Online Reputation Fraud Campaign Detection in User Ratings

Chang Xu, Jie Zhang, Zhu Sun
School of Computer Science and Engineering, Nanyang Technological University, Singapore
xuch0007@e.ntu.edu.sg, zhangj@ntu.edu.sg, sunzhu@ntu.edu.sg

Abstract

Reputation fraud campaigns (RFCs) distort the reputations of rated items, by generating fake ratings through multiple spammers. One effective way of detecting RFCs is to characterize their collective behaviors based on rating histories. However, these campaigns are constantly evolving and changing tactics to evade detection. For example, they can launch early attacks on the items to quickly dominate the reputations. They can also whitewash themselves through creating new accounts for subsequent attacks. It is thus challenging for existing approaches working on historical data to promptly react to such emerging fraud activities. In this paper, we conduct RFC detection in online fashion, so as to spot campaign activities as early as possible. This leads to a unified and scalable optimization framework, FRAUDSCAN, that can adapt to emerging fraud patterns over time. Empirical analysis on two real-world datasets validates the effectiveness and efficiency of the proposed framework.

1 Introduction

Crowdsourced online ratings have been an informative source for users to assess the quality of various items, such as product ratings on Amazon and Page Likes on Facebook. As being more influential, these ratings are capable of affecting the profitability of items being evaluated [Forman *et al.*, 2008]. While positive ratings could boost item reputation and bring increase in sales, negative commentary could damage public image and lead to business failure. This, however, has spurred the rise of an underground business, where *fake* ratings are crafted by spammers to either promote or degrade item reputation [Jindal and Liu, 2008; De Cristofaro *et al.*, 2014]. To stay ahead of defensive systems, spammers are constantly evolving. They are found recently to rely on *astroturfing* [Xu *et al.*, 2015; Wang *et al.*, 2014] to commit large-scale fraud practices. With popular crowdsourcing services, spammers now can easily orchestrate reputation fraud campaigns (RFCs) that take advantage of the efforts of multiple users [Fayazi *et al.*, 2015]. Figure 1 shows a real RFC found in a crowdsourcing platform, which is described as a crowdsourcing task instructing each spammer to leave a positive rating (with monetary rewards) for a targeted item during a specific time period.

Write a Product Review on Amazon.com

Employer: Sunncry | Job ID: 30785 | Category: Sale & Marketing

Job Status: Open (4 days left) **Bidding Started:** May 09, 2016 23:54

Job Reward: US$5.00 **Bidding Ends:** May 17, 2016 23:54

Description

1. You should have an account and have purchase history on amazon.com.
2. One account and one IP can only write one review.
3. I will send you the review and you will only need to copy and paste.
4. Send us a screen shot once you finish it!

Figure 1: A real reputation fraud campaign in the wild.

These campaigns are aggressive and highly damaging due to the availability of manpower for large-scale manipulation. Existing approaches to RFC detection rely on offline behavior analysis and batch models built on static rating history [Li *et al.*, 2011; Feng *et al.*, 2012; Mukherjee *et al.*, 2012; Akoglu *et al.*, 2013; Xu and Zhang, 2015; Ye and Akoglu, 2015; Wentao *et al.*, 2015]. However, the built batch models may quickly become obsolete once the campaigns adopt different strategies. First, they can change attack timing with various intentions. For example, early attacks can be launched to hijack and preconceive an item's initial reputation [Lim *et al.*, 2010; Mukherjee *et al.*, 2012]. They can also generate massive ratings with opposite polarity right after dissenting votes to timely inverse the opinion orientation [Jindal and Liu, 2008]. A built model may need frequent updates to handle attacks with changed timings. Second, RFCs can also change their constituents. To whitewash established attack histories, new accounts with clean background can easily be created or bought for new campaigns. Again, the batch approaches need to train a new model to account for campaigns with changed members. Moreover, from an algorithmic view, existing batch approaches scan the input data multiple times for model building upon each single update, rendering it too costly to quickly respond to evolving campaigns.

To efficiently adapt built models to new attack modes, we take an *online learning* based approach, which has been successfully applied to various situations such as image retrieval [Gao *et al.*, 2014] and natural language processing [Sun *et al.*, 2016], but has not yet been used to detect reputation fraud campaigns. In this approach, the detection model is trained incrementally upon the arrival of new ratings, and thus can adapt to emerging adversarial patterns in the data. However, two challenges are needed to be addressed

when seeking for online solutions. First, many existing features for RFC detection are not suitable for online scenarios; they are designed to capture *occurred* campaigns, but cannot effectively handle *emerging* campaigns due to the lack of collectible evidence when the campaigns are still in progress. Second, the rating data collected in online settings are incomplete and sparse, which could in turn cause the sparsity of the features and deteriorate the models built on the features.

In this paper, to overcome the above issues, we propose a novel and unified framework, FRAUDSCAN, to conduct RFC detection online. In this framework, instead of directly comparing RFCs with normal activities, internal interactions between campaign members are exploited. Such interactions are presumably more indicative of emerging campaigns, since they become observable once the campaigns have begun. In addition, to effectively model the sparse rating behaviors, FRAUDSCAN takes an embedding-based approach to compactly encode user-rating-item relations with dense embeddings. A novel campaign-aware optimization scheme is also proposed for efficient online update. Results of extensive experiments on two real datasets show the superiority of the proposed framework over state-of-the-art methods in both detection accuracy and efficiency.

2 Problem Formulation and Challenges

We introduce the notations and definitions related to the studied problem. To adapt built detection model to constantly changing campaign strategies, we formulate the task of online reputation fraud campaign detection and identify the challenges of fulfilling this task.

Assume that there is a set of n items $\mathcal{V} = \{v_j\}_{j=1}^{n}$ receiving ratings from a set of m users $\mathcal{U} = \{u_i\}_{i=1}^{m}$. A typical item rating consists of four fundamental elements: a *user id* specifying the subject giving the rating, an *item id* referring to the rated item, a numerical *star* instantiating the rating, and a *timestamp* recording the time of the rating. We use a 4-tuple $r_{ij} =< u_i, e_{ij}, v_j, t_{ij} >$ to represent an item rating r_{ij} generated by user u_i for item v_j with star e_{ij} at time t_{ij}.

Definition 1 *Along the time dimension, we can observe a **rating sequence** comprised of an unbounded series of incoming ratings in temporal order $\{r_t\}$, with timestamp $t = \{1, 2, 3, ...\}$. Denote $\mathcal{U}^t$ and $\mathcal{V}^t$ as the sets of users and items up to time t respectively, and a rating matrix $\mathbf{R}^t \in \mathbb{R}^{|\mathcal{U}^t| \times |\mathcal{V}^t|}$ as all the ratings from $\mathcal{U}^t$ to $\mathcal{V}^t$.*

Definition 2 *A **reputation fraud campaign (RFC)** consists of a group of **campaigned spammers** who are asked to conduct reputation fraud on a set of targeted items. A RFC can be denoted by a 3-tuple $< \mathcal{U}_c, \mathcal{R}_c, \mathcal{V}_c >$ with $\mathcal{V}_c$ the set of targeted items, $\mathcal{R}_c$ the set of created ratings, and $\mathcal{U}_c$ the set of involved campaigned spammers.*

Definition 3 *To quantify the tendency of a user to participate in RFCs, a real-valued **spamicity** score $s_u^t \in \mathbb{R}^+$ is assigned to each user $u \in \mathcal{U}^t$ at time t. A ranked list of κ users with the largest spamicities S_κ^t will be produced as the **detection result** at time t.*

Given the definitions above, we formulate the task of **online reputation fraud campaign detection** as below: *Given the old rating matrix $\mathbf{R}^{t-1}$ at time $t-1$, the model $\mathbf{M}^{t-1}$ learned from $\mathbf{R}^{t-1}$, and the detection result S_κ^{t-1}, upon a new rating comes in at time t, the goal is to efficiently update the new model $\mathbf{M}^t$ for reputation fraud campaign detection on the current rating matrix $\mathbf{R}^t$, and output the up-to-date detection result S_κ^t.*

The fulfillment of the above task, however, requires two challenges to be addressed. The first is concerned with effective emerging campaign characterization. Various features have been proposed to characterize *occurred* campaigns on different dimensions. For example, deviation-based features [Lim *et al.*, 2010; Mukherjee *et al.*, 2012; 2013] are commonly used to capture the differences between the ratings given by spammers and the remaining users. However, when dealing with emerging campaigns, if an item has not yet received sufficient ratings to form a general "ground-truth" rating at the beginning, the deviation of any incoming rating at this point would be no longer statistically significant to indicate manipulation. Another example is the profile-based features that focus on the external properties of campaigns, such as *Group Size* and *Group Support Count* used in [Mukherjee *et al.*, 2012]. These features may fail to accurately reflect the traits of emerging campaigns, since the size of a campaign is yet undetermined before the campaign ends.

The second challenge is about effective user rating behavior modeling. In online environments, the rating matrix $\mathbf{R}$ is inherently incomplete and sparse, which would cause many existing features to also be very sparse, and further hurt the performance of models built on those features. More robust representations are thus needed to encode user rating behaviors so as to effectively differentiate between campaigned spammers and normal users.

3 The Proposed Framework

To overcome the above challenges, we propose a novel framework, FRAUDSCAN, for online reputation fraud campaign detection. In light of the first challenge, we introduce the notion of campaign context to capture the *interactions* between campaigned spammers. Such interactions are driven by ongoing collaborations between spammers from the same campaigns, as they are instructed to target the same items and give ratings with the same polarity during the campaign period (as illustrated in Figure 1). By monitoring such interactions across time, it is possible to spot emerging campaigns once they have launched. Second, to model the sparse rating behaviors, we take an embedding-based approach [Hofmann, 1999] where each user or item is mapped to a dense embedding formed by considering all user-rating-item relations in the data. A novel regularized matrix factorization model is proposed to substantiate the campaign embedding in the campaign context, and a campaign-aware online update scheme to further efficiently adapt the built detection model to new ratings.

3.1 Campaign Activity Monitoring

In this section, we introduce the campaign context built to monitor emerging campaign activities. Different from batch solutions, the campaign context is maintained incrementally upon the arrival of new ratings.

Target conformity

The first type of campaign context is focused on the items targeted by campaigned spammers. Due to the involvement in the same campaigns, spammers would naturally co-rate the

same items, and are tied by such co-rating actions. We model such co-rating behaviors as a matrix $\mathbf{C}^t \in \mathbb{R}^{|\mathcal{U}^t|\times|\mathcal{U}^t|}$, where each entry $c^t_{i,i'}$ is the normalized number of common items rated by both user u_i and $u_{i'}$ up to time t. The update rule for $\mathbf{C}^t$ is as follows: upon the arrival of a new rating r_{ij} from user u_i to item v_j, we update $c^t_{i,i'}$ as,

$$\begin{aligned} {c^*_{i,i'}}^t &= {c^*_{i,i'}}^{t-1} + 1 \quad \text{if } u_{i'} \text{ had also rated } v_j \\ c^t_{i,i'} &= {c^*_{i,i'}}^t / (\sqrt{|\mathcal{V}^t_i|} \cdot \sqrt{|\mathcal{V}^t_{i'}|}) \end{aligned} \tag{1}$$

where $c^*_{i,i'}$ is a helper variable for counting; $\mathcal{V}^t_i$ and $\mathcal{V}^t_{i'}$ are the items rated by u_i and $u_{i'}$ up to time t respectively.

Agreement intensity

The ratings given by campaigned spammers should also agree on the co-rated items, as they are instructed to post either positive ratings for promotion or negative ratings for demotion. Thus, the differences between their ratings given to the co-rated items should be small. We use a matrix $\mathbf{G}^t \in \mathbb{R}^{|\mathcal{U}^t|\times|\mathcal{U}^t|}$ to capture the intensity of spammers' agreements. Each entry $g^t_{i,i'}$ keeps the up-to-date average rating difference between u_i and $u_{i'}$, and is updated as follows,

$$\begin{aligned} {g^*_{i,i'}}^t &= {g^*_{i,i'}}^{t-1} + e^{-|r_{ij}-r_{i'j}|^2} \quad \text{if } u_{i'} \text{ had also rated } v_j \\ g^t_{i,i'} &= {g^*_{i,i'}}^t / {c^*_{i,i'}}^t \end{aligned} \tag{2}$$

where $g^*_{i,i'}$ is the helper variable, and RBF (radial basis function) kernel is used to compute the rating deviation.

Operational constraint

RFCs are often associated with time schedules specifying for how long the campaigns would last. In order to achieve the desired fraud effects, spammers are required to finish their jobs in time, such that their utilities can be aggregated. Such a constraint, however, would necessarily lead to small gaps between the timings of fraud activities. To account for this intuition, a matrix $\mathbf{O}^t \in \mathbb{R}^{|\mathcal{U}^t|\times|\mathcal{U}^t|}$ is used for tracking the proximity of users in time, with each entry $o^t_{i,i'}$ recording the average time interval between ratings given by u_i and $u_{i'}$ on the co-rated items so far, and being updated as below,

$$\begin{aligned} {o^*_{i,i'}}^t &= {o^*_{i,i'}}^{t-1} + e^{-|t_{ij}-t_{i'j}|^2} \quad \text{if } u_{i'} \text{ had also rated } v_j \\ o^t_{i,i'} &= {o^*_{i,i'}}^t / {c^*_{i,i'}}^t \end{aligned} \tag{3}$$

where $o^*_{i,i'}$ is the helper variable, and $t_{ij}(t_{i'j})$ is the timestamp of $u_i(u_{i'})$'s rating for v_j.

3.2 Fraud Campaign Embedding

As discussed, to address the sparsity of incoming rating data, we take an embedding-based approach for robust user rating behavior modeling, and propose a regularized matrix factorization model for RFC embedding. Concretely, we map each entity (a user or an item) in the data to a real-valued, dense vector in a low-dimensional space by taking all user-rating-item relations into account. This can be achieved by factorizing the original rating matrix $\mathbf{R}^t$ into two matrices, i.e., user matrix $\mathbf{U}^t \in \mathbb{R}^{k\times m}$ and item matrix $\mathbf{V}^t \in \mathbb{R}^{k\times n}$, where $k \ll \min(m,n)$ is the rank of the embedded space, which is commonly set to be low (e.g., within $[10, 50]$). Each column $\mathbf{u}^t_i$ of $\mathbf{U}^t$ (or $\mathbf{v}^t_j$ of $\mathbf{V}^t$) corresponds to the current embedding of user u_i (or item v_j). Meanwhile, to effectively capture campaign activities, we regularize such factorization in a proper way so that *users who have stronger interactions in the campaign context are mapped to closer regions in the embedded space*. As such, spammers from the same campaigns can be clustered together due to their strong connections in the campaign context, while normal users on the other hand would be dispersed randomly in the embedded space. Formally, we model the task of online RFC detection as a regularized embedding problem:

$$\begin{aligned} \arg\min_{\mathbf{u}^t,\mathbf{v}^t} L^t &= \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}(r_{ij} - {\mathbf{u}^t_i}^T\mathbf{v}^t_j)^2 \\ &+ \frac{\alpha_c}{2}\sum_{i=1}^{m}\sum_{i'>i}^{m} c^t_{i,i'}\|\mathbf{u}^t_i - \mathbf{u}^t_{i'}\|^2_F + \frac{\alpha_g}{2}\sum_{i=1}^{m}\sum_{i'>i}^{m} g^t_{i,i'}\|\mathbf{u}^t_i - \mathbf{u}^t_{i'}\|^2_F \\ &+ \frac{\alpha_o}{2}\sum_{i=1}^{m}\sum_{i'>i}^{m} o^t_{i,i'}\|\mathbf{u}^t_i - \mathbf{u}^t_{i'}\|^2_F + \frac{\lambda_u}{2}\|\mathbf{U}^t\|^2_F + \frac{\lambda_v}{2}\|\mathbf{V}^t\|^2_F \end{aligned} \tag{4}$$

where I_{ij} is the indicator function that equals to 1 if u_i rated v_j, and 0 otherwise. $\|\cdot\|_F$ is Frobenius Norm. $\alpha_c, \alpha_g, \alpha_o, \lambda_u, \lambda_v$ are non-negative regularization parameters. The term $\sum_{i=1}^m\sum_{j=1}^n I_{ij}(r_{ij} - {\mathbf{u}^t_i}^T\mathbf{v}^t_j)^2$ is the objective function that factorizes the input rating matrix $\mathbf{R}^t$ into user and item matrices $\mathbf{U}^t$ and $\mathbf{V}^t$. The terms $\sum_{i=1}^m\sum_{i'=1}^m (*)^t_{i,i'}\|\mathbf{u}^t_i - \mathbf{u}^t_{i'}\|^2_F$ are inspired by fused lasso [Tibshirani *et al.*, 2005], and used to constrain the former objective function, so as to bridge the campaign context and the embedded space. More specifically, they are designed to penalize large differences between the embeddings of two users if they are strongly correlated in the campaign context (i.e., with large $c^t_{i,i'}, g^t_{i,i'}, o^t_{i,i'}$). As such, campaigned spammers will obtain similar embeddings (in the sense of small Frobenius distance) due to their strong connections in the campaign context. The last two terms $\|\mathbf{U}^t\|^2_F$ and $\|\mathbf{V}^t\|^2_F$ are regularizers for avoiding overfitting.

To solve this optimization problem, stochastic gradient descent (SGD) is commonly used in the literature. In our case, for each rating $r_{ij} \in \mathbf{R}^t$, the following updating equations are used to learn the detection model $\mathcal{M}^t = \{\mathbf{U}^t, \mathbf{V}^t\}$:

$$\begin{aligned} \mathbf{u}^t_i &\leftarrow \mathbf{u}^t_i + \beta_u\Bigg((r_{ij} - {\mathbf{u}^t_i}^T\mathbf{v}^t_j)\mathbf{v}^t_j - \alpha_c\sum_{i'}^{m} c^t_{i,i'}(\mathbf{u}^t_i - \mathbf{u}^t_{i'}) \\ &\quad - \alpha_g\sum_{i'}^{m} g^t_{i,i'}(\mathbf{u}^t_i - \mathbf{u}^t_{i'}) - \alpha_o\sum_{i'}^{m} o^t_{i,i'}(\mathbf{u}^t_i - \mathbf{u}^t_{i'}) - \lambda_u\mathbf{u}^t_i\Bigg) \\ \mathbf{v}^t_j &\leftarrow \mathbf{v}^t_j + \beta_v\Big((r_{ij} - {\mathbf{u}^t_i}^T\mathbf{v}^t_j)\mathbf{u}^t_i - \lambda_v\mathbf{v}^t_j\Big) \end{aligned} \tag{5}$$

where β_u and β_v are the learning rates.

Once SGD converged, for each user pair $(u_i, u_{i'})$, we compute the (inverted) Frobenius distance $s^t_{i,i'} = \frac{1}{1+\|\mathbf{u}^t_i - \mathbf{u}^t_{i'}\|^2_F}$ to measure the degree of their closeness in the embedded space. Then, the spamicity of a user u_i is derived as the highest closeness she has with any other user so far, i.e., $s^t_i = \max(\{s^t_{i,i'} | u_{i'} \in \mathcal{U}^t\})$. To generate the detection result S^t_κ, we rank the users by their current spamicities, and obtain the list of κ users with the largest values.

3.3 Online Campaign Detection

The above optimization procedure, however, is not sufficient for efficient and effective online RFC detection. First, the update scheme is in batch-mode; the model is not updated across time. Second, it implicitly assumes that the users are independent, and does not utilize the fact that campaigned spammers are correlated in the campaign context. Here, we propose a *campaign-aware* online optimization scheme to update the built detection model. The key observation is that, whenever a new rating arrives, it is not necessary to make updates for all users and items in the data. Instead, only the parts of the model that are related to the incoming rating should be affected. More specifically, let r_{ij} be the current rating from a user u_i to an item v_j, and U_j^t the set of users who had rated v_j up to time t. Then, the arrival of r_{ij} would only change the relations between u_i and those in U_j^t in the campaign context (see Eq. (1-3)). Moreover, as rating data increase, the impact of each rating to the embedding model decreases, and new ratings would not significantly change the entire embedded space. Thus, to obtain the current user matrix $\mathbf{U}^t$ and item matrix $\mathbf{V}^t$, it is sufficient to only update the columns of $\mathbf{U}^{t-1}$ that correspond to u_i and the users in U_j^{t-1}, and the column of $\mathbf{V}^{t-1}$ that corresponds to v_j. In other words, to obtain the current detection model $\mathcal{M}^t = \{\mathbf{U}^t, \mathbf{V}^t\}$, we approximate $\mathbf{U}^t$ and $\mathbf{V}^t$ by updating the embeddings of u_i, U_j^{t-1}, and v_j that undergo changes in the campaign context, and keep the remaining part fixed as $\mathcal{M}^{t-1}$. This leads to Algorithm 1 for incremental RFC detection model update.

Algorithm 1: Incremental RFC Detection Flow

Input: $\mathbf{C}^t$, $\mathbf{G}^t$, $\mathbf{O}^t$, $\mathbf{U}^{t-1}$, $\mathbf{V}^{t-1}$, S_κ^{t-1}, $\mathbf{R}^{t-1} \cup \{r_{ij}\}$
Output: $\mathbf{U}^t$, $\mathbf{V}^t$, S_κ^t

```
// (1) Initialization
if u_i is new then
    Append a new column to the rightmost of U^{t-1};
    Initialize u_i, i.e., the u_i's column in U^{t-1};
if v_j is new then
    Append a new column to the rightmost of V^{t-1};
    Initialize v_j, i.e., the v_j's column in V^{t-1};
// (2) Update user and item matrix
Initialize u_i;
repeat
    foreach r_{i*} ∈ R_i^{t-1} ∪ {r_ij} do
        Update u_i and v_* according to Eq. (5);
    foreach u_{i'} ∈ U_j^t do
        Update u_{i'} according to Eq. (5);
until Convergent or maximum iteration reached;
// (3) Update detection report
foreach (u_i, u_{i'}) do
    s_{i,i'}^t = 1 / (1 + ||u_i^t - u_{i'}^t||_F^2);
    s_i^t = max({s_{i,i'}^t | u_{i'} ∈ U^t});
    Insert {s_i^t, u_i} into S_κ^{t-1};
return U^t, V^t, S_κ^t
```

The algorithmic flow consists of three stages. First, we initialize the embeddings of u_i and v_j if either of them is new (line 1–6). Then, the campaign embedding is updated (line 7–13). In this stage, we re-train the embedding of u_i by first initializing $\mathbf{u}_i$ and then updating the embedding according to her up-to-date rating profile $\mathbf{R}_i^t = \mathbf{R}_i^{t-1} \cup \{r_{ij}\}$ (line 9-10). Then, we update the embeddings of users in U_j^t who are related to the current rating in the campaign context (line 11-12). Finally, we generate the detection report (line 14–17). For efficient update, the report S_κ is implemented as a max-heap that stores the largest κ spamicities and the corresponding users, which supports $O(\log \kappa)$ insertion. Through only updating the entities undergoing changes in campaign context, the complexity of online RFC detection has been considerably reduced. The time complexity of Algorithm 1 is $O((|\mathbf{R}_i^t| + |U_j^t|) \times k \times Iter)$ for campaign embedding update and $O(|U_j^t| \log \kappa)$ for result reporting. Note that $\mathbf{R}_i^t$ and U_j^t are only the ith and jth columns of rating matrix $\mathbf{R}^t$, which are typically small in real applications (e.g., power-law distributed).

4 Experimental Analysis

We validate the proposed FRAUDSCAN for online reputation fraud campaign detection based on extensive experiments on real-world datasets. The evaluation is focused on the effectiveness of the proposed framework in RFC detection, and the efficiency of the proposed incremental scheme for detection model update.

4.1 Datasets

Our experiments are conducted on two real-world datasets.

Restaurant Reviews on Yelp (YelpZip): This dataset was used in [Rayana and Akoglu, 2015], which contains reviews about restaurants on Yelp.com. These reviews were collected through restaurant searching on Yelp via zipcode. The annotation is based on Yelp's own mechanism that classifies reviews as recommended or filtered. Users who authored filtered reviews are considered as the golden standard of spammers. However, there is no annotation for campaigned spammers in the dataset. We then follow the procedure in [Mukherjee *et al.*, 2012] to extract user groups who have co-ratings, and regard groups whose members are all spammers as fraud campaigns.

Product Reviews on Amazon (AmazonCn): This dataset was created by [Xu *et al.*, 2013]. It contains consumer reviews on amazon.cn. It already has the gold standard of campaigned spammers, which was obtained in a similar way by first locating users whose reviews were filtered by Amazon, and then grouping those exhibiting co-rating behaviors.

Table 1 shows the statistics of the two datasets.

Table 1: Statistics of the used datasets.

	AmazonCn	**YelpZip**
# ratings	1,205,125	608,598
# users	645,072	260,277
# items	136,785	5,044
# campaigned spammers	1,937	474

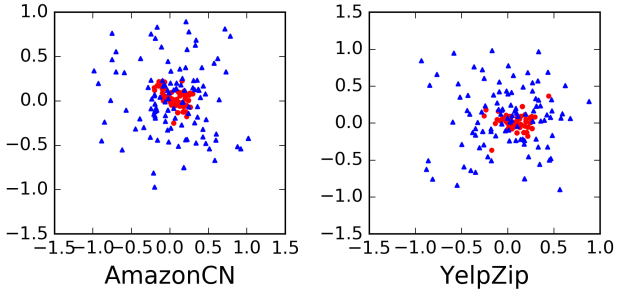
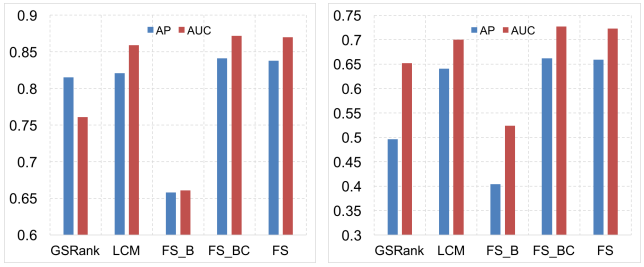


Figure 2: User projections in the embedded space (the first two principal components are used). Samples (uniform) consisting of 100 campaigned spammers (in red/circle) and 100 normal users (in blue/triangle) are shown.



(a) AmazonCn (b) YelpZip

Figure 3: Effectiveness comparison between FRAUDSCAN and state-of-the-art approaches on the two datasets.

4.2 Evaluation Protocol

For effectiveness comparison, as FRAUDSCAN outputs a ranked list of users with numerical spamicity scores, we adopt two well-known metrics - Average Precision (AP) and Area Under ROC Curve (AUC) - for ranking performance evaluation. Average Precision is essentially the area under precision-recall curve, and AUC is commonly used to validate binary classification systems by varying the discrimination threshold. For efficiency evaluation, runtime of the detection algorithm is used as the performance metric.

4.3 Performance Evaluation

In this section, we evaluate the effectiveness of FRAUDSCAN for reputation fraud campaign detection.

Qualitative analysis

We first conduct qualitative analysis. The success of FRAUDSCAN lies in the regularized campaign embedding that maps spammers from the same campaigns to close regions in the embedded space. Figure 2 shows the user projections in the embedded space on the two datasets. We can see that on both datasets, the campaigned spammers are effectively clustered together as expected, while the normal users are more scattered over the space, which validates the efficacy of the proposed embedding model. In addition, this also shows that our method can be used for visualization, which is lacking in other unsupervised methods that yield spamicity scores only.

Comparison analysis

Next, we compare FRAUDSCAN with state-of-the-art approaches, with the following baselines used:

- GSRank [Mukherjee *et al.*, 2012]: a power iteration-based approach for spotting campaigned spammers based on the causal relations between users, reviews, and items. A set of Group Spam Behavior Indicators are proposed to capture suspicious collective behaviors of campaigned spammers.
- LCM [Xu and Zhang, 2015]: a statistical framework to model the collective behaviors of campaigned spammers for campaign inference and prediction.

In addition, two variants of FRAUDSCAN (FS) are also used:

- FS_B: this is a batch variant of the proposed framework. In this variant, only the matrix factorization objective is optimized, without the use of campaign context-based regularization.
- FS_BC: this is a batch variant of the proposed framework, with campaign context-based regularization used.

We empirically set optimal parameters for the hyperparameters (e.g., learning rate) of each method using a grid search in $\{0.001, 0.01, 0.1, 0.5, 1\}$. For FRAUDSCAN and its variants, four parameters α_1, α_2, α_3, and k need to be tuned. We will show later the sensitivity analysis conducted on these parameters. Here, the optimal settings $(\alpha_1, \alpha_2, \alpha_3, k)$ are used, i.e., $(0.1, 0.1, 0.01, 20)$ for AmazonCn and $(0.01, 0.1, 0.01, 30)$ for YelpZip. Figure 3 shows the performance comparison on the two datasets. In the figure, each result is the average over 10 runs of the corresponding experiment. All the improvements in the results are significant at $p < .05$ with paired t-test. The following observations are made based on the result comparison.

First, we can see that our proposed methods FS_BC and FRAUDSCAN achieve better results than other baselines on both datasets. In particular, by comparing FS_BC with the two start-of-the-art methods GSRank and LCM, the performance is improved by a margin of [2.4%, 3.2%] in AP and [1.5%, 14.5%] in AUC on AmazonCn, and [3.2%, 33.4%] in AP and [3.8%, 11.5%] in AUC on YelpZip. This might be ascribed to the embedding-based approach used in FRAUDSCAN for user rating behavior modeling, which compactly encodes rich user-rating-item relationships in the sparse original rating matrix in a more robust way.

Second, by comparing different variants of FRAUDSCAN, we observe that FS_B performs worse than the other two on both datasets, suggesting that the factorization of the input rating matrix alone is not sufficient for campaign activity characterization. By further utilizing the campaign context proposed for capturing interactions between campaigned spammers, the performance of FS_BC is largely improved. This shows that the incorporation of campaign-level information is crucial for collusive behavior modeling. Finally, FS_BC and FRAUDSCAN achieve comparable results on both datasets, suggesting that the approximation adopted in the proposed online detection approach does not harm the embedded model building and the detection performance.

4.4 Earliness of Detection

One important objective of online detection is to spot emerging campaign activities as early as possible, so as to limit the caused damage in time. To this regard, we run the compared methods across time, with a one-day interval, and measure the earliness of detection by computing the lag between the time when a spammer appeared for the first time in the sys-

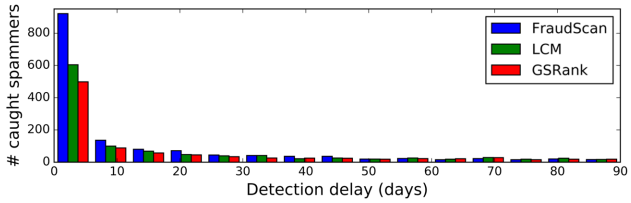


Figure 4: Histograms of detection time lags for spammers caught by the compared methods. The histograms are truncated at 90-day lag due to the long-tail shape.

tem and the time when (s)he was first detected. Ideally, the lag should be zero for all spammers.

Figure 4 shows the results on AmazonCn[1], by drawing histograms of detection time lags (in days) for caught spammers (true positives)[2]. We find that all the evaluated methods can discover more than half of the caught spammers in less than 10 days after their first ratings. In particular, FRAUDSCAN achieves better results than others by catching more spammers with less delay. Surprisingly, 921 spammers are caught by FRAUDSCAN right after they have posted the first ratings (zero delay), which is 34.4% more than LCM and 45.9% more than GSRank. This shows that compared to other methods, FRAUDSCAN is more effective in quickly responding to emerging reputation fraud campaigns.

4.5 Sensitivity Analysis

In this section, we evaluate the influences of model parameters on the detection performance. They are α_1, α_2, α_3, and k in our framework controlling the importances of the three types of campaign context and the rank of the embedded space. For each α parameter, an independent experiment is conducted by setting other parameters to zero. The results on AmazonCn are shown in Figure 5.

We can see that all the results show similar patterns. The detection performance first rises as the parameter values increase. It then reaches the peak at particular settings and decreases afterwards. This is because when the parameters are

[1]We omit the results on YelpZip due to less significant improvements being observed. The main reason is that the activeness of campaigns on YelpZip is relatively low, resulting in inherently large time lags between consecutive ratings.

[2]We regard the users ranked at the top 2000 as positives, since there are 1,937 annotated campaigned spammers in the dataset.

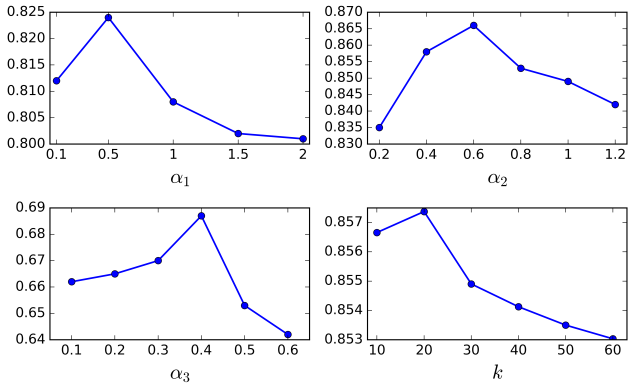


Figure 5: The effects of parameters α_1, α_2, α_3, and k on the detection performance (AUC).

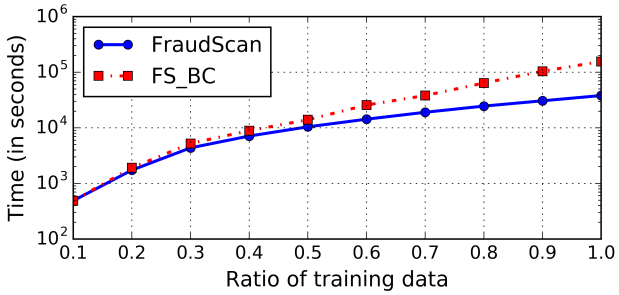


Figure 6: The cumulative distribution of detection latency.

set too small, the campaign context cannot be fully exploited to improve detection performance. However, the overuse of the campaign context can hurt the detection performance, since each type of the campaign context is just a necessary condition for characterizing RFCs. It can be seen that as long as the parameters are set within a medium range, the detection performance can always be improved compared to that without the campaign context. We can also observe that the rank of the embedded space should be set sufficiently large to achieve an optimal representation capacity.

4.6 Efficiency Analysis

Finally, we evaluate the runtime performance of our framework to validate its efficiency in RFC detection. In particular, the performance of the batch model FS_BC and that of the online learning model FRAUDSCAN are compared. The experiments are conducted on a machine with a single-CPU, 3.20Ghz and 16G memory. Figure 6 shows the results on AmazonCn due to its larger size. We vary the size of training data, and record the execution time of each baseline. It can be observed that FRAUDSCAN requires less execution time than the batch version FS_BC on datasets with different sizes. This shows that our online learning method is more efficient than the batch counterpart in detection. Moreover, as the size of training data increases, the differences in efficiency between the two approaches become more significant. Specifically, when the full dataset is used, the proposed online learning approach achieves around 4.1x speedup in detection.

5 Conclusion and Future Work

In this paper, we study the problem of reputation fraud campaign detection in user ratings, and propose a principled optimization framework for emerging campaign detection. The new framework is based on the online learning scheme, and operates by monitoring campaign activities across time. The notion of campaign context is introduced to capture emerging campaigns via monitoring interactions among spammers. Moreover, to accurately model users' sparse rating behavior, an embedding approach is adopted to compactly map users and items into low-dimensional space. In the approach, the campaign context serves as model regularizers which can project campaigned spammers onto close regions in the embedded space. A campaign-aware online learning scheme is finally developed for efficient model update. The experiments show that our method achieves superior detection accuracy and efficiency over batch-based competitors. As future work, a potential extension of our approach is to analyze and incorporate other useful side information such as review text to improve model building.

Acknowledgments

We thank the anonymous reviewers for their valuable comments. This work was supported by the MOE AcRF Tier 2 Grant M4020110.020 awarded to Dr. Jie Zhang.

References

[Akoglu *et al.*, 2013] Leman Akoglu, Rishi Chandy, and Christos Faloutsos. Opinion fraud detection in online reviews by network effects. In *Proceedings of the 7th International AAAI Conference on WebBlogs and Social Media (ICWSM)*., 2013.

[De Cristofaro *et al.*, 2014] Emiliano De Cristofaro, Arik Friedman, Guillaume Jourjon, Mohamed Ali Kaafar, and M Zubair Shafiq. Paying for likes?: Understanding facebook like fraud using honeypots. In *Proceedings of the 14th Internet Measurement Conference (IMC)*, pages 129–136. ACM, 2014.

[Fayazi *et al.*, 2015] Amir Fayazi, Kyumin Lee, James Caverlee, and Anna Squicciarini. Uncovering crowdsourced manipulation of online reviews. In *Proceedings of the 38th International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 233–242. ACM, 2015.

[Feng *et al.*, 2012] Song Feng, Longfei Xing, Anupam Gogar, and Yejin Choi. Distributional footprints of deceptive product reviews. In *Proceedings of the International AAAI Conference on WebBlogs and Social Media (ICWSM)*, 2012.

[Forman *et al.*, 2008] Chris Forman, Anindya Ghose, and Batia Wiesenfeld. Examining the relationship between reviews and sales: The role of reviewer identity disclosure in electronic markets. *Information Systems Research*, 19(3):291–313, 2008.

[Gao *et al.*, 2014] Xingyu Gao, Steven CH Hoi, Yongdong Zhang, Ji Wan, and Jintao Li. Soml: Sparse online metric learning with application to image retrieval. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*, pages 1206–1212, 2014.

[Hofmann, 1999] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 50–57. ACM, 1999.

[Jindal and Liu, 2008] Nitin Jindal and Bing Liu. Opinion spam and analysis. In *Proceedings of the 1st International Conference on Web Search and Data Mining (WSDM)*, pages 219–230. ACM, 2008.

[Li *et al.*, 2011] Fangtao Li, Minlie Huang, Yi Yang, and Xiaoyan Zhu. Learning to identify review spam. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, 2011.

[Lim *et al.*, 2010] Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan 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. ACM, 2010.

[Mukherjee *et al.*, 2012] Arjun Mukherjee, Bing Liu, and Natalie Glance. Spotting fake reviewer groups in consumer reviews. In *Proceedings of the International Conference on World Wide Web (WWW)*, 2012.

[Mukherjee *et al.*, 2013] Arjun Mukherjee, Abhinav Kumar, Bing Liu, Junhui Wang, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. Spotting opinion spammers using behavioral footprints. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 632–640. ACM, 2013.

[Rayana and Akoglu, 2015] Shebuti Rayana and Leman Akoglu. Collective opinion spam detection: Bridging review networks and metadata. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 985–994. ACM, 2015.

[Sun *et al.*, 2016] Fei Sun, Jiafeng Guo, Yanyan Lan, Jun Xu, and Xueqi Cheng. Sparse word embeddings using l1 regularized online learning. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2915–2921, 2016.

[Tibshirani *et al.*, 2005] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

[Wang *et al.*, 2014] Gang Wang, Tianyi Wang, Haitao Zheng, and Ben Y Zhao. Man vs. machine: Practical adversarial detection of malicious crowdsourcing workers. In *Proceedings of the 23rd USENIX Security Symposium (USENIX Security 14)*, pages 239–254, 2014.

[Wentao *et al.*, 2015] Tian Wentao, Xu Yinqing, Shi Bei, and Wai Lam. A unified model for unsupervised opinion spamming detection incorporating text generality. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 725–731, 2015.

[Xu and Zhang, 2015] Chang Xu and Jie Zhang. Towards collusive fraud detection in online reviews. In *Proceedings of the 15th IEEE International Conference on Data Mining (ICDM)*, pages 1051–1056. IEEE, 2015.

[Xu *et al.*, 2013] Chang Xu, Jie Zhang, Kuiyu Chang, and Chong Long. Uncovering collusive spammers in chinese review websites. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (CIKM)*, pages 979–988. ACM, 2013.

[Xu *et al.*, 2015] Haitao Xu, Daiping Liu, Haining Wang, and Angelos Stavrou. E-commerce reputation manipulation: The emergence of reputation-escalation-as-a-service. In *Proceedings of the 24th International Conference on World Wide Web (WWW)*, pages 1296–1306. ACM, 2015.

[Ye and Akoglu, 2015] Junting Ye and Leman Akoglu. Discovering opinion spammer groups by network footprints. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 267–282. Springer, 2015.