The Fifteenth International Symposium on Artificial Life and Robotics 2010 (AROB 15th '10),
B-Con Plaza, Beppu,Oita, Japan, February 4-6, 2010

Midpoint-Validation Algorithm for Support Vector Machine Classification

Hiroki Tamura, Shingo Yamashita and Koichi Tanno

Faculty of Engineering, University of Miyazaki
1-1 Gakuken kibanadai-nishi, Miyazaki-shi, 889-2192 Japan
(htamura @ cc.miyazaki-u.ac.jp)

Abstract: In this paper, we propose midpoint-validation algorithm for Support Vector Machine, which improves the generalization of support vector machine so that midpoint-validation error is minimized. We compare its performance with the other paper techniques of support vector machine and also tested our proposed method on fifth benchmark problems. The results obtained from the simulation shows the effectiveness of the proposed method.

Keywords: Support Vector Machine; Midpoint-Validation; Pattern Classification Problem

1. Introduction

Support vector machine (abbr. SVM) proposed by Vapnik [1] is one of the most influential and powerful tools for solving classification[2-6]. The main concept is based on the formation of a Lagrange multiplier equation combining both objective terms and constraints. The most attractive notions are the idea of the large margin and kernel. It has produced a remarkable performance in a number of difficult learning tasks without requiring prior knowledge and with guarantee on its generalization behavior dues to the method of structural risk minimization.

A number of improved SVM have been proposed to improve the generalization. Weston [8] proposed an algorithm to leverage the Universum by maximizing the number of observed contradictions, and showed experimentally that this approach delivers accuracy improvements over using labeled data alone. Raicharoen [12] proposed the learning algorithms that do not need any kernel functions to map the input vectors into a linearly separable feature space. The separability is based on the critical Support vectors essential to determine the locations of all separating hyperplanes. A separating hyperplane is placed in the middle of the line connecting two support vectors, one from each class, and it is also orthogonal to this line.

We proposed a midpoint-validation method which improves the generalization of neural network (Tamura & Tanno, Oct. 2008[9]). This method creates midpoint data in input space, and calculates criteria using the midpoint data and previous training data. We stop training as soon as the criteria is higher than it was the last time it was checked. Further, we proposed the adjustment method of SVM from the result obtained from the SVM that used the midpoint data (Tamura & Tanno, Jul. 2008[10]). This method is adjusted for threshold value in order for the output of the midpoint data to be set to nearly 0. This technique had better results than SVM, Multilayer Perceptron using midpoint-validation method or cross-validation method.

In this paper, we propose midpoint-validation algorithm, which improves the generalization of SVM so that midpoint-validation error is minimized. This idea applies midpoint-validation method to learning algorithm of SVM and has the new rules of adjustment method. We compare performance of midpoint-validation algorithm with those of the SVM, *soft margin* SVM, SVM using midpoint-validation method and tested our proposed method on fifth benchmark problems. The simulation results carried out shows the effectiveness of the midpoint-validation algorithm.

The rest of this paper is organized into six sections. Section 2 reviews the concept of original SVM algorithm. Section 3 introduces Midpoint-Validation Method for SVM about paper [10]. Section 4 presents our proposed midpoint-validation algorithm for SVM. Section 5 provides the experimental results performed with several benchmark data and compares them with the others'. Section 6 concludes the paper.

2. Review of SVM

The SVM is a mechanical learning system that uses a hypothesis space of linear functions in a high dimensional feature space. Assume the training sample $S=((\mathbf{x}_1,y_1),....(\mathbf{x}_N,y_N))$ consisting of vectors $\mathbf{x}_i \in R$ with i=1,..., N , and each vector $\mathbf{x}_i$ belongs to either of the two classes. Thus it is given a label $y_i \in \{-1,1\}$. The pair of ($\mathbf{w}$,b) defines a separating hyper-plane of equation as follows:

$$(\mathbf{w},\mathbf{x})+b=0 \qquad (1)$$

However, Eq.(1) can possibly separate any part of the feature space, therefore one needs to establish an optimal separating hyper-plane (abbr. OSH) that divides ***S*** leaving all. The points of the same class are accumulated on the same side while maximizing the margin which is the distance of the closest point of ***S***. The closest vector $\mathbf{x}_i$ is called support vector and the OSH($\mathbf{w}'$,b') can be determined by solving an optimization problem. The solution of this optimization problem is given by the saddle point of the Lagrangian.

Maximize margin $\frac{1}{2}(\mathbf{w},\mathbf{w})$

Subject to $y_i((\mathbf{w}\cdot\mathbf{x}_i)+b)\geq 1$ (2)

To solve the case of nonlinear decision surfaces, the OSH is carried out by nonlinearly transforming a set of original feature vectors $\mathbf{x}_i$ into a high-dimensional feature space by mapping $\mathbf{\Phi}$: $\mathbf{x}_i \rightarrow \mathbf{z}_i$ and then performing the linear separation. However, it requires an enormous computation of inner products ($\mathbf{\Phi}(\mathbf{x}) \cdot \mathbf{\Phi}(\mathbf{x}_i)$) in the high-dimensional feature space. Therefore, using a Kernel function which satisfies the Mercer's theorem significantly reduces the calculations to solve the nonlinear problems. In this paper, we used the Gaussian kernel given in Eq.(3) as the kernel function while the SVM decision function $g(\mathbf{x})$ and output of SVM are as given in Eq.(4), (5).

$$K(\mathbf{x},\mathbf{x}_j)=\exp\left(\frac{-\|\mathbf{x}-\mathbf{x}_j\|^2}{2\sigma^2}\right) \quad (3)$$

$$g(\mathbf{x})=\sum_{i=1}^{N} w_i K(\mathbf{x}_i,\mathbf{x})+b \quad (4)$$

$$O=sign(g(\mathbf{x})) \quad (5)$$

3. Midpoint-Validation Method [10]

3.1 Creation of Midpoint Data

Step 1: A training data (x_i) that belongs to group ***D0*** is selected accordingly.
Step 2: A training data (x_j) that has nearest distance in group ***D1*** is selected.
Step 3: A training data (x_i*) that has nearest distance in groups ***D0*** is also selected.
Step 4: Go to Step 5 when training data (x_i*) and training data (x_i) are the same.
Else to Step 2 and (i^*) is substituted for (i).
Step 5: A midpoint of training data (x_i) and (x_j) is decided as the new midpoint data (x_k^{new}).

Fig 1. Flow of Midpoint Data Creation Algorithm

Midpoint data are created from the existing known training data which has different teacher signal. The midpoint data created is midpoint of the known training data and it is expected that by doing so, the generalization would improve. As for the teacher signal, it is assumed to have two classes (-1or 1). Training data groups that belongs to each teacher signal is assumed to be ***D0*** and ***D1***.The creation process of the midpoint data from training data groups ***D0*** is stated as Fig 1.

This processing is done by all training data, and midpoint data is created. Midpoints data (*x, y*) are made up from Step1 to Step5.

3.2 Midpoint-validation method for SVM

Step 1: SVM is created by the known training data.
Step 2: Create the midpoint data using Midpoint data creation algorithm.
Step 3: The output value of SVM of the midpoint data and training data are computed.
Step 4: The value of B is computed according to Eq.(8).
Step 5: The output of the proposed SVM is also computed according to Eq.(6) and Eq.(7).

Fig 2. Flow of Midpoint-Validation Method for SVM

We introduce the adjustment method of SVM from the result obtained from the SVM that used the midpoint data created with Section 3.1. First, SVM is created using known training data. Next, the output value of SVM of the midpoint data and training data are computed. It is assumed that the desired output of SVM by the midpoint data is a value as nearly 0. Therefore, we assume that the midpoint data is near to the classifier line. Then, B from Eq.(6) is adjusted so that the SVM output of the midpoint data may become close to 0. The method is shown in Eq.(8), where M is number of midpoint data.

$$h(\mathbf{x})=g(\mathbf{x})+B \quad (6)$$

$$O=sign(h(\mathbf{x})) \quad (7)$$

$$B=-\frac{1}{M}\sum_{m=1}^{M} g(\mathbf{x}_m) \quad (8)$$

Therefore, SVM is adjusted in order for the output of the midpoint data to be set to nearly 0. We call this technique midpoint-validation method for SVM. The flow of this method is shown in Fig 2.

Midpoint-validation method is applicable to all the techniques of SVM and it is also easy since only one value of parameter B has to be computed.

4. Midpoint-Validation Algorithm

The published paper [10] showed that it was effective to use midpoint data in input space. The method of paper [10] is only adjusted for parameter B in order for the output of the midpoint data to be set to nearly 0. But parameter $\boldsymbol{w}$ were not adjusting. Then, our proposed new method used midpoint data to learning algorithm of SVM.

We propose midpoint-validation algorithm for SVM, which improves the generalization of SVM so that midpoint-validation error in input space is minimized. Eq.(9) is objective function of midpoint-validation error.

$$E = \frac{1}{M}\left|\sum_{m=1}^{M} \mathbf{w} \cdot \mathbf{x}_m + b\right| \tag{9}$$

Eq.(9) of midpoint-validation error is computed simultaneously. Furthermore, the state of SVM in case midpoint-validation error is the minimum is saved for use model. When midpoint-validation error is 1.0 or more, our proposed technique judges that deviation is large than margin of SVM and midpoint-validation method [10] is applied. We call this technique midpoint-validation algorithm for SVM. The flow of midpoint-validation algorithm is shown as below.

Step 1: Create the midpoint data using
Midpoint data creation algorithm.
Step 2: Learning Algorithm of original SVM.
And midpoint-validation error (Eq.(9))
is computed.
Step 3: The state of SVM (in case midpoint-validation
error is the minimum) is saved.
Step 4: **If** Training error less than ***e1***.
: Learning Algorithm stop and go to **Step 5**.
Else : go to **Step2**.
Step 5: **If** Midpoint-validation error lager than ***1.0***.
: Adjusting parameter B using
midpoint-validation method

Fig 3. Midpoint-Validation Algorithm for SVM

Improvement in generalization capability is more expectable by abolishing the deviation of the classifier line in input space by this proposed method using midpoint-validation error.

5. Simulations

In order to test the effectiveness of the midpoint-validation algorithm, we compared its performance with those of the original SVM (from our simulation results using Eq.(4) and Eq.(5)), SVM using midpoint-validation method [10] and tested our proposed method on several benchmark problems. We also applied it to a realistic 'real-world' problem. The data set was created by Johns Hopkins University and obtained from the database [7]. In this paper, we tested on several benchmark problems of Ionosphere, Pima Diabetes, Wisconsin breast cancer (Wisconsin), Sonar and Liver Disorders. We performed only one time using the proposed method. All experiments are the same conditions as database [7] in terms of separating the training data and test data. In this simulation, we used ***e1*** was *1e-6*.

We performed 10 trials (The value of Gaussian kernel parameter σ of different 10) with each method and the simulation results are shown in Table 1. The number of training data vectors, number of test data vectors and the dimensions of the data vectors are set to *n1* be, *n2* and *d* respectively. These three values of *n1, n2* and *d* used in our experiments are shown in Table 1. And, M is number of midpoint data. The results of SVM using midpoint-validation algorithm are better than SVM in fifth benchmark problems. The midpoint-validation algorithm almost has improved SVM in this simulation results. Although midpoint-validation algorithm is almost the same as midpoint-validation method, midpoint-validation algorithm has better results in the conditions of good parameters.

The comparison between SVM using midpoint-validation algorithm and other paper technique are summarized in Table 1. We compare its performance with those of *soft margin* SVM from software LIBSVM [11] (*C*-SVC). The other SVM experimental results are obtained from the published papers [12]. The results show that the proposed method has the best performance in two benchmark problems. But published paper [12] results showed the best performance in Sonar problem and Liver Disorders, where as in other problems the proposed method has better or same results. The method of paper [12] resembles our proposed method by creating a kernel using midpoint data. From Table 1, our proposed method and paper [12] technique have a speciality and a non-speciality by problem.

Table 1. Simulation Results (Testing Correctness [%]).
* t-test (significance level of 5%) had statistical significance than SVM

Dataset (*n1,n2* × *d*)	**SVM**		***C*-SVC[11]**	**CSVM[12]**	**SVM+MV[10]**		**Proposed Method**	
	Ave	**Best**	**-**	**-**	**Ave**	**Best**	**Ave**	**Best**
Ionosphere **(200,151 ×34) *M*=15**	88.7	97.4	**98.0**	92.7	89.8	97.4	87.2	**98.0**
Pima Diabetes **(576,192 ×8) *M*=79**	77.5	77.6	81.3	**82.3**	80.1*	81.3	80.6*	81.7
Wisconsin **(342,341 ×9) *M*=12**	91.7	97.1	**98.8**	-	98.5*	98.5	98.4*	**98.8**
Sonar **(104,104 ×60) *M*=9**	92.1	93.3	87.5	**98.1**	92.3	93.3	91.9	93.3
Liver Disorders **(230, 115×6) *M*=30**	63.3	63.5	51.3	**81.4**	77.0*	77.4	77.2*	78.3

6. Conclusions

SVM performs the large margin in the feature space. However, many experiment results showed that the boundary line created by SVM has deviation in input space. The deviation of SVM is considered to have been generated under the influence of kernel function parameter. Moreover, it is thought that "overfitting" was caused in SVM. Therefore, we think that the adjusting method gives SVM the improvement result in many problems by using the new function made by using the middle point data. The new function is one standard of fine-tuning to finality.

In this paper, we proposed a midpoint-validation algorithm which could be utilized to improve the generalization of SVM. The simulation results on some benchmark problems showed that proposed method be able to find the best performance in the two benchmark problems. The midpoint-validation algorithm had improved the deviation of the output of SVM by kernel function using midpoint data and also simpler than the previous SVM algorithm. Moreover, we think that the proposed method has eased the condition of the labyrinthian problem by using the average value of midpoint data outputs for Eq.(8), (9).

How to select more optimal parameter of proposed method can be further studied.

References

[1]C. Cortes and V.N. Vapnik, "Support Vector Networks", Machine Learning, 20(3), pp. 273-297, 1995.

[2]C.J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", Data Mining and Knowledge Discovery, vol.2,no. 2, 1998.

[3]T. Joachims, "Making Large-Scale Support Vector Machine Learning Practical", in Advances in Kernel Methods: Support Vector Learning, Shoelkopf, B.,Burges, C., Smola, A., eds., pp. 169-184, 1999.

[4]O.L. Mangasarian and D.R. Musicant, "Active support vector machines", Technical Report 00-04, Data Mining Institute, Computer Sciences Department, University of Wisconsin, madison,Wisconsin, April 2000.

[5]O.L. Mangasarian and D.R. Musicant, "Lagrange Support Vector Machines", Journal of Machine Learning Research Vol. 1, pp. 161-177, 2001.

[6]L.Prechelt, ``Automatic Early Stopping Using Cross Validation Quantifying the Criteria '', Neural Networks Vol, 11 No.4 , pp. 761-767, 1998.

[7]C.L. Blake, C.J Merz, UCI Repository of machine learning databases,
http://www.ics.uci.edu/~mlearn/MLRepository.html,
University of California, Department of Information and Computer Science, Irvine, CA, 1998.

[8]J. Weston, R. Collobert, F. Sinz, L. Bottou, V. Vapnik "Inference with the Universum", Proc of ICML'2006, Pittsburgh, USA, 2006.

[9]H.Tamura and K.Tanno, "Midpoint-Validation Method of Neural Networks for Pattern Classification Problems", International Journal of Innovative Computing, Information and Control, Vol.4, No.10 pp.2475-2482. 2008.

[10]H.Tamura and K.Tanno, "Midpoint-Validation Method for Support Vector Machine Classification", IEICE Trans. on Information and Systems Vol.E91-D No.7 pp.2095-2098. 2008.

[11]LIBSVM: http://www.csie.ntu.edu.tw/~cjlin/libsvm/
Chang and Lin 2001.

[12] T.Raicharoen, T. C.Lursinsap, "Critical support vector machine without kernel function", Proceedings of ICONIP '02, vol.5, pp.2532-2536, 2002.