An Algorithm Model For Incremental Dectection of Spam Reviews

Maoan Wang, Jun Sun, Yifan Wu, and Guoshi Wu

***Abstract*—Surging smartphone use and pervasive O2O services mean customers can post their reviews about restaurants and shops online. However, many merchants may hire some people to post positive but fraud reviews in order to attract more customers. Therefore, a model need to be built to detect spam reviews. In this paper, firstly, we build a detection model using traditional batch processing which view the detection as a binary classification problem. Next, since many reviews are coming sequentially, batch processing is not efficient and useful. We will use another incremental algorithm—Hoeffding Option Tree to update the model without processing the past data repeatedly. We find that the incremental method can drastically improve the speed and the accuracy is also satisfying.**

Index Terms—Incremental classification, fraud review, hoeffding option tree.

I. Introduction

Spam reviews is a huge topic which can be categorized into several major fields—bump, advertisement, hype and etc. According to Jindal N, Liu B [1], there are generally three types of spam reviews:

1) Untruthful opinions (fraud reviews).
2) Reviews on brands only.
3) Non-reviews.

Based on the data acquired from Dianping.com (like the Yelp in the US), 331,415 reviews can be used to conduct this experiment. And the most spam reviews belong to the 1). Also, obviously, detecting the fraud reviews will be the most demanding job. Therefore, our model just focuses on the fraud reviews. The precise definition for 1) is that: those that deliberately mislead readers or opinion mining systems by giving undeserving positive reviews to some target objects in order to promote the objects (which we call hyper spam). Besides, each review will be labeled with a number that can tell us whether it belongs to fraud review.

The raw review consists of several features which include user id, shop id, review body, star, type, etc.

Firstly, we build a *Static Detection Model* which can classify the review in batch process. The model will extract several features from each review through which we can calculate a valued called *credibility* of each user and merchant. In other words, we build a credibility table based on the past records. This is called *Credibility Building Process* and is constructed mainly by using Logistic Regression algorithm. Then, by using Random Forest—a bootstrap sampling of decision trees [2], we will build the final static detection model. The detail will be discussed in Section II.

Also, there exists a drawback for the static detection model which is that reviews are not coming once and for all. Imaging for Dianping.com, many new reviews will be posted every day and many new users and merchants will show up. So the static detection model has to be updated and the whole data need to be processed over and over again. Therefore, in order to avoid repeatedly evaluating past data and to make the model be updated continually, we propose one model that uses incremental algorithm—Hoeffding Option Tree.

This model also relies on the credibility building process. And during each update, all the data must be used to create a new credibility table. This process costs time but it cannot be ameliorated by using incremental algorithm. Then, how can we improve the efficiency? For example, we already train our model by using 100,000 reviews based on their credibility table. And there comes 1000 new reviews. So we calculate their user's and merchant's credibility through previously established credibility table but, instead of training the model by using the whole 101,000 reviews, we simply update the model by processing the new 1000 reviews based on previous results. The whole process will be discussed step by step through Section III.

In the Section IV and Section IV, we will show the reader the results of our experiments which include the accuracy and efficiency. At last, we will have a brief conclusion.

II. Static Detection Model

A. Crediblity Building Process

1) Logistic regression

First, we define several important features or values that can be calculated through the original reviews. Then, by using Logistic Regression algorithm, we can obtain a number or possibility whose value ranges from 0 to 1. That is the credibility.

$$P_a = h_\theta(x_a) = g\left({\theta_a}^t x_a\right) = \frac{1}{1 + e^{-{\theta_a}^t x_a}} \tag{1}$$

Equation (1) clearly shows how the credibility is calculated. The x_a represents all the important features of a user or merchant which we will be discussed later. The $\theta_a$is comprised of the weights of related features. This vector is an estimated value obtained from the training process.

Manuscript received August 25, 2015; revised December 25, 2015. This research is supported by the Science and Technology Projects of MOT China (NO.2014364222100).

Maoan Wang, Yifan Wu, Guoshi Wu are with Beijing University of Posts and Telecommunications, Beijing, China (e-mail: 2011213174@bupt.edu.cn, pain@bupt.edu.cn, guoshiwu@bupt.edu.cn).

Jun Sun is with Information Center, China Waterborne Transport Research Institute, MOT, Beijing, China (e-mail: sunjun@wti.ac.cn).

DOI: 10.7763/IJMO.2016.V6.501

2) *Important features*

Next, we will present to you the important features. Since users and merchants have different features. We have to define them separately.

a) User important features

For users, the first feature called RD, which represents the difference between user's grade and this merchant's average grade. Then, the C value, calculated like this:

$$C_a = \sum_{p \in P_a} n_{ap}\left(1 - \frac{\sigma_{ap}}{\mu_{ap}}\right) \tag{2}$$

C_a is the C value of the user a. P_a represents all the merchants that a has ever reviewed. n_{ap} is the number of reviews that user a has commented on merchant p. σ_{ap} and μ_{ap} represents the standard deviation and mean of the scores of these reviews. This feature wants to measure the similarity of scores since some paid posters may grade one merchant several times with same score. Another important feature is called ETF.

Since so many merchants would like to open up markets at early times, they are more likely to recruit people to post fraud comments.

$$ETF_a = \max_{p \in P_a} ETF_{ap} \tag{3}$$

$$ETF_{ap} = \begin{cases} 0, & if\ L_{ap} - A_p > \beta \\ 1 - \dfrac{L_{ap} - A_p}{\beta}, & x \geq 0 \end{cases} \tag{4}$$

L_{ap} records the time that user a posted the last review to merchant p. A_p is the time that the merchant had the first review. β is a fixed value and we define it as half year in this model. At last, we want to find those users who only write reviews to a single merchant, labeled as 1. This feature is called SR. y_a represents the type of the review—fraud or true.

In summary, $x_a = (RD_a, C_a, SR_a, ETF_a)$, $h_\theta(x_a) = y_a$.

b) Merchant important features

For merchants, we use three features— SRr, AVG and RCV. First, SRr represents, to a merchant, what percentage the single review accounts for the whole reviews. As to RCV:

$$\mathrm{R}CV_p = \frac{\sigma(scores)}{\mu(scores)} \tag{5}$$

The $\sigma(scores)$ and $\mu(scores)$ means the standard deviation and mean of all scores of a merchant. AVG simply means the average of all the scores of a merchant. y_p represents the type of the review—fraud or true.Again, just like the user, $x_p = (SRr_p, \mathrm{R}CV_p, AVG)$, $h_\theta(x_p) = y_p$.

Once we have all these features calculated, we can use the review type given by Dianping.com to train the dataset and obtain related weights θ_a. And we can store all the users' and merchants' credibility value into a table. Therefore, if the new review comes, we can find its user's and merchant's credibility by searching the credibility table.

B. *Random Forest Classification*

Acquiring the credibility of both the customer and the merchant in each review is the first step. Then, by using Ansj [3] which is a tokenizer based on Chinese, we can obtain much information about each review such as the percent of adjectives, verbs and nouns or the length of the review. Each of these features will act as a factor in the final Random Forest algorithm. In our experiments, after training, we build the random forest which has 200 independent decision trees. Each tree is built by using 4 features randomly chosen in all the features.

III. Incremental Detection Model

Just like the static detection model, this incremental model also relies on the credibility building process. So, if the data come several times and the original credibility table is not enough for all the users and merchants, then, all the data must be used to re-create the credibility table. This process can't be done incrementally. And that makes sense because we don't need to update the table when only 100 reviews come. And we use Hoeffding Option Tree to replace the random forest to classify the reviews.

At first, we use x_r to represent all the features:

$$x_r = (P_a, P_p, P_{parts}, stard_r, len) \tag{6}$$

P_a, P_p represents the credibility of user and merchant separately. And P_{parts}means all the features calculated by Ansj tool which consists of the percentages of different parts of speech. len represents the total length of the review. $stard_r$ means the difference between user's score and the average score of the merchant. So, each raw review can be transferred into this format.

The whole process can be shown in the Fig. 1.

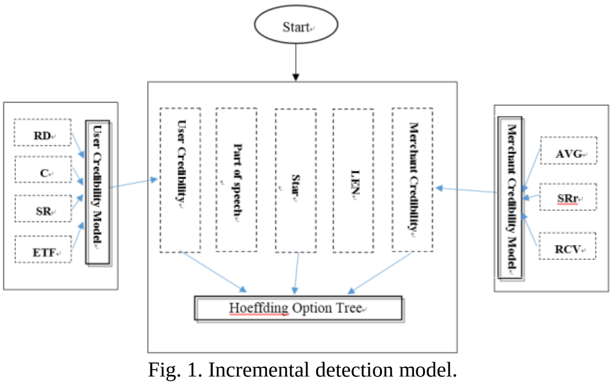

Fig. 1. Incremental detection model.

The next step is training Hoeffding Option tree by processing the data composed by Equation (6) sequentially.

A. *Hoeffding Tree*

Hoeffding trees were first introduced by Domingos and Hulten in the paper "Mining High-Speed Data Streams" [4]. And its advantage is obvious: that all the data will be inspected only once without storing them for further update.

Hoeffding tree can process stream data but the more data we use to train each time, the more accurate our tree will be.

Because the idea behind it is that we want to use the sample data to estimate the whole. Therefore, we define the value called n_{min} and the number of reviews must exceed it before any process begin.

B. The Workflow

When data formatted in Equation (6) comes, they will be sorted in related nodes. Then, when the number of data in one node is larger than n_{min}, we can update the tree.

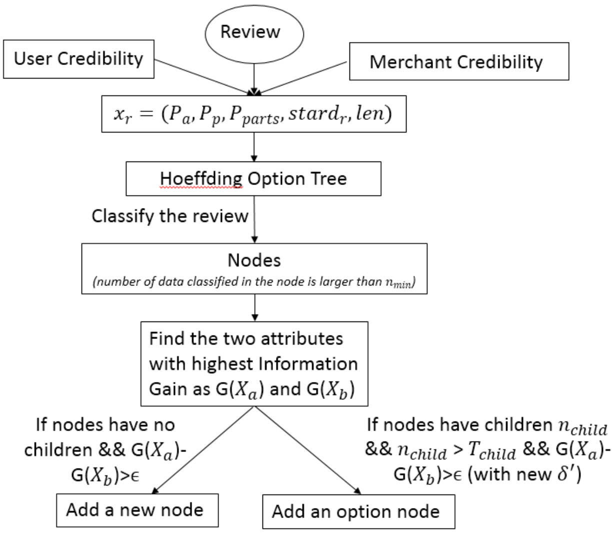
Review
User Credibility
Merchant Credibility
$x_r = (P_a, P_p, P_{parts}, stard_r, len)$
Hoeffding Option Tree
Classify the review
Nodes
(number of data classified in the node is larger than n_{min})
Find the two attributes with highest Information Gain as G(X_a) and G(X_b)
If nodes have no children && G(X_a)-G(X_b)>ϵ
If nodes have children n_{child} && n_{child} > T_{child} && G(X_a)-G(X_b)>ϵ (with new δ')
Add a new node
Add an option node

Fig. 2. Hoeffding option tree framework.

For example, new 100 reviews arrives at one node, and the n_{min} is 100. So these 100 reviews are the sample data. Each feature will be used to calculate the Information Gain which can be trusted due to the Hoeffding Bound. We will discuss it later.

If the node has no children, we will compute the information gain $\overline{G}(x_i)$ for each attribute i and find the highest and second-highest value X_a , X_b. Then we compute the Hoeffding bound ϵ , if the difference between X_a and $X_b$is larger than ϵ, a new node will be added that split on attribute a.

Otherwise, if the node has children but the number of option nodes is less than the defined maximum value—T_{child}, we can choose another attribute to split on this node. Let S be existing child split with highest $\overline{G}$, X be (non-used) attribute with highest $\overline{G}$. Computing a new Hoeffding bound ϵ with a much bigger probability δ . If the difference between X and S is larger than ϵ, then an addition node will be added that split on X.

In this way, we find the tree has been updated just using the 100 reviews and some statistics about calculating the Information Gain will be updated too. The whole process is shown in Fig 2.

C. Why It Works?

1) The obstacles for decision trees

We know that split criteria is essential to traditional decision tree. It chooses the feature we split on one node. For example, the information gain is one of the methods which is popularized by Quinlan [5]. For a dataset D, the expected information needed to classify or entropy is given by

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i) \tag{7}$$

Ref. [6] Where p_i is the nonzero probability that an arbitrary tuple in D belongs to class C_i. It represents the average amount of information needed to identify the class label of a tuple in D

As you can see, since we don't have all the data every time, so this method can't be used directly. In each update, we simply have the sample data. So how can we estimate the population by using each sample? This is solved by Hoeffding Bound

2) Hoeffding bound

The Hoeffding bound states that with probability $1-\delta$, the true mean of a random variable of range R will not differ from the estimated mean after n independent observations by more than:

$$\epsilon = \sqrt{\frac{R^2 \ln\frac{1}{\delta}}{2n}} \tag{8}$$

The ϵ represents the difference in information gain between splitting on the best and second best attributes. For example, if the difference in gain between the best two attributes is estimated to be 0.3, and ϵ is computed to be 0.1, then the bound guarantees that the maximum possible change in difference will be 0.1 [7].

3) Hoeffding option tree

As to the Hoeffding Option tree, it is an ensemble method of Hoeffding trees. Option trees were introduced by [8] Buntine (1992a) as a generalization of decision trees. Then, [9] Ron Kohavi and Clayton Kunz proposed a new option decision trees with majority votes.

Simply speaking, instead of using just a single node, this method creates a set of option nodes so there will be several results to one example. And we choose the ultimate result by combining the predictions of all the option nodes.

Apart from the Hoeffding bound ϵ, it will construct a new confidence δ' that can be used to calculate a new Hoeffding bound. And this new bound is responsible for building extra option nodes. This δ' needs to be looser than the original δ so additional attribute choices can be made.

IV. Experiments

A. Model Performance

The tests have been conducted on both models. The total number of reviews used are 331,415. For static detection model, we use 70% of the data to train the model, and exploit the rest 30% to test the model. The result is shown in Table I.

As to the Incremental Detection Model, we also use the 70% of the data to train. The rest 30% of the data is using to test the model. And, since this model includes two phases—building credibility table and building Hoeffding Option tree, we decide to make the reviews needed to build credibility table be fixed at 40% percent and the rest 30% will be processed sequentially. This means we will change the number of reviews processed for each update in each experiment which represents the value of n_{min}. These reviews will not be inspected again. The result is presented in Table II.

TABLE I: STATIC DETECTION MODEL PERFORMANCE

	ACCURACY	TIME(*SECONDS*)
1	74.862%	2820
2	75.683%	2840
3	75.451%	2880
AVERAGE	75.332%	2847.7

TABLE II: INCREMENTAL DETECTION MODEL PERFORMANCE

	n_{min}	FINAL TREE SIZE(LEAVES)	ACCURACY	TIME(*SECONDES*)
1	200 reviews	474	82.59%	4.56
2	2000 reviews	237	82.416%	2.78
3	10000 reviews	88	80.235%	3.06

Besides, the “TIME” column only records the time required to complete the Random Forest or Hoefding Option Tree since the rest steps are the same.

B. Result Analysis

First, Incremental Detection Model is obviously more efficient. We can’t compare them directly since random forest consists of 200 decision trees in our model. But, it still takes nearly 14.24 seconds to build each decision tree so the Hoeffding option tree is still faster.

Second, we can infer from their performances that incremental detection model predicts more accurate than the static detection model. It may seem strange that Incremental Detection Model can trump Static Detection Model in both accuracy and efficiency. But according to the study of Ron Kohavi [9], the Option Decision trees can outperform the bagging method of traditional decision trees in some circumstances.

V. CONCLUSION

This paper introduces a new way of detecting fraud reviews which includes several significant features of users and merchants. And we build the credibility table for both of them. Then, we adopted the Hoeffding Option trees to act as the classifier. With this model, we can detect the fraud reviews sequentially. And we demonstrated that it has satisfying accuracy and remarkable efficiency. We are confident that this model can put into practical use.

REFERENCES

[1] N. Jindal and B. Liu, "Opinion spam and analysis," in *Proc. the International Conference on Web Search and Web Data Mining*, 2008, pp. 219-230.

[2] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Monterey, 1984.

[3] J. Sun. Chinese word segmentation. [Online]. Available: https://github.com/NLPchina/ansj_seg/

[4] P. Domingos and G. Hulten, "Mining high-speed data streams," in *Proc. the Sixth International Conference on Knowledge Discovery and Data Mining*, 2000.

[5] J. R. Quinlan, *Programs for Machine Learning*, Morgan Kaufmann Publishers, 1993.

[6] J. Han and J. Pei, "Classification: Basic concepts," *Data Mining: Concepts and Techniques*, China Machine Press, 2014.

[7] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, Data stream mining: A practical approach. [Online]. Available: https://moa.cms.waikato.ac.nz/documenttation/

[8] W. Buntine, “Learning classification trees,” *Statistics and Computing*, vol. 2, no. 2, pp. 63-67,1992.

[1] R. Kohavi and C. Kunz, "Option decision trees with majority votes," in *Proc. International Conference on Machine Learning*, pp. 161–169, 1997.

Jun Sun was born in China, in 1985. He received the Ph.D. degree in GIS from Graduated University of Chinese Academy of Sciences, in Beijing, China, in 2012. His research interests include temporal-spatial analysis method, data mining method and so on.

He is a research assistant and postdoctoral researcher in China Waterborne Transport Research Institute (WTI) of MOT, in Beijing. Dr. Sun is a member of China Computer Federation. He received Surveying Entrepreneurship Award of Academician Xia Jianbai in 2006.

Yifan Wu was born in 1991, in Beijing, China. He got his bachelor of information security in Beijing University of Posts and Telecommunications in 2003. Since then, he has majored software engineering in BUPT for master degree.

He paid great efforts to research the detection of review spam during working at the research center of intelligent information processing in BUPT. He has taken up an internship since July, 2015 in a startup company. Now, he is focusing on the research and developmen t of topic crawler.

Maoan Wang was born in 1993. He received the bachelor of management degree and bachelor of science (engineering) degree with first honors in the Sino-British program of Beijing University of Posts and Telecommunications. His major was e-commerce engineer with law.

He worked at shenzhen ZNV Technology Co., Ltd as an intern at 2014 to help build the logistic management system. Then, he joined the Enterprise Informatization Teaching and Research Section. Now, he is focusing on data science and applying for graduate programs in the US.

Guoshi Wu was born in China. He received the master degree in Northeastern University, in Shengyang, China in 1989. His research interests include intelligent information processing, big data modeling, data miningand multi content information fusion etc.

He is a professor of Software School of Beijing University of Posts and Telecommunications, a director of research center of intelligent information processing in Beijing, China. His papers include: G.Wu and K. P. Liu, *Research on Text Classification Algorithm by Combining Statistical and Ontology Methods*, *Computational Intelligence and Software Engineering*, 2009, “G. S. Wu and Y. R. Wang, "Research on e-learning system prototype based on semantic web Service technology, computing and intelligent systems," *Communications in Computer and Information Science*, pp. 247-254, Springer-Verlag Berlin Heidelberg, 2011” “G. S. Wu and F. F. Liu, "Web crawler for event-driven crawling of AJAX-based web applications," *Emerging Technologies for Information Systems, Computing, and* Management, pp. 191-200, Springer Science+Business Media NewYork, 2013”, His research interests include intelligent information processing, big data modeling, data mining, Software Engineering, and multi content information fusion etc.

Prof. Wu is a member of SOA standard of the Ministry of industry and information, China.