A New Method of Transductive SVM-Based Network Intrusion Detection

Manfu Yan, Zhifang Liu

A New Method of Transductive SVM-based Network Intrusion Detection

Manfu Yan[1] Zhifang Liu[2]

[1]Department of Mathematics, Tangshan Teacher's College, Tangshan Hebei, China
Email: 3005@tstc.edu.cn
[2]Network Technology Center, Tangshan Teacher's College, Tangshan Hebei, China
Email: lzf@tstc.edu.cn

Abstract. Based on the existing Transductive SVM and via introducing smooth function $P(\Delta,\lambda)$ to construct smooth cored unconstrained optimization problem, this article will build the optimization model accessible to degenerate solutions to generate an improved transductive SVM, introduce simulated annealing to degenerate the optimization problem, and apply such a Support Vector Classifier to generate a new method of network intrusion detection.

Key words: optimization; unconstrained problem; transductive SVM; network intrusion detection; simulated annealing.

1. Introduction

Network intrusion detection is a safe mechanism with dynamic monitoring, prevention or resistance against network intrusion [1]. The network-intrusion detection system can be used to discover and identify the behavior and attempt of intrusion in the system via monitoring and analyzing the network flow and system audit records to give out an alarm of intrusion in order to facilitate the administer to take effective measures to mend the loopholes of the system and fill up the system [2].

Network intrusion detection is used to separate user behavior's normal data of from its abnormal data, which essentially can be regarded as the classification. The data to describe the behaviors of users is of multi-index as usual. Therefore, it can be expressed with an n-dimensional vector. In this way, the network-intrusion detection problem can be summarized as data group for normal or abnormal behavior of users, i.e. two kinds of classification problems of n-dimensional vector, which can help create the detection methods and system via the support vector classifier. But studies and practices indicate that as to the network-intrusion detection problem, the methods and system built by the use of ordinary SVM (e.g. C-SVM) are not desirable in the precision of detection. Thus, we try to apply the transductive support vector classifier to create the detection methods and introduce the simulated annealing method to degenerate the optimized model.

2. The Improvement of Transductive Support Vector Machine

Generally, for Support Vector Machine, it is set up by given trainingset

$$T=\{(x_1,y_1),\cdots,(x_l,y_l)\} \qquad (1)$$

Here, $x_i\in\mathcal{X}=R^n, y_i\in\mathcal{Y}=\{1,-1\}, i=1,2,\cdots,l$. We normally call them Inductive Support Vector Machine [3] .

Vapnik has discussed a kind of sorting algorithm for Transductive Support Vector Machine, it is different from Inductive Support Vector Machine. It provides a mutual independent set, which follows joint distribution, besides the given trainingset T.

$$S=\{x_1^*,\cdots, x_m^*\}, \qquad (2)$$

Here $x_1^* \in \mathcal{X} = R^n$.

For TSVM, we want to find a optimized function $f(x, w_0)$ from a particular function set $F = \{f(x, w)\}$, so that the risk

$$R(w) = \frac{1}{m}\sum_{i=1}^{m} L\left(y_i^*, f\left(x_i^*, w\right)\right) \tag{3}$$

is minimized. Here w is a general parameter of the function, $L(y, f(x, w))$ represents the loss due to estimation of y by $f(x, w)$, that is to say, here we interested in the function value of $f(x, w_0)$ on given fixed points x_i^*, not all the function values within the field of definition.

2.1 Unconstraint Problem

Initially, the optimization of TSVM is [4]

$$\min_{w\in H, b\in R, y^*\in R, \xi, \xi^*} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i + C^*\sum_{j=1}^{m}\xi_j^* \tag{4}$$

$$\text{s.t.} \quad y_i\left(\left(w\cdot x_i\right)+b\right) \geq 1-\xi_i, i=1,\cdots,l, \tag{5}$$

$$y_j^*\left(\left(w\cdot x_j^*\right)+b\right) \geq 1-\xi_j^*, j=1,\cdots,m, \tag{6}$$

$$\xi_i \geq 0, i=1,\cdots,l, \tag{7}$$

$$\xi_j^* \geq 0, j=1,\cdots,m, \tag{8}$$

We would like to simplify initial problem (4)—(8) in this section, it is reduced to unconstraint problem.

Theorem Consider the solution for (4)—(8), it must satisfy

$$y_j^*\left(\left(w\cdot x_j^*\right)+b\right) \geq 0 \tag{9}$$

for all x_j^*

Prove: Suppose the solution for the problem is $(w, b, \xi, \xi^*, y_1^*, \cdots, y_m^*)$, if for some x_j^*, $(w\cdot x_j^*)+b \geq 0$, then $y_j^* = 1$. Since when $y_j^* = 1$, $y_j^*\left(\left(w\cdot x_j^*\right)+b\right) \geq 0, \xi_j^* \geq 1-y_j^*\left(\left(w\cdot x_j^*\right)+b\right)$, as minimized objective function, it must satisfy $\xi_j^* = 0$ or $0 \leq \xi_j^* = 1-y_j^*\left(\left(w\cdot x_j^*\right)+b\right) < 1$.

When $y_j^* = -1$, $y_j^*\left(\left(w\cdot x_j^*\right)+b\right) \leq 0, \xi_j^* \geq 1,$ which is large than then objective function value when $y_j^* = 1$.

Similarly, for some x_j^*, $\left(w\cdot x_j^*\right)+b \leq 0$, then $y_j^* = -1$

Namely, for all x_j^*, $y_j^*\left(\left(w\cdot x_j^*\right)+b\right) \geq 0,$

Based upon the theorem above, we can change the constraint (6) and (8) of problem (4)—(8) into

$$\xi_j^* = \left(1-\left|\left(w\cdot x_j^*\right)+b\right|\right)_+, \quad j=1,2,\cdots,m \tag{10}$$

However, for variable $\xi_i, i=1,\cdots,l,$ we got

$$\xi_j = \left(1-y_i\left(\left(w\cdot x_i\right)+b\right)\right)_+, \quad i=1,2,\cdots,l \tag{11}$$

Here, function $(\cdot)_+$ is single variable function,

$$(\Delta)_+ = \begin{cases} \Delta, \Delta \geq 0; \\ 0, \Delta < 0 \end{cases} \tag{12}$$

Base on this, we could convert problem (4)—(8) into unconstraint optimization.

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(1-y_i((w\cdot x_i)+b))_+ + C^*\sum_{j=1}^{m}(1-\left|(w\cdot x_j^*)+b\right|)_+ \qquad (13)$$

2.2 Smooth Unconstraint Problem

Since unconstraint problem(13) is not smooth，it is not able to be solved by regular optimization method. As a result, we think about modifying the second and third part of objective function for problem (13), so that it becomes smooth, in order to construct a smooth unconstraint problem that is similar to unsmooth and unconstraint problem (13). Because of that, we introduce an approximate function for unsmooth function $(\Delta)_+$

$$P(\Delta,\lambda) = \Delta + \frac{1}{\lambda}\ln(1+e^{-\lambda\Delta}), \qquad (14)$$

Here, parameter λ>0, obviously the function above is smooth; we could prove it as well. When $\lambda\to\infty$, function $P(\Delta,\lambda)$ converges at $(\Delta)_+$ such that the second part of unconstraint optimization (13) is transformed into

$$C\sum_{i=1}^{l}P(1-y_i((w\cdot x_i)+b),\lambda). \qquad (15)$$

and the third part is transformed to

$$C^*\sum_{j=1}^{m}P(1-\left|(w\cdot x_j^*)+b\right|,\lambda) \qquad (16)$$

The unsmooth term $|\Delta'|$ is still inside (16), so we decide to use following function to approximate $|\Delta'|$ smoothly.

$$P'(\Delta',\mu) = \Delta' + \frac{1}{\mu}\ln(1+e^{-2\mu\Delta'}) \qquad (17)$$

We could deduce some the following theorem by making use of the properties of $P(\Delta,\lambda)$:

Now, unconstraint problem (13) approximates optimization problem

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}P(1-y_i((w\cdot x_i)+b),\lambda) + C^*\sum_{j=1}^{m}P(1-P'((w\cdot x_j^*)+b,\mu),\lambda). \qquad (18)$$

When λ,μ is large enough, the solution of smooth unconstraint problem (18) will most approximate unsmooth unconstraint problem (13).

2.3 Smooth Unconstraint Problem with Kernel

If we take account of linear partition of input space, we could introduce a mapping from input space X to Hilbert space H

$$\Phi: \begin{matrix} X \to H \\ x \to X = \Phi(x) \end{matrix} \qquad (19)$$

and kernel function

$$K(x,x') = (\Phi(x)\cdot\Phi(x')), \qquad (20)$$

Apply l_1 module $\|w\|_1$ on objective function of problem (4)---(8), we got optimization problem

$$\min_{w,b,\xi,\xi^*} \|w\|_1 + C\sum_{i=1}^{l}\xi_i + C^*\sum_{j=1}^{m}\xi_j^* \qquad (21)$$

$$\text{s.t.} \quad y_i((w\cdot x_i)+b)\geq 1-\xi_i, i=1,\cdots,l, \tag{22}$$

$$y_j^*((w\cdot x_j^*)+b)\geq 1-\xi_j^*, j=1,\cdots,m, \tag{23}$$

$$\xi_i\geq 0, \xi_j^*\geq 0, i=1,\cdots,l, j=1,\cdots,m. \tag{24}$$

We know that if β,β^* is the solution of dual problem for problem(4)—(8), then the solution of initial problem(21)—(24) to W could approximately represented as

$$W=\sum_{i=1}^{l} y_i\beta_i X_i+\sum_{j=1}^{m} y_j^*\beta_j^* X_j^*=\sum_{i=1}^{l}(\alpha_i-\bar{\alpha}_i)\Phi(x_i)+\sum_{j=1}^{m}(\alpha_j^*-\bar{\alpha}_j^*)\Phi(x_j^*). \tag{25}$$

We could alter problem (21)—(24) to following problem, by making use of above expression

$$\min_{\alpha,\bar{\alpha},\alpha^*,\bar{\alpha}^*,b,\xi,\xi^*,} \sum_{i=1}^{l}(\alpha_i+\bar{\alpha}_i)+\sum_{j=1}^{m}(\alpha_j^*+\bar{\alpha}_j^*)+C\sum_{i=1}^{l}\xi_i+C^*\sum_{j=1}^{m}\xi_j^* \tag{26}$$

$$y_i(\sum_{k=1}^{l}(\alpha_k-\bar{\alpha}_k)K(x_k,x_i)+\sum_{k=1}^{m}(\alpha_k^*-\bar{\alpha}_k^*)K(x_k^*,x_i)+b)\geq 1-\xi_i, i=1,\cdots,l, \tag{27}$$

$$y_j^*(\sum_{k=1}^{l}(\alpha_k-\bar{\alpha}_k)K(x_k,x_j^*)+\sum_{k=1}^{m}(\alpha_k^*-\bar{\alpha}_k^*)K(x_k^*,x_j^*)+b)\geq 1-\xi_j^*, i=1,\cdots,l, \tag{28}$$

$$\xi_i\geq 0, i=1,\cdots,l, \tag{29}$$

$$\xi_j^*\geq 0, j=1,\cdots,m, \tag{30}$$

Here we use $\sum_{i=1}^{l}(\alpha_i+\bar{\alpha}_i)+\sum_{j=1}^{m}(\alpha_j^*+\bar{\alpha}_j^*)$ to replace $\|w\|_1$.

The method is similar to that of previous section，we could transform problem (26)—(30) into smooth unconstraint optimization problem by introducing smooth function $P(\Delta,\lambda)$ and $P'(\Delta',\mu)$.

$$\begin{aligned}\min_{\alpha,\bar{\alpha},\alpha^*,\bar{\alpha}^*,b,\xi,\xi^*,} \quad & \sum_{i=1}^{l}(\alpha_i+\bar{\alpha}_i)+\sum_{j=1}^{m}(\alpha_j^*+\bar{\alpha}_j^*)\\ & +C\sum_{i=1}^{l}P(1-y_i(\sum_{k=1}^{l}(\alpha_k-\bar{\alpha}_k)K(x_k,x_i)+\sum_{k=1}^{m}(\alpha_k^*-\bar{\alpha}_k^*)K(x_k^*,x_i)+b),\lambda)\\ & +C^*\sum_{j=1}^{m}P(1-P'(\sum_{k=1}^{l}(\alpha_k-\bar{\alpha}_k)K(x_k,x_i)+\sum_{k=1}^{m}(\alpha_k^*-\bar{\alpha}_k^*)K(x_k^*,x_j^*)+b),\mu),\lambda)\end{aligned} \tag{31}$$

When λ,μ is large enough, above problem is similar to problem(26)—(30)，consequently, we could create decision function after we got the optimum solution ($\alpha,\bar{\alpha},\alpha^*,\bar{\alpha}^*,b,\xi,\xi^*$)of this problem.

$$f(x)=\operatorname{sgn}(\sum_{k=1}^{l}\alpha_k-\bar{\alpha}_k)K(x_k,x)+\sum_{k=1}^{m}(\alpha_k^*-\bar{\alpha}_k^*)K(x_k^*,x)+b), \tag{32}$$

Additionally, using this decision function to decide the category of the points in test set S.

2.4 Conclusion：Improvement of TSVM

a) Assume known trainingset $T=\{(x_1,y_1),\cdots,(x_1,y_1)\}$, here $x_i\in X=R^n, y_i=\{-1,1\}, i=1,\cdots,l;$ known test set $S=\{x_1^*,\cdots,x_m^*\}$, here $x_i^*\in X=R^n$;

b) Choose suitable parameter C and C^*, choose suitable kernel function $K(x,x')$; construct and find the unconstraint problem (31), thus got optimum solution $(\tilde{\alpha},\tilde{\bar{\alpha}},\tilde{\alpha}^*,\tilde{\bar{\alpha}}^*,\tilde{b})$;

c) Create decision function：

$$f(x)=\mathrm{sgn}(\sum_{k=1}^{l}\tilde{\alpha}_k-\tilde{\tilde{\alpha}}_k)K(x_k,x)+\sum_{k=1}^{m}(\tilde{\alpha}_k^*-\tilde{\tilde{\alpha}}_k^*)K(x_k^*,x)+\tilde{b})$$

Thereby, for any test point belongs to S, the decision function will provide the category for it.

3. New Method on Network Intrusion Detection

As the objective function of problem 31 has a continuous gradient and hesse matrix, as well as unconstrained, it can be solved by basic algorithm to unconstrained problem. However its objective function is not convex function thus a number of local optimal solutions may exist, and through the general unconstrained algorithm may not acquire global optimal solution. In the following global optimal algorithm – simulated annealing algorithm will be introduced to solve problem (31), and the model will be introduced to network intrusion detection.

3.1 Simulated Annealing Algorithm

First, we will introduce simulated annealing algorithm. Simulated annealing algorithm is a kind of random search method known as Monte Carlo method, which allows the objective function to have random changes in the increasing direction. Therefore, simulated annealing algorithm can jump out of local minimum point. This algorithm was proposed by Metropolis as early as 1953, originated from simulation to solid annealing process. The annealing process starts at a certain high enough temperature, and almost every random motion are acceptable under this temperature. Then the temperature decreases slowly according to some cooling rule and tends to zero. Enough time is needed for the system to reach a stable state at each temperature point, and finally places in a state with lowest energy, to obtain a relative global optimal solution to the optimization problem. In which, one solution x_k to the optimization problem and its target value $f(x_k)$ correspond to a solid microstate k and its energy E_k respectively. The temperature T in the annealing process is a control parameter decreasing by the algorithm process. The algorithm adopts Metropolis acceptance criteria. In each step of the algorithm, a new candidate solution generates randomly. If the new solution decreases the objective function, it is acceptable; otherwise whether to accept it will be decided in form of exponential probability. Probability P to accept the new solution is:

$$p=\begin{cases}\exp(-\Delta f/T) & -\Delta f>0,\\ 1 & -\Delta f\le 0.\end{cases} \qquad (33)$$

In which, Δf is the variation of objective function caused by random disturbance and T represents temperature. From formula (33) we can see that for a given Δf, when T is relatively high, acceptance probability to the new solution which increases the function is larger than the probability when T is relatively low. Thus the entire algorithm keeps the iterative process of "generate new solution – judge – accept or discard" till find the optimal solution finally. The specific algorithm is as follows:

Algorithm A. Simulated annealing algorithm

a) Suppose k=0, T= T_0, in which T_0 is the initial temperature. Parameter L and initial value x_0 are given;
b) Generate a new candidate x_{k+1} by random disturbance of x_k;
c) Calculate $\Delta f = f(x_{k+1}) - f(x_k)$;
d) If $\Delta f \le 0$, accept the new solution $x_{k+1}=x_k$. If the stop criteria is satisfied, the algorithm stops and $x= x_{k+1}$; otherwise give a random value λ in the range of 0 to 1 obeying uniform distribution. If $\exp(-\Delta f/T)>\lambda$, accept the new solution $x_{k+1}=x_k$;
e) Suppose k=k+1, if $k\le L$, jump to step b);
f) Decrease T_0 according to temperature cooling rule. Suppose $x_0=x_k$ and k=0, and jump to step b).

In the above algorithm, parameters we need to select include the initial temperature value T_0. Simulated annealing algorithm requires a large enough T_0 to ensure jumping out of the local optimal solutions, that is to ensure $\exp(-\Delta f/T_0)\approx 1$. Selection of a too large T_0 will cause a too long algorithm period, while a too small T_0 will cause the algorithm traps in the local optimal solution too early. The other parameters need to be selected are iteration times L under each temperature, initial solution x_0. Besides the decreasing rule of temperature T also needs to be known,

generally taken as $T_{k+1}=\beta T_k, 0<\beta<1$; and the final value of temperature, actually the final temperature value often chose as close to 0.

3.2 Network Intrusion Detection

As the widespread use of network, log data is very large. Some network attacks such as DNS spoofing, denial of service, port scanning, etc. are generally very difficult to be directly discovered. Using data mining technology, normal and abnormal action model can be acquired from massive logs, and then detect intrusion action [5]. Data mining technology commonly used in intrusion detection system includes neural network, genetic algorithm [6] and so on. There are also researchers using support vector machine to conduct some tests on actual intrusion detection [7].

In essence, intrusion detection is actually a classification problem, that is to separate the normal action data and abnormal action data of users through detection. In which the data describing user action is often multi-index. The feasibility of using support vector machine to conduct intrusion detection has been verified in [7]. As we focuses only on behavior of the current user, thus hereby we attempt to use improved deduction support vector machine to discuss the new method of network intrusion detection.

Data adopted in the test is a batch of network connection record set [8]. This batch of original data is a recovered connection information based on the data obtained in IDS evaluation by U.S. Defense Advanced Research Projects Agency (DARPA) in 1998 [9], including 7 weeks' network traffic with about 5 million connection records, in which there were a large number of normal network traffic and various attacks, thus has a strong representative. As the amount of data is significantly large, here we only select attacks of DOS type to conduct our experiments, and determine the dimension number of the problem is 18 according to detection attribute set needed by DOS type attack provided by [9]. 200 normal connection information data are extracted from the original data set as the positive-type set, and 200 DOS type attack connection information data are extracted as the negative-type set, then the positive-type point set and negative-type point set are randomly separated into training set and test set according to some ratio (6:4).

C-support vector machine [10] and improved deduction support vector machine are used respectively to solve the classification problem of the above composition. When solving optimization problem in the improved deduction support vector machine, simulated annealing algorithm introduced in last section is used. Both models adopt RBF kernel function. During the test, the parameter C, C* , as well as σ in RBF kernel function adopt multiple values respectively. Given different combinations of these parameters, test the performance of the two algorithms under different combinations. Table 6.1 is a test result under one group of the combinations as $C=C^*=100$ and $\sigma=2$, in which detection accuracy is the ratio of correctly detected samples in the test set to the total number of samples in the test set; false positive rate is the ratio of normal samples that are mistaken detected to abnormal samples to total number of normal samples; detection rate is the ratio of detected abnormal samples to total number of abnormal samples.

Table 1. Result comparison

Detection Result	C—SVC	Algorithm2.4
Detection Accuracy	79.9%	81.2%
False Positive Rate	0.54%	0.47%
Detection Rate	77.5%	80.1%

Data in the table indicate that, usage of improved deduction support vector machine can obtain higher detection accuracy. Certainly, the test result depends on rational selection of parameters. In practical applications, cross validation or LOO error method (can refer to [3]) can be adopted to determine optimal parameters.

4. Conclusion and Perspective

With reference to the results of the said discussion, the detection precision of improved transductive SVM instead of C-SVM is higher, indicating that only by studying and creating SVM intentionally according to the particularity of problems when using SVM to solve practical problems can it achieve more desirable effects of application. As for

solving the network intrusion problem, the results of detection can depend on the proper selection of parameters besides the selection and study of the varieties of SVM. And in fact, it is available to adopt the cross validation or LOO err (see [3]) to determine the optimized parameter, which is also one of the directions to study on transductive SVM in the future. Besides SVM, neural net and genetic algorithm can be applied to conduct the network intrusion detection via the data extraction technology. And it is required to make a study of these technologies applied to the detection on network intrusion and make comparisons between them, and even compare them with the detection methods of other technologies to further argument the advantages of these new methods, all of which are the problems requiring further studies on theory and practice.

References

[1]Jianchun Jiang, et. al.: Study and Review on Network Intrusion Detection, Journal of Software, 2001, 11

[2]Yuanming Nei, et. al: Network Information Safety Technology [M] Beijing: Science Press, 2001

[3]Naiyang Deng, Yingjie Tian. Support Vector Machine—A New Method in Data Mining. Beijing: Science Press, 2004: 77-162, 224-272. (in Chinese)

[4]Manfu Yan. Support Vector Machines for Classification and Its Application. Beijing: China Agricultural University, 2005(Doctor's Degree Paper)

[5]Lee W, stolfo S. Data mining approaches for intrusion detection [EB/OL]. http://www.cs.columbia.edu~wenke: papers/usenix/usenix.html, 2000-10-12/2002-03-01

[6]Balajinath B, Raghavan S.V. Intrusion detection through learning behavior model. Computer Communication, 2001, 24(12):1202-1212

[7]Hui Li, et al. SVM-based Network Intrusion Detection. Journal 0f Computer Research and Development. VOL 40, No 6, 2003,6

[8]http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[9]Lee W, stolfo S, Mok K.W. A datamining framework for building intrusion detection medels. The 1999 IEEE Symposium on Security and Privary, Oakland, CA, 1999

[10]Naiyang Deng, Yingjie Tian. Support Vector Machine – Theory, Algorithm and Expansion. Beijing: Science Press 2009, p97-p98