Simultaneously Detecting Fake Reviews and Review Spammers using Factor Graph Model

Yuqing Lu, Lei Zhang, Yudong Xiao, and Yangguang Li
Tsinghua National Laboratory for Information Science and Technology(TNList),
Graduate School at Shenzhen, Tsinghua University
Yuqing.lu@keg.cs.tsinghua.edu.cn, zhanglei@sz.tsinghua.edu.cn, Yudongxiao@gmail.com,
liyangguang1988@gmail.com

ABSTRACT
Review spamming is quite common on many online shopping platforms like Amazon. Previous attempts for fake review and spammer detection use features of reviewer behavior, rating, and review content. However, to the best of our knowledge, there is no work capable of detecting fake reviews and review spammers at the same time. In this paper, we propose an algorithm to achieve the two goals simultaneously. By defining features to describe each review and reviewer, a Review Factor Graph model is proposed to incorporate all the features and to leverage belief propagation between reviews and reviewers. Experimental results show that our algorithm outperforms all of the other baseline methods significantly with respect to both efficiency and accuracy.

ACM Classification Keywords
H.2.8 Database Management: Data Mining

General Terms
Algorithms, Experimentation

Author Keywords
opinion spam, fake review, factor graph

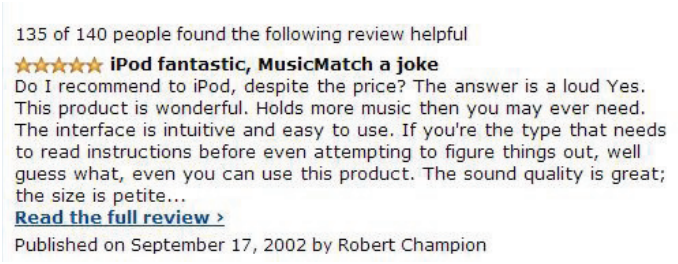
INTRODUCTION

With the ever-increasing popularity of E-Commerce(e.g., Amazon[1]), people are more likely to write reviews on the E-Commerce web to express their views or opinions on the target product. There comes an increasing potential for monetary gain through fake review in order to promote the target product or to damage target products' reputation.

With the large number of user-generated content, the researchers have developed various methods to analyze sentiment in reviews, including natural language processing and data mining techniques[2, 5, 16, 17, 20]. However, all these works have the same assumption: the opinion resources are truthful, which obviously do not conform to the reality. Prior work in [3] have shown that 10-15% reviews essentially echo the earliest reviews and may potentially be influenced by earliest fake review.

These fake reviews are called *opinion spam*[6]. Previous work mainly focus on detecting disruptive opinion spam which have significant fake signals, e.g., high text similarity[6], large deviation with average rating[11]. In [14], several group features based on text similarity and rating have been used to spot fake reviewer groups. In [10], utilizing two views of review: features of review and features of reviewers, they use co-training algorithm to identify review spam.

Before proceeding further, let's have a look at an example of spam review shown in Figure 1. The review content sounds reasonable and details the advantage of the target product, what's more, it even get quite high helpful feedback number and high helpful feedback rate. You may judge this review as a truthful review. However, all the other reviews written by the reviewer are with similar content. That is to say, the reviewer should be spammer and the label of reviewer can help us judge the label of reviews. As a result, in our work, this review is judged as a insidious fake review.


135 of 140 people found the following review helpful

☆☆☆☆☆ **iPod fantastic, MusicMatch a joke**

Do I recommend to iPod, despite the price? The answer is a loud Yes. This product is wonderful. Holds more music then you may ever need. The interface is intuitive and easy to use. If you're the type that needs to read instructions before even attempting to figure things out, well guess what, even you can use this product. The sound quality is great; the size is petite...

Read the full review ›

Published on September 17, 2002 by Robert Champion

Figure 1. Example of fake review

Apparently, these disruptive fake reviews can be identified by an expert reader. There are also some other work still need to be done: (i) *Detecting insidious fake review*: There also exist some insidious opinion spam meanwhile. They don't have similar content with other reviews and have similar rating with average rating. They don't have similar review content with other reviews and have small deviation with average rating. (ii) *Detecting fakers*: We want to detect fake reviewers, i.e., spammers. They are the main source of fake reviews and these reviewers are more likely to post fake reviews.

In this work, we propose an interesting problem: Detecting

[1]http://www.amazon.com

WebSci'13, May 1 – May 5, 2013, Paris, France.
ACM 978-1-4503-1889-1.

fake reviews and fakers simultaneously in an united framework. There are several key challenges: (i) How to incorporate all these various features into an united framework; (ii) It is very difficult to manually label fake reviews or reviewers for model training, especially we have to detect not only obvious spam review but also insidious fake review.

This research makes the following main contributions:

- **Labeled Dataset**: Several college students were first trained to spot fake review. For the simplify, we only label each review and reviewer with binary value. To ensure the quality of label, each review have been annotated by two students independently. If a review or reviewer get different label, it will be annotated by another two students (Section 3).
- **Factor Graph Model**: Referring to previous work, we get abundant of features, in addition, we also define a review group feature and several mutual reinforcement features between reviewers and reviews (Section 4). We further define a review factor graph(RFG) model to incorporate all features (Section 4).
- **Model Learning and Inference**: For learning our defined RGF model, we design an efficient algorithm to optimize the joint probability via belief propagation (Section 4). By means of five-fold cross validation, we conduct experiments to validate the effectiveness of our method. Our method achieve significant improvement compared with other baseline methods (Section 5).

The overview of our proposed method is a supervised learning framework. Our experiments were conducted in the five-cross validation procedure. In model training, we estimated the best parameter configuration so that the log-likelihood of observation value are maximized in our training dataset. In test, given attributes of all reviewers and reviews and the relation between review and review in our test dataset, we perform the belief propagation in our model to estimated the probality of each reviewer to be a spammer or each review to be a fake review. The experiments results show that by leveraging the belief propagation between reviewers and reviews, our approach can significantly improve the accuracy compared to other baseline methods.

The rest of the paper is organized as follows. Section 2 formally formulates the problem; Section 3 details our dataset collection; Section 4 explain our model framework; Section 5 gives and compares experimental results; Section 6 discuss related work and Section 7 concludes the paper.

PROBLEM FORMULATION

In this section, after presenting several definitions, we formally define the targeted problem in this work. We formulate the problem in the context of Amazon dataset[6].

In our work, the set of reviews can be represented as $R = \{r_1, r_2, ..., r_n\}$, where n denotes the amount of reviews in the dataset, the set of reviewers (users) can be represented as $U = \{u_1, u_2, ..., u_{n'}\}$, where $n^{'}$ is the amount of reviewers in the dataset. Then the amazon reviews and reviewers

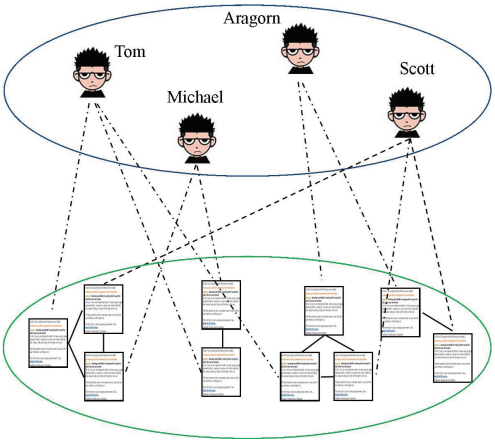


Figure 2. Example of the review graph

can be modeled as an undirected graph $G = (V, E)$, where $V = \{R \cup U\}$ is the union collection of reviews set R and reviewers set U, and $E \subset \{(R \times U) \cup (U \times U)\}$ is the set of undirected links, each undirected link between user and review $e_{ij} = (u_i, r_j)\}$ indicates that user u_i write the review r_j, each undirected link between reviews $e_{ij} = (r_i, r_j)$ indicates that reviews r_i and r_j are reviews on the same product. To clearly illustrate the review graph, Figure 2 gives an example of the review graph.

Definition 1. **Fake review**: Opinion spam is first presented by Jindal[6]. In our work, fake reviews include three types of reviews: (i) untruthful opinion. Those giving undeserving positive reviews in order to promote the targeted products or giving unjust negative reviews to damage the targeted products' reputation; (ii) reviews only with empty adjectives. Those only with empty adjectives but without concrete characters even if those reviews' rating are nearly equal to average rating are judged as fake reviews. (iii) non-reviews. Those reviews are advertisements or other irrelevant reviews. For review node r_i, we use fake value $z_i = 1$ to denote that review r_i is a fake review.

Definition 2. **Review spammer**: Review spammers or fake reviewers are the main source of fake reviews. We set a fake review ratio threshold value θ, if and only if reviewer u_i fake review ratio is no less θ, u_i can be judged as a fake reviewer. For reviewer node u_i, we use faker value $y_i = 1$ to denote that reviewer u_i is a review spammer. Having defined fake review and review spammer, we are concerned with the following problem:

Problem. **Detecting fake review and review spammer in an united framework**: Given review graph $G = (V, E)$, we need to detect fake review and review spammer in an united framework, i.e. our task is to find a prediction function f:

$$f : (V, E) \rightarrow (Y, Z) \tag{1}$$

where the Y denotes the set of reviewer faker value and the Z denote the set of review fake value.

DATA COLLECTION

As mentioned before, there were no public labeled dataset for fake review and spammers before our project. We built a labeled dataset by human experts.

Opinion spam and labeling viability: There were no public gold standard labeled dataset in which reviews and reviewers all have been labeled meanwhile before our project. Research on Web[24], email[1], blogs[8], and social spam[13] and fake reviewer group[14] all rely on manually labeled dataset for detection. Because of the distinct difference from existed fake review detection, the only way to get gold standard dataset for our work, is to label data by using human expert knowledge.

Amazon dataset: In this work, we use product reviews from Amazon[6]. The reason for using this dataset is that it covers a very wide range of products. Amazon.com is also considered one of the most successful e-commerce web site which has a large number of reviews for its products and a relatively long history. We can detect fake reviews and review spammers from product reviews. Each review consists of the following 8 parts:

<member id> <product id> <date> <number of helpful feedback> <number of feedback> <rating> <title> <body>

We can generate all our factors by utilizing these 8 parts. The amazon dataset has also been used in [7, 11, 14]. In our work, we only use reivews of Electronic products in our experiments, which covers 141,501 reviewers, 195,174 reviews and 300,864 products.

Review pre-processing: Due to the huge amount of reviews and reviewers we have to label manually, several work have to be done with our Electronic-dataset before it is used[11].

Removal of duplicate products. There are some products with different product id in Amazon.com, but in our intuition they are the same product and should own nearly the same quality: (i) products with exactly the same product name and brand; (ii) products with the same product name and brand but with different color; (iii) product with the same product name and brand but with different size. In this work, we remove all these duplicate products and leave only one as the representative of all these duplicate products. Then we process all the reviews on these duplicate products as the reviews on the remained representative product.

Removal of inactive reivewers and unpopular products. In order to simplify our following labeling process,we remove inactive reviewers and unpopular products by repeating two steps iteratively: (i) remove reviewers with less than 4 reviews, then remove all these inactive reviewers' reviews; (ii) remove products with less than 4 reviews, then remove all reviews on these unpopular products.

Finally no duplicate products exist in our dataset and our dataset includes only reviewers with no fewer than 4 reviews and products with no fewer than 4 reviews. We get the preprocessed dataset with statistics shown in the Table 1 in which the $Number_1$ indicates the number before preprocessing and the $Number_2$ indicates the remained number after preprocessing.

Table 1. Dataset Statistics

	$Number_1$	$Number_2$
U: set of users	141,501	1,078
V: set of reviews	195,174	6,489
O: set of objects	300,864	851

How to Spot Fake Reviews Manually: We learned to spot fake reviews by referring ro prior work[2]. We obtained a list of spam signals, e.g. (i) left within a short period of time; (ii) have zero caveats; (iii) read like a sales ad, (iv) Multiple reviews that are exactly the same,etc. What's more, we also get a list of suggestions on how to spot fake reviews, e.g, (i) Check out the 3-star and 4-star reviews first; (ii) If you're suspicious, turn to Google; (iii) Be cautious of too many five-star reviews, e.g. All these signals and suggestions on the web and research papers are given by domain experts. For the accuracy of labeling, we reminded our labeling judges that these spam signals and suggestions can be used in their labeling procedure.

Annotator training: We employ several college students to annotate the Electronic reviews dataset. They are first asked to read all these guideline website and research papers. After learning these spam signals and suggestions on how to spot fake reviews, then they independently label the review dataset. Each review and reviewer has to be annotated by two different students independently. If a review of a reviewer get different label, it will be annotated by another two different students.

MODEL FRAMEWORK

In this section, we propose a novel Review Factor Graph(RFG) model to incorporate all the information about reviews and reviewers for better predicting reviews and reviewers fake value. We further propose our model learning algorithm.

Features

A lot of prior research[14, 6, 10, 22, 11, 15, 25] have been conducted on detecting fake reviews and fake reviewers. But to our knowledge, there are little model or algorithm can detect fake reviews and fake reviewers in an unified model. In the work, we refer to existed research papers and guideline websites on how to spot fake reviews and define features for our model. We mainly divide the features into four groups: (i) review related; (ii) reviewer related; (iii) features between reviewer and reviews; (iv) review group feature.

Review related feature

Second Person: Fake reviews are more likely to use the second personal pronouns, such as "you","your". We use the ratio of the second personal pronouns as a real number feature. If a sentence contains second personal pronouns, we judge it as a second personal pronouns sentence, we also use the ratio of second personal pronouns sentence as a real number feature.

[2]http://www.cs.uic.edu/ liub/FBS/fake-reviews.html

Length: The length of review, normalized by maximum length is extracted as a real number feature.

Positive: If a review only express positive sentiment or negative sentiment on the product, it tends to be spam. We compute the ratio of positive and negative text at the word, both are in real number.

Product Average Rating: We use the average rating, normalized by maximum rating which in our work is 5, as a real number feature.

Product Popular Rating: We make statistics on the review number of each product, normalized by the max review number, as a real number feature. The feature indicate the popular degree of each product.

Absolute Rating Difference: We compute absolute value of the difference between the review rating and the average rating of the target product, normalized by maximum rating. We use the normalized rating difference as a real number feature.

Helpful Feedback Rate and Number: We compute the helpful feedback rate and use it as a real number feature.What's more, We also use the helpful feedback number, normalized by maximum helpful feedback number, as another real number feature.

The First Product Review: We consider the post time in our work. We judge if the review is the first review for the target product. We use a binary indicator feature for this feature.

Similarity Score: We represent each review as a word vector, and select the highest cosine similarity score with other reviews as a real number feature.

Reviewer related features

Total Helpful Feedback Number and Rate: According to all the reviews written by the same reviews, we calculate the total helpful feedback rate as a real number feature, we also calculate the total helpful feedback number normalized by maximum helpful feedback number as a real number feature.

Minimum Time Interval: Fake reviewers are more likely to review multiple products in a short time window. According to all the post time written by the same reviewer, we calculate the minimum time interval of each reviewer and judge whether the minimum time interval less than empirical time interval two days. We use a binary feature for this feature.

Reviewer Rating Difference: Having calculated the difference between the review rating and the average rating of the target product, we calculate the average value of all the absolute difference values from the same reviewer, normalized by maximum average absolute difference value.

Review Number: We sum the number of all reviews the reviewer has written and normalized by the max review number. We use it as a real number feature. It can reflect the active degree of reviewer.

Features between reviewer and review

Indicator Between reviewer and review: Based on the fake value of reviewer and review, we define four binary indicator feature $I_m(u_i, r_j) = 1(m \in \{1, 2, 3, 4\})$ if and only if some condition holds, otherwise the indicator equals 0: (i) if and only if $u_i = r_j = 1$, $I_1(u_i, r_j) = 1$; (ii) if and only if the $u_i = 1, r_j = 0$,$I_2(u_i, r_j) = 1$; (iii) if and only if the $u_i = 0, r_j = 1$,$I_3(u_i, r_j) = 1$; (iv) if and only if the $u_i = 0, r_0 = 0$,$I_4(u_i, r_j) = 1$.

Table 2. Feature List

	No	Description
review local factors	0	second personal pronouns word rate
	1	second personal sentence rate
	2	review body length
	3	the number of helpful feedback
	4	the number of helpful feedback rate
	5	if the positive word rate
	6	the negative word rate
	7	the average rating of target product
	8	the popularity of target product
	9	the rating diff with average rating
	10	the first review indicator
	11	the max similarity with other reviews
reviewer local factors	12	reviewer helpful rate
	13	the sum of helpful feedback
	14	the min review time interval
	15	the sum of absolute diff with average rating
	16	the number of written review
reviewer to review factors	16	fake reviewer and fake review indicator
	17	fake reviewer and truthful review indicator
	18	truthful reviewer and fake review indicator
	19	truthful reviewer and truth review indicator

Review group features

For each target product, according to all the reviews about the product, we calculate the average rating and denote it by Avg. For each review about the product, we divide them into four subset: (i) R_1: For each review $r_i \in R_1$, $|Rating_i - Avg| > 1.5$ and review r_i is labeled as fake review,i.e. $d_i = 1$; (ii) R_2: For each review $r_i \in R_2$, $d_i = 0$ and $|Rating_i - Avg| > 1.5$; (iii) R_3: For each review $r_i \in R_3$, $|Rating_i - Avg| < 1.5$ and $d_i = 1$; (iv) R_4: For each review $r_i \in R_4$, $|Rating_i - Avg| < 1.5$ and $d_i = 0$.

Group Rating Feature: Based on the four subset, we define group rating feature as $rgf = \frac{2}{1+e^{-(|R_1|+|R_4|-|R_2|-|R_3|)}} - 1$, $rfg \in (0, 1)$. It's obvious that, for each target product, if the corresponding group rating feature value rfg is bigger, then our estimation for all these reviews about the product are more reliable.

For convenience, we summarize all defined features and denote them with corresponding symbol in Table 2

The proposed model

Factor graph assumes observation are cohesive with local features and relationships. It has been successfully applied in many applications such as social influence analysis[18], social relation mining[4, 19, 21], knowledge linking[23] and summarization[26].

In this work, we formalized our problem **Detecting fake review and fake reviewer** into a Review Factor Graph(RFG) model. To clearly illustrate the Review Factor Graph, we give

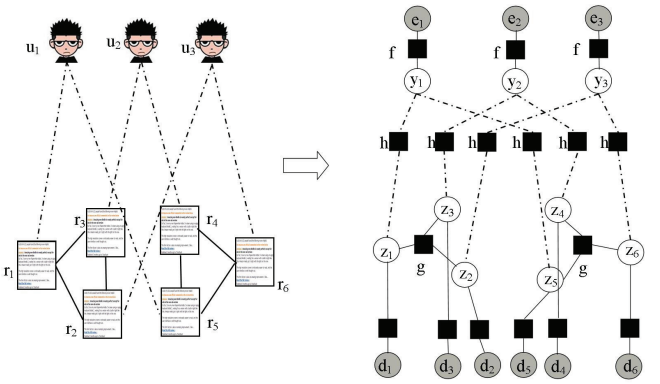


Figure 3. An example of the problem and how to transform review data into Review Factor Graph. In the right figure, each circle means variable and each diamonds means factor. Each gray circle represents local feature, i.e. e_i indicates the local feature of reviewer u_i and d_i indicates the feature of review r_i. Each white circle which means fake value is connected to local features by local factor f. All the reviews targeting the same product are connected together by group factor g. Each Reviewer fake value y_i is connected with his review fake value z_j by cross domain factor h

a example which illustrates how to transform review data into our defined Review Factor Graph model shown in Figure 3

As shown in Figure 3, based on features we have defined in 1, we define three types of factors: (i)local feature factor; (ii) group factor; (iii) cross domain factor.

Local feature factor: The probability that a review is a fake review or a reviewer is a faker can be estimated by local attributes. In this work, we use $d_i = (d_{i1}, d_{i2}, ..., d_{in})$ and $e_i = (e_{i1}, e_{i2}, ..., e_{in})$ to represent review feature vector and reviewer feature vector, respectively. To estimate the significance of each feature, we introduce a feature weight λ_c for each local feature c, then we can define the local feature factor by the local entropy formally as :

$$f_c(y_i|e_{ic}) = exp(\lambda_c \cdot e_{ic} \cdot y_i) \tag{2}$$

where the e_{ic} means the c-th local feature of reviewer u_i and the y_i means the fake value of reviewer u_i.

$$f_c(z_i|d_{ic}) = exp(\lambda_c \cdot d_{ic} \cdot z_i) \tag{3}$$

where the d_{ic} means the c-th local feature of review r_i and the z_i means the fake value of the review r_i

Review group factor: In our intuition, we can use review group rating feature to measure the degree of reliability of all rating giving to a target product. The factor take all reviews targeting the same product into consideration, rather than a special review. We can define the group factor as :

$$g(\mu, rgf) = exp(\mu \cdot rgf) \tag{4}$$

where the value rgf is the review group feature value we have defined before.

Cross domain factor: Each reviewer u_i and any his review r_j are not independent. In our intuition, a fake reviewer is more likely to post a fake review and a honest reviewer is more likely to post a veritable review. In our work, we utilize the features between reviewer and review and define the cross domain factor:

$$h_c(y_i, z_j) = exp(\nu_c \cdot I_c(y_i, z_j)), (r_j \in R(u_i)) \tag{5}$$

where ν_c is the weight of the c-th cross domain factor and I_c is the c-th binary indicator we have defined in 1.

Objective function: Finally, we can define model objective function as the normalized product of equation 2-4 for all instances. In our work, we use Θ to denote the union collection of all weight, i.e., $\Theta = \{\lambda_c\}_c \cup \{\mu\} \cup \{\nu_c\}_c$

$$p(Y, Z|E, D, \Theta) = \frac{1}{Z} \prod_{u_i \in U} \prod_{r_j \in R} \prod_{c \in C} \prod_{grf \in G} f_c(y_i|e_{ic}) \cdot f_c(z_j|d_{jc}) \cdot g(grf) \cdot h_c(y_i, z_j) \tag{6}$$

where Z is the normalized value, which sums up the conditional likelihood $P(Y, Z|E, D, \Theta)$ over all the possible labels of all the instances

Model learning and inference

Model learning: Learning review factor graph model is to estimate a parameter configuration Θ, so that the log-likelihood of observation value are maximized. For simplicity, we use $y_i\overrightarrow{e_i}$ to represent $y_i\overrightarrow{e_i} = (e_{i1} \cdot y_i, e_{i2} \cdot y_i, ...)_{i \in N_u}$, $z_i\overrightarrow{d_i}$ to represent $z_i\overrightarrow{d_i} = (d_{i1} \cdot z_i, d_{i2} \cdot z_i, ...)_{i \in N_r}$ and $\overrightarrow{I(i,j)} = (I_1(y_i, z_j), I_2(u_i, z_j)...)$. We use the Then the conditional likelihood $P(Y, Z|E, D, \Theta)$ defined in Eq.6 can be written as:

$$\begin{aligned} p =& \frac{1}{Z} \prod_{u_i \in U} exp\{\overrightarrow{\lambda} \cdot (y_i\overrightarrow{e_i})^T\} \prod_{r_j \in R} exp\{\overrightarrow{\lambda} \cdot (z_i\overrightarrow{d_i})^T\} \\ & \cdot \prod_{g \in G} exp\{\mu \cdot grf\} \prod_{r_j \in R(u_i)} exp\{\overrightarrow{\nu} \cdot \overrightarrow{I(i,j)}^T\} \\ =& \frac{1}{Z} exp\{\overrightarrow{\lambda}_u \sum_{u_i \in U} (y_i\overrightarrow{e_i})^T + \overrightarrow{\lambda}_r \sum_{r_j \in R} (z_i\overrightarrow{d_i})^T \\ & + \mu \cdot \sum_{grf \in G} (grf) + \overrightarrow{\nu} \sum_{r_j \in R(u_i)} (\overrightarrow{I(i,j)}^T)\} \\ =& \frac{1}{Z} exp\{\overrightarrow{\lambda}_u \mathbf{S}_u + \overrightarrow{\lambda}_r \mathbf{S}_r + \mu \mathbf{S}_g + \overrightarrow{\nu} \mathbf{S}_{I(i,j)}\} \end{aligned} \tag{7}$$

where $\mathbf{S}_u$ is the aggregation vector of local factor unweighted features times labeled value over all reviewer(user) nodes, $\mathbf{S}_r$ is the aggregation vector of local factor unweighted features over all review nodes, $\mathbf{S}_g$ is the aggregation vector of all review group factor unweighted features over all review groups

and $\mathbf{S}_{I(i,j)}$ is the aggregation vector of all cross domain factor unweighted features over all reivewer to review links.

To learn our RFG model, based on Eq.7 we can define the following log-likehood object function $\mathcal{O}(\Theta)$:

$$\begin{aligned}\mathcal{O}(\Theta) &= \log(p) \\ &= \log(\frac{1}{Z} exp\{\overrightarrow{\lambda}_u \mathbf{S}_u + \overrightarrow{\lambda}_r \mathbf{S}_r + \mu \mathbf{S}_g + \overrightarrow{\nu} \mathbf{S}_{I(i,j)}\}) \\ &= \overrightarrow{\lambda}_u \mathbf{S}_u + \overrightarrow{\lambda}_r \mathbf{S}_r + \mu \mathbf{S}_g + \overrightarrow{\nu} \mathbf{S}_{I(i,j)} - \log(Z) \\ &= \overrightarrow{\lambda}_u \mathbf{S}_u + \overrightarrow{\lambda}_r \mathbf{S}_r + \mu \mathbf{S}_g + \overrightarrow{\nu} \mathbf{S}_{I(i,j)} \\ &\quad - \log(\sum_{\mathbf{Y},\mathbf{Z}} \overrightarrow{\lambda}_u \mathbf{S}_u + \overrightarrow{\lambda}_r \mathbf{S}_r + \mu \mathbf{S}_g + \overrightarrow{\nu} \mathbf{S}_{I(i,j)})\end{aligned} \tag{8}$$

To solve the problem, in our work, we use L-BFGS[12] which is a quasi-Newton method for solving non-linear optimization problem. Thus, we can calculate the gradient for each parameter θ by the following function:

$$\frac{\mathcal{O}(\Theta)}{\partial \theta} = \mathbf{S}_\theta - \mathbf{E}_{\mathbf{Y},\mathbf{Z}}(\mathbf{S}_\theta) \tag{9}$$

where $\mathbf{S}_\theta$ represents the sum of unweighted features times labeled data over all nodes on the dimensionality θ, and $\mathbf{E}_{\mathbf{Y},\mathbf{Z}}(\mathbf{S}_\theta)$ means the expectation of the sum of unweighted features times labeled data over all nodes on dimensionality θ. A main challenge here is to calculate the expectation $\mathbf{E}_{\mathbf{Y},\mathbf{Z}}(\mathbf{S}_\theta)$. In our work, we utilize the Loopy belief Propagation (LBP) algorithm to estimate the approximate marginal probability over each unobservable node $p(u_i|\Theta)$ or $p(d_i|\Theta)$. Having acquired these approximate marginal probability, we can get $\mathbf{E}_{\mathbf{Y},\mathbf{Z}}(\mathbf{S}_\theta)$ by summing the expectation of unweighted features times estimated label over all nodes.

Model inference: With the learned parameter $\Theta = \{\lambda_c\}_c \cup \{\mu\} \cup \{\nu_c\}_c$, given test dataset, we can detect fake reviewers and fake reviews. Model inference is to predict the label of each reviewer and review by finding a label configuration which maximizes the joint probability as shown in Eq. 6, i.e.:

$$(Y, Z)^* = \arg\max p(Y, Z|E, D, \Theta) \tag{10}$$

In our work, we utilize the LBP algorithm to achieve an near-optimal solution. The algorithm contains multiple iterations for updating beliefs. Each iteration contains two stages: (i) all variable nodes transmit belief to their related neighbor factor nodes simultaneously; (ii) all factor nodes transmit belief to their related neighbor variable nodes. In our work, we denote the messages for delivering belief between variable nodes and factor nodes by $m_{x\to f}$ and $m_{f\to x}$. The symbol $m_{x\to f}$ represents the message sent to factor node from variable node and the symbol $m_{f\to x}$ represents the message sent to variable from factor node. The message can be formulated as following:

$$m_{x\to f} = \sum_{h\in N(x)} m_{h\to x} \tag{11}$$

$$m_{f\to x} = \sum_{\sim\{x\}} \{f + \sum_{x'\in n(f)\setminus\{x\}} m_{x'\to f}\} \tag{12}$$

where the notation $\sum_{h\in N(x)} m_{h\to x}$ denote the sum of all belief having received from all neighbour factor nodes and the notation $\sum_{x'\in n(f)\setminus\{x\}} m_{x'\to f}\}$ denote the sum of all belief having received from all neighbour variable nodes except for x. Interested reader please refer to [9] for details of message propagation.

Algorithm 1 Detailed algorithm for inference

Require:
All local attributes set on review D and all local attributes set on reviewer U; The set of review to review relation H; The set of review group relation G_r; The set of feature weight parameter Θ and the number of iteractions $Iter$

Ensure:
All review fake value Z and reviewer fake value Y

// update messages
for $i \leftarrow 1$ *to* $Iter$ **do**
  update message $m_{x\to f}$ according to Eq.11;
  update message $m_{f\to x}$ according to Eq.12;
end for
// obtain reviewer label
calculate y_i for each article according to Eq.13
// obtain review label
calculate z_i for each editor according to Eq.13

By performing these two stages in each iteration until our RFG model arrive at the global convergence state. Then we can get the approximate global optimal configuration label for each node. We can get label for each node by Eq.13

$$y_i, z_i = \begin{cases} 1 & if\ m_{x\to f} + m_{f\to x} > 0\ for\ some\ f \\ 0 & otherwise \end{cases} \tag{13}$$

Details on inference procedure is given in Algorithm 1.

Algorithm complexity: We use I to denote the number of iteration, then the algorithm computational complexity is proportional to $I \cdot (E_u + E_r)$, where the E_u and E_r denote the number of users(reviewers), the number of reviews respectively.

EXPERIMENT

Experiment setup

Dataset: We conduct experiments on our Amazon electronic reviews. The details on the procedure of preprocessing and labeling have been described in Section 1. The prepared dataset consist of 6,489 reviews written by 1,078 reviewers.

Baseline methods: We define several methods as baseline methods to compare with our approach, including Support

Vector Machine(SVM), Logistic Regression Classifiers(LR), Conditional Random Field(CRF). Details on how we use these methods has been given below:

- **SVM**: Given all local features on reviews. Based on our labeled review, we can represent each review with an attribute vector and train a SVM classification model. With learned model, we can perform inference on test data. It is the same with reviewers.
- **LR**: We train LR models on reviews and reviewers, respectively. Then we use our trained review LR classifier to infer review label and use our trained reviewer LR classifier to infer reviewer label.
- **CRF**: As we have defined a review group feature, we also train a review CRF model. We use our learned CRF model to infer review label and compare performance with our approach.

Experiment results and analysis

Compare performance: As our dataset which has been annotated manually is so limited, we use the five-fold cross validation to compare our method with other baseline methods. We calculate the average F_1 and Accuracy according to five predicting results. The performance results are shown in Table 3 and 4 and the best performance have been highlighted in bold. Need to note that, because in our dataset there exist no relation or feature between reviewers, we don't need to conduct experiment on reviewer dataset by utilizing CRF model. In this context, CRF model degenerate to LR model.

From Table 3 and 4, it can be easily found that by introducing features between reviewers and reviews, we unify the problem detecting fake reviewers and fake reviews into one united framework. What's more, owning to the factors between reviewers and reviews, each reviewer can transmit message to all reviews written by him/her to help label these reviews, each review can also transmit message to its reviewer to help label the reviewer. With these mutual reinforcement features between reviewers and reviews, in estimating reviews procedure, our model achieve 4.73% improvement measured by accuracy and 8.19% improvement measured by F_1, in estimating reviewers procedure, our model achieve 1.5% improvement measured by accuracy and 6.72% improvement measured by F_1;

Table 3. Experiment results comparison on review dataset

	F_1	**Accuracy**
SVM	0.3512	0.8437
LR	0.3522	0.8319
CRF	0.3663	0.8510
RFG	**0.3961**	**0.8820**

Table 4. Experiment results comparison on reviewer dataset

	F_1	**Accuracy**
SVM	0.4732	0.8873
LR	0.4810	0.8879
RFG	**0.5133**	**0.9013**

Factor contribution and analysis: We further analyse the contribution of each factor. By model learning, we can get the parameter configuration, i.e., the estimated weight value for each factor. We have shown the weights of each review local feature factor, i.e., λ_0, λ_1,...,λ_{11} in Figure 4. We also show the weights of each reviewer local feature factor, i.e., λ_{12},λ_{13},...λ_{15}, and the weights of each reviewer-review feature factor, i.e., ν_{16}, ν_{17},...,ν_{19} in Figure 5 and 6 respectively.

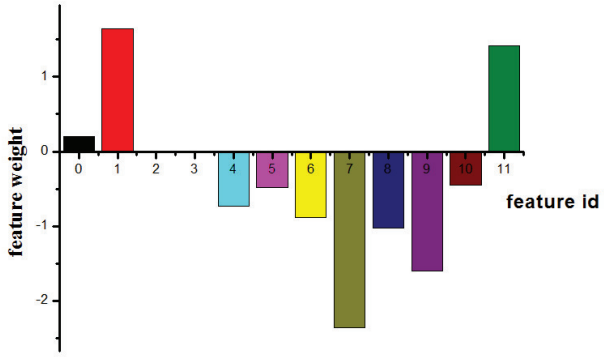


Figure 4. Results of parameters estimation on review local feature.

As all attributes for review have been normalized, from Figure 4, we can find that the feature 1 (second personal pronoun sentence rate), the feature 7(the average rating of product), the feature 9(the rating difference with average rating) and the feature 11(the max similarity with other reviews) are the most important features to identify fake review. It's easy to explain with our intuition, reviews with large rate of sentence containing personal pronoun, are more likely to spam; Spam reviews may have large difference with average rating and are more likely to have similar review content with other review.

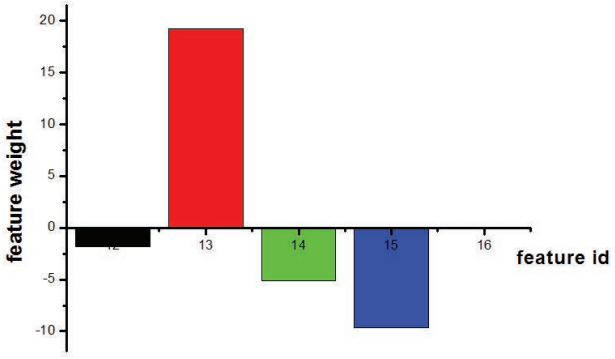


Figure 5. Results of parameters estimation on reviewer local feature

From Figure 5, we can find that the feature 13(the sum of helpful feedback) is the most important feature to identify faker, while the feature 16(the number of reviews have been written by the reviewer) is the least important feature.

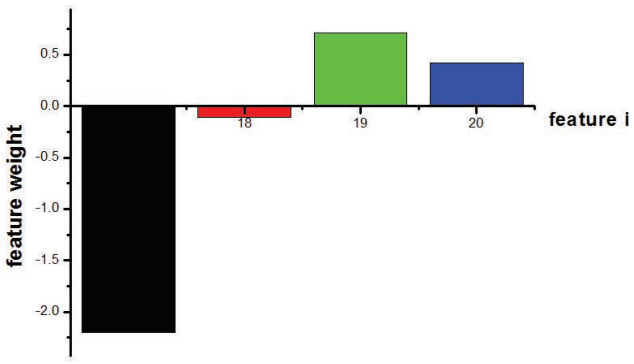


Figure 6. Results of parameters estimation between reviewers and reviews

From Figure 6, we can find that the feature 17(the reviewer is fake and the review is fake) has the largest weight, which means if a reviewer u_i is a fake reviewer, his reviews are more likely to be spam. Conversely, if a review r_i is a fake review, the author of the review are more likely to be faker.

RELATED WORK

Spam has been widely studied in the contexts of email and Web. With the development of web especially E-Commerce, opinion spam[6] has also attract lots of researches.

[6] first present the problem, detecting fake review, which is called opinion spam. In their work, duplicate or similar reviews were viewed as opinion spam and non-duplicate opinions were viewed as truthful reviews.They further train their models by using features based on the review text, reviewer and product. A distortion based method was also proposed in [25] in the absence of gold-standard data. Both of these methods use heuristic evaluation, while we generate gold-standard labeled by manual.

[27] collected 40 truthful and 42 deceptive hotel reviews and manually compared the psychologically relevant linguistic differences between them. [15] employed standard word and part-of-speech(POS) n-gram features for supervised learning on a dataset comprised of 400 truthful and 400 deceptive review. In our work, we annotated a much larger dataset comprised of 6489 Amazon reviews.

[10] utilize two views of review: features about reviews and features about reviewers, to train their model by using co-training algorithm. [22] used an unsupervised method to learn a graph-based model.

On the other hand, [11] detect review spammer by utilizing rating behaviors and [14] detect group spammers by using group features based on review text, rating and some other metadatas.

There are some key difference from previous works need to point out: (i) To the best of our knowledge, our labeled dataset is much larger than previous golden-standard dataset; (ii) We solve our problem in factor graph model and train our model by message propagation algorithm, to the best of our knowledge, little work in literature has tried to use probability graph model to detect fake reviews and fakers; (iii) In our model, by introducing several mutual reinforcement features between reviewers and reviews, we can detect fake reviews and fakers in the united framework meanwhile, what's, more, our model achieve significant improvement than other baseline methods.

CONCLUSION

In this paper, we propose a new problem, predicting fake reviews and review spammers simultaneously. With Amazon dataset, we employ several college students to label fake reviews manually. We define a large number of features to describe reviews and reviewers respectively. Distinct from previous work, we also define a set of features between reviewers and reviews. To incorporate all features we have defined, we propose a Review Feature Graph model. We further design an efficient max-sum algorithm which utilize belief propagation to perform model learning and inference. We also conduct experiments to compare our method with other baseline methods.

By leveraging the belief propagation between reviewers and reviews, the belief of reviewer and the belief of reviews can propagate to and influence each other. With the mutual enhancement effect, our approach achieve significant improvement over all other baseline methods. On average, other baseline methods can estimate the label of reviews with 0.8422 accuracy and reviewers with 0.8876 accuracy. Our approach result can achieve 0.8820 (improved by 4.73%) accuracy for reviews and 0.9013 (improved by 1.5%) accuracy for reviewers respectively. With respect to F_1 measure, our approach can improve by 8.19% on review dataset and 6.72% on reviewer dataset at least.

Finally based on our learned weights, we analyse the contribution of each factor to detect fake reviews and review spammers.

ACKNOWLEDGEMENTS

The paper was finished under grant support of Upgrading Projects of Shenzhen Key Laboratories under grants CXB201005260071A and CXB201104220042A.

REFERENCES

1. P. Chirita, J. Diederich, and W. Nejdl. Mailrank: using ranking for spam detection. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 373–380. ACM, 2005.

2. K. Dave, S. Lawrence, and D. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, pages 519–528. ACM, 2003.

3. E. Gilbert and K. Karahalios. Understanding deja reviewers. In *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, pages 225–228. ACM, 2010.

4. J. Hopcroft, T. Lou, and J. Tang. Who will follow you back?: reciprocal relationship prediction. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1137–1146. ACM, 2011.
5. M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.
6. N. Jindal and B. Liu. Opinion spam and analysis. In *Proceedings of the international conference on Web search and web data mining*, pages 219–230, 2008.
7. N. Jindal, B. Liu, and E. Lim. Finding unusual review patterns using unexpected rules. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1549–1552. ACM, 2010.
8. P. Kolari, A. Java, T. Finin, T. Oates, and A. Joshi. Detecting spam blogs: A machine learning approach. In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, volume 21, page 1351. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.
9. F. Kschischang, B. Frey, and H. Loeliger. Factor graphs and the sum-product algorithm. *Information Theory, IEEE Transactions on*, 47(2):498–519, 2001.
10. F. Li, M. Huang, Y. Yang, and X. Zhu. Learning to identify review spam. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, pages 2488–2493. AAAI Press, 2011.
11. E. Lim, V. Nguyen, N. Jindal, B. Liu, and H. Lauw. Detecting product review spammers using rating behaviors. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 939–948. ACM, 2010.
12. D. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.
13. B. Markines, C. Cattuto, and F. Menczer. Social spam detection. In *Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web*, pages 41–48. ACM, 2009.
14. A. Mukherjee, B. Liu, and N. Glance. Spotting fake reviewer groups in consumer reviews. In *Proceedings of the 21st international conference on World Wide Web*, pages 191–200. ACM, 2012.
15. M. Ott, Y. Choi, C. Cardie, and J. Hancock. Finding deceptive opinion spam by any stretch of the imagination. *arXiv preprint arXiv:1107.4557*, 2011.
16. B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.
17. A. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 339–346. Association for Computational Linguistics, 2005.
18. J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 807–816. ACM, 2009.
19. W. Tang, H. Zhuang, and J. Tang. Learning to infer social ties in large networks. *Machine Learning and Knowledge Discovery in Databases*, pages 381–397, 2011.
20. P. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 417–424. Association for Computational Linguistics, 2002.
21. C. Wang, J. Han, Y. Jia, J. Tang, D. Zhang, Y. Yu, and J. Guo. Mining advisor-advisee relationships from research publication networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 203–212. ACM, 2010.
22. G. Wang, S. Xie, B. Liu, and P. Yu. Review graph based online store review spammer detection. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 1242–1247. IEEE, 2011.
23. Z. Wang, J. Li, Z. Wang, and J. Tang. Cross-lingual knowledge linking across wiki knowledge bases. In *Proceedings of the 21st international conference on World Wide Web*, pages 459–468. ACM, 2012.
24. B. Wu, V. Goel, and B. Davison. Topical trustrank: Using topicality to combat web spam. In *Proceedings of the 15th international conference on World Wide Web*, pages 63–72. ACM, 2006.
25. G. Wu, D. Greene, B. Smyth, and P. Cunningham. Distortion as a validation criterion in the identification of suspicious reviews. In *Proceedings of the First Workshop on Social Media Analytics*, pages 10–13. ACM, 2010.
26. Z. Yang, K. Cai, J. Tang, L. Zhang, Z. Su, and J. Li. Social context summarization. In *Proceedings of the 34th ACM SIGIR Conference*, 2011.
27. K. Yoo and U. Gretzel. Comparison of deceptive and truthful travel reviews. *Information and communication technologies in tourism 2009*, pages 37–47, 2009.