Online Anti-Opinion Spam: Spotting Fake Reviews from the Review Sequence

[1]Pankaj Chaudhary, [2]Rajat Shahni

[1]Joint Director & Dean (Academic), [2] Software Engineer,

[1]JB Institute of Technology, Dehradun, [2] Contour Software Systems Pvt. Ltd, Noida

Abstract-Detecting review spam is important for current e-commerce applications. However, the posted order of review has been neglected by the former work. In this paper, we explore the issue on fake review detection in review sequence, which is crucial for implementing online anti-opinion spam. We analyze the characteristics of fake reviews firstly.

Based on review contents and reviewer behaviours, six time sensitive features are proposed to highlight the fake reviews. And then, we devise supervised solutions and a threshold-based solution to spot the fake reviews as early as possible. The experimental results show that our methods can identify the fake reviews orderly with high precision and recall.

I. INTRODUCTION

More and more users prefer to post reviews on products and services for sharing their opinions and experiences in the eBusiness web sites, such as Amazon. Many potential consumers would make their purchase decisions based on these online reviews. This motivates some unscrupulous merchants to apply fake reviews on misleading the potential consumers by enhancing their reputation or diminishing the competitors'. Thus, it brings an urgent demand to detect fake reviews as early as possible for reducing their influence.

The prior works on spotting review spam can be divided into three groups roughly based on the detected targets : review spam (e.g. [1], [2], [4], [5]), review spammer (e.g. [3], [6]) and the group of spammers (e.g. [7]). These works have pushed the anti-opinion spam forward, but they have ignored the review orders. However, identifying the review spam according to their presence orders is very important for the target of online opinion spam detection. Therefore, we focus on the problem of identifying fake reviews from the review sequences.

Our main idea is to highlight the fake reviews in the review sequences with the time sensitive features firstly based on the review contents and the reviewer behaviors. Secondly, we devise a supervised solution and a threshold based one to detect the fake reviews in the review sequences separately.

II. HIGHLIGHTING THE FAKE REVIEWS

In this section, we propose six features updated dynamically to highlight the review spam.

A. Modeling the review contents

Let R = {r[l], r[2], ... , r[n]} be a review sequence, and number indicates the posted order of review, the review r[i] contains multiple information: reviewer ID r[i].u, posted time r[i].t, review content r[i].c and product ID r[i].p.

• **Personal content similarity** (F 1)

If the reviewer r[i].u posts hislher own reviews repeatedly, r [i]. c would has a relative high similarity with his/her reviews. We maintain a review centroid for each reviewer, which consists of the terms' average occurrence frequencies in the reviews posted by r[i].u. Thus, we can evaluate the personal content similarity of the detected review as follows.

$$Su = similarity(r[i].c, Cr[ij.u) \quad (1)$$

where similarity is the similarity function likes cosine, Cr[ij.u is the review content centroid of r[i].u. After review r[i] is detected, the centroid Cr[iju is updated immediately.

• **Similarity with reviews on a product** (F2)

A fake review might be the duplicate or near-duplicate of an existing one on the same product. If so, it would be closely related with the product. If there are multiple reviews between the fake review and the normal ones copied, readers always can not identify the fake review because most of them prefer to read the reviews in the first few pages.

Compared with the normal reviews, fake reviews would have higher similarity with the "review centroid" of the product. Thus, we can calculate the similarity of the detected review with those on the same product as follows.

$$Sp = similarity(r[i].c, Cr[ij.p) \quad (2)$$

where Cr[ij.p is the centroid of reviews on product r[i].p, which is similar to the Cr[ij.u' The review centroid of a product would be updated after a review is processed.

• **Similarity with reviews on other products** (F3)

It is thorny to identify whether r[i] is a duplicate or near duplicate of r[j] for all reviews on different products and j < i, when i is a large number, r[i].p i- r[j].p and r[i].u i-r [j]. u.

Firstly, it is impractical to calculate the similarity of r[i].c with each r[j].c, since the total count of review pairs would be very large. Secondly, if we apply the methods like Fl and F2, the discriminating components of centroid would tend to be 0 because of so many reviews. Thus, we propose a solution for this case based on min hashing.

Given a string collection S = {Ol,' .. , Om} and a hash function h, the minhashing value of S is defined as follows.

mh(S) = argmoEiSn (h(o)) where the hash function h is used to permutate the elements in S randomly.

For a set of hash functions generated randomly, if each corresponding min hasing value of review r[i].c is equal to that of review r[j].c, r[i].c would be a duplicate of r[j].c with relatively high probability. Therefore, min hasing can measure the probability that the r[i].c contains the repeated contents in the existing reviews. By this measurement, we can determine whether r[i].c is a duplicate/near-duplicate of the existing one or not in a probabilistic sense.

Firstly, we generate d hash functions {hI, h2,' " , hd} randomly. And then, we calculate the minhashing value for each hash function, and use these minhashing values to construct a hash signature for each review content (r[i].c), namely,

Sig(r[i].c) = H(mhl (r[i].c), mh2(r[i].c), ... , mhd(r[i].c)) where H is a message-digest algorithm, which can generate a unique signature for a set of minhasing values.

The signatures of all reviews are maintained in a set. If the signature of r[i].c already exists in this set, r[i].c would be a duplicate or near-duplicate with very high probability. Otherwise, we generate multiple signatures for each review by repeating the processes above. By this way, we can evaluate the probability of that the near-duplicate's signature is different with the source review. Thus, The more times the signatures of r[i] are the same with those of r[j].c, r[i].c would be more likely to be a near-duplicate of r[j].c.

Formally, let hil, hi2,' .. , hid (i = 1,2,⋯ , b) denote b sets of hash functions. The Hl,H2,' " ,Hb denote b signature sets respectively. Thus, the probability of r[i].c be a duplicate or near-duplicate can be evaluated as follows.

So = L□=l e x ist(Sigi) b {I Sigi E Hi e x ist(Sigi) = 0 Sigi rt. Hi

B. Modeling the reviewer behaviors

Some spammers' behaviors could imply that their reviews are fake, such as a spammer posts multiple reviews in a short period of time. Thus, we can model such behaviors to highlight the fake reviews.

• **The review frequency of reviewer** (F4)

Spammers prefer to post many fake reviews in a small time window for more benefits. Thus, we can measure the probability of being a fake review by the time interval between two consecutive posted time points for one reviewer.

S _ _ Iu[pre(r[i].u),r[i] .t] uf -1 max(Iu)

where pre(r[i].u) is the latest review time of r[i].u, r[i].t is the posted time of review r[i], Iu[x, y] is the time interval between x and y, and max(Iu) is the maximum time interval among all pairs of adjacent reviews posted by r[i].u.

• **The review frequency of a product** (F5)

Multiple fake reviews occurring in a short time interval would make greater impact. If a product is c Olmnented very frequently in a small time window, it might be attacked. Of cause, this could be caused by other reasons like promotions. But we also treat it as an index of spammer's behavior.

S -1- Ip[pre(r[i].p),r[i] .t]
pf - max(Ip)

where pre(r[i].p) is the latest review time on product r[i].p, Ip[x, y] is similar to that defined in Equation 4, but x and y are the posted time poinits of two reviews on the same product respectively, max(Ip) is the maximum time interval among all pairs of adjacent reviews on product r[i].p.

• **The repeatability measure** (F6)

Spammers might comment a product repeatedly. Therefore, we make a complement for F4 and F5 with checking whether r[i].u has cOlmnented the product r[i].p or not.

Sr = {1 r[i].uEUp
o r[i].u rt. Up

where Up is the set of reviewers commented product r [i]. p.

III. THE FAKE REVIEW DETECTION METHODS

A. **The supervised detection methods**

Based on the six features proposed above, we can transform the review r[i] into a vector Xi' Given a labelled review set {(Xl,yI),' " , (Xn,Yn)}, where Yi is the X;'s label. Based on these labeled samples, we can apply Logistic regression or SVM to predict the labels of reviews.

For the Logistic regression, the parameter vector w can be evaluated by Maximum Likelihood Estimations. Thus, the Logistic regression model for fake review detection can be defined as follows.

exp(w · X) P(y = LspamlX) = 1 + exp(w . X)
P(y 1 = LnormallX) = 1 + exp(w . X)

We also predict the review labels with the SVM, which tries to find a high-dimensional separating hyperplane between two groups of data. To simplify feature analysis in later section, we restrict our evaluation to linear SVM, which learns a weight vector wand bias term b, such that a review Xj can be classified by:

f) = sign(w . Xj + b)

It is worth noting that we need to update the corresponding centroids for Fl and F2, the signature sets for F3, the maximum time intervals for F4 and F5 and the comment status of each product for F6, after each review is processed.

B. **The threshold-based detection method**

Although the trained classifiers have stronger generalization ability. In practice, The fake reviews occupies a small proportion of all reviews. Thus, it would take some time to gather a certain amount of fake reviews for training. Until that, the fake reviews would make the negative influence.

Therefore, we devise a threshold-based solution for detecting the fake reviews without labeled samples. Recalled the features F 1-F6 discussed in Section II, each of them tries to highlight the fake reviews from different perspectives.

Intuitively, if a review is a spam, the sum of the feature values of FI-F6 would tend to be close to 6, since all of them locate in [0, 1] according to their definitions. Thus, we can evaluate the spam score of a review as following:

S(alSu + a2Sp + a3So + a4Su core j + a5Spj + a6Sr) = 6
Lk=l ak

where aI, a2,' .. , a6 are the weight parameters turning the contributions of feature F 1, F2, . . . , F6 separately. Since the spam score is normalized in [0,1]' we can determine whether a review is a spam with a predetermined threshold

T: { Lnormal SCOre r[i] < T, Lr[i] = Lspam SCOrer[i] > T
random SCOrer[i] = T

where the random means predicting r[i]'s label randomly.

IV. EXPERIMENTS

A. **Dataset**

One of the main challenges encountered by fake review detection is the absence of ground-truth, because fake reviews are difficult to find, even for human readers. Thus, we treat the duplicates/near-duplicates of reviews as the fake ones. This way is applied in some prior works like [1] and [4].

We reconstruct our fake review dataset with the following processes based on Liu's review dataset I , which was crawled from Amazon:

1) Removing the inactive products (review count < 10). 1 http://www.cs.uic.edu/ liubIFBS /sentiment-analysis.html
2) Sorting the reviews of each product with the displaying model "Newest First" on Amazon.
3) Calculating the Jaccard similarity for each pair reviews based on bigrams.
4) Treating the reviews as fake review candidates, whose Jaccard similarity is greater than or equal 0.7.
5) Sorting each pairs of candidates by their posted orders. We get some order copy chains. We remark the review at the head of each chain as the normal ones, because we can not be sure whether they are review spam.

We have collected 2000 fake reviews with above process together with the corresponding normal reviews based on the Liu's dataset. These reviews are sorted by their posted orders, which constitute an ordered review dataset. The reconstructed dataset includes 6824 products, 155080 reviews, 122672 reviewers and 2000 review spam.

B. **Experimental results and discussion**

For the supervised solutions, the training samples should be chosen based on the reviews' posted orders. Thus, we can simulate the arrival orders of reviews in real-word, and avoid to use the later reviews to predict the front ones' labels.

In the first experiment, the count of training fake reviews is fix, and we compare the detection effects with different proportions of training fake reviews and normal ones by increasing the normal reviews' quantity. Namely, we use 50 fake reviews and 50 x i (2 :s; i :s; 20) normal ones for training, the remaining reviews are treated as the testing samples.

(FI (s» of Logistic regression and SVM for fake review detection. The detection effects of both methods are improved, when the normal review count is increased until it reaches 700. When we apply 50 fake reviews and 800 normal ones for training, the SVM achieves the highest FI(s) value (0.923), and the detection precision on spam (pre c isions) is 0.939, the recall of spam (re calls) is 0.908. Moreover, we can observe that the SVM is not very sensitive to the normal review count.

For the contributions of the proposed features for fake review detection, Table I shows the effect of SVM with different feature options respectively, in which 50 fake reviews and 800 normal ones are used to train. According to the order from F 1 to F6, the "1" indicates the feature in corresponding location would be used. For example, the "11 0 Ill" means the feature F3 is excluded only.

the feature Fl, F2, F3, F5 and F6 are active for spotting fake review, since we remove any one of them will lead to the declining effect. Recalled the definition of F4 in Section II, it means that the smaller time interval between the posted time points of two adjacent reviews wrote by one reviewer, the greater probability of the review be fake. But the time granularity of review is the day in our dataset, while some normal reviewers also write multiple reviews in one day.

This may be the main reason that the feature F4 does not work. If we could obtain finer time granularity, we consider the feature F4 would make its contribution to fake review detection.

We can observe that the review content is more effective than the reviewer behavior for fake review detection. We consider the main reason still may be the time granularity.

The last experiment focuses on the threshold-based method. . Clearly, if a review is fake, its spam score would be close to l. We get the highest value of Fl (s) with the T = 0.55 and the feature weight setting "212111", which is very close to the best effect of the Logistic-based method. Moreover, with the increase of T, the pre c isions is increasing and the recalls is decreasing for the same set of parameters. Thus, we can tune the T to reach various effects for different application demands.

V. CONCLUSION

In this paper, we explore the problem of identifying fake reviews from the review sequences, which consists of the ordered reviews according to their posted time points.

THE EFFECTIVENESS OF THE THRESHOLD-BASED DETECTION METHOD

Feature precisions recalls Fl(S) option accuracy Firstly, we highlight the fake reviews with six time sensitive features based on the review contents and the reviewer behaviors. Secondly, we devise the supervised solutions and a threshold-based one for spotting fake review respectively.

At last, we carry out a series of experiments on a real review set to verify the effects of the proposed methods, and analyze the utilities of the six proposed features in different cases. The experimental results show that the supervised methods can work with high precision and recall on fake reviews based on few training samples, and the threshold-based one can achieve a relatively good effect without training samples.

REFERENCES

[1] N. Jindal and B. Liu. Analyzing and detecting review spam. In ICDM 2007, pages 547-552. IEEE, 2007.

[2] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock. Finding deceptive opinion spam by any stretch of the imagination. arXiv preprint arXiv:1107.4557, 2011.

[3] E.-P. Lim, y'-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw. Detecting product review spammers using rating behaviors. In CIKM 2010, pages 939-948. ACM, 2010.

[4] N. Jindal and B. Liu. Opinion spam and analysis. In WSDM 2008, pages 219-230. ACM, 2008.

[5] S. Xie, G. Wang, S. Lin, and P. S. Yu. Review spam detection via temporal pattern discovery. In SIGKDD 2012, pages 823-83l. ACM, 2012.

[6] G. Wang, S. Xie, B. Liu, and P. S. Yu. Identify online store review spammers via social review graph. ACM Transactions on Intelligent Systems and Technology (TIST), 3(4):61, 2012.

[7] A. Mukherjee, B. Liu, and N. Glance. Spotting fake reviewer groups in consumer reviews. In WWW 2012, pages 191-200. ACM, 2012.