Support Vector Machine with Customized Kernel

Guangyi Chen[1], Tien Dai Bui[1], Adam Krzyzak[1], and Weihua Liu[2]

[1] Department of Computer Science and Software Engineering, Concordia University, 1455 de Maisonneuve West, Montreal, Quebec, Canada H3G 1M8
{guang_c,bui,krzyzak}@cse.concordia.ca
[2] State Key Lab. of Virtual Reality Technology and Systems, Beihang University, ZipCode 100191, No 37, Xueyuan Rd., Haidian District, Beijing, P.R. China
liuwh_99@hotmail.com

Abstract. In the past two decades, Support Vector Machine (SVM) has become one of the most famous classification techniques. The optimal parameters in an SVM kernel are normally obtained by cross validation, which is a time-consuming process. In this paper, we propose to learn the parameters in an SVM kernel while solving the dual optimization problem. The new optimization problem can be solved iteratively as follows:

(a) Fix the parameters in an SVM kernel; solve the variables α_i in the dual optimization problem.
(b) Fix the variables α_i, solve the parameters in an SVM kernel by using the Newton–Raphson method.

It can be shown that (a) can be optimized by using standard methods in training the SVM, while (b) can be solved iteratively by using the Newton-Raphson method. Experimental results conducted in this paper show that our proposed technique is feasible in practical pattern recognition applications.

Keywords: Support vector machine (SVM), feature extraction, SVM kernels, pattern recognition, pattern classification.

1 Introduction

Support vector machine (SVM) was developed by Vapnik et al. ([1], [2], [3]) for pattern recognition and function regression. The SVM assumes that all samples in the training set are independent and identically distributed. It uses an approximate implementation to the structure risk minimization principal in statistical learning theory, rather than the empirical risk minimization method. A kernel is utilized to map the input data to a higher dimensional feature space so that the problem becomes linearly separable. An SVM kernel plays a very important role in the performance of the SVM applications.

We briefly review recent advances in SVM applications. Chen and Dudek [4] developed the auto-correlation wavelet kernel for pattern recognition. It was shown that this kernel is better than the wavelet kernel [5] because the auto-correlation

C. Guo, Z.-G. Hou, and Z. Zeng (Eds.): ISNN 2013, Part I, LNCS 7951, pp. 258–264, 2013.

wavelet is shift-invariant whereas the wavelet is not. This shift-invariant property is very important in pattern recognition. Chen [6] also proposed the dual-tree complex wavelet (DTCWT) kernel for SVM classification. The DTCWT developed by Kingsbury [7] has the approximate shift invariant property and better orientation selectivity. These good properties have made the DTCWT a better candidate for pattern recognition.

In this paper, we propose to learn the parameters in an SVM kernel while solving the SVM optimization problem. We break the optimization problem into two smaller optimization problems: (a) Fix the parameters in an SVM kernel, and then solve the variables α_i in the dual optimization problem. (b) Fix the variables α_i, and solve the parameters in an SVM kernel by using the *Newton–Raphson* method. We solve (a) and (b) iteratively for at most τ iterations in each round of optimization, respectively. We repeat the optimization of (a) and (b) in a loop manner until they converge or the maximum number of iterations is reached. Our simulation results show that our proposed method achieves higher classification rates than the standard SVM for recognizing traffic light signals and the vowel dataset (τ=100).

The organization of this paper is as follows. Section 2 proposes to learn the parameters in an SVM kernel while training the SVM for pattern recognition. Section 3 conducts some experiments in order to show that by optimizing the parameters in an SVM kernel we can achieve higher classification rates. Finally, Section 4 draws the conclusions of this paper, and gives future research direction.

2 Proposed Method

An SVM can be used as a classifier for a pattern recognition problem with $n>2$ classes, which can be resolved by solving $n\times(n-1)/2$ two-class SVM problems. A two-class SVM problem can be summarized as follows. Let (x_i,y_i) be a set of training samples, where x_i is the feature vector and $y_i=+1$ or -1.

The primal form of an SVM problem is formulated as:

$$\text{Min: } \tfrac{1}{2}\|w\|^2$$
$$\text{Subject to: } y_i(w^T x_i - b) \geq 1 \text{ for all } i=1, 2, \ldots, n.$$

This is an optimization problem that can be solved by introducing a set of Lagrange multiplier $\alpha_i \geq 0$. We have to solve the following quadratic dual optimization problem:

$$\text{Max: } \sum_{i=1}^{n} \alpha_i - \tfrac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$
$$\text{Subject to: } 0 \leq \alpha_i \leq C \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0.$$

In this paper, we will restrict the kernel to be the radial basis function (RBF) kernel:

$$k(x_i, x_j)=\exp(-\gamma \| x_i - x_j \|^2)$$

or the exponential radial basis function (ERBF) kernel:

$$k(x_i, x_j)=\exp(-\gamma \| x_i - x_j \|),$$

where the parameter $\gamma \geq 0$. In the above dual optimization problem, the parameter γ can be chosen by the user or it can be learned by cross-validation, which is a time-consuming process. We have decided to fix the parameter *C* in this paper.

We propose to solve the above dual optimization problem in two steps:

(a) Fix the parameter γ in an SVM kernel; solve the variables α_i in the dual optimization problem.
(b) Fix the variables α_i; solve the parameters in an SVM kernel by using the *Newton–Raphson* method iteratively.

The first optimization problem (a) can be solved by the standard optimization method in training an SVM. We restrict the number of iteration in solving this optimization problem to be at most τ iterations, instead of looping for many iterations. We modify the C++ code of LIBSVM [8] to solve this optimization problem. After obtaining the approximate parameters α_i, we will solve the second optimization problem (b) iteratively by using the *Newton–Raphson* method.

Let us derive the formula to solve the second optimization problem (b). Since

$$\sum_{i=1}^{n} \alpha_i y_i = 0,$$

We have

$$\sum_{i=1}^{n-1} \alpha_i y_i y_n = - \alpha_n.$$

By plugging α_n into the dual optimization problem, we obtain the following optimization problem without any constrains:

$$\max_{\gamma} W(\gamma) = \sum_{i=1}^{n-1} \alpha_i (1-y_n y_i) - \tfrac{1}{2} \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} \alpha_i \alpha_j y_i y_j$$

$$k(x_i,x_j)+ (\sum_{i=1}^{n-1} \alpha_i y_i)(\sum_{j=1}^{n-1} \alpha_j y_j k(x_n,x_j))- ½ (\sum_{j=1}^{n-1} \alpha_i y_i)^2$$

In order to obtain the maximization, we need to set the first derivative $W'(\gamma)=0$. From the above equation, we can derive:

$$W'(\gamma)= ½ \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} \alpha_i \alpha_j y_i y_j k(x_i,x_j)(-||x_i-x_j||^2) - \alpha_n y_n (\sum_{j=1}^{n-1} \alpha_j y_j k(x_n,x_j))(-||x_n-x_j||^2) \quad \textit{for RBF.}$$

$$W'(\gamma)= ½ \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} \alpha_i \alpha_j y_i y_j k(x_i,x_j)(-||x_i-x_j||) - \alpha_n y_n (\sum_{j=1}^{n-1} \alpha_j y_j k(x_n,x_j))(-||x_n-x_j||) \quad \textit{for ERBF.}$$

and

$$W''(\gamma)= ½ \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} \alpha_i \alpha_j y_i y_j k(x_i,x_j)(||x_i-x_j||^4) - \alpha_n y_n (\sum_{j=1}^{n-1} \alpha_j y_j k(x_n,x_j))(||x_n-x_j||^4) \quad \textit{for RBF.}$$

$$W''(\gamma)= ½ \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} \alpha_i \alpha_j y_i y_j k(x_i,x_j)(||x_i-x_j||^2) - \alpha_n y_n (\sum_{j=1}^{n-1} \alpha_j y_j k(x_n,x_j))(||x_n-x_j||^2) \quad \textit{for ERBF.}$$

From the *Newton-Raphson* method, we obtain the following formula for the second optimization problem (b):

$$\gamma_{k+1} = \gamma_k - W'(\gamma_k) / W''(\gamma_k).$$

We would like to restrict the number of iterations for the second optimization problem (b) to be at most $\tau=100$ iterations, and then switch to the first optimization problem (a). We repeat to solve the two optimization problems (a) and (b) interchangeably until convergence or the maximum number of iterations is reached. The above solutions are for a two-class classification problem. Let $\Delta\gamma_k = \gamma_{k+1} - \gamma_k$ be for a two-class classification problem. Since we have to solve $n\times(n-1)/2$ two-class SVM

problems, we can take the mean of $\Delta\gamma_{k,}$ over all these $n\times(n-1)/2$ two-class SVM problems. Therefore, the iterative formula for solving the parameter γ_k can be given as

$$\gamma_{k+1} = \gamma_k + \varepsilon \times mean(\Delta\gamma_k).$$

It is expected that by solving the two optimization problems interchangeably, we can obtain better solutions for pattern recognition. The decision function of a two-class SVM problem is

$$f(x) = sign(\sum_{i=1}^{n} \alpha_i y_i k(x_i,x)+b)$$

where

$$b = y_r - \sum_{i=1}^{n} \alpha_i y_i k(x_i,x_r)$$

and (x_r,y_r) is a training sample. For the one-versus-one SVM problem, classification is performed by a max-wins voting approach, in which every classifier assigns the instance to one of the two classes. The vote for the assigned class is increased by one, and the class with most votes determines the instance classification.

3 Experimental Results

We conducted some experiments by using the data sets *svmguide4* and *vowel* provided in [8]. The *svmguide4* data set is for traffic light signals, which has 6 classes with 300 training samples and 312 testing samples. The number of features was chosen to 10. The *vowel* data set has 11 classes with 528 training samples and 462 testing samples. The number of features was also chosen to 10. We used the following Code One and Two to train and test the standard LIBSVM and our proposed SVM, where the parameters *C* and *g* can be changed as desired. In our experiments, we choose τ=100 and ε=0.01 for the traffic light dataset and for the vowel dataset.

----------Code One----------
svm-scale *-l 0 -s range1 svmguide4 > svmguide4.scale*
svm-scale *-r range1 svmguide4.t > svmguide4.t.scale*
svm-train *-c 100 -g 0.2 svmguide4.scale*
svm-predict *svmguide4.t.scale svmguide4.scale.model svmguide4.t.predict*

----------Code Two----------
svm-scale *-l -1 -u 1 -s range3 vowel.scale > vowel.scale.scale*
svm-scale *-r range3 vowel.scale.t > vowel.scale.t.scale*
svm-train *-c 100 -g 0.2 -t 2 vowel.scale.scale*
svm-predict *vowel.scale.t.scale vowel.scale.scale.model vowel.scale.t.predict*

Tables 1-2 tabulate the parameters *C* and *g*, and the recognition rates for both the standard LIBSVM and our proposed SVM for the *traffic light* dataset and *vowel* dataset, respectively. From the two tables, it can be seen that our proposed SVM obtains higher classification rates than the standard LIBSVM due to the learning strategy introduced in our proposed SVM. Note that we only used the RBF in our experiments. We leave ERBF to our future research.

Table 1. A comparison between the standard LIBSVM and our proposed SVM for the traffic light dataset

Parameter *C*	Parameter *g*	Classification rate (LIBSVM)	Classification rate (Proposed SVM)
100	0.2	78.53%	**81.73%**
10	0.2	54.17%	**66.67%**
1	0.2	29.17%	**46.79%**

Table 2. A comparison between the standard LIBSVM and our proposed SVM for the vowel dataset

Parameter *C*	Parameter *g*	Classification rate (LIBSVM)	Classification rate (Proposed SVM)
100	0.2	55.19%	**64.94%**
10	0.2	53.03%	**63.20%**
1	0.2	59.74%	**61.26%**

4 Conclusions and Future Work

We have proposed a solution for solving an n-class classification problem by using SVM. We resolve the n-class SVM classification problem by solving $n \times (n-1)/2$ two-class SVM problems. Each two-class SVM classification problem can be resolved by (a) fixing the parameter γ in an SVM kernel and then solve the variables α_i in the dual optimization problem, and by (b) fixing the variables α_i and solve the parameter γ in an SVM kernel by using the *Newton–Raphson* method iteratively. We solve for (a) and (b) interchangeably until they converge or the maximum number of iterations τ=100 is reached. Experimental results show that the proposed method in this paper is feasible in pattern recognition.

Further research needs to be done by learning the upper bound *C* as well while solving the optimization problems. It is believed that, by optimizing both *C* and γ, we can obtain higher classification rates for *n*-class pattern recognition problems. We may also apply our proposed SVM to the recognition of handwritten digits and handwritten characters. We are very interested in extracting the dual-tree complex wavelet features, the ridgelet features, the contourlet features, the curvelet features, etc. ([7], [9], [10]).

Acknowledgments. This research was supported by the research grant from the Natural Science and Engineering Research Council of Canada (NSERC) and Beijing Municipal Science and Technology Plan: Z111100074811001.

References

1. Vapnik, V.N.: The nature of statistical learning. Springer, New York (1995)
2. Vapnik, V.N.: Statistical learning theory. Wiley, New York (1998)
3. Cortes, C., Vapnik, V.N.: Support vector networks. Machine Learning 20, 273–297 (1995)
4. Chen, G.Y., Dudek, G.: Auto-correlation wavelet support vector machine. Image and Vision Computing 27(8), 1040–1046 (2009)
5. Zhang, L., Zhou, W., Jiao, L.: Wavelet support vector machine. IEEE Transactions on Systems, Man, and Cybernetics - Part B 34(1), 34–39 (2004)
6. Chen, G.Y.: Dual-tree Complex Wavelet Support Vector Machine. International Journal of Tomography & Statistics 15(F10), 1–8 (2010)
7. Kingsbury, N.G.: Complex wavelets for shift invariant analysis and filtering of signals. Journal of Applied and Computational Harmonic Analysis 10(3), 234–253 (2001)
8. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2(3) (2011), Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
9. Candes, E.J.: Ridgelets and the representation of mutilated Sobolev functions. SIAM J. Math. Anal. 33(2), 2495–2509 (1999)
10. Do, M.N., Vetterli, M.: The contourlet transform: an efficient directional multiresolution image representation. IEEE Transactions on Image Processing 14(12), 2091–2106 (2005)