International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 18 (2015) pp 39269-39273

Incremental Spammer Feature Extraction for the Spam Detection in Social Networking Service

Jin Seop Han

Ph.d. student Department of Computer Science Kwangwoon University, 20, Gwangun-ro, Nowon-gu Seoul, Korea, ZIPCODE 01897 jshan74@kw.ac.kr

ByungJoon Park

Professor, Corresponding author Department of Computer Science Kwangwoon University, 20, Gwangun-ro, Nowon-gu Seoul, Korea, ZIPCODE 01897 bjpark@kw.ac.kr

Abstract
With the rise of social networks such as Facebook and Twitter, the problem of spam with a malicious purpose has become larger and more intractable. In this paper, we present an efficient method for detecting spam on Twitter. Especially, in order to efficiently extract feature values of the classifier model for the detection of spam, this paper propose methods for detecting spam based on the incremental mining approach using the only incremental data rather than a batch process using the entire data. And, we evaluated by experiment that classification accuracy of the proposed method is maintained, and time efficiency of the proposed method is improved more than existing batch process.

Keywords: Social networking service, Spam Detection, Incremental Mining, Feature Extraction.

Introduction

Social Networking Service (SNS) is internet service to enable people to have relationship with their acquaintances and other people that they don't know. In SNS, it is possible to create an account quickly through a simple subscription process, have relationship with multiple users easily, and share messages. For the reason, some people can request a relation and post a message for impure purposes, and thus spam actions polluting SNS can frequently occur. In order for spam detection, most researchers use supervised learning technique based on artificial intelligent machine learning. The SNS spam detection technique based on the supervised learning uses training data to identify the features that are used to distinguish spam from non-spam, extracts relevant values to make a classifier model, and predicts spam and non-spam through the learned classification model. In this case, identifying the features used to separate spam from non-spam effectively and extracting relevant values significantly influence classification performance. In fact, predicting spammers in the way of extracting the feature values from all data every time is an inefficient process. Therefore, this study proposes an incremental feature extraction method in which features are extracted in consideration of a data increment only in extracting the feature values necessary to detect spam in SNS.

This paper is comprised of as follows: section 2 describes related works on incremental mining and existing SNS spam detection; section 3 explains the proposed incremental feature extraction method; section 4 presents the results of the test conducted in the application of the proposed method; section 5 shows the drawn conclusion.

Related Work

In data mining, it is inefficient to perform mining with all data every time to process mass data and find a mining pattern. Incremental mining is applied mainly to association rule mining used to find a sequential pattern [1, 2, 3], and scans newly added data only, not the whole previously mined data, in creating a data pattern and a rule [7]. The incremental mining method used to be applied to the studies on Betweenness Centrality [4] and Closeness Centrality [5] which were aimed at measuring centrality of one node in the dynamic SNS network structure, and to the study on finding a node aggregate with similar tendencies [8].

With regard to the works related to SNS spam detection, there is the research [12] that proposed spam detection framework in consideration of messages, users, and user relationship [12], and Defensios system that detects spam messages in Facebook was proposed [15]. The system based on SVM (Support Vector Machine) detects malicious messages and URLs mainly. Also, the spam detection system based on logistic regression analysis technique was proposed in consideration of Twitter words and URLs [13]. The proposed method [9] was to use the number of friends and the number of friend requests in Twitter, and clustering technique to analyze the association structure of messages and identify the features necessary to separate from legitimate messages. The spam classification method [10] analyzed such features as the number of message transmissions to users who had no relations. Although the proposed method [10] achieved higher accuracy, it used a relatively smaller amount of data. Therefore, if a massive amount of data is used, it is hard for the method to guarantee its performance and efficiency. Social-Honeypot suggested relatively many features of Twitter users, conducted spam detection test with more data than, and thereby proposed a classifier model with high classification accuracy (98.42%) [14]. Table 1 presents the comparison of existing works in terms of approach and data set.

TABLE.1. Existing work on SNS spam detection

Author	Approach	Data set
Hongyu Gao et al.[9]	SVM Algorithm, Total of 6 features including Average time interval	217,802 posts in Facebook and 467,390 tweets in Twitter, as spam
Maarten Bosma et al.[10]	HITS Algorithm, Relation between messages, Authors, User spam reports, Reporters	28,998 spam reports, 13,188 messages, and 9,491 users
Jonghyuk Song et al.[12]	Total of 11 features including Mentions sent to non-followers	1,000 non-spammers, 300 spammers 50 tweets per user
Lee, K. et al.[14]	Random Forest Algorithm, Total of 19 features including the length of the screen name	41,499 user, 5,643,297 post in Twitter

This paper referred to research on Social-Honeypot which is described in detail as follows.

To collect the training data necessary for classification learning, Social-Honeypot arranges multiple software agent accounts (Honeypot) in internet and collects the accounts with which unwanted friend requests are tried. In the Social-Honeypot environment, message transmission and friend request can be tried between Honeypot accounts. Therefore, a non-Honeypot account that made a friend request to a Honeypot account is considered to be contents polluter or spam. And, the user profiles of the induced accounts, relationship between users, messages, and user history information were analyzed, and thus a total of 19 features were identified. The identified features are presented in the following:

- User Profile: Length of the screen name, length of the user description, and longevity of the account
- User Relationship: Number of followers, number of followings, ratio of followings to followers, percentage of bidirectional friends, percentage of bidirectional followers, standard deviation of unique numerical IDs of the followings, and standard deviation of unique numerical IDs of the followers
- User Contents: Number of posted tweets, number of posted tweets per day, average number of links in a tweet, average number of unique links in a tweet, average number of @usernames in a tweet, average number of unique @usernames in a tweet, average content similarity over all pairs of tweets posted by a user, and ratio of the uncompressed to the compressed size of tweets
- User History: Average daily change of the number of followings over a period of time for each user

Social-Honeypot identifies such various features, calculates their values, and creates a classifier model to predict spammers.

Previous studies including the above Social-Honeypot are based on batch process in which feature values are extracted from all data at every time when feature values are extracted in order for classification prediction. Therefore, this study proposes an incremental feature extraction method which uses newly accumulated data only without batch process of all data every time in order to extract feature values and thereby can efficiently predict spammers.

Incremental Feature Extraction Method

Incremental mining is aimed at using knowledge of previous mining result and newly added data and thereby producing the same result as the batch process of all data. As shown in Fig. 1, this study proposes a method to extract feature values on the basis of incremental mining approach.

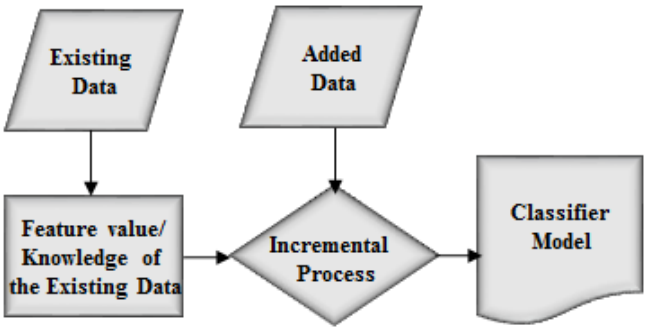

Fig.1. Incremental Feature Extraction

In the proposed method, incremental process is applied only to the feature values and knowledge of the existing data and newly added data and creates a classifier model. This study analyzed existing 19 features of Social-Honeypot and the features proposed by the previous research [6], and classified them with batch process and incremental process as shown in Table 2.

TABLE.2. Batch Process and Incremental Process

Batch Process (Thirteen features)	Incremental Process (Seven features)
- Length of the screen name, length of the user description, longevity of the account - Number of followers, number of followings, ratio of followings to followers - Percentage of bidirectional friends, percentage of bidirectional followers - Standard deviation of unique numerical IDs of the followings, standard deviation of unique numerical IDs of the followers - Number of posted tweets, number of posted tweets per day - Ratio of the uncompressed to the compressed size of tweets	- Average number of links in a tweet, average number of unique links in a tweet - Average number of @usernames in a tweet, average number of unique @usernames in a tweet - Average content similarity over all pairs of tweets posted by a user - Average daily change of the number of followings over a period of time for each user - Average daily change of the number of messages a user has posted over a period of time per user [6]

The features classified by batch process are related to user profile and relationship. In terms of user contents features, there are 'the number of messages', 'the average number of daily messages', and 'message compression ratio'. When the values of the features-'user profile', 'user relationship', 'the number of messages', and 'the average number of messages'-are obtained through the execution of Twitter API each time, it is possible to extract the feature values suitable to mining. The 'message compression ratio' means the ratio of the size of an original message size and the size of its compressed message. When the feature values are calculated with all data, it is possible to extract accurate values. Therefore, batch process is effective.

The features classified by incremental process are mostly related to user contents and history. To extract the feature values, this study defined and applied the following heuristic formulas.

- Average number of links in a tweet

$$(PreVal \times PreMsgNum + URLNum)/TotalMsgNum \quad (1)$$

- Average number of unique links in a tweet

$$(PreVal \times PreMsgNum + UniqueURLNum)/TotalMsgNum \quad (2)$$

The formulas (1) and (2) are about the ratio of URL inclusion in messages ((2) excludes duplication). A spam message tends to have a higher value than a legitimate user's message. The ratio of previously extracted URLs without scanning of existing data messages (PreVal), the number of URLs (URLNum) extracted through scanning of the added data to the existing message number (PreMsgNum), and the total message number (TotalMsgNum) are applied to the formulas for calculation. The below (3) and (4) represent the formulas for the ratio of '@' inclusion in message. '@' is used when a message is sent to a particular user. A spam message tends to have a smaller value than a legitimate user's message. They are calculated in the same way as the formulas (1) and (2).

- Average number of @usernames in a tweet

$$(PreVal \times PreMsgNum + AtNum)/TotalMsgNum \quad (3)$$

- Average number of unique @usernames in a tweet

$$(PreVal \times PreMsgNum + UniqueAtNum)/TotalMsgNum \quad (4)$$

- Average content similarity over all pairs of tweets posted by a user

$$(PreVal \times PrePair + Val \times Pair + AddVal \times AddPair)/(PrePair + Pair + AddPair) \quad (5)$$

The formula (5) is average contents similarity over all pairs of messages posted by a user. In formula, contents similarity is calculated by using the cosine similarity algorithm the same as Social-Honeypot. A spam message tends to have a larger value than a legitimate user's message. Existing similarity average value (PreVal), the similarity average value (Val) of an added message, the similarity average value (AddVal) of the pair of existing message and the added message are calculated. The number of each message pair (PrePair, Pair, AddPair) is applied to the formula in order for incremental process.

- Average daily change of the number of followings over a period of time per user

$$\sqrt{(PreVal^2 \times (Period-2) + |Following - PreFollowing|)/(Period-1)} \quad (6)$$

The formula (6) is used to calculate the average daily change of the number of friend requests over a certain period. A spam message tends to have a larger value. In consideration of the formulas of Social-Honeypot [14], existing change ratio (PreVal), a period (Period), the number of friend requests today (Following), and the number of friend request previous day (PreFollowing) are applied in order for incremental process.

- Average daily change of the number of messages a user has posted over a period of time per user

$$(PreVal \times (Period-2) + |MsgNum - PreMsgNum|)/(Period-1) \quad (7)$$

The 'average change of the number of messages' used in this study is the feature proposed by the previous research [6]. The average change of the number of messages posted by a user over a certain period of time is calculated. A spam message tends to have a larger value. As shown in the formula (7), existing change (PreVal), a period (Period), the number of messages today (MsgNum), and the number of messages previous day (PreMsgNum) are applied in order for incremental process.

As shown above, this study classified the features for incremental process, defined formulas, and extracted feature values.

The next section describes the experimental result to evaluate the performance of the proposed method.

Experimental result

This section presents the experiment and evaluation of the proposed method, and describes experimental dataset.

A. Dataset

For the experiment of the proposed incremental feature extraction method, the data of Social-Honeypot [14] are used. Social-Honeypot consists of 19,276 legitimate users (46.4%), 22,223 contents polluters (53.6%). In terms of messages, the number of legitimate users' messages is 3,263,238, and the number of polluter messages is 2,380,059. The details of the dataset are shown in Table 3.

TABLE.3. Details of the Social-Honeypot data

File Name	Format
Profile.txt	UserIDCreatedTimeCollectedTime Followings Followers Tweets LenOfScreenNameLenOfDesc
Tweets.txt	UserIDTweetID Tweet CreatedTime
Followings.txt	UserIDSeriesOfNumberOfFollowings
Supplement.txt	UserIDrateOfBifriendsOfFollowingsrateOfBifriendsOfFollowersSDofFollowingsSDofFollowers

Social-Honeypot had arranged legitimate users from Feb. 2007 to Nov. 29, 2009, and had induced spam from Apr. 2007 to Aug. 2010, during which the period of intensively inducing spam data was from Dec. 30, 2009 to Aug. 2, 2010. With the Social-Honeypot data, this study redesigned the original data in order to meet its purpose and test its proposed method effectively. As a result, Twitter data, which had been posted from Oct. 31 to Nov. 29, 2009 (30days), the period when legitimate users and polluter were collected equally, were redesigned by day and then incremental feature extraction test was conducted. Over the 30 days, 2,523,219 messages of legitimate users and 78,847 of polluter messages were used for the test of this study.

B. Experiment and evaluation

In the experiment, existing batch process and the incremental process proposed in section 3 are applied to extract feature values, and comparison evaluation between the two methods was made in terms of accuracy and time-efficiency. The experiment was implemented with C language in Windows 7, and its hardware system has Intel(R) Core(TM) i7-2600k 3.40 GHz CPU and 8GB RAM. A classifier model was created by each method, and its accuracy performance was measured with Weka toolkit Version 3.7.6 [11]. Twenty features were measured by batch process and by incremental process. As the result, the accuracy of both methods was 99.427%, equally. With the experimental dataset daily redesigned over 30 days, the extraction time of batch process and of incremental process by day was measured. As shown in Table 4, when the average time was measured, the seven features extracted by incremental process had shorter time on average than by batch process.

TABLE.4. Average time of the feature extraction over 30 days

Process \ Feature	Batch (min:sec)	Increment (min:sec)
- Average number of links in a tweet, average number of unique links in a tweet - Average number of @usernames in a tweet, average number of unique @usernames in a tweet	00:1.777	00:0.192
- Average content similarity over all pairs of tweets posted by a user	19:52.688	01:32.760
- Average daily change of the number of followings over a period of time for each user	00:0.077	00:0.035
- Average daily change of the number of messages a user has posted over a period of time per user	00:1.225	00:0.109

Especially, it takes a lot of time to extract the value of the feature 'message similarity average'. When it was extracted by incremental process, around 18 minutes and 30 seconds shortened. Fig. 2, Fig. 3, Fig. 4, and Fig. 5 illustrate the comparison of feature values over 30 days.

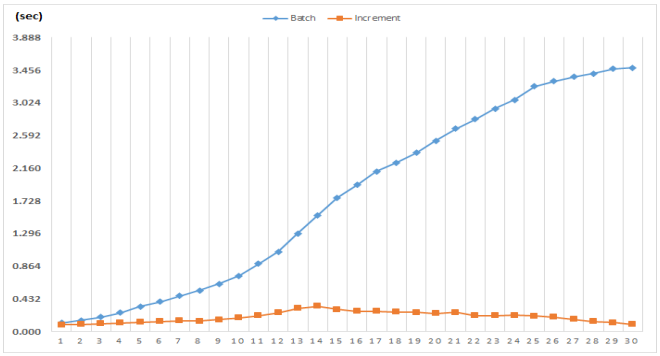

Fig.2. Extraction time of the number of URL, unique URL, @, and unique @ per message

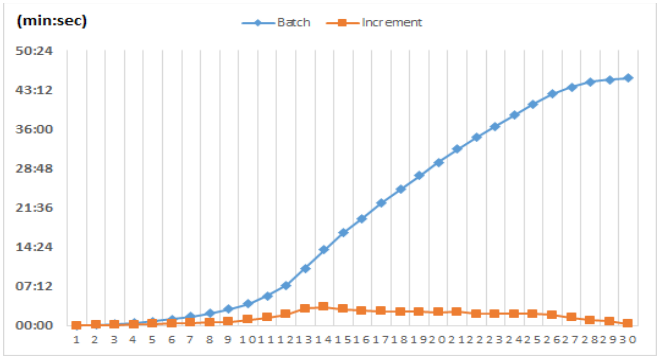

Fig.3. Extraction time of the average content similarity over all pairs of message

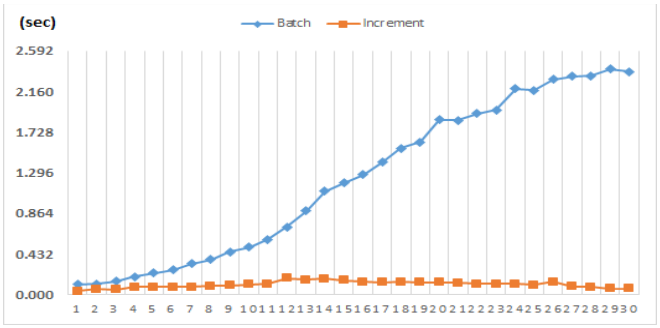

Fig.4. Extraction time of the change rate of the number of followings

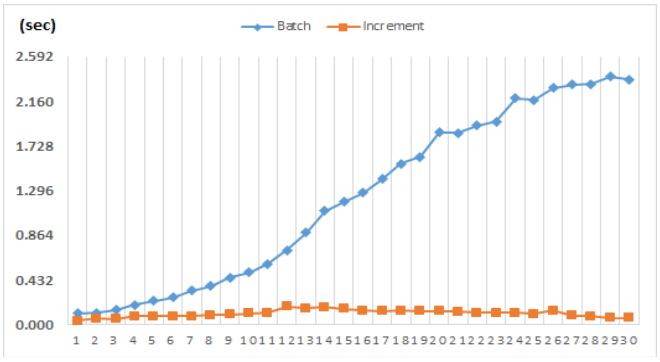

Fig.5. Extraction time of the average change rate of the number of the messages

As shown in the figures, daily data are accumulated and increased so that batch process based on all data causes rises in the extraction time of feature values with the lapse of time. On contrary, since incremental process extracts feature values only with the use of a daily increment in data, its extraction time changes a little with the lapse of time.

Conclusion

This paper proposed an incremental feature extraction method to detect spam efficiently in SNS. Unlike the batch mining process based on all data to extract the feature values of a classifier model for spam detection, the proposed incremental mining method took into account a data increment only to extract feature values. For the method, the formulas to identify the features for incremental process and extract the feature values were defined. Based on the formulas, the extraction experiment of batch process and of the proposed incremental process had been conducted over 30 days. As the result, the proposed method had the same accurate classification performance as batch process and showed better time-efficiency than batch process.

Acknowledgments

This research was supported by Research Grant of Kwangwoon University in 2013.

References

[1] Sreedevi, M., Kumar, and G. V., “Parallel and Distributed Approach for Mining Closed Regular Patterns on Incremental Databases at User Thresholds”, Proc. of the 2014 International Conference on Information and Communication Technology for Competitive Strategies, ACM, pp. 59-63, 2014.

[2] Chen, Y. C., Weng, J. T. Y., Wang, J. Z., Chou, C. L., Huang, J. L., and Lee, S. Y., “Incrementally Mining Temporal Patterns in Interval-based Databases”, Data Science and Advanced Analytics (DSAA), pp. 304-311, 2014.

[3] Mehta, Gunjan, Deepa Sharma, and Ekta Chauhan, “Application of Incremental Mining and Apriori Algorithm on Library Transactional Database”, International Journal of Computer Applications, pp. 73-78, 2013.

[4] MirayKas, Matthew Wachs, Kathleen M. Carley, and L. Richard Carley, “Incremental Algorithm for Updating Betweenness Centrality in Dynamically Growing Networks”, Proc. of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. ACM, pp. 33-40, 2013.

[5] MirayKas, Kathleen M. Carley, L. and Richard Carley, “Incremental Closeness Centrality for Dynamically Changing Social Networks”, Advances in Social Networks Analysis and Mining (ASONAM), pp. 1250-1258, 2013.

[6] Jin Seop Han, ByungJoon Park, “Efficient Detection of Content Polluters in Social Networks”, IT Convergence and Security 2012(Lecture Notes in Electrical Engineering, pp. 991-996, 2013.

[7] Shah, Siddharth, N. C. Chauhan, and S. D. Bhander, “Incremental Mining of Association Rules: A Survey”, International Journal of Computer Science and Information Technologies, Vol. 3, no. 3, pp. 4071-4074, 2012.

[8] Papadopoulos, S., Kompatsiaris, Y., Vakali, A., and Spyridonos, P., “Community detection in Social Media Performance and application considerations", Data Mining and Knowledge Discovery 24.(3), pp. 515-554, 2012.

[9] Hongyu Gao, Yan Chen, Kathy Lee, Diana Palsetia, and AlokChoudhary, “Towards Online Spam Filtering in Social Networks”, Proc. of 19th Network Distributed System Security (NDSS) Symposium, http://www.internetsociety.org, 2012.

[10] Maarten Bosma, Edgar Meij, and WouterWeerkamp, W., “A Framework for Unsupervised Spam Detection in Social Networking Sites”, Proc. of European Conference on In-formation Retrieval (ECIR), pp. 364-375, 2012.

[11] Eibe Frank, Available from http://www.cs.waikato.ac.nz/ml/weka/downloading.html, 2012.

[12] Jonghyuk Song, Sangho Lee and Jong Kim, “Spam filtering in twitter using sender-receiver relationship”, Proc. of the 14th International Symposium on Recent Advances in Intrusion Detection (RAID), pp. 301-317, 2011.

[13] KristoFer Beck, “Analyzing Tweets to Identify Malicious Messages”, Proc. of Electro/Information Technology (EIT) IEEE International Conference, pp. 1-5, 2011.

[14] Kyumin Lee, Brian David Eoff, and James Caverlee, “Seven Months with the Devils: A Long-Term Study of Content Polluters on Twitter”, Proc. of International AAAI Conference on Weblogs and Social Media (ICWSM), pp. 185-192, 2011.

[15] Saeed Abu-Nimeh, Thomas M. Chen, and Omar Alzubi, “Malicious and Spam Posts in Online Social Networks”, IEEE Computer Society, Vol. 9, pp. 23-28, 2011.