Analysis of Data Mining and Machine Learning Techniques to Detect Fake Reviews on Commercial Websites

CSE 543: Information Assurance and Security, Individual Project Report

Abhishek Zambre
ASU ID 1210933864
MCS, CIDSE
Ira A. Fulton Schools of Engineering
Arizona State University
abhishek.zambre@asu.edu

Abstract – **This document is an individual project report for Analysis of Data Mining and Machine Learning Techniques to Detect Fake Reviews on Commercial Websites. Most of the e-commerce websites encourage their customers to write reviews for the products sold on their websites. But some professionals are being hired to write fake reviews for some products which may either boost their popularity or may damage the existing one. This report mainly covers various techniques that are being developed to identify those fake reviews posted by fraud users. There are many methods which involve machine learning, data mining and graph based algorithms to identify such reviews. Some of them include analysis on review content, spam detection, temporal pattern discovery, network effects, rating behaviors, text mining and probabilistic language modeling. This report covers these concepts in detail.**

I. Introduction

With the advent of the Internet, people have started sharing their opinion about various things they care about. One such thing they share their opinion about is the products they purchase online. People generally rate a product from 1 to 5 and leave their opinion as text stating what they like or dislike about the product they purchased. These reviews influence how other people view a product being sold online. If it gets good reviews, people are more likely to purchase that product trusting other reviewers. However, some retailers take advantage of this trust that people put on online reviews. They unscrupulously manipulate the reviews by adding in fake reviews. These fake reviews tend to excessively glorify the product and attempt to improve the rating by giving 5 star reviews. As a result, the true rating is not reflected in the scores that the product receives online.

We want to ensure that the consumer can make a well-informed decision about the product he is about to purchase. This should not be affected by retailers who inappropriately manipulate the reviews and ratings by paying people to praise their product. The recent advances in machine learning and data mining have led to the creation of a new area called opinion mining. In this study, we analyze many such techniques which would allow us to identify and remove fraudulent reviews.

In this report, we study the review spam identification task in product review mining system. We manually build a review spam collection based on the crawled reviews. We first employ supervised learning methods and analyze the effect of different features in review spam identification. We also observe that the spammer consistently writes spam. This provides us another view to identify review spam: we can identify if the author of the review is spammer. Based on the observation, we provide a two-view semi-supervised methods to exploit the large amount of unlabeled data. The experiment results show that the two-view co-training algorithms can achieve better results than the single-view algorithm. Our designed machine learning methods achieve significant improvements as compared with the heuristic baselines.

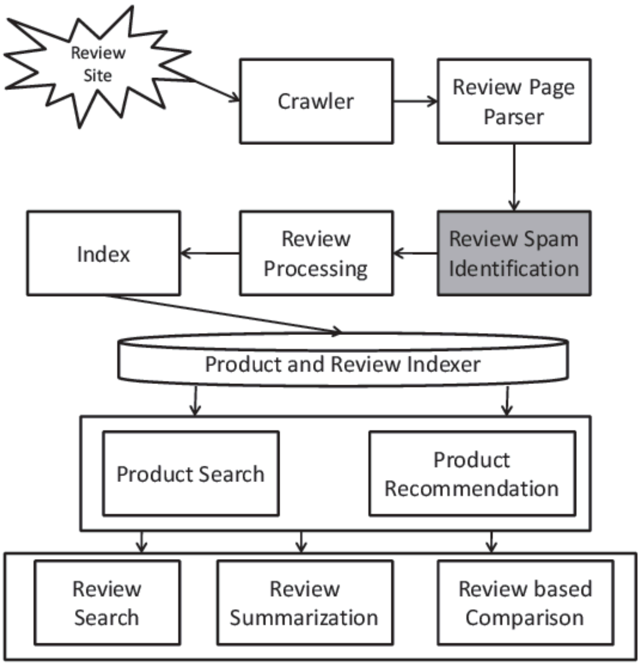
II. Overview

In today's world, the competition between ecommerce websites is at its peak. On many ecommerce sites, people write faked reviews, called review spam, to promote their products, or defame their competitors' products. It is therefore important to identify and filter out such review spams. To detect them, heuristic rules are in used today, such as helpfulness voting, or rating deviation, which limits the performance of this task. Machine learning methods can be used to identify review spam.

To implement this, first a spam collection needs to be built from crawled reviews. After this analysis of the effects of various features in spam identification needs to be performed. One important observation is that the review spammer consistently writes spam. This provides another view to identify review spam: one can identify if the author of the review is spammer. Based on this observation, a two-view semi-supervised method is designed for co-training and to exploit the large amount of unlabeled data. The experiment results show that the proposed method is effective. This designed machine learning method achieve significant improvements in comparison to the heuristic baselines.

II. System Description

The overall framework is shown in Figure 1. First the system crawls the review pages from the review site, then parses these html pages with several regular expressions, to extract the review relevant text parts. Before the review analyzer module, system needs to identify and filter out the fake reviews, called review spam, to provide the consumer real and trustful reviews. The review analyzer mainly predicts the overall sentiment and extract the topic and opinion words for each review. Finally, indexing is done on all the analyzed reviews into the indexer.



System provides several applications. First, the consumer can target their product with product search. The system can also recommend products based on the product reviews. After targeting the product, the system directs the user into the product pages. This page provides the review summary for this product based on the review topic and opinion extraction. Since the number of reviews can be very large, especially for the popular products, system also provide the review search module. The consumer can search the reviews with sentiment queries, such as “positive opinions on battery” for a camera. System also provide a product comparison module. Review spam identification is an important component in the system.

III. Review Spam Corpus Construction

To identify review spam, a review spam corpus is built manually. Part of crawled product reviews are obtained from Epinions. The data set consists of about 60k reviews. On the review sites, after the review is posted, other users can evaluate the posted review. There is a large amount of helpfulness evaluations and comments in the review sites Annotation of the review spam is done by taking advantages of these evaluations. One assumption made is that there is a relation between review helpfulness and spam, as follows: the review spam will not help the people know the product. Therefore, the low-helpful reviews contain more review spam. Low helpful reviews are then manually annotated. Based on this assumption, spam data set is built as follows:

Data Pre-processing:

1. The duplicate products are removed based on full name match. The reviews with anonymous reviewers are also removed. After that, the reviews with enough social evaluations whose number of helpfulness and comment evaluation is above 5 are selected.
2. Ranked all the left reviews (about 30k), based on their overall helpfulness, and divide them into three sets: top-helpful set, middle-helpful set, low-helpful set. 1000, 1000, and 4000 reviews are selected from the three sets separately.
3. Extracted various contexts for the review to be annotated, including all the comments and helpfulness evaluations, the target product description, and the reviewer’s profile and history reviews.

Annotator Training:

In this part, a principled way to conduct annotation is presented. In public blogs and forums, there is a great interest in identifying review spam. 10 college students are employed to annotate the review spam data set. They are first asked to read all the related articles and discussions to know what the review spam looks like. They then independently label the review data. When they determine if the review is spam or not, they are asked to carefully read the contexts provided in the data pre-processing step. Each review is labelled by two people. The conflict is resolved by the third one. Finally, in the 6000 reviews, total 1398 spam reviews are marked. The result also verifies the hypothesis: most of the spam reviews are from the low-helpful review set. System identify 1256, 112, 30 spam reviews from the low, middle and top helpful set separately.

IV. Review Spam Identification

4.1 Methods

a) Supervised Methods

With labelled review spam data set, fully supervised method to identify review spam can be designed. Several supervised methods are tested, including SVM, logistic regression, Naive Bayes, based on the public machine learning software Weka. Naive Bayes achieved best results in experiments. Naive Bayes assumes the features are conditionally independent given the review’s category.

$$P_{NB}(c|d) = \frac{P(c)\prod_{i=0}^{m} P(f_i|c)}{(P(d))}$$

Despite its simplicity and the fact that its conditional independence assumption doesn’t hold in real-world situations, Naïve Bayes-based categorization still tends to perform surprisingly well.

b) Semi-Supervised Methods

Since it is a labour-intensive task to manually label the review spam, only a small set of review data is annotated. There are still many unlabelled data, which may boost the performance. In this section, semi-supervised method is used to utilize the unlabelled reviews.

Before designing semi-supervised method, it is observed that the spammers consistently write review spam. To verify this observation, 40 spammers are randomly selected from labelled data set by removing the reviewers with low number (below 3) of reviews. For each spammer, 10 his reviews are randomly extracted. These reviews are manually labelled, which aims to check if the spammer consistently writes review spam. Among 40 spammers, 25 always write spams, 3 write about 80% spams, 6 write about 70% spams, 4 write about 40% spams, 2 write about 30% spams. On average, the spammers have about 85% possibility to write spam. This provides two views to identify the review spams: the first view is to directly detect if the review is review spam; the other view is to detect if the author of the review is spammer. If the author of the review is a spammer, this review has a very high probability to be a review spam.

Based on the above observations, a two-view semi-supervised method for review spam detection is designed. The framework of the co-training algorithm is employed. The co-training algorithm [Blum and Mitchell, 1998] is a typical bootstrapping method, which starts with a set of labelled data, and increases the amount of annotated data by adding unlabelled data incrementally. One important aspect of co-training algorithm is property of two views. The separation of two views proves to be more effective than the single view in practice and theory. In the context of review spam identification, each review has two views of features: features about review itself and features about corresponding reviewers. The overall framework algorithm is shown below.

```
Require: two views of feature sets for each review: review features Fr and reviewer features Fu; a small set of labeled reviews L; a large set of unlabeled reviews U.
Ensure: Loop for I iterations
1: Learn the first view classifier Cr from L based on review features Fr;
2: Use Cr to label reviews from U based on Fr;
3: Choose p positive and n negative most confidently predicted reviews Treview from U.
4: Learn the second view classifier Cu from L based on reviewer features Fu;
5: Use Cu to label reviews from U based on reviewer features Fu;
6: Choose p positive and n negative most confidently predicted reviews T'reviewer from U;
7: Extract the reviews T'review authored by T'reviewer;
8: Move Reviews Treview ∪ T'review from U to L with their predicted labels.
```

In practice, there are always noises in data. The assumptions of co-training, such as conditional independent views, may not hold in practice. Only the p positive instances and n negative instances are selected, when the two view classifiers agree most: T ∪ T' in Step 8 is changed to T ∩ T'.

4.2 Features

The feature engineering is a key task for review spam identification task. Various observations to identify review spam are acquired by analyzing data set and reading discussions from public blogs and forums. But transferring these observations to the features is still a challenging task. In this section, extracted features for review spam identification are introduced. The features are mainly divided into two groups. One is related with review, the other is related with reviewer.

a) Review Related Features: This type of features contains four groups: content features, sentiment features, product features and meta data features.

- Content Features:

i. Unigram and Bigram : feature selection metric X2 is used to select the text classification features: the top 100 unigrams and top 100 bigrams.

ii. Square of normalized length: a length of the review, normalized by the maximum length, is also extracted as a real number feature.

iii. First Person vs. Second Person: there is a finding that in the fake review, if sometimes says "you" should do something, rather than how "I" experienced, the ratio is used of the first personal pronouns, such as "I", "my", "we", and the second personal pronouns, such as "you", "your", as a real number feature.

iv. High Similarity Score: the spammer may just change the product name in the review, or post the same review on more than one products. Each review is represented as a word vector, and select the highest cosine similarity score with other reviews as a real number feature.

v. Other content features are extracted as follows: ratio of the question and exclamation sentences, where these sentences are identified simply by regular expressions; ratio of the capital letters.

- Sentiment Features

i. Subjective vs. Objective: if the review consists of much objective information, it may just describe the products' attributes or off topic advertisements. The ratio of subjective and objective is computed at the word and sentence level. The subjective word is identified by subjective lexicons, SentiWordNet and HowNet. If the sentence contains at least one subjective word, it is considered as subjective.

ii. Positive v.s. Negative: if the review only express positive sentiment or negative sentiment on the product, it tends to be spam. Because the real reviews will express both sides of sentiments. The ratio of positive and negative text is computed at the word and sentence level. The positive and negative sentiment are also identified by sentiment lexicons.

- Product Features

i. Product Centric Features: the number of reviews are employed under this product to denote the popularity of the product. The average rating of the product as a feature is also used.

ii. Product Description Features: it is a good indicator that how the product is described in the review. If the product name is not mentioned, this review may be an off-topic advertisement. If the brand name or product name is mentioned many times, this review may be an advocator for this product. The percent of brand and product name is calculated in all words as a real number feature.

- Meta-data Features

The meta-data features include the rating of the reviews. The difference between the review rating and the average rating of the target product is calculated. The post time is also considered. Binary feature is used to denote if the review is the first product review.

b) Reviewer related Features: All the reviewer related features are divided into two groups: Profile Features and Behavior Features.

- Profile Features

The profile features are all extracted from the reviewer profile page. It contains the reviewer id, the number of written reviews, whether contains real name, homepage, and self-descriptions, the rank of popularity in the whole site and the specific category.

- Behavior Features

i. Authority Score: on Epinions site, one reviewer can "trust" another reviewer, if the former thinks the reviews written by the latter is trustful. This is similar to the web page links. A directed reviewer graph is first constructed based on the "trust" relation. The reviewer's authority score is computed based on the link analysis algorithm PageRank.

$$PR(u_i) = \frac{1-d}{N} + d \sum_{u_j \in M(u_i)} \frac{PR(u_j)}{L(u_j)}$$

where $u_1, \ldots, u_N$ are the reviewers in the collection, N is the total number of reviewers, $M(u_i)$ is the set of reviewers that "trust" reviewer u_i, $L(u_i)$ is the number of reviewers that reviewer u_i "trust", d is a damping factor, which is set as 0.85. The PageRank score can be computed iteratively with random initial values.

ii. Brand Deviation Score: The spammer may focus on specific brands, the distribution of the review numbers is computed over different brands. Entropy is used to denote this score:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i)$$

where x_i is the i-th brand, $p(x_i)$ is the probability with the number of the i-th brand reviews divided by the total reviews.

iii. Rating Deviation Score The spammer may give different brands with different ratings. The variance is computed as a real number features:

$$Var(X) = \sum_{i=1}^{n} p(x_i)(s(x_i) - \mu)^2$$

where $s(x_i)$ is the average rating for the i-th brand, μ is the overall average rating on all the brands.

V. EXPERIMENTS

5.1 Experiment Setup

The data has been described in Section 3. For supervised methods, data set is divided into training set and test set. 10-fold cross-validation is conducted: the data set is randomly split into ten folds, where nine folds are selected for training and the tenth fold is selected for test. Co-training method is applied on the same test data set as the supervised method, for the convenience of comparison.

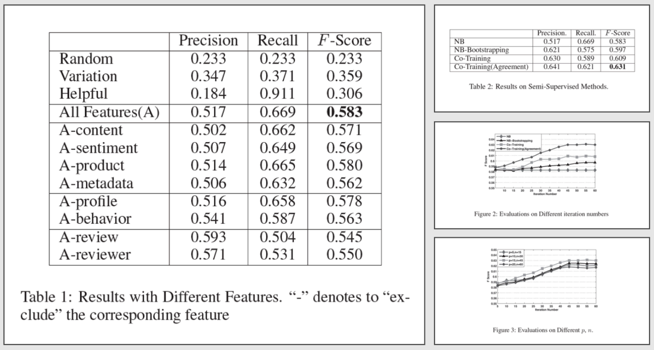
The evaluation metrics are precision ($[S_p \cap S_c]/S_p$), recall (($[S_p \cap S_c]/S_c$) and F-score ([2*precision*recall] / precision+recall), where S_c is the true review spams, and S_p is the set of predicted review spams.

5.1 Experiment Results

- Supervised Method Results: Table 1 shows the experiment results. The machine learning method Naive Bayes (NB) achieves significant improvement compared with the heuristic methods. With all features, NB can achieve the best result 0.583 in F-Score.

- Semi-Supervised Method Results: From the previous section, we have analyzed the effect of various features. In this section, we exploit co-training method to utilize the large number of unlabeled data. Table 2 shows the experiment results. NB-Bootstrapping is a bootstrapping version of NB, which uses all features as a single view.

-Parameter Sensitivity: Figure 2 shows the results for different iteration numbers. When the iteration numbers are above 40, it can achieve good results. p and n are the numbers of newly added positive and negative samples in each iteration.



	Precision	Recall	F-Score
Random	0.233	0.233	0.233
Variation	0.347	0.371	0.359
Helpful	0.184	0.911	0.306
All Features(A)	0.517	0.669	**0.583**
A-content	0.502	0.662	0.571
A-sentiment	0.507	0.649	0.569
A-product	0.514	0.665	0.580
A-metadata	0.506	0.632	0.562
A-profile	0.516	0.658	0.578
A-behavior	0.541	0.587	0.563
A-review	0.593	0.504	0.545
A-reviewer	0.571	0.531	0.550

Table 1: Results with Different Features. "-" denotes to "exclude" the corresponding feature

	Precision.	Recall.	F-Score
NB	0.517	0.669	0.583
NB-Bootstrapping	0.621	0.575	0.597
Co-Training	0.630	0.589	0.609
Co-Training(Agreement)	0.641	0.621	**0.631**

Table 2: Results on Semi-Supervised Methods.

Figure 2: Evaluations on Different iteration numbers

Figure 3: Evaluations on Different p, n.

VI. CONCLUSION

Supervised Method Results: The machine learning method Naive Bayes (NB) achieves significant improvement compared with the heuristic methods. With all features, NB can achieve the best result 0.583 in F-Score.

Semi-Supervised Method Results: From the previous section, we have analyzed the effect of various features. In this section, we exploit co-training method to utilize the large number of unlabeled data. NB-Bootstrapping is a bootstrapping version of NB, which uses all features as a single view.

Parameter Sensitivity: When the iteration numbers are above 40, it can achieve good results. p and n are the numbers of newly added positive and negative samples in each iteration.

VII. REFERENCES

[1] [Blum and Mitchell, 1998] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In COLT, pages 92–100, New York, NY, USA, 1998. ACM.

[2] [Collins and Singer, 1999] Michael Collins and Yoram Singer. Unsupervised models for named entity classification. In EMNLP-VLC, pages 100–110, 1999.

[3] [Furuse et al., 2007] Osamu Furuse, Nobuaki Hiroshima, Setsuo Yamada, and Ryoji Kataoka. Opinion sentence search engine on open-domain blog. In IJCAI, pages 2760–2765, CA, USA, 2007. Morgan Kaufmann Inc.

[4] [Hall et al., 2009] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. The WEKA data mining software: an update. SIGKDD Explorations, 11(1):10–18, 2009.

[5] [Jindal and Liu, 2008] Nitin Jindal and Bing Liu. Opinion spam and analysis. In WSDM, pages 80 219–230, New York, NY, USA, 2008. ACM.

[6] [Kim et al., 2006] Soo-Min Kim, Patrick Pantel, Tim Chklovski, and Marco Pennacchiotti. Automatically assessing review helpfulness. In EMNLP, pages 423–430, Morristown, NJ, USA, 2006. ACL.

[7] [Lewis, 1998] David D. Lewis. Naive (Bayes) at forty: The independence assumption in information retrieval. In C. Nedellec and C. Rouveirol, editors, ´ ECML, number 1398, pages 4–15, Chemnitz, DE, 1998. Springer Verlag.

[8] [Li et al., 2010] Fangtao Li, Minlie Huang, and Xiaoyan Zhu. Sentiment analysis with global topics and local dependency. In AAAI, 2010.