NESTING SUPPORT VECTOR MACHINTE FOR MUTI-CLASSIFICATION

BO LIU[1], ZHI-FENG HAO[2], XIAO-WEI YANG[2]

[1] College of Computer Science and Engineering, South China University of Technology, Guangzhou 510640, P. R. China
[2] Department of Applied Mathematics, South China University of Technology, Guangzhou510640, P. R. China
E-MAIL: Csbliu@yahoo.com.cn

Abstract:

Support vector machines (SVMS) were originally designed for binary classifications. As for multi-classifications, they are usually converted into binary ones, where unclassifiable regions usually exist. To overcome this drawback, a novel method called Nesting Support Vector Machine (NSVMS) for multi-classification is presented in this paper. Our ideas are as follows: Firstly, construct the optimal hyperplanes based on One-against-One algorithm. Secondly, if there exist data points in the middle unclassifiable region, select them to construct optimal hyperplanes with the same hyperparameters. Thirdly, repeat the second step until there are no data points in the unclassifiable regions or the regions disappear. In order to examine the training accuracy and the generalization performance of the proposed algorithm, One-against-One algorithm, Fuzzy Least Square Support Vector Machine (FLS-SVM) and the proposed algorithm are applied to two UCI datasets. The results show that the training accuracy of the proposed algorithm is higher than the others, and its generalization performance is also comparable with them.

Keywords:

Support vector machines; Least Squares Support Vector Machine; One-against-One algorithm; FLS-SVM; Nesting Support Vector Machine; Multi-classification

1. Introduction

The standard support vector machines (SVM) [1] were originally designed for binary classifications. Unfortunately, many practical applications consist of multi-classification problems, which are usually converted into binary ones. Up to now, several methods have been proposed to decompose and reconstruct multi-classification problems, which one can refer to a good view of these methods written by Rifkin [2].

In the previous work, Vapnik proposed One-against-All algorithm [3], in which a $C-$ class problem is transformed into C two-class problems and for the i th two-class problem, class i is separated from the remaining classes. However, there exist some unclassifiable regions and the accuracy of this algorithm is very low. To overcome this problem, Vapnik [1] introduced the continuous decision functions instead of the discrete decision functions. KreBel [4] converted the $C-$ class problem into $\frac{C(C-1)}{2}$ two-class problems, which is called pairwise classification (One-against-One). In this case, the middle unclassifiable region still remains. Platt et al. [5] proposed decision-tree-based pairwise classification, which was called Decision Directed Acyclic Graph (DDAG). Inoue and Abe [6] presented fuzzy support vector machines for One-against-All classification, in which fuzzy membership functions are defined instead of the continuous decision functions. Recently, they put forward to a fuzzy support vector machine based on pairwise classification [7]. Tsujinishi and Abe [8] introduced a new method depending on pairwise classification, in which the average membership function is defined instead of the minimum membership function. In order to improve the accuracy of the multi-classification, Hao et al [9] proposed TMSVM algorithm.

In this paper, a novel method called the Nesting Support Vector Machines is proposed to deal with the unclassifiable regions in the One-against-One algorithm.

The rest of this paper is organized as follows: In Section 2, we review the standard SVM, LS-SVM and Kernel Function. One-against-One algorithm and Fuzzy Least Squares Support Vector Machine are briefly described in Section 3. In Section 4, the proposed algorithm is given. Experimental results are shown in Section 5 and some conclusions will be given in Section 6.

2. Least Squares Support Vector Machine

In this section, we will first shortly review the standard support vector machine, least squares support vector machine (LS-SVM) and kernel function. As for the details one can refer to [3] [10].

2.1. Standard Support Vector Machine

Let $S=\{(x_1,y_1),(x_2,y_2),\ldots,(x_l,y_l)\}$ be a training set with input data $x_i \in R^n$ and corresponding binary class labels $y_i \in \{-1,+1\}$. According to the Vapnik's original formulation, the SVM classifier is:

$$D(x)=w^T\psi(x)+b \tag{1}$$

where $\psi(x)$ is a nonlinear function, which maps the input space x into a feature space (and possibly infinite), w^T is an m-dimensional vector, and b is a scalar.

To separate the data linearly in the feature space, the decision function satisfies the following conditions:

$$y_i(w^T\psi(x_i)+b)\geq 1 \qquad for \quad i=1,\cdots,l \tag{2}$$

In order to determine the optimal separating hyperplane, which has the maximum margin between two classes, we can formulate the following optimization problem:

$$\min_{w,b} J(w,b)=\frac{1}{2}w^t w \tag{3}$$

s. t.

$$y_i(w^T\psi(x_i)+b)\geq 1 \qquad for \quad i=1,\cdots,l \tag{4}$$

When the training data are linearly nonseparable, we need introduce slack variables ξ_i into Eq. (2) as follows:

$$y_i(w^T\psi(x_i)+b)\geq 1-\xi_i \qquad for \quad i=1,\cdots,l \tag{5}$$

$$\xi_i \geq 0 \qquad for \quad i=1,\cdots,l \tag{6}$$

To obtain the optimal separating plane, we need solve the following optimization problem:

$$\min_{w,b,\xi} J(w,b,\xi)=\frac{1}{2}w^T w+\gamma\frac{1}{2}\sum_{k=1}^{l}\xi_i\ , \tag{7}$$

s. t.

$$y_i(w^T\psi(x_i)+b)\geq 1-\xi_i \qquad for \quad i=1,\cdots,l \tag{8}$$

$$\xi_i \geq 0 \qquad for \quad i=1,\cdots,l \tag{9}$$

where γ is a parameter which determines the tradeoff between the maximum margin and the minimum classification error.

The corresponding Lagrangian is:

$$L(w,b,\xi;\alpha,\beta)=J(w,b,\xi)-\sum_{i=1}^{l}\alpha_i\{y_i[w^T\psi(x_i)+b]-1+\xi_i\} - \sum_{i=1}^{l}\beta_i\xi_i \tag{10}$$

(the minimum with respect to w,b,ξ_i and the maximum with respect to nonnegative multipliers α_i,β_i). The parameters that minimize the Larangian must satisfy the following conditions:

$$\frac{\partial L}{\partial w}=0 \rightarrow w=\sum_{i=1}^{l}\alpha_i y_i\psi(x_i). \tag{11}$$

$$\frac{\partial L}{\partial b}=0 \rightarrow \sum_{i=11}^{l}\alpha_i y_i=0. \tag{12}$$

$$\frac{\partial L}{\partial \xi_i}=0 \rightarrow \alpha_i+\beta_i=\gamma, \qquad i=1,\cdots,l. \tag{13}$$

Substituting Eq. (11) into the Lagrangian and taking into account Eqs. (12) and (13), we obtain the following dual problem:

Maximize

$$Q(\alpha)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j\psi(x_i)^T\psi(x_j) \tag{14}$$

s. t

$$\sum_{i=11}^{l}\alpha_i y_i=0, \qquad 0\leq\alpha_i\leq\gamma \tag{15}$$

2.2. Least Squares Support Vector Machine

In contrast to the SVMS, the LS-SVM [10] is trained by minimizing:

$$\min_{w,b,e} J(w,b,e)=\frac{1}{2}w^T w+\gamma\frac{1}{2}\sum_{i=1}^{l}e_i^2 \tag{16}$$

s. t.

$$y_i(w^T\psi(x_i)+b)=1-e_i \qquad for \quad i=1,\cdots,l \tag{17}$$

The corresponding Lagrangian is:

$$L(w,b,e;\alpha)=J(w,b,e)-\sum_{i=1}^{l}\alpha_i\{y_i[w^T\psi(x_i)+b]-1+e_i\} \tag{18}$$

where α_i are Lagrange multipliers, in which $\alpha_i \neq 0$ are called support values.

The conditions for optimality are as follows:

$$\frac{\partial L}{\partial w}=0 \rightarrow w=\sum_{i=1}^{l}\alpha_i y_i\psi(x_i). \tag{19}$$

$$\frac{\partial L}{\partial b}=0 \rightarrow \sum_{i=1}^{l}\alpha_i y_i=0. \tag{20}$$

$$\frac{\partial L}{\partial e_i}=0 \rightarrow \alpha_i=\gamma e_i, \qquad i=1,\cdots,l. \tag{21}$$

In a matrix form, Eqs. (19)-(21) are expressed by

$$\begin{bmatrix} \Omega & Y \\ Y^t & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ b \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{22}$$

where Ω, Y and 1 are, respectively,

$$\Omega_{ij} = y_i y_j \psi(x_i)^T \psi(x_j) + \frac{\delta_{ij}}{\gamma} \tag{23}$$

$$Y = [y_1, \cdots, y_l]^T \tag{24}$$

$$1 = [1, \cdots, 1]^T \tag{25}$$

here,

$$\delta_{ij} = \begin{cases} 1 & i = j. \\ 0 & i \neq j. \end{cases} \tag{26}$$

From Eq. (21), one can see that the support values α_k are proportional to the errors in LS-SVM. In general, most of the errors are not equal to zero. So, the sparseness of the SVM will be lost, which is the main drawback of LS-SVM.

2.3. Kernel Function

One of the characteristics of the SVM is that it uses the kernel trick. In Eqs. (14) and (23), defining

$$K(x, x_i) = \psi(x)^T \psi(x_i) \tag{27}$$

we can avoid treating variables in the feature space. For the kernel function $K(.,.)$ one typically has the following choices:

$K(x, x_i) = x^t x$ (Linear kernel)

$K(x, x_i) = (1 + x^t x / c)^d$ (Polynomial kernel)

$K(x, x_i) = \exp\{-\|x - x_i\|_2^2 / \sigma^2\}$ (RBF kernel)

$K(x, x_i) = \tanh(k x_i^t x + \theta)$ (MLP kernel)

3. Multi-Classification Algorithms

As for multi-classification, they are usually converted into binary ones. Up to now, several methods have been proposed to decompose and reconstruct multi-classification problems. In present section, let us introduce the conventional multi-classification algorithms.

3.1. One-against-One Algorithm

According to the conventional pairwise classification [4], one needs to determine $\frac{C(C-1)}{2}$ decision functions for the C-classes. The optimal hyperplane for class i against j, which has the maximum margin between them, is

$$D_{ij}(x) = w_{ij}^T x + b_{ij} = 0 \tag{28}$$

where w_{ij}^T is an m-dimensional vector, and b_{ij} is a scalar.

Here define the orientation of the optimal hyperplane via the following equation.

$$D_{ij}(x) = -D_{ji}(x) \tag{29}$$

For the input vector, one computes

$$D_i(x) = \sum_{j \neq i, j=1}^{c} sign(D_{ij}(x)) \tag{30}$$

where

$$sign(x) = \begin{cases} 1 & x > 1 \\ 0 & x \leq 1 \end{cases} \tag{31}$$

and classifies x into the class

$$\arg \max_{i=1,\ldots,c} (D_i(x)) \tag{32}$$

If i is not unique in Eq. (32), then x is unclassifiable. For example, $D_i(x) = 1$ for i=1, 2, 3 in the shaded region of Fig. 1. Therefore, the shaded region is unclassifiable [7].

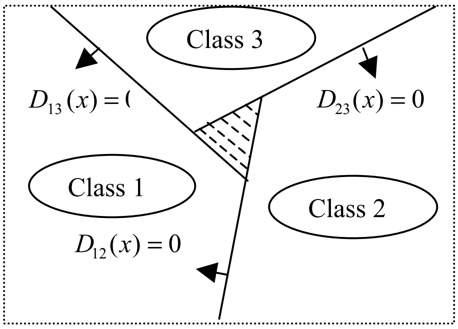

Figure 1 One-against-One

3.2. Fuzzy Least Squares Support Vector Machines Algorithm

In order to overcome the drawback of the One-against-One, Daisuke, Tsujinishi and Abe [8] introduced the fuzzy membership functions based on the One-against-One classification. Constructing the hyperplanes in the feature space, they defined the one-dimensional membership functions $m_{ij}(x)$ as follows:

$$m_{ij} = \begin{cases} 1 & D_{ij}(x) \geq 1 \\ D_{ij}(x) & otherwise \end{cases} \tag{33}$$

In their work, using the minimum operator or the average operator, the membership functions are given by

$$m_i(x) = \min_{j=1,\ldots,c}(m_{ij}(x)) \tag{34}$$

or

$$m_i(x) = \frac{1}{k-1}\sum_{j\neq i, j=1}^{c} m_{ij}(x) \tag{35}$$

where C is the total number of classes. Using either Eq. (34) or Eq. (35), the input data is classified into the class

$$\arg\max_{i=1,\cdots,c} m_i(x) \tag{36}$$

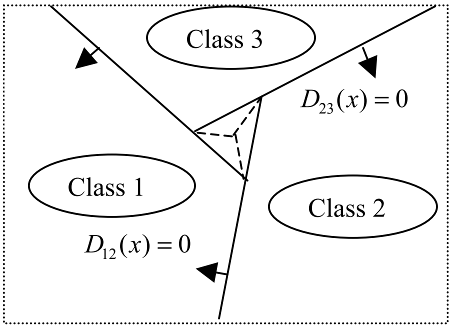

Figure 2 FLS-SVMS

Using the minimum operator, the unclassifiable region shown in Fig. 1 is resolved as shown in Fig.2.

4. Proposed Algorithm

Let $S=\{(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),\ldots,(\mathbf{x}_l,y_l)\}$ be a training set from an unknown distribution, where $\mathbf{x}_i \in R^n$ and $y_i \in \{1,2,\ldots,C\}$. In order to solve the unclassifiable regions in the One-against-One algorithm, we propose a novel algorithm in this section, which is called Nesting Support Vector Machine (NSVMS).

One can train the SVMS by the following steps which can be shown in Fig. 3:

Firstly, construct $\frac{C(C-1)}{2}$ hyperplanes in the feature space based on One-against-One approach [4].

Secondly, select the data points in the middle unclassifiable region according to Eq. (32).

Thirdly, use the data points in the unclassifiable region alone to construct hyperplanes with the same hyperparameters.

Finally, repeat the second and third steps until there are no data points in the middle region or the region disappear.

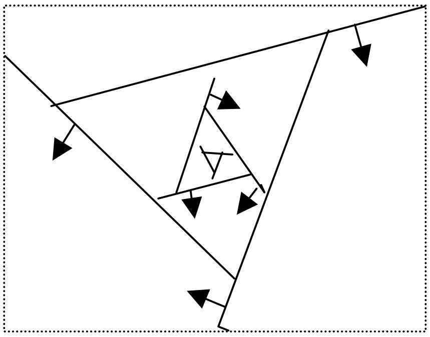

Figure 3 LS-NSVMS

By the proposed approach, the data points can be separated and the computational complexity is little higher than One-against-One and FLS-SVM, which can be proved as follows:

- One constructs the SVMs according to the proposed method. The data points in the unclassifiable region are one of the following cases within the finite steps.
  1. There is one class or no data points in the middle unclassifiable region.
  2. There are two class data points in the middle unclassifiable region.

  If the first case appears, the data points in the unclassifiable region belong to one class. If the second one happens, construct the SVMs to separate the two classes and then the unclassifiable region disappears.
- The computational complexity of training binary LS-SVM is in proportion to $O(l^3)$, where l is the number of training data points. Assume that the data points in the space is well-proportioned and l is the total number of the training data samples, then

$$T_{One-Against-One} = T_{FLS-SVM} \sim C_C^2 O(\frac{2l}{C})^3 \sim \frac{C(C-1)}{2} O(\frac{2l}{C})^3 \tag{37}$$

$$T_{NSVM} \sim C_{C_1}^2 O(\frac{2l_1}{C_1})^3 + C_{C_2}^2 O(\frac{2l_2}{C_2})^3 + C_{C_3}^2 O(\frac{2l_3}{C_3})^3 + \cdots + C_{C_m}^2 O(\frac{2l_m}{C_m})^3$$

$$a = \max(\frac{l_2}{l_1}, \frac{l_3}{l_2}, \frac{l_4}{l_3} \cdots, \frac{l_m}{l_{m-1}}) \tag{38}$$

$$\beta = mim(\frac{C_2}{C_1}, \frac{C_3}{C_2}, \frac{C_4}{C_3} \cdots, \frac{C_m}{C_{m-1}}) \tag{39}$$

$$T_{NSVM} \le C_{C_1}^2 O(\frac{2l_1}{C_1})^3 + C_{C_2}^2 O(\frac{2\alpha l_1}{C_2})^3 + C_{C_3}^2 O(\frac{2\alpha^2 l_1}{C_3})^3$$

$$+ \cdots + C_{C_m}^2 O(\frac{2\alpha^{m-1} l_1}{C_m})^3$$

$$= \alpha^{3*0} C_{C_1}^2 O(\frac{2l_1}{C_1})^3 + \alpha^{3*1} C_{C_2}^2 O(\frac{2l_1}{C_2})^3 + \alpha^{3*2} C_{C_3}^2 O(\frac{2l_1}{C_3})^3$$

$$+ \cdots + \alpha^{3*(m-1)} C_{C_m}^2 O(\frac{2l_m}{C_m})^3$$

$$\le \alpha^{3*0} C_{C_1}^2 O(\frac{2l_1}{C_1})^3 + \alpha^{3*1} C_{C_1}^2 O(\frac{2l_1}{C_2})^3 + \alpha^{3*2} C_{C_1}^2$$

$$O(\frac{2l_1}{C_3})^3 + \cdots + \alpha^{3*(m-1)} C_{C_1}^2 O(\frac{2l_1}{C_m})^3$$

$$\le \alpha^{3*0} C_{C_1}^2 O(\frac{2l_1}{C_1})^3 + \alpha^{3*1} C_{C1}^2 O(\frac{2l_1}{\beta C_1})^3 + \alpha^{3*2} C_{C_1}^2$$

$$O(\frac{2l_1}{\beta^2 C_1})^3 + \cdots + \alpha^{3*(m-1)} C_{C_1}^2 O(\frac{2l_1}{\beta^{m-1} C_1})^3$$

$$= C_{C_1}^2 O(\frac{2l_1}{C_1})^3 (\frac{\alpha^{3*0}}{\beta^{3*0}} + \frac{\alpha^{3*1}}{\beta^{3*1}} + \frac{\alpha^{3*2}}{\beta^{3*2}} + \cdots + \frac{\alpha^{3*(m-1)}}{\beta^{3*(m-1)}})$$

According to

$$C_1 = C, \qquad \beta \ge \frac{1}{C}, \qquad l = l_1,$$

one has

$$T_{NSVM} \le C_C^2 O(\frac{2l}{C})^3 ((C\alpha)^{3*0} + (C\alpha)^{3*1} + (C\alpha)^{3*2} +$$

$$\cdots + (C\alpha)^{3*(m-1)})$$

$$= (\frac{1-(C\alpha)^{3m}}{1-(C\alpha)^3}) C_C^2 O(\frac{2l}{C})^3 \tag{40}$$

where l_i denote the number of the data points which construct svms, m is the number of layers of svms, C_i mean the number of class, and T is the computational complexity. According to the experience of experiment, a is near to zero, $C\alpha$ is also close to zero. From Eq. (37) and Eq. (40), we can easily conclude that the computational complexity of NSVM is little higher than those of One-against-One and FL-SVM.

5. Performance Evaluation

The experiments are run on a PC with a 2.8GHz Pentium IV processor and a maximum of 512MB memory. All the programs are written in C++, using Microsoft's Visual C++ 6.0 compiler. In order to evaluate the performance of the proposed algorithm, One-Against-One, FLS-SVM and NSVM are applied to three UCI data sets available from the UCI Machine Learning Repository [11]. Data preprocessing is in the following:

- Abalone dataset: This dataset consists of 4177 data with 6 features and 29 classes and hasn't class 2. We reconstructed the data by combining classes 1, 3, 4 into one class, class 5 into one class, class 6 into one class, class 7 into one class, classes 8, 9 into one class, classes 10, 11 into one class, classes 12, 13 into one class, the rest classes into one class. We select 2177 samples randomly for training, and the rest for testing.
- Thyroid dataset: This dataset involves 3772 data points for training and 3428 data points for testing with 21 features and 3 classes.

An RBF kernel function is employed, the parameter values and the results are shown in Tables 1and 2. Although the parameters may be not the optimal hyperparameters, we consider them reasonable.

Table1 The results of total data points

Dataset	Parameter (γ, σ)	One-Against-One (%)		LS-NSVMS (%)	
		Testing	Training	Testing	Training
Abalone	(3,25)	45.35	48.002	45.65	48.32
Thyroid	(1,1)	94.136	94.936	94.428	95.387

Table2 The results of total data points

Dataset	Parameter (γ, σ)	FLS-SVM (%)		LS-NSVMS (%)	
		Testing	Training	Testing	Training
Abalone	(3,25)	45.6	48.277	45.65	48.32
Thyroid	(1,1)	94.340	95.201	94.428	95.387

Table3 The results of data in the unclassifiable region

Dataset	Parameter (γ, σ)	FLS-SVM (%)		LS-NSVMS (%)	
		Testing	Training	Testing	Training
Abalone	(3,25)	71.43	85.71	85.714	100
Thyroid	(1,1)	50.0	58.823	71.429	100

Tables 1 and 2 show the accuracy of total data points, the accuracy of data points in the unclassifiable region is listed in the Table 3.

According to Tables 1, 2 and 3, we can easily conclude that the training and test accuracy in the unclassifiable region is much higher than those of FLS-SVM, but the accuracy of total data points is enhanced little. Because compared with all of the data points, the data points in the middle unclassifiable region are much fewer. In this way, we can well solve the unclassifiable region in the one-against-one algorithm.

6. Conclusions

In order to solve unclassifiable region, a novel method is presented in this paper. One-Against-One, FLS-SVM and NSVMS are applied to two UCI data sets to examine the training accuracy and the generalization performance of NSVM. The results show that the training accuracy of NSVM is higher than those of One-Against-One and FLS-SVM, and its generalization performance is also comparable with them.

Acknowledgements

This work has been supported by the National Natural Science Foundation of China (10471045, 10471045), Guangdong Key Laboratory of Computer Network (No.CN200403), Natural Science Foundation of Guangdong Province (04020079, 970472, 000463, 04020079), Fok Ying Tong Education Foundation (91005), Guangdong Key Laboratory of Computer Network Foundation (No.CN200403) and Natural Science Foundation of South China University of Technology (D76010).

References

[1] V. N. Vapnik. Statistical Learning Theory. John Wiley & Sons, 1998.

[2] R. Rifkin, A. Klautau, ”In Defense of One-Vs-All Classification”, Journal of Machine Learning Research, Vol, 5, pp. 101-141, 2004.

[3] V. N. Vapnik. The Nature of Statistical Learning Theory. Springer-Verlag, London, UK, 1995.

[4] U. H. G. KreBel, “Pairwise classification and support vector machines”, In B. Schölkopf, C. J. Burges, and A. J. Smola, editors, Advances in Kernel Methods: Support Vector Learning, pp 255-268. The MIT Press, Cambridge, MA, 1999.

[5] J. C. Platt, N. Cristianini, and J. Shawe.-Taylor, “Large Margin DAGs for multiclass classification”, In S. A. Solla, T. K. Leen, and K. R. Müller, editors, Advances in Neural Information Processing Systems 12, pp 547-553. The MIT Press 2000.

[6] T. Inoue and S. Abe, “Fuzzy support vector machines for pattern classification”, In Proceedings of International Joint Conference on Neural Networks (IJCNN’01), Vol. 2, pp 1449-1454, July 2001.

[7] S. Abe and T. Inoue, “Fuzzy support vector machines for multiclass problem”, In Proceedings of 10th European Symposium on Artificial Neural Networks (ESANN”2002), Bruges, Belgium pp. 113-118, April 2002.

[8] D. Tsujinishi and S. Abe, ”Fuzzy least squares support vector machines for multiclass problems”, Neural Network 16(2003) 785-792..

[9] Z. F. Hao el al, “Twi-map Support Vector Machine for Multi-classification Problems”, Proceeding of ISNN 2005 Conference. Lecture Notes in Computer Science 3496. Chongqing, PT1, pp 869-874. May 2005.

[10] J. A. K. Suykens, “Least Squares Support Vector Machine for Classification and Nonlinear Modelling,” Neural Network World, Vol. 10, No. 1-2, pp. 29-47, 2000.

[11] P. M. Murphy, D. W. Aha. UCI repository of machine learning database, 1992.