Detecting Collusive Spammers in Online Review Communities

Conference Paper · November 2013

DOI: 10.1145/2513166.2513176

CITATIONS
3

READS
156

1 author:

Chang Xu
Nanyang Technological University
5 PUBLICATIONS **18** CITATIONS

SEE PROFILE

Detecting Collusive Spammers in Online Review Communities

Chang Xu
School of Computer Engineering,
Nanyang Technological University, Singapore
xuch0007@e.ntu.edu.sg

ABSTRACT

In this paper, we first define our research problem as to detect collusive spammers in online review communities. Next we present our current progress on this topic, in which we have spotted anomalies by evaluating 15 behavioral features proposed in the state-of-the-art approaches. Then we propose a novel hybrid classification/clustering method to detect colluders in our dataset based on selected informative features. Experimental results show that our method promisingly improve the performance of traditional classifiers by incorporating clustering for the smoothing. Finally, possible extensions of our current work and challenges in achieving them are discussed as our future directions.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Filtering

Keywords

Collusive Spammer; Spam Review Detection; Smoothing

1. RESEARCH PROBLEM

With the growing availability of review services at online stores (e.g., amazon.com) and opinion sharing websites (e.g., epinions.com), product reviews have become an indispensable part of online shopping; shoppers nowadays will not purchase a product without reading a product review. Unfortunately, many of the reviews they read may not be that genuine as expected. It has been found that some paid professionals fabricate reviews without even using the products, with the sole goal of promoting a product or demoting a competitor's product.

Our research is within the field of opinion mining in the Web. Specifically, we aim at detecting opinion spam in online review communities. Opinion spam [3] refers to malicious activities (e.g., posting deceptive reviews) carried out to intentionally mislead readers or opinion analysis systems by giving unfair positive opinions to some target entities (e.g., products, hotels, and restaurants) so as to boost their credits and/or by providing negative opinions to demote their reputations. Instead of directly identifying spam reviews from the huge amount of online review corpus, we are particularly interested in addressing the problem of detecting collusive spammers (colluders) in online review communities. Collusive spammers refers to a groups of hired malicious users (or controlled user accounts) who *collaboratively* write fake reviews for some specified entities, coordinated and/or guided by one or more shadow organizers or organizations. We focus on colluders because:

- Paid spammers are more likely to be well organized and collaborate on tasks coordinated by a shady organizer, thus spamming campaigns can be committed in a much more cost-effective and undetectable way [1];
- Collaborating spammers will exert immense control over the opinions of a target, thereby undermining the trustworthiness of the review site [6];
- By colluding with each other on many target entities within different spamming campaigns, colluders are more likely to be given away by their collective behaviors (e.g., by copying each other's reviews). Then interesting observations might be made and corresponding methods can be invented to effectively and efficiently catch them collectively by exploiting the collusive behavioral evidence.

2. RELATED WORK

Many existing approaches attempt to detect fraud in online review communities by either carefully constructing specific features or exploiting the relationships among different types of entities (e.g., reviews, reviewers, reviewer groups, and reviewed targets) or the combination of both. The literature can be further classified into three categories according to their respective detecting objectives.

Review centric approaches. In terms of spam review detection, Jindal and Liu [3] use supervised learning to classify spam reviews into three categories: untruthful reviews, reviews on brands only, and non-reviews. A number of engineered features, which are constructed according to different levels of information (i.e., the text of the review, the reviewer writing the review, and the product being reviewed), are used to detect the last two types of spam reviews. To fight against the untruthful reviews, duplicate spam reviews are used as the positive examples to train a logistic regression

PIKM'13, November 1, 2013, San Francisco, CA, USA.

ACM 978-1-4503-2411-3/13/10 ...$15.00
http://dx.doi.org/10.1145/2513166.2513176.

model for the classification. Li *et al.* [4] also identify spam reviews using carefully extracted features. However, based on the observation that spammers will consistently write spam reviews, they proposed a two-view semi-supervised method to detect spam reviews by using two sets of features - review features and reviewer features - for the co-training. By taking advantage of the studies from psychology and computational linguistics, Ott *et al.* [7] concentrates on detecting deceptive opinion spam through insights obtained from the review text only. They formulate the task from three complementary perspectives, i.e., text categorization, psycholinguistic deception detection, and genre identification. Corresponding features are used to train traditional classifiers (Naive Bayes and Support Vector Machine) on a collection of pseudo-ground truth - fake reviews written by non-experts hired from Crowdsourcing services. They report that nearly 90% accuracy has been achieved by using a combined classifier with both n-gram and psychological deception features.

Reviewer centric approaches. Instead of focusing on the problem of detecting spam reviews, Lim *et al.* [5] argue that review spammer detection is preferred because gathering behavioral evident of spammers is much easier and scalable than that of spam reviews. Thus they propose a user behavior driven review spammer detection approach via an aggregated scoring model, which ranks reviewers based on their behavioral patterns. A collection of target-based spamming patterns and deviation-based spamming patterns are considered. They also demonstrate that by removing the reviewers with very high spam scores, the corresponding spammed products are shown to be subjected to considerable changes in terms of the aggregate rating and review count. Recently, a more sophisticated graph-based method have been proposed by Wang *et al.* [9] to detect review spammers by considering the relationships among reviews, reviewers and reviewed stores. They construct a heterogeneous graph to capture such relationships. Besides, three types of concepts - the trustiness of reviewers, the honesty of reviews, and the reliability of stores - are quantified to represent the degree of goodness of the entities. The quantities are computed iteratively through propagation over the graph according to specific interactions among the entities. Finally, suspicious reviewers are expected to be revealed via ranking their trustiness scores.

Reviewer group centric approaches. Mukherjee *et al.* [6] study collusive spammers, which are group of spammers who write spam reviews collaboratively to promote or demote some target products. Reviewer-group based features are proposed to capture the aggregated behavioral patterns exhibited by spammer groups. Finally, an iterative-based ranking algorithm is proposed to rank candidate groups by exploiting the relationships among groups, group members and reviewed products.

3. PROGRESS TO DATE

3.1 Overview

As our first step, we currently exert our efforts on detecting colluders in Chinese online review systems. Because we found that, on one hand, spamming activities are especially critical in China given the great rich-poor divide and massive Internet population, and little research has been conducted on Chinese review datasets. According to China Internet Network Information Center (CNNIC)[1], China's Internet users have hit 564 million by the end of 2012. A whopping 37.1% of the entire users are students and unemployed, who are the ideal source of low-cost manpower for information diffusion on the Internet. In fact, nowadays people are also found to join the "Internet Water Army" [1] to post spam reviews in online review sites, for 0.10 to 0.50 CNY per posting. The "Internet Water Army" has evolved to operate in small scales, where small coalitions of spammers collude to generate a desired result without showing up on the radar of site moderators and regulators. On the other hand, we want to find out whether those proposed state-of-the-art features are still capable to catch spammers in today's online review systems. We believe that, from an adversarial view of point, colluders may gradually evolve to adapt to evade existing anti-spam approaches that equip with the outdated features by changing their tactics accordingly. Once features are found to be inferior to discriminate spammers from non-spammers, new techniques become urgent to fill the gaps. Up to now, we have:

- empirically evaluated 15 behavioral features proposed by the state-of-the-art approaches on a newly crawled Chinese-Language review dataset, anomalies are spotted in the behaviors of spammers. Analysis and explanations are provided for such anomalies;

- proposed a hybrid classification/clustering method to detect collusive spammers in our dataset, based on the observation that collusive spammers are implicitly correlated through their belonging groups. Such groups are formed by capturing specific collective behaviors (e.g., write fake reviews for common target products) of a number of spammers. This type of information is considered to be hard to fake in practice because for example if they do not review the target products they will not get paid in the end. Experiments show that smoothing the results of traditional classifiers via clustering can achieve better results by revealing the colluders that belong to small groups.

3.2 Spotting Anomalies

3.2.1 Data Acquisition and Annotation

Since to the best of our knowledge no annotated dataset about colluders in Chinese online reviews has been published yet, we then created one by crawling consumer reviews from Amazon.cn (the Chinese counterpart of Amazon.com). Specifically, we crawled a snapshot of *manufacturing product* reviews which contains 1,205,125 reviews written by 645,072 reviewers on 136,785 products (e.g., electronics, housewares). Each review has 6 attributes: ReviewerID, ProductID, Product Brand, Rating, Date and Review Text. To collect sufficient colluders for evaluation, we use frequent itemset mining (FIM), similar with [6], to search for the places where colluders would probably be found. In such context, reviewer IDs are regarded as *items*, each transaction is the set of reviewer IDs who have reviewed a particular product. Through FIM, groups of reviewers who have reviewed multiple common products can be found. Here we use maximal frequent itemset mining (MFIM) to discover groups with maximal size since we focus on the worst spam-

[1]http://www.cnnic.net.cn/

ming activities in our dataset. Following the same parameter settings as [6] that minimum support is 3 and minimum group size is 2, 8,915 groups are found, each represented as a mapping from a set of members to a set of commonly reviewed products.

During the annotation, we consider a reviewer as colluder if (s)he involves in at least one of the spam groups. A group is considered as a spam group if the behaviors of its members matches some practical spamming indicators w.r.t colluders from previous studies [9, 6], online discussions and our own observations. As a result, due to the limited time, only 2,280 groups have been examined, of which 837 are labeled as spam group and 1,443 as non-spam group. Moreover, in terms of reviewers, according to the above mentioned rules, a total of 742 colluders and 1,882 non-colluders are identified from those 2,280 groups.

3.2.2 Feature Evaluation

Nowadays, most of the state-of-the-art spam review tackling techniques follow the procedure of learning classification models using strong engineered features. The trained models are then used to conduct further detection. However, one major problem is that such feature based solutions do not generalize. From an adversarial view of point, colluders may eventually evolve to adapt to evade existing anti-spam approaches that equip with these outdated features by changing their tactics accordingly. Thus models trained today may not be able to catch smarter colluders tomorrow any more. Moreover, feature selection becomes a necessary subprocedure to handle the difference between the distributions of different datasets. For example, we have evaluated 15 behavioral features by using cumulative histograms (CHs) on the dataset mentioned in Section 3.2.1[2]. The cumulative histograms (CHs) is a good way to show the distribution of feature scores over colluders and non-colluders respectively, which provides a visual intuition for how well each feature discriminates colluders. The larger the gap between the two curves of two distributions are, the better the discrimination capability will be achieved. Such behavioral features can be divided by two sub-categories, i.e., individual behavioral features and collusive behavioral features. The list of individual behavioral features (IBFs) is as follows (these features were also used in [4] and [5]):

For each reviewer:

- Number of posted reviews;
- Brand Deviation Score (BDS) measures the deviation in his review counts over different brands;
- Rating Deviation Score (RDS) measures the variance of his ratings over different brands.
- Targeting Products (TP)[3] evaluates for a particular product how similar all his reviews are in terms of ratings and contents;
- Targeting Product Groups (TPG) measures the pattern of ratings towards a set of products sharing common attributes (e.g., brand) within a short duration;
- General Deviation (GD) measures the average difference between the reviewer rating on one product versus the product's average rating;
- Early Deviation (ED) measures how early ratings of the product deviate from the average rating of that same product.

The CHs of IBFs are shown in Figure 1.

Figure 1: Distributions of colluders (solid) and non-colluders (dashed) vs. IBFs.

The list of collusive behavioral features (CBFs) is as follows (these features were also used in [6]):

- Group Time Window (GTW) evaluates how close, temporally, the members of a group write reviews for a particular product;
- Group Deviation (GD) flags a group as suspicious if its ratings diverge significantly from other reviewers on the same product;
- Group Content Similarity (GCS) calculates the maximum average pairwise content similarity among group member reviews;
- Group Member Content Similarity (GMCS) calculates the maximum average pairwise content similarity among individual member's reviews;
- Group Early Time Frame (GETF) reveals how early a group post reviews of common products;
- Group Size Ratio (GSR) measures the average ratio of group size to the total number of reviewers for each product;
- Group Size (GS) is the number of members in a group;
- Group Support Count (GSUP) records the number of common products.

Note that these CBFs are originally group-based while our task is reviewer-based. We thus first generate the group-based scores for each reviewer group, and then each reviewer will inherit the group-based score from the corresponding group. For those involved in multiple groups the highest score will be chosen, as we intend to capture the worst spamming behaviors of colluders. The CHs of CBFs are shown in Figure 2.

Figure 2: Distributions of colluders (solid) and non-colluders (dashed) vs. CBFs.

Particularly, we found that 8 out of the 15 features (General Deviation, ED, "# of reviews", Group Deviation, GCS, GMCS, GS, GSUP) evaluated above showed unexpected characteristics compared to previous study results. To illustrate further, we randomly pick a colluder who is a member of 13 groups and make the following observation:

- (o1) Low GSUP: 12 out of the 13 (92.3%) have only 3 common products while the remaining one has 4;

[2]Our work only considers behavioral features because in [6] behavioral features are shown to overwhelmingly outperform non-behavioral features such as linguistic features that extracted from the text of reviews.

[3]TP is meaningless in our case since only a small portion of reviewers write multiple reviews for the same products.

- (o2) Low GS: 5 have size 2, 3 have size 3, 4 have size 4, and 1 has size 5;
- (o3) Low brand variations: all 13 groups target products of one particular brand;
- (o4) Similar ratings: among all 124 product-rating pairs, 21 (16.9%) are 4 stars, and 103 (83.1%) are 5 stars. There are no ratings lesser than 4 stars;
- (o5) Overlapping colluders: among all 78 group-group pairs, 69 (88.5%) have one common member, 8 have two, and one pair even has three common members.
- (o6) Overlapping products: among all 78 group-group pairs, 57 (73.1%) have at least one common product. Note that these groups have very low GSUP (see o1).

To better understand how these tiny colluder groups cooperate with each other, we use a bipartite graph to model the relationship among groups, colluders and products, an example of which is shown in Figure 3. In this example, all 4 groups share the common member "Z4". C_1 and C_2 both review products "1C" and "88" while C_3 and C_4 both review products "7Y" and "1C". Such a sophisticated and deceptive arrangement is designed to specifically evade state-of-the-art techniques as follows:

- Groups C_1 and C_2 (likewise C_3 and C_4) would have been merged into one bigger group if both groups write reviews for at least three common products. We regard at least two reviewers as a candidate group if they together review at least three products for the same reason as [6]. Now that they are single groups, GS will fail to distinguish them from normal ones;
- The GSUP of merged groups are also in the same scale as normal reviewer groups who may coincidentally review a few common products;
- From (o4) we know that tiny groups that target products of the same brand would provide consistently high/low ratings. If a less popular product is overwhelmed by many such groups, the majority rule abide by ED, General Deviation and Group Deviation would become invalid since the ratings given by the colluders would be very close to the average rating of the compromised products.

Figure 3: Collusive Groups: 4 groups (dashed boxes) collaboratively review 6 products (squares). Colluders are represented by circles.

3.2.3 Feature Selection

Finally, to improve the performance of colluder detection in our Chinese-Language review dataset, feature selection is employed as a sub-procedure. AUC (Area Under the ROC Curve) score is used as a metric to select well-performed features (Tables 1). a linear SVM model is trained to generate the AUC scores. The ranking of the features by AUC is roughly consistent with the situation illustrated in the corresponding cumulative histograms. Brand Deviation Score, Rating Deviation Score and GTW achieve the best with the AUC scores ≥ 0.80, followed by General Dev., GCS, GETF and GSR, of which the AUC scores are above 0.60. According to this ranking, 7 behavioral features (i.e., General Dev., BDS, RDS, GTW, GCS, GETF and GSR) are selected to form as the Selected Strong Features (SSF) due to their good performance in terms of both the AUC score and the cumulative histogram. Note that GS (AUC: 0.77) has been abandoned since its cumulative histogram shown in Figure 2 implies its vulnerability to collusive groups with small size. Note that although in practice features may not be used individually, our individual evaluation and selection can give a more fine-grained perspective of how the feasibility of *each* feature would be affected by the changes of the spamming patterns in real world data, guided by which the potential correlations among the features can be further assessed in practice.

(a) AUC scores of Individual Behavioral Features

Features	**General Dev.**	**TPG**	**ED**
AUC	0.61	0.49	0.59
Features	**Brand Dev.**	**Rating Dev.**	**# of reviews**
AUC	0.82	0.80	0.49

(b) AUC scores of Collusive Behavioral Features

Features	**GTW**	**GD**	**GCS**	**GMCS**
AUC	0.80	0.58	0.67	0.44
Features	**GETF**	**GSR**	**GS**	**GSUP**
AUC	0.68	0.67	0.77	0.43

Table 1: Area under ROC curves (AUC) for all 14 behavioral features (exclude TP). Larger AUC denotes better detection trade-off between true and false positives based on the corresponding feature.

3.3 The Hybrid Classification/Clustering Method

3.3.1 Motivation

The cause of the emergence of the phenomenon revealed by Figure 3 should be ascribed to one main factor which occurs when we sampling colluders in the dataset by using the FIM algorithm as shown in Section 3.2.1. In reality, a group of spammers may not want to write fake reviews for too many common products together[4], to form a complete sub-graph in a reviewer-product bipartite graph. Thus given a dense reviewer-product bipartite graph, by using the FIM algorithm with fixed parameter settings, the whole graph would be decomposed into many small pieces of fully connected sub-graphs (groups) that may have many nodes overlapped (members and products). Such small sub-graphs may dilute the effectiveness of some CBFs as discussed in Section 3.2.2.

Figure 3 also gives us a hint that a committed spamming campaign nowadays may involve a bunch of hired collaborative spammers. Their deceptive movements are in fact correlated, together guided by the same task specification of that spamming campaign. For example as shown in Figure 3, it is less likely that those 7 spammers happen to write reviews for those 6 products belong to the same brand in a lockstep

[4]In fact we found that colluders prone to write fake reviews for many different products belonging to the same brand.

way. In other words, the reviewer-product edges in the bipartite graph shown in Figure 3 are not placed at random, which from a spammer group level view implies certain degree of similarity between such correlated spammer groups (e.g., having overlapped members and products). On the other hand, in terms of regular reviewers, such highly overlapping (correlated) sub-bipartite graphs are less likely to be observed. Thus in order to achieve a more robust and accurate colluder detecting solution, we argue that when modeling their behaviors, such collusive review spammers should be treated *relationally*, rather than independently. While in our case, the colluders can be implicitly "connected" through their corresponding FIM generated groups.

Concretely, we propose to detect collusive spammers by adopting a two-step strategy, also following the classification setting. Firstly, each reviewer in the test set will be labeled as colluder or non-colluder by a base classifier trained from the training examples, with each represented as a vector of strong behavioral feature scores (Section 3.2.2) associated with a specific reviewer. Secondly, a metric based clustering algorithm is introduced to improve the prediction obtained from the classification algorithm by performing clustering on all reviewers in the test set. The metrics for each pair of reviewers are measured by taking advantage of the similarities (see Section 3.3.2) of their corresponding belonging groups. Intuitively, if the majority of a cluster is predicted to be colluder then the prediction for all reviewers in the cluster is colluder. Similarly if the majority of a cluster is predicted to be non-colluder then we relabel all reviewers in this cluster as non-colluder.

3.3.2 The Method

Basic Classifier. Given a set of reviewer groups $\{g_j\}_{j=1}^{m}$ and the set of individual reviewers $\{v_i\}_{i=1}^{n}$ with each associated with a feature vector $\mathbf{f}_i$, our goal is to assign each reviewer a label $y_i \in \{pos, neg\}$.[5] In the first step, a binary traditional classifier is employed to make initial prediction about the labels of all reviewers in the test set. Strong features selected in Section 3.2.2 are used to train such a base classifier.

Clustering. In the second step, a metric based clustering algorithm is used to smooth the classification results obtained in the first step. Fundamentally, a "collusiveness" metric is designed to capture the relations (more specifically, similarities) between reviewer pairs by exploiting their corresponding group level information. Note that the higher the degree of the similarity is, the closer the two reviewers would be, in the sense of collusion, and thus the lower the metric would be achieved. The metric is measured as following steps:

- **Common Member Ratio**. A colluder may work with different set of colluders on multiple assigned spamming tasks, causing groups sharing some common members. Intuitively, the more common members shared between group g and g', the higher the likelihood that their members collude with each other:

$$d_{\text{CMR}} = 1 - \frac{|g \bigcap g'|}{|g \bigcup g'|} \tag{1}$$

  where $|g|(|g'|)$ denotes the size of group $g(g')$.

- **Common Product Ratio**. The same set of target products (especially with the same brand) are often assigned to more than one colluder for reviewing so as to achieve the impression of reviewer variability, causing groups sharing some common products. Intuitively, the more common products shared between group g and g', the higher the likelihood that their members collude with each other:

$$d_{\text{CPR}} = 1 - \max_{b \in B} \frac{|P_{b,g}| + |P_{b,g'}|}{|P_g| + |P_g|} \tag{2}$$

  where B is the set of brands targeted by both group g and g', $P_{b,g}(P_{b,g'})$ is the set of products with brand b reviewed by $g(g')$, and $P_g(P'_g)$ is the set of products reviewed by $g(g')$.

- **Common Brand Rating Difference**. In order to abide by the task specifications of spamming campaigns, collusive groups always give consistent ratings (either promote or demote) to target products of a specific brand. Thus the difference between the average ratings given by group g and g' to that brand can be used to evaluate their rating consistency. Note that in a 5-star rating scale review system, 4-star is the largest possible difference:

$$d_{\text{CBRD}} = \min_{b \in B} \frac{|\overline{r}_{b,g} - \overline{r}_{b,g'}|}{4} \tag{3}$$

  where $\overline{r}_{b,g}(\overline{r}_{b,g})$ is the average rating given by group $g(g')$ to brand b.

- **Collusiveness**. The pairwise collusiveness of two groups is computed as the weighted average of the above measurements:

$$d_{g,g'} = \frac{\sum_i w_i d_i}{\sum_i w_i} \tag{4}$$

  where $d_i \in \{d_{CMR}, d_{CPR}, d_{CBRD}\}$ and w_i is the non-negative weight for d_i such that $\sum_i w_i = 1$. Finally, the pairwise collusiveness of two reviewers v and v' is computed by taking the average over the pairwise collusiveness of each pair of their respective groups:

$$d_{v,v'} = \frac{\sum_{g \in G} \sum_{g' \in G'} d_{g,g'}}{|G| \cdot |G'|} \tag{5}$$

  where $G(G')$ is the set of groups each of which has $v(v')$ as one of its members.

Our hybrid classification/clustering method is implemented as shown in Alg.1. We split the annotated dataset (Section 3.2.1) into training set T and test set T'. For clustering, a collusiveness matrix D is computed based on all reviewers in the test set T'. In Alg.1, a basic classifier $\mathbf{F}$ is trained on the training set T by using the selected strong behavioral feature set F (Line 1), and then is used to make initial prediction on the test set T' (Line 2). Then comes to the smoothing phase. First, based on the collusiveness matrix D, a number of non-overlapped clusters are generated by using a metric based clustering algorithm $\mathbf{K}$ (Line 3). Next, relabeling is performed (Line 4-7) as follows: for each cluster C_k, if the majority of this cluster is predicted to be positive then we relabel all reviewers in the cluster as positive, and vice versa.

[5]Reviewers labeled as colluders are positive examples.

$I(\cdot)$ is the identity function that takes value 1 if the input equality holds and 0 otherwise. Finally, the smoothed labels are returned as the output of the algorithm.

Alg. 1 : Colluder Detection Algorithm
Input : Selected strong behavioral feature set F; Training set: T; Test set: T'; Collusiveness matrix D; Clustering algorithm $\mathbf{K}$;
Output: Spam label of all reviewers in the test set.

1 Train a classifier $\mathbf{F}$ on training set T with feature set F;
2 For each v_i in T', predict the corresponding label as y'_i by using classifier $\mathbf{F}$;
// Smoothing
3 $C_1, C_2, ..., C_n = \mathbf{K}(D)$ where $C_i \cap C_j = \varnothing$ for any $1 < i < j < n$;
// Relabeling
4 **for** $k = 1$ *to* n **do**
5 **if** $\sum_{v_i \in C_k} I(y'_i = pos) \geq \sum_{v_i \in C_k} I(y'_i = neg)$ **then**
6 for any $v_i \in C_k$, $y_i \leftarrow$ pos;
7 **else** for any $v_i \in C_k$, $y_i \leftarrow$ neg;
8 **return** $\{y_i\}$;

3.4 Experimental Analysis

Experiment settings. Support Vector Machines (Linear) is selected as the basic classifier $\mathbf{F}$ to be consistent with previous studies [6, 7]. Ten-fold cross-validation is employed to create dataset splits for training and testing. The standard metrics - precision, recall, and f-score are used to measure the effectiveness of the classifier, wherein precision and recall are the ratio of predicted true colluders to the predicted reviewers and true colluders, respectively. DBSCAN, a density-based clustering algorithm [2], is applied to cluster reviewers. DBSCAN requires two parameters: the maximum distance ε between two neighboring samples and the minimum number m of samples required to form a cluster. We choose DBSCAN for two reasons: 1) we have no prior knowledge about the number of true clusters in our dataset; 2) DBSCAN treats the samples that belong to neither of the clusters as noise. This is applicable to our problem since non-colluders are expected to be dissimilar to colluders in terms of the collusiveness metric defined in Section 3.3.2. Hence they are expected to be treated as noise, further ignored by the smoothing phase of Alg.1. Homogeneity, which evaluates the purity of each cluster [8], is used to evaluate the performance of DBSCAN. We use homogeneity because the performance of Alg.1 (Line 4-7) relies heavily on the purity of each cluster.

Evaluation of our method. We first evaluate the basic classifier by using different feature sets, as shown in Table 2. It can be seen that IBF performs poorly, which is not surprising since individual based features cannot capture collaborative behaviors of colluders. By performing feature selection (using SSF), the F-score has been improved remarkably by 8.0% compared to CBF because we have removed all inferior features from IBF and CBF.

Next, based on the results obtained from the classification phase, we then conduct the clustering phase to see whether this smoothing technique would bring some benefits by exploiting the similarities between potential collusive spammers. In this experiment, we smooth the results obtained from the best performed SSF classifier. The final results are shown in Table 3.

	Precision	Recall	F-score
IBF	0.650	0.407	0.499
CBF	0.801	0.735	0.766
SSF	0.845	0.808	0.827

Table 2: Precision, recall and f-score with different feature settings: IBF - 6 Individual Behavioral Features, CBF - 8 Collusive Behavioral Features, SSF - 7 Selected Strong Features.

We can see the the proposed hybrid classification/clustering method achieves a promising improvement compared to the basic classifiers. Recall is notably improved compared to the basic classifier with SSF since some colluders labeled as negative by the basic classifier are re-labeled as positive in the clustering phase (Line 4-7 of Alg.1); mis-classified colluders can further be corrected if they are clustered together with most of correctly identified colluders. In general, as long as most of reviewers in a cluster are correctly classified and the cluster is pure enough, any wrongly labeled examples will finally be revised.

Sensitivity Analysis. To evaluate the sensitivity of choosing parameter settings for our DBSCAN algorithm, we vary the cluster parameter $\varepsilon \in [0.1, 0.9]$ at an interval of 0.05 and $m \in [2, 250]$ at an interval of 5. Figures 4(a) and 4(b) show the clustering performance over the parameter ranges. It can be observed that homogeneity becomes instability when $\varepsilon > 0.7$ (Figure 4(a)). This is because a large ε would allow many colluders and non-colluders to merge into one cluster. However, when $\varepsilon \leq 0.45$, the noise ratio is higher (Figure 4(b)) because a low ε would cause many examples to have insufficient peers within ε range, which are then treated as noise by the clustering algorithm. Note that a high noise ratio will not lead to improvement even if the corresponding homogeneity is high, as the predicted label of a reviewer will be ignored if the reviewer happens to be treated as noise. In addition, the value of parameter m generally has little impact on the homogeneity of resulting clusters.

Next we evaluate how clustering will affect the final detection performance. We use ΔF-score, the variation of the F-score caused by the clustering based smoothing, to quantify the improvement. The relationships among homogeneity, noise ratio and ΔF-score are plotted in Figures 4(c-e). In Figure 4(e), when the homogeneity>0.8, the ΔF-score is restrict positive, implying the results of basic classifier have been improved. However, when the homogeneity attains lower than 0.5, the outcome becomes deteriorated. This is because impure clusters will cause the minorities to be labeled that same as the majorities, resulting in more mislabelings. From Figure 4(d) we can see that, as the noise ratio becomes greater than 0.2, the ΔF-score is shown to slowly decrease towards zero. This explains why the noise ratio measurement must be taken into account. Since noise will be ignored by DBSCAN when computing homogeneity, even a high homogeneity may not be able to directly result in improvement if the noise ratio is much too high. Note that when noise ratio<0.2, performance becomes worse. This is because low noise ratio (<0.2) corresponds to low homogene-

	SSF	SSF+Clustering											
		ε=0.3 m=40	ε=0.4 m=40	ε=0.4 m=60	ε=0.5 m=40	ε=0.5 m=60	ε=0.5 m=80	ε=0.6 m=40	ε=0.6 m=60	ε=0.6 m=80	ε=0.7 m=60	ε=0.7 m=80	ε=0.7 m=100
Precision	0.845	0.854	0.878	0.863	0.881	0.864	0.864	0.888	0.879	0.871	0.898	0.894	0.887
Recall	0.808	0.852	0.914	0.919	0.930	0.930	0.926	0.936	0.936	0.933	0.941	0.937	0.890
F-score	0.827	0.852	0.895	0.890	0.904	0.896	0.894	0.911	0.906	0.901	0.919	0.915	0.888

Table 3: Results of the proposed hybrid classification/clustering algorithm.

(a) (b) (c) (d) (e)

Figure 4: Parameter Sensitivity. The 3-D scatter plots in (a) and (b) show how homogeneity and noise ratio are affected by ε and m. The 2-D scatter plots in (c), (d), and (e) show the pairwise relationships among homogeneity, noise ratio, and ΔF-score.

ity (< -0.3) (Figure 4(c)), resulting in performance deterioration (negative ΔF-scores) (Figure 4(e)).

In summary, the basic classifier **F** will be improved by the smoothing technique if the clustering is performed with high homogeneity value and low noise ratio. This is not hard to achieve just by tuning the clustering algorithm by the grid-search of the best parameter setting(Figures 4(a-b)), while in our case, decent parameter settings can be found when $\varepsilon \in [0.3, 0.7]$ and $m \geq 40$ (Table 3).

4. FUTURE RESEARCH

In the future, we plan to extend our current colluder detection approach. More specifically, we are pursuing a new colluder detection system which is expected to achieve the following design goals:

- *Generalization*: 1) The system should work well on online review communities designed for users from different regions; how to design a comprehensive colluder detection system that takes all the observed spamming patterns into account is really challenging. 2) The system should handle review data from different domains, e.g., restaurant vs product reviews; as an opinion mining problem, models trained for one particular domain will not necessarily be applicable to other domains because opinions are expressed differently across domains. These goals require our colluder detection system to rely on the information/evidence as general as possible. For example, review text information may not be used when designing the system. On the other hand, behavior-related evidence (e.g., who writes reviews with who for what entities) can be more favorable because they can be easily collected in any online review community and they contain fundamental information which is sufficient to reveal the collusive tactics adopted by colluders. We will also conduct evaluations on more datasets collected from online review communities with different characteristics.

- *Robustness*: As shown in Section 3.2.2, while new approaches are constantly invented aiming at efficiently catching spammers, spammers also unceasingly make smarter changes to cover their trails for being undetected. This fact requires our system to be able to handle unseen spamming patterns and also the specific attacks against the system itself. To achieve these goals, we plan to exploit the underlying implicit connections between collusive spammers as briefly discussed in Section 3.3.1. These potential correlations are expected to be hard to fake in practice and thus are more robust than the engineered features constructed via temporal observations. More specifically, explicit "links" can be created between colluders once specific conditions are satisfied. Then their corresponding "spamicity" can be inferred collectively based on their interactions implied by such "links".

- *Flexibility*: The system should be capable to easily integrate previous spam-tackling approaches that have been proven to work well on specific spamming patterns (e.g., literatures reviewed in Section 2). Moreover, prior knowledge inferred from side information (e.g. review text, review time series) should also be able to be plugged into the system to improve the overall performance.

- *Scalability*: Our system should be scalable to large data with a low computational complexity. Also, due to the precious labeled spam review data, how to take advantage of the vast majority of unlabeled data is still an open problem. In such case, semi-supervised learning techniques may be of great help to achieve this goal.

5. REFERENCES

[1] C. Chen, K. Wu, V. Srinivasan, and X. Zhang. Battling the internet water army: Detection of hidden paid posters. *CoRR arXiv:1111.4297*, 2011.

[2] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 226–231, 1996.

[3] N. Jindal and B. Liu. Opinion spam and analysis. In *Proceedings of the International Conference on Web Search and Web Data Mining (WSDM)*, pages 219–230, 2008.

[4] F. Li, M. Huang, Y. Yang, and X. Zhu. Learning to identify review spam. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2488–2493, 2011.

[5] E. Lim, V. Nguyen, N. Jindal, B. Liu, and H. Lauw. Detecting product review spammers using rating

behaviors. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 939–948, 2010.

[6] A. Mukherjee, B. Liu, and N. Glance. Spotting fake reviewer groups in consumer reviews. In *Proceedings of the 21st International Conference on World Wide Web (WWW)*, 2012.

[7] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock. Finding deceptive opinion spam by any stretch of the imagination. In *ACL*, 2011.

[8] A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007.

[9] G. Wang, S. Xie, B. Liu, and P. S. Yu. Review graph based online store review spammer detection. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 1242–1247. IEEE, 2011.