Information Sciences 378 (2017) 475–483

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Granularity selection for cross-validation of SVM

Yong Liu, Shizhong Liao*

School of Computer Science and Technology, Tianjin University, Tianjin 300072, P. R. China

ARTICLE INFO

Article history:
Received 30 August 2015
Revised 25 April 2016
Accepted 28 June 2016
Available online 29 June 2016

Keywords:
Cross-validation
Model selection
Fold of cross-validation
Granular computing
Granularity selection

ABSTRACT

Granularity selection is fundamental to granular computing. Cross-validation (CV) is widely adopted for model selection, where each fold of data set of CV can be considered as an information granule, and the larger the number of the folds is, the smaller the granularity of each fold is. Therefore, for CV, granularity selection is equal to the selection of the number of folds. In this paper, we explore the granularity selection for CV of support vector machine (SVM). We first use the Huber loss to smooth the hinge loss used in SVM, and to approximate CV of SVM. Then, we derive a tight upper bound of the discrepancy between the original and the approximate CV with a high convergence rate. Finally, based on this derived tight bound, we present a granularity selection criterion for trading off the accuracy and time cost. Experimental results demonstrate that the approximate CV with the granularity selection criterion gives the similar accuracies as the traditional CV, and meanwhile significantly improves the efficiency.

1. Introduction

Granular computing has a wide range of applications in data mining, pattern recognition and machine learning [21,28–30]. How to select a proper granularity is one of the fundamental issues in the research and application of granular computing [8,21,28,32,33]. Cross-validation (CV) [20,24] is a tried and tested approach for selecting the optimal model [9,18,19], which is widely used in granular computing. In t-fold CV, the data set is split into t disjoint subset of (approximately) equal size and the algorithm (or model) is trained for t times, each time leaving out one of subsets from training, but using the omitted subset to compute the validation error. The t-fold CV estimate is then the average of the validation errors observed in t iterations, or folds. Each subset of t-fold CV can be considered as the information granule [13,21,23,32], the larger number of folds means the smaller granularity of each subset and the higher cost. Therefore, for CV, granularity selection is equal to the selection of the number of folds, which is a key problem to CV.

Support vector machine (SVM) is an important machine learning method widely adopted in granular computing [22,25,26,31]. The performance of SVM greatly depends on the choice of some hyper-parameters (such as the kernel parameter and regularization parameter), hence how to select the optimal hyper-parameters is important to SVM [1,14,16]. Although the t-fold CV is a commonly used approach for selecting the hyper-parameters for SVM [2,4,17], it requires training t times, which is computationally intensive. For the sake of efficiency, some approximate leave-one-out CV for SVM are given: such as generalized approximate cross-validation (GACV) [27], radius-margin bound [26], span bound [5], support vector count [26]. However, there is few work on the approximation of the general t-fold CV (for all t). Instead of using the full grid, the local search heuristics is used to find local minima in the validation error to speed up the computation of CV

* Corresponding author.
E-mail addresses: szliao@mail.tju.edu.cn, szliao@tju.edu.cn (S. Liao).

http://dx.doi.org/10.1016/j.ins.2016.06.051
0020-0255/

[10,11]. In [12], an improved CV procedure is proposed, which uses nonparametric testing coupled with sequential analysis to determine the best parameter set on linearly increasing subsets of the data. Different from the above approximate CV methods that speed up the grid-search procedure, in our previous work [15], we present a strategy for approximating the CV error for a class of kernel-based algorithms, in which the loss function must be differentiable. Unfortunately, the hinge loss used in SVM is not differentiable, so the approximate strategy proposed in [15] can not be used for SVM.

In this paper, we present an approximate CV approach for SVM, and further present a novel granularity selection method for it. Specifically, we first use the Huber loss to approximate the hinge loss, and give an approach to approximating the CV of SVM using the Huber loss. Then, we derive a tight upper bound of the discrepancy between the original and approximate CV errors of order $\mathcal{O}\left(\frac{1}{t\cdot r}\right)$, where t is the number of folds and r is the order of Taylor expansion. Finally, based on the derived tight bound, we present a granularity selection criterion to trade off the performance of approximation and the computational cost. The proposed approximate CV requires training on the full data set only once, hence it can significantly improve the efficiency. Experimental results demonstrate that the approximate CV with the granularity selection criterion is sound and efficient.

The rest of the paper is organized as follows. We start by introducing some preliminaries and notations in Section 2. We then propose a novel strategy for approximating the CV of SVM in Section 3. In Section 4, we present a granularity selection criterion to choose the number of folds. We empirically analyze the performance of our approximate CV with the granularity selection criterion in Section 5. We end in Section 6 with conclusion. All the proofs are given in Appendix.

2. Preliminaries and notations

We consider the supervised learning where a learning algorithm receives a sample of n labeled points

$$S = \{z_i = (\mathbf{x}_i, y_i)\}_{i=1}^n, z_i \in \mathcal{Z} = (\mathcal{X} \times \mathcal{Y}),$$

where $\mathcal{X}$ denotes the input space and $\mathcal{Y} = \{-1, +1\}$ the output space. We assume S is drawn identically and independently from a fixed, but unknown probability distribution P on $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a Mercer kernel [3], and assume $K(\mathbf{x}, \mathbf{x}) \le 1$, $\forall \mathbf{x} \in \mathcal{X}$[1] . The reproducing kernel Hilbert space (RKHS) associated with K is defined to be the completion of the linear span of the set of functions $\mathcal{H}_K = \text{span}\{\Phi(\mathbf{x}) = K(\mathbf{x}, \cdot) : \mathbf{x} \in \mathcal{X}\}$ with the inner product denoted as $\langle \cdot, \cdot \rangle_K$ satisfying $\left\langle K(\mathbf{x}, \cdot), K(\mathbf{x}', \cdot) \right\rangle_K = K(\mathbf{x}, \mathbf{x}')$. The learning algorithms we study is SVM [7,26]:

$$f_{P_S}^{\text{svm}} := \arg\min_{f \in \mathcal{H}_K} \frac{1}{|S|} \sum_{z_i \in S} \ell(y_i f(\mathbf{x}_i)) + \lambda \|f\|_K^2,$$

where $\ell(\cdot)$ is the hinge loss $\ell(t) = \max(0, 1 - t)$, λ is the regularization parameter, and $|S|$ is the size of S.

Let $S_1, \ldots, S_t$ be a random equipartition of S into t parts, called folds. For simplicity, assume that n mod t, and hence, $|S_i| = \frac{n}{t} =: l$, $i = 1, \ldots, t$. Each S_i can be considered as an information granule [13,23,32]. Note that the larger t, the smaller size of S_i, which implies the smaller granularity of S_i. Thus, the selection of fold can be regarded as the granularity selection in CV.

Let $P_{S\setminus S_i}$ be the empirical distribution of the sample S without the observations S_i, that is

$$P_{S\setminus S_i} = \frac{1}{n-l} \sum_{z_i \in S\setminus S_i} \delta_{z_i}, \tag{1}$$

where δ_{z_i} is the Dirac distribution in z_i. The hypothesis learned on all of the data excluding S_i can be written as:

$$f_{P_{S\setminus S_i}}^{\text{svm}} := \arg\min_{f \in \mathcal{H}_K} \frac{1}{n-l} \sum_{z_i \in S\setminus S_i} \ell(y_i f(\mathbf{x}_i)) + \lambda \|f\|_K^2.$$

Then, the t-fold CV error can be written as

$$t\text{-CV} := \frac{1}{n} \sum_{i=1}^{t} \sum_{z_j \in S_i} I\left(y_j f_{P_{S\setminus S_i}}^{\text{svm}}(\mathbf{x}_j)\right),$$

where, $I(c) = 1$ if $c < 0$, otherwise 0. Although the t-CV is wildly used for model selection, it requires training t times, which is computationally expensive. In our previous work [15], we present a strategy to approximate the t-CV based on Bouligand influence function (BIF) [6] for some kernel-based algorithms, in which the loss function must be differentiable. This approximate CV needs to be trained only once, hence it can significantly improve the efficiency. Unfortunately, the hinge loss used in SVM is not differentiable so that the approximate strategy proposed in [15] can not be used for SVM, directly. To address this problem, in the next section, we will propose to use a differentiable approximation of the hinge loss, inspired by the Huber loss.

[1] $K(\mathbf{x}, \mathbf{x}) \le 1$ is a common assumption, for example, met for the popular Gaussian kernel and Laplace kernel.

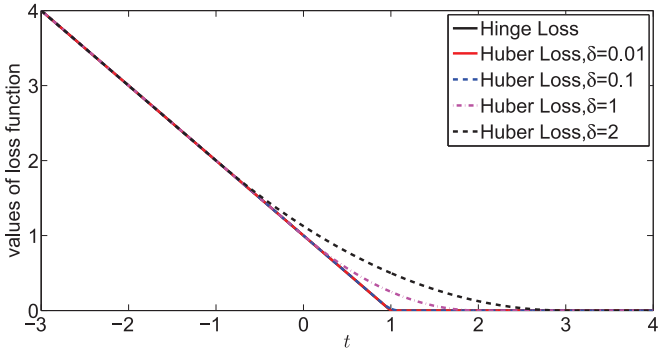

Fig. 1. Hinge loss vs. Huber loss with different δ, $\delta = 2, 1, 0.1, 0.01$.

3. Approximate cross-validation of SVM with Huber loss

In this section, we will use the Huber loss to smooth the hinge loss, and further to approximate CV of SVM.

3.1. Huber loss

The Huber loss with respect to $\delta > 0$ is given as follows:

$$\ell_\delta(t) = \begin{cases} 0 & \text{if } t > 1+\delta, \\ \dfrac{(1+\delta-t)^2}{4\delta} & \text{if } |1-t| \le \delta, \\ 1-t & \text{if } t < 1-\delta. \end{cases}$$

Noth that Huber loss are differentiable. In Fig. 1, we can see that Huber loss is a very good approximation of hinge loss when δ is not very large. In fact, when $\delta \le 0.1$, hinge loss and Huber loss are almost the same.

The following theorem will theoretically verify the effectiveness of the approximation of hinge loss using Huber loss for SVM.

Theorem 1. *Denote*

$$R(f) = \frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y_i f(\mathbf{x}_i)) + \lambda \|f\|_K^2$$

and

$$R^\delta(f) = \frac{1}{n}\sum_{i=1}^{n} \ell_\delta(y_i f(\mathbf{x}_i)) + \lambda \|f\|_K^2,$$

where $\ell_\delta(\cdot)$ is the Huber loss with respect to δ. Let $f_{P_S}^{\text{svm}}$ and $f_{P_S}^{\delta}$ be the minimizer of $R(f)$ and $R^\delta(f)$, that is

$$f_{P_S}^{\text{svm}} = \arg\min_{f\in\mathcal{H}_K} R(f) \text{ and } f_{P_S}^{\delta} = \arg\min_{f\in\mathcal{H}_K} R^\delta(f)$$

Then, the following inequation holds:

$$R(f_{P_S}^{\text{svm}}) \le R(f_{P_S}^{\delta}) \le R(f_{P_S}^{\text{svm}}) + \frac{\delta}{2}.$$

According to the above theorem, we know that the small δ can guarantee the performance of approximation.

3.2. Approximate cross-validation with Huber loss

In this subsection, we will approximate the t-CV with Huber loss. The Huber SVM is defined as follows:

$$f_{P_S}^{\delta} := \arg\min_{f\in\mathcal{H}_K} \frac{1}{|S|}\sum_{z_i\in S} \ell_\delta(y_i f(\mathbf{x}_i)) + \lambda \|f\|_K^2. \tag{2}$$

We say that a point $\mathbf{x}_i$ is a *support vector* if $|1 - y_i(f^{\delta}_{P_S}(\mathbf{x}_i))| \le \delta$. Let $\mathbf{I}^0$ be the $n \times n$ diagonal matrix with the first n_{sv} entries being 1 and the others 0. Note that Huber loss is differentiable,

$$\ell'_\delta(t) = \begin{cases} 0 & \text{if } t > 1+\delta, \\ \dfrac{-(1+\delta-t)}{2\delta} & \text{if } |1-t| \le \delta, \\ -1 & \text{if } t < 1-\delta, \end{cases} \text{ and } \ell''_\delta(t) = \begin{cases} 0 & \text{if } t > 1+\delta, \\ \dfrac{1}{2\delta} & \text{if } |1-t| \le \delta, \\ 0 & \text{if } t < 1-\delta. \end{cases}$$

So the approximate CV method proposed in [15] can be directly used for Huber SVM.

In the following, we will briefly show how to use BIF to approximate CV (more detail seen in [15]). To this end, let $P_S = \frac{1}{n}\sum_{z_i \in S} \delta_{z_i}$ be the sample distribution, and $P_{S_i} = \frac{1}{l}\sum_{z_i \in S_i} \delta_{z_i}$ be the empirical distribution corresponding to the ith fold S_i. One can see that

$$P_{S\setminus S_i} = \left(1 - \left(\frac{-1}{t-1}\right)\right)P_S + \frac{-1}{t-1}P_{S_i}.$$

Thus, the $P_{S\setminus S_i}$ can be considered as a perturbation of the P_S.

Let $f^{\delta} : P \to f^{\delta}(P) =: f^{\delta}_P \in \mathcal{H}_K$. The BIF and high order BIF [15] are used to measure the impact of an infinitesimal small amount of contamination of the original distribution P, and the BIF and high order BIF are the first and high derivative order of f^{δ} at P. Thus, from Taylor expansion, if all BIFs exist, we have

$$f^{\delta}_{P_{S\setminus S_i}}(\mathbf{x}_j) \approx f^{\delta}_{P_S}(\mathbf{x}_j) + \sum_{s=1}^{r}\left[\frac{-1}{t-1}\right]^s \frac{1}{s!}[\mathbf{B}^i_s]_j, \tag{3}$$

where r is the order of Taylor expansion, $\mathbf{B}^i_1$ and $\mathbf{B}^i_{k+1}$ are the first and $k+1$th order BIF at P_S with respect to S_i, which can be computed as

$$\mathbf{B}^i_1 = -\mathbf{L}_n^{-1}\left[\frac{1}{l}[\mathbf{K}\otimes\mathbf{S}_i]\begin{bmatrix} y_1\ell'_\delta(y_1 f^{\delta}_{P_S}(\mathbf{x}_1)) \\ \vdots \\ y_n\ell'_\delta(y_n f^{\delta}_{P_S}(\mathbf{x}_n)) \end{bmatrix} + 2\lambda\begin{bmatrix} f^{\delta}_{P_S}(\mathbf{x}_1) \\ \vdots \\ f^{\delta}_{P_S}(\mathbf{x}_n) \end{bmatrix}\right],$$

$$\mathbf{B}^i_{k+1} = (k+1)\mathbf{L}_n^{-1}\left[\frac{1}{n}\mathbf{K}\mathbf{I}_0\mathbf{B}^i_k - \frac{1}{2\delta l}[[\mathbf{K}\mathbf{I}_0]\otimes\mathbf{S}_i]\mathbf{B}^i_k\right],$$

where, $\mathbf{L}_n := 2\lambda\mathbf{I}_n + \frac{1}{2\delta n}\mathbf{K}\mathbf{I}^0$, $\mathbf{S}_i$ is an $n \times n$ matrix with $[\mathbf{S}_i]_{j,k} = 1$ if $\mathbf{x}_k \in S_i$, 0 otherwise, and $\otimes$ is the entrywise matrix product (also known as the Hadamard product).

Thus, the approximate t-CV of Huber SVM can be written as:

$$\mathrm{BIFCV}^t_r = \frac{1}{n}\sum_{i=1}^{t}\sum_{z_j\in S_i} I\left(y_j\left(f^{\delta}_{P_S}(\mathbf{x}_j) + \sum_{s=1}^{r}\left[\frac{-1}{t-1}\right]^s\frac{[\mathbf{B}^i_s]_j}{s!}\right)\right). \tag{4}$$

To compute BIFCV^t_r, we need to compute $f^{\delta}_{P_S}$ on the full data set, the inversion of $\mathbf{L}_n$ and the BIF matrices. The time complexity of computing $f^{\delta}_{P_S}$ and the inversion of $\mathbf{L}_n$ is $O(n^3_{\mathrm{sv}})$, and the time complexity of BIF matrices is $O(t\cdot n\cdot n_{\mathrm{sv}} + r\cdot n\cdot n_{\mathrm{sv}})$, where n_{sv} is the size of support vectors, n is the size of the data set, t is the number of folds, and r is the order of the Taylor expansion. Thus, the overall time complexity is $O(n^3_{\mathrm{sv}} + t\cdot n\cdot n_{\mathrm{sv}} + r\cdot n\cdot n_{\mathrm{sv}})$.

For the traditional t-CV, the algorithm need to be executed t times, so the time complexity is $O\left(tn^3_{\mathrm{sv}}\right)$. Thus, the proposed approximate t-CV is much more efficient.

4. Granularity selection

In this section, we will first derive a tight upper bound of the discrepancy between the original (hinge SVM) and the approximate CV, and then present a granularity selection criterion to select the number of folds.

The upper bound of the discrepancy between t-CV and BIFCV^t_r is given as follows:

Theorem 2. *Let $t-\mathrm{CV}$ be the t-fold CV error,*

$$t-\mathrm{CV} := \frac{1}{n}\sum_{i=1}^{t}\sum_{z_j\in S_i} I(y_j f^{\mathrm{svm}}_{P_{S\setminus S_i}}(\mathbf{x}_j)),$$

and BIFCV^t_r the approximate t-fold CV error

$$\mathrm{BIFCV}^t_r := \frac{1}{n}\sum_{i=1}^{t}\sum_{z_j\in S_i} I\left(y_j\left(f^{\delta}_{P_S}(\mathbf{x}_j) + \sum_{s=1}^{r}\left[\frac{-1}{t-1}\right]^s\frac{[\mathbf{B}^i_s]_j}{s!}\right)\right).$$

Table 1
The selections of number of folds and the order of Talor expansion with respect to the approximation ϵ. In this table, we set $\delta = 0.01$, $\lambda = 1$, $\kappa = 1$.

Approximation error ϵ	Fold t	Order r	Time
0.2	4	2	$O(n_{sv}^3 + 6n \cdot n_{sv})$
0.1	5	3	$O(n_{sv}^3 + 8n \cdot n_{sv})$
0.05	6	4	$O(n_{sv}^3 + 10n \cdot n_{sv})$
0.01	16	14	$O(n_{sv}^3 + 30n \cdot n_{sv})$

Table 2
Datasets.

Datasets	♯Instances	♯Attributes
Sonar	208	60
Heart	270	13
Liver-disorders	345	112
Ionosphere	351	34
Breast-cancer	683	10
Australian	690	14
Diabetes	768	8
Fourclass	862	2
German.numer	1000	24
a2a	2265	123

Then, the following inequation holds:

$$\left| t - \mathrm{CV} - \mathrm{BIFCV}_r^t \right| \leq \frac{\delta}{2} + \frac{1}{\lambda(r+1)(t-1)}.$$

According to the above theorem, one can see that the performance of the approximation of CV is dependent on t and r. The larger t and r, the better performance of approximation but the higher computational cost. Thus, we need to select the t and r to trade off the performance of approximation and the time cost.

Assume ϵ is the approximation error that the customer can bear. It is ideal to choose the smallest number of t and r that satisfies the requirement of customer. Thus, it is reasonable to use the following criterion to choose t and r:

$$\begin{aligned} &\arg\min_{t,r\in\mathbb{N}_+} t + r, \\ &\text{s.t. } \frac{1}{\lambda(r+1)(t-1)} \leq \epsilon - \frac{\delta}{2}. \end{aligned} \tag{5}$$

Using the above criterion, from Theorem 2, we know that the approximation error is smaller than ϵ, which satisfies the requirement of customer. Note that the solution of the optimization problem (5) can be written as

$$t - 1 = r + 1 = \left\lceil \sqrt{\frac{\kappa}{\lambda(\epsilon - \delta/2)}} \right\rceil, \tag{6}$$

where $c = \lceil x \rceil$ is the smallest integer satisfying $c \geq x$. Thus, we can choose the t and r according to equation (6). Some special cases of the selection of t and r by equation (6) are given in Table 1.

Remark 1. The selection of number of folds is equal to the granularity selection for CV. Thus, for different granularity, that is different number of folds, the relative order of performances of all compared algorithms may be different. But, for any fixed number of folds, the proposed approximate CV is approximately consistent, that is the approximate CV converges to the original CV when $\delta \to 0$, $r \to \infty$.

5. Experiments

In this section, we will empirically analyze the performance of the proposed approximate CV with the granularity selection criterion.

The data sets are 10 publicly available sets from LIBSVM Data[2] seen in Table 2. We use the popular Gaussian kernel

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(\frac{-\|\mathbf{x} - \mathbf{x}'\|_2^2}{2\sigma}\right)$$

[2] http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Table 3
The average test errors (%) of ϵ-BIF, t-CV and GACV [27] for Gaussian kernel and Polynomial kernel, $\epsilon = 0.2, 0.1, 0.05$, $t = 5, 10, 20$. For each training set, we choose the kernel parameter and regularization parameter by each criterion on the training set, and evaluate the test error for the chosen parameters on the test set.

Gaussian Kernel							
Data sets	0.2-BIF	0.1-BIF	0.05-BIF	5-CV	10-CV	20-CV	GACV
Australian	15.01 ± 1.22	14.78 ± 1.10	14.67 ± 1.49	15.19 ± 1.12	14.96 ± 0.84	14.90 ± 0.78	15.29 ± 1.02
Heart	18.37 ± 4.10	17.19 ± 2.31	17.04 ± 2.16	18.37 ± 0.97	17.78 ± 1.89	17.33 ± 1.78	16.67 ± 7.86
Ionosphere	7.89 ± 1.48	7.43 ± 1.07	7.31 ± 1.10	7.09 ± 1.04	6.63 ± 1.25	7.09 ± 1.04	7.86 ± 2.69
Breast-cancer	3.28 ± 0.60	3.11 ± 0.49	3.11 ± 0.49	3.28 ± 0.48	3.05 ± 0.74	2.99 ± 0.52	4.74 ± 0.34
Diabetes	23.85 ± 1.86	24.22 ± 1.41	24.06 ± 1.31	23.33 ± 1.99	24.06 ± 1.82	23.59 ± 1.48	23.91 ± 1.23
Fourclass	0.19 ± 0.13	0.09 ± 0.13	0.09 ± 0.13	0.19 ± 0.30	0.28 ± 0.30	0.28 ± 0.30	1.74 ± 0.27
German.numer	26.48 ± 1.08	26.36 ± 1.30	26.36 ± 1.30	25.00 ± 0.55	25.08 ± 0.64	24.96 ± 0.46	24.50 ± 0.24
Liver-disorders	32.79 ± 1.46	32.21 ± 3.20	33.12 ± 1.87	32.67 ± 1.95	31.98 ± 1.01	32.21 ± 3.80	32.77 ± 6.80
Sonar	22.50 ± 8.78	20.77 ± 5.55	20.77 ± 5.55	21.35 ± 8.34	17.31 ± 4.56	17.12 ± 7.05	19.48 ± 3.42
a2a	18.82 ± 4.39	18.73 ± 4.43	18.71 ± 4.42	18.52 ± 1.38	18.98 ± 1.33	18.66 ± 1.33	19.00 ± 2.29
Polynomial Kernel							
Data sets	0.2-BIF	0.1-BIF	0.05-BIF	5-CV	10-CV	20-CV	GACV
Australian	14.03 ± 1.02	14.20 ± 1.08	13.97 ± 0.97	14.03 ± 1.02	13.97 ± 0.97	14.20 ± 1.08	13.97 ± 1.68
Heart	20.30 ± 1.71	19.41 ± 2.42	18.96 ± 3.08	20.30 ± 1.71	19.26 ± 2.46	18.67 ± 2.53	19.22 ± 3.65
Ionosphere	5.34 ± 1.37	5.23 ± 1.36	5.11 ± 1.21	5.00 ± 1.23	5.34 ± 1.37	5.34 ± 1.37	7.93 ± 0.93
Breast-cancer	3.76 ± 1.15	3.58 ± 1.25	3.56 ± 1.52	3.57 ± 1.19	3.68 ± 1.05	3.74 ± 1.14	3.51 ± 1.09
Diabetes	22.66 ± 1.21	22.76 ± 1.63	22.45 ± 1.53	22.66 ± 1.21	22.55 ± 1.28	22.60 ± 1.49	22.19 ± 1.43
Fourclass	4.36 ± 0.89	4.36 ± 0.89	4.36 ± 0.89	4.36 ± 0.89	4.36 ± 0.89	4.36 ± 0.89	4.36 ± 0.89
German.numer	25.32 ± 2.20	24.76 ± 2.06	23.45 ± 2.13	24.80 ± 2.09	25.36 ± 2.22	24.92 ± 2.07	25.04 ± 1.65
Liver-disorders	31.10 ± 2.50	31.79 ± 2.98	31.68 ± 2.93	31.45 ± 1.99	31.79 ± 2.98	31.68 ± 2.93	31.79 ± 2.80
Sonar	16.54 ± 2.08	16.73 ± 1.99	15.96 ± 0.86	16.15 ± 1.72	16.92 ± 2.85	15.38 ± 1.36	15.77 ± 1.61
a2a	19.51 ± 0.59	19.63 ± 0.74	19.73 ± 0.81	19.21 ± 0.56	19.31 ± 0.64	19.40 ± 0.61	19.17 ± 0.68

and polynomial kernel

$$K(\mathbf{x}, \mathbf{x}') = \left(\langle \mathbf{x}, \mathbf{x}' \rangle + 1\right)^d$$

as our candidate kernels, $\sigma \in \{2^i, i = -10, -9, \ldots, 9, 10\}$ and $d \in \{1, 2, \ldots, 8\}$. The regularization parameter $\lambda \in \{2^i, i = -3, -2, \ldots, 11\}$. For each data set, we run all the methods 10 times with data sets being split randomly (50% of all the examples for training and the other 50% for testing). The use of multiple training/test partitions allows an estimate of the statistical significance for the performance of different methods. Let A_i and B_i be the test errors of methods A and B in partition i, and $d_i = B_i - A_i$, $i = 1, \ldots, 10$. Let $\bar{d}$ and S_d be the mean and standard error of d_i. Then under t-test, with confidence level 95%, we claim that A is significantly better than B if the t-statistic $\frac{\bar{d}}{S_d/\sqrt{10}} > 1.833$. All statements of statistical significance in the paper refer to a 95% level of significance. Experiments are conducted on a Dell PC with 3.1 GHz 4-core CPU and 4 GB memory.

5.1. Accuracy

The average test error of traditional t-CV ($t = 5, 10, 20$), the popular generalized approximate cross-validation (GACV) [27] and our ϵ-BIF ($\epsilon = 0.2, 0.1, 0.05$) with the granularity criterion are reported in Table 3. For each training set, for ϵ-BIF, we first select the t and r based on the granularity selection criterion, and then choose the optimal σ for Gaussian kernel or d for polynomial kernel, and λ by minimizing the approximate CV proposed in (4). Finally, we evaluate the test errors for the chosen parameters on the test set. For t-CV and GACV, they don't need to choose the t and r, and the other steps are the same as ϵ-BIF. The results in Table 3 can be summarized as follows:

(1) The smaller ϵ the better performance of our proposed approximate CV, which is conform to our theoretical analysis.
(2) The test errors of ϵ-BIF and CV are very close. In particular, (a) for polynomial kernel, neither BIF with $\epsilon = 0.1$ (or $\epsilon = 0.05$, or $\epsilon = 0.2$) nor t-CV ($t = 5, 10, 20$) criterion is significantly better than the other on any of the data sets; (b) for Gaussian kernel, neither BIF with $\epsilon = 0.2$ (or 5-CV) nor 10-CV (or 20-CV) is significantly better than the other on 9/10 data sets, but significantly worse than on sonar.
(3) BIF is significantly better than GACV on breast-cancer and fourclass for Gaussian kernel, on ionosphere for polynomial kernel, but without being significantly worse on any of the remaining data sets.

The above experimental results demonstrate that the quality of the proposed approximate CV with the granularity selection criterion is quite good.

Table 4
The average computational time (in second) of ϵ-BIF, t-CV and GACV [27] for Gaussian kernel and polynomial kernel, $\epsilon = 0.2, 0.1, 0.05$, $t = 5, 10, 20$.

Gaussian Kernel							
Data sets	0.2-BIF	0.1-BIF	0.05-BIF	5-CV	10-CV	20-CV	GACV
Australian	4.35 ± 0.15	5.52 ± 0.12	6.99 ± 0.07	8.79 ± 0.11	19.49 ± 0.21	41.29 ± 0.56	2.75 ± 0.12
Heart	1.15 ± 0.41	1.41 ± 0.30	1.87 ± 0.12	2.21 ± 0.03	4.78 ± 0.08	9.80 ± 0.09	0.73 ± 0.03
Ionosphere	1.97 ± 0.14	2.25 ± 0.22	3.12 ± 0.10	4.70 ± 0.10	10.24 ± 0.26	21.27 ± 0.56	1.34 ± 0.04
Breast-cancer	2.74 ± 0.07	3.90 ± 0.13	4.29 ± 0.13	5.60 ± 0.10	12.42 ± 0.23	26.00 ± 0.40	1.77 ± 0.06
Diabetes	4.43 ± 0.06	5.86 ± 0.09	6.58 ± 0.07	7.59 ± 0.16	17.04 ± 0.29	35.84 ± 0.43	2.03 ± 0.04
Fourclass	6.90 ± 0.09	8.70 ± 0.09	12.23 ± 0.13	6.37 ± 0.21	14.33 ± 0.44	30.30 ± 0.97	1.83 ± 0.03
German.numer	10.83 ± 0.23	14.34 ± 0.20	18.35 ± 0.17	22.66 ± 0.30	50.88 ± 0.88	107.24 ± 1.67	6.23 ± 0.12
Liver-disorders	0.92 ± 0.08	1.38 ± 0.13	2.12 ± 0.07	2.37 ± 0.05	5.22 ± 0.13	10.51 ± 0.19	0.62 ± 0.01
Sonar	1.00 ± 0.05	1.32 ± 0.04	1.82 ± 0.02	3.01 ± 0.06	6.47 ± 0.12	13.28 ± 0.12	0.95 ± 0.02
a2a	64.5 ± 1.09	82.2 ± 1.28	103.50 ± 1.30	129.8 ± 0.93	292.9 ± 2.56	618.1 ± 5.10	42.03 ± 0.96
Polynomial Kernel							
Data sets	0.2-BIF	0.1-BIF	0.05-BIF	5-CV	10-CV	20-CV	GACV
Australian	2.20 ± 0.06	2.76 ± 0.12	3.91 ± 0.12	4.80 ± 0.20	11.18 ± 0.31	23.19 ± 0.44	1.39 ± 0.09
Heart	0.38 ± 0.00	0.52 ± 0.01	0.78 ± 0.01	0.85 ± 0.00	1.99 ± 0.02	4.09 ± 0.01	0.21 ± 0.00
Ionosphere	0.58 ± 0.01	0.76 ± 0.01	1.11 ± 0.01	1.32 ± 0.01	3.03 ± 0.03	6.57 ± 0.11	0.34 ± 0.02
Breast-cancer	2.16 ± 0.04	2.65 ± 0.01	3.68 ± 0.03	4.54 ± 0.02	10.62 ± 0.08	22.43 ± 0.45	1.31 ± 0.03
Diabetes	3.36 ± 0.17	4.38 ± 0.20	6.30 ± 0.03	5.65 ± 0.11	13.71 ± 0.27	32.46 ± 0.39	1.77 ± 0.02
Fourclass	4.23 ± 0.02	5.56 ± 0.02	8.22 ± 0.09	7.32 ± 0.03	18.11 ± 0.12	39.34 ± 0.76	2.30 ± 0.02
German.numer	5.92 ± 0.01	7.77 ± 0.04	11.79 ± 0.58	10.53 ± 0.29	24.94 ± 0.07	55.59 ± 0.25	3.13 ± 0.02
Liver-disorders	0.54 ± 0.00	0.72 ± 0.01	1.06 ± 0.01	1.24 ± 0.01	2.90 ± 0.01	6.05 ± 0.06	0.31 ± 0.01
Sonar	0.26 ± 0.00	0.37 ± 0.01	0.57 ± 0.00	0.65 ± 0.00	1.36 ± 0.01	2.81 ± 0.01	0.13 ± 0.00
a2a	38.32 ± 0.76	48.23 ± 0.56	70.26 ± 0.79	61.31 ± 0.54	148.37 ± 1.94	306.15 ± 7.97	21.99 ± 0.17

5.2. Time cost

The computational time of ϵ-BIF, t-CV and GACV are listed in Table 4. The results in Table 4 can be summarized as follows: (1) The time cost of our BIF much lower than that of CV. Thus, the proposed approximate CV with granularity selection can significantly improve the efficiency of t-CV for model selection. (2) The GACV is faster than BIF. This can be explained by the fact that although BIF and GACV only need to training the algorithm once, BIF still needs to compute the BIF matrices.

6. Conclusion

In this paper, we present a novel granularity selection method for CV of SVM. This is the first attempt to select the number of folds and approximate the t-fold CV for the non-differentiable loss based regularization algorithms. We propose a strategy to approximate the CV of SVM with smooth Huber loss, and derive an upper bound of the discrepancy between the original and the approximate CV errors with a high convergence rate. Furthermore, we give a granularity selection criterion that can cut the time cost and sustain the accuracy requirement. Theoretical and experimental results show that our proposed approximate CV with the granularity selection criterion has sound theoretical foundation and high computational efficiency.

Acknowledgement

This work was supported in part by 973 Program (2013CB329304), Key Program of National Natural Science Foundation of China (No.61432011) and National Natural Foundation of China (No.61222210).

Appendix

In this section, we will give the proofs of Theorem 1 and Theorem 2.

Appendix A: Proof of Theorem 1

Proof. Notice that $\max(0, 1-t) \le \ell_\delta(t) \le \max(0, 1-t) + \delta/2, \forall t$, which yields the following inequalities

$$R(f) \le R^\delta(f) \le R(f) + \frac{\delta}{2}, \forall f \in \mathcal{H}_K. \tag{7}$$

Thus, one can obtain that

$$\begin{aligned} R(f_{P_S}^{\text{svm}}) &= \inf_{f\in\mathcal{H}_K} R(f) \le R(f_{P_S}^{\delta}) \\ &\le R^{\delta}(f_{P_S}^{\delta}) \le R^{\delta}(f_{P_S}^{\text{svm}}) \\ &\le R(f_{P_S}^{\text{svm}}) + \delta/2, \end{aligned}$$

which completes the proof of Theorem 1. □

Appendix B: Proof of Theorem 2

Denote $\tilde{f}^{\delta}_{P_{S\setminus S_i}}(\mathbf{x}_j) = f^{\delta}_{P_S} + \sum_{s=1}^{r}\left(\frac{-1}{t-1}\right)^s \frac{[\mathbf{B}_s]_j}{s!}$. Note that

$$\begin{aligned} &\left|I\left(y_j f^{\text{svm}}_{P_{S\setminus S_i}}(\mathbf{x}_j)\right) - I\left(y_j \tilde{f}^{\delta}_{P_{S\setminus S_i}}(\mathbf{x}_j)\right)\right| \\ &\le \left|I\left(y_j f^{\text{svm}}_{P_{S\setminus S_i}}(\mathbf{x}_j)\right) - I\left(y_j f^{\delta}_{P_{S\setminus S_i}}(\mathbf{x}_j)\right)\right| \\ &+ \left|I\left(y_j f^{\delta}_{P_{S\setminus S_i}}(\mathbf{x}_j)\right) - I\left(y_j \tilde{f}^{\delta}_{P_{S\setminus S_i}}(\mathbf{x}_j)\right)\right|. \end{aligned}$$

From Theorem 1, we know that

$$\left|I\left(y_j f^{\text{svm}}_{P_{S\setminus S_i}}(\mathbf{x}_j)\right) - I\left(y_j f^{\delta}_{P_{S\setminus S_i}}(\mathbf{x}_j)\right)\right| \le \frac{\delta}{2}.$$

In the following, we will bound

$$\left|I\left(y_j f^{\delta}_{P_{S\setminus S_i}}(\mathbf{x}_j)\right) - I\left(y_j \tilde{f}^{\delta}_{P_{S\setminus S_i}}(\mathbf{x}_j)\right)\right|.$$

Since $I(\cdot)$ is 1-Lipschitz continuous, we have

$$\begin{aligned} &\left|I\left(y_j f^{\delta}_{P_{S\setminus S_i}}(\mathbf{x}_j)\right) - I\left(y_j \tilde{f}^{\delta}_{P_{S\setminus S_i}}(\mathbf{x}_j)\right)\right| \\ &\le |y_j f^{\delta}_{P_{S\setminus S_i}}(\mathbf{x}_j) - y_j \tilde{f}^{\delta}_{P_{S\setminus S_i}}(\mathbf{x}_j)| \\ &= |f^{\delta}_{P_{S\setminus S_i}}(\mathbf{x}_j) - \tilde{f}^{\delta}_{P_{S\setminus S_i}}(\mathbf{x}_j)|. \end{aligned}$$

Using the reproducing property $f(\mathbf{x}_i) = \langle f, \Phi(\mathbf{x}_i)\rangle_K$, we can differentiate (2) with respect to f and at the optimal solution $f^{\delta}_{P_S}$, the gradient vanishes, yielding

$$-2\lambda f^{\delta}_{P_S} = \frac{1}{n}\sum_{i=1}^{n}\left[y_i \ell'_{\delta}(y_i f(\mathbf{x}_i))K(\mathbf{x}_i, \cdot)\right].$$

Note that $|\ell'_{\delta}| \le 1$, and $\sup_{\mathbf{x}\in\mathcal{X}} K(\mathbf{x},\mathbf{x}) \le 1$, so the following inequation holds:

$$|f^{\delta}_{P_S}(\mathbf{x})| \le \frac{1}{2\lambda}, \forall \mathbf{x} \in \mathcal{X}. \tag{8}$$

By the definition of $\mathbf{B}_1^i$ and $\mathbf{L}_n$, it is easy to verity that

$$|[\mathbf{B}_1^i]_j| \le \|\mathbf{L}_n^{-1}\|_2 \left(\frac{1}{l}\sum_{\mathbf{x}_h\in S_i}|K(\mathbf{x}_j,\mathbf{x}_h)| + 2\lambda|f^{\delta}_{P_S}(\mathbf{x}_j)|\right) \le \frac{1}{\lambda}.$$

From the definition of $\mathbf{B}_{k+1}^i$, one sees similarly that the upper bound of high order terms can be obtained by, $\forall \mathbf{x}_j \in S_i$,

$$\begin{aligned} |[\mathbf{B}_{k+1}^i]_j| &\le 2n\delta(k+1)\frac{t-1}{2\delta n}|\text{BIF}_k(P_{S_i}; f^{\delta}, P_S)(\mathbf{x}_j)| \\ &= (k+1)(t-1)|[\mathbf{B}_k^i]_j|, \end{aligned}$$

Therefore, from Taylor's Theorem, it is easy to verity that

$$\begin{aligned}|f^{\delta}_{P_{S\setminus S_i}}(\mathbf{x}_j) - \tilde{f}^{\delta}_{P_{S\setminus S_i}}(\mathbf{x}_j)| &\le \frac{|[\mathbf{B}^i_{r+1}]_j|}{(t-1)^{r+1}(r+1)!}\\ &\le \frac{|[\mathbf{B}^i_1]_j|}{(r+1)(t-1)}\\ &= \frac{1}{\lambda(r+1)(t-1)}.\end{aligned}$$

Thus, we have

$$\left|I\left(y_j f^{\mathrm{svm}}_{P_{S\setminus S_i}}(\mathbf{x}_j)\right) - I\left(y_j \tilde{f}^{\delta}_{P_{S\setminus S_i}}(\mathbf{x}_j)\right)\right| \le \frac{\delta}{2} + \frac{1}{\lambda(r+1)(t-1)}.$$

This completes the proof of Theorem 2.

References

[1] E. Abbasnejad, D. Ramachandram, R. Mandava, A survey of the state of the art in learning the kernels, Knowl. Inf. Syst. 31 (2) (2012) 193–221.
[2] S. An, W. Liu, S. Venkatesh, Fast cross-validation algorithms for least squares support vector machine and kernel ridge regression, Pattern Recognit. 40 (8) (2007) 2154–2162.
[3] N. Aronszajn, Theory of reproducing kernels, Trans. Am. Math. Soc. 68 (1950) 337–404.
[4] G. Cawley, N. Talbot, Preventing over-fitting during model selection via Bayesian regularisation of the hyper-parameters, J. Mach. Learn. Res. 8 (2007) 841–861.
[5] O. Chapelle, V. Vapnik, O. Bousquet, S. Mukherjee, Choosing multiple parameters for support vector machines, Mach. Learn. 46 (1-3) (2002) 131–159.
[6] A. Christmann, A.V. Messem, Bouligand derivatives and robustness of support vector machines for regression, J. Mach. Learn. Res. 9 (2008) 915–936.
[7] A. Christmann, I. Steinwart, Support Vector Machines, Springer Verlag, 2008.
[8] B. Huang, C. Guo, H. Li, G. Feng, X. Zhou, Hierarchical structures and uncertainty measures for intuitionistic fuzzy approximation space, Inf. Sci. 336 (2016) 92–114.
[9] J. Josse, F. Husson, Selecting the number of components in principal component analysis using cross-validation approximations, Comput. Stat. Data Anal. 56 (6) (2012) 1869–1879.
[10] S. Keerthi, V. Sindhwani, O. Chapelle, An efficient method for gradient-based adaptation of hyperparameters in svm models, in: Advances in Neural Information Processing Systems 19 (NIPS 2006), 2006, pp. 673–680.
[11] R. Kohavi, G. John, Automatic parameter selection by minimizing estimated error, in: Proceedings of the 12nd International Conference on Machine Learning (ICML 1995), 1995, pp. 304–312.
[12] T. Krueger, D. Panknin, M. Braun, Fast cross-validation via sequential testing, The Journal of Machine Learning Research 16 (2015) 1103–1155.
[13] J. Liang, Z. Shi, The information entropy, rough entropy and knowledge granulation in rough set theory, Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 12 (01) (2004) 37–46.
[14] Y. Liu, S. Jiang, S. Liao, Eigenvalues perturbation of integral operator for kernel selection, in: Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (CIKM 2013), 2013, pp. 2189–2198.
[15] Y. Liu, S. Jiang, S. Liao, Efficient approximation of cross-validation for kernel methods using Bouligand influence function, in: Proceedings of The 31st International Conference on Machine Learning (ICML(1) 2014), 2014, pp. 324–332.
[16] Y. Liu, S. Liao, Kernel selection with spectral perturbation stability of kernel matrix, Sci. Chin. Inf. Sci. 57 (11) (2014) 1–10.
[17] Y. Liu, S. Liao, Preventing over-fitting of cross-validation with kernel stability, in: Proceedings of the 7th European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2014), Springer, 2014, pp. 290–305.
[18] Y. Liu, S. Liao, Y. Hou, Learning kernels with upper bounds of leave-one-out error, in: Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM 2011), 2011, pp. 2205–2208.
[19] W. Mao, X. Mu, Y. Zheng, G. Yan, Leave-one-out cross-validation-based model selection for multi-input multi-output support vector machine, Neural Comput. Appl. 24 (2) (2014) 441–451.
[20] C. Mosier, The need and means of cross validation, Edu. Psychol. Measur. 11 (1951) 5–11.
[21] W. Pedrycz, Granular Computing: Analysis and Design of Intelligent Systems, CRC press, 2013.
[22] B. Schölkopf, A. Smola, Learning with Kernels, MIT Press, Cambridge, MA, 2002.
[23] A. Skowron, J. Stepaniuk, Information granules: towards foundations of granular computing, Int. J. Intell. Syst. 16 (1) (2001) 57–85.
[24] M. Stone, Cross-validatory choice and assessment of statistical predictions, J. Roy. Stat. Soc.. Series B (Methodological) 36 (1974) 111–147.
[25] J. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Process. Lett. 9 (3) (1999) 293–300.
[26] V. Vapnik, The Nature of Statistical Learning Theory, Springer Verlag, 2000.
[27] G. Wahba, Y. Lin, H. Zhang, Generalized approximate cross-validation for support vector machines for support vector machines, in: Advances in Large Margin Classifiers, 2000, pp. 297–309.
[28] Y. Yao, Granular computing: basic issues and possible solutions, in: Proceedings of the 5th Joint Conference on Information Sciences (JCIS 2000), vol. 1, Citeseer, 2000, pp. 186–189.
[29] L. Zadeh, Fuzzy sets, Inf. Control 8 (3) (1965) 338–353.
[30] L. Zadeh, Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, Fuzzy Sets Syst. 90 (2) (1997) 111–127.
[31] J. Zhang, Y. Wang, A rough margin based support vector machine, Inf. Sci. 178 (9) (2008) 2204–2214.
[32] P. Zhu, Q. Hu, Adaptive neighborhood granularity selection and combination based on margin distribution optimization, Inf. Sci. 249 (2013) 1–12.
[33] P. Zhu, Q. Hu, W. Zuo, M. Yang, Multi-granularity distance metric learning via neighborhood granule margin maximization, Inf. Sci. 282 (2014) 321–331.