



GSLDA: LDA-based group spamming detection in product reviews

Zhuo Wang¹ · Songmin Gu¹ · Xiaowei Xu²

Published online: 7 February 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

Online product reviews are becoming increasingly important due to their guidance function in people's purchase decisions. As being highly subjective, online reviews are subject to opinion spamming, i.e., fraudsters write fake reviews or give unfair ratings to promote or demote target products. Although there have been much efforts in this field, the problem is still left open due to the difficulties in gathering ground-truth data. As more and more people are using Internet in everyday life, group review spamming, which involves a group of fraudsters writing hype-reviews (promote) or defaming-reviews (demote) for one or more target products, becomes the main form of review spamming. In this paper, we propose a LDA-based computing framework, namely GSLDA, for group spamming detection in product review data. As a completely unsupervised approach, GSLDA works in two phases. It first adapts LDA (Latent Dirichlet Allocation) to the product review context in order to bound the closely related group spammers into a small-sized reviewer cluster, and then it extracts high suspicious reviewer groups from each LDA-clusters. Experiments on three real-world datasets show that GSLDA can detect high quality spammer groups, outperforming many state-of-the-art baselines in terms of accuracy.

Keywords Review spam · Group spamming · LDA · Opinion spamming

1 Introduction

Online reviews are becoming an increasingly important user-generated content since they provide testimonials of real people rather than merchants' own advertisements. Today as more and more businesses are completed through the Internet, positive review content and/or high rating scores can yield considerable financial gains, while negative reviews and/or low rating scores can cause monetary loss. As a result, there exist individuals and organizations who write fake reviews to praise their products (services), or to depreciate others'. As the commercialization of such illegal acts, fraudsters are organized to collaboratively write fake reviews to magnify the effect of review manipulation. Such group spamming activities are even more harmful than individual review spammers, since they can take full control of the sentiment of a product.

Since the seminal work by Jindal and Liu [7], there have been a large number of research papers addressing the fake review detection problem. The main difficulty of this problem lies in that, it is extremely difficult to get ground-truth data for model building and model testing. One can hardly tell which reviews are fake by merely reading the review content [7, 15]. This makes data-driven approaches, i.e. traditional machine learning approaches, less attractive. Therefore, model-driven approaches and graph-based approaches seem to be more promising [18]. On the other hand, fake review detection problem can also be viewed as an anomaly detection problem. As it is often hard to tell whether a review/reviewer is fake/genuine, ranking methods are frequently used, which sort the reviews/reviewers according to how likely they involve in committing fraud.

In recent years, there were increasingly research interests in group spamming detection. Mukherjee et al first introduced the group spammer detection problem and proposed an iterative algorithm GSRank [13]. GSRank first exploits frequent itemset mining (FIM) to generate candidate spammer groups, and then rank these candidate groups based on the relationships among reviewers, products and reviews. Wang et al [20, 21] seek graph-based methods which only use the topological structure of review

✉ Zhuo Wang
zhuowang@sylu.edu.cn

¹ Shenyang Ligong University, Shenyang, China

² University of Arkansas at Little Rock, Little Rock, AR 72204, USA

graph to spot group spammers. Wang et al. [21] proposed a divide and conquer based algorithm, namely GSBP, to detect *loose* spammer groups, i.e., each reviewers in a group does not necessarily review all the target products, instead of *tight* groups generated by FIM. To further improve the detection precision, [20] refined GSBP and proposed GSBC which models the reviewer collusiveness by both considering review time and rating scores, and spammer groups are modeled as bi-connected graphs. Some researchers proposed methods to detect collusive review spammers, that is, they only rank reviewers according to the suspiciousness of being collusive, rather than find the whole spammer groups [23]. Although these methods are effective in spotting certain kinds of group spammers, the overall detection precision is still unsatisfactory. As spamming strategies advance over time, it is infeasible to design a single method that can recall all kinds of group spammers with a high accuracy.

In this paper, we propose a simple yet effective computing framework, namely GSLDA, for group spam detection in review data. The method not only significantly improves the detection precision, but also is scalable to large datasets. We claim the following contributions:

- We first introduce LDA to perceive the group spamming behavior in product review data. We adapt the mature LDA model that works in a *document-topic-word* setting to the *product-cluster-reviewer* setting by transforming review burstiness and rating score homogeneity into reviewer occurrences in a product. As a result, it can generate reviewer clusters (*topics*, in analogy to LDA for document clustering), each having a set of reviewers that are closely related in reviewing activities. To the best of our knowledge, this is the first attempt to use LDA upon non-textual information in detecting review spammers.
- We seek methods to extract high quality review spammer groups from each GSLDA cluster. The top N reviewers with largest probabilities in each cluster are selected as the abnormal reviewers (in analogy to the *topic* words in LDA), and the SCAN algorithm is used to extract highly suspicious spammer groups from these abnormal reviewers. The detected groups are ordered by a set of diverse group spam indicators.
- We conduct extensive experiments on three real-world datasets which contain labeled or unlabeled data. Experimental results show that our proposed GSLDA outperforms many state-of-the-art baselines significantly.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 interprets the GSLDA method. Section 4 gives the experimental results. We conclude the paper in Section 5.

2 Related work

Review spam detection problem can be roughly categorized into two types: individual review spam detection [1, 7, 10–12, 15, 16, 19] and group review spam detection [2, 13, 20–23, 25]. For individual review spam detection, supervised or unsupervised machine learning based methods were widely adopted, heavily relying on review content-based or behavior-based review spam indicators. Nowadays, however, review spammers are often organized and work collaboratively to game the review systems. The co-review behavior contains more clues for identifying fraudulent reviewers, which sheds light on group spam-based approaches for review spam detection.

The first attempt to detect group spammers in product review data is by [13], which exploits FIM technique to mine candidate co-reviewing groups. These candidate groups are further processed using GSRank method which computes the spam degree of each candidate groups. [2, 23] also use FIM to mine candidate groups. Wang et al [20, 21] proposed GSBP and GSBC which only exploit the topological structure of a reviewer graph, a projection of a *reviewer-product* bipartite graph. Unlike GSRank, GSBP/GSBC does not consider review content related features. It is reported that review content not only is of little help in distinguishing between fake and genuine reviews, but also incurs additional computation cost [14]. Some researchers aim to detect review spammers by exploiting the collusive behaviors, e.g., review time bursts [6], homogeneous review patterns [22]. Since review data inherently forms a bipartite network, many graph-based (or relationship-based) methods were proposed to evaluate the spamicity of individual reviews/reviewers or review spammer groups. Akoglu et al. [1, 16] proposed Markov Random Field (MRF)-based frameworks (FraudEagle and SpEagle) to rank the spamicity of review/reviewers, and group spammers can be obtained by doing graph clustering on an induced subgraph that contains only the top ranked review spammers.

Former researchers also have introduced LDA to the fake review detection problem [8, 9, 17]. However, their methods apply LDA directly upon review text. Li et al. [9] proposed a LDA-based model for detecting deceptive reviews, which applies LDA on review content by comparing the subtle differences between deceptive reviews and truthful ones. Nonetheless, their method is supervised, i.e., applying LDA on fake review content (crowd-sourced using Amazon Mechanical Turk, AMT) and genuine review content, respectively. On the other hand, in our method we use LDA to analogize the *reviewer-cluster-product* context. Besides, our method is completely unsupervised. Similarly, [8] proposed a topic model based approach to detecting fake reviews, where five types of topics are identified to be

correlated with truthful and fake reviews. Sandulescu and Ester [17] also uses LDA to generate review text topics and exploits the similarity of topic distributions to detect one-time review spammers.

3 GSLDA for group spamming detection

3.1 Characterizing group spamming

As more and more people engage in online reviewing, individual spamming usually has little impact on the sentiment of a product or service. Group spamming is referred to as a group of spammers working together to commit fraud for certain purposes. Mukherjee et al. [13] shows that group spamming has more spam clues than individual spamming hence is easier to be identified. Spammers are often required to complete the spam campaign within a given time period. Also, in product review systems, group spammers are enforced to rate extremity scores (eg. {4,5} for promoting, or {1,2} for bad-mouthing) for target products. Although sometimes spam campaign organizers may require spammers to write *normal* reviews to camouflage, such efforts are usually far less made than on real spam. In comparison, individual spamming has far less clues for identifying spam, especially for the one-time reviews, where almost only the review content can be used [17].

Review time and rating scores, as well as the reviewer-product network are hard to fabricate, thus are more reliable in detecting group spammers. Review content, on the other hand, is highly subjective, and well-crafted fake review text is difficult to be identified even by human judges [15]. Therefore, in this study, we do not use review content information.

3.2 The LDA model

For a better understanding of GSLDA, we briefly introduce the principle of LDA, a generative topic model which was first proposed by David Blei et al in 2003 [3]. In machine learning and natural language processing, a topic model is a type of statistical model for discovering the hidden topics that occur in a collection of documents. Each document can be viewed as a mixture of various topics, and each topic is revealed by a set of frequently used words. In LDA, each document is akin to the standard “bag of words model” assumption, that is, word sequence is not considered. Therefore, LDA can also be used in other domains where data can be represented as “bag of words model”. For instance, by treating an image as a document, and small patches of the image as words, LDA can be used to automatically put natural images into categories.

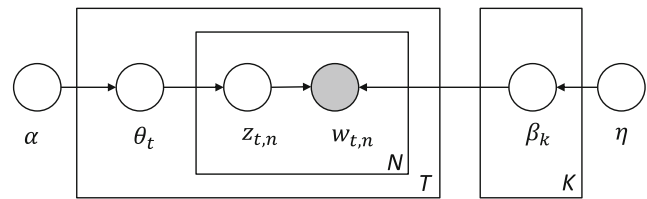


Fig. 1 Plate notation for LDA with Dirichlet-distributed topic-word distributions

Figure 1 illustrates the plate notation of LDA, which is widely used in probabilistic graphical model representation. Let T be the number of documents, K be a specified number of topics, and N is the vocabulary size. All documents are represented by $T N$ -dimensional vectors $w_t, t = 1, 2, \dots, T$, all topics are represented by $K N$ -dimensional vectors $\beta_k, k = 1, 2, \dots, K$. $w_{t,n}$ denotes the word frequency of the n -th word in the t -th document, which are the only observable variables. $\beta_{k,n}$ denotes the word frequency of the n -th word in the k -th topic. The probability distribution that each topic belongs to the t -th document is denoted by θ_t , thus $\theta_{t,k}$ is the probability that the k -th topic belongs to the t -th document. The hidden topic structure of a document collection is represented in the hidden random variables: the per-topic word proportions β_k , the per-document topic proportions θ_t , and the per-word topic assignments $z_{t,n}$. LDA assumes the following variable distributions:

$$\theta_t \sim \text{Dirichlet}(\alpha)$$

$$\beta_k \sim \text{Dirichlet}(\eta)$$

$$z_{t,n} \sim \text{Multinomial}(\theta_t)$$

$$w_{t,n} \sim \text{Multinomial}(\beta_{z_{t,n}})$$

Given the LDA model, words in a document are produced as a generative process:

1. For each topic, draw a distribution over words $\beta_k \sim \text{Dirichlet}(\eta)$;
2. For each document,
 - (a) draw a vector of topic proportions $\theta_t \sim \text{Dirichlet}(\alpha)$;
 - (b) For each word,
 - i. Draw a topic assignment $z_{t,n} \sim \text{Multinomial}(\theta_t), z_{t,n} \in \{1, \dots, K\}$;
 - ii. Draw a word $w_{t,n} \sim \text{Multinomial}(\beta_{z_{t,n}}), w_{t,n} \in \{1, \dots, N\}$.

The inference problem of LDA is to compute the topic distribution of each document θ_t , the word distribution of each topic β_k , and the topic assignment of each word in each document $z_{t,n}$, given the observed data $\mathbf{W} = \{w_t\}$ and the Dirichlet prior parameters α and η , i.e. $p(z, \beta, \theta | \mathbf{W}, \alpha, \eta)$. Approximation techniques, including variational inference and Gibbs sampling, are often used to

compute the hidden variables. Note that β_k contains the per-topic word distribution, which is the main concern of our proposed GSLDA (per-cluster reviewer distribution).

3.3 GSLDA reviewer clustering

In product review data, reviewers and products naturally form a *product-reviewer* bipartite network [1]. A bipartite network is a graph whose vertices can be divided into two disjoint and independent sets U and V such that every edge connects a vertex in U to one in V. Normal reviewers usually review or rate separately thus genuine reviews usually arrive randomly. In group spamming, however, review spammers inevitably form a tightly coupled sub-bipartite graph of the global bipartite network. Reviewers in such spam groups can be viewed as outliers of the vast majority of the reviewers. Therefore, in detecting group spamming, the goal is to search all the spammer groups that the members in each group are closed related, i.e., to review a set of target products in a close time window, rate similar scores, and, to go further, have homogenous review behaviors or preferences.

In LDA topic model, documents and words also form a *document-word* bipartite network. Motivated by LDA, we treat products as documents, and the reviewers who review the same product can be seen as a “bag of reviewers”. We then model reviewer groups as hidden, which are analogous to *topics* in LDA. To distinguish, we call these *topics clusters*. By treating spam groups as hidden topics, we can gracefully adapt LDA to the *product-cluster-reviewer* setting. Unfortunately, this straightforward analogy of LDA has many drawbacks. (1) In LDA, the words in a document is viewed as the *bag of words* model, that is, it infers the distributions only by the word frequency in a document. In review data, however, reviewers post reviews in chronological sequence. Review time burst, i.e., a time interval during which many reviewers review a product simultaneously, is often a crucial signal for review spamming; (2) A word may appear in a document for many times, while in review data, a reviewer usually reviews a product only once. To leverage LDA for group spam detection, we artificially increase the number of occurrences of a reviewer who is likely to be group spammers. The intuition is to boost the frequency of suspicious reviewers based on the two key group spamming signals: review time burstiness and rating score similarity.

3.3.1 Review time boosting

To increase the frequency of a reviewer who reviews in a review burst, we introduce kernel density estimation with Gaussian kernel. Kernel density function, which estimates the probability density function of a random variable, can well reflect the review burstiness, i.e., the magnitude of

review burstiness on date d is proportional to the value of kernel density function in d . The kernel density function is defined as follow:

$$\hat{f}_h(x) = \frac{1}{n} \sum_{i=1}^n K_h(x - x_i) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right) \quad (1)$$

where $K(\cdot)$ is the kernel function, and h is the band width. We use Gaussian kernel, thus

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

For a given product p , let D_p be the review date set of product p , the reviewer who reviews on date $d \in D_p$ is boosted to $N_t(p, d)$ times, where:

$$N_t(p, d) = \frac{\hat{f}_h(d)}{\min_{d' \in D_p} \hat{f}_h(d')} - 1 \quad (2)$$

$N_t(p, d)$ is defined as the ratio of the density of the review on date d to the minimum density of all review dates of p , and minus one. For most reviews that do not lie in review bursts, their $N_t(p, d)$ will be 0. Therefore, such reviewers will be filtered out, which can greatly reduce the input size of LDA. This mechanism also facilitates multiple reviewing, i.e., a single user reviews a product for many times, which is often thought of as a suspicious spam signal. h is chosen to determine how close that two review dates are considered to be collusive.

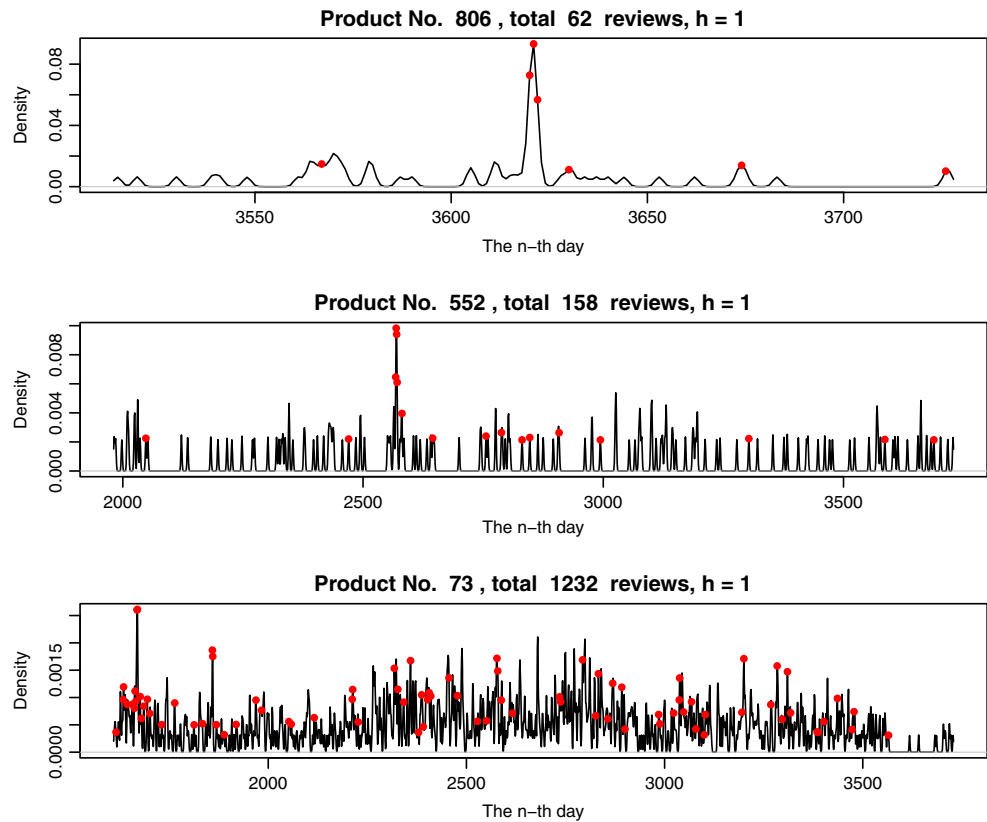
Figure 2 plots the kernel density of three products with various numbers of reviews in YelpNYC dataset. Each red point indicates a group spamming activity on that date, according to the labels of the dataset. We can see that group spam is more likely to happen in review bursts.

Note that kernel density estimation was also used in [6] for detecting review burstiness. They take the derivative of the kernel density function, and set it to zero to find the time when a review peak is formed. This is notably different from our method in that, we use kernel density function to magnify the frequency of a reviewer who is suspicious for group spamming.

3.3.2 Rating score boosting

Another way to boost the frequencies of review spammers is by taking into account the rating scores. As we have mentioned earlier, group spammers are more likely to give similar rating scores toward a target product than those in randomly formed groups. We model this behavior by considering both the variance and the entropy of the rating scores lie in a relatively narrow time window τ , where a review burst is probably likely to happen. Intuitively, the

Fig. 2 The kernel density of three products with various number of reviews in YelpNYC dataset. Each red point indicates a spam activity on that date. The bank width h is set to 1



smaller the variance and the entropy of the scores, the more suspicious those reviewers will be. Thereby, the times of the occurrences of a reviewer who review product p on date d is boosted to

$$N_r(p, d) = \left(2 - \frac{\text{var}(\Psi_\tau^d)}{8} - \text{entropy}(\Psi_\tau^d)\right) \text{avg}_{d' \in D_p} N_t(p, d') \quad (3)$$

where Ψ_τ^d is the rating score set in which each score is rated in $[d - \frac{\tau}{2}, d + \frac{\tau}{2}]$. The variance is divided by 8 because 8 is the largest variance in a 5-star rating system, which is the variance of $\{1, 5\}$. The rating score boosting is multiplied by the average value of review time boosting for all dates in D_p , so that the impact of rating score boosting is in a comparable order of magnitude with that of the review time boosting.

Put it altogether, the final boosting times for a reviewer who review on date d toward product p is defined as

$$N(p, d) = \text{Round}(N_t(p, d)) + \text{Round}(N_r(p, d)) \quad (4)$$

As a result, each product now contains new reviewer bags with magnified number of reviewers. The new data are fed into LDA and accordingly we get K reviewer clusters, each having its own reviewer distribution, i.e., the probability of a reviewer to be in that cluster. We call such a reviewer cluster a GSLDA cluster. According to the rationale of LDA, the top N reviewers are the “keywords” that best describe

that cluster, which contains abnormal reviewers for further discovery of group spamming.

3.4 Spammer group extraction

In LDA, words with the largest N probabilities in a topic can be thought of as the representatives of that topic. Although the top N reviewers in each GSLDA cluster are highly suspicious, simply output these reviewers as spammers will lead to poor detection precision. This is because that a *cluster* in GSLDA has different semantics from a *topic* in LDA. Reviewers in the same cluster implies that they have directly or indirectly reviewed the same products. In fact, reviewers with a large number of reviews toward a large number of products intend to be the representative reviewers in a GSLDA cluster. Such reviewers are usually long-term users and are not likely to be spammers. In GSLDA, since we have boosted the number of reviewers based on the review time burstiness and rating score similarity, members in the same group spam campaign are even more likely to be in the same cluster. As such, detecting group spammers from the entire dataset is transformed into detecting group spammers within each GSLDA cluster.

In this subsection, we first give a model for describing the relationship between two reviewers in a GSLDA cluster, then we use this relationship to construct a reviewer graph that reveals the collusiveness between each pair of

reviewers. This graph is further processed using SCAN [24], a structural clustering algorithm for networks, to generate the candidate spammer groups. Finally, we define 7 group spamming indicators based on which the spamicity of each candidate group is computed.

To extract spammer groups from a GSLDA cluster is similar to that from the full bipartite graph projection, as done in [20, 21]. However, there are indeed some differences between the two data sources. (1) Since we only choose a small fraction of the total reviewers in the cluster, the number of related reviewers and products are significantly reduced compared to the number of reviewers and products in the full datasets. (2) Through LDA, reviewers in each GSLDA cluster become closely related, the induced reviewer graph becomes extremely dense. As such, Algorithms in [20, 21] become less effective in processing GSLDA clusters. Thus we need a new method to extract high quality review spammer groups in a GSLDA cluster.

3.4.1 Choosing abnormal reviewers from a GSLDA cluster

Simply choosing the top N reviewers in each GSLDA cluster could be problematic since there might be less than N reviewers whose probabilities belonging to a cluster are non-zeros. Therefore, we first choose the outliers (abnormal reviewers) of the reviewer probability distribution, then if the number of outliers is greater than N , we choose the top N reviewers, otherwise, only the outliers are chosen. The standard definition of outliers for a Box-and-Whisker plot is the points outside of the range $\{Q_1 - 1.5IQR, Q_3 + 1.5IQR\}$, where $IQR = Q_3 - Q_1$, and Q_1 is the first quartile and Q_3 is the third quartile of the data.

3.4.2 Reviewer graph construction

GSBP, which models relationship between two reviewers as the number of co-reviewed products, is shown to be inferior to GSBC, which models the relationship between two reviewers by both considering review time and rating score deviation. Therefore, we use a similar way as used in GSBC to model the relationship between two reviewers, which defines the co-review collusiveness between two reviewers i and j toward product k as

$$Collu(i, j, k) = \begin{cases} 0, & |t_i^k - t_j^k| > \tau \vee |\psi_i^k - \psi_j^k| \geq 2 \\ \alpha(1 - \frac{|t_i^k - t_j^k|}{\tau}) + (1 - \alpha) \left(1 - \frac{|\psi_i^k - \psi_j^k|}{2}\right), & otherwise \end{cases} \tag{5}$$

where τ is a user specified time window, t_i^k stands for the review time of reviewer i toward product k , ψ_i^k stands for the rating score of reviewer i toward product k , α is a coefficient to balance the importance of review time and rating score.

Equation (5) implies that if reviewer i and j co-review product k beyond time window τ or their rating score deviation is greater than 1, the co-review will not be considered as spamming, which significantly reduces the number of co-reviews by coincidence. Unlike GSBC, we no longer consider the suspicion of a product, since we only pick a small fraction of reviewers in a cluster, so that only a small number of products are involved. After all, the LDA process already filters out irrelevant products.

Then we define the collusiveness between reviewer i and j as the maximum collusiveness between reviewer i and j toward the commonly reviewed products of reviewer i and j , and multiplies the Jaccard Similarity of the product set of reviewer i and j :

$$\sigma(i, j) = \frac{|P_i \cap P_j|}{|P_i \cup P_j|} \max_{k \in P_i \cap P_j} Collu(i, j, k) \tag{6}$$

where P_i and P_j are the sets of products reviewed by i and j , respectively. In Eq. 6, we introduce Jaccard similarity coefficient to reflect the fact that collusion is more likely to happen when most of the products reviewed by the two reviewers are in common, rather than by coincidence, which is crucial to gain a high detection accuracy.

Small edge weights are trivial thus we prune them by setting an edge weight threshold δ . So that we can obtain a reviewer graph of a GSLDA cluster by taking the nodes as the reviewers and the weights between two nodes i and j as $\sigma(i, j) \geq \delta$.

3.4.3 Generate candidate groups using SCAN

With the reviewer graph of a GSLDA cluster as described above, the goal is to mine dense regions from the graph that represent the spammer groups. GSBC exploits bi-connected components to model review spammer groups, and uses a divide-and-conquer strategy to search spammer groups in the global reviewer graph. A bi-connected graph is a connected graph that, for each pair of node i and j , there exist at least two disjoint paths between node i and j . Since our reviewer graph consists of only the top N reviewers in a GSLDA cluster, GSBC algorithm does not work well on it. Instead, we exploit graph clustering algorithm SCAN to mine the dense parts of a given weighted graph. These dense parts of the graph are called candidate review spammer groups. Note that the final number of groups is impacted by parameter ϵ , a structural similarity threshold defined in SCAN. The larger the ϵ value, the less number of clusters SCAN will generate, and the denser the clusters will be.

3.4.4 Spammer group ranking

In order to quantitatively determine the spamicity of the groups detected by SCAN, we use group spam indicators

to rank the detected groups. Note that these indicators might be domain knowledge related. It is worth noting that these indicators only affect the final ranking list of the detected groups by SCAN, but not the groups themselves. In comparison, in GSBP/GSBC, group spam indicators not only determine the final generated groups, but also the ranking results as well. Intuitively and wisely selecting a subset of the group spam indicators will generate user-intended ranking results. For comparison, we use the same group spam indicators that were used in [20]. The *MR* (multiple review) indicator in [20] is not used in this study, because our datasets do not involve multiple reviewing, i.e., a single reviewer reviews a product for many times. For completion, we also briefly list the seven group spam indicators here. All these indicators are normalized to [0,1], and a larger value indicates being more suspicious. For detailed information, please refer to [20]. Note that, unlike individual review centric or reviewer centric features [5], these group spammer features can only be computed after a reviewer group has been generated using a group spam detection algorithm, e.g., FIM [13], GSBP or GSBC.

To reduce the contingency of small-sized spammer groups, a penalty function is introduced using a variant of the Sigmoid function as follow:

$$L(g) = \frac{1}{1 + e^{-(|R_g|+|P_g|-3)}} \quad (7)$$

where R_g denotes the reviewer set of g , P_g denotes the target product set of g . A target product is referred to as a product which is co-reviewed by two reviewers in R_g within time window τ , and the rating score difference is less than 2. In a spammer group, the minimum number of reviewers is 2 and the minimum number of products is 1, so that $L(g) \in [0.5, 1)$.

1. *Review tightness (RT)*: In tightly coupled spammer groups, group members intend to write reviews for every target product. Given a spammer group g , we define the review tightness of g , denoted by $RT(g)$, as the ratio of the number of suspicious reviews in g to the cardinality of the cartesian product of R_g and P_g , and multiplies $L(g)$. For suspicious reviews, which form a review set V_g , we mean those reviews that are co-reviewed by two reviewers in time window τ and their rating score difference is less than or equal to 1.

$$RT(g) = \frac{|V_g|}{|R_g||P_g|} L(g) \quad (8)$$

2. *Neighbor tightness (NT)*: In tightly coupled groups, the collusiveness among reviewers tends to be stronger than those in genuine reviewer groups. Thus the neighbor tightness of group g is defined as the average edge

weight of the review spammer graph induced by group g , and multiplied by $L(g)$:

$$NT(g) = \frac{\sum_{i,j \in R_g} \sigma(i,j)}{\binom{|R_g|}{2}} L(g) \quad (9)$$

3. *Product tightness (PT)*: Studies show that spammers are usually not long term users. If a reviewer only writes reviews for the target products and does not review any other products, they are very likely to be spammers. Thus we define the product tightness of g as the ratio of the number of commonly reviewed products by all the members in g to the number of products reviewed by all the members in g :

$$PT(g) = \frac{|\cap_{r \in R_g} P_r|}{|\cup_{r \in R_g} P_r|} \quad (10)$$

where P_r denotes the product set of reviewer r .

4. *Average time window (TW)*: In a spammer group, reviewers often post fake reviews in a narrow time window. Thus we define the time window spamicity of product p as:

$$TW_p(g, p) = \begin{cases} 1 - \frac{SD_p}{T}, & SD_p \leq T \\ 0, & SD_p > T \end{cases}$$

where SD_p is the standard deviation of review dates for product p reviewed by reviewers in R_g , T is a user specified time threshold, say, 10 days. The TW indicator of g is then defined as the average time window spamicity of all products in P_g , and multiplied by $L(g)$:

$$TW(g) = \text{avg}_{p \in P_g} TW_p(g, p) L(g) \quad (11)$$

5. *Rating variance (RV)*: Group spammers tend to rate identical or similar scores. Thus we define the *RV* indicator as:

$$RV(g) = 2 \left(1 - \frac{1}{1 + e^{-\text{avg}_{p \in P_g} \text{var}(p,g)}} \right) L(g) \quad (12)$$

where $\text{var}(p, g)$ denotes the variance of the rating scores of product p by reviewers in g . We use sigmoid function to smooth the average variance to interval (0,1).

6. *Reviewer ratio (RR)*: If the target products in P_g are mainly reviewed by the reviewers in R_g , and reviewers not in R_g are rare, then the group is more suspicious. We define *RR* as the maximum ratio of the number of reviewers in R_g who review product p to the number of all the reviewers of p , $p \in P_g$:

$$RR(g) = \max_{p \in P_g} \frac{|R_{gp}|}{|R_p|} \quad (13)$$

where R_p stands for the reviewer set of product p , and R_{gp} is the reviewer set of product p and these reviewers must be in g .

Table 1 Review dataset statistics

Dataset	#Reviews	#Reviewers	#Products	Time span	Labeled
YelpNYC	359,052	160,225	923	2004.10-2015.1	Yes
YelpZip	608,598	260,277	5,044	2004.10-2015.1	Yes
AmazonCDs	972,105	536,264	118,122	2012.1-2014.7	No

7. *Group size (GS)*: Groups that only contain 2 or 3 reviewers are more likely to be formed by accident. Groups involve more people and products are more suspicious and interesting than small groups. We define *GS* indicator as

$$GS(g) = \frac{1}{1 + e^{-(|R_g|-3)}} \tag{14}$$

Since the minimum value of $|R_g|$ is 2, $GS \in [0.27, 1)$.

3.5 The GSLDA computing framework

In a nutshell, Algorithm 1 describes the GSLDA computing framework for group spam detection. The algorithm is self-explanatory.

Algorithm 1 GSLDA computing framework

Input:

- 1: \mathcal{B} : bipartite review graph (product-reviewer);
- 2: h : band width in kernel density estimator;
- 3: τ : co-review time window;
- 4: K : number of clusters to be generated;
- 5: α : review time and rating score balance coefficient;
- 6: N : maximum number of reviewers selected from a GSLDA cluster;
- 7: δ : reviewer graph edge weight threshold;
- 8: ϵ : structural similarity threshold defined in SCAN;

Output:

- 9: ranking list of spammer groups

Description:

- 10:
 - 11: **for** each product p in \mathcal{B} **do**
 - 12: Boost reviewer occurrences using (4);
 - 13: **end for**
 - 14: Apply LDA to generate K GSLDA clusters;
 - 15: **for** each GSLDA Cluster gc **do**
 - 16: Construct reviewer graph G using the top N abnormal reviewers in gc as vertices, using (6) to compute edge weights ($\geq \delta$);
 - 17: Apply SCAN on G to generate candidate groups;
 - 18: **for** each candidate group cg **do**
 - 19: Compute the spamicity of cg using the average of the 7 group spam indicators;
 - 20: **end for**
 - 21: **end for**
-

For Line 1-3 of Algorithm 1, the time complexity is $O(|V|)$, where $|V|$ is the number of reviews in the dataset. For the most tough part, LDA in Line 4, the time complexity is $O(k|P||R|)$, where k is the number of iterations, $|P|$ is the number of products, $|R|$ is the number of reviewers, according to the implementation of LDA using Gibbs sampling. The final part, through Line 5-11, which only processes a small fraction of each GSLDA cluster, is very fast. Thus, the overall time is linear to the number of products and reviewers, which ensures that the approach is scalable to large review datasets. The approach is also space efficient which has a space complexity of $O(K(|P| + |R|))$, where K is the number of topics in LDA.

4 Experimental study

4.1 Datasets

We conducted extensive experiments on two labeled datasets (YelpNYC and YelpZip) and an unlabeled dataset (AmazonCDs) which were also used in [20]. Table 1 shows the dataset statistics. YelpNYC and YelpZip were first used in [16] and contain review data in restaurants and hotels from 2004 to 2015, and each review tuple (review time, reviewerID, productID, rating score) is labeled as recommended (genuine) or filtered (fake) by Yelp anti-fraud filtering algorithm. Since Yelp review’s filter is no way perfect, the dataset is called near ground-truth. AmazonCDs contains Amazon CDs and Vinyl reviews from 2012 to 2014, and is unlabeled. Unlike Yelp datasets, AmazonCDs involves a large number of products.

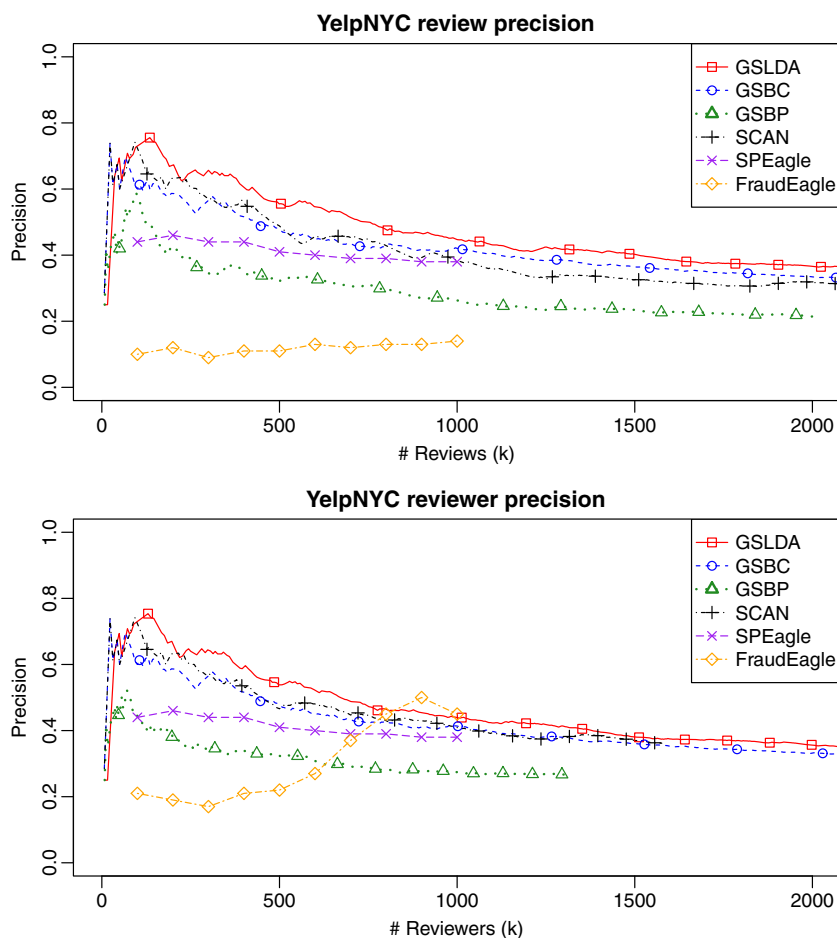
4.2 Compared baselines

First, we compare GSLDA with GSBP and GSBC. Since we use SCAN to find spammer groups from the reviewer graph of each GSLDA cluster, we also compare to SCAN

Table 2 Parameter setting for datasets

Dataset	τ	h	K	α	N	ϵ	δ
YelpNYC	10	1	300	0.5	100	0.84	0.6
YelpZip	20	1	300	0.5	100	0.84	0.6
AmazonCDs	10	1	300	0.5	100	0.84	0.6

Fig. 3 Review/reviewer precision @ top k on YelpNYC dataset. The curves for GSLDA, GSBC, GSBP, SCAN are marked with a point every 40 groups. Curves for FraudEagle and SpEagle are marked every 100 reviews/reviewers



that finds spammer groups from the global reviewer graph that constructed from the entire review bipartite graph. We also compare with FraudEagle and SpEagle, two Markov Random Field based approaches, which also try to detect (group) review spammers in Yelp datasets.

4.3 Performance on Yelp datasets

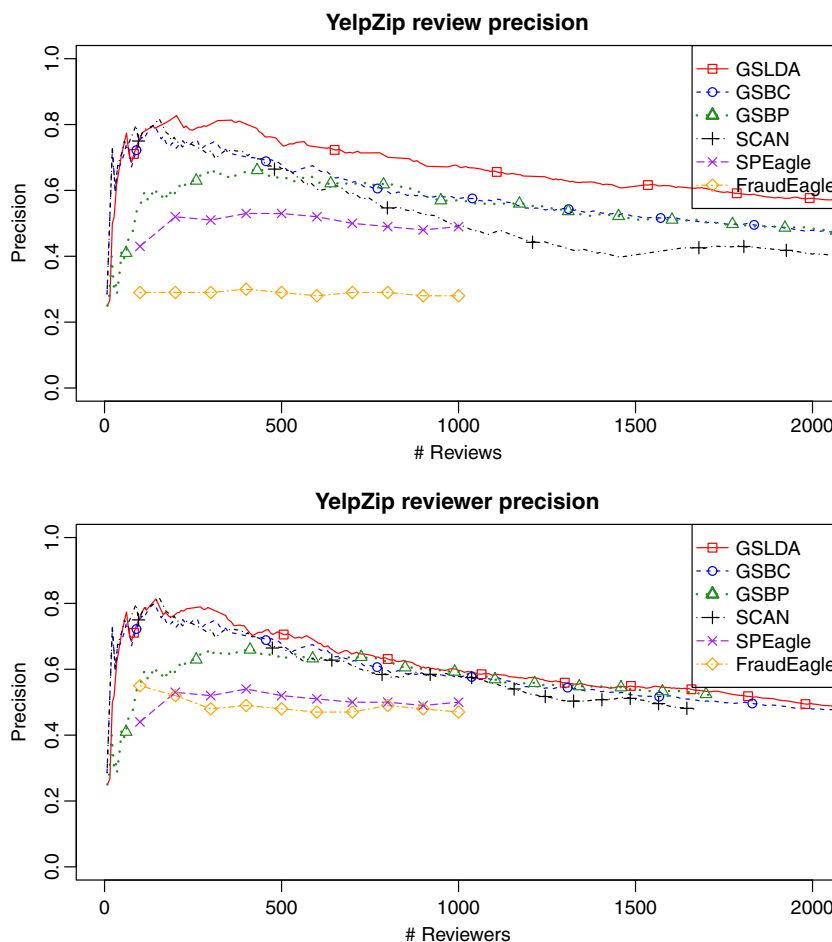
For the labeled Yelp datasets, we compared the detection precision at top k reviews/reviewers of each method. The parameters used in GSLDA for YelpNYC and Yelp Zip are shown in Table 2. The Dirichlet hyperparameters for topic proportions (α) and topic multinomials (η) are both set to 0.05. Note that we set a larger time window for YelpZip than YelpNYC because we found that the review density of YelpNYC is larger than that of YelpZip. We plot the precision of the top 2000 reviews/reviewers detected by GSLDA, GSBC, GSBP, and SCAN. For FraudEagle and SpEagle, we use the experimental data in [16], where the top 1000 review/reviewer precision was provided. For review ranking of GSLDA, GSBC, GSBP, and SCAN, we fetch all the reviews in each group, and rank all these reviews according to the ranking of the groups returned by

each method. The reviews in the same group are randomly placed. We mark a reviewer as a spammer if and only if the reviewer has written at least one fake review, as defined in [16].

Figure 3 shows the precision for YelpNYC dataset. We can see that GSLDA outperforms all the baselines with a large margin, especially for the top 1000 reviews/reviewers. The curves for GSLDA, GSBC, GSBP, and SCAN are marked with a point every 40 groups. The more points, the more groups are required to generate the top 2000 reviews/reviewers. For the fixed number of groups, larger group size leads to better detection recall. We can see that GSBC has the least number of points both for reviews and reviewers, indicating that GSBC intends to generate larger spammer groups than others for this dataset. GSLDA generates slightly smaller groups than GSBC. GSLDA also generates large groups for the top ranked groups, and the group size gets smaller as more groups are fetched. Figure 4 shows the precision for YelpZip dataset. We can see that GSLDA again performs the best among all the 6 approaches. This time it generates the largest number of reviews for the groups.

To study the impact of parameters used in GSLDA, we set different parameter values to run GSLDA on YelpNYC

Fig. 4 Review/reviewer precision @ top k on YelpZip dataset



dataset. We use the setting in Table 2 as a baseline, and each time we change the value of a parameter to compare the detection precision. Figure 5 shows the result. The red curve stands for the parameter value for the baseline. We can see that all the parameters except for δ (reviewer graph edge weight threshold) have not much impact to the precision. Parameter δ , which determines the reviewer graph structure, has significant impact to precision: too small δ results in low precision. $\delta=0$, i.e., no edge weight threshold is specified, yields poor detection precision, indicating that SCAN is sensitive to the structure of the graph.

Note that there is a trade-off between precision and recall. For example, if we specify a larger N or lower δ , more groups will be returned, thus the precision might drop, but the recall might increase.

4.4 Performance on AmazonCDs dataset

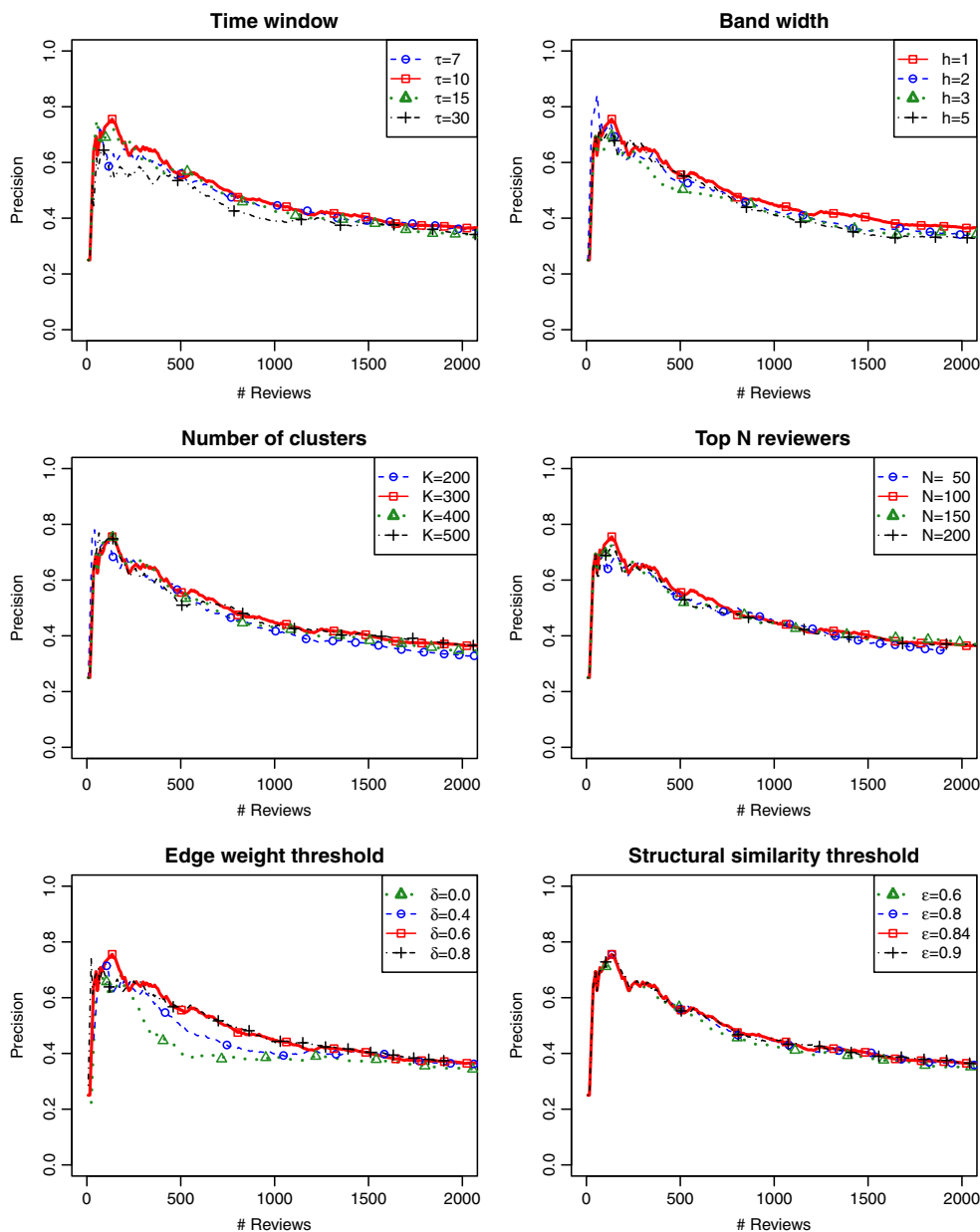
We run GSLDA on AmazonCDs dataset using the parameters shown in Table 2. Since AmazonCDs is an unlabeled dataset, we conduct a CDF (Cumulative Distribution Function) comparison among GSLDA, GSBC, GSBP, and SCAN. CDF comparison is frequently used

to evaluate the quality of detected items from a single dimension in fake review detection research [4, 13, 23]. The closer the curve is to the right, the higher value the indicator will achieve. From Fig. 6 we can see that, GSLDA performs quite well on all indicators except for RR, for which GSBP has an astonishing performance. On average, the AVG indicator has comparable performance with GSBC.

Since CDF can not completely determine the overall detection precision, to further study the performance of GSLDA on AmazonCDs, we perform a human evaluation of the spammer groups generated by GSLDA and GSBC. Labeling fake reviews is a challenging task which requires concrete evidence, business environment, user feedbacks, and enough man-power. Only commercial review sites (Yelp, Amazon, etc.) have labeling services [8]. Therefore, instead of labeling the detected groups or reviews as spam/non-spam, we give a brief summarization of the results of GSLDA and GSBC.

First, we compared the group similarity between GSBC and GSLDA for the top 500 groups. GSLDA groups contain 7602 reviewers and GSBC groups contain 4335 reviewers, and total 2745 reviewers are in common. This shows that GSLDA tends to find suspicious reviewers with higher

Fig. 5 Parameter impact for YelpNYC dataset. The red curve stands for the baseline using parameters in Table 2



recall. We also try to match the similar groups between GSLDA and GSBC using Jaccard similarity. Given two spammer groups g and g' , the Jaccard similarity between the two groups is defined as

$$JaccardSim(g, g') = \frac{|R_g \cap R_{g'}|}{|R_g \cup R_{g'}|}$$

For a given GSLDA group g_1 , we find a group g_2 in GSBC with the maximum Jaccard similarity with g_1 , and vice versa. Table 3 shows the Jaccard similarity distribution for GSLDA and GSBC, respectively. From the table we can see that GSLDA and GSBC generate a large portion of similar groups (170) and also a large portion of quite different

groups (278/242). This shows that they are complementary in spotting fake review groups. On the other hand, the common groups shall be more suspicious than those that are significantly different. Considering such groups can further improve the detection precision.

We also manually checked the top 100 spammer groups returned by GSLDA and GSBC, by both checking the AmazonCDs dataset and the Amazon online customer profiles, and found that all these 200 groups are to some extent suspicious for group spamming. The main characteristics of these groups are summarized as follows.

- Most of the reviewers review one or more target products in the same day or in a very narrow time window;

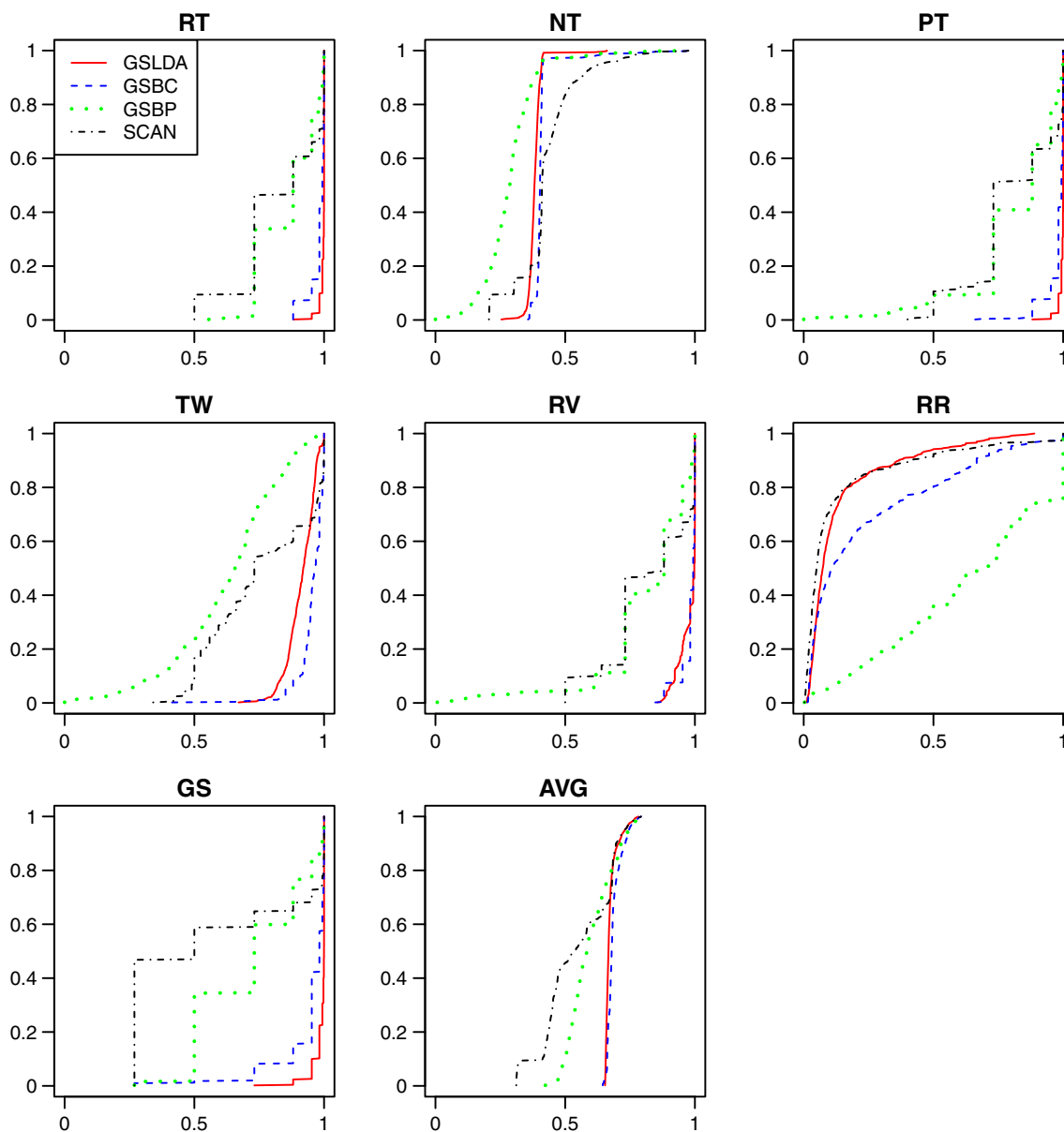


Fig. 6 CDF comparison for AmazonCDs dataset

- The rating scores are almost all 5 stars;
- The target products were almost only reviewed by the reviewers in the spammer group;
- Most of the group members were not active after they reviewed the target products.

Table 3 Jaccard similarity comparison between GSLDA and GSBC on AmazonCDs

Jaccard Similarity	[0,0.2]	(0.2,0.5]	(0.5,1]
GSLDA vs GSBC	278	52	170
GSBC vs GSLDA	242	88	170

5 Conclusion

Fake review detection problem has become increasingly important for online review systems. Nowadays group spamming is the main form of review spamming. In this paper, we proposed a group spamming detection approach based on the mature LDA model and SCAN algorithm. We adapt LDA for review spammer detection by boosting the number of suspicious reviewers according to review time burstiness and rating score homogeneity. The method only uses review time and rating score data, plus the review bipartite graph, no review content is needed, hence is robust to opinion spamming. Experimental study indicates that our proposed method can dramatically improve the detection

precision, outperforming many state-of-the-art baselines by a large margin. Future work includes developing other effective group spamming indicators, automatically determining the number of clusters K , and combining with other review spamming detection techniques to further improve the detection precision.

References

- Akoglu L, Chandy R, Faloutsos C (2013) Opinion fraud detection in online reviews by network effects. In: Proceedings of the seventh International Conference on Weblogs and Social Media, ICWSM 2013, Cambridge, Massachusetts, USA, July 8–11, 2013
- Allahbakhsh M, Ignjatovic A, Benatallah B, Beheshti S-M-R, Bertino E, Foo N (2013) Collusion detection in online rating systems. In: Web Technologies and Applications, vol. 7808 of Lecture Notes in Computer Science. Springer, Berlin, pp 196–207
- Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. *J Mach Learn Res* 3:993–1022
- Choo E, Yu T, Chi M (2015) Detecting opinion spammer groups through community discovery and sentiment analysis. In: Proceedings of the 29th Annual IFIP WG 11.3 Working Conference on Data and Applications Security and Privacy XXIX, DBSec 2015, Fairfax, VA, USA, July 13–15, 2015, pp 170–187
- Crawford M, Khoshgoftaar TM, Prusa JD, Richter AN, Al Najada H (2015) Survey of review spam detection using machine learning techniques. *J Big Data* 2(1):23
- Fei G, Mukherjee A, Liu B, Hsu M, Castellanos M, Ghosh R (2013) Exploiting burstiness in reviews for review spammer detection. In: Seventh international AAAI conference on weblogs and social media
- Jindal N, Liu B (2008) Opinion spam and analysis. In: Proceedings of the 2008 International Conference on Web Search and Data Mining (New York, NY, USA). ACM, pp 219–230
- Lee KD, Han K, Myaeng S (2016) Capturing word choice patterns with LDA for fake review detection in sentiment analysis. In: Proceedings of the 6th International Conference on Web Intelligence, Mining and Semantics, WIMS 2016, Nimes, France, June 13–15, 2016, pp 9:1–9:7
- Li J, Cardie C, Li S (2013) Topicspam: a topic-model based approach for spam detection. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4–9 August 2013, Sofia, Bulgaria, Volume 2: Short Papers, pp 217–221
- Li J, Ott M, Cardie C, Hovy E (2014) Towards a general rule for identifying deceptive opinion spam. In: Proceedings of the 52nd annual meeting of the association for computational linguistics (Volume 1: Long Papers) (Baltimore, Maryland), pp 1566–1576
- Lim E-P, Nguyen V-A, Jindal N, Liu B, Lauw HW (2010) Detecting product review spammers using rating behaviors. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management (New York, NY, USA), CIKM'10, pp 939–948
- Mukherjee A, Kumar A, Liu B, Wang J, Hsu M, Castellanos M, Ghosh R (2013) Spotting opinion spammers using behavioral footprints. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (New York, NY, USA), KDD'13, pp 632–640
- Mukherjee A, Liu B, Glance N (2012) Spotting fake reviewer groups in consumer reviews. In: Proceedings of the 21st International Conference on World Wide Web (New York, NY, USA). ACM, pp 191–200
- Mukherjee A, Venkataraman V, Liu B, Glance NS (2013) What yelp fake review filter might be doing? In: Proceedings of the Seventh International Conference on Weblogs and Social Media, ICWSM 2013, Cambridge, Massachusetts, USA, July 8–11, 2013
- Ott M, Choi Y, Cardie C, Hancock JT (2011) Finding deceptive opinion spam by any stretch of the imagination. In: Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies - Volume 1 (Stroudsburg, PA, USA), pp 309–319
- Rayana S, Akoglu L (2015) Collective opinion spam detection: Bridging review networks and metadata. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10–13, 2015, pp 985–994
- Sandulescu V, Ester M (2015) Detecting singleton review spammers using semantic similarity. In: Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy, May 18–22, 2015 - Companion Volume, pp 971–976
- Viviani M, Pasi G (2017) Credibility in social media: opinions, news, and health information - a survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 7(5):e1209
- Wang G, Xie S, Liu B, Yu PS (2011) Review graph based online store review spammer detection. In: ICDM, pp 1242–1247
- Wang Z, Gu S, Zhao X, Xu X (2017) Graph-based review spammer group detection. *Knowledge and Information Systems*
- Wang Z, Hou T, Song D, Li Z, Kong T (2016) Detecting review spammer groups via bipartite graph projection. *Comput J* 59(6):861–874
- Xu C, Zhang J (2015) Towards collusive fraud detection in online reviews. In: 2015 IEEE International Conference on Data Mining, ICDM 2015, Atlantic City, NJ, USA, November 14–17, 2015, pp 1051–1056
- Xu C, Zhang J, Chang K, Long C (2013) Uncovering collusive spammers in chinese review websites. In: Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management (New York, NY, USA). ACM, pp 979–988
- Xu X, Yuruk N, Feng Z, Schweiger TAJ (2007) SCAN: a structural clustering algorithm for networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August 12–15, 2007, pp 824–833
- Ye J, Akoglu L (2015) Discovering opinion spammer groups by network footprints. In: Appice A, Rodrigues PP, Santos Costa V, Soares C, Gama J, Jorge A (eds) Machine Learning and Knowledge Discovery in Databases, Volume 9284 of Lecture Notes in Computer Science. Springer International Publishing, pp 267–282



Zhuo Wang received his B.S. degree in Computer Software from Xidian University, Xi'an, China, in 1992 and M.S. degree in Computer Software and Theory from Northeastern University, Shenyang, China, in 2004. He is currently an associate professor in Shenyang Ligong University, China. His research interests include data mining, machine learning, especially anomaly detection in online social data. He has published nearly 20 papers in

many international/domestic academic journals and conferences.



Songmin Gu received his B.S. degree in Computer Science and Technology from Nanjing University of Posts and Telecommunications, China, in 2015. He is currently a master's student in Shenyang Ligong University. His research interests include data mining, machine learning. He has won the secondary prize of Huawei Cup in the 13th domestic master student mathematical modeling contest and the first prize in the first big data contest in Liaoning Province.



Xiaowei Xu is a tenured full professor in the Department of Information Science at the University of Arkansas at Little Rock (UALR). He received his Ph.D. in computer science from the University of Munich. Prior to his appointment at UALR, Dr. Xu was a senior research scientist and the head of a research team at Siemens. Dr. Xu is a recipient of the prestigious ACM SIGKDD Test of Time Award for his pioneered work in the density-based clustering algorithm DBSCAN. He is a program committee member and session chair for premier forums including ACM SIGKDD Conference on Knowledge Discovery and Data Mining. He is a consultant for companies and government agencies including Siemens, Acxiom, and FDA. Dr. Xu's research focuses on data mining, machine learning, and deep learning. Published over 80 research papers with over 15863 citations, he is one of the most cited researchers according to Google Scholar.