Web robot detection with semi-supervised learning method

Wang Dong[1, 2, a *], Xi Lei[1, 3, b *], Zhang Hui[1, 3, c],
Liu Hebing[1, 3, d], Zhang Hao[1, 3, e], Song Ting[3, f]

[1]Collaborative Innovation Center of Henan Grain Crops, Zhengzhou, 450002, China

[2]Department of Information Management, Henan Agricultural University, Zhengzhou, 450002, China

[3]College of Information and Management Science, Henan Agricultural University, 450002, China

[a]dwang@henau.edu.cn, [b]hnaustu@126.com, [c]zhhnau@163.com

[d]liuhebing@henau.edu.cn, [e]zhanghaohnnd@126.com, [f]ms_songting@163.com

* these authors contributed equally to this work

Keywords: Web robot detection, Semi-supervised learning, Support Vector Machine.

Abstract. Web robot is an automated information gathering program that has brought a lot of problems such as information leakage, resource occupation and network security threaten. It is necessary to effectively detect and control the web access comes from web robots. Summarizes the existing categories of web robot detection method, we propose a new detection method with semi-supervised support vector machine. The experiments based on the same test data set are presented to show that the new method is superior to other robot detection methods.

Introduction

As the Internet increasingly prevail, more and more traditional essential services, such as education, medicine, transportation, banking and shopping, are being offered by means of web-based applications which can be directly accessed over the network. These changes bring us convenient and efficient, but also face more security threats. Web robots form a critical component of search engines, undertaking information discovery task, has becoming an issue can't be ignored. These automatic programs always been used to collect sensitive information, personal privacy or confidential data in case we did not realize. Recently, web robot was also used to execute automated DDoS attacks on web sites. Therefore, detection and discovery of Web robot has become an important topic in the field of network security.

Current detection methods are mainly divided into four categories: a) approaches based on the HTTP header analysis [1]. These methods rely on user-agent field and other information in the access requests to identify robot program. A well-behaved web robot always declared their identity in the user-agent field. According the user-agent field to identify web robot is simple and effective. As the user-agent information is easy to deceive, malicious robots tending to hide their access by masquerading as a normal browser visit, only through the HTTP header information discrimination is difficult to find web robot in disguise. b) honey pot technique [2]. Based on the active offense theory, honey pot is a newly arisen technology which is valued by the realm in computer network security increasingly. Typically web robot identity methods based on honey pot include robot protocol detection method and hidden links method. Robot protocol detection method identity robot.txt access or add a file record in robot.txt which is not really exists. Clients that accessed to these documents are likely to be malicious robots. These methods could find the robots which read the robots.txt file but not comply with the robots protocol. In contrast, the robots that don't read the robots.txt file are incapable of action. Hidden links methods always add some hidden links in the webpage which can't be seen by the eye but could be accessed by robots. For example, the link text color could be set to the background color or covered with graphic and other page elements. Hidden links methods can easily capture web robots, but for some of the more intelligent robots or robots only for specific file such as video, picture

© 2015. The authors - Published by Atlantis Press

are often not achieve very good detection results. c) access feature analysis methods [3,4,5]. These methods identify robot access by analyzing the characteristics of Web access to find the different of person and program [6]. Study concluded some important features: click number, HTML-to-image ratio, percentage of error responses, link relationships, standard deviation of requested page's depth, and so on. The accuracy of the analysis method based on the selection of access characteristic feature. Selection different features can cause greater differences detected result. Access features are often combined in machine learning algorithms to find web robot. d) detection algorithms based on machine learning [7,8,9]. Currently a large number of machine learning algorithms been used to detect robots [10]. These methods got superior accuracy to other methods. The typical methods including: C4.5 algorithm(a decision tree classifier), RIPPER, k-nearest neighbor algorithm, Naive Bayesian model [11]. Studies have shown that using the supervised algorithm to classify Web access into different sets (human access and robot visit, robots can be further divided into well-behaved robots and malicious robots) can achieve more satisfactory performance. However, the performance of supervised learning algorithms relied heavily on the label data. The supervised algorithm can only achieve the same effect with labeled data. The error in label data will cause great error in classification. Unsupervised neural networks [12], hidden Markov model [13], the Turing test algorithms [14] has also been used to detect robots. These algorithms do not need to label data, using statistical information identify robots, but got worse accuracy than supervised learning algorithms. How to combine the advantages of supervised learning and unsupervised learning, using plenty of unlabeled data to improve the accuracy of the supervised learning algorithm is a direction of current research [15]. This paper presents a semi-supervised robot detection method.

Semi-supervised support vector machine method

Support vector machine. Support vector machine (SVM) is a new general knowledge discovery and machine learning method over the past decade. This technology has a solid foundation of statistical theory and been used in many areas (network intrusion detection, natural language understanding, text classification, handwriting recognition, protein classification) [16]. The main idea of support vector machine is looking for the largest classification edge in the high-dimensional feature space (also referred to as the optimal classification plane), and ensuring minimum classification error in unknown samples. Such SVM problem was described as a constrained optimization problem:

$$\min_{w} \frac{\|w\|^2}{2} \qquad s.t. \quad y_i(w \cdot x_i + b) \geq 1, \quad (i = 1,2,3,\mathbf{L},n) \tag{1}$$

Linearly separable SVM is to ensure all training samples were correctly classified, at the same time to get the best generalization performance by maximizing classification interval. Usually the ideal state is unable to reach, in order to allow SVM construct a linear decision boundary in the case can't be separated linear, we introduced positive relaxation factor x_i in constrained optimization problem. In this case, the optimization problem becomes:

$$\min_{w} \left(\frac{\|w\|^2}{2} + C\sum_{i=1}^{n} x_i \right) \quad s.t. \quad y_i[(w \cdot x_i) + b] \geq 1 - x_i, i = 1,\mathbf{L},n \quad x_i \geq 0, \quad i = 1,\mathbf{L}\, n \tag{2}$$

As it involves a large number of parameters w,b,x_i, the solution to this optimization problem is a thorny issue. Using Lagrange method the optimal classification problem can be transformed into its dual problem:

$$\min_{l} \left(\frac{1}{2}\sum_{i}\sum_{j} l_i l_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{n} l_i \right) \quad s.t. \qquad \sum_{i=1}^{n} l_i y_i = 0 \qquad 0 \leq l_i \leq C \tag{3}$$

Where l_i is the Lagrange multiplier. According to KKT conditions, a large number of training samples not on the classification plane is necessarily meet $l_i = 0$, and a few training samples on the classification plane meet $l_i > 0$ are called support vectors. The support vector meet $0 < l_i < C$ was called non-border support vector, and the vector $l_i = C$ was called boundary support vector. Using

optimization methods get l_i and to further determine the parameters b, the following classification function could be used classify the test samples:

$$f(x) = sign(w \cdot \Phi(z) + b) = sign\left(\sum_{i=1}^{n} l_i y_i \Phi(x_i) \cdot \Phi(z) + b\right) \quad (4)$$

To avoid dimensional disaster problem, we usually compute kernel function $K(x_i, z) = \Phi(x_i) \cdot \Phi(z)$ instead of the vector dot product.

Feature extraction. In order to classify access records into human or robot access, we should first extract structured access session in the access log. Web session is a client requests sequence during a web site visit. In order to classify Web session, we need to construct characteristic features describing each session. Web access features must have integrity, that feature items must be able to confirm that the contents of the target. But too much feature will cause the classification algorithm complexity increasing and the weight of key features reducing, affect the accuracy of classification. Therefore, we must choose the least possible information to reflect the characteristics of robots from the numerous sessions dimensional vector and assign appropriate weight. According to previous research, combined with the concrete practice, we use to extract the 11 most important features for robots detection: User-Agent, Request number, HTML-to-Image Ratio, Percentage of PDF / PS file requests, Percentage of 4xx error responses, Percentage of HTTP requests of type HEAD, Percentage of requests with unassigned referrers, Robots.txt file request, Request's depth, Percentage of consecutive sequential HTTP requests, Multi IP requests. Under normal circumstances, the request during a session robots access to far more than the number of requests that the human user access. For some access has a handful of requests but claim robot in the User-Agent, which may cause potentially identification conflict, we indicate the session which claim robot first and use support vector machine to classify another session.

Weight calculation. For access features extracted from the session, we must find a way to express it which easily processed by computer. In this paper, VSM (Vector Space Model) to represent the Web access session. An access log file, you can build an n-dimensional vector space, each dimension represents a different access sessions feature. Access logging can be represented by a set of such vectors, called feature vectors. Feature vector can be mathematically described by an n-tuple, composed by n items (or index entries), each feature item has a certain weight. The entire access log can be abstract to a collection of vectors. After weight calculation, the access log has been converted into a feature vector sets with N-dimensional, the value of each element of the vector is digital. The feature vectors could be put into the support vector machine, after training it can get the maximum interval hyper plane and be used to classify and identify web robot. However, as each feature item represents a completely different meaning, its value is not comparable. Although the different weights can be balance by SVM training, a good input data can often improve the accuracy of classification detection. In this algorithm we use the formula $f(x_i) = \frac{x_i}{\max(x)}$ normalizing weights of each item, to ensure that all the data are in the 0-1 range, so that each input vector is a point in a multi-dimensional vector space with radius 1.

Semi-supervised robot dection with RBF kernel. Although the weights calculation converted the data into a vector with radius 1, the normalization process without considering the ratio between the feature item, directly classification the original vector may linearly inseparable. According to the Mercer theorem, the samples which are linear inseparable in the low dimensional may linearly separable by nonlinear mapping into a high-dimensional feature space. In this paper we choose Gaussian radial basis function (RBF) as the support vector machine kernel function. First, normalizing the input vector with a basic kernel $K(X,Y) = \frac{X \cdot Y}{\sqrt{(X \cdot X)(Y \cdot Y)}}$,then we converted the basic SVM kernel into a Gaussian radial basis function kernel, so the kernel function is transformed into: $\hat{K}(X,Y) = e^{-\frac{K(X,X)-2K(X,Y)+K(Y,Y)}{2s^2}} + 1$, where s is Euclidean distance median of all the positive training

samples to the nearest negative training samples. The constant 1 is added to the kernel to make the data leave the original point.

After training with a small amount of expert labeled data, the support vector machine could be used to detection robot by the hyper plane discriminant function. As the generated discriminant function is completely dependent on the initial training data, the accuracy will be unsatisfactory if the training data is not typical. In order to solve this problem, we involve the unlabeled data into support vector machines training. After each classifying we add the top 10% confidence data as new training data with added label into the training set, and then generate a new discriminant function. In the iterative process, the bias from training sample was gradual corrected and the classifier achieve better performance.

Experiments

In the experimental stage of our study, the training data sets were constructed by pre-processing web server access log files provided by Henan Agricultural University. The log files contained detailed information about user web-based access into the domain www.henau.edu.cn during a 7-day interval – between Jan 5 2015 and Jan 11 2015. A total of about 2.9 million log entries were extracted from the file. All data divided into seven data sets. The first one data set is the training set and the other six data sets are test sets. For each data set, the original access records were aggregated in access sessions based on 11 important features. Table 1 lists data set partitioning with session and records.

Table 1. data set partitioning with session and records

Data Set	Total access		well-behaved robots		malicious robots		human access	
	session	record	session	record	session	record	session	record
1	5416	451383	1058	165581	682	116387	3676	169415
2	6213	471749	1093	168235	815	122090	4305	181424
3	6352	475110	1251	179546	966	123031	4135	172533
4	5047	419067	958	166008	724	117339	3365	135720
5	5829	469418	1129	172156	803	129437	3897	167825
6	3267	306708	927	160171	698	104081	1642	42456
7	3648	313362	1014	163715	732	108143	1902	41504
Total	35772	2906797	7430	1175412	5420	820508	22922	910877

As can be seen from table1, there are 12850 access sessions are robot access in total 35772 access sessions, about 40% of the total number of access sessions. The 12850 robot access sessions include 2 million access records, accounting for 69% of all access records. Robot accesses over the weekend take up 85% of the total number of accesses as the reduced human visit. So we can see that the robot access take up a large amount of web server resources. The goodwill robot visit come from MSN, Google, Baidu, Yahoo and other robots accounted for 59% in all robot visits. The malicious robot which disguised itself as browser access accounted for 41%. The results showed that malicious robot access problem couldn't be solved by consciously abide the robot agreement.

Web robot detection target is identified robot visit from all web access. The most important indicators to judge detection effect are accuracy rate, coverage rate and F1 measure. Accuracy rate, also known as precision, is the percentage of the correct detected robot access number and full robot access (the correct detected robot access plus error detected robot access). Coverage rate, also called recall rate, is the percentage of the correct detected robot access and the actual total number of robot access (the correct detected robot access plus undetected robot access). In practice only the coverage rate or the accuracy rate often do not reflect the classifier's true capacity, we often use F1 score to measure the classifier's performance, which is defined as $F1 = \frac{2rp}{r+p}$. The three important indicators were used to evaluate the detection method in following experiments.

In the semi-supervised experiments, using the first day's access data as the training set to process SVM training to acquire the discriminant function. Then we used the trained classifier test the second day data and combine the top 10% confidence access into training data set and training SVM again. The process was iterative training and testing using the 2-7 days' data. Fig 1 shows the classifier performance changes in the iterative process of semi-supervised classifier.

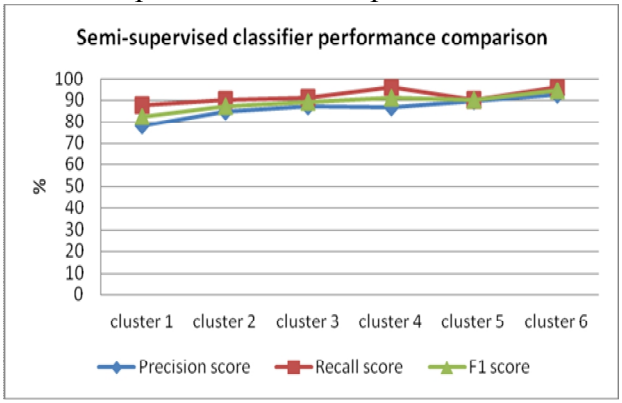

Fig 1. Semi-supervised classification performance comparison

Experimental results show that the SVM classifier's performance has been further improved in using semi-supervised method. In the cluster5 data set (Saturday), the classifier's performance showed a decline as the large web access change. In the next data set (Sunday) the classifier realized the self-correcting with semi-supervised learning.

In subsequent experiments, comparing the semi-supervised support vector machine method (S3VMs) with several major web robot detection methods (the HTTP header analysis, Honey pot, Feature analysis, C4.5 discrimination trees, RIPPER, K-Nearest Neighbor algorithm, Naive Bayesian model, Neural Network) on the same test set.

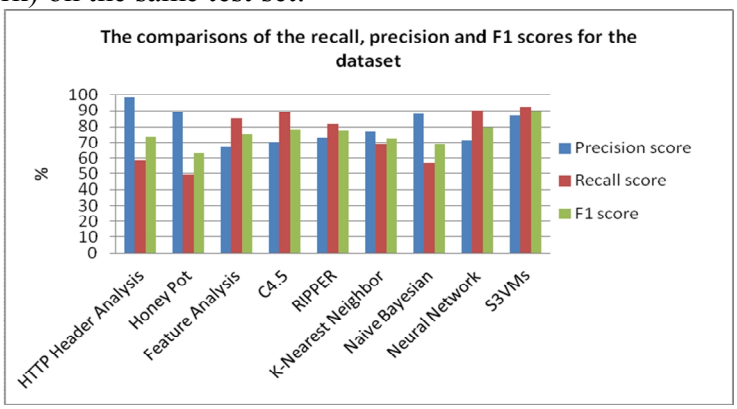

Fig 2. The comparisons of the recall, precision and F1 scores for the data set

The result (Figure 2) shows that HTTP Header Analysis and Honey Pot have high precision score and low recall score. The F1 score is also lower than the feature analysis and machine learning-based approach. The methods based on machine learning are often integrated with the results of feature analysis methods and achieve better performance. The semi-supervised support vector machine method (S3VMs) superior to other robot detection methods as it introduced the tag data and the unlabeled data participating classification.

Conclusion

This paper presents a new web robot detection method based on semi-supervised support vector machine which got a better dection performance than the other methods. During the experiment we also found that some of the old methods do not made the purported effects. It also shows that the robots are becoming more intelligent with the development of technology.The war between robot and robot detection will continue. Tracking the latest robot technology, designing a better feature

extraction method, using the latest artificial intelligence research to improve web robot detection is a direction which needs our further efforts.

Acknowledgements

This work was financially supported by the Henan Province Major Science and Technology Projects(131100110400).

References

[1] Doran D, Gokhale S S. Web robot detection techniques: overview and limitations[J]. Data Mining and Knowledge Discovery, 2011, 22(1-2): 183-210.

[2] Kaur H, Madan S, Sehgal R K. UAC: A Lightweight and Scalable Approach to Detect Malicious Web Pages[M]. Modern Trends and Techniques in Computer Science. Springer International Publishing, 2014: 241-261.

[3] AlNoamany Y A, Weigle M C, Nelson M L. Access patterns for robots and humans in web archives[C]. Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries. ACM, 2013: 339-348.

[4] Doran D, Gokhale S S. Detecting Web Robots Using Resource Request Patterns[C]. Machine Learning and Applications (ICMLA), 2012 11th International Conference on. IEEE, 2012, 1: 7-12.

[5] Kwon S, Kim Y G, Cha S. Web robot detection based on pattern-matching technique[J]. Journal of Information Science, 2012, 38(2): 118-126.

[6] Kwon S, Oh M, Kim D, et al. Web robot detection based on monotonous behavior[J]. Proceedings of the Information Science and Industrial Applications, 2012, 4.

[7] Sardar T H, Ansari Z. Detection and confirmation of web robot requests for cleaning the voluminous web log data[C]. IMpact of E-Technology on US (IMPETUS), 2014 International Conference on the. IEEE, 2014: 13-19.

[8] Stevanovic D, An A, Vlajic N. Feature evaluation for Web robot detection with data mining techniques[J]. Expert Systems with Applications, 2012, 39(10): 8707-8717.

[9] Stevanovic D, An A, Vlajic N. Detecting Web robots from web server access logs with data mining classifiers[M]. Foundations of Intelligent Systems. Springer Berlin Heidelberg, 2011: 483-489.

[10] Hou Y T, Chang Y, Chen T, et al. Malicious web content detection by machine learning[J]. Expert Systems with Applications, 2010, 37(1): 55-60.

[11] Stassopoulou A, Dikaiakos M D. Web robot detection: A probabilistic reasoning approach[J]. Computer Networks, 2009, 53(3): 265-278.

[12] Stevanovic D, Vlajic N, An A. Detection of malicious and non-malicious website visitors using unsupervised neural network learning[J]. Applied Soft Computing, 2013, 13(1): 698-708.

[13] Lu W Z, Yu S Z. Web robot detection based on hidden Markov model[C]. Communications, Circuits and Systems Proceedings, 2006 International Conference on. IEEE, 2006, 3: 1806-1810.

[14] Stevanovic D, Vlajic N, An A. Unsupervised clustering of Web sessions to detect malicious and non-malicious website users[J]. Procedia Computer Science, 2011, 5: 123-131.

[15] Chapelle O, Sindhwani V, Keerthi S S. Optimization techniques for semi-supervised support vector machines[J]. The Journal of Machine Learning Research, 2008, 9: 203-233.

[16] Wang D, Song T, Liu B. Optimizing Discriminant Model for Improved Classification of Protein[J]. Applied Mechanics and Materials, 2013, 411: 3227-3231.