Proceedings of the First International Conference on Machine Learning and Cybernetics, Beijing, 4-5 November 2002

EVOLVING SUPPORT VECTOR MACHINE PARAMETERS

ANH TRAN QUANG, QIAN-LI ZHANG, XING LI

Department of Electronics Engineering, Tsinghua University, China
E-MAIL: qa00@mails.tsinghua.edu.cn, zhang@cernet.edu.cn, xing@cernet.edu.cn

Abstract:

The kernel type, kernel parameters and upper bound C control the generalization of Support Vector Machines. The best choice of kernel or C depends on each other and the art of researchers. This paper presents a general optimization problem of Support Vector Machine parameters including a mixed kernel and different upper bounds for unbalanced data. The objectives are ξα-estimators of error rate, recall and precision. Evolutionary Algorithms are used to solve the problem. The performance of this method is illustrated with a standard data set of Intrusion Detection application.

Keywords:

Support vector machine; Evolutionary algorithm; Mixed kernel; Training evaluation; Intrusion detection

1 Introduction

SVM's (Support Vector Machines) are learning machines that implement the empirical risk minimization principle in learning theory [1]. To guarantee a high rate of generalization of the learning machine, i.e. ability to predict unseen data, one has to construct a structure

$$S_1 \subset S_2 \subset , \dots , \subset S$$

on the set of decision functions $S = \{Q(z, \alpha\},\alpha \in \Lambda\}$ and then choose both an appropriate element S_k of the structure and a function $Q\{z, \alpha^k{}_m\} \in S_k$ within this element that minimizes risk bound. The bound can be written as follows:

$$R(\alpha^k{}_m) \leq R_{emp}(\alpha^k{}_m) + \Omega\,(l/h_k) \qquad (1)$$

where the first term is empirical risk and the second is the confidence interval (l is the training data set size and h_k is the VC dimension of S_k).

In SVM, kernels are used to map the input vectors into high dimension feature space and a structure of decision function set is constructed in this space by Δ-margin separating hyperplanes. Thus, the kernels decide the flexibility of the decision function set, in other words they affect both empirical risk and confidence interval values in inequality (1). One can improve SVM generalization ability by choosing appropriate kernel [2]. However, the choice of kernel type and its parameters depends in many respects on the art of the researcher.

As the chosen structure element S_k expands, the minima of empirical risk are decreased, but the term responsible for the confidence interval is increased. The SVM uses an upper bound C to control a tradeoff between the minimal of empirical risk and the confidence interval [1]. When the kernel is fixed, one can find an appropriate value of C by picking different value of it [3]. Yet, the best choice of C depends on the choice of kernel.

EA's (Evolutionary Algorithms) are search methods for optimization problems, in which a mechanics of natural evolution principle is used to obtain the global optimal solution. EA's have demonstrated considerable success in combination with other methods, such as Neural Networks [4] and Fuzzy Systems [5]. In this paper, evolutionary approach is used to optimize SVM parameters.

The remainder of this paper is organized as follows. Section 2 presents optimization problem. Section 3 describes algorithm to solve the problem. Section 4 shows experiment result with a practical data set. Finally, we conclude with remarks and plans for future work.

2 Optimization Problem Representation

A. SVM parameter general model

The SVM generalization ability is controlled by kernel type, kernel parameters and upper bound C. Three typical kernel kinds are:

Polynomial: $K_{poly}(\boldsymbol{u}, \boldsymbol{v}) = (\sigma_1 * \boldsymbol{u} \cdot \boldsymbol{v} + r_1)^d$

RBF: $K_{rbf}(\boldsymbol{u}, \boldsymbol{v}) = \exp(-|\boldsymbol{u} - \boldsymbol{v}|^2/\sigma_2)$

Sigmoid: $K_{sig}(\boldsymbol{u}, \boldsymbol{v}) = \tanh(\sigma_3 * \boldsymbol{u} \cdot \boldsymbol{v} + r_3)$

Every kernel has its advantages and disadvantages and a mixed kernel approach was introduced [2]. In this paper, the kernel is a convex combination of the three kernels above. The convex combination kernel is written in the following forms:

$$K = \lambda_1 K_{poly} + \lambda_2 K_{rbf} + \lambda_3 K_{sig}$$

Where

$$0 \leqslant \lambda_1, \lambda_2, \lambda_3 \leqslant 1$$
$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$

To be an admissible kernel in SVM, kernels must satisfy Mercer's Theorem [1]. Since K_{poly}, K_{rbf} and K_{sig} all satisfy Mercer's Theorem, a convex combination of them also satisfy Mercer's Theorem. Thus our kernel is admissible. In addition, for some classification problems, numbers of data in different classes are unbalanced. Hence, a method using different upper bound C+ and C- for each class was proposed [6]. Therefore a general model of parameters using in a SVM training process are represented as a vector

$$V = \{\lambda_1, \lambda_2, \lambda_3, \sigma_1, r_1, d, \sigma_2, \sigma_3, r_3, C-, C+\}.$$

B. SVM performance measures

0-7803-7508-4/02/$17.00 ©2002 IEEE

Three typical performance measures of learning machine are: error rate, recall and precision [8]. Each measure has its advantages and disadvantages. It is used in different applications. Leave-one-out is a popular method to examine these performance measures [7]. While the method is usually very accurate, it is very expensive to compute. $\xi\alpha$-estimator is a particular method for the case of SVM that overcomes this problem. It uses an upper bound of the number of leave-one-out errors instead of calculating them brute force. For a training data set

$$\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_l, y_l)\}$$

where $\mathbf{x}_i \in \mathbf{R}^n$, $y \in \{-1,1\}$, the $\xi\alpha$-estimator of error rate, recall and precision can be written as follows [8],

Error rate: $Err_{\xi\alpha} = d/l$

Recall: $Rec_{\xi\alpha} = 1 - d_+/l_+$

Precision: $Prec_{\xi\alpha} = (l_+ - d_+) / (l_+ - d_+ + d_-)$

Where

$l_+ = |\{i : y_i = 1\}|$

$d = |\{i : (2\alpha_i R^2_\Delta + \xi_i) \geqslant 1\}|$

$d_+ = |\{i : y_i = 1 \wedge (2\alpha_i R^2_\Delta + \xi_i) \geqslant 1\}|$

$d_- = |\{i : y_i = -1 \wedge (2\alpha_i R^2_\Delta + \xi_i) \geqslant 1\}|$

R^2_Δ is an upper bound on $K(\mathbf{x}_i, \mathbf{x}_i) - K(\mathbf{x}_i, \mathbf{x}_j)$ for all $\mathbf{x}_i$, $\mathbf{x}_j$. ξ is the vector of training losses. $\boldsymbol{\alpha}$ is solution of the dual SVM training problem. Note, ξ and $\boldsymbol{\alpha}$ are both available after training SVM at no extra cost.

C. Optimization problem

Since there are three measures of SVM performance, searching for the optimal solution of SVM parameters is a multi-objective programming. Three objective functions are $Err_{\xi\alpha}(V)$, $Rec_{\xi\alpha}(V)$ and $Prec_{\xi\alpha}(V)$, which are the $\xi\alpha$-estimators of error rate, recall and precision respectively. The optimization problem can be written in the following forms,

$$\begin{aligned} &\text{Max } G\,(1-Err_{\xi\alpha}(V), Rec_{\xi\alpha}(V), Prec_{\xi\alpha}(V)) \\ &\text{subject to:} \qquad (2) \\ &\quad 0 \leqslant \lambda_1, \lambda_2, \lambda_3 \leqslant 1 \\ &\quad \lambda_1 + \lambda_2 + \lambda_3 = 1 \\ &\quad \sigma_1, r_1, d, \sigma_2, \sigma_3, r_3, C-, C+ \geqslant 0 \end{aligned}$$

where V is a vector representing the SVM parameters. G is a compromise model of the objective functions. A well-known compromise model is set up by weighting the objective function

$$G = \rho_1(1-Err_{\xi\alpha}) + \rho_2 Rec_{\xi\alpha} + \rho_3 Prec_{\xi\alpha}$$

where weight ρ_1, ρ_2, ρ_3 are nonnegative numbers with $\rho_1+\rho_2+\rho_3 = 1$.

3 Algorithm

EA's is used to solve the multi-objective problem (2). First, we define SVM parameters V as a chromosome. The fitness of each chromosome is determined by the objectives: $\xi\alpha$-estimators of error rate, recall and precision of SVM using the chromosome as its parameters. The chromosomes are processed by evolution functions, i.e. crossover, mutation and selection, to produce the optimal solution. The evolutionary procedure can be written as follows:

Step 1: Initialize chromosomes at random.

Step 2: Update the chromosomes by crossover and mutation operations.

Step 3: Calculate the objective values $\xi\alpha$-estimators of error rate, recall and precision of SVM using each chromosome as its parameters.

Step 4: Compute the fitness of each chromosome according to the objective values.

Step 5: Select chromosomes for the next generation.

Step 6: Repeat the second to fifth steps for a given number of cycles.

Step 7: Report the best chromosome as the optimal solution.

In step 1 and 2, the initialized and updated chromosomes must be satisfied the constraints in (2). The crossover, mutation and selection function in step 2 and 4 affect the algorithm convergence process. Appropriate designs of these functions can improve the convergence time. The fitness is computed by weighting the objective values. The choice of weights depends on a particular application.

4 Experiments

In this section, experiment of evolving SVM is carried out upon a data set of ID (Intrusion Detection) application. Network-based attack is a computer attack that exploits the vulnerabilities of network services. Misuse detection [9] is a typical technique of IDS that can detect network-based attacks by searching for specific patterns in the data of network packets. However, this technique cannot detect a new kind or variance of attacks. Usually, a network-based attack relates to one or some network connections. The behaviors of attack-related connections differ to the normal ones. In this section, SVM is used to categorize normal and attack-related network connections. A standard IDS evaluation data source is provided by the Massachusetts Institute of Technology, Lincoln Lab [10]. According to the provided description of the attack activities embedded in the data, we obtain a training data set consisted of 200 normal connections and 50 attack-related connections. For each connection, we extract the same features as described in [11]. Normal and attack-related connections are labeled -1 and 1 respectively. SVM training code is a modification of LIBSVM [12].

Evolutionary experiments on different selections of objective weights (i.e. ρ_1, ρ_2, ρ_3) are carried out. After about 300 generations of evolution, the optimal solutions of different objective weights are brought out as shown in table 1

Table 1. Optimal solutions of different objective weights

Obj. weights			Kernel parameters									Upper bounds		Obj.
ρ_1	ρ_2	ρ_3	λ_1	λ_2	λ_3	σ_1	r_1	d	σ_2	σ_3	r_3	C-	C+	G
1.0	0.0	0.0	0.1	0.6	0.3	0.7	61.0	1.8	0.03	34.4	63.5	670.4	293.0	0.886
0.6	0.2	0.2	0.2	0.5	0.3	3.0	64.5	1.2	0.05	68.2	78.6	543.9	173.6	0.874
0.4	0.3	0.3	0.4	0.4	0.2	6.0	84.5	0.9	0.04	98.3	76.7	742.4	325.6	0.865
0.0	0.5	0.5	0.5	0.4	0.1	0.4	38.0	1.9	0.07	10.0	40.9	997.0	255.7	0.805

As described above, ρ_1, ρ_2, ρ_3 are the weights of the objective 1- Err, Rec and Precision and λ_1, λ_2, λ_3 are the proportions of the polynomial, RBF and Sigmoid kernel in the mixture kernel respectively. The results show that, for almost the optimal solutions, the proportion of RBF kernel is the highest. As the weight of the objective Err increases, the proportion of polynomial kernel increases and the proportion of the sigmoid kernel decreases. It also demonstrates that a polynomial kernel has good ability to increase the recall and precision where a sigmoid kernel has good ability to decrease the error rate. Figure 1 illustrates these characteristics.

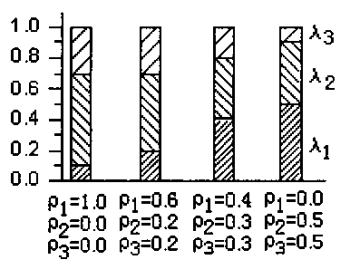

Fig.1. Proportion of different kernels in the optimal solutions

For the case ρ_1=0.6, ρ_2=0.2, ρ_3=0.2, the evolutionary progresses of parameter λ_1, λ_2, λ_3 and the objective are illustrated in figure 2. We can see that after about 100 generations, the objective is larger than 0.85 and the parameter λ_1, λ_2, λ_3 start to converge.

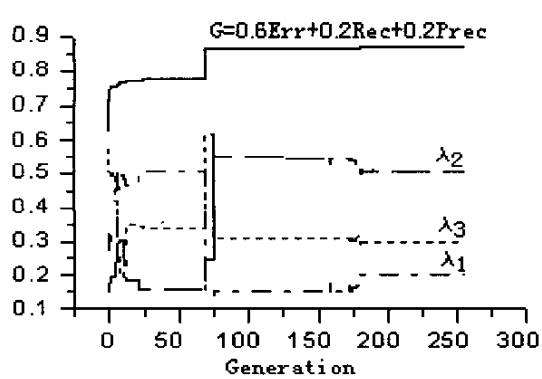

Fig.2. The evolutionary progresses of λ_1, λ_2, λ_3 and objective

The evolutionary progresses of the upper bounds C- and C+ are shown in figure 3. We can see that the evolutionary curve of C- and C+ are very similar. The ratio of C- to C+ seems to be constant, for some extent. The ratio should be depended on the unbalanced data. This characteristic can be used to improve the convergence speed by designing a better mutation of crossover function.

In fact, all the other parameters converge after a certain number of generations. The convergence characteristics are depended on the crossover, mutation and selection functions in the EA's.

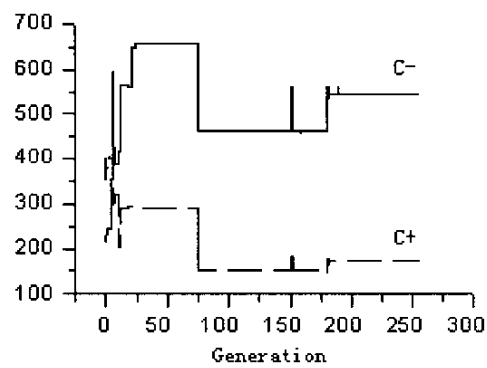

Fig.3. The evolutionary progresses of C- and C+

5 Conclusions

EA's have a good ability to solve the multi-objective programming of SVM parameters optimization problem. The chromosome is coded as a general model of SVM parameters, which is represented by a mixture of kernels and different upper bounds for unbalance data. We use $\xi\alpha$-estimators to calculate the objectives: error rate, recall and precision of each SVM training process. The fitness of each chromosome is computed by weighting the objective function.

For a training data set of IDS application, RBF kernel has a high proportion in the mixture kernel of the optimal solutions. A polynomial kernel has good ability to increase the recall and a sigmoid kernel has good ability to decrease the error rate. All the parameters converge after a certain number of evolutionary generations. The convergence speed depends on the crossover, mutation and selection functions in EA's.

Further research can focus on a more general representation of kernel, a better design of crossover, mutation function in EA based on the characteristics of

SVM parameters to increase the convergence speeds.

References

[1] Vapnik V. N. An Overview of Statistical Learning Theory. IEEE Transactions on Neural Networks, Vol 10, Issue 5, Sep 1999,pp 988 -999.

[2] Smits, G.F.; Jordaan, E.M. Improved svm regression using mixtures of kernels. Proceedings of the 2002 International Joint Conference on Neural Networks, Vol. 3, 2002, Pages: 2785 -2790.

[3] Drucker H., Wu D., Vapnik V. N. Support Vector Machines for Spam Categorization. IEEE Transactions on Neural Networks, Vol. 10, Issue 5, Sep 1999.

[4] Xin Yao. Evolving artificial neural networks. Proceedings of the IEEE, Vol. 87 Issue: 9, Sept. 1999, Pages: 1423 -1447.

[5] Baoding Liu. Fuzzy random chance-constrained programming. Fuzzy Systems, IEEE Transactions on Fuzzy Systems, Vol. 9 Issue: 5, Oct. 2001, pp 713 - 720.

[6] Osuna E., Freund R., Girosi F. Support vector machines: Training and applications. Massachusetts Institute of Technology, AI Memo No. 1602. 1997.

[7] Lunts A., Brailovskiy V. Evaluation of attributes obtained in statistical decision rules. Engineering Cybernetics, 1967 (3): pp 98-109.

[8] Joachims T. Estimating the generalization performance of a SVM efficiently. Proceedings of the Seventeenth International Conference on Machine Learning, San Francisco, 2000.

[9] Denning E.D. "An Intrusion Detection Model". IEEE Symposium on Security and Privacy, 1986: 118-133.

[10] DARPA. Intrusion Detection Evaluation. URL: http://www.ll.mit.edu/IST/ideval/index.html.

[11] Frank J. "Artificial Intelligence and Intrusion Detection: Current and future Directions." Proceedings of the 17th National Computer Security Conference. Oct. 1994.

[12] Chih-Chung Chang, Chih-Jen Lin. "LIBSVM: a Library for Support Vector Machines". URL: http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf.