Fast Cross-Validation for Kernel-Based Algorithms

Yong Liu, Shizhong Liao, Shali Jiang, Lizhong Ding, Hailun Lin, and Weiping Wang

Abstract—Cross-validation (CV) is a widely adopted approach for selecting the optimal model. However, the computation of empirical cross-validation error (CVE) has high complexity due to multiple times of learner training. In this paper, we develop a novel approximation theory of CVE and present an approximate approach to CV based on the Bouligand influence function (BIF) for kernel-based algorithms. We first represent the BIF and higher order BIFs in Taylor expansions, and approximate CV via the Taylor expansions. We then derive an upper bound of the discrepancy between the original and approximate CV. Furthermore, we provide a novel computing method to calculate the BIF for general distribution, and evaluate BIF criterion for sample distribution to approximate CV. The proposed approximate CV requires training on the full data set only once and is suitable for a wide variety of kernel-based algorithms. Experimental results demonstrate that the proposed approximate CV is sound and effective.

Index Terms—Cross-validation, approximation, bouligand influence function, model selection, kernel methods

1 Introduction

Cross-validation (CV) is a tried and tested approach for selecting the optimal model [1], [2]. The empirical cross-validation error (CVE) is a reliable estimate of the generalization error for performance estimation [3]. In t-fold CV, 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 the subsets from training, but using the omitted subset to compute the validation error. The t-fold CV estimate is then simply the average of the validation errors observed in each of the t iterations, or folds. Discussions and theoretical studies about the t-fold CV can be found, for example, in [4], [5], [6], [7], [8], [9] and the references therein.

Kernel-based algorithms, such as SVM [10], least squares SVM [11], kernel ridge regression (KRR) [12] and support vector regression (SVR) [13], have demonstrated great success in pattern recognition problems. The performance of these methods 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 fundamental to kernel-based algorithms [14], [15], [16].

Although the t-fold CV is a commonly used approach for selecting the hyper-parameters of kernel-based algorithms [17], [18], [19], it requires training t times. In this paper, we develop a novel approximation theory of CVE and present an approach to approximating the CVE based on Bouligand influence function (BIF) [20] for a variety of kernel-based algorithms, including LSSVM, KRR, SVM and SVR. We first establish the relationship between BIF and CVE, and further give an approach for approximating the CVE via BIF and higher order BIFs. This is the first attempt to apply the theoretical notion of BIF for practical model selection. We then derive an upper bound of the discrepancy between the original and approximate CVEs, with a convergence rate $O\left(\frac{1}{r!t^r}\right)$, where r is the order of the Taylor expansion and t the number of the folds. Thus, if r and t are not very small, the value of $\frac{1}{r!t^r}$ is nearly equal to 0, which demonstrates the effectiveness of the proposed approximate CV. Moreover, we provide a novel computing method to calculate the BIF and higher order BIFs for general distribution, and use them as the criteria to approximate the CVE for model selection of kernel-based algorithms. The proposed approximate CV requires training on the full data only once, hence can significantly improve the efficiency. Experimental results demonstrate that the approximate CV gives almost the same accuracy results as the traditional CV, but meanwhile significantly improves the efficiency.

This paper is an extension of [21], [22] published in ICML and IJCAI, respectively. Compared with the conference versions, this paper contains much new material, including

- an approximation theory of CVE based on the Bouligand influence function for kernel-based algorithms;
- an upper bound of the discrepancy between the original and the approximate CVEs;

- *Y. Liu and H. Lin are with Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100864, China. E-mail: {liuyong, linhailun}@iie.ac.cn.*
- *S. Liao is with College of Intelligence and Computing, Tianjin University, Tianjin 300457, China. E-mail: szliao@tju.edu.cn.*
- *S. Jiang is with Washington University, St. Louis, MO 63130. E-mail: shalijiang@gmail.com.*
- *L. Ding is with Inception Institute of Artificial Intelligence (IIAI), Abu Dhabi, UAE. E-mail: lizhong.ding@inceptioniai.org.*
- *W. Wang is with Institute of Information Engineering, Chinese Academy of Sciences, National Engineering Research Center for Information Security, and National Engineering Laboratory for Information Security Technology, Beijing 100864, China. E-mail: wangweiping@iie.ac.cn.*

Manuscript received 14 July 2018; revised 31 Oct. 2018; accepted 1 Jan. 2019. Date of publication 14 Jan. 2019; date of current version 1 Apr. 2020. (Corresponding author: Yong Liu.) Recommended for acceptance by K. Weinberger. Digital Object Identifier no. 10.1109/TPAMI.2019.2892371

- an extension of our approximate method to various kernel-based algorithms, such as L1-SVM, L2-SVM and SVR;
- lots of experimental investigations of our approximate method for various kernel-based algorithms.

1.1 Related Work

In this section, we will introduce the related work about approximate CV methods of kernel-based algorithms and Bouligand influence function.

1.1.1 Approximate CV of Kernel-Based Algorithms

The extreme form of t-fold CV, where t equals the sample size, is known as leave-one-out CV. For the sake of efficiency, much work has been done to reduce the time complexity of leave-one-out CV for some specific kernel-based learning algorithms, see [23], [24], [25], [26], [27], [28] for SVM, [17], [29] for LSSVM, [30] for sparse LSSVM, [31] for kernel logistic regression, [32] for kernel Fisher discriminant classifiers, [33], [34] for Gaussian processes, and [35], [36] for kernel-based regression. There is much work on improving the efficiency of the leave-one-out CV, but little work focuses on the general t-fold CV. Based on the fact that the LSSVM and KRR have closed-form solutions, [18], [37] applied the matrix inverse formula to develop a new formula to expedite the t-fold CV process. [8], [38] proposed a framework for kernel multi-class models, and used the approximation of t-fold CVE log likelihood to learn the kernel parameters.

There is little work on the approximation of the general t-fold CV (for all t). Moreover, the existing approximation techniques for the leave-one-out CV or general t-fold CV can only be used for a specific kernel-based algorithm. There exist no effective methods for approximating the general t-fold CVE for a wide variety of kernel-based algorithms.

1.1.2 Bouligand Influence Function

In recent years, some researchers have studied the robustness of kernel-based algorithms, see [20], [35], [39], [40], [41], [42], [43], [44], [45], [46], [47] and the references therein. In the field of robust statistics, the notion of influence function (IF) [48] is introduced in order to analyze the effects of outliers on the algorithm. [39], [40], [41], [42], [44], [45], [46] showed that SVMs for classification and regression have a bounded influence function under some assumption on the loss function. [35] provided an approach to estimating the leave-one-out CVE via the influence function for some kernel-based regression. [20] generalized the notion of influence function, and introduced a new notion from Bouligand-derivatives [49] called Bouligand influence function, which measures the impact of an infinitesimal small amount of contamination of the original distribution. They also showed that SVMs have a bounded BIF with some assumptions on loss function.

The focus of the above work lies on deriving the theoretical analysis of robust statistics for some kernel-based algorithms. But little work aims at developing practical procedures for model assessment.

1.2 Contributions

Our contributions are given as follows:

1) We develop a novel approximation theory of CVE based on the BIF for kernel-based algorithms. A tight upper bound of the difference between the original and the approximate t-fold CVEs is established (see Theorem 1).
2) We show that the BIF and higher order BIFs are closely related to terms of a Taylor expansion, and present an approach to approximating the CVE via the Taylor expansion. This is the first attempt to apply the theoretical notion of BIF for practical model selection.
3) We provide a novel computing method to calculate the first order BIF for the general distribution. Furthermore, we extend the above result of the first order BIF to the higher order BIFs (see Theorem 3).
4) We evaluate the first and higher order BIFs for sample distribution, and further present an efficient method for approximating t-fold CVE. Different from the existing approximate leave-one-out CV or t-fold CV, the proposed approximate CV is suitable for a wide variety of kernel-based algorithms, such as SVM, LSSVM, KRR and SVR.
5) The proposed approximate t-fold CV requires training on the full data set only once, hence can significantly improve the efficiency. The proposed approximate CV offers a new paradigm for practical model selection.

1.3 Outlines

The rest of the paper is organized as follows. We start by introducing some preliminaries and notations in Section 2. In Section 3, we give the notion of BIF and higher order BIFs, and propose a novel strategy for approximating the CVE. A method to calculate the BIF and higher order BIFs for the general continuous distribution is proposed in Section 4. In Section 5, we evaluate these BIFs for the sample distribution, and show how to use BIFs to approximate the t-fold CV in practice. We empirically analyze the performance of our proposed approximate CV in Section 6. We end in Section 7 with conclusion.

2 Preliminaries and Notations

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

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

where $\mathcal{X}$ denotes the input space and $\mathcal{Y}$ denotes the output space, $\mathcal{Y} \subset \mathbb{R}$ in the regression case and $\mathcal{Y} = \{-1, +1\}$ in classification case. We assume $\mathcal{S}$ is drawn identically and independently from a fixed, but unknown probability distribution $\mathbb{P}$ on $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$.

Let $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel, that is, κ is symmetric and for any finite set of points $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathcal{X}$, the kernel matrix $\mathbf{K} = \left[\kappa(\mathbf{x}_i, \mathbf{x}_j)\right]_{i,j=1}^n$ is positive semidefinite. The reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ associated with the kernel κ is defined to be the completion of the linear span of the set of functions

$$\{\Phi(\mathbf{x}) = \kappa(\mathbf{x}, \cdot) : \mathbf{x} \in \mathcal{X}\}$$

with the inner product satisfying $\langle \kappa(\mathbf{x}_i, \cdot), \kappa(\mathbf{x}'_i, \cdot) \rangle_\kappa = \kappa(\mathbf{x}_i, \mathbf{x}'_j)$.

The operator $f_\kappa : \mathbb{P} \to f_\kappa(\mathbb{P}) = f_{\kappa,\mathbb{P}} \in \mathcal{H}$ is defined by

$$f_{\kappa,\mathbb{P}} = \arg\min_{f \in \mathcal{H}} \mathbb{E}_{\mathbb{P}}[\ell(y, f(\mathbf{x}))] + \lambda \|f\|_\kappa^2,$$

where $\ell(\cdot,\cdot)$ is a loss function, λ is the regularization parameter and $\mathcal{H}$ is the RKHS associated with κ. When the sample distribution $\mathbb{P}_\mathcal{S}$ is used,

$$f_{\kappa,\mathbb{P}_\mathcal{S}} = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(\mathbf{x}_i)) + \lambda \|f\|_\kappa^2.$$

KRR, LSSVM, L1-SVM, L2-SVM and ϵ-SVR are only different in the choice of the loss function:

- KRR and LSSVM: $\ell(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2$;
- L1-SVM: $\ell(y, f(\mathbf{x})) = \max(0, 1 - y(f(\mathbf{x})))$;
- L2-SVM: $\ell(y, f(\mathbf{x})) = \max(0, 1 - y(f(\mathbf{x})))^2$;
- ϵ-SVR: $\ell(y, f(\mathbf{x})) = \max(0, |y - f(\mathbf{x})| - \epsilon)$.

Let $\mathcal{S}_1, \ldots, \mathcal{S}_t$ be a random equipartition of $\mathcal{S}$ into t parts, called folds, with $|\mathcal{S}_i| = \lfloor \frac{n}{t} \rfloor =: l$. Let $\mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}$ be the empirical distribution of the sample $\mathcal{S}$ without the observations $\mathcal{S}_i$, that is

$$\mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}(z) = \begin{cases} \frac{1}{n-l} & \text{if } z \in \mathcal{S} \setminus \mathcal{S}_i, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Let $f_{\kappa,\mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}}$ be the hypothesis learned on all of the data excluding $\mathcal{S}_i$. The t-fold cross-validation error is defined as

$$t - \mathrm{CV} := \frac{1}{n} \sum_{i=1}^{t} \sum_{z_j \in \mathcal{S}_i} V\Big(y_j, f_{\kappa,\mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}}(\mathbf{x}_j)\Big),$$

where $V(\cdot,\cdot)$ is an appropriate loss function.[1]

3 Approximation of Cross-Validation

In this section, we will first introduce the notion of Bouligand influence function [20] and define higher order BIFs. We then represent the BIF and higher order BIFs as terms of Taylor expansions, and approximate t-fold CVE via the Taylor expansions. Finally, we derive an upper bound of the discrepancy between the original and approximate CVEs.

3.1 Bouligand Influence Function

Definition 1. *Let $\mathbb{P}$ be a distribution and T be an operator $T : \mathbb{P} \to T(\mathbb{P})$. Then the* Bouligand influence function (BIF) *of T at $\mathbb{P}$ in the direction of a distribution $\mathbb{Q} \neq \mathbb{P}$ is defined as*

$$\mathrm{BIF}(\mathbb{Q}; T, \mathbb{P}) = \lim_{\epsilon \to 0} \frac{T((1-\epsilon)\mathbb{P} + \epsilon\mathbb{Q}) - T(\mathbb{P})}{\epsilon}.$$

One can see that BIF is used to measure the impact of an infinitesimal small amount of contamination of the original distribution $\mathbb{P}$ in the direction of $\mathbb{Q}$ on the quantity of $T(\mathbb{P})$. The notion of influence function [48] popularly used in the field of robust statistics is a special case of BIF when setting $\mathbb{Q}$ to be the Dirac distribution δ_z at a point $z \in \mathcal{Z}$. BIF is a generalization of influence function.

Remark 1. The existence of BIF has been discussed in [20]. They show that if the $\mathcal{X}$ and $\mathcal{Y}$ are closed sets, κ is a bounded and continuous kernel function, ℓ is a convex loss function and is Lipschitz continuous w.r.t. the second argument, then the BIF is existence. In this paper, we assume the BIF is existence in the following.

Denote $\mathbb{P}_{\epsilon,\mathbb{Q}} = (1-\epsilon)\mathbb{P} + \epsilon\mathbb{Q}$. Note that the derivative of $T(\mathbb{P}_{\epsilon,\mathbb{Q}})$ at ϵ can be written as

$$\lim_{\Delta\epsilon \to 0} \frac{T(\mathbb{P}_{\epsilon+\Delta\epsilon,\mathbb{Q}}) - T(\mathbb{P}_{\epsilon,\mathbb{Q}})}{\Delta\epsilon}. \quad (2)$$

If setting $\epsilon = 0$, Equation (2) gives

$$\lim_{\Delta\epsilon \to 0} \frac{T(\mathbb{P}_{\Delta\epsilon,\mathbb{Q}}) - T(\mathbb{P})}{\Delta\epsilon} = \mathrm{BIF}(\mathbb{Q}; T, \mathbb{P}).$$

Thus, $\mathrm{BIF}(\mathbb{Q}; T, \mathbb{P})$ is the first order derivative of $T(\mathbb{P}_{\epsilon,\mathbb{Q}})$ at $\epsilon = 0$. The higher order BIFs can be defined as follows:

Definition 2. *Let $\mathbb{P}$ be a distribution and T be an operator $T : P \to T(\mathbb{P})$. Then the kth order BIF of T at $\mathbb{P}$ in the direction of a distribution $\mathbb{Q} \neq \mathbb{P}$ is defined as*

$$\mathrm{BIF}_k(\mathbb{Q}; T, \mathbb{P}) = \frac{\partial}{\partial^k \epsilon} T(\mathbb{P}_{\epsilon,\mathbb{Q}})|_{\epsilon=0},$$

where $\mathbb{P}_{\epsilon,\mathbb{Q}} = (1-\epsilon)\mathbb{P} + \epsilon\mathbb{Q}$.

If BIF and high BIFs existence, from Taylor expansion, we have

$$T(\mathbb{P}_{\epsilon,\mathbb{Q}}) = T(\mathbb{P}) + \sum_{i=1}^{\infty} \frac{\epsilon^i}{i!} \mathrm{BIF}_i(\mathbb{Q}; T, \mathbb{P}). \quad (3)$$

3.2 Approximation of Cross-Validation Error with BIFs

In this section, we will give a novel approach to approximating the t-fold CVE with BIFs, and derive a bound of the discrepancy between the original and approximate CVEs.

To this end, let $\mathbb{P}_{\mathcal{S}_i}$ be the empirical distribution corresponding to the ith fold $\mathcal{S}_i$ and $\mathbb{P}_\mathcal{S}$ the sample distribution, respectively. We have

$$\mathbb{P}_{\mathcal{S}_i}(z) = \begin{cases} \frac{1}{l} & \text{if } z \in \mathcal{S}_i, \\ 0 & \text{otherwise;} \end{cases} \text{ and } \mathbb{P}_\mathcal{S}(z) = \begin{cases} \frac{1}{n} & \text{if } z \in S, \\ 0 & \text{otherwise.} \end{cases}$$

One can see that

$$\mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i} = \left(1 - \left(\frac{-1}{t-1}\right)\right)\mathbb{P}_\mathcal{S} + \frac{-1}{t-1}\mathbb{P}_{\mathcal{S}_i},$$

where $\mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}$ is the empirical distribution of the sample $\mathcal{S}$ without $\mathcal{S}_i$ defined in (1). Thus, if taking

$$\mathbb{Q} = \mathbb{P}_{\mathcal{S}_i}, \epsilon = \frac{-1}{t-1}, \mathbb{P}_{\epsilon,\mathbb{Q}} = \mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}, \mathbb{P} = \mathbb{P}_\mathcal{S} \text{ and } T = f_\kappa,$$

Equation (3) gives

$$f_{\kappa,\mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}} = f_{\kappa,\mathbb{P}_\mathcal{S}} + \sum_{s=1}^{\infty} \left(\frac{-1}{t-1}\right)^s \frac{\mathrm{BIF}_s(\mathbb{P}_{\mathcal{S}_i}; f_\kappa, \mathbb{P}_\mathcal{S})}{s!}. \quad (4)$$

1. V does not need to be the same as the loss function ℓ used to obtain $f_{\kappa,\mathbb{P}_\mathcal{S}}$.

Thus, we can take the low order approximation of the Taylor expansion to effectively approximate the t-fold CV by cutting it off at some step r

$$\mathrm{BIF}_t^r := \frac{1}{n}\sum_{i=1}^{t}\sum_{z_j\in\mathcal{S}_i} V\Bigg(y_j, f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_j) + \sum_{s=1}^{r}\left[\frac{-1}{t-1}\right]^s \frac{\mathrm{BIF}_s(\mathbb{P}_{\mathcal{S}_i}; f_\kappa, \mathbb{P}_{\mathcal{S}})(\mathbf{x}_j)}{s!}\Bigg). \tag{5}$$

Note that BIF_t^r only depends on the calculation of $f_{\kappa,\mathbb{P}_{\mathcal{S}}}$ and $\mathrm{BIF}_s(\mathbb{P}_{\mathcal{S}_i}; f_\kappa, \mathbb{P}_{\mathcal{S}})$. Thus, if given the $\mathrm{BIF}_s(\mathbb{P}_{\mathcal{S}_i}; f_\kappa, \mathbb{P}_{\mathcal{S}})$, we need to train the algorithm only once on the full data set $\mathcal{S}$ to obtain $f_{\kappa,\mathbb{P}_{\mathcal{S}}}$ for approximating the $f_{\kappa,\mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}}$, $i = 1, 2, \ldots$. In the next section, we will provide a procedure to calculate $\mathrm{BIF}_s(\mathbb{P}_{\mathcal{S}_i}; f_\kappa, \mathbb{P}_{\mathcal{S}})$ for kernel-based algorithms.

The upper bound of the difference between the original t-fold CVE and the approximate BIF_t^r is given as follows:

Theorem 1. *Let t-CV be the t-fold CVE,*

$$t-\mathrm{CV} := \frac{1}{n}\sum_{i=1}^{t}\sum_{z_j\in\mathcal{S}_i} V(y_j, f_{\kappa,\mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}}(\mathbf{x}_j)),$$

and BIF_t^r the approximate t-fold CVE defined in (5). Assume that $\mathrm{BIF}_s(\mathbb{P}_{\mathcal{S}_i}; f_\kappa, \mathbb{P}_{\mathcal{S}})$ is bounded, that is, there exists a constant Q, such that for all $\mathbf{x}\in\mathcal{X}$, $|\mathrm{BIF}_s(\mathbb{P}_{\mathcal{S}_i}; f_\kappa, \mathbb{P}_{\mathcal{S}})(\mathbf{x})| \le Q$, and $V(\cdot,\cdot)$ is C-Lipschitz continuous with respect to the second variable. Then

$$\left|t-\mathrm{CV}-\mathrm{BIF}_t^r\right| \le \frac{CQ}{(r+1)!(t-1)^r(t-2)}.$$

The proof is given in Appendix A.

The assumption that $\mathrm{BIF}_s(\mathbb{P}_{\mathcal{S}_i}; f_\kappa, \mathbb{P}_{\mathcal{S}})$ is bounded is common, we will show that this assumption is satisfied under certain general conditions in Section 5. One can see that for a not very small number r, $\frac{1}{(r+1)!(t-1)^r(t-2)}$ is nearly equal to 0. For example, if setting $t = 10$, $r = 5$, then $\frac{1}{(r+1)!(t-1)^r(t-2)} = \frac{1}{6!\cdot 9^5\cdot 3} = 7.84\text{e-}9$. This implies the effectiveness of the proposed approximate CV.

4 Calculation of BIFs for General Distribution

In this section, we will provide a novel method to calculate the first order BIF and higher order BIFs at a general distribution $\mathbb{P}$ on $\mathcal{Z}$.

Let $\ell'(\cdot,\cdot)$ be the derivative of $\ell(\cdot,\cdot)$ with respect to the second variable. The first order BIF at the general distribution $\mathbb{P}$ is given in the following theorem:

Theorem 2. *Let $\mathcal{H}$ be the RKHS of a bounded continuous kernel κ on $\mathcal{X}$, and $\mathbb{P}$ a distribution on $\mathcal{Z}$, then the* BIF *of f_κ in the direction of a distribution $\mathbb{Q}\neq\mathbb{P}$ is*

$$\mathrm{BIF} = L^{-1}\Big[-2\lambda f_{\kappa,\mathbb{P}} - \mathbb{E}_{\mathbb{Q}}\big[\ell'(y, f_{\kappa,\mathbb{P}}(\mathbf{x}))\Phi(\mathbf{x})\big]\Big],$$

where the operator $L:\mathcal{H}\to\mathcal{H}$ is defined by

$$L(f) = 2\lambda f + \mathbb{E}_{\mathbb{P}}[\ell''(y, f(\mathbf{x}))f(\mathbf{x})\Phi(\mathbf{x})].$$

The proof is given in Appendix B.

The calculation of higher order BIFs are given in the following theorem:

Theorem 3. *Let $\mathcal{H}$ be the RKHS of a bounded continuous kernel κ on $\mathcal{X}$, and $\ell(\cdot,\cdot)$ a convex loss function such that the third derivative with respect to the second variable is 0. Furthermore, let $\mathbb{P}$ be a distribution on $\mathcal{Z}$, then the $(k+1)$th order BIF_{k+1} of f_κ in the direction of a distribution $\mathbb{Q}\neq\mathbb{P}$ is*

$$\mathrm{BIF}_{k+1} = (k+1)L^{-1}\Big[\mathbb{E}_{\mathbb{P}}\big[\mathrm{BIF}_k(\mathbb{Q}; f_\kappa, \mathbb{P})\ell''(y, f_{\kappa,\mathbb{P}}(\mathbf{x}))\Phi(\mathbf{x})\big] - \mathbb{E}_{\mathbb{Q}}\big[\mathrm{BIF}_k(\mathbb{Q}; f_\kappa, \mathbb{P})\ell''(y, f_{\kappa,\mathbb{P}}(\mathbf{x}))\Phi(\mathbf{x})\big]\Big].$$

where the operator $L:\mathcal{H}\to\mathcal{H}$ is defined by

$$L(f) = 2\lambda f + \mathbb{E}_{\mathbb{P}}[\ell''(y, f(\mathbf{x}))f(\mathbf{x})\Phi(\mathbf{x})].$$

The proof is given in Appendix C.

Remark 2. In Theorem 3, the loss function ℓ is assumed to be convex with zero third derivative w.r.t. the second variable. It seems that this assumption might be too strong. In fact, from the proof of this theorem, one can see that the assumption of the third derivative w.r.t. the second variable is 0 can be removed. We add this assumption is to make the form of the high order BIFs simple.

Furthermore, if the loss function is also symmetric, meaning $f(t_1,t_2) = f(t_2,t_1)$, we only need to assume that the high order derivative (no matter w.r.t. the first or second variable) of the loss function is existence.

5 Calculation of BIFs for Sample Distribution

In this section, we will evaluate the BIFs for the sample distribution $\mathbb{P}_{\mathcal{S}}$ for a wide of kernel-based algorithms, including LSSVM, KRR, L1-SVM, L2-SVM and ϵ-SVR, and further apply these results as criteria to approximate the t-fold CVE.

5.1 LSSVM and KRR

LSSVM and KRR are two popular learning machines with square loss for solving classification and regression problems, respectively. From Theorem 2, the operator L at sample distribution $\mathbb{P}_{\mathcal{S}}$ maps $f_{\kappa,\mathbb{P}_{\mathcal{S}}}\in\mathcal{H}$ to

$$L(f_{\kappa,\mathbb{P}_{\mathcal{S}}}) = 2\lambda f_{\kappa,\mathbb{P}_{\mathcal{S}}} + \frac{2}{n}\sum_{j=1}^{n} f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_j)\Phi(\mathbf{x}_j).$$

Thus, one can see that

$$\begin{bmatrix} L(f_{\kappa,\mathbb{P}_{\mathcal{S}}})(\mathbf{x}_1) \\ \vdots \\ L(f_{\kappa,\mathbb{P}_{\mathcal{S}}})(\mathbf{x}_n)\end{bmatrix} = 2\left[\lambda\mathbf{I}_n + \frac{1}{n}\mathbf{K}\right]\begin{bmatrix} f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_1) \\ \vdots \\ f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_n)\end{bmatrix}, \tag{6}$$

where $\mathbf{K} = [\kappa(\mathbf{x}_i,\mathbf{x}_j)]_{i,j=1}^n$, $\mathbf{I}_n$ is the $n\times n$ identity matrix. Equation (6) indicates that the matrix

$$2\mathbf{L}_n := 2\left[\lambda\mathbf{I}_n + \frac{1}{n}\mathbf{K}\right],$$

is the finite sample version of the operator L at $\mathbb{P}_{\mathcal{S}}$. From Theorem 2, it is now clear that

$$\begin{bmatrix} \mathrm{BIF}(\mathbf{x}_1) \\ \vdots \\ \mathrm{BIF}(\mathbf{x}_n) \end{bmatrix} = \mathbf{L}_n^{-1}\left[-\frac{1}{l}[\mathbf{K}\circ\mathbf{S}_i]\mathbf{g} - \lambda \mathbf{f}_{\kappa,\mathbb{P}_{\mathcal{S}}}\right], \tag{7}$$

where

$$\mathbf{g} = \frac{1}{2}\begin{bmatrix} \ell'(y_1, f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_1)) \\ \vdots \\ \ell'(y_n, f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_n)) \end{bmatrix} = -\begin{bmatrix} y_1 - f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_1) \\ \vdots \\ y_n - f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_n) \end{bmatrix};$$

$\mathbf{f}_{\kappa,\mathbb{P}_{\mathcal{S}}} = (f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_1), \ldots, f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_n))^{\mathrm{T}}$;

$\mathbf{S}_i$ is an $n \times n$ matrix with $[\mathbf{S}_i]_{j,k} = \begin{cases} 1 & \text{if } \mathbf{x}_k \in \mathcal{S}_i, \\ 0 & \text{otherwise;} \end{cases}$

$\circ$ is the entrywise matrix product.

For t-fold CV, define $\mathbf{B}_k$ as the $n \times t$ matrix with

$$[\mathbf{B}_k]_{j,i} = \mathrm{BIF}_k(\mathbb{P}_{\mathcal{S}_i}; f_\kappa, \mathbb{P}_{\mathcal{S}})(\mathbf{x}_j), j \in \{1, \ldots, n\}, i \in \{1, \ldots, t\},$$

and $\mathbf{B}_k^i$ the ith column of $\mathbf{B}_k$. From Theorem 3, one sees similarly that the higher order terms can be computed by

$$\mathbf{B}_{k+1}^i = (k+1)\mathbf{L}_n^{-1}\left[\frac{1}{n}\mathbf{K}\mathbf{B}_k^i - \frac{1}{l}[\mathbf{K}\circ\mathbf{S}_i]\mathbf{B}_k^i\right]. \tag{8}$$

Let $\mathbf{Q}$ be the $n \times t$ matrix with

$$[\mathbf{Q}]_{j,i} = \begin{cases} 1 & \text{if } \mathbf{x}_j \in \mathcal{S}_i, \\ \frac{1}{1-t} & \text{otherwise,} \end{cases} \text{ and } \mathbf{G} = \frac{1}{n}\mathbf{L}_n^{-1}\mathbf{K},$$

then Equation (8) gives

$$\mathbf{B}_{k+1} = (k+1)\mathbf{G}[\mathbf{B}_k \circ \mathbf{Q}](1-t). \tag{9}$$

According to Equation (4), by cutting it off at some step r, we have

$$f_{\kappa,\mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}}(\mathbf{x}_j) \approx f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_j) + \sum_{s=1}^{r}\left[\frac{-1}{t-1}\right]^s \frac{1}{s!}[\mathbf{B}_s]_{j,i}. \tag{10}$$

Remark 3. Note that $\mathbf{K}$ is a positive semidefinite matrix, thus $\mathbf{L}_n = \lambda\mathbf{I}_n + \frac{1}{n}\mathbf{K}$ a positive definite matrix, and $\|\mathbf{L}_n^{-1}\|_2 \le \frac{1}{\lambda}$. Therefore, if the kernel, the first derivative of the loss function and the element of $\mathcal{H}$ are bounded, then from Equations (7) and (8), it is easy to verity that the BIF and High order BIFs are all bounded.

The similar analysis can easy be extended to L2-SVM, L1-SVM and ϵ-SVR.

5.2 L2-SVM

The loss function of L2-SVM is $\ell(y,t) = \max(0, 1-yt)^2$. Thus, we have

$$\ell'(y,t) = \begin{cases} 0 & \text{if } yt > 1 \\ -2y(1-yt) & \text{if } yt < 1, \end{cases}$$

$$\ell''(y,t) = \begin{cases} 0 & \text{if } yt > 1 \\ 2 & \text{if } yt < 1. \end{cases}$$

Note that the derivatives with respect to the second variable of $\ell(y,t)$ at $yt = 1$ do not exist, but in practice the probability that $yt = 1$ is 0, so we can ignore this possibility.[2]

A point $\mathbf{x}_i$ is a *support vector* if $y_i(f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_i)) < 1$. Let us reorder the training points such that the first n_{sv} points are support vectors. Let $\mathbf{I}^0$ be the $n \times n$ diagonal matrix with the first n_{sv} entries being 1 and the others 0.

From Theorem 2, the operator L at sample distribution $\mathbb{P}_{\mathcal{S}}$ maps any $f_{\kappa,\mathbb{P}_{\mathcal{S}}} \in \mathcal{H}$ to

$$L(f_{\kappa,\mathbb{P}_{\mathcal{S}}}) = 2\lambda f_{\kappa,\mathbb{P}_{\mathcal{S}}} + \frac{2}{n}\sum_{j=1}^{n_{\mathrm{sv}}} f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_j)\Phi(\mathbf{x}_j).$$

Thus, we have

$$\begin{bmatrix} L(f_{\kappa,\mathbb{P}_{\mathcal{S}}})(\mathbf{x}_1) \\ \vdots \\ L(f_{\kappa,\mathbb{P}_{\mathcal{S}}})(\mathbf{x}_n) \end{bmatrix} = 2\left[\lambda\mathbf{I}_n + \frac{1}{n}\mathbf{K}\mathbf{I}^0\right]\begin{bmatrix} f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_1) \\ \vdots \\ f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_n) \end{bmatrix}.$$

So, the matrix

$$2\mathbf{L}_n := 2\left[\lambda\mathbf{I}_n + \frac{1}{n}\mathbf{K}\mathbf{I}^0\right],$$

is the finite sample version of the operator L at $\mathbb{P}_{\mathcal{S}}$. Similar to LSSVM and KRR, the following equations hold:

$$\mathbf{B}_1^i = \mathbf{L}_n^{-1}\left[-\frac{1}{l}[\mathbf{K}\circ\mathbf{S}_i]\mathbf{I}^0\mathbf{g} - \lambda\mathbf{f}_{\kappa,\mathbb{P}_{\mathcal{S}}}\right],$$

$$\mathbf{B}_{k+1} = (k+1)\mathbf{G}[\mathbf{B}_k \circ \mathbf{Q}](1-t),$$

where $\mathbf{G} = \frac{1}{n}\mathbf{L}_n^{-1}\mathbf{K}\mathbf{I}^0, \mathbf{g} = (g_1, \ldots, g_n)^{\mathrm{T}}, g_i = f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_i) - y_i$, and $\mathbf{f}_{\kappa,\mathbb{P}_{\mathcal{S}}} = (f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_1), \ldots, f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_n))^{\mathrm{T}}$. Thus, we have

$$f_{\kappa,\mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}}(\mathbf{x}_j) \approx f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_j) + \sum_{s=1}^{r}\frac{[\mathbf{B}_s]_{j,i}}{(1-t)^s s!}.$$

5.3 L1-SVM

The hinge loss used in L1-SVM is not differentiable. We propose to use a differentiable approximation of it, inspired by the Huber loss

$$\ell(y,t) = \begin{cases} 0 & \text{if } yt > 1+h, \\ \frac{(1+h-yt)^2}{4h} & \text{if } |1-yt| \le h, \\ 1-yt & \text{if } yt < 1-h. \end{cases}$$

If $h \to 0$, the Huber loss converges to the hinge loss. It is easy to verify that

$$\ell'(y,t) = \begin{cases} 0 & \text{if } yt > 1+h, \\ \frac{-y(1+h-yt)}{2h} & \text{if } |1-yt| \le h, \\ -y & \text{if } yt < 1-h, \end{cases}$$

$$\ell''(y,t) = \begin{cases} 0 & \text{if } yt > 1+h, \\ \frac{1}{2h} & \text{if } |1-yt| \le h, \\ 0 & \text{if } yt < 1-h. \end{cases}$$

We say that a point $\mathbf{x}_i$ is a *support vector*if $|1 - y_i(f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_i))| \le h$. Similar with L2-SVM, we can obtain that

2. Alternatively, we can use subdifferentials of these points.

$$\mathbf{L}_n := 2\lambda \mathbf{I}_n + \frac{1}{2hn}\mathbf{K}\mathbf{I}^0,$$

is the finite sample version of the operator L at $\mathbb{P}_{\mathcal{S}}$, and the following equations hold:

$$\mathbf{B}_1^i = \mathbf{L}_n^{-1}\left[-\frac{1}{l}[\mathbf{K}\circ\mathbf{S}_i]\mathbf{g} - 2\lambda \mathbf{f}_{\kappa,\mathbb{P}_{\mathcal{S}}}\right],$$
$$\mathbf{B}_{k+1} = (k+1)\mathbf{G}[\mathbf{B}_k\circ\mathbf{Q}](1-t),$$

where $\mathbf{g} = (g_1,\ldots,g_n)^{\mathrm{T}}$, $g_i = \ell'(y_i, f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_i))$, $\mathbf{G} = \frac{1}{2hn}\mathbf{L}_n^{-1}\mathbf{K}\mathbf{I}^0$ and $\mathbf{f}_{\kappa,\mathbb{P}_{\mathcal{S}}} = (f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_1),\ldots,f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_n))^{\mathrm{T}}$.

5.4 ϵ-SVR

The loss $\max(0,|y-t|-\epsilon)$ of ϵ-SVR is not differentiable. Let $d = |y-t|$, we consider the use of the following Huber loss

$$\ell(y,t) = \begin{cases} 0 & \text{if } d < \epsilon - h, \\ \frac{(h+d-\epsilon)^2}{4h} & \text{if } -h \le d-\epsilon \le h, \\ d-\epsilon & \text{if } d > \epsilon + h. \end{cases}$$

If $h \to 0$, this Huber loss converges to $\max(0, d-\epsilon)$. Note that

$$\ell'(y,t) = \begin{cases} 0 & \text{if } d < \epsilon - h, \\ -\frac{(h+d-\epsilon)\mathrm{sign}(y-t)}{2h} & \text{if } -h \le d-\epsilon \le h, \\ -\mathrm{sign}(y-t) & \text{if } d > \epsilon + h, \end{cases}$$

and

$$\ell''(y,t) = \begin{cases} 0 & \text{if } d < \epsilon - h, \\ \frac{1}{2h} & \text{if } -h \le d-\epsilon \le h, \\ 0 & \text{if } d > \epsilon + h. \end{cases}$$

We say that a point $\mathbf{x}_i$ is a *support vector*if $\epsilon - h \le |y_i - f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_i)| \le \epsilon + h$.

Note that the matrix

$$\mathbf{L}_n := 2\lambda \mathbf{I}_n + \frac{1}{2hn}\mathbf{K}\mathbf{I}^0$$

is the finite sample version of the operator L at $\mathbb{P}_{\mathcal{S}}$. Similar with L1-SVM, we have

$$\mathbf{B}_1^i = \mathbf{L}_n^{-1}\left[-\frac{1}{l}[\mathbf{K}\circ\mathbf{S}_i]\mathbf{g} - 2\lambda \mathbf{f}_{\kappa,\mathbb{P}_{\mathcal{S}}}\right],$$
$$\mathbf{B}_{k+1} = (k+1)\mathbf{G}[\mathbf{B}_k\circ\mathbf{Q}](1-t),$$

where $\mathbf{G} = \frac{1}{2hn}\mathbf{L}_n^{-1}\mathbf{K}\mathbf{I}^0$.

5.5 BIF Criterion

The traditional t-fold CVE is given by

$$t-\mathrm{CV} = \frac{1}{n}\sum_{i=1}^{t}\sum_{z_j\in\mathcal{S}_i} V(y_j, f_{\kappa,\mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}}(\mathbf{x}_j)).$$

The idea is to replace $f_{\kappa,\mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}}(\mathbf{x}_j)$ by

$$f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_j) + \sum_{s=1}^{r}\frac{[\mathbf{B}_s]_{j,i}}{(1-t)^s s!}.$$

Thus, the rth order BIF criterion of the approximation of t-fold CVE is defined as

$$\mathrm{BIF}_t^r := \frac{1}{n}\sum_{i=1}^{t}\sum_{z_j\in\mathcal{S}_i} V\left(y_j, f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_j) + \sum_{s=1}^{r}\frac{[\mathbf{B}_s]_{j,i}}{(1-t)^s s!}\right).$$

To compute BIF_t^r, we need to compute $f_{\kappa,\mathbb{P}_{\mathcal{S}}}$ on the full data set, the inversion of $\mathbf{L}_n$ and BIF matrices. For LSSVM and KRR, the calculation of $f_{\kappa,\mathbb{P}_{\mathcal{S}}}$ includes the calculation of the inversion of $\mathbf{L}_n$. For L1-SVM, L2-SVM and ϵ-SVR, the time complexity of computing the inversion of $\mathbf{L}_n$ is $O(n_{\mathrm{sv}}^3)$, where n_{sv} is the size of support vectors that is usually much smaller than the size of data n, i.e., $n_{\mathrm{sv}} \ll n$. The time complexity of computing BIF matrices is $O(tn^2 + rn^2)$, where t is the fold of CV and r is the order of the Taylor expansion. Thus, the overall time complexity of BIF_t^r is

$$O(n^3 + rn^2 + tn^2),$$

for LSSVM and KRR, and

$$O(n^3 + n_{\mathrm{sv}}^3 + rn^2 + tn^2),$$

for L1-SVM, L2-SVM and ϵ-SVR.

For the traditional t-fold CV, the algorithm need to be executed t times, the time complexity is $O(t(\frac{t-1}{t}n)^3) = O(\frac{(t-1)^3}{t^2}n^3)$. Thus, the proposed approximate t-fold CV is much more efficient for a moderate value of t.

6 Experiments

In this section, we will empirically analyze the performance of the proposed approximate t-fold cross-validation using the criterion of t-BIF.

The data sets are 20 publicly available data sets from LIBSVM Data:[3] 10 data sets for classification and 10 data sets for regression. We use the popular Gaussian kernel

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

and polynomial kernel[4]

$$\kappa(\mathbf{x},\mathbf{x}') = (\mathbf{x}^{\mathrm{T}}\mathbf{x}' + 1)^d,$$

with $\sigma \in \{2^i, i = -10,-9,\ldots,10\}$, $d \in \{1,2,\ldots,10\}$. The regularization parameter[5] $\lambda \in \{2^i/n, i = -3,-2,\ldots,11\}$. For each data set, we run all the methods 10 times with data sets being split randomly (50 percent of all the examples for training and the other 50 percent for testing). The use of multiple training/test partitions allows an estimate of the statistical significance for the performance of different cross-validation 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 $\mathcal{S}_d$ be the mean and standard error of d_i. Then under t-test, with

3. http://www.csie.ntu.edu.tw/~cjlin/libsvm.

4. In this paper, we only report the results with polynomial kernel for LSSVM. For other learning machines, such as L2-SVM, L1-SVM, KRR, ϵ-SVR, the results of polynomial kernel are similar to LSSVM.

5. The value of λ we set seems too small at first sight, but in fact, the regularized algorithm we considered in this paper is $\frac{1}{n}\sum_{i=1}^{n}\ell(f_{\kappa,\mathbb{P}_{\mathcal{S}}}(x_i), y_i) + \lambda\|f\|_{\kappa}^2$, while other authors usually ignore $1/n$. Therefore, the value of λ in our paper is $1/n$ time of that of regularized algorithms other authors considered.

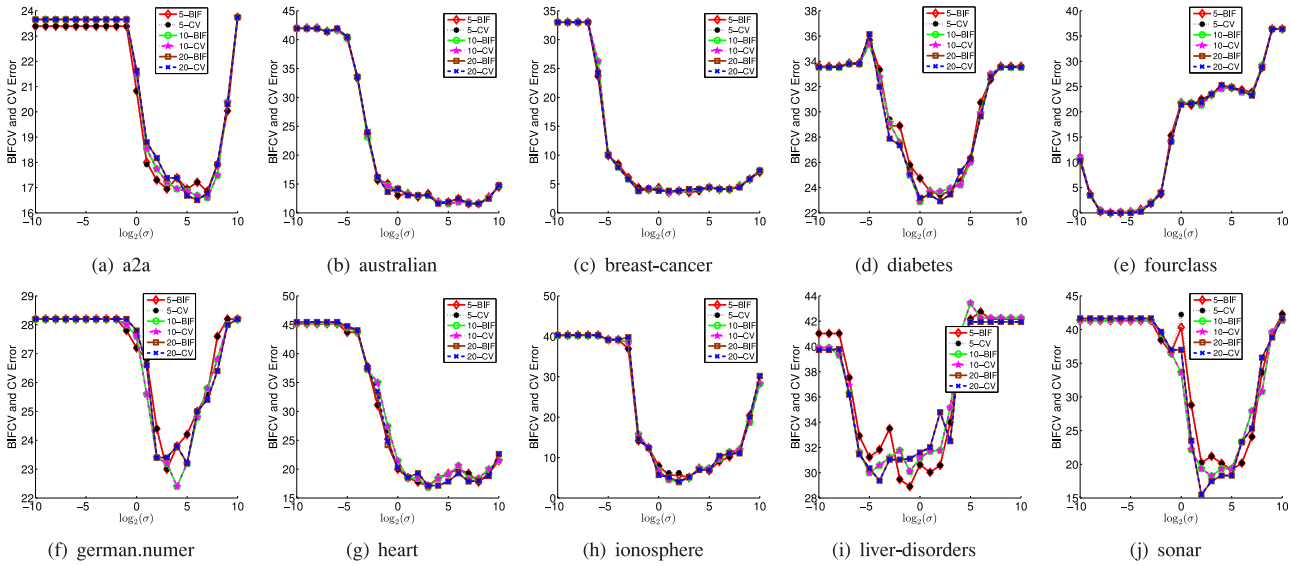


Fig. 1. The traditional t-CV errors and t-BIF errors for LSSVM with $t = 5, 10, 20$. The order of the Taylor expansion $r = 5$ and the regularization parameter $\lambda = 1/n$.

TABLE 1
The Average Test Errors (%) of t-BIF and t-CV on the Classification Data Sets for LSSVM with $t = 5, 10, 20$

LSSVM	5-CV	5-BIF	10-CV	10-BIF	20-CV	20-BIF
australian	15.19 ± 0.88	15.19 ± 0.88	14.67 ± 0.78	14.67 ± 0.78	15.01 ± 0.38	15.01 ± 0.38
heart	17.33 ± 3.42	17.33 ± 3.42	16.89 ± 3.20	17.78 ± 1.74	17.63 ± 2.94	17.63 ± 2.94
ionosphere	7.20 ± 1.11	7.20 ± 1.11	6.97 ± 0.85	6.74 ± 0.85	6.74 ± 0.48	6.74 ± 0.48
breast-cancer	3.28 ± 0.56	3.28 ± 0.56	3.40 ± 0.61	3.40 ± 0.61	3.40 ± 0.61	3.40 ± 0.61
diabetes	23.44 ± 1.55	23.44 ± 1.55	25.05 ± 0.50	25.05 ± 0.50	24.22 ± 1.43	24.22 ± 1.43
fourclass	0.19 ± 0.42	0.23 ± 0.40	0.23 ± 0.40	0.23 ± 0.40	0.23 ± 0.40	0.23 ± 0.40
german.numer	24.56 ± 0.61	24.56 ± 0.61	24.96 ± 0.79	25.04 ± 0.89	24.84 ± 0.59	24.84 ± 0.59
liver-disorders	31.86 ± 2.83	31.86 ± 2.83	31.16 ± 1.06	31.16 ± 1.06	29.53 ± 2.83	29.53 ± 2.83
sonar	17.85 ± 8.46	17.31 ± 5.48	17.88 ± 5.51	17.50 ± 5.46	17.12 ± 6.17	17.88 ± 5.34
a2a	18.39 ± 1.20	18.39 ± 1.20	18.57 ± 1.64	18.57 ± 1.64	18.07 ± 0.79	18.07 ± 0.79

For each training set, we choose the kernel parameter σ and regularization parameter λ by the two criteria on the training set respectively, and evaluate the test errors for the chosen parameters on the test set. The order of the Taylor expansion $r = 5$.

confidence level 95 percent, we claim that A is significantly better than B (or equivalently B significantly worse than A) if the t-statistic

$$\frac{\bar{d}}{\mathcal{S}_d/\sqrt{10}} > 1.833.$$

All statements of statistical significance in the paper refer to a 95 percent level of significance. The learning machines we considered including LSSVM, L2-SVM, KRR, L1-SVM and ϵ-SVR. The codes of L2-SVM is from [50] implemented in Matlab, L1-SVM and ϵ-SVR are from LIBSVM [51] implemented in C++. LSSVM, KRR and the calculation of BIFs are implemented in Matlab. Experiments are conducted on a Dell PC with 3.1-GHz 4-core CPU and 4-GB memory.

6.1 LSSVM

In this section, we will verify the performance of t-BIF for LSSVM, $t = 5, 10, 20$.

6.1.1 Gaussian Kernel

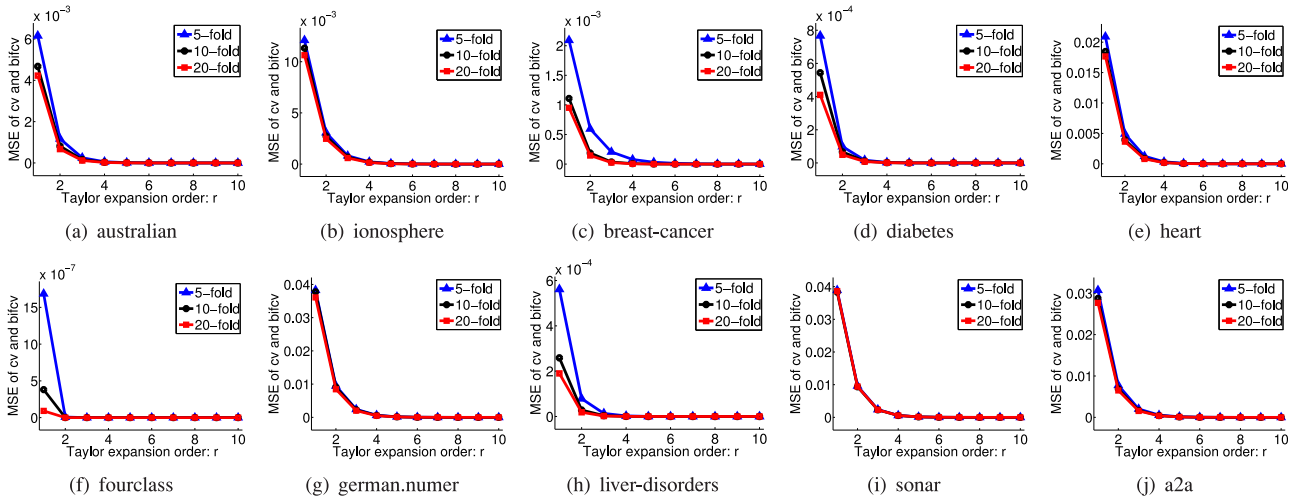
In our first experiment, we set the order of the Taylor expansion $r = 5$ (we will explore the influence of this parameter next). The traditional t-fold cross-validation error (t-CV) and t-BIF error with fixed $\lambda = 1/n$ are shown in Fig. 1. For each data set, we compute the t-CV and t-BIF errors with different σ, $\sigma \in \{2^i, i = -10, -9, \ldots, 10\}$. One can see that t-BIF is coincident with traditional t-CV on all data sets, $t \in \{5, 10, 20\}$. This demonstrates that t-BIF approximates t-CV perfectly well.

The average test errors of t-BIF and t-CV for Gaussian kernel are reported in Table 1. For each training set, we choose the $\sigma \in \{2^i, i = -10, -9, \ldots, 9, 10\}$ and $\lambda \in \{2^i/n, i = -3, -2, \ldots, 11\}$ by the two criteria on the training set respectively, and evaluate the test errors for the chosen parameters on the test set. The test errors of t-BIF and t-CV are very similar, neither t-BIF nor t-CV criterion is shown to be significantly better than the other on any of the data sets. In particular, on australian, heart, ionosphere, breast-cancer, diabetes, german.numer and a2a data sets, 5-BIF gives the same test errors as 5-CV. On the remaining data sets, both 5-BIF and 5-CV give the similar results. The results of $t = 10, 20$ are similar to that of $t = 5$. The above experimental results demonstrate that the quality of our approximation of CV based on BIF is quite good.

The computational time of t-BIF and t-CV are listed in Table 2. We can find that the time cost of t-BIF is much

TABLE 2
The Average Computational Time (in Seconds) of t-BIF and t-CV for LSSVM with $t = 5, 10, 20$, and the Order of the Taylor Expansion $r = 5$

LSSVM	5-CV	5-BIF	10-CV	10-BIF	20-CV	20-BIF
australian	11.22 ± 0.10	7.50 ± 0.11	28.06 ± 0.49	9.76 ± 0.18	59.14 ± 0.20	14.42 ± 0.09
heart	2.19 ± 0.03	1.46 ± 0.02	5.15 ± 0.06	2.20 ± 0.11	11.24 ± 0.10	3.58 ± 0.03
ionosphere	3.42 ± 0.02	2.03 ± 0.01	8.02 ± 0.06	2.93 ± 0.01	17.18 ± 0.10	4.72 ± 0.02
breast-cancer	10.52 ± 0.09	6.99 ± 0.03	25.53 ± 0.40	9.24 ± 0.06	57.96 ± 0.69	14.00 ± 0.04
diabetes	13.45 ± 0.06	11.92 ± 0.02	33.26 ± 0.77	16.68 ± 0.24	85.65 ± 0.19	25.90 ± 0.16
fourclass	14.19 ± 0.05	14.49 ± 0.03	41.12 ± 0.17	20.56 ± 0.05	92.98 ± 0.30	32.60 ± 0.09
german.numer	27.05 ± 0.10	21.91 ± 0.08	68.54 ± 0.14	30.79 ± 0.14	157.59 ± 0.31	48.31 ± 0.15
liver-disorders	2.89 ± 0.04	1.96 ± 0.05	6.80 ± 0.03	2.80 ± 0.01	14.46 ± 0.04	4.53 ± 0.01
sonar	1.69 ± 0.02	1.06 ± 0.01	3.72 ± 0.03	1.62 ± 0.01	7.64 ± 0.05	2.73 ± 0.01
a2a	175.57 ± 0.83	152.77 ± 0.33	463.19 ± 0.86	205.96 ± 0.65	1076.65 ± 5.63	312.96 ± 0.53



(a) australian (b) ionosphere (c) breast-cancer (d) diabetes (e) heart

(f) fourclass (g) german.numer (h) liver-disorders (i) sonar (j) a2a

Fig. 2. The mean square discrepancies between t-CV and t-BIF with different r, where r is the order of the Taylor expansion. For each training set, we choose the σ and λ by t-fold cross validation on the training set. Plotted are the mean square error of the approximate $f_{\kappa,\mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}}(\mathbf{x})$ and $f_{\kappa,\mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}}(\mathbf{x})$ for the chosen parameters on the validation sample $\mathcal{S}_i$.

TABLE 3
The Average Test Errors (%) of t-BIF and t-CV on the Classification Data Sets for LSSVM with Polynomial Kernel, $t = 5, 10, 20$

LSSVM	5-CV	5-BIF	10-CV	10-BIF	20-CV	20-BIF
australian	14.43 ± 1.44	15.19 ± 2.17	14.14 ± 1.53	14.84 ± 2.39	14.26 ± 1.09	14.96 ± 2.11
heart	18.81 ± 1.71	18.81 ± 1.71	19.85 ± 2.06	19.85 ± 2.06	18.22 ± 3.17	18.52 ± 2.82
ionosphere	6.06 ± 1.11	5.37 ± 0.87	5.83 ± 1.10	5.37 ± 0.87	6.17 ± 1.10	5.37 ± 0.87
breast-cancer	3.28 ± 0.56	3.17 ± 0.43	2.93 ± 0.69	3.11 ± 0.49	3.11 ± 0.84	3.11 ± 0.49
diabetes	22.29 ± 1.61	22.29 ± 1.61	22.24 ± 1.24	22.24 ± 1.24	21.93 ± 1.60	21.93 ± 1.60
fourclass	0.19 ± 0.13	0.19 ± 0.13	0.24 ± 0.13	0.24 ± 0.13	0.24 ± 0.13	0.24 ± 0.13
german.numer	25.16 ± 2.49	26.68 ± 1.04	24.72 ± 1.72	26.68 ± 1.04	24.52 ± 1.35	26.68 ± 1.04
liver-disorders	29.30 ± 1.06	29.30 ± 1.06	28.72 ± 1.52	28.72 ± 1.52	29.19 ± 1.56	29.19 ± 1.56
sonar	16.92 ± 2.77	17.88 ± 3.88	18.65 ± 1.75	18.08 ± 3.56	17.50 ± 1.25	18.85 ± 2.60
a2a	19.49 ± 0.93	19.60 ± 0.70	19.58 ± 1.09	19.58 ± 1.09	19.63 ± 0.96	19.63 ± 0.96

For each training set, we choose the kernel parameter σ and regularization parameter λ by the two criteria on the training set respectively, and evaluate the test errors for the chosen parameters on the test set. The order of the Taylor expansion $r = 5$.

lower than that of t-CV, especially for large number of folds. Thus, t-BIF can significantly improve the efficiency of t-CV for model selection.

Now we explore the influence of the order r of the Taylor expansion. The discrepancies between t-CV and t-BIF with different r are given in Fig. 2. For each training set, we choose the σ and λ by t-CV on the training set. Plotted are the mean square error of the approximate $f_{\kappa,\mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}}(\mathbf{x})$ (computed by t-BIF) and $f_{\kappa,\mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}}(\mathbf{x})$ (computed by t-CV) for the chosen parameters on the validation sample $\mathcal{S}_i$. We can see that on most data sets there is no discrepancy between t-CV and t-BIF when $r \geq 4$. Thus, we can select $t = 4$ or 5 in practice without sacrificing accuracy.

6.1.2 Polynomial Kernel

The average test errors of t-BIF and t-CV for polynomial kernel are reported in Table 3. For each training set, we choose the $d \in \{i = 1, 2, \ldots, 10\}$ and $\lambda \in \{2^i/n, i = -3, -2, \ldots, 11\}$

TABLE 4
The Average Computational Time (in Seconds) of t-BIF and t-CV for LSSVM with Polynomial Kernel, $t = 5, 10, 20$ and the Order of the Taylor Expansion $r = 5$

LSSVM	5-CV	5-BIF	10-CV	10-BIF	20-CV	20-BIF
australian	11.24 ± 0.31	7.82 ± 0.09	28.09 ± 1.33	9.05 ± 1.46	53.70 ± 1.44	14.92 ± 1.27
heart	2.60 ± 0.02	1.42 ± 0.19	5.65 ± 0.16	2.29 ± 0.41	11.88 ± 0.85	3.55 ± 0.81
ionosphere	3.42 ± 0.14	2.05 ± 0.17	8.82 ± 0.55	2.26 ± 0.49	17.69 ± 1.90	4.80 ± 0.33
breast-cancer	10.78 ± 0.03	6.89 ± 0.05	25.09 ± 0.02	9.63 ± 0.04	16.37 ± 0.02	4.42 ± 0.08
diabetes	13.08 ± 0.09	11.10 ± 0.09	33.98 ± 0.05	15.81 ± 0.54	86.69 ± 0.14	24.50 ± 0.92
fourclass	14.13 ± 0.23	13.37 ± 0.21	41.55 ± 0.11	20.56 ± 0.02	96.44 ± 0.49	34.08 ± 0.90
german.numer	27.07 ± 0.02	21.41 ± 0.09	68.65 ± 0.71	30.81 ± 0.40	156.06 ± 0.40	47.12 ± 0.30
liver-disorders	2.83 ± 0.09	1.97 ± 0.04	6.69 ± 0.02	2.90 ± 0.07	14.46 ± 0.03	4.51 ± 0.04
sonar	1.63 ± 0.01	1.02 ± 0.04	3.77 ± 0.04	1.64 ± 0.07	7.94 ± 0.05	2.74 ± 0.03
a2a	175.34 ± 0.86	152.75 ± 0.32	436.11 ± 0.75	213.27 ± 0.74	1076.95 ± 2.69	312.76 ± 0.73

TABLE 5
The Average Test Errors (%) of t-BIF and t-CV for L2-SVM with the Order of the Taylor Expansion $r = 5$

L2-SVM	5-CV	5-BIF	10-CV	10-BIF	20-CV	20-BIF
australian	15.13 ± 1.07	15.13 ± 1.07	14.38 ± 0.86	14.61 ± 0.86	14.55 ± 1.33	14.55 ± 1.33
heart	17.04 ± 3.19	17.04 ± 3.19	17.19 ± 3.41	17.19 ± 3.41	17.78 ± 3.05	17.33 ± 2.49
ionosphere	7.54 ± 1.17	7.09 ± 1.65	7.31 ± 0.85	6.86 ± 0.57	6.74 ± 0.48	6.29 ± 1.07
breast-cancer	3.11 ± 0.61	3.11 ± 0.61	3.23 ± 0.59	3.23 ± 0.59	3.52 ± 0.46	3.52 ± 0.46
diabetes	23.70 ± 1.46	23.70 ± 1.46	24.01 ± 1.37	24.01 ± 1.37	24.17 ± 1.38	23.80 ± 1.59
fourclass	0.23 ± 0.40	0.23 ± 0.40	0.28 ± 0.38	0.05 ± 0.10	0.28 ± 0.38	0.05 ± 0.10
german.numer	24.92 ± 0.88	24.96 ± 0.89	24.52 ± 0.48	24.52 ± 0.48	25.44 ± 0.55	25.36 ± 0.62
liver-disorders	30.81 ± 3.13	30.70 ± 3.14	30.12 ± 2.19	29.77 ± 1.81	28.84 ± 1.87	28.84 ± 1.87
sonar	19.62 ± 7.98	17.69 ± 6.03	17.88 ± 5.12	17.69 ± 4.88	17.12 ± 4.87	17.12 ± 4.87
a2a	18.23 ± 1.27	18.14 ± 1.30	18.45 ± 1.45	18.46 ± 1.48	18.43 ± 1.41	18.45 ± 1.40

For each training set, we choose the kernel parameter σ and regularization parameter λ by each criterion on the training set respectively, and evaluate the test errors for the chosen parameters on the test set.

TABLE 6
The Average Computational Time (in Seconds) of t-BIF and t-CV for L2-SVM with the Order of the Taylor Expansion $r = 5$

L2-SVM	5-CV	5-BIF	10-CV	10-BIF	20-CV	20-BIF
australian	25.42 ± 2.82	8.50 ± 0.21	55.85 ± 3.41	10.86 ± 0.25	105.60 ± 4.42	15.69 ± 0.30
heart	8.94 ± 0.28	2.19 ± 0.03	19.09 ± 0.56	2.89 ± 0.06	35.35 ± 1.14	4.23 ± 0.05
ionosphere	17.31 ± 1.05	2.77 ± 0.07	31.15 ± 0.76	3.68 ± 0.10	52.24 ± 1.82	5.59 ± 0.21
breast-cancer	33.19 ± 1.86	9.18 ± 0.05	61.48 ± 0.92	11.41 ± 0.15	118.46 ± 1.44	16.27 ± 0.09
diabetes	31.07 ± 0.95	14.20 ± 0.12	64.32 ± 0.50	19.77 ± 0.07	130.19 ± 2.52	30.73 ± 0.09
fourclass	37.45 ± 1.25	18.07 ± 0.08	76.66 ± 1.37	25.20 ± 0.07	150.90 ± 1.19	39.40 ± 0.12
german.numer	40.40 ± 0.29	25.01 ± 0.12	92.77 ± 1.04	35.06 ± 0.14	185.18 ± 1.33	55.29 ± 0.18
liver-disorders	13.23 ± 0.69	2.36 ± 0.05	23.47 ± 4.08	3.25 ± 0.03	40.93 ± 4.57	5.13 ± 0.12
sonar	7.18 ± 1.31	1.85 ± 0.03	17.18 ± 0.21	2.41 ± 0.02	31.83 ± 0.44	3.60 ± 0.02
a2a	172.39 ± 1.25	165.69 ± 0.41	440.68 ± 1.64	226.81 ± 0.28	955.76 ± 4.60	350.81 ± 0.37

by the two criteria on the training set respectively, and evaluate the test errors for the chosen parameters on the test set. Neither t-BIF nor t-CV criterion with polynomial kernel proving significantly better than the other on any of the data sets, $t = 5, 10, 20$. The above experimental results demonstrate that our proposed approximate CV method can also be applied for the polynomial kernel.

The computational time of t-BIF and t-CV for polynomial kernel are listed in Table 4. One can see that the time cost of t-BIF is much lower than that of t-CV, especially for large number of folds.

Remark 4. Theoretically, our approximate CV can be suitable for all the data sets when the BIF and high order BIFs are existence. However, from the experiments, one can see that for some data sets of small fold ($t = 5$), the computation time benefits much more reduction than others. This is because of the time complexity of BIF-criterion is $O(n^3 + tn^2 + rn^2)$, which is not much faster than the traditional CV of time complexity of $O(\frac{(t-1)^3}{t^2}n^3)$ for small fold.

6.2 L2-SVM

The learning machine used in this section is L2-SVM. The average test errors and computational time of t-BIF and t-CV are respectively reported in Tables 5 and 6. For each training set, we choose the σ and λ by each criterion on the training set respectively, and evaluate the test errors for the

TABLE 7
The Average Test Errors (%) of t-BIF and t-CV on the Classification Data Sets for L1-SVM with the Order of Taylor Expansion $r = 5$

Classification	5-CV	5-BIF	10-CV	10-BIF	20-CV	20-BIF
australian	15.19 ± 1.12	15.01 ± 1.22	14.96 ± 0.84	14.78 ± 1.10	14.90 ± 0.78	14.67 ± 1.49
heart	17.78 ± 1.89	18.37 ± 4.10	17.33 ± 1.78	17.04 ± 2.16	18.37 ± 0.97	17.19 ± 2.31
ionosphere	7.09 ± 1.04	7.43 ± 1.07	6.63 ± 1.25	7.31 ± 1.10	7.09 ± 1.04	7.89 ± 1.48
breast-cancer	3.05 ± 0.74	3.11 ± 0.49	2.99 ± 0.52	3.11 ± 0.53	3.28 ± 0.48	3.28 ± 0.60
diabetes	23.33 ± 1.99	23.85 ± 1.86	24.06 ± 1.82	24.22 ± 1.41	23.59 ± 1.48	24.06 ± 1.31
fourclass	0.19 ± 0.30	0.19 ± 0.30	0.28 ± 0.30	0.09 ± 0.13	0.28 ± 0.30	0.09 ± 0.13
german.numer	25.00 ± 0.55	26.48 ± 1.08	25.08 ± 0.64	26.36 ± 1.30	24.96 ± 0.46	26.36 ± 1.30
liver-disorders	32.67 ± 1.95	32.79 ± 1.46	31.98 ± 1.01	32.21 ± 3.20	32.21 ± 3.80	33.49 ± 1.87
sonar	21.35 ± 8.34	20.77 ± 5.55	17.31 ± 4.56	22.50 ± 8.78	17.12 ± 7.05	20.77 ± 5.55
a2a	18.52 ± 1.38	18.73 ± 4.43	18.98 ± 1.33	18.82 ± 4.39	18.66 ± 1.33	18.71 ± 4.42

For each training set, we choose the kernel parameter σ and regularization parameter λ by each criterion on the training set, and evaluate the test error for the chosen parameters on the test set.

TABLE 8
The Average Computational Time (in Seconds) of t-BIF and t-CV for L1-SVM with $t = 5, 10, 20$ and the Order of the Taylor Expansion $r = 5$

Classification	5-CV	5-BIF	10-CV	10-BIF	20-CV	20-BIF
australian	8.79 ± 0.11	6.35 ± 0.15	19.49 ± 0.21	7.52 ± 0.12	41.29 ± 0.56	9.99 ± 0.07
heart	2.21 ± 0.03	2.15 ± 0.41	4.78 ± 0.08	2.41 ± 0.30	9.80 ± 0.09	2.87 ± 0.12
ionosphere	4.70 ± 0.10	2.97 ± 0.14	10.24 ± 0.26	3.25 ± 0.22	21.27 ± 0.56	4.12 ± 0.10
breast-cancer	5.60 ± 0.10	4.74 ± 0.07	12.42 ± 0.23	5.90 ± 0.13	26.00 ± 0.40	8.29 ± 0.13
diabetes	7.59 ± 0.16	6.43 ± 0.06	17.04 ± 0.29	7.86 ± 0.09	35.84 ± 0.43	10.58 ± 0.07
fourclass	6.37 ± 0.21	6.90 ± 0.09	14.33 ± 0.44	8.70 ± 0.09	30.30 ± 0.97	12.23 ± 0.13
german.numer	22.66 ± 0.30	15.83 ± 0.23	50.88 ± 0.88	18.34 ± 0.20	107.24 ± 1.67	23.35 ± 0.17
liver-disorders	2.37 ± 0.05	1.92 ± 0.08	5.22 ± 0.13	2.38 ± 0.13	10.51 ± 0.19	3.12 ± 0.07
sonar	3.01 ± 0.06	2.00 ± 0.05	6.47 ± 0.12	2.32 ± 0.04	13.28 ± 0.12	2.82 ± 0.02
a2a	129.86 ± 0.93	144.55 ± 1.09	292.99 ± 2.56	156.29 ± 1.28	618.18 ± 5.10	181.50 ± 1.30

TABLE 9
The Testing Mean Square Error of t-BIF and t-CV for KRR, the Order of Taylor Expansion $r = 5$

KRR	5-CV	5-BIF	10-CV	10-BIF	20-CV	20-BIF
bodyfat	1.81e-5 $\pm$ 6.8e-6	1.79e-5 $\pm$ 6.6e-6	1.81e-5 $\pm$ 6.8e-6	1.81e-5 $\pm$ 6.8e-6	1.81e-5 $\pm$ 6.8e-6	1.81e-5 $\pm$ 6.8e-6
housing	13.51 ± 2.09	13.37 ± 2.26	13.23 ± 1.96	13.27 ± 2.33	12.95 ± 2.04	12.77 ± 2.48
mpg	7.84 ± 1.13	7.56 ± 0.89	7.63 ± 0.71	7.57 ± 0.69	7.57 ± 0.69	7.57 ± 0.69
pyrim	9.92e-3 $\pm$ 5.0e-3	9.92e-3 $\pm$ 5.0e-3	9.92e-3 $\pm$ 5.0e-3	9.92e-3 $\pm$ 5.0e-3	1.00e-2 $\pm$ 4.9e-3	9.92e-3 $\pm$ 5.0e-3
triazines	2.02e-2 $\pm$ 3.9e-3	2.01e-2 $\pm$ 3.9e-3	2.03e-2 $\pm$ 3.9e-3	2.03e-2 $\pm$ 3.8e-3	2.04e-2 $\pm$ 4.0e-3	2.02e-2 $\pm$ 3.9e-3
eunite2001	458.30 ± 33.50	458.30 ± 33.50	453.20 ± 38.04	453.20 ± 38.04	453.20 ± 38.04	453.20 ± 38.04
space-ga	1.19e-2 $\pm$ 1.3e-3	1.20e-2 $\pm$ 1.5e-3	1.20e-2 $\pm$ 1.5e-3	1.20e-2 $\pm$ 1.5e-3	1.20e-2 $\pm$ 1.5e-3	1.20e-2 $\pm$ 1.5e-3
cpusmall	10.47 ± 0.44	10.47 ± 0.44	10.35 ± 0.21	10.47 ± 0.44	10.35 ± 0.21	10.35 ± 0.21
mg	1.45e-2 $\pm$ 9.5e-4	1.44e-2 $\pm$ 1.1e-3	1.44e-2 $\pm$ 1.10e-3	1.44e-2 $\pm$ 1.1e-3	1.44e-2 $\pm$ 1.1e-3	1.44e-2 $\pm$ 1.1e-3
abalone	4.46 ± 0.25	4.46 ± 0.25	4.44 ± 0.26	4.44 ± 0.26	4.44 ± 0.26	4.44 ± 0.26

For each training set, we choose the kernel parameter σ and regularization parameter λ by each criterion on the training set, and evaluate the test error for the chosen parameters on the test set.

chosen parameters on the test set. Similar with the results of LSSVM, t-BIF gives almost the same accuracy results as t-CV but meanwhile significantly improves the efficiency. In practice, t-CV is not statistically superior to t-BIF at the 95 percent level of significance for accuracy on any of the data sets, but much slower than t-BIF, especially for large t.

6.3 L1-SVM

In this section, the h of Huber loss is set to be 0.01. The test errors of t-BIF and t-CV are reported in Table 7. t-CV is not statistically superior than t-BIF on any of data sets. However, compared with the performance of t-BIF for LSSVM and L2-SVM, we find that t-BIF for L1-SVM is a little worse, that is the accuracy results of t-BIF and t-CV are not always the same for L1-SVM. This may be explained by the fact that we substitute the Huber loss with small h for the hinge loss in this paper.

The computational times of t-BIF and t-CV are reported in Table 8. Note that L1-SVM is implemented in C++, but our approximate method is implemented in Matlab. Even so, we find that our t-BIF gives the comparable results with that of t-CV for small t, and is much faster for large t.

6.4 KRR

In this section, we use KRR for regression. The test mean square errors and computational times of t-BIF and t-CV are respectively reported in Tables 9 and 10. Similar to the results of classification, one can see that t-BIF gives nearly

TABLE 10
The Average Computational Time (in Seconds) of t-BIF and t-CV for KRR with the Order of the Taylor Expansion $r = 5$

KRR	5-CV	5-BIF	10-CV	10-BIF	20-CV	20-BIF
bodyfat	2.12 ± 0.07	1.26 ± 0.02	4.73 ± 0.04	1.96 ± 0.16	10.14 ± 0.51	3.15 ± 0.05
housing	6.27 ± 0.09	3.39 ± 0.02	14.74 ± 0.10	4.62 ± 0.06	32.66 ± 0.18	6.99 ± 0.06
mpg	4.18 ± 0.14	2.16 ± 0.01	9.45 ± 0.08	3.03 ± 0.02	20.80 ± 0.14	4.80 ± 0.06
pyrim	0.76 ± 0.01	0.54 ± 0.01	1.63 ± 0.02	0.90 ± 0.02	3.26 ± 0.03	1.59 ± 0.04
triazines	1.54 ± 0.02	0.91 ± 0.01	3.36 ± 0.02	1.41 ± 0.02	6.87 ± 0.07	2.37 ± 0.02
eunite2001	2.73 ± 0.03	1.62 ± 0.02	6.43 ± 0.07	2.39 ± 0.02	13.57 ± 0.06	3.93 ± 0.02
space-ga	422.81 ± 0.99	310.54 ± 0.34	1287.07 ± 2.88	402.34 ± 0.24	3131.25 ± 6.44	586.30 ± 0.50
cpusmall	273.93 ± 1.44	212.75 ± 0.49	794.09 ± 3.76	282.92 ± 0.45	1944.67 ± 8.03	423.66 ± 0.93
mg	56.92 ± 0.10	42.67 ± 0.08	151.48 ± 0.13	60.02 ± 0.05	344.30 ± 0.47	94.73 ± 0.13
abalone	1379.41 ± 3.98	802.11 ± 0.98	4028.64 ± 8.78	968.80 ± 1.19	9687.75 ± 17.95	1298.82 ± 1.47

TABLE 11
The Testing Mean Square Error on the Regression Data Sets for SVR with $r = 5$ and $h = 0.01$

ϵ-SVR	5-CV	5-BIF	10-CV	10-BIF	20-CV	20-BIF
bodyfat	$9.10\text{e-}5 \pm 1.7\text{e-}5$	$1.09\text{e-}4 \pm 3.9\text{e-}5$	$9.11\text{e-}5 \pm 1.7\text{e-}5$	$9.74\text{e-}5 \pm 2.4\text{e-}5$	$9.04\text{e-}5 \pm 1.78\text{e-}5$	$9.74\text{e-}5 \pm 2.4\text{e-}5$
housing	46.81 ± 3.91	46.72 ± 3.85	46.72 ± 3.85	47.59 ± 5.03	46.72 ± 3.85	47.59 ± 5.03
mpg	19.90 ± 2.84	19.69 ± 2.68	19.81 ± 2.94	19.69 ± 2.68	19.81 ± 2.94	19.69 ± 2.68
pyrim	$1.56\text{e-}2 \pm 5.6\text{e-}3$	$1.55\text{e-}2 \pm 5.4\text{e-}3$	$1.53\text{e-}2 \pm 5.9\text{e-}3$	$1.56\text{e-}2 \pm 5.4\text{e-}3$	$1.56\text{e-}2 \pm 5.5\text{e-}3$	$1.58\text{e-}2 \pm 6.5\text{e-}3$
triazines	$2.24\text{e-}2 \pm 3.5\text{e-}3$	$2.50\text{e-}2 \pm 4.6\text{e-}3$	$2.24\text{e-}2 \pm 3.4\text{e-}3$	$2.50\text{e-}2 \pm 4.6\text{e-}3$	$2.17\text{e-}2 \pm 2.6\text{e-}3$	$2.50\text{e-}2 \pm 4.6\text{e-}3$
eunite2001	1758.36 ± 173.3	1756.46 ± 172.6	1758.36 ± 173.3	1756.46 ± 172.6	1758.36 ± 173.3	1756.46 ± 172.6
space-ga	$1.41\text{e-}2 \pm 1.0\text{e-}3$	$1.41\text{e-}2 \pm 1.0\text{e-}3$	$1.41\text{e-}2 \pm 9.0\text{e-}4$	$1.41\text{e-}2 \pm 9.0\text{e-}4$	$1.41\text{e-}2 \pm 9.0\text{e-}4$	$1.41\text{e-}2 \pm 9.0\text{e-}4$
cpusmall	107.18 ± 10.76	106.24 ± 12.74	107.18 ± 10.76	106.24 ± 12.74	107.18 ± 10.76	106.24 ± 12.74
mg	$1.77\text{e-}2 \pm 8.9\text{e-}4$	$1.76\text{e-}2 \pm 7.5\text{e-}4$	$1.75\text{e-}2 \pm 6.8\text{e-}4$	$1.76\text{e-}2 \pm 8.3\text{e-}4$	$1.75\text{e-}2 \pm 6.8\text{e-}4$	$1.76\text{e-}2 \pm 8.3\text{e-}4$
abalone	5.51 ± 0.06	5.50 ± 0.05	5.48 ± 0.05	5.50 ± 0.05	5.50 ± 0.06	5.50 ± 0.06

For each training set, we choose the kernel parameter σ and regularization parameter λ by each criterion on the training set, and evaluate the test error for the chosen parameters on the test set.

TABLE 12
The Average Computational Time (in Seconds) for SVR with the Order of the Taylor Expansion $r = 5$

ϵ-SVR	5-CV	5-BIF	10-CV	10-BIF	20-CV	20-BIF
bodyfat	1.24 ± 0.02	2.16 ± 0.10	2.54 ± 0.09	2.42 ± 0.07	5.07 ± 0.12	3.03 ± 0.05
housing	2.96 ± 0.25	3.58 ± 0.17	6.02 ± 0.26	4.33 ± 0.08	12.11 ± 0.51	5.75 ± 0.08
mpg	2.26 ± 0.08	2.59 ± 0.09	4.49 ± 0.11	3.14 ± 0.08	8.87 ± 0.18	4.06 ± 0.09
pyrim	0.98 ± 0.10	0.84 ± 0.10	1.73 ± 0.09	1.06 ± 0.12	3.36 ± 0.13	1.66 ± 0.08
triazines	1.85 ± 0.04	1.91 ± 0.08	3.85 ± 0.08	2.23 ± 0.07	7.84 ± 0.24	2.70 ± 0.05
eunite2001	1.76 ± 0.10	2.47 ± 0.07	3.49 ± 0.10	2.83 ± 0.07	6.76 ± 0.18	3.61 ± 0.05
space-ga	32.73 ± 1.32	51.13 ± 2.35	75.61 ± 3.60	68.73 ± 2.16	161.55 ± 7.12	82.27 ± 2.18
cpusmall	17.07 ± 0.82	30.97 ± 0.36	36.99 ± 1.23	38.07 ± 0.60	77.71 ± 2.54	48.33 ± 0.58
mg	10.19 ± 0.29	15.03 ± 0.11	22.93 ± 0.37	20.10 ± 0.09	49.26 ± 0.80	28.33 ± 0.12
abalone	73.48 ± 1.70	105.92 ± 0.98	172.97 ± 3.73	138.51 ± 0.92	373.10 ± 7.02	184.35 ± 1.02

the same accuracy results as that of t-CV, meanwhile, significantly improves the efficiency on most data sets.

6.5 ϵ-SVR

In this experiment, we set $\epsilon = d$ and $h = 0.01\epsilon$, where d is standard error of the output y_i (in this experiment, we only report the results of $\epsilon = d$, similar results can be found with other values, e.g., $\epsilon \in \{2^i \cdot d, i = -4, -3 \ldots, 4\}$). The test mean square errors of t-BIF and t-CV are reported in Table 11. We can see that t-BIF gives the similar result with that of t-CV. We can also find that the performance of t-BIF for ϵ-SVR is a little worse than that for KRR. This may be caused by the replacement of $\max\big(0, |y - f_{\kappa,\mathbb{P}_S}(\mathbf{x})| - \epsilon\big)$ with Huber loss.

Although the ϵ-SVR is implemented in C++, from Table 12, we find that t-BIF gives the comparable results with that of t-CV for small t, and is much faster for large t.

The above results for LSSVM, L2-SVM, L1-SVM, KRR and ϵ-SVR demonstrate that the proposed approximate CV based on Bouligand influence function is sound and effective for their model selection.

7 Conclusion

In this paper, we develop a novel approximation theory of t-fold empirical cross-validation error based on the Bouligand influence function. We express the t-fold CVE by the BIF and higher order BIFs via Taylor expansions, and derive an upper bound of the discrepancy between the original and approximate t-fold CVEs with a very fast convergence rate $O(\frac{1}{r!t^r})$. We further give the BIF-based approximation of the t-fold CVEs for a wide variety of kernel-based algorithms, and propose an approximate CV using the

BIF-based approximation as model selection criteria instead of the t-fold CVE. Theoretical and experimental results show that the proposed approximate CV has sound theoretical foundation, high computational efficiency and wide application, and provides a new paradigm for model selection with cross-validation.

APPENDIX A
PROOF OF THEOREM 1

Proof. Let $f^r_{\kappa,\mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}} = f_{\kappa,\mathbb{P}_{\mathcal{S}}} + \sum_{j=1}^{r}\left(\frac{-1}{t-1}\right)^j \frac{\mathrm{BIF}_j(\mathbb{P}_{\mathcal{S}_i}; f_\kappa, \mathbb{P}_{\mathcal{S}})}{j!}$. From Equation (4), we have

$$\begin{aligned}
&|f_{\kappa,\mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}} - f^r_{\kappa,\mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}}| \\
&= \left|\sum_{j=r+1}^{\infty} \frac{\mathrm{BIF}_j(\mathbb{P}_{\mathcal{S}_i}; f_\kappa, \mathbb{P}_{\mathcal{S}})}{j!(1-t)^j}\right| \\
&\le \sum_{j=r+1}^{\infty} \frac{Q}{j!(t-1)^j} \le \frac{Q}{(r+1)!}\sum_{j=r+1}^{\infty}\frac{1}{(t-1)^j} \\
&= \frac{Q}{(r+1)!(t-1)^{r+1}}\sum_{j=0}^{\infty}\frac{1}{(t-1)^j} \\
&= \frac{Q}{(r+1)!(t-1)^{r+1}}\frac{1}{1-1/(t-1)} \\
&= \frac{Q}{(r+1)!(t-1)^r(t-2)}.
\end{aligned}$$

Note that $V(\cdot,\cdot)$ is C-Lipschitz continuous with respect to the second variable, thus we have

$$\begin{aligned}
&\left|V\left(y_j, f_{\kappa,\mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}}(\mathbf{x}_j)\right) - V\left(y_j, f^r_{\kappa,\mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}}(\mathbf{x}_j)\right)\right| \\
&\le C\left|f_{\kappa,\mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}} - f^r_{\kappa,\mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}}\right| \le \frac{CQ}{(r+1)!(t-1)^r(t-2)}.
\end{aligned}$$

This completes the proof of Theorem 1. □

APPENDIX B
PROOF OF THEOREM 2

Proof. From Theorem 2 in [52], we have

$$-2\lambda f_{\kappa,\mathbb{P}} = \mathbb{E}_{\mathbb{P}}[\ell'(y, f_{\kappa,\mathbb{P}}(\mathbf{x}))\Phi(\mathbf{x})] \tag{11}$$

Let $f_\epsilon = f_{\kappa,\mathbb{P}_{\epsilon,Q}}$. Note that $\mathbb{P}_{\epsilon,\mathbb{Q}} = (1-\epsilon)\mathbb{P} + \epsilon\mathbb{Q}$, hence we can obtain that

$$-2\lambda f_\epsilon = (1-\epsilon)\mathbb{E}_{\mathbb{P}}[\ell'(y, f_\epsilon(\mathbf{x})] + \epsilon\mathbb{E}_{\mathbb{Q}}[\ell'(y, f_\epsilon(\mathbf{x})]. \tag{12}$$

Taking the first derivative on both sides of (12) with respect to ϵ yields

$$\begin{aligned}
-2\lambda\frac{\partial}{\partial\epsilon}f_\epsilon = {}&(1-\epsilon)\mathbb{E}_{\mathbb{P}}\left[\left(\frac{\partial}{\partial\epsilon}f_\epsilon(\mathbf{x})\right)\ell''(y, f_\epsilon(\mathbf{x}))\Phi(\mathbf{x})\right] \\
&- \mathbb{E}_{\mathbb{P}}[\ell'(y, f_\epsilon(\mathbf{x}))\Phi(\mathbf{x})] \\
&+ \epsilon\mathbb{E}_{\mathbb{Q}}\left[\left(\frac{\partial}{\partial\epsilon}f_\epsilon(\mathbf{x})\right)\ell''(y, f_\epsilon(\mathbf{x}))\Phi(\mathbf{x})\right] \\
&+ \mathbb{E}_{\mathbb{Q}}[\ell'(y, f_\epsilon(\mathbf{x}))\Phi(\mathbf{x})].
\end{aligned} \tag{13}$$

Setting $\epsilon = 0$ and according to Equation (11), we have

$$\begin{aligned}
&2\lambda\frac{\partial}{\partial\epsilon}f_\epsilon|_{\epsilon=0} + \mathbb{E}_{\mathbb{P}}\left[\left(\frac{\partial}{\partial\epsilon}f_\epsilon(\mathbf{x})|_{\epsilon=0}\right)\ell''(y, f_{\kappa,\mathbb{P}}(\mathbf{x}))\Phi(\mathbf{x})\right] \\
&= \mathbb{E}_{\mathbb{P}}\left[\ell'(y, f_{\kappa,\mathbb{P}}(\mathbf{x}))\Phi(\mathbf{x})\right] - \mathbb{E}_{\mathbb{Q}}\left[\ell'(y, f_{\kappa,\mathbb{P}}(\mathbf{x}))\Phi(\mathbf{x})\right] \\
&= -2\lambda f_{\kappa,\mathbb{P}} - \mathbb{E}_{\mathbb{Q}}\left[\ell'(y, f_{\kappa,\mathbb{P}}(\mathbf{x}))\Phi(\mathbf{x})\right].
\end{aligned}$$

□

APPENDIX C
PROOF OF THE THEOREM 3

Proof. First we prove the following for all $2 \le k \in \mathbb{N}$:

$$\begin{aligned}
-2\lambda\frac{\partial}{\partial^k\epsilon}f_\epsilon = {}&(1-\epsilon)\mathbb{E}_{\mathbb{P}}\left[\left(\frac{\partial}{\partial^k\epsilon}f_\epsilon(\mathbf{x})\right)\ell''(y, f_\epsilon(\mathbf{x}))\Phi(\mathbf{x})\right] \\
&- k\mathbb{E}_{\mathbb{P}}\left[\left(\frac{\partial}{\partial^{k-1}\epsilon}f_\epsilon(\mathbf{x})+\right)\ell''(y, f_\epsilon(\mathbf{x}))\Phi(\mathbf{x})\right] \\
&+ k\mathbb{E}_{\mathbb{Q}}\left[\left(\frac{\partial}{\partial^{k-1}\epsilon}f_\epsilon(\mathbf{x})\right)\ell''(y, f_\epsilon(\mathbf{x}))\Phi(\mathbf{x})\right] \\
&+ \epsilon\mathbb{E}_{\mathbb{Q}}\left[\left(\frac{\partial}{\partial^k\epsilon}f_\epsilon(\mathbf{x})\right)\ell''(y, f_\epsilon(\mathbf{x}))\Phi(\mathbf{x})\right].
\end{aligned} \tag{14}$$

Taking the derivative on both sides of (13) with respect to ϵ yields

$$\begin{aligned}
-2\lambda\frac{\partial}{\partial^2\epsilon}f_\epsilon = {}&(1-\epsilon)\mathbb{E}_{\mathbb{P}}\left[\left(\frac{\partial}{\partial^2\epsilon}f_\epsilon(\mathbf{x})\right)\ell''(y, f_\epsilon(\mathbf{x}))\Phi(\mathbf{x})\right] \\
&- 2\mathbb{E}_{\mathbb{P}}\left[\left(\frac{\partial}{\partial\epsilon}f_\epsilon(\mathbf{x})\right)\ell''(y, f_\epsilon(\mathbf{x}))\Phi(\mathbf{x})\right] \\
&+ 2\epsilon\mathbb{E}_{\mathbb{Q}}\left[\left(\frac{\partial}{\partial^2\epsilon}f_\epsilon(\mathbf{x})\right)\ell''(y, f_\epsilon(\mathbf{x}))\Phi(\mathbf{x})\right] \\
&+ \mathbb{E}_{\mathbb{Q}}\left[\left(\frac{\partial}{\partial\epsilon}f_\epsilon(\mathbf{x})\right)\ell''(y, f_\epsilon(\mathbf{x}))\Phi(\mathbf{x})\right],
\end{aligned}$$

Thus for $k = 2$, the Equation (14) is satisfied. Taking the derivatives of both sides in (14),

$$\begin{aligned}
-2\lambda\frac{\partial}{\partial^{k+1}\epsilon}f_\epsilon = {}&(1-\epsilon)\mathbb{E}_{\mathbb{P}}\left[\left(\frac{\partial}{\partial^{k+1}\epsilon}f_\epsilon(\mathbf{x})\right)\ell''(y, f_\epsilon(\mathbf{x})+b_\epsilon)\Phi(\mathbf{x})\right] \\
&- (k+1)\mathbb{E}_{\mathbb{P}}\left[\left(\frac{\partial}{\partial^k\epsilon}f_\epsilon(\mathbf{x})\right)\ell''(y, f_\epsilon(\mathbf{x}))\Phi(\mathbf{x})\right] \\
&+ (k+1)\mathbb{E}_{\mathbb{Q}}\left[\left(\frac{\partial}{\partial^k\epsilon}f_\epsilon(\mathbf{x})\right)\ell''(y, f_\epsilon(\mathbf{x}))\Phi(\mathbf{x})\right] \\
&+ \epsilon\mathbb{E}_{\mathbb{Q}}\left[\left(\frac{\partial}{\partial^{k+1}\epsilon}f_\epsilon(\mathbf{x})\right)\ell''(y, f_\epsilon(\mathbf{x}))\Phi(\mathbf{x})\right],
\end{aligned}$$

from which it follows that (14) holds for $k+1$ indeed. Set $\epsilon = 0$:

$$\begin{aligned}
&2\lambda\frac{\partial}{\partial^{k+1}\epsilon}f_\epsilon|_{\epsilon=0} + \mathbb{E}_{\mathbb{P}}\left[\left(\frac{\partial}{\partial^{k+1}\epsilon}f_\epsilon(\mathbf{x})|_{\epsilon=0}\right)\ell''(y, f_{\kappa,\mathbb{P}}(\mathbf{x}))\Phi(\mathbf{x})\right] \\
&= (k+1)\mathbb{E}_{\mathbb{P}}\left[\left(\frac{\partial}{\partial^k\epsilon}f_\epsilon(\mathbf{x})|_{\epsilon=0}\right)\ell''(y, f_{\kappa,\mathbb{P}}(\mathbf{x}))\Phi(\mathbf{x})\right] \\
&\quad - (k+1)\mathbb{E}_{\mathbb{Q}}\left[\left(\frac{\partial}{\partial^k\epsilon}f_\epsilon(\mathbf{x})|_{\epsilon=0}\right)\ell''(y, f_{\kappa,\mathbb{P}}(\mathbf{x}))\Phi(\mathbf{x})\right].
\end{aligned}$$

□

ACKNOWLEDGMENTS

This work is supported in part by the National Natural Science Foundation of China (No.61703396, No.61673293, No.61602467), the Youth Innovation Promotion Association CAS, the Science and Technology Project of Beijing (Z181100002718004), the National Key Research and Development Program of China (2016YFB1000604), and the Excellent Talent Introduction of Institute of Information Engineering of CAS (Y7Z0111107).

REFERENCES

[1] J. Josse and F. Husson, "Selecting the number of components in principal component analysis using cross-validation approximations," *Comput. Statist. Data Anal.*, vol. 56, no. 6, pp. 1869–1879, 2012.
[2] F. Pedregosa, "Hyperparameter optimization with approximate gradient," in *Proc. 33th Int. Conf. Mach. Learn.*, 2016, pp. 737–746.
[3] W. Mao, X. Mu, Y. Zheng, and G. Yan, "Leave-one-out cross-validation-based model selection for multi-input multi-output support vector machine," *Neural Comput. Appl.*, vol. 24, no. 2, pp. 441–451, 2014.
[4] D. Allen, "The relationship between variable selection and data agumentation and a method for prediction," *Technometrics*, vol. 16, no. 1, pp. 125–127, 1974.
[5] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *J. Roy. Statistical Soc.. Series B (Methodological)*, vol. 36, pp. 111–147, 1974.
[6] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. 14th Int. Conf. Artif. Intell.*, 1995, pp. 1137–1143.
[7] Y. Bengio and Y. Grandvalet, "No unbiased estimator of the variance of k-fold cross-validation," *The J. Mach. Learn. Res.*, vol. 5, pp. 1089–1105, 2004.
[8] M. W. Seeger, "Cross-validation optimization for large scale structured classification kernel methods," *The J. Mach. Learn. Res.*, vol. 9, pp. 1147–1178, 2008.
[9] R. Kumar, D. Lokshtanov, S. Vassilvitskii, and A. Vattani, "Near-optimal bounds for cross-validation via loss stability," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 27–35.
[10] V. Vapnik, *The Nature of Statistical Learning Theory*. Berlin, Germany: Springer Verlag, 2000.
[11] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, 1999.
[12] C. S. A. Gammerman and V. Vovk, "Ridge regression learning algorithm in dual variables," in *Proc. 15th Int. Conf. Mach. Learn.*, 1998, pp. 515–521.
[13] B. Schölkopf and A. J. Smola, *Learning with Kernels*. Cambridge, MA, USA: MIT Press, 2002.
[14] E. Abbasnejad, D. Ramachandram, and R. Mandava, "A survey of the state of the art in learning the kernels," *Knowl. Inf. Syst.*, vol. 31, no. 2, pp. 193–221, 2012.
[15] Y. Liu, S. Liao, H. Lin, Y. Yue, and W. Wang, "Infinite kernel learning: Generalization bounds and algorithms," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 2280–2286.
[16] L.-Z. Ding and S. Liao, "An approximate approach to automatic kernel selection," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 554–565, Mar. 2017.
[17] G. C. Cawley and N. L. Talbot, "Preventing over-fitting during model selection via Bayesian regularisation of the hyper-parameters," *The J. Mach. Learn. Res.*, vol. 8, pp. 841–861, 2007.
[18] S. An, W. Liu, and S. Venkatesh, "Fast cross-validation algorithms for least squares support vector machine and kernel ridge regression," *Pattern Recognit.*, vol. 40, no. 8, pp. 2154–2162, 2007.
[19] Y. Liu and S. Liao, "Preventing over-fitting of cross-validation with kernel stability," in *Proc. Eur. Conf. Mach. Learn. Principles Practice Knowl. Discovery Databases*, 2014, pp. 290–305.
[20] A. Christmann and A. V. Messem, "Bouligand derivatives and robustness of support vector machines for regression," *The J. Mach. Learn. Res.*, vol. 9, pp. 915–936, 2008.
[21] Y. Liu, S. Jiang, and S. Liao, "Efficient approximation of cross-validation for kernel methods using Bouligand influence function," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 324–332.
[22] Y. Liu, H. Lin, L. Ding, W. Wang, and S. Liao, "Fast cross-validation," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 2497–2503.
[23] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Mach. Learn.*, vol. 46, no. 1/3, pp. 131–159, 2002.
[24] O. Chapelle and V. Vapnik, "Model selection for support vector machines," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 1999, pp. 230–236.
[25] V. Vapnik and O. Chapelle, "Bounds on error expectation for support vector machines," *Neural Comput.*, vol. 12, no. 9, pp. 2013–2036, 2000.
[26] G. Wahba, Y. Lin, and H. Zhang, "Generalized approximate cross-validation for support vector machines for support vector machines," in *Proc. Advances Large Margin Classifiers*, 2000, pp. 297–309.
[27] M. Opper and O. Winther, "Gaussian processes and SVM: Mean field and leave-one-out," in *Proc. Advances Large Margin Classifiers*, 1999, pp. 311–326.
[28] S. Keerthi, "Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms," *IEEE Trans. Neural Netw.*, vol. 13, no. 5, pp. 1225–1229, Sep. 2002.
[29] G. C. Cawley, "Leave-one-out cross-validation based model selection criteria for weighted LS-SVMs," in *Proc. Int. Joint Conf. Neural Netw.*, 2006, pp. 1661–1668.
[30] G. C. Cawley and N. L. Talbot, "Fast leave-one-out cross-validation of sparse least-squares support vector machines," *Neural Netw.*, vol. 17, no. 10, pp. 1467–1475, 2004.
[31] G. C. Cawley and N. L. Talbot, "Efficient approximate leave-one-out cross-validation for Kernel logistic regression," *Mach. Learn.*, vol. 71, no. 2/3, pp. 243–264, 2008.
[32] G. C. Cawley and N. L. Talbot, "Efficient leave-one-out cross-validation of kernel Fisher discriminant classifiers," *Pattern Recognit.*, vol. 36, no. 11, pp. 2585–2592, 2003.
[33] C. E. Rasmussen, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.
[34] S. Sundararajan and S. S. Keerthi, "Predictive approaches for choosing hyperparameters in Gaussian processes," *Neural Comput.*, vol. 13, no. 5, pp. 1103–1118, 2001.
[35] M. Debruyne, M. Hubert, and J. A. Suykens, "Model selection in Kernel based regression using the influence function," *The J. Mach. Learn. Res.*, vol. 9, pp. 2377–2400, 2008.
[36] M. Debruyne, "Robustness of censored depth quantiles, pca and kernel based regression, with new tools for model selection," Department of Mathematics, Ph.D. dissertation, Katholieke Universiteit Leuven, Leuven, Belgium, 2007.
[37] T. Pahikkala, J. Boberg, and T. Salakoski, "Fast n-fold cross-validation for regularized least-squares," in *Proc. 9th Scandinavian Conf. Artif. Intell.*, 2006, pp. 83–90.
[38] M. Seeger, "Cross-validation optimization for large scale hierarchical classification Kernel methods," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2006, pp. 1233–1240.
[39] A. Christmann and I. Steinwart, "On robustness properties of convex risk minimization methods for pattern recognition," *The J. Mach. Learn. Res.*, vol. 5, pp. 1007–1034, 2004.
[40] A. Christmann and I. Steinwart, "Consistency and robustness of kernel based regression," *Bernoulli*, vol. 13, pp. 799–819, 2007.
[41] A. Christmann and I. Steinwart, *Support Vector Machines*. Berlin, Germany: Springer Verlag, 2008.
[42] A. Christmann, A. V. Messem, and I. Steinwart, "On consistency and robustness properties of support vector machines for heavy-tailed distributions," *Statist. Interface*, vol. 2, pp. 311–327, 2009.
[43] H. Xu, C. Caramanis, and S. Mannor, "Robustness and regularization of support vector machines," *The J. Mach. Learn. Res.*, vol. 10, pp. 1485–1510, 2009.
[44] A. V. Messem and A. Christmann, "A review on consistency and robustness properties of support vector machines for heavy-tailed distributions," *Advances Data Anal. Classification*, vol. 4, no. 2/3, pp. 199–220, 2010.
[45] R. Hable and A. Christmann, "On qualitative robustness of support vector machines," *J. Multivariate Anal.*, vol. 102, no. 6, pp. 993–1007, 2011.
[46] A. Christmann and R. Hable, "Consistency of support vector machines using additive Kernels for additive models," *Comput. Statist. Data Anal.*, vol. 56, no. 4, pp. 854–873, 2012.
[47] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 1885–1894.

[48] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust Statistics: The Approach Based on Influence Functions*. New York, NY, USA: Wiley, 1986.
[49] S. M. Robinson, "An implicit-function theorem for a class of nonsmooth functions," *Math. Operations Res.*, vol. 16, pp. 292–309, 1991.
[50] O. Chapelle, "Training a support vector machine in the primal," *Neural Comput.*, vol. 19, no. 5, pp. 1155–1178, 2007.
[51] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, 2011, Art. no. 27.
[52] E. D. Vito, L. Rosasco, A. Caponnetto, M. Piana, and A. Verri, "Some properties of regularized Kernel methods," *The J. Mach. Learn. Res.*, vol. 5, pp. 1363–1390, 2004.

Yong Liu received the PhD degree in computer science from Tianjin University, Tianjin, China, in 2016. He is currently an associate researcher at Institute of Information Engineering, Chinese Academy of Sciences. His research interests include large-scale kernel methods, large-scale model selection, machine learning.

Shizhong Liao received the PhD degree in computer science from Tsinghua University, Beijing, China, in 1997. He is currently a professor at College of Intelligence and Computing, Tianjin University, Tianjin, China. His research interests include artificial intelligence and theoretical computer science.

Shali Jiang received the MS degree in computer science from Tianjin University, China, in 2015. He is currently working toward the PhD degree in computer science at Washington University in St. Louis. His current research interests are active learning, Bayesian optimization, Gaussian processes and kernel methods.

Lizhong Ding received the PhD degree in computer science from Tianjin University, Tianjin, China, in 2015. He is currently a research scientist at Inception Institute of Artificial Intelligence (IIAI), Abu Dhabi, UAE. He spent two years as a postdoctoral fellow at King Abdullah University of Science and Technology (KAUST), Saudi Arabia. His research interests include large-scale kernel methods, deep generative models, model selection and matrix analysis.

Hailun Lin received the PhD degree from the Institute of Computing Technology, Chinese Academy of Sciences, in 2015. She is currently an assistant professor at Institute of Information Engineering, Chinese Academy of Sciences. Her main research interests include open knowledge network, information extraction.

Weiping Wang received the PhD degree in computer science from the Harbin Institute of Technology, China, in 2008. He is currently a professor at Institute of Information Engineering, Chinese Academy of Sciences, National Engineering Research Center for Information Security, and National Engineering Laboratory for Information Security Technology. His research interests include database and storage systems.