Journal of Computational Information Systems 7: 6 (2011) 2078-2086
Available at http://www.Jofcis.com

Online Internet Traffic Classification Based on Proximal SVM

Chengjie GU[1,†], Shunyi ZHANG[1], He HUANG[2]

[1]*Institute of Information Network Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003, China*

[2]*School of Software, Beijing University of Aeronautics and Astronautics, Beijing 100083, China*

Abstract

Online and accurate traffic classification is a key challenge for network management. Internet traffic classification based on flow statistics using machine learning method has attracted great attention. To solve the drawback of the previous classification scheme to meet the requirements of the network activities, our work mainly focuses on how to build an online Internet traffic classification system based on flow statistics. We propose an online Internet traffic classification based on proximal SVM. Experiment results illustrate this method can classify online traffic with first p packet of flow with high accuracy. Meanwhile, the proximal SVM method is computationally more efficient than the previous SVM methods with similar accuracies.

Keywords: Traffic Classification; Proximal SVM; Supervised Learning; Feature Selection

1. Introduction

Internet traffic will increase 46% annually from 2007 to 2012 according to measurement study in the literature [1]. The demand for bandwidth management methods that optimize network performance and provide QoS guarantees has increased substantially in recent years [2]. Therefore, accurate and online classification of network traffic plays important roles in many areas such as traffic engineering, QoS, and intrusion detection etc.

Machine learning technique which is a powerful tool in data separation in many disciplines aims to classify data based on either a priori knowledge or statistical information extracted from raw dataset. This method can be well suited with Internet traffic classification, as long as the traffic classified into categories that exhibit similar characteristics in parameters. Therefore, traffic classification method based on flow statistics shows effective performance in this field [3]. Substantial attention has been invested in data mining techniques and machine learning algorithms using flow features for traffic classification.

While traffic classification methods based on flow statistics offer various degrees of successes, there are several limitations. 1) Classifier with certain machine learning methods, such as bayesian estimating, C4.5, nearest neighbor, may be trapped into local optimization. 2) Although a relatively high accuracy is achieved, these methods do not fit into the real-time situation due to their requirement on computation and storage.

To address the above-mentioned problems, we take two aspects to improve the accuracy and speed of the

[†] Corresponding author.
Email addresses: jackiee.gu@gmail.com (Chengjie GU).

machine learning methods for Internet traffic classification. 1) In order to achieve early detection, we allow the classifier to classify traffic flows early in the connection using the first *p* packets of flow. 2) We use proximal SVM method which can obtain high accuracy with faster computational time as a maximum margin classifier to avoid local optimization.

The remainder of this paper is structured as follows. Related work is represented in Section2. Section 3 describes our proposed online Internet traffic classification based on proximal SVM. Section 4 presents the experimental results and analysis. The conclusions and potential future work are listed in Section 5.

2. Related Work

The port-based traffic classification relies on well-known port number to classify different Internet applications [4]. However, the new P2P applications use different strategies to camouflage their traffic in order to evade detection. This method is no longer valid because of the inaccuracy and incompleteness of its classification results [5].

In order to deal with the disadvantages of the above method, payload-based classification method is proposed to inspect the packet payload [6]. Although this solution does can achieve high classification accuracy, it can't work with encrypted traffic or newly P2P applications. On the other hand, signature searching in the payload of every packet produces a high consume of resources.

The host-behavior-based approach is developed to capture social interaction observable even with encrypted payload [7]. However, this method such as BLINC can't classify exactly the applications. It could suppose a problem with applications that are theoretically from different groups but with similar behavior (e.g., Game and P2P).

The other branch that appears to solve the limitations of the network traffic classification methods is flow statistics analysis based on machine learning (ML) [8]. Machine learning algorithms are generally divided into supervised learning and unsupervised learning. Supervised learning requires training data to be labeled in advance and produces a model that fits the training data. Moore et al. [9] used a Naive Bayes classifier which was a supervised machine learning approach to classifying internet traffic. But only 65% accuracy rate, which was not good enough to classify Internet traffic. Williams et al. [10] conducted a comparison of five machine learning algorithms which were widely used to classify empirical study of Internet traffic. Among these algorithms, C4.5 achieved the highest accuracy in their results. Auld [11] proposed supervised machine learning based on a Bayesian neural network to classify the traffic with higher accuracy and better stability, but it was not capable for real-time applications. Unsupervised learning essentially clusters flows with similar characteristics together [12]. The advantage is that it does not require training, and new applications can be classified by examining known applications in the same cluster. Zander et al. [13] extended this work by using an EM algorithm called Auto Class, and found the optimal feature subset for classifying traffic. Erman et al. [14] compared the performance of unsupervised machine learning algorithms in traffic classification. Since our main focus is on evaluating the predictive power of a built/trained traffic classifier rather than on detecting new applications or flow clustering. Bernaille et al. [15] used a simple K-Means clustering algorithm to perform classification by using only the first five packets of the flow to implement online traffic classification.

3. Online Internet Traffic Classification based on Proximal SVM

3.1. Traffic Classification Scheme

Classifiers based on machine learning use a training dataset that consists of *N* tuples (x_i, y_i) and learn a mapping $f(x) \to y$. In the traffic classification context, examples of attributes include flow statistics such as duration and total number of packets. The terms attributes and features are used interchangeably in the machine learning literature. In our supervised Internet traffic classification system, Let $X = \{x_1, x_2, \cdots, x_n\}$ be a set of flows. A flow instance x_i is characterized by a vector of attribute values, $x_i = \{x_{ij} \mid 1 \le j \le m\}$, where *m* is the number of attributes, and x_{ij} is the value of the j^{th} attribute of the i^{th} flow, and x_i is referred to as a feature vector. Also, let $Y = \{y_1, y_2, \cdots, y_q\}$ be the set of traffic classes, where *q* is the number of classes of interest. The y_i can be classes such as "HTTP", "MEDIA", and "P2P". Therefore, our goal is to learn a mapping from an *m*-dimensional variable X to Y.

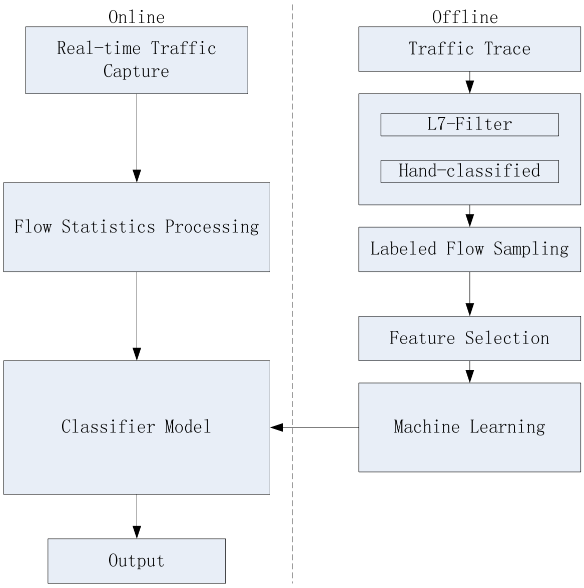

Fig. 1 Architecture of Online Internet Traffic Classification Based on Proximal SVM

As illustrated in Figure 1, the architectture of online Internet traffic classification based on proximal SVM includes two portions: off-line ML modeling and on-line ML classification. In the stage of off-line ML modeling, traffic trace should be labeled to different application using L7-Filter or Hand-classification method. Training subsets are uniformly sampled for the specific ML classification. Once the datasets are ready, the system carries out feature selection to eliminate the redundant andirrelevant features to get optimal feature subset. In the stage of online ML classification, the flows which can be realtime collected are trained using the selected ML classifier. Flow statistics are computed in terms of each selected feature and stored them to the corresponding database in the flow preprocessing module. The output classifier model produced by off-line ML modeling is applied to classify the captured traffic. Eventually, the classification output would be applied to network activities such as network surveillance, QoS.

3.2. Proximal Support Vector Machine

Support vector machine (SVM) based on the statistical learning theory by Vapnik is a new and powerful classification technique and has drawn much attention in recent years [16]. Support Vector Machine (SVM) is known as one of the best machine learning algorithms for classification purpose and has been successfully applied to many classification problems such as image recognition, text categorization, medical diagnosis, remote sensing, and motion classification [17-18]. SVM method is selected as our classification algorithm due to its ability for simultaneously minimizing the empirical classification error and maximizing the geometric margin classification space. These properties reduce the structural risk of over-learning with limited samples. But standard SVM still has some limitations. Proximal support vector machine (PSVM) is proposed instead of SVM, which leads to an extremely fast and simple algorithm for generating a system of linear equations [19]. The formulation of PSVM greatly simplifies the problem with considerably faster computational time than SVM [20].

Assume that a training set S is given as $S=\{(x_1,y_1),\cdots,(x_l,y_l)\}$, where $x_i \in R^n$, and $y_i \in \{-1,+1\}$. The goal of SVM is to find an optimal separating hyperplane

$$w'x-b=0 \tag{1}$$

that classifies training samples correctly or basically correctly, where $w \in R^n$, and the scalar $b \in R^n$. Now to find the optimal separating hyperplane is to solve the following constrained optimization problem

$$\min \frac{1}{2} w'w + ce'\xi \tag{2}$$

$$s.t. \quad D(Aw-eb)+\xi \geq e \tag{3}$$

$$\xi \geq 0 \tag{4}$$

Where $\xi=(\xi_1,\xi_2,\cdots,\xi_l)'$, ξ_i is slack variable, $i=1,2,\cdots,l$, e denotes a column vector of ones of arbitrary dimension, $A=(x_1,x_2,\cdots,x_l)'$, *D* is a diagonal matrix whose entries are given by $D_{ii}=y_i$, and $C>0$ is a fixed penalty parameter of labeled samples. It controls the tradeoff between complexity of the machine and generalization capacity.

PSVM modifies SVM formulation based on maximizing the separating margin $1/w'w+b^2$ in the space of R^{n+l} and changes the slack variables ξ from the L_1 norm to the L_2 one. Note that the nonnegative constraint on the slack variables ξ in (3) is no longer needed. Furthermore, a fundamental change is replacing the inequality constraint with an equality constraint. This leads to the optimization problem as follows

$$\min \frac{1}{2}(w'w+b^2)+c\frac{1}{2}\|\xi\|^2 \tag{5}$$

$$s.t.\quad D(Aw-eb)+\xi=e \tag{6}$$

This modification not only adds advantages such as strong convexity of the objective function, but changes the nature of optimization problem significantly. The planes $w'x-b=\pm 1$ are not bounding planes any more, but can be thought of as "proximal" planes, around which points of the corresponding class are clustered. The formulation of PSVM greatly simplifies the problem and generates a classifier by merely solving a single system of linear equations. However, sometimes the result of PSVM is not accurate when the training set is inadequate or there is a significant deviation between the training and working sets of the total distribution.

4. Experimental Results and Analysis

4.1. Empirical Traces

This subsection describes the empirical traces in our work. The overall network traffic trace consists of three traces. Moore_Set was collected from the experiment of Pro. Moore from Cambridge University; Handmade_Set was simply labeled by manual classification in our laboratory using payload-based technique or port-based method; Univetsity_Set was collected from Nanjing University of Posts and Telecommunications.

Moore_Set trace consists of bidirectional network traffic of some biological research institute during 0 to 24 o'clock on Aug 20th, 2003. The application names of each type and the quality as well as the respective proportion of each network flow are shown in Table 1.

Unlike the usual way to obtain traces, we set a local experimental network with around 100 hosts to generate traffic manually to get Handmade_Set. Let each host run the specific application (HTTP, MAIL, FTP, DATABASE, P2P, GAME, etc.) at the same time. Since the applications run in the host is predetermined, it is easy to classify and categorize the traffic flow by the IP address. Table 2 summarizes the applications in our experiments.

To facilitate our work, we collect traces in all academic units and laboratories on the campus from the Internet gateway of Nanjing University of Posts and Telecommunications. Univetsity_Set was collected over a span of six months from April 10, 2009 to October 10, 2009. Table 3 summarizes the applications found in the 20 1-hour Campus traces.

4.2. Evaluation Metrics

In this paper, *TP*, *FP*, and *FN* are the numbers of true positives, false positives, and false negatives, respectively. True Positives is the number of correctly classified flows, False Positives is the number of flows falsely ascribed to a given application, and False Negatives is the number of flows from a given application that are falsely labeled as another application.

Table 1 Statistics of Moore_Set

Type of flow	Application names	Num of flow	Percent(%)
WWW	http, https	328091	86.91
MAIL	Imap, pop3, smtp	28567	7.567
BULK	ftp	11539	3.056
DATABASE	oracle, mysql	2648	0.701
SERVER	ident,ntp,x11,dns	2099	0.556
P2P	kazaa,bittorrent	2094	0.555
ATTACK	worm, virus	1793	0.475
MEDIA	real, media player	1152	0.305
INT	telnet,ssh,rlogin	110	0.029
GAME	half-life	8	0.002
Total	26 applications	377526	100

Table 2 Statistics of Handmade_Set

Type of flow	Num of flow	Percent(%)
WWW	1000	12.5
MAIL	1000	12.5
BULK	1000	12.5
DATABASE	1000	12.5
SERVER	1000	12.5
P2P	1000	12.5
MEDIA	1000	12.5
GAME	1000	12.5
Total	8000	100

Table 3 Statistics of University _Set

Type of flow	Num of flow	Percent(%)
WWW	4606712	64.44
MAIL	561994	7.86
BULK	11786	0.16
DATABASE	1528681	21.38
SERVER	2876	0.04
P2P	29596	0.43
MEDIA	1698	0.02
GAME	13453	0.19
UNKNOWN	392075	5.48
Total	7148871	100

Accuracy is the ratio of the sum of all True Positives to the sum of all the True Positives and False Positives for all classes. We apply this metric to measure the accuracy of a classifier on the whole trace set. The latter two metrics are to evaluate the quality of classification results for each application class.

$$accuracy = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FP_i} \times 100\% \tag{7}$$

4.3. Comparing Algorithm Performance

In this section, we principally test on different datasets consists of public Moore_Set, Handmade_Set, and Univetsity_Set. For each dataset, we give the 10-fold average testing correctness. In this experiment we compare the performance using LSVM [21], SSVM [22] and SVM methods for classification.

Table 4 Comparing Algorithm Performance

Dataset	PSVM
	Accuracy(%)
	Time(Sec)
		SSVM
		Accuracy(%)
		Time(Sec)
			LSVM
			Accuracy(%)
			Time(Sec)
				SVM
				Accuracy(%)
				Time(Sec)

Moore_Set	91.66	91.83	**92.07**	91.92
	3.6	22.1	38.5	63.7
Handmade_Set	92.73	93.01	**93.08**	92.94
	0.7	6.6	8.4	17.3
Univetsity_Set	90.58	90.73	**91.62**	90.69
	75.3	693.1	812.7	1668.4

As shown in Table 4, bold type indicates the best result, the accuracy of the four algorithms is very similar but the execution time including ten-fold cross validation for PSVM is smaller by as much as one order of magnitude or more than the other three methods tested. In contrast, standard SVMs solve a quadratic or a linear program that requires considerably longer computational time. Computational results on different datasets indicate that the PSVM classifier has comparable test set correctness to that of standard SVM classifiers, but with considerably faster computational time that can be an order of magnitude faster. The PSVM can easily handle large datasets such as network traffic.

4.4. Impact of Selecting Different Kernel Function on Classification Accuracy

Selecting different kernel plays an important role in the SVM-based classification, commonly used kernel functions include LINEAR, POLY, RBF and SIGMOID. Different kernel functions create different non-linear separation surfaces.

To choose the fitted kernel functions, four commonly kernel functions as LINEAR, POLY, RBF, SIGMOID are used to evaluate their classification accuracy. In default, each experiment is repeated on 10 independently sampled datasets to eliminate bias. Results are shown in Table 5. RBF kernel function gives the best classification accuracy. Therefore, RBF kernel function was used in the subsequent experiment.

Table 5 Impact of Selecting Different Kernel Function on Classification Accuracy

Kernel Function	Accuracy(%)
SIGMOID	75.92
LINEAR	85.43
POLY	87.08
RBF	**91.66**

4.5. Impact of Number of Packets for Statistics on Classification Accuracy

In order to classify the network applications associated with a flow as early as possible, we follow the idea presented in Bernaille et al. [15] and conduct experiments to determine the appropriate packet number *p*. The statistics information of several packets in each flow could distinguish network traffic from Internet traffic accurately with the least *p*. For our experiments, we classify network traffic using flows from Moore_Set, Handmade_Set and Univetsity_Set respectively.

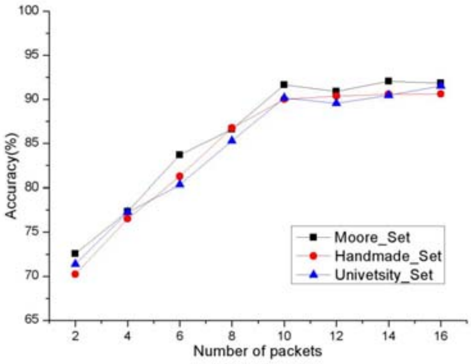

Fig. 2 Impact of Number of Packets for Statistics on Classification Accuracy

The experimental result shown in Figure 2 indicates that proximal SVM algorithm could achieve better accuracy when we choose 10 packets for statistics. It is noticeable that the statistics of the first 10 packets could classify traffic with high accuracy (mostly over 90%) in different traces. At the same time, the classification accuracy improves marginally using more than 10 packets. Considering our goal to detect network traffic as fast as possible with high accuracy, we choose 10 packets for flow feature statistics.

5. Conclusion

Internet traffic classification plays important roles in numerous areas such as network management, traffic engineering, QoS provisioning etc. However, as many newly-emerged P2P applications use dynamic port numbers and masquerading techniques, it causes the most challenging problem in network traffic classification. In this paper, we propose online Internet traffic classification based on proximal SVM. We show that our technique can achieve high classification accuracy with faster computational time. This method is suitable for realtime network traffic classification. Several opportunities exist for future work. Supervised machine learning method is the requirement on a large number of labeled training samples. We can propose semi-supervised classification method to solve this issue. Moreover, we also need more experiments to find out which features are efficient and suitable for improving the classification accuracy.

References

[1] B.Marco, Mellia, Antonio Pescape and LucaSalgarelli, Traffic classification and its applications to modern networks, Computer Networks, 2009,53(6):759-76.

[2] Soysal, Murat,Schmidt, Ece Guran.Machine learning algorithms for accurate flow-based network traffic classification: Evaluation and comparison. Performance Evaluation, 2010, 67(6):451-467.

[3] J. Erman, M. Arlitt, and A. Mahanti. Traffic classification using clustering algorithms. In Proceedings of SIGCOMM workshop on Mining network data, 2006: 281-286.

[4] T. Karagiannis, K. Papagiannaki, and M. Faloutsos. BLINC: multilevel traffic classification in the dark. In Proceedings of the 2005 conference on Applica- tions, technologies, architectures, and protocols for computer communications, pages 229-240. ACM New York, NY, USA, 2005.

[5] Constantinou F, Mavrommantis P. Identifying known and unknown peer-to-peer traffic. In IEE NCA'06 Conference, 2006: 93-102.

[6] S. Sen, O. Spatscheck, and D. Wang. Accurate, scalable in-network identification of p2p traffic using application signatures. In Proceedings of the 13th international conference on World Wide Web, ACM New York, NY, USA, 2004: 512-521.

[7] T. Karagiannis, K. Papagiannaki, N. Taft, and M. Faloutsos. Profiling the end Host. Lecture Notes Computer Science, 2007.

[8] T. Nguyen and G. Armitage. A Survey of Techniques for Internet Traffic Classification using Machine Learning. IEEE Communications Surveys and Tutorials, 2008,11(3):37-52.

[9] A. W. Moore and D. Zuev. Internet traffic classification using bayesian analysis tech- niques. In Proceedings of ACM SIGMETRICS international conference on Measurement and modeling of computer systems, 2005:50-60.

[10] N. Williams, S. Zander, and G. Armitage. A preliminary performance comparison of five machine learning algorithms for practical ip traffic flow classification. ACM SIGCOMM Computer Communication Review,2006, 30(5):5-16.

[11] T. Auld, A. W. Moore, and S. F. Gull. Bayesian neural networks for internet traffic classification. IEEE Transaction on Neural Network,2007,18(1):223-239.

[12] A. Mcgregor, P. Lorier M. Hall, and J. Brunskill. Flow clustering using machine learning techniques. In Passive and Active Network Measurement, 2004: 205 - 214.

[13] S. Zander, T. Nguyen, and G. Armitage. Automated traffic classification and application identification using machine learning. In Proceedings of the IEEE Conference on Local Computer Networks 30th Anniversary, 2005:250 - 257.

[14] J. Erman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson. Offline/realtime traffic classification using semi-supervised learning. Technical report, University of Calgary,2007.

[15] Bernaille L, Teuxeira R, Akodkenous I, Soule A, Slamatian K. Traffic classification on the fly. ACM SIGCOMM Computer Communication Review 2006; 36: 23-26.

[16] Vapnik. SVM method of estimating density, conditional probability, and conditional density.IEEE International Symposium on Circuits and Systems, 2002:749-752.

[17] Yan Hongsen, Xu Duo. An approach to estimating product design time based on fuzzy ν-support vector machine[J]. IEEE Transactions on Neural Networks, 2007, 18(3):721-732.

[18] Sahbi Hichem, Geman Donald. A hierarchy of support vector machines for pattern detection[J]. Journal of Machine Learning Research,2006, 7(10):2087-2123.

[19] Tran Duc A, Nguyen Thinh. Localization in wireless sensor networks based on support vector machines[J]. IEEE Transactions on Parallel and Distributed Systems, 2008, 19(7):981-994.

[20] Hao P Y, Chiang J H. Fuzzy regression analysis by support vector learning approach[J]. IEEE Transactions on Fuzzy Systems, 2008, 16(2):428-441.

[21] Yuh-Jye Lee, O.L.Mangasarian. SSVM: A smooth support vector machine.Computational Optimization and Applications, 2001, 20(1):5-22.

[22] O.L.Mangasarian, D.R.Musicant. Lagrangian support vector machines. Journal of Machine Learning Research, 2001,1(3):161-177.