LETTER

Midpoint-Validation Method for Support Vector Machine Classification

Hiroki TAMURA[†a)] ***and*** **Koichi TANNO**[†], ***Members***

SUMMARY In this paper, we propose a midpoint-validation method which improves the generalization of Support Vector Machine. The proposed method creates midpoint data, as well as a turning adjustment parameter of Support Vector Machine using midpoint data and previous training data. We compare its performance with the original Support Vector Machine, Multilayer Perceptron, Radial Basis Function Neural Network and also tested our proposed method on several benchmark problems. The results obtained from the simulation shows the effectiveness of the proposed method.
key words: *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 and regression problems [2]. 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 guaranteed generalization behavior due to the method of structural risk minimization.

A number of improved implementations of quadratic programming problems have been proposed to overcome problems such as decomposing into smaller problems like chunking, SVM*Light* [3], and Sequential Minimal Optimization. Other approaches of implementations to find the maximal margin are the Successive Over Relaxation, Relaxed Online Maximum Margin Algorithm, Active Support Vector Machine (abbr. ASVM) [4], and Lagrangian Support Vector Machines (abbr. LSVM) [5]. Also another category of implementation converts the problem to a problem of computing the nearest point between two convex polytopes and finding the closest points of opposite class like DirectSVM. Weston [14] proposed an algorithm to leverage the Universum by maximizing the number of observed contradictions, showing experimentally that this approach delivers accuracy improvements over using labeled data alone.

Cross-validation method is typical technique used in order to prevent the occurrence of over fitting [6] and it is also an evaluation method that has proven to be better than residuals. The disadvantage of using a residual evaluation is that it does not provide any indication of how well the learner will do when it is asked to make new predictions for data it has not already seen. This is resolved by refraining from the entire data set when training a learner. Instead, some of the data is removed before training begins, and when training is completed, the data that has been removed are then used to test the performance of the learned model on "new" data.

In this paper, we propose a midpoint-validation method which improves the generalization of SVM. The proposed method creates midpoint data and turning adjustment parameter of SVM the midpoint data and previous training data. With our proposed method it is no longer necessary for the data to be separated into two sets as in cross-validation. We compare its performance with those of the original SVM, Multilayer Perceptron (abbr. MLP), Radial Basis Function Neural Network (abbr. RBF) and tested our proposed method on several benchmark probelms. The results obtained from the simulation carried out shows the effectiveness of the proposed method.

The rest of this paper is organized into four sections. Section 2 reviews the concept of original SVM algorithm. Section 3 presents our proposed Midpoint-Validation Method for SVM. Section 4 provides the experimental results performed with several benchmark data and compares them with the others'. Section 5 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. The simplest model is called Linear SVM, and it works for data that are linearly separable in the original feature space only. In the early 1990s, nonlinear classification in the same procedure as Linear SVM became possible by introducing nonlinear functions called Kernel functions without being conscious of actual mapping space. This extended techniques of nonlinear feature spaces called as Nonlinear SVM.

Assume the training sample $S = ((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N))$ consisting of vectors $\boldsymbol{x}_i \in R$ with $i = 1, \ldots, N$, and each vector $\boldsymbol{x}_i$ belongs to either of the two classes. Thus it is given a label $y_i \in \{-1, 1\}$. The pair of $(\boldsymbol{w}, b)$ defines a separating hyper-plane of equation as follows:

$$(\boldsymbol{w}, \boldsymbol{x}) + b = 0 \tag{1}$$

Manuscript received November 30, 2007.
Manuscript revised March 20, 2008.
[†]The authors are with the Faculty of Engineering, University of Miyazaki, Miyazaki-shi, 889–2192 Japan.
a) E-mail: htamura@cc.miyazaki-u.ac.jp
DOI: 10.1093/ietisy/e91–d.7.2095

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 $\boldsymbol{x}_i$ is called support vector and the OSH (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.

$$\text{Maximize margin } \frac{1}{2}(\boldsymbol{w}, \boldsymbol{w})$$
$$\text{Subject to } y_i((\boldsymbol{w} \cdot x_i) + b) \geq 1$$

To solve the case of nonlinear decision surfaces, the OSH is carried out by nonlinearly transforming a set of original feature vectors $\boldsymbol{x}_i$ into a high-dimensional feature space by mapping $\boldsymbol{\Phi}$: $x_i \rightarrow z_i$ and then performing the linear separation. However, it requires an enormous computation of inner products $(\boldsymbol{\Phi}(\boldsymbol{x}) \cdot \boldsymbol{\Phi}(\boldsymbol{x}_i))$ in the high-dimensional feature space. Therefore, using a Kernel function which satisfies the Mercer's theorem given in Eq. (2) 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 $g(\boldsymbol{x})$ and output of SVM are as given in Eqs. (4) and (5).

$$(\Phi(\boldsymbol{x}) \cdot \Phi(\boldsymbol{x}_i)) = K(\boldsymbol{x}, \boldsymbol{x}_i) \tag{2}$$

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

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

$$O = sign(g(\boldsymbol{x})) \tag{5}$$

3. Midpoint-Validation Method for SVM

SVM performs the large margin in the feature space. However, many experiment results showed that the boundary line created by SVM has deviation. Figure 1 is the example of deviation of SVM. Figure 1 (a) consist of two classes problem by the input vector of X1 and X2. Training data $\boldsymbol{D0}$ is ○, $\boldsymbol{D1}$ is □ and test data is black diamond shape. The output of SVM which built from training data $\boldsymbol{D0}$ (○) and $\boldsymbol{D1}$ (□) is shown in Fig. 1 (b). Based on Fig. 1 (b), it is obvious that the output of SVM inclines toward $\boldsymbol{D1}$. The deviation of SVM is considered to have been generated under the influence of kernel function parameter. Although SVM constitutes of a large margin in feature space, it can be said that SVM has deviated from the influence of kernel function parameter in input space.

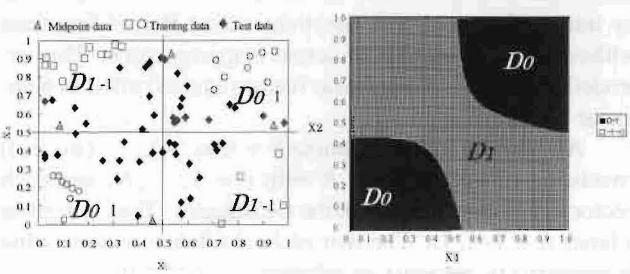
In this paper, we propose the method of decreasing the deviation of SVM by creating a midpoint data. The data based on the midpoint between classes are created for midpoint data. We expect that generalization will be improved by tuning the appropriate midpoint data. The SVM classifier generated on the information of the midpoint data is more reliable and accurate than the original SVM.

3.1 Creation of Midpoint Data

Midpoint data is 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 (−1 and 1). Training data groups that belongs to each teacher signal is assumed to be $\boldsymbol{D0}$ and $\boldsymbol{D1}$. The creation process of the midpoint data from training data groups $\boldsymbol{D0}$ is stated as below.

Midpoint data creation algorithm
Step 1: A training data (x_i) that belongs to group $\boldsymbol{D0}$ is selected accordingly. (Fig. 2: a)
Step 2: A training data (x_j) that has nearest distance in group $\boldsymbol{D1}$ is selected. (Fig. 2: b)
Step 3: A training data $(x_i{}^*)$ that has nearest distance in groups $\boldsymbol{D0}$ is also selected. (Fig. 2: c)
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. 2: x)

This processing is done by all training data, and mid-



(a) Input data (b) Output of SVM

Fig. 1 The example of deviation of SVM.

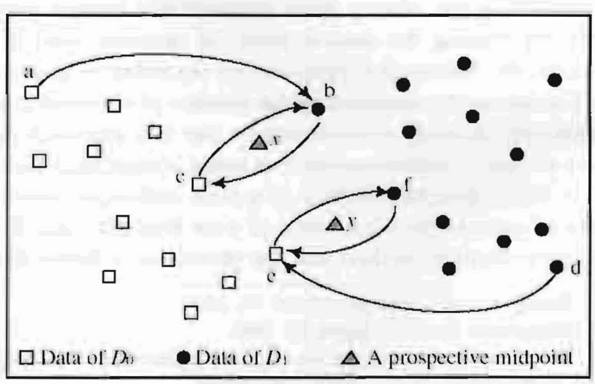


Fig. 2 Midpoint data creation. Midpoints data (x, y) are made up from Step1 to Step5.

point data is created (Fig. 2).

3.2 Adjustment Method for SVM Using Midpoint Data

We propose the adjustment method of SVM from the result obtained from the SVM that used the midpoint data created with Sect. 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.

$$h(\boldsymbol{x}) = g(\boldsymbol{x}) + B \tag{6}$$

$$O = sign(h(\boldsymbol{x})) \tag{7}$$

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

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 Classification. The flow of this method is shown as below.

Midpoint-Validation Method for Support Vector Machine Classification
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 Eqs. (6) and (7).

As a result of midpoint-validation method, the margin of SVM in feature space may turn to non-maximum. But, the improvement in generalization capability is more expectable by abolishing the deviation of the classifier line in input space by the midpoint-validation method. The midpoint-validation method is expected for the effect which eases the deviation of both input space and future space. Moreover, the proposed 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.

The result of the output of SVM using midpoint-validation method with regard to the problem in Fig. 1 (a) is shown in Fig. 3. Figure 3 shows that the output of SVM using midpoint-validation method had improved from the previous result of Fig. 1 (b).

4. Simulation Results

In order to test the effectiveness of the proposed method,

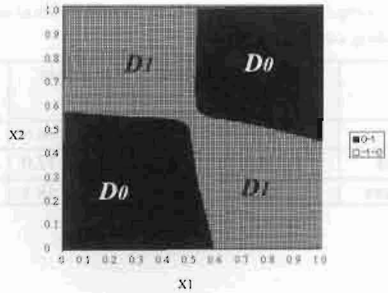


Fig. 3 The output of SVM using midpoint-validation method.

Table 1 Simulation results (Testing correctness [%]).

Dataset (n1,n2 × d)	RBF	MLP +CV	Original SVM	Proposed Method	
Ionosphere (200,151 ×34)	96.0[12]	96.0[9]	84.1	**97.4**	M=15
Pima Diabetes (576,192 ×8)	75.9[13]	76.4[10]	78.6	**80.7**	M=79
Wisconsin (342,341 ×9)	96.6[12]	96.7[10]	90.6	**98.5**	M=12
Sonar (104,104 ×60)	–	90.4[11]	93.3	**93.3**	M=9
Liver Disorders (230, 115×6)	–	76.8[our]	63.5	**77.4**	M=30

we compared its performance with those of the original SVM (from our simulation results using Eqs. (4) and (5)), MLP (from [8]–[11] and our simulation results), RBF (from [8], [12], [13]), and tested our proposed method on several benchmark problems. MLP results were obtained with the former cross-validation method. 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 conducted with the same conditions as the database [7] in terms of separating the training data and test data.

We performed 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 the proposed method are better than RBF, MLP in fifth benchmark problems. The original SVM and proposed method showed the same results in Sonar problem, where as in other problems the proposed method has improved. The comparison between our proposed method and other SVM techniques (LSVM, ASVM, SVM*light*) are summarized in Table 2. The other SVM experimental results are obtained from the published papers [5]. The results show that the proposed method has the best performance in third benchmark problems.

Table 2 The comparison between our proposed method and other SVM techniques (Testing correctness [%]).

	LSVM	ASVM	SVM*light*	Proposed Method
Ionosphere	87.8	87.8	88.6	**97.4**
Pima Diabetes	78.1	78.1	77.0	**80.7**
Liver Disorders	68.7	67.3	68.1	**77.4**

5. Conclusion

In this paper, we proposed a midpoint-validation method which could be utilized to improve the generalization of SVM. The simulation results on some benchmark problems showed that the proposed method is able to find the best performance in the fifth benchmark problems. The advantage of the proposed method is that the method becomes much more efficient compared to the former cross-validation method due to the numerical simulation used. Furthermore, the implementation is also very straightforward since only a minimal amount of calculation is required, which is only the computation for value B.

However, the margin of SVM using midpoint-validation method in feature space may turn to non-maximum. Although midpoint-validation method was effective for SVM in fifth benchmark problems as shown in the simulation, midpoint-validation method might not necessarily have an effective guarantee on all the problems. Therefore, in our future works, we intend to find the method of determining the conditions which are effective using midpoint-validation method.

References

[1] C. Cortes and V.N. Vapnik, "Support vector networks," Mach. Learn., vol.20, no.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, pp.121–167, 1998.

[3] T. Joachims, "Making large-scale support vector machine learning practical," in Advances in Kernel Methods: Support Vector Learning, eds. B. Shoelkopf, C. Burges, and A. Smola, pp.169–184, MIP Press, 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 Netw., vol.11, no.4, pp.761–767, 1998.

[7] C.L. Blake and 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] W. Duch, Datasets used for classification: Comparison of results http://158.75.5.90/kmk/projects/datasets.html, Nicolaus Copernicus University.

[9] V.G. Sigillito, S.P. Wing, L.V. Hutton, and K.B. Baker, "Classification of radar returns from the ionosphere using neural networks," Johns Hopkins APL Technical Digest, vol.10, pp.262–266, Johns Hopkins University, Baltimore, MD, 1989.

[10] B. Ster and A. Dobnikar, "Neural networks in medical diagnosis: Comparison with other methods," Proc. International Conference EANN '96, pp.427–430, 1996.

[11] R.P. Gorman and T.J. Sejnowski, "Analysis of hidden units in a layered network trained to classify sonar targets," Neural Netw., vol.1, pp.75–89, 1988.

[12] M. Wallace, N. Tsapatsoulis, and S. Kollias, "Intelligent initialization of resource allocating RBF networks," Neural Netw., vol.18, no.2, pp.117–122, 2005.

[13] D. Michie, D.J. Spiegelhalter, and C.C. Taylor, Machine learning, neural and statistical classification, Elis Horwood, London, 1994.

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