Towards Online Review Spam Detection

Yuming Lin[1,2], Tao Zhu[1], Xiaoling Wang[1],Jingwei Zhang[2], Aoying Zhou[1]
[1]Center for Cloud Computing and Big Data, East China Normal University, Shanghai, China
[2]Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guangxi, China
{ymlinbh, zhutaofjmlzt, gtzhjw}@gmail.com, {xlwang, ayzhou}@sei.ecnu.edu.cn

ABSTRACT

User reviews play a crucial role in Web, since many decisions are made based on them. However, review spam would misled the users, which is extremely obnoxious. In this poster, we explore the problem of online review spam detection. Firstly, we devise six features to find the spam based on the review content and reviewer behaviors. Secondly, we apply supervised methods and an unsupervised one for spotting the review spam as early as possible. Finally, we carry out intensive experiments on a real-world review set to verify the proposed methods.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering*

Keywords

review spam; online detection; review analysis

1. INTRODUCTION

More and more users prefer to post reviews for sharing their opinions in the eBusiness web sites, such as Amazon. Based on such user-generated contents, manufacturers can improve the product quality, while consumers can make purchase decisions handily. A Cone's survey[1] has reported that 80% of consumers reverse purchase decisions after reading negative reviews, and 87% affirm purchase decisions for positive ones. This motivates some merchants to post review spam for misleading the costumers. Thus, it brings an urgent demand for detecting review spam as early as possible.

Review spam are divided into three categories in [1], and the first type of spam, the reviews containing false opinion, is focused in our works, since such spam is more harmful and is difficult for customers to identify [1]. There are some prior works on spotting review spam, such as [1, 2, 3, 4], which have pushed the anti-review spam forward. But these works ignore the order of reviews, which is important for online review spam detection. Therefore, we focus on identifying review spam in the order of their presences.

Our main idea is to highlight the spam in review sequence firstly based on different features. Secondly, we apply supervised or unsupervised methods to detect the spam online separately. The former works well based on few labeled samples, the latter also achieves the fairly good effect without labeled samples in our experiments.

[1]www.conecomm.com/contentmgr/showdetails.php/id/4008

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2577293.

2. HIGHLIGHTING THE REVIEW SPAM

Spam can be highlighted based on review contents and reviewer behaviors. Review $r[i]$ contains multiple information: reviewer ID $r[i].u$, post time $r[i].t$, content $r[i].c$ and product ID $r[i].p$.

(1) Personal content similarity (F1)

If the reviewer $r[i].u$ copies his/her 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 written by $r[i].u$. Thus, we can evaluate the personal content similarity:

$$S_u = similarity(r[i].c, \overline{C}_{r[i].u}) \quad (1)$$

where similarity is the function measuring text similarity likes *cosine* similarity, $\overline{C}_{r[i].u}$ is the review content centroid of $r[i].u$. After review $r[i]$ is detected, the $\overline{C}_{r[i].u}$ is updated.

(2) Similarity with reviews on a product (F2)

A review spam might be the duplicate or near-duplicate of an existing review on the target product. Thus, we evaluate the review similarity with those on the same product.

$$S_p = similarity(r[i].c, \overline{C}_{r[i].p}) \quad (2)$$

where $\overline{C}_{r[i].p}$ is the centroid of reviews on product $r[i].p$.

(3) Similarity with reviews on other products (F3)

It is thorny to identify whether $r[i].c$ is a near-duplicate among massive reviews. It is unrealistic to calculate the similarity of $r[i].c$ with each review, since the count of such review pairs is too large. Moreover, if the methods like F1 and F2 are applied, the discriminating components of centroid tend to 0. Thus, we propose a solution based on *minhashing*.

Firstly, we calculate the *minhashing* values with multiple hash functions, and use these values to construct a hash signature for each review content ($r[i].c$), namely,

$$Sig(r[i].c) = H(mh_1(r[i].c), mh_2(r[i].c), \cdots, mh_d(r[i].c))$$

where H is a message-digest algorithm, which can generate a unique signature for a set of *minhasing* values.

Let $h_{i1}, \cdots, h_{id}$ $(i = 1, \cdots, b)$ denote b sets of hash functions generating different random permutations. The $H_1, \cdots, H_b$ denote b signature sets. The probability of $r[i].c$ be a near-duplicate can

be evaluated as follows.

$$S_O = \frac{\sum_{i=1}^{b} exist(Sig_i)}{b} \quad (3)$$

$$exist(Sig_i) = \begin{cases} 1 & Sig_i \in H_i \\ 0 & Sig_i \notin H_i \end{cases}$$

(4) The review frequency of reviewer (F4)

If $r[i].t$ is very close to the previous review time of $r[i].u$, $r[i]$ maybe a review spam. Thus, we can evaluate a review's probability of being a review spam by the time interval between two consecutive post time for the same reviewer.

$$S_{uf} = 1 - \frac{I_u[pre(r[i].u), r[i].t]}{\max(I_u)} \quad (4)$$

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

(5) The reviewed frequency of a product (F5)

If a product is commented very frequently with a burst mode, it might be attacked by review spam. Of cause, such case could be caused by other reasons, such as promotions. But we also treat it as an index of spammer's behavior.

$$S_{pf} = 1 - \frac{I_p[pre(r[i].p), r[i].t]}{\max(I_p)} \quad (5)$$

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

(6) The repeatability index (F6)

Review spammer might comment one product repeatedly. We make a complement for F4 and F5 with checking whether $r[i].u$ has commented $r[i].p$ or not.

$$S_r = \begin{cases} 1 & r[i].u \in U_p \\ 0 & r[i].u \notin U_p \end{cases} \quad (6)$$

where U_p is the set of reviewers commented product $r[i]_p$.

3. DETECTION METHODS

Review spam detection can be viewed as a classification problem. Based on the above features, we can applied supervised methods on it, such as Logistic regression and SVM.

On the other hand, the proposed features try to highlight the review spam from different perspectives. Thus, we devise an unsupervised method to detect the spam:

$$Score = \frac{(a_1 S_u + a_2 S_p + a_3 S_O + a_4 S_{uf} + a_5 S_{pf} + a_6 S_r)}{\sum_{k=1}^{6} a_k} \quad (7)$$

where $a_1, \cdots, a_6$ are the weight parameters turning the contributions of feature F1, $\cdots$, F6 separately.

In Equation 7, the $Score$ is normalized in $[0, 1]$. Thus, we can detect review spam with a threshold τ, such as 0.5.

4. EXPERIMENTS

We sort the reviews in Jindal and Liu's review dataset [1] according to their orders with displaying model "Newest First" on Amazon. Similar to [1, 2], we treat the reviews with *Jaccard* similarity over 0.7 as spam, and then we collect 2000 review spam together with 155080 normal reviews.

To simulate the online detection, the ordered reviews are detected one by one in our experiments. We apply the detection $precision$ on spam, $recall$ of spam and the corresponding $F1-measure$ to evaluate the effect of proposed methods. Figure 1 shows the $F_1(spam)$ of Logistic regression and SVM on spam detection with 50 spam and variable number normal reviews for training. We can observe that SVM outperforms the Logistic regression, and SVM is not sensitive to the count of normal reviews. The detection $precision$ on spam is 0.939, the $recall$ of spam is 0.908, when SVM achieves the highest $F_1(spam)$ value (0.923).

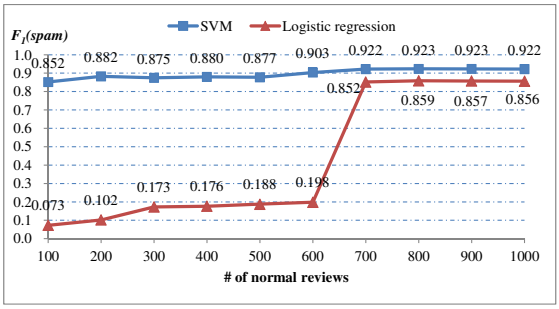

Figure 1: Spam detection with supervised methods

Table 1 shows the effect of unsupervised method, where the feature option means the weights of six features. We find that F1 and F3 play more important roles for spotting the spam. But other features are effective complements too, since any of them is omitted, the performance would decrease. Due to space limitations, we can not show all experiment results. Overall, the unsupervised method can achieve a relatively good effect without training samples.

Table 1: Spam detection with unsupervised method

spam threshold	feature option	$precision_s$	$recall_s$	$F_1(spam)$
$\tau = 0.50$	111111	0.804	0.791	0.797
	212111	0.875	0.821	0.847
$\tau = 0.55$	111111	0.822	0.755	0.787
	212111	0.916	0.796	**0.851**

5. ACKNOWLEDGMENTS

This work is supported by the 973 project (No. 2010CB328106), the NSFC grant (No. 61363005 and 61033007), the Guangxi Natural Science Foundation (No. 2013GXNSFBA019267), the general project of Guangxi Provincial Department of Education (No. 2013YB095).

6. REFERENCES

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

[2] C. Lai, K. Xu, R. Y. Lau, and L.Jing. Toward a language modeling approach for consumer review spam detection. In *ICBE 2010*, pages 241–244. ACM, 2010.

[3] 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.

[4] G. Wu, D. Greene, and P. Cunningham. Merging multiple criteria to identify suspicious reviews. In *RecSys 2010*, pages 241–244. ACM, 2010.