K * k$, denotes a part of the original numerical feature. Then, a group can be represented as a $K * k$ dimension vector $g^d$, in which the $l$-th entry $g_l \in \{0, 1\}$, $1 \le l \le K*k$, denotes if the group contains the feature value in $f_l^d$. If $g_l = 1$, we say group $g$ contains the new feature $f_l^d$. Regard the vector $g^d$ as a transaction in frequent itemset mining, we could obtain the support count of feature $f_l^d$ in $P$, denoted as $SC_P(f_l^d)$. Analogously, Regard the vector $g^d$ as a document in text classification, we could obtain the inverse document frequency of the feature $f_l^d$ in $P + U$, denoted as $IDF(f_l^d)$. Thus, we define the feature strength function $D_{f_l^d}$ as follows:

$$D_{f_l^d} = SC_P(f_l^d)IDF(f_l^d) \tag{4}$$

where $SC_P(f_l^d)$ equals to the number of groups who contain $f_l^d$ in $P$, and $IDF(f_l^d)$ could be calculated as

$$IDF(f_l^d) = \log \frac{N_G}{1 + N_G^{f_l^d}} \tag{5}$$

where $N_G$ is the number of groups in $P + U$ and $N_G^{f_l^d}$ is the number of groups who contains feature $f_l^d$.

Given a feature, it is obviously that $N_G^{f_l^d} \swarrow \Rightarrow D_{f_l^d} \nearrow$, thus we could maximize the objection function by removing the instances containing $f_l^d$ from $U$, i.e., let $N_G^{f_l^d} = 0$. Therefore, the problem of $RN$ extraction could be described as: given $RN = U$ initially and the sorted list of features (obtained in terms of the feature strength function), to remove the instances containing this feature in $RN$, until $|RN| \simeq |P|$. Algorithm 1 summarizes the complete procedure of reliable negative set extraction, including feature discretizing, sorting and instance removing.

**Algorithm 1** Reliable Negative Set Extraction

**Input:** $P$: Labeled spammer group set; $U$: Unlabeled group set;
$k$: Discretization parameter, the number of categories for feature discretization.
**Output:** $RN$: Reliable negative instances set, a set of non-spammer groups.

**for** each feature $f \in P + U$ **do** ▷ Bisecting V-Clustering
  Calculate the sorted value list $S$ for each feature $f$;
  $\mathcal{C} \leftarrow \emptyset$; ▷ Initialize the set of cut points
  **while** $|\mathcal{C}| < k$ **do**
    Select a sub-list denoted as $S_j$, $1 \le j \le |\mathcal{C}| + 1$, with the largest range;
    $\forall c_i \in S_j$, calculate $\delta_{c_i}$ on $S_j$ according to Eq. (3);
    $p = \arg\max_i \delta_{c_i}$, $\mathcal{C} \leftarrow \mathcal{C} \bigcup \{c_p\}$;
  **end while** ▷ Now $\mathcal{C} = \{c_1, c_2, \ldots, c_{k-1}\}$
  Divide $S$ into $k$ parts according to $\mathcal{C}$;
**end for**
Construct a new feature space $F^d = \{f_1^d, f_2^d, \ldots, f_{K*k}^d\}$;
**for** each feature $f_l^d \in P$ **do** ▷ Only consider features appearing in $P$
  Calculate $D_{f_l^d}$ according to Eq. (4);
**end for**
Sort every $f_l^d \in P$ in $D$-decreasing order to form a list $F^d$;
$RN \leftarrow U$; ▷ Initially, $RN$ contains all instances of $U$
**for** each feature $f_l^d \in F^d$ from top to bottom **do**;
  Remove instances containing $f_l^d$ from $RN$;
  **if** $|RN|$ is close to $|P|$ **then**
    **return** $RN$;
  **end if**
**end for**

### B. SEMI-SUPERVISED LEARNING

After extracting the reliable negative set $RN$, we could obtain the labeled data set containing both positive and negative instances. Thus the PU-Learning problem could be converted into the well-known semi-supervised learning problem. Denote the labeled data set as $L$ ($L = P + RN$), we first train a Naive Bayes classifier on $L$ and then incorporate the unlabeled data set (denoted as $U$) with an Expectation Maximization (EM) algorithm to improve the initial classifier. It is also worth to mention that the discretized features are only utilized in reliable negative set extraction and we still use the numerical $K$ dimension feature space here for the classification.

Suppose a group $g$ is represented as a $K$ dimension vector $g = \{f_1, f_2, \cdots, f_K\}$ with each entry $f_i$, $1 \le i \le K$,

stands for a group feature. Assume the feature $f_i$ follows the normal probability distribution with mean $\mu_i$ and standard deviation $\sigma_i$. The probability of a group belonging to the class $Y$ (spammer group or non-spammer group) and having a feature $f_i = x_i$ can be calculated as

$$P(x_i|Y) = \frac{1}{\sqrt{2\pi}\sigma_{Yi}} exp(-\frac{(x_i - \mu_{Yi})^2}{2(\sigma_{Yi})^2}) \tag{6}$$

where $\sigma_{Yi}$ and $\mu_{Yi}$ are the mean and standard deviation of $f_i$ of the labeled data in class $Y$. Then, the probability of a spammer group candidate $g$ belonging to the class $Y$ can be calculated as

$$P(g|Y) = \prod_{i=1}^{K} P(x_{gi}|Y) \tag{7}$$

where $x_{gi}$ is the value of the $i$-th feature $f_i$ of group $g$. With Eqs. (6) and (7), we can calculate the probability of an unknown group belonging to a certain class (spammer group or non-spammer group). In this procedure, the labeled instances ($L$) are used to determine the parameters (mean and standard deviation) of probability distribution of each class. In semi-supervised learning, the unlabeled instances ($U$) could also be used to estimate more exact parameters, which means the improvement of the classifier.

To exploit the unlabeled data, we employ the Expectation Maximization (EM) algorithm, a widely used approach which iteratively re-estimates parameters by repeating the two kinds of steps (E-Step and M-Step) until the parameters converging to stationary values. In spammer group detection, the estimated parameters are the mean and standard deviation of both spammer group class and non-spammer group class. Particularly, to modulate the influence of the unlabeled data, we employ EM-λ, a variation of EM proposed in [36], which adds a weighting factor λ in the estimation. The detail iterative process of EM-λ is as follows:

- **E-Step:** Calculate the probability of each group $g_m$ belonging to a class as follows:

$$P(g_m \in Y) = P(Y|g_m) = \frac{P(Y)P(g_m|Y)}{P(g_m)} \tag{8}$$

  where $P(g_m)$ is constant. Assume the probabilities of the two class are equal for unknown instances, then $P(Y|g_m)$ is only determined by $P(g_m|Y)$ which can be obtained from Eqs. (6) and (7).
- **M-Step:** Estimate the parameters based on the probability obtained in E-Step. The mean of feature $f_i$ on the instances belonging to class $Y$ can be calculated as

$$\mu_{Yi} = \frac{1}{|Y|} \sum_{g=1}^{|Y|} \Omega_g x_{gi}. \tag{9}$$

  The standard deviation of feature $f_i$ on the instances belonging to class $Y$ can be calculated as

$$\sigma_{Yi} = \sqrt{\frac{1}{|Y|} \sum_{g=1}^{|Y|} \Omega_g^2 (x_{gi} - \mu_{Yi})^2} \tag{10}$$

In Eqs. (9) and (10), $|Y|$ represents the number of spammer groups or non-spammer groups, which is calculated by adding weight $\Omega_g$ as follows:

$$|Y| = \sum_{g=1}^{|L|+|U|} \Omega_g \tag{11}$$

where $|L|$ and $|U|$ denote the numbers of labeled and unlabeled instances, respectively. In Eqs. (9), (10) and (11), the weight $\Omega_g$ could be calculated by the probability of a group belonging to a certain class as follows:

$$\Omega_g = P(Y|g_m) = \frac{P(g \in Y)}{\sum_j P(g \in Y_j)}. \tag{12}$$

To modulate the influence of unlabeled data, an additional parameter $\Lambda(g)$ is defined as

$$\Lambda(g) = \begin{cases} \lambda, & if\ g \in U \\ 1, & if\ g \in L \end{cases} \tag{13}$$

where λ is the weighting factor. Then, we could rewrite $\Omega_g$ as

$$\Omega_g = P(Y|g_m) = \frac{\Lambda(g)P(g \in Y)}{\sum_j P(g \in Y_j)}. \tag{14}$$

Obviously, EM-λ has the same E-Step as EM and involves an additional parameter $\Lambda(g)$ in M-Step to modulate the influence of unlabeled data. When λ is close to zero, the unlabeled data will have little influence to the shape of EM's hill-climbing surface. However, when $\lambda = 1$, each unknown group will be weighted as known spammer group or non-spammer group, and EM-λ squints towards to the original EM algorithm. Algorithm 2 summarizes the procedure of training a classifier using semi-supervised learning.

## V. EXPERIMENT

In this section, we evaluate the performance of PSGD on a real-life dataset, including the overall performance, impact of parameter λ and the effectiveness of *RN* extraction algorithm.

### A. EXPERIMENTAL SETUP

#### 1) DATASET

The dataset is collected from Amazon China (http://www.amazon.cn), one of the most popular e-commerce platforms in China, from September 2000 to December 2011. We totally crawled 2,151,963 reviews of 88,972 users on 504,171 products. To remove those cold-start products and users, we first extract the products that have been reviewed over 15 times, and then pick out users who have reviewed over 20 times. As a result, we finally obtain 469,392 reviews and ratings from 9,423 users on 19,185 products.

To evaluate the performance of PSGD for solving the spammer group detection problem, we first extract the spammer group candidates from the 9,423 users using FIM. By setting the support count as 20, which means the group members

must have co-reviewed at least 20 products, we finally extract 4,298 spammer group candidates. As there is no ground-truth for evaluation, we employ 8 post-graduate students as experts to label these candidates. All of them are very familiar with e-commerce environment and have studied the topic of review spam detection more than 2 years. Each expert works alone and is encouraged to use his/her own understanding to label groups. Due to the large number of the instances (spammer group candidates), the experts are divided into 4 groups with each consisting of two experts, and each expert group labels a quarter of all instances, i.e., about 1074 instances. In the procedure of labeling, if the two experts vote consistently, we take their judgment as the final label. Otherwise, if the two experts give conflictive judgment, we temporarily add another expert from other expert groups and the final label obeys the majority's judgement. Summarizing their judgment, we finally obtain 1,512 spammer groups and 1,786 non-spammer groups from the 4,298 group candidates.

**Algorithm 2** Semi-Supervised Learning

**Input:** $L$: Labeled group set, $U$: Unlabeled group set.
**Output:** $\theta$: The classifier as spammer group detector.

1: Train an initial Naive Bayes classifier $\theta$ based on the labeled group set, $L$, only;
2: **repeat**
3: Utilize the current $\theta$ to calculate the probability of $g_m$ by Eq. (8); ▷ E-Step
4: Improve $\theta$ by re-estimating the parameters utilizing Eqs. (9) to (14); ▷ M-Step
5: **until** none of estimated parameters $\mu_{Yi}$ and $\sigma_{Yi}$ changes
6: Return the classifier $\theta$ as the spammer group detector;

### 2) BASELINES AND EVALUATION METRICS

Two supervised learning classification models: Naive Bayes Classifier (NB) and Support Vector Machine (SVM), and two unsupervised ranking methods: GSRank [12] and PCA [14], are used as the baselines for the purpose of comparison. Both NB and SVM are run on `WEKA`[1] in their default settings. The parameters of GSRank and PCA are set following the suggestion in their original papers [12], [14].

To evaluate the effectiveness of PSGD, we use Precision ($P$), Recall ($R$) and F1-Score ($F1$) as the metrics where

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \quad F1 = \frac{2PR}{P + R} \tag{15}$$

where $TP$ is the number of truly identified spammer groups, $FP$ is the number of wrongly identified spammer groups, and $FN$ is the number of missed spammer groups.

### 3) FEATURE CONSTRUCTION

A number of group features are defined for distinguishing spammer groups against normal groups in the literature [12], [14], and [37]. In this paper, we construct a total of 12 features for the spammer group detection problem. Among them, 10 features are selected from the literatures, including *Group Time Window* (*GTW*), *Group Deviation* (*GD*), *Group Early Time Frame* (*GETF*), *Group Size Ratio* (*GSR*), *Group Size* (*GS*), *Group Support Count* (*GSUP*) from [12], *Product Tightness* (*PT*), *Rating Variance* (*RV*), *Product Reviewer Ratio* (*PRR*) from [37] and *Average Active Interval* (*AAI*) from [14]. However, the linguistic features in the literatures, such as *Group Content Similarity* (*GCS*) and *Group Member Content Similarity* (*GMCS*), are excluded because they often underperform in discriminating spammer/non-spammers according to the study in [13]. In addition, we define two new group features: *Group Common Products Ratio* (*GCPR*) and *Group Amazon Verified Purchase* (*GAVP*). In particular, if the members of a group concentrate on a certain number of products and barely review any other products, this group is more likely to be a spammer group. In allusion to this rule, we could calculate the common products ratios (common products mean products co-reviewed by all the group members) of the members and then aggregate them to evaluate the group spamicity. Thus we construct the group feature *GCPR*, which utilizes geometric mean to aggregate the group members' common products ratios and is defined as

$$GCPR(g) = \sqrt[|g|]{\prod_{r \in g} \frac{|P_g|}{|P_r|}}$$

$$P_g = \bigcap_{r \in g} P_r \tag{16}$$

where $g$ and $r$ are the group and reviewer, respectively, and $|g|$ is the number of reviewers in group $g$. $P_r$ is the products set reviewed by reviewer $r$.

In the common sense, if a reviewer has really bought the product he/she reviewed, this reviewer is more likely to be a normal customer instead of a spammer. As Amazon provides purchase verification to indicate if the reviewer has really brought the product, we could exploit this mechanism to evaluate the spamicity of a group by calculating the verified purchase ratio of this group. Thus we construct the group feature *GAVP*, which is defined as

$$GAVP(g) = Avg_{p \in P_g}(1 - \frac{|V_g(p)|}{|R_g(p)|}) \tag{17}$$

where $g$ and $p$ are the group and product, respectively, and $P_g$ is the common products set as in Eq. (16). $R_g$ and $V_g$ are the reviews and verified purchases come from the group $g$, respectively.

To evaluate the new proposed group features, we construct two feature sets. One consists of the traditional group features in the existing literatures, denoted as $GFS\_T$, and $GFS\_T = \{GTW, GD, GETF, GSR, GS, GSUP, PT, RV, PRR, AAI\}$. The other feature set adds the new group features, denoted as $GFS\_N$, and $GFS\_N = GFS\_T \cup \{GCPR, GAVP\}$. We will

[1] http://www.cs.waikato.ac.nz/ml/weka/

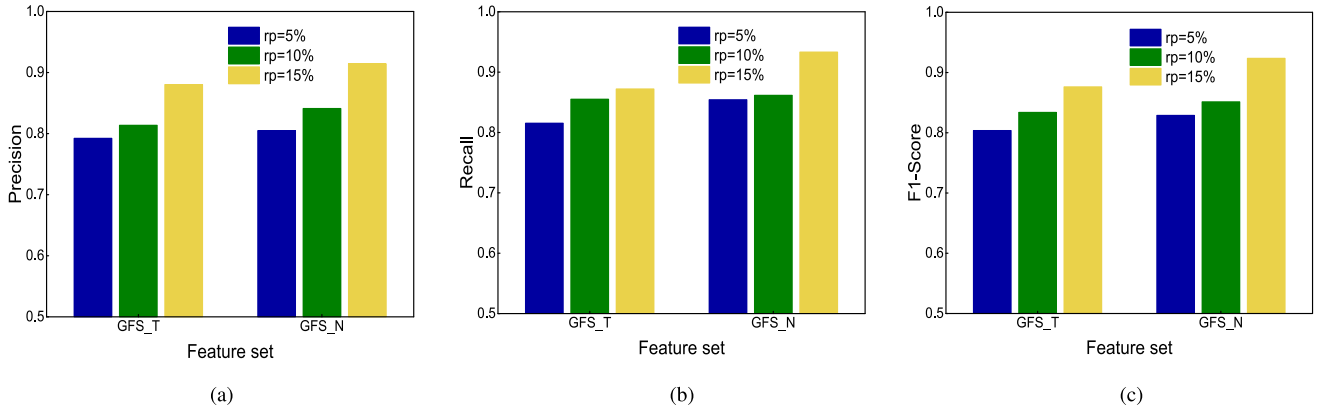


**FIGURE 2.** The precision, recall and F1-score of PSGD. (a) Precision. (b) Recall. (c) F1-score.

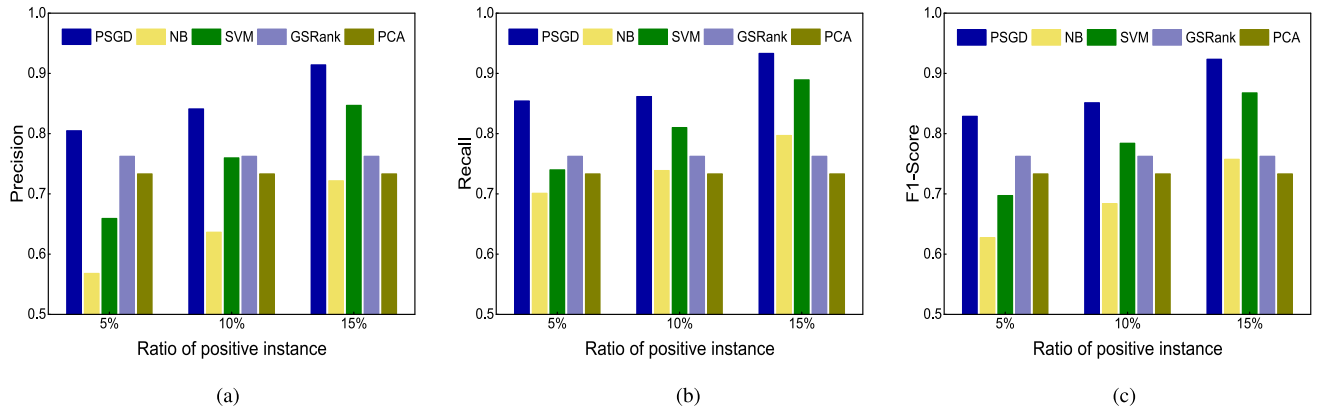


**FIGURE 3.** Performance comparison of different detection methods. (a) Precision. (b) Recall. (c) F1-score.

compare the detection performance on these two feature sets in the following section.

### B. OVERALL PERFORMANCE

Here we present the overall performance of PSGD. We compare the performance on different scales of $P$ set by setting the ratios of positive instance (denoted as $rp$) as 5%, 10% and 15%, i.e., 75, 150 and 225 out of the 1,512 spammer groups are picked out to construct $P$ sets, respectively. Set the parameter $\lambda = 0.5$, we evaluate the Precision, Recall and F1-Score of PSGD and the experimental results are shown in Fig. 2. As can be seen, except the Precision on $GFS\_T$ when $rp = 5\%$, the values of all the three metrics are more than 0.8, and even over 0.9 when $rp = 15\%$ on $GFS\_N$. These results demonstrate that PSGD is effective on spammer group detection. Moveover, PSGD performs better on $GFS\_N$ than on $GFS\_T$, which demonstrates that our new proposed group features, i.e., GCPR and GAVP, have high degree of distinction between spammer groups and non-spammer groups, and could significantly improve the performance of detection. Another observation is that the performance of PSGD improves as the ratio of positive instance gets larger, which demonstrates that although our PSGD model involves both labeled and unlabeled data, the labeled data plays a more important role.

We also give the performance comparison with the baselines on $GFS\_N$ when $rp = 5\%$, 10% and 15%. To detect spammer groups using supervised learning methods NB and SVM, we additionally pick out some negative instances from the 1,786 non-spammer groups to construct training set combining with $P$. The number of selected negative instances equals to the size of the relative $P$ set, and we finally obtain 3 training sets containing 150, 300 and 450 labeled instances, respectively. As GSRank and PCA output ranked group lists in terms of the calculated suspicious degree of groups, it needs a threshold to divide the ranked group list into spammer groups and non-spammer groups. In this experiment, we output the lists in descending order and set the threshold as 1,512, the number of spammer groups in the ground-truth, and the groups above the threshold in the list are identified as spammer groups. The experimental results are shown in Fig. 3. As can be seen, our PSGD model outperforms all the baselines on all the three ratios of positive instance. However, NB generally performs worse than GSRank and PCA until $rp$ gets to 15%. Moreover,

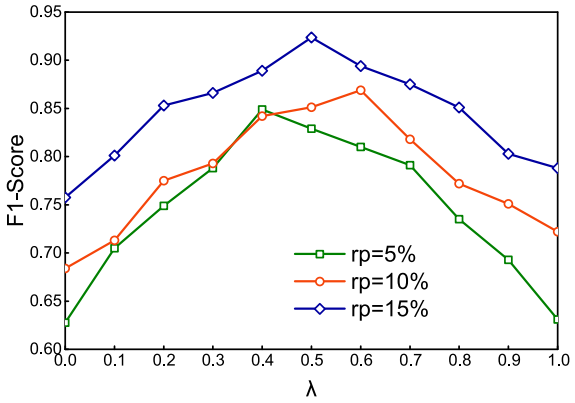


**FIGURE 4.** Impact of λ on F1-score.

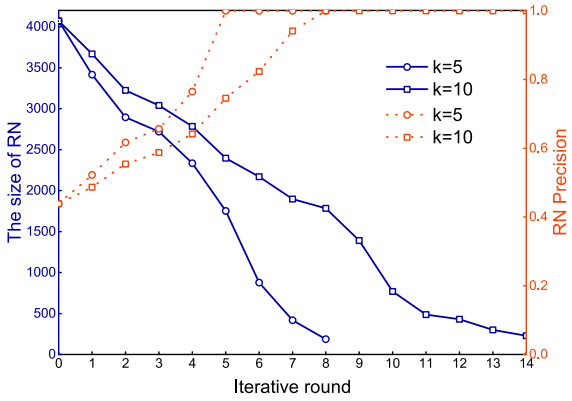


**FIGURE 5.** Size and precision tracking of *RN*.

SVM performs better than NB that it outperforms GSRank and PCA when $rp = 10\%$. This is because SVM usually has better adaptability on small training set than NB. The comparison results demonstrate that to obtain high performance, the supervised learning models require a lot of labeled instances to train the classifier, however, as involving the unlabeled instances, the partially learning model PSGD needs less labeled instance and only positive instances. This is significant because the instances can only be labeled by manual and large-scale labeling is daunting and even impossible.

It is worth mentioning that as some active measures need to be taken for the detected spammer groups (e.g., to delete the member's account), the profiles of each group member need strict further checking because falsely deleting a normal user is usually much worse than neglecting a spammer. However, the detection results provided by PSGD could significantly decrease the checking cost for administrators.

### C. IMPACT OF WEIGHTING FACTOR λ

The parameter of weighting factor λ modulates the weight of unlabeled data in our PSGD model. In this section, we investigate the impact of λ on detection performance. By ranging the parameter λ from 0 to 1 with 0.1 as the interval, we evaluate the performance of PSGD on *GFS_N* when $rp = 5\%$, $10\%$ and $15\%$, respectively. Fig. 4 shows the experimental results, from which we can see that the F1-Score varies with the increase of λ on all the three ratios of positive instance. All the three curves in Fig. 4 represent similar variation tendency and reach a peak when the λ value is intermediate. From Fig. 4, we can also find that the curve with small ratio of positive instance changes more significantly than that with larger ratio of positive instance. This indicates that when the number of labeled positive instances is very small, carefully selecting λ will significantly improve the practical performance of spammer group detection even further. Therefore, we can utilize cross-validation approach to determine the optimal λ value for the specific data. In this case, λ set between 0.4 and 0.6 resolves in satisfying results.

### D. EFFECTIVENESS OF RN EXTRACTION

As mentioned in Sec. IV-A, PSGD iteratively removes instances containing the highest discriminative feature from the current *RN*. To demonstrate the effectiveness of this *RN* extraction algortihm, we take $|P| = 225$ (i.e., $rp = 15\%$) as an example and track the *RN* Precision (defined as $|TRN|/|RN|$ where $|TRN|$ stands for the number of truly negative instances in *RN*) in each iterative round. We also investigate the impact of the discretization parameter, $k$, on *RN* extraction algorithm. Fig. 5 presents the variation of *RN* Precision and the size of *RN* when the discretization parameter $k$ is set as 5 and 10, respectively. Initially, the *RN* Precision is in fact the ratio of non-spammer groups to all groups, i.e., about 44%. With the operation of the algorithm, it removes the identified positive instances and adjusts the scale of *RN* in each iterative round. It can be seen that all positive instances have been removed (i.e., the *RN* Precision soars to 1) before $|RN|$ has been decreased to close to $|P|$. However, in practice, as there is no ground-truth to tell us that all the positive instances have been removed, the algorithm needs to go on to remove the instances until $|RN| \simeq |P|$. As all the remainder instances in the current *RN* are negative instances because all the positive instances have been removed, no matter what instances are further removed, we still have $|TRN| = |RN|$. As a result, the extracted reliable negative set yields 100% accuracy in our experiments.

In particular, when $k = 5$, the size of *RN* decreases more quickly and the *RN* Precision soars to 1 earlier. This is expected as smaller discretization parameter brings less partitions of feature value list, causing more instances to contain the specific features which are then removed in each iterative round. However, it does not mean that the smaller the discretization parameter is, the better performance the *RN* extraction algorithm is. The reason is that each specific feature will be contained in more instances when discretization parameter is small, thus the discriminative power of the features will decrease. In this case, we are able to set discretization parameter between 5 and 15 to obtain satisfying results.

## VI. CONCLUSION AND FUTURE WORK

This paper proposes a partially supervised learning based model PSGD to detect spammer groups from product reviews. First, the PSGD model uses frequent item mining (FIM) to discover spammer group candidates from the review data. Then, by manually labeling some spammer groups as positive instances, the PSGD employs PU-Learning to construct a classifier from the positive and unlabeled instances to identify the real spammer groups from group candidates. In particular, the PSGD defines a feature strength function to measure the discriminative power of group features, and then iteratively removes instances containing high discriminative features from the unlabeled instances set to obtain a reliable negative set consisting of only non-spammer groups. By combining the positive, negative and unlabeled instances, we convert the PU-Learning problem into the well-known semi-supervised learning problem, and employ Naive Bayesian model and EM algorithm to construct a classifier as spammer group detector. Experiments on Amazon.cn demonstrate that the proposed PSGD model outperforms both supervised and unsupervised learning methods on spammer group detection.

Our future work in the area will focus on the improvement of the PSGD model. Beyond the Naive Bayesian model used in PSGD, we will investigate and incorporate more classification models such as neural network, Semi-Supervised SVM (S3VM) and even ensemble methods. On the positive instances acquisition and *RN* extraction, we plan to involve active learning to improve the accuracy and efficiency of data labeling. Another problem needs to be solve is that how to verify if *RN* Precision reaches 1 when $|RN| \simeq |P|$, because the purity of reliable negative set do has significant influence for the accuracy of classification. Moreover, if the *RN* indeed has a precision lower than 1, a method evaluating the influence of the mislabeled instances will be needed to decide if the *RN* can be used.

## REFERENCES

[1] F. Zhu and X. Zhang, "Impact of online consumer reviews on sales: The moderating role of product and consumer characteristics," *J. Marketing*, vol. 74, no. 2, pp. 133–148, 2010.

[2] K. C. Santosh and A. Mukherjee, "On the temporal dynamics of opinion spamming: Case studies on yelp," in *Proc. 25th Int. Conf. World Wide Web*, 2016, pp. 369–379.

[3] N. Jindal and B. Liu, "Opinion spam and analysis," in *Proc. Int. Conf. Web Search Data Mining*, 2008, pp. 219–230.

[4] F. Li, M. Huang, Y. Yang, and X. Zhu, "Learning to identify review spam," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2011, vol. 22. no. 3, pp. 219–230.

[5] S. Xie, G. Wang, S. Lin, and P. S. Yu, "Review spam detection via temporal pattern discovery," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 823–831.

[6] A. Mukherjee *et al.*, "Spotting opinion spammers using behavioral footprints," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 632–640.

[7] L. Akoglu, R. Chandy, and C. Faloutsos, "Opinion fraud detection in online reviews by network effects," in *Proc. ICWSM*, vol. 13. 2013, pp. 2–11.

[8] S. Shehnepoor, M. Salehi, R. Farahbakhsh, and N. Crespi, "NetSpam: A network-based spam detection framework for reviews in online social media," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 7, pp. 1585–1595, Jul. 2017.

[9] Z. Wu, Y. Wang, Y. Wang, J. Wu, J. Cao, and L. Zhang, "Spammers detection from product reviews: A hybrid model," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2015, pp. 1039–1044.

[10] A. Heydari, M. Tavakoli, and N. Salim, "Detection of fake opinions using time series," *Expert Syst. Appl.*, vol. 58, pp. 83–92, Oct. 2016.

[11] H. Li *et al.*, "Bimodal distribution and co-bursting in review spam detection," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 1063–1072.

[12] A. Mukherjee, B. Liu, and N. Glance, "Spotting fake reviewer groups in consumer reviews," in *Proc. 21st Int. Conf. World Wide Web*, 2012, pp. 191–200.

[13] C. Xu, J. Zhang, K. Chang, and C. Long, "Uncovering collusive spammers in Chinese review websites," in *Proc. 22nd ACM Int. Conf. Conf. Inf. Knowl. Manag.*, 2013, pp. 979–988.

[14] Y. Wang, Z. Wu, Z. Bu, J. Cao, and D. Yang, "Discovering shilling groups in a real e-commerce platform," *Online Inf. Rev.*, vol. 40, no. 1, pp. 62–78, 2016.

[15] B. Liu and W. S. Lee, "Partially supervised learning," in *Web Data Mining*. Berlin, Germany: Springer, 2011, pp. 171–208.

[16] L. Zhang, Y. Yuan, Z. Wu, and J. Cao, "Semi-SGD: Semi-supervised learning based spammer group detection in product reviews," in *Proc. 5th Int. Conf. IEEE Adv. Cloud Big Data (CBD)*, Aug. 2017, pp. 368–373.

[17] A. Heydari, M. A. Tavakoli, N. Salim, and Z. Heydari, "Detection of review spam: A survey," *Expert Syst. Appl.*, vol. 42, no. 7, pp. 3634–3642, 2015.

[18] C. L. Lai, K. Q. Xu, R. Y. K. Lau, Y. Li, and L. Jing, "Toward a language modeling approach for consumer review spam detection," in *Proc. IEEE 7th Int. Conf. e-Bus. Eng. (ICEBE)*, Nov. 2010, pp. 1–8.

[19] S. Shojaee, M. A. A. Murad, A. B. Azman, N. M. Sharef, and S. Nadali, "Detecting deceptive reviews using lexical and syntactic features," in *Proc. IEEE 13th Int. Conf. Intell. Syst. Design Appl. (ISDA)*, Dec. 2013, pp. 53–58.

[20] N. Jindal, B. Liu, and E.-P. Lim, "Finding unusual review patterns using unexpected rules," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manag.*, 2010, pp. 1549–1552.

[21] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock, "Finding deceptive opinion spam by any stretch of the imagination," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1. 2011, pp. 309–319.

[22] M. Ott, C. Cardie, and J. Hancock, "Estimating the prevalence of deception in online review communities," in *Proc. 21st Int. Conf. World Wide Web*, 2012, pp. 201–210.

[23] A. Mukherjee, V. Venkataraman, B. Liu, and N. Glance, "What yelp fake review filter might be doing?" in *Proc. 7th Int. AAAI Conf. Weblogs Soc. Media*, 2013, pp. 409–418.

[24] G. Fei, A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, and R. Ghosh, "Exploiting burstiness in reviews for review spammer detection," in *Proc. ICWSM*, vol. 13. 2013, pp. 175–184.

[25] E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw, "Detecting product review spammers using rating behaviors," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manag.*, 2010, pp. 939–948.

[26] S. Rayana and L. Akoglu, "Collective opinion spam detection: Bridging review networks and metadata," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 985–994.

[27] A. Fayazi, K. Lee, J. Caverlee, and A. Squicciarini, "Uncovering crowdsourced manipulation of online reviews," in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2015, pp. 233–242.

[28] G. Wang, S. Xie, B. Liu, and P. S. Yu, "Review graph based online store review spammer detection," in *Proc. IEEE 11th Int. Conf. Data Mining (ICDM)*, Dec. 2011, pp. 1242–1247.

[29] D. H. Fusilier, M. Montes-y-Gómez, and P. Rosso, "Using PU-learning to detect deceptive opinion spam," in *Proc. WASSA*, 2013, pp. 1–8.

[30] H. Li, Z. Chen, B. Liu, X. Wei, and J. Shao, "Spotting fake reviews via collective positive-unlabeled learning," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Dec. 2014, pp. 899–904.

[31] H. Li, B. Liu, A. Mukherjee, and J. Shao, "Spotting fake reviews using positive-unlabeled learning," *Comput. Sistemas*, vol. 18, no. 3, pp. 467–475, 2014.

[32] Y. Ren, D. Ji, and H. Zhang, "Positive unlabeled learning for deceptive reviews detection," in *Proc. EMNLP*, 2014, pp. 488–498.

[33] X.-L. Li, P. S. Yu, B. Liu, and S.-K. Ng, "Positive unlabeled learning for data stream classification," in *Proc. SIAM Int. Conf. Data Mining*, 2009, pp. 259–270.

[34] G. P. C. Fung, J. X. Yu, H. Lu, and P. S. Yu, "Text classification without labeled negative documents," in *Proc. 21st Int. Conf. Data Eng. (ICDE)*, Apr. 2005, pp. 594–605.
[35] J. Yuan *et al.*, "T-drive: Driving directions based on taxi trajectories," in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geogr. Inf. Syst.*, 2010, pp. 99–108.
[36] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using EM," *Mach. Learn.*, vol. 39, no. 2, pp. 103–134, 2000.
[37] Z. Wang, T. Hou, D. Song, Z. Li, and T. Kong, "Detecting review spammer groups via bipartite graph projection," *Comput. J.*, vol. 59, no. 6, pp. 861–874, 2016.

**LU ZHANG** received the B.S. and Ph.D. degrees in computer science from Southeast University, Nanjing, China, in 2005 and 2012, respectively. He is currently a Lecturer with the Jiangsu Provincial Key Laboratory of E-Business, School of Information Engineering, Nanjing University of Finance and Economics. His current research interests include data mining, machine learning, and recommender systems. He is a member of the ACM and CCF.

**ZHIANG WU** (M'17) received the B.S. degree in computer science from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2004, and the Ph.D. degree in computer science from Southeast University, Nanjing, in 2009. He is currently a Full Professor with the Jiangsu Provincial Key Laboratory of E-Business, School of Information Engineering, Nanjing University of Finance and Economics. His recent research focuses on distributed computing, data mining, e-commerce intelligence, and social network analysis. He has published over 30 refereed journal and conference papers in these areas. He is a member of the ACM and a Senior Member of CCF.

**JIE CAO** received the B.S. degree from the Nanjing University of Science and Technology, Nanjing, China, in 1992, and the M.S. and Ph.D. degrees from Southeast University, Nanjing, in 1998 and 2002, respectively, all in computer science. He is currently a Chief Professor and the Dean of the School of Information Engineering, Nanjing University of Finance and Economics. His main research interests include data mining, big data, and e-commerce intelligence. He is a member of the ACM and CCF. He has been selected in the Program for New Century Excellent Talents in University and received the Young and Mid-aged Expert with Outstanding Contribution in Jiangsu Province.