Twofold Cross-Validation Is Better than Five- or Tenfold

Shigeo Abe
Kobe University
Kobe, Japan
abe@kobe-u.ac.jp

***Abstract*—Model selection by k-fold cross-validation is a very useful way of realizing a support vector machine with a high generalization ability. The number of folds is usually five or ten. And thus, cross-validation is time consuming especially for large-size problems. In this paper first we show that under some assumption, for any classifier the cross-validation accuracies increase as the number of folds increases. (This does not imply that the larger number of folds realizes the better generalization ability.) Then, we propose using twofold cross-validation instead of five- or tenfold. We discuss the advantage of twofold cross-validation from the speed-up, and adjustment of parameter values when underfitting occurs. By computer experiments using two-class and multiclass problems, we demonstrate that the generalization abilities by twofold cross-validation are statistically comparable to five- or tenfold cross-validation with a considerable speed-up.**

I. Introduction

Model selection is very important in realizing a support vector machine (SVM) with a high generalization ability. To estimate the generalization ability, cross-validation is usually used. In cross-validation, we split the training set into k ($k \geq 2$) subsets with nearly equal sizes. And holding out one subset, we train the SVM using the remaining subsets, and calculate the number of correctly classified data of the held-out subset. This procedure is iterated k times changing the held-out subsets, and the accuracy for the total training set is calculated. In model selection, which is usually done by grid search, cross-validation is repeated for different values of hyperparameters at the grid points and the hyperparameter values with the best accuracy are selected as optimal. (For the speed-up of grid search please see [1].)

Because cross-validation is slow, there have been proposed several speed-up methods [2]–[7]. In [4], considering that for $k \geq 3$, the SVM trained in cross-validation shares $k-2$ training subsets, cross-validation is sped up by reusing the solution of the previous SVM for training for the current SVM.

In addition to model selection, k-fold cross-validation is used for estimating the generalization ability of the classification problem [8]–[11]. In [10], it is experimentally studied whether cross-validation should be repeatedly evaluated to obtain a reliable accuracy estimate. Computer experiments show that cross-validation with a large number of k and a small number of repetition is desired.

There have been many discussions on how to set the number of folds in cross-validation. Usually, 5 or 10 is selected [12], [13]. But because the model selection time increases rapidly as k increases, $k = 5$ is widely used. In [14], the number of folds ($k \geq 3$) is determined during model selection. According to the experiment using the two-class problems, which are also used in our experiments, the value mainly lies between 3 and 4, and $k = 4$ is a good choice if a fixed value is used.

The hyperparameter values selected by model selection may not be optimal for training the SVM using the training data because the number of data for model selection is $(k-1)/k$ times the number of the training data. In [13], the value of margin parameter C is proposed to multiply by $k/(k-1)$.

In [5], to speed up model selection by cross-validation, k-fold cross-validation of the SVM is approximated by the continuous Huber loss instead of the hinge loss and the criterion to select the number of folds is proposed.

In this paper, we propose to use twofold cross-validation to speed up model selection. For model selection, an accurate estimate discussed in [10] is not necessary. If the highest accuracies are near among different numbers of folds, model selection with a small folds of cross-validation will give a reasonably good result. The aim of our paper is to clarify whether twofold cross validation gives comparable performance with five- or tenfold cross-validation with a considerable speed up.

First, we discuss the effect of the number of folds to the cross-validation accuracy. Under some assumptions, we show that the cross-validation accuracy increases as the number of folds increases with the upper bound of the accuracy without cross-validation, namely, the training data accuracy. We call this onefold cross-validation accuracy. Then we discuss how cross-validation time changes as the number of folds is changed.

If we use twofold cross-validation, the number of data used for cross-validation is half the number of training data. Therefore, there may be cases where underfitting occurs because of the small C value. We propose a method for detecting underfitting and tuning of the C value for avoiding underfitting.

Using two-class and multiclass problems, the advantages of twofold cross-validation against five- ten-, and onefold cross-validation is demonstrated and the avoidance of underfitting is tested.

In Section II, we briefly summarize the architecture of the SVM. Then in Section III, we discuss the effect of the number of folds to the cross-validation accuracy, the computation cost

of cross-validation, and avoidance of underfitting. In Section IV, we evaluate the advantage of twofold cross-validation over five-, ten-, and onefold cross-validation for the two-class and multiclass problems.

Codes will be included in Programs in https://www2.kobe-u.ac.jp/˜abe/home-svmpc/index.html.

II. Support Vector Machines

The standard SVM, which is often called L1 SVM, classifies two-class data by the following separating hyperplane in the l-dimensional feature space that is mapped from the input vector $\mathbf{x}$ by the mapping function $\boldsymbol{\phi}(\mathbf{x})$:

$$D(\mathbf{x}) = \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}) + b = 0, \tag{1}$$

where $\mathbf{w}$ is the l-dimensional coefficient vector and b is the bias term.

The L1 SVM maximizes the minimum margin between the separating hyperplane and the training data and minimizes the classification error with the positive margin parameter C. It is given by

$$\min\ Q(\mathbf{w}, b, \boldsymbol{\xi}) = \frac{1}{2}\,\|\mathbf{w}\|^2 + C \sum_{i=1}^{M} \xi_i \tag{2}$$

$$\text{s.t. } y_i\,(\mathbf{w}^\top\,\boldsymbol{\phi}(\mathbf{x}_i) + b) + \xi_i \geq 1,\ \xi_i \geq 0,\ i = 1, \ldots, M, \tag{3}$$

where $\{\mathbf{x}_i, y_i\}$ $(i = 1, \ldots, M)$ are M training data and their labels, $\mathbf{x}_i$ are the n-dimensional input vectors and $y_i = 1$ for Class 1 and -1 for Class 2, ξ_i are the slack variables for the ith inequality constraints, and $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_M)^\top$.

The dual form of the L1 SVM is given by

$$\max \quad Q(\boldsymbol{\alpha}) = \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{M} \alpha_i\,\alpha_j\,y_i\,y_j\,K_{ij} \tag{4}$$

$$\text{s.t.} \quad \sum_{i=1}^{M} y_i\,\alpha_i = 0, \tag{5}$$

$$C \geq \alpha_i \geq 0,\ i = 1, \ldots, M, \tag{6}$$

where α_i $(i = 1, \ldots, M)$ are the Lagrange multipliers, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_M)^\top$, K_{ij} is the kernel, and $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\phi}^\top(\mathbf{x}_i)\,\boldsymbol{\phi}(\mathbf{x}_j)$.

In our following study we use the RBF kernel:

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma\,\|\mathbf{x} - \mathbf{x}'\|^2), \tag{7}$$

where γ is the parameter to control the width spread.

The Lagrange multipliers α_i $(i = 1, \ldots, M)$ are associated with the inequality constraints given by (3) and work to maximize the minimum margin.

The KKT complementarity conditions are given by

$$\alpha_i \left(y_i \left(\sum_{j=1}^{M} y_j\,\alpha_j\,K_{ij} + b \right) - 1 + \xi_i \right) = 0, \quad i = 1, \ldots, M, \tag{8}$$

$$(C - \alpha_i)\,\xi_i = 0, \quad i = 1, \ldots, M, \tag{9}$$

$$\alpha_i \geq 0, \quad \xi_i \geq 0, \quad i = 1, \ldots, M. \tag{10}$$

If $\alpha_i = 0$, then $\xi = 0$. We call $\mathbf{x}_i$ non-support vectors. They are correctly classified and have margins more than or equal to 1. If $C \geq \alpha_i > 0$, $\mathbf{x}_i$ are called support vectors. Especially, if $C > \alpha_i > 0$, then $\xi_i = 0$. We call $\mathbf{x}_i$ unbounded support vectors. They are correctly classified and have margins of 1. (We call the margin of 1 full margin.)

If $\alpha_i = C$, then $\xi_i \geq 0$. The associated $\mathbf{x}_i$ are called bounded support vectors. If $1 \geq \xi_i \geq 0$, $\mathbf{x}_i$ are correctly classified although the margins may not be full. We call them correctly classified support vectors. If $\xi_i \geq 1$, $\mathbf{x}_i$ are misclassified. We call them misclassified support vectors.

The decision function given by (1) is rewritten as follows:

$$D(\mathbf{x}) = \sum_{i \in S} \alpha_i\,y_i\,K(\mathbf{x}_i, \mathbf{x}) + b, \tag{11}$$

where S is the set of support vector indices.

For non-support vectors $\mathbf{x}_i$, $y_i D(\mathbf{x}) \geq 1$. For unbounded support vectors, $\mathbf{x}_i$, $y_i D(\mathbf{x}) = 1$. And for bounded support vectors, $\mathbf{x}_i$, $y_i D(\mathbf{x}) = 1 - \xi_i$.

III. Cross-validation

In the following, first we explain k-fold cross-validation. Then, we discuss the dependence of cross-validation accuracy on the number of folds and the computation cost of cross-validation. And finally, we discuss avoiding underfitting caused by the gap between the number of data used for cross-validation and that for training after model selection.

A. Procedure for Cross-validation

In k-fold cross-validation, first we randomly split the training data set into k subsets with almost equal sizes, V_i^k $(i = 1, \ldots, k)$. Leaving V_j^k $(j = 1, \ldots, k)$ out, we train the SVM using $\cup_{i=1, i\neq j}^{k} V_i^k$ with the given hyperparameter values. Then we classify V_j^k and count the number of correctly classified data, $N_{V_j^k}$. We repeat the above procedure for $j = 1, \ldots, k$ and calculate the cross-validation accuracy, $A_{\rm cv}^k$

$$A_{\rm cv}^k = \frac{\sum_{j=1}^{k} N_{V_j^k}}{M}, \tag{12}$$

where $M = |\cup_{i=1}^{k} V_i^k|$.

In model selection, we do not repeat cross-validation for the given hyperparameter values. Therefore, to make model selection stable, we fix V_i^k throughout model selection.

B. Effect of the Number of Folds on Cross-Validation Accuracy

In this section, we discuss the relation between the training data accuracy and the cross-validation accuracy and how the cross-validation accuracy changes as the number of folds changes. In the following we consider using the L1 SVM with the fixed γ and C values and assume that initially the training data are randomly ordered, and are split into k subsets in order. Therefore, for instance for tenfold cross-validation, the first two subsets are equal to the first subset of fivefold cross-validation.

The training data accuracy is given by

$$A_{\rm tr} = \frac{\sum_{j=1}^{k} N^0_{V_j^k}}{M}. \tag{13}$$

where $N^0_{V_j^k}$ is the number of correctly classified data for data subset V_j^k, and the SVM is trained using all the training data.

Let the number of misclassified training data be #MSV. Then,

$$A_{\rm tr} = 1 - \frac{\#MSV}{M}. \tag{14}$$

If the classification problem is linearly separable under the given hyperparameter values, #MSV$= 0$. Therefore,

$$A_{\rm tr} \geq A^k_{\rm cv} \quad \text{for } i = 1, \ldots, k. \tag{15}$$

To extend the above relation to nonlinear problems, we assume that *the test data accuracy is non-decreasing if the test data are added to the training data:*

$$N^0_{V_j^k} \geq N_{V_j^k}. \tag{16}$$

In the left-hand side of the above inequality, the classifier is trained using the training data used in the right-hand side plus data in V_j^k, namely all the training data. Because the data in V_j^k are more likely to be correctly classified if data in V_j^k are included in the training data, (16) is a reasonable assumption.

From (12), (13), and (16), (15) also holds for the general case.

To consider the relation of accuracies when the number of folds changes, we further assume that *if more training data are added, the test data accuracy is non-decreasing:*

$$N_{V_{j'}^{k'} \cap V_j^k} \geq N_{V_j^k \cap V_{j'}^{k'}} \quad \text{for } \; k' \geq k, \tag{17}$$

where $N_{V_{j'}^{k'} \cap V_j^k}$ is the number of correctly classified data for $V_{j'}^{k'} \cap V_j^k$ classified by the classifier trained using the training data excluding $V_{j'}^{k'}$.

In the left-hand side of the above inequality, the classifier is trained using the training data used in the right hand side plus the data in $V_j^k - V_{j'}^{k'}$. But because the added data in $V_j^k - V_{j'}^{k'}$ are not included in $V_{j'}^{k'} \cap V_j^k$, the assumption given by (17) is more likely to be violated than that given by (16). Violation of (17) may occur when the numbers of correctly classified data are small as the following computer experiment shows.

It is clear that

$$\sum_{j'=1}^{k'} \sum_{j=1}^{k} N_{V_{j'}^{k'} \cap V_j^k} = \sum_{j'=1}^{k'} N_{V_{j'}^{k'}}, \tag{18}$$

$$\sum_{j'=1}^{k'} \sum_{j=1}^{k} N_{V_j^k \cap V_{j'}^{k'}} = \sum_{j=1}^{k} \sum_{j'=1}^{k'} N_{V_j^k \cap V_{j'}^{k'}} = \sum_{j=1}^{k} N_{V_j^k}. \tag{19}$$

Therefore, from (12), (17)-(19),

$$A^{k'}_{\rm cv} \geq A^k_{\rm cv} \quad \text{for } \; k' \geq k. \tag{20}$$

Combining (15) with the above, we obtain

$$A_{\rm tr} \geq A^k_{\rm cv} \geq A^{k-1}_{\rm cv} \geq \cdots \geq A^2_{\rm cv}. \tag{21}$$

For example, if $k' = 2k$,

$$V_{2j}^{2k} \cup V_{2j-1}^{2k} = V_j^k \quad \text{for } j = 1, \ldots, k. \tag{22}$$

Applying (17) to (22),

$$N_{V_{2j}^{2k}} + N_{V_{2j-1}^{2k}} \geq N_{V_j^k} \quad \text{for } j = 1, \ldots, k. \tag{23}$$

Therefore, from (12),

$$A^{2k}_{\rm cv} \geq A^k_{\rm cv}. \tag{24}$$

We must notice that (21) holds for any classifier so long as (16) and (17) are satisfied.

C. Cost of Cross-validation

In one epoch of k-fold cross-validation, training with the data size of $(k-1)M/k$ is repeated k times. The SVM is formulated by quadratic programming. Thus, complexity of computation of the SVM with M training data is M^3 if the problem is solved without decomposition. Accordingly, we assume the amount of computation of k-fold cross-validation T_k by

$$T_k = a\, k \left(\frac{k-1}{k}\right)^3 M^3. \tag{25}$$

Here we assume that a is a constant independent of k.

Then the ratio of T_k/T_2 is given by

$$T_k/T_2 = \frac{4(k-1)^3}{k^2} \tag{26}$$

For instance $T_5/T_2 = 10.24$ and $T_{10}/T_2 = 29.16$. Thus we can expect speed-up of cross-validation employing twofold, instead of five- or tenfold.

D. Adjustment of C Values

The optimal values obtained by cross-validation may not be optimal for the original training set because the data sizes for cross-validation and training are different. To adjust non-optimality of the selected values by this gap, adjustment of the C value is proposed [13]. Because the hyperparameter C is multiplied to the sum of slack variables, its effect changes according to the number of training data. To adjust this, the following adjustment is proposed

$$C = \frac{k}{k-1}\, C_{\rm opt}, \tag{27}$$

where $C_{\rm opt}$ is the C value selected by model selection.

For $k = 5$, $C = 1.25\, C_{\rm opt}$ and for $k = 2$, $C = 2\, C_{\rm opt}$.

The problem of using $C_{\rm opt}$ for training the original set is a possibility of underfitting. As an initial experiment, we used this for $k = 2$. In rare cases, underfitting occurred even when we used (27) and in most cases, applying (27) resulted in accuracy degradation.

Therefore, instead of (27), we use the following method.

From (21), if

$$A_{\rm tr} < A^k_{\rm cv}, \tag{28}$$

we consider that underfitting occurred. This is avoided by increasing the C value, which works to improve the training

data accuracy, $A_{\rm tr}$. Assume that C_i is selected by model selection, where C_i is the ith candidate of C and C_i is arranged in the increasing order, i.e., $C_i < C_{i+1}$. The procedure of detecting and avoiding underfitting is as follows:

1) Set $C_{\rm opt} \leftarrow C_i$, train the SVM with $C_{\rm opt}$, and go to Step 2).
2) If underfitting occurs, i.e., (28) is satisfied, $C_i \leftarrow C_{i+1}$ and go to Step 1). Otherwise, terminate the algorithm.

In Step 2), if C_i is the upper most candidate, we terminate the procedure. In this, case we need to increase the upper bound of C to resolve underfitting.

IV. Performance Evaluation

We used two-class and multiclass problems used in [15], [16]. The two-class problems have 100 or 20 pairs of training and test data sets. The multiclass problems have 20 or 5 training and test data sets.

For model selection we used twofold, fivefold, and tenfold cross-validation, and as a reference we used training data accuracy for model selection. We call the model selection using the training data accuracy onefold cross-validation.

In cross-validation, the candidate values for γ and C were the same as those used in [15], [16]; γ values from $\{0.01, 0.1, 0.5, 1, 5, 10, 15, 20, 50, 100, 200\}$ and C values from $\{0.1, 1, 10, 50, 100, 500, 1000, 2000\}$.

After model selection, we trained the SVM using the selected hyperparameter values and tested the test data, and calculated the average accuracy and the standard deviation for each problem. Statistical difference caused by the number of folds was tested by the t-test with the confidence level of 95%.

If underfitting occurred after training the SVM with the determined hyperparameter values, namely, (28) was satisfied, as discussed in Section III-D we increased the C value until the underfitting was resolved.

A. Trends of Cross-Validation Accuracies

To examine how the number of folds affects the cross-validation accuracies, we plot the trends of cross-validation accuracies of the first file of the breast-cancer problem for the change of the γ or C value. Figure 1 shows the cross-validation accuracies for the change of the γ value fixing $C = 1$. For twofold, the accuracy is peaked at $\gamma = 10$. Both for fivefold and tenfold, the accuracies are peaked at $\gamma = 15$. For twofold, the accuracies for the $\gamma = 10$ and 15 are the same. Thus smaller value of 10 is selected. For the three folds, the tendencies of peaks are similar. But for onefold, the accuracies increases as the value of γ is increased.

For $\gamma = 200$, the opposite directions of inequalities occurred in (21). This means that the assumption given by (17) does not hold at this value.

Fig. 2 shows the cross-validation accuracies for the changes of the C value with $\gamma = 15$. For twofold, fivefold, and tenfold, the peaks are reached at $C = 1$. And for $C = 50$, the accuracies of twofold and fivefold are reversed: the relation given by (21) is violated for twofold and fivefold at this value.

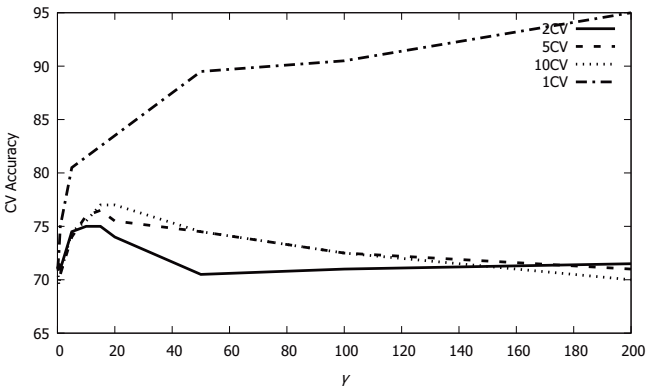

Fig. 1. Cross-validation accuracies of the breast-cancer problem for the changes of the γ values

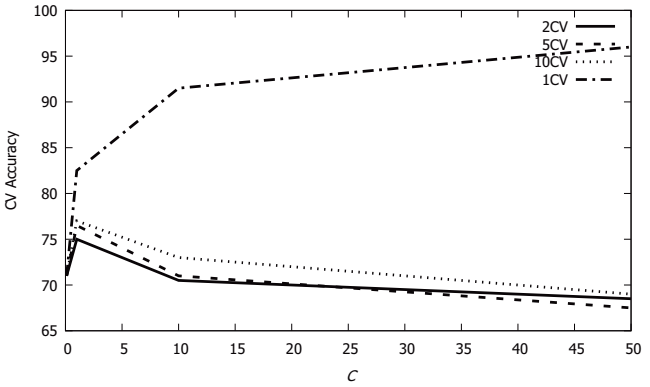

Fig. 2. Cross-validation accuracies of the breast-cancer problem for the changes of the C values

For onefold, the accuracies increase as the value of C is increased. Therefore, it is unfit for model selection.

The above results exemplify the validity of taking twofold cross-validation for model selection.

The violations of assumption (21) occurred when the differences of accuracies are small. Namely, the assumption (17) tends to be violated when the difference of accuracies is small.

B. Two-class Problems

Table I shows the average test data accuracies for the two-class problems. In the "Problem I/Tr/Te" column, "Problem" lists the name of the problem and each of "I/Tr/Te" denotes the number of inputs, the number of training data, and the number of test data, respectively. For each problem the average accuracy and the standard deviation are listed from the second to the fifth columns.

The bold and underlined figures show the best and worst accuracies among the first to the third results. "Average" denotes the average accuracy of the accuracies for the 13 two-class problems and each in "B/S/W" in parentheses denotes that the best accuracies were obtained B times, the second best, S times, and the worst, W times. "W/T/L" denotes the

results of statistical analysis: "W" denotes the number of wins for the SVM by twofold cross-validation over that by five-, ten- or onefold cross-validation, "T", the number of ties, and "L", the number of losses. The "+" symbol shows that the twofold cross-validation shows the statistically better accuracy than onefold and the "−" symbol, statistically worse.

There is no statistical difference between twofold and fivefold cross-validation, and between twofold and tenfold cross-validation. But between twofold and onefold cross-validation, twofold cross-validation is statistically better than onefold 10 times, and worse once. This means that by onefold cross-validation, overfitting occurred for most of the problems and that at least twofold cross-validation was necessary except for the thyroid and titanic problems.

From the Average measure and the B/S/W measure, twofold cross-validation performed best.

The standard deviations of the ringnorm and titanic problems by twofold cross-validation are large compared to those by other numbers of folds. This indicates very low accuracies caused by underfitting. For the ringnorm problem, underfitting occurred for the 78th file. The twofold cross-validation accuracy was 98.50% with $\gamma = 200$ and $C = 0.1$. The training data accuracy was 84.50%. Thus according to (28), overfitting occurred. With the increase of the C value to 1, the accuracies for the training and test data sets became 100% and 96.643%, respectively. Thus the overfitting was resolved. The corrected average accuracy and the standard deviation were 98.215%, which was the best, and 0.288, respectively.

For the titanic problem, overfitting occurred for the 38th file. The best twofold cross-validation accuracy was 80.67% with $\gamma = 1$ and $C = 10$. The training and test data accuracies were 80.00% and 69.797%, respectively. By increasing the C value to 50, the training and test data accuracies were 81.333% and 73.574%, respectively. The corrected average accuracy and the standard deviation were 77.411%, which was the best, and 0.782, respectively.

Therefore, for the both problems, underfitting was detected and resolved by the proposed method.

Table II list the average cross-validation accuracies for the two-class problems. Excluding the onefold results, for each problem, we show the best accuracy in bold and the worst, underlined. For each problem, the maximum average accuracy was obtained by onefold cross-validation, which validates (15).

The tenfold cross-validation shows the best accuracy among the three folds except for the diabetes problems and the twofold cross-validation the worst. This shows the validity of (21) for most of the problems.

Although the accuracies of tenfold cross-validation show the best and the twofold the worst, this does not affect the test data accuracies: on average twofold cross-validation tends to show the best accuracies.

Table III shows the execution time of cross-validation, training the SVM, and classification of the test data set, normalizing the twofold cross-validation time to 1. In the last two rows, we show the average and median values of the execution time. Compared to twofold cross-validation, on average, fivefold cross-validation is 25.7 times slower and the tenfold cross-validation is 76.4 times slower. By the median values, they are halved but still much slower than twofold cross-validation. From the execution time also, twofold cross-validation has an advantage over fivefold and tenfold cross-validation.

C. Multiclass Problems

The data sets for each of the multiclass problems originally consist of one training data set and one test data set. So we combine them into one, and randomly split it to one training data set and one test data set, and generate 20 training and test data sets.

Table IV shows the average test data accuracies and their standard deviations of the eleven multiclass problems. For the MNIST problem, we use the first five training and test data set to shorten the experiment time. In the first row of the first column, "O" denotes the number of classes. In the MNIST (r) problem, the roles of the training and test data sets of the original MNIST problem were switched.

We used fuzzy pairwise classification [17] to resolve unclassifiable regions.

To speed up model selection, we selected the γ value from 1, 5, 10, and 15 for the USPS problem, from 10, 15, 20, and 50 for the MNIST (r) and MNIST problems, and from 50, 100, and 200 for the letter problem.

For the USPS problem, the average accuracies and the standard deviations for the twofold and fivefold cross-validation are the same. The γ values of 20 were selected for both twofold and fivefold cross-validation, and the C values of 10 were selected for 18 times each. The C value of 50 was selected twice for twofold cross-validation and once for fivefold cross-validation. And the C value of 100 was selected once for fivefold cross-validation. Thus, the model selection results for the one file differed in C values but the accuracies were the same. Therefore, the average accuracies and the standard deviations by twofold and fivefold cross-validation were the same, although the cross-validation accuracies were different (see Table V).

For the MNIST problem, for two-, five-, and tenfold cross-validation, the same C and γ values were selected for all the five training data sets and thus the average test data accuracies are the same.

Tenfold cross-validation shows the best average accuracies among 11 problems and the twofold cross-validation, the worst. But the difference of the average accuracies is 0.021% and is much smaller than that for the two-class problems shown in Table I (0.126%). The "B/S/W" measures for the twofold and fivefold are similar but that for the tenfold is a little worse.

From the statistical analysis, twofold and fivefold, and twofold and tenfold have no statistical difference but twofold cross-validation is statistically better than onefold for nine problems. This also shows the importance of cross-validation in model selection.

TABLE I
AVERAGE TEST DATA ACCURACIES FOR THE TWO-CLASS PROBLEMS

Problem I/Tr/Te	Twofold	Fivefold	Tenfold	Onefold
Banana 2/400/4900	**89.280**±0.619	89.151±0.790	89.161±0.589	$^{+}$84.650±1.281
Cancer 9/200/77	**73.351**±4.348	72.935±4.649	72.714±4.586	$^{+}$65.247±4.922
Diabetes 8/468/300	**76.443**±1.727	76.347±1.823	76.150±1.706	$^{+}$67.933±2.146
Flare-solar 9/666/400	66.965±1.961	**67.063**±2.102	66.983±2.164	$^{+}$63.905±1.889
German 20/700/300	76.057±2.236	76.060±2.140	**76.193**±2.026	$^{+}$69.493±2.622
Heart 13/170/100	**82.990**±3.573	82.900±3.610	82.420±3.705	$^{+}$76.210±3.626
Image 18/1300/1010	**97.356**±0.423	97.158±0.410	97.282±0.594	97.342±0.435
Ringnorm 20/400/7000	98.030±2.007	**98.136**±0.342	98.094±0.391	$^{+}$97.102±0.506
Splice 60/1000/2175	**89.039**±0.738	88.892±0.842	88.922±0.854	$^{+}$88.262±0.701
Thyroid 5/140/75	95.267±2.266	**95.427**±2.468	95.293±2.418	$^{-}$95.933±2.169
Titanic 3/150/2051	77.343±1.260	77.361±0.773	**77.392**±0.690	77.626±1.023
Twonorm 20/400/7000	**97.410**±0.292	97.389±0.265	97.384±0.294	$^{+}$95.992±0.713
Waveform 21/400/4600	89.657±0.827	**89.722**±0.689	89.560±0.802	$^{+}$87.637±0.853
Average (B/S/W)	**85.322** (7/1/5)	85.272 (4/6/3)	85.196 (2/6/5)	82.103 (—)
W/T/L	—	0/13/0	0/13/0	10/2/1

TABLE II
AVERAGE CROSS-VALIDATION ACCURACIES FOR THE TWO-CLASS PROBLEMS

Problem	Twofold	Fivefold	Tenfold	Onefold
Banana	90.183	90.528	**90.603**	97.847
Cancer	75.260	75.715	**75.735**	98.395
Diabetes	77.615	**78.015**	77.996	100.000
Flare-solar	67.523	67.584	**67.638**	70.437
German	76.099	76.481	**76.609**	100.000
Heart	83.729	84.441	**84.665**	100.000
Image	96.408	97.204	**97.415**	100.000
Ringnorm	98.465	98.588	**98.623**	100.000
Splice	87.180	88.650	**89.015**	99.985
Thyroid	97.007	97.507	**97.636**	100.000
Titanic	79.087	79.380	**79.420**	80.447
Twonorm	97.933	98.067	**98.085**	100.000
Waveform	90.278	90.797	**91.073**	100.000
Average	85.905	86.381	**86.501**	95.932
B/S/W	0/0/13	1/12/0	12/1/0	—

TABLE III
AVERAGE EXECUTION TIME COMPARISON

Problem	Twofold	Fivefold	Tenfold	Onefold
Banana	1.0	24.0	64.9	2.1
Cancer	1.0	150.2	505.4	41.6
Diabetes	1.0	12.6	33.9	9.2
Flare-solar	1.0	4.9	11.9	1.8
German	1.0	14.9	42.5	8.5
Heart	1.0	40.8	163.6	4.8
Image	1.0	5.5	12.2	1.7
Ringnorm	1.0	11.1	30.9	2.9
Splice	1.0	12.8	32.9	6.6
Thyroid	1.0	20.8	7.0	0.8
Titanic	1.0	9.0	13.5	156.5
Twonorm	1.0	12.6	33.5	4.8
Waveform	1.0	15.3	41.5	4.4
Average	1.0	25.7	76.4	18.9
Median	1.0	12.8	33.5	4.8

Table V shows the average cross-validation accuracies. For all the eleven multiclass problems, (21) is satisfied. Therefore, the higher cross-validation accuracy does not leads to higher test data accuracy. So the comparison of generalization ability by the cross-validation accuracy may lead to erroneous results.

Table VI shows the average execution time setting the twofold cross-validation to 1. Execution time includes cross-validation time, training time using the hyperparameter values determined by cross-validation, and classification time of the test data set. Using twofold cross-validation, the execution speed is 4.3 times and 9.6 times faster than using fivefold and tenfold cross-validation, respectively.

From the multiclass problems also, twofold cross-validation is shown to speed up cross-validation with the comparable test accuracies compared to those by five- or tenfold cross-validation.

V. CONCLUSIONS

In this paper we proposed using twofold cross-validation instead of five- or tenfold. Assuming reasonable assumptions, we first showed that for any classifier the cross-validation accuracies increase as the number of folds is increased. And the upper bound is the onefold cross-validation accuracy, which is the training data accuracy without cross-validation. We also proposed a method for detecting underfitting and avoiding it.

By computer experiments using the SVM, we showed that the test data accuracy by twofold cross-validation was statistically comparable to that by fivefold or tenfold cross-validation and the execution speed was much faster.

Underfitting rarely occurred by twofold cross-validation. But when it happened, it was detected and avoided by the proposed method.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number 25K07795.

REFERENCES

[1] S. Abe. Accelerating grid search by pruning. In *2025 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9, Rome, Italy, 2025.

[2] Z. Wen, R. Zhang, K. Ramamohanarao, J. Qi, and K. Taylor. MASCOT: Fast and highly scalable SVM cross-validation using GPUs and SSDs. In *2014 IEEE International Conference on Data Mining*, pages 580–589, 2014.

TABLE IV
AVERAGE TEST DATA ACCURACIES FOR THE MULTICLASS PROBLEMS

Problem I/O/Tr/Te	Twofold	Fivefold	Tenfold	Onefold
Numeral 12/10/810/820 [17]	99.579±0.257	**99.598**±0.291	99.591±0.265	99.652±0.314
Thyroid 21/3/3772/3428 [18]	97.491±0.232	**97.535**±0.215	97.493±0.242	+96.152±0.288
Blood cell 13/12/3097/3100 [17]	**94.437**±0.280	94.390±0.269	94.344±0.287	+93.269±0.412
Hiragana-50 50/39/4610/4610 [17]	99.728±0.078	**99.734**±0.084	99.730±0.086	+99.528±0.112
Hiragana-13 13/38/8375/8356 [17]	99.683±0.073	**99.720**±0.062	99.710±0.062	99.685±0.064
Hiragana-105 105/38/8375/8356 [17]	**99.967**±0.030	99.964±0.030	99.965±0.034	+99.828±0.074
Satimage 36/6/4435/2000 [18]	91.475±0.590	91.608±0.616	**91.780**±0.548	+90.290±0.517
USPS 256/10/7291/2007 [19]	**97.783**±0.324	**97.783**±0.324	97.755±0.340	+97.200±0.531
MNIST(r) 784/10/10000/60000 [20]	97.090±0.074	97.091±0.072	**97.096**±0.071	+96.827±0.072
Letter 16/26/16000/4000 [18]	**97.836**±0.155	97.815±0.153	97.830±0.171	+97.426±0.209
MNIST 784/10/60000/10000 [20]	**98.566**±0.060	**98.566**±0.060	**98.566**±0.060	+98.420±0.065
Average (B/S/W)	97.603 (5/0/6)	97.618 (6/3/2)	**97.624** (3/6/2)	97.116 (—)
W/T/L	—	0/11/0	0/11/0	9/2/0

TABLE V
AVERAGE CROSS-VALIDATION ACCURACIES FOR THE MULTICLASS PROBLEMS

Problem	Twofold	Fivefold	Tenfold	Onefold
Numeral	99.500	99.741	**99.747**	100.000
Thyroid	97.010	97.418	**97.513**	99.948
Blood cell	94.026	94.583	**94.696**	100.000
Hiragana-50	99.150	99.608	**99.710**	100.000
Hiragana-13	99.283	99.640	**99.704**	100.000
Hiragana-105	99.839	99.950	**99.964**	100.000
Satimage	90.882	91.723	**92.052**	100.000
USPS	97.202	97.657	**97.768**	99.999
MNIST(r)	96.093	96.780	**96.908**	100.000
Letter	96.353	97.426	**97.643**	100.000
MNIST	98.134	98.471	**98.548**	100.000
Average	97.043	97.545	**97.659**	99.995
B/S/W	0/0/11	0/11/0	11/0/0	—

TABLE VI
AVERAGE EXECUTION TIME FOR THE MULTICLASS PROBLEMS

Problem	Twofold	Fivefold	Tenfold	Onefold
Numeral	1.0	3.2	6.8	1.1
Thyroid	1.0	5.5	12.5	1.9
Blood cell	1.0	3.1	6.3	1.3
Hiragana-50	1.0	3.7	8.5	1.3
Hiragana-13	1.0	2.7	4.9	1.2
Hiragana-105	1.0	5.2	13.1	1.5
Satimage	1.0	4.5	11.0	2.2
USPS	1.0	3.8	8.4	1.8
MNIST(r)	1.0	4.2	9.2	1.4
Letter	1.0	4.2	9.7	1.9
MNIST	1.0	2.7	5.6	1.4
Average	1.0	4.3	9.6	1.7

[3] J. Zhang and S. Wang. A fast leave-one-out cross-validation for SVM-like family. *Neural Computing and Applications*, 27:1717–1730, 2016.

[4] Z. Wen, B. Li, R. Kotagiri, J. Chen, Y. Chen, and R. Zhang. Improving efficiency of SVM k-fold cross-validation by alpha seeding. arXiv:1611.07659v2, 2017.

[5] Y. Liu and S. Liao. Granularity selection for cross-validation of SVM. *Information Sciences*, 378:475–483, 2017.

[6] H. A. Fayed and A. F. Atiya. Speed up grid-search for parameter selection of support vector machines. *Applied Soft Computing*, 80:202–210, 2019.

[7] F. Mohr and J. N. van Rijn. Fast and informative model selection using learning curve cross-validation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):9669–9680, 2023.

[8] J. D. Rodriguez, A. Perez, and J. A. Lozano. Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):569–575, 2010.

[9] G. Jiang and W. Wang. Error estimation based on variance analysis of k-fold cross-validation. *Pattern Recognition*, 69:94–106, 2017.

[10] T.-T. Wong and P.-Y. Yeh. Reliable accuracy estimates from k-fold cross validation. *IEEE Transactions on Knowledge and Data Engineering*, 32(8):1586–1594, 2020.

[11] L. R. Olsen. Multiple-k: Picking the number of folds for cross-validation https://cran.r-project.org/web/packages/cvms/vignettes/picking_the_number_of_folds_for_cross-validation.html. 2025.

[12] C.-W. Su, C.-C. Chang, and C.-J. Lin. A practical guide to support vector classification. In *Technical report, Dept. of Computer Science, National Taiwan University*, 2003.

[13] D. Anguita, A. Ghio, S. Ridella, and D. Sterpi. K-fold cross validation for error rate estimate in support vector machines. In *International Conference on Data Mining*, pages 291–297, 2009.

[14] D. Anguita, L. Ghelardoni, A. Ghio, L. Oneto, and S. Ridella. The 'K' in K-fold cross validation. In *ESANN*, volume 102, pages 441–446, 2012.

[15] S. Abe. Training minimal complexity support vector machines with multiple kernels. In *Proceedings of 2023 IEEE International Conference on Systems, Man and Cybernetics (SMC 2023)*, pages 913–919, Honolulu, Hawaii, October 2023.

[16] S. Abe. Are bounded support vectors harmful to stable classification? In *Proceedings of International Joint Conference on Neural Networks*, pages 1–7, Yokohama, Japan, July 2024.

[17] S. Abe. *Support Vector Machines for Pattern Classification*. Springer-Verlag, London, UK, second edition, 2010.

[18] A. Asuncion and D. J. Newman. UCI machine learning repository, http://www.ics.uci.edu/∼mlearn/MLRepository.html. 2007.

[19] USPS Dataset. https://www.kaggle.com/bistaumanga/usps-dataset.

[20] Y. LeCun and C. Cortes. The MNIST database of handwritten digits http://yann.lecun.com/exdb/mnist/.