Efficient Approximation of Cross-Validation for Kernel Methods using Bouligand Influence Function

Yong Liu YONGLIU@TJU.EDU.CN
Shali Jiang SLJIANG@TJU.EDU.CN
Shizhong Liao SZLIAO@TJU.EDU.CN
School of Computer Science and Technology, Tianjin University, Tianjin 300072, P. R. China

Abstract

Model selection is one of the key issues both in recent research and application of kernel methods. Cross-validation is a commonly employed and widely accepted model selection criterion. However, it requires multiple times of training the algorithm under consideration, which is computationally intensive. In this paper, we present a novel strategy for approximating the cross-validation based on the Bouligand influence function (BIF), which only requires the solution of the algorithm once. The BIF measures the impact of an infinitesimal small amount of contamination of the original distribution. We first establish the link between the concept of BIF and the concept of cross-validation. The BIF is related to the first order term of a Taylor expansion. Then, we calculate the BIF and higher order BIFs, and apply these theoretical results to approximate the cross-validation error in practice. Experimental results demonstrate that our approximate cross-validation criterion is sound and efficient.

1. Introduction

Kernel methods, such as SVM (Steinwart & Christmann, 2008; Vapnik, 2000), least squares support vector machine (LSSVM) (Suykens & Vandewalle, 1999) and support vector regression (SVR) (Shawe-Taylor & Cristianini, 2000), have been widely used in data mining and machine learning. The performance of these kernel methods greatly depends on the choice of some hyper-parameters (such as the kernel parameter and regularization parameter), therefore the model selection problem becomes an important topic in kernel methods. A related problem is the evaluation of the generalization ability of the learning algorithms. In fact, it is common to select the optimal hyper-parameters by choosing the ones with the lowest generalization error.

Obviously, the generalization error is not directly computable, as the probability distribution generating the data is unknown, therefore it is necessary to resort to estimates of its value. This error can be estimated either via testing on some data which has not been used for learning (hold-out validation or cross-validation techniques) or via a bound given by theoretical analysis (Chapelle et al., 2002). To establish the upper bounds of the generalization error, some measures are introduced: such as VC dimension (Vapnik, 2000), Rademacher complexity (Bartlett & Mendelson, 2002), maximal discrepancy (Bartlett et al., 2002), regularized risk (Schölkopf & Smola, 2002), radius-margin bound (Vapnik, 2000), compression coefficient (Luxburg et al., 2004) and eigenvalues perturbation (Liu et al., 2013).

While there have been many interesting attempts to use the above bounds or other techniques to pick the hyper-parameters, the most commonly used and widely accepted methods for selecting the hyper-parameters are still the k-fold cross-validation (KCV) and leave-one-out cross-validation (LOO). However, KCV and LOO requires the solution of the algorithm under consideration several times, which are computationally expensive. For the sake of efficiency, some approximate LOO criteria for some specific algorithms are given: such as generalized cross-validation (GCV) (Golub et al., 1979), influence function (Debruyne et al., 2008), generalized approximate cross-validation (GACV) (Wahba et al., 1999) and span bound (Chapelle et al., 2002).

In this paper, we will present a novel strategy for approximating the k-fold cross-validation based on the Bouligand influence function (BIF) (Christmann & Messem, 2008). To our knowledge, an effective strategy for approximating the k-fold cross-validation error (for all k) for kernel methods has never been given before. We establish the link

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32. Copyright 2014 by the author(s).

between the concept of BIF and the concept of KCV, and present a novel method to calculate the BIF and higher order BIFs at the continuous distribution. Furthermore, we evaluate these BIFs at the sample distribution and use these BIFs to obtain an approximation of KCV. Our method requires the solution of the algorithm only once, which can dramatically improve the efficiency. Experimental results demonstrate that our BIF criterion is a good choice for model selection.

Related Work

In recent years, some researchers study the robustness of the kernel methods. In the field of robust statistics the influence function (IF) (Hampel et al., 1986) is introduced in order to analyze the effects of outliers on the algorithm. This influence function is defined for continuous distributions that are slightly perturbed by adding a small amount of probability mass at a certain place. Christmann and Steinwart (Christmann & Steinwart, 2004; 2007), Steinwart and Christmann (Steinwart & Christmann, 2008), Christmann et al (Christmann et al., 2009), and Messem and Christmann (Messem & Christmann, 2010) show that SVMs for classification and regression have a bounded influence function under some assumptions of the loss function. Debruyne et al (Debruyne et al., 2008) presented a method to estimate the LOO via the influence function. Christmann and Messem (Christmann & Messem, 2008) generalize the notion of influence function, and introduce a new notion from Bouligand-derivatives (Robinson, 1991) called Bouligand influence function (BIF), which measures the impact of an infinitesimal small amount of contamination of the original distribution. They show that SVMs have a bounded BIF with some weaker assumptions of loss function.

For kernel methods, such as SVM, LSSVM and SVR, the form of the decision function is $f(\boldsymbol{x}) = \sum_i \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b$. The above work about the robust statistics of kernel methods all ignore the bias b. However, sometimes the bias b plays an important role in the performance of kernel methods. In this paper, we consider the b, and present a theoretical result to calculate the BIF at the continuous distribution. This result generalizes the result of Christmann and Messem (Christmann & Messem, 2008) with a much simpler proof. Debruyne et al (Debruyne et al., 2008) present a method to calculate the higher order IFs, and apply these results to approximate the LOO. We generalize the results of IFs to BIFs, and apply these results of BIFs to approximate the cross-validation error.

The rest of the paper is organized as follows. In Section 2, we introduce some elementary facts. In Section 3, we introduce the concept of BIF, and give a novel strategy for approximating the cross-validation error. A method to calculate the BIF and higher order BIFs is proposed in Section 4. In Section 5, we show how to use these BIFs to approximate the cross-validation estimator in practice. We empirically analyze the performance of our proposed approximate cross-validation criterion in Section 6. We end in Section 7 with conclusion.

2. Preliminaries

Let $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ be a sample set of size n drawn identically and independently from a fixed, but unknown probability measure P on $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, $\mathcal{X} \subseteq \mathbb{R}^d$, $\mathcal{Y} \subseteq \mathbb{R}$ for regression, and $\mathcal{Y} \subseteq \{+1, -1\}$ for classification. Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel, that is, K is symmetric and for any finite set of points $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subset \mathcal{X}$, the kernel matrix $\boldsymbol{K} = [K(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j=1}^m$ is positive semidefinite. The reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ associated with the kernel K is defined to be the completion of the linear span of the set of functions $\{\Phi(\boldsymbol{x}) = K(\boldsymbol{x}, \cdot) : \boldsymbol{x} \in \mathcal{X}\}$ with the inner product denoted as $\langle \cdot, \cdot \rangle_K$ satisfying $\langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{x}') \rangle_K = K(\boldsymbol{x}, \boldsymbol{x}')$ (Aronszajn, 1950).

The operator $f_{\lambda,K} + b_{\lambda,K} : P \to f_{\lambda,K,P} + b_{\lambda,K,P}$ is defined by $f_{\lambda,K,P} + b_{\lambda,K,P} =$

$$\arg\min_{f \in \mathcal{H}, b \in \mathbb{R}} \mathbb{E}_P V(y - f(\boldsymbol{x}) - b) + \lambda \|f\|_K^2,$$

where $V(\cdot)$ is a loss function and λ is the regularization parameter. When the sample distribution P_n is used, one has that $f_{\lambda,K,P_n} + b_{\lambda,K,P_n} =$

$$\arg\min_{f \in \mathcal{H}, b \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^n V(y_i - f(\boldsymbol{x}_i) - b) + \lambda \|f\|_K^2.$$

Such estimators have been studied in detail, see for example (Wahba, 1990; Vapnik, 2000).

LSSVM (Suykens & Vandewalle, 1999; Cawley & Talbot, 2007), ϵ-insensitive support vector regression (ϵ-SVR) (Shawe-Taylor & Cristianini, 2000) and quadratic ϵ-insensitive support vector regression (quadratic ϵ-SVR) (Shawe-Taylor & Cristianini, 2000) are only different in the choice of the loss function. For LSSVM, $V(r) = r^2$, for ϵ-SVR, $V(r) = \max\{|r| - \epsilon, 0\}$, and for quadratic ϵ-SVR, $V(r) = (\max\{|r| - \epsilon, 0\})^2$.

Unless specially stated, we respectively write $f_{\lambda,K,P}$ and $b_{\lambda,K,P}$ as f_P and b_P in the following.

3. A Strategy for Fast Approximation of Cross Validation

In this section, we introduce the Bouligand influence function (BIF) (Christmann & Messem, 2008) and higher order BIFs, and show how to use these BIFs to approximate the k-fold cross-validation (KCV).

3.1. Bouligand Influence Function

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

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

The BIF measures the impact of an infinitesimal small amount of contamination of the original distribution P in the direction of Q on the quantity of $T(P)$.

Denote $P_{\epsilon,Q} = (1-\epsilon)P + \epsilon Q$. One can see that the BIF is a first order derivative of $T(P_{\epsilon,Q})$ at $\epsilon = 0$. Higher order BIFs can be defined too:

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

$$BIF_k(Q;T,P) = \frac{\partial}{\partial^k \epsilon} T(P_{\epsilon,Q})|_{\epsilon=0}.$$

If all BIFs exist then the following Taylor expansion holds:

$$T(P_{\epsilon,Q}) = T(P) + \sum_{i=1}^{\infty} \frac{\epsilon^i}{i!} BIF_i(Q;T,P). \tag{1}$$

3.2. A Strategy for Approximating the KCV using BIF

Assume the sample set $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ is divided into k disjoint parts $\{S_i\}_{i=1}^k$. Let $P_n^{-S_i}$ be the empirical distribution of the sample S without the observations S_i, that is $P_n^{-S_i}(\boldsymbol{x}) = \frac{1}{n-M}$ if $\boldsymbol{x} \in S \setminus S_i$, otherwise 0, where M is the size of S_i.

For k-fold cross-validation, the $T(P_n^{-S_i})$ should be computed for every i. This means that the algorithm under consideration has to be executed k times, which is computationally intensive.

If the BIFs of T can be calculated, we can provide a fast alternative. First note that

$$P_n^{-S_i} = \left(1 - \left(\frac{-M}{n-M}\right)\right) P_n + \frac{-M}{n-M}\Delta_{S_i},$$

where Δ_{S_i} is the sample distribution corresponding to the sample S_i, that is, $\Delta_{S_i}(\boldsymbol{x}) = \frac{1}{M}$ if $\boldsymbol{x} \in S_i$, otherwise 0. Thus, taking $Q = \Delta_{S_i}$, $\epsilon = -\frac{M}{n-M}$, $P_{\epsilon,Q} = P_n^{-S_i}$, $P = P_n$ and $T = f_{\lambda,K} + b_{\lambda,K}$, Equation (1) gives

$$f_{P_n^{-S_i}} + b_{P_n^{-S_i}} = f_{P_n} + b_{P_n} + \sum_{j=1}^{\infty} \left(\frac{-M}{n-M}\right)^j \frac{BIF_j(\Delta_{S_i}; f_{\lambda,K} + b_{\lambda,K}, P_n)}{j!}. \tag{2}$$

The right hand side now only depends on the full sample P_n and Δ_{S_i}. Given the $BIF_j(\Delta_{S_i}; f_{\lambda,K} + b_{\lambda,K}, P_n)$, the k-fold cross validation error can be written as

$$k\text{-}CV = \frac{1}{n}\sum_{i=1}^{k}\sum_{\boldsymbol{x}_j\in S_i} \ell\Big(y_j, f_{P_n} + b_{P_n} + \sum_{p=1}^{\infty}\left(\frac{-M}{n-M}\right)^p \frac{BIF_p(\Delta_{S_i}; f_{\lambda,K} + b_{\lambda,K}, P_n)}{p!}\Big),$$

where $\ell(\cdot,\cdot)$ is an appropriate loss function. It only requires the solution of the algorithm once.

Note that $-M/(n-M) = -1/(k-1)$, the $\left|\frac{(-1)^p}{(k-1)^p p!}\right|$ is very small for some large p. Thus, we can take the low order approximation of the Taylor expansion to effectively approximate the k-fold cross-validation:

$$k\text{-}CV \approx \frac{1}{n}\sum_{i=1}^{k}\sum_{\boldsymbol{x}_j\in S_i} \ell\Big(y_j, f_{P_n} + b_{P_n} + \sum_{p=1}^{r}\left(\frac{-M}{n-M}\right)^p \frac{BIF_p(\Delta_{S_i}; f_{\lambda,K} + b_{\lambda,K}, P_n)}{p!}\Big).$$

Remark 1. *In our experiments, when the order of Taylor expansion $r \geq 3$, we find that the value of the approximate cross-validation error is almost the same as original one.*

4. The Calculation of BIFs

In this section, we first provide a novel method to calculate the BIF and higher order BIFs at the continuous distribution P, and then estimate these BIFs at the specific sample distribution P_n.

4.1. The Calculation of BIFs at Continuous Distribution

By the definition of the k-th order BIF of $f_{\lambda,K} + b_{\lambda,K}$, $k = 1, 2, \ldots$, it is easy to verify that

$$BIF_k(Q; f_{\lambda,K} + b_{\lambda,K}, P) = \frac{\partial}{\partial^k \epsilon} f_{P_{\epsilon,Q}}|_{\epsilon=0} + \frac{\partial}{\partial^k \epsilon} b_{P_{\epsilon,Q}}|_{\epsilon=0}.$$

Let $V_P = V(y - f_P(\boldsymbol{x}) - b_P))$, the first order BIF at the P will be given in the following theorem.

Theorem 1. *Let $\mathcal{H}$ be the RKHS of a bounded continuous kernel K on $\mathcal{X}$. Furthermore, let P be a distribution on $\mathcal{X} \times \mathcal{Y}$, then the BIF of $f_{\lambda,K} + b_{\lambda,K}$ in the direction of a distribution $Q \neq P$ is*

$$[\frac{\partial}{\partial\epsilon} f_{P_{\epsilon,Q}}|_{\epsilon=0}, \frac{\partial}{\partial\epsilon} b_{P_{\epsilon,Q}}|_{\epsilon=0}] = L^{-1}[-2\lambda f_P + \mathbb{E}_Q(V_P'\Phi(\boldsymbol{x})), \mathbb{E}_Q V_P'],$$

where the operator $L : (\mathcal{H}, \mathbb{R}) \to (\mathcal{H}, \mathbb{R})$ is defined by

$$L(f,b) = \Big[2\lambda f + \mathbb{E}_P(V_P'' f(\boldsymbol{x})\Phi(\boldsymbol{x})) + b\mathbb{E}_P(V_P''\Phi(\boldsymbol{x})), \\ \mathbb{E}_P(V_P'' f(\boldsymbol{x})) + b\mathbb{E}_P(V_P'')\Big].$$

The proof is given in Appendix A.

Remark 2. *The first order BIF of the decision function without the bias term ($b_{\lambda,K} = 0$) has been given in (Christmann & Messem, 2008). Our above theorem generalizes their result. Moreover, our proof is much simpler.*

The higher order BIF is given in the following theorem:

Theorem 2. *Let $\mathcal{H}$ be the RKHS of a bounded continuous kernel K on $\mathcal{X}$. Let V be a convex loss function such that the third derivative is 0. Furthermore, let P be a distribution on $\mathcal{X} \times \mathcal{Y}$, then the $(k+1)$ order BIF of $f_{\lambda,K} + b_{\lambda,K}$ in the direction of a distribution $Q \neq P$ is*

$$\Big[\frac{\partial}{\partial^{k+1}\epsilon} f_{P_{\epsilon,Q}}|_{\epsilon=0}, \frac{\partial}{\partial^{k+1}\epsilon} b_{P_{\epsilon,Q}}|_{\epsilon=0}\Big] = \\ (k+1)L^{-1}\Big[2\mathbb{E}_P(BIF_k(Q; f_{\lambda,K}+b_{\lambda,K}, P)V_P''(\Phi(\boldsymbol{x}))) - \\ \mathbb{E}_Q(BIF_k(Q; f_{\lambda,K}+b_{\lambda,K}, P)V_P''\Phi(\boldsymbol{x})), \\ \mathbb{E}_P(BIF_k(Q; f_{\lambda,K}+b_{\lambda,K}, P)V_P'') - \\ \mathbb{E}_Q(BIF_k(Q; f_{\lambda,K}+b_{\lambda,K}, P)V_P'')\Big].$$

where the operator $L : (\mathcal{H}, \mathbb{R}) \to (\mathcal{H}, \mathbb{R})$ is defined by

$$L(f,b) = \Big[2\lambda f + \mathbb{E}_P(V_P'' f(\boldsymbol{x})\Phi(\boldsymbol{x})) + b\mathbb{E}_P(V_P''\Phi(\boldsymbol{x})), \\ \mathbb{E}_P(V_P'' f(\boldsymbol{x})) + b\mathbb{E}_P(V_P'')\Big].$$

The proof is given in Appendix B.

Remark 3. *For the common loss function V, such as $V(r) = r^2$ and $V(r) = (\max(|r| - \epsilon, 0)^2$, the third derivative is 0. Thus, the assumption of the above Theorem is feasible.*

4.2. The Calculation of BIFs at the Sample Distribution

In this subsection, we will estimate the BIF at the sample distribution P_n to obtain $BIF_j(\Delta_{S_i}; f_{\lambda,K} + b_{\lambda,K}, P_n)$.

4.2.1. LSSVM

First consider taking the least squares loss $V(r) = r^2$. From Theorem 1, the operator L at P_n maps any $(f,b) \in (\mathcal{H}, \mathbb{R})$ to

$$L(f,b) = \Big[2\lambda f + \frac{2}{n}\sum_{j=1}^n f(\boldsymbol{x}_j)\Phi(\boldsymbol{x}_j) + \frac{2b}{n}\sum_{j=1}^n \Phi(\boldsymbol{x}_j), \\ \frac{2}{n}\sum_{j=1}^n f(\boldsymbol{x}_j) + 2b\Big].$$

Denote $\boldsymbol{f} = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n))^{\mathrm{T}}$, $\mathbf{1} = (1, \ldots, 1)^{\mathrm{T}}$, kernel matrix $\boldsymbol{K} = [K(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j=1}^n$. Note that

$$\begin{bmatrix} L(f,b)(\boldsymbol{x}_1) \\ \vdots \\ L(f,b)(\boldsymbol{x}_n) \end{bmatrix} = 2\begin{bmatrix} \lambda\boldsymbol{I}_n + \frac{1}{n}\boldsymbol{K} & \frac{1}{n}\boldsymbol{K}\mathbf{1} \\ \frac{1}{n}\mathbf{1}^{\mathrm{T}} & 1 \end{bmatrix}\begin{bmatrix} \boldsymbol{f} \\ b \end{bmatrix},$$

which means that the matrix

$$2\boldsymbol{L}_n := 2\begin{bmatrix} \lambda\boldsymbol{I}_n + \frac{1}{n}\boldsymbol{K} & \frac{1}{n}\boldsymbol{K}\mathbf{1} \\ \frac{1}{n}\mathbf{1}^{\mathrm{T}} & 1 \end{bmatrix}$$

is the finite sample version of the operator L at P_n. Denote

$$\frac{\partial}{\partial^k\epsilon}\boldsymbol{f}_{P_{\epsilon,\Delta_{S_i}}}|_{\epsilon=0} \\ = (\frac{\partial}{\partial^k\epsilon} f_{P_{\epsilon,\Delta_{S_i}}}(\boldsymbol{x}_1)|_{\epsilon=0}, \ldots, \frac{\partial}{\partial^k\epsilon} f_{P_{\epsilon,\Delta_{S_i}}}(\boldsymbol{x}_n)|_{\epsilon=0})^{\mathrm{T}}.$$

From Theorem 1, it is now clear that

$$\begin{bmatrix} \frac{\partial}{\partial\epsilon}\boldsymbol{f}_{P_{\epsilon,\Delta_{S_i}}}|_{\epsilon=0} \\ \frac{\partial}{\partial\epsilon} b_{P_{\epsilon,\Delta_{S_i}}}|_{\epsilon=0} \end{bmatrix} = \boldsymbol{L}_n^{-1}\begin{bmatrix} \frac{1}{M}[\boldsymbol{K} \bullet \boldsymbol{S}_i]\boldsymbol{g} - \lambda\boldsymbol{f}_{P_n} \\ \frac{1}{M}\boldsymbol{g}_{S_i}^{\mathrm{T}}\mathbf{1} \end{bmatrix}$$

where $\boldsymbol{g} = (g_1, \ldots, g_n)^{\mathrm{T}}, g_i = y_i - f_{P_n}(\boldsymbol{x}_i) - b_{P_n}$, $\boldsymbol{g}_{S_i} = (g_{S_i,1}, \ldots, g_{S_i,n})^{\mathrm{T}}, g_{S_i,j} = g_j$ if $\boldsymbol{x}_j \in S_i$, otherwise 0, $\boldsymbol{f}_{P_n} = (f_{P_n}(\boldsymbol{x}_1), \ldots, f_{P_n}(\boldsymbol{x}_n))^{\mathrm{T}}$, $\boldsymbol{S}_i$ denote the $n \times n$ matrix as $[\boldsymbol{S}_i]_{j,k} = 1$ if $\boldsymbol{x}_k \in S_i$, otherwise 0, and $\bullet$ is the entrywise matrix product (also known as the Hadamard product).

From Theorem 2, one sees similarly that the higher order terms can be computed

$$\begin{bmatrix} \frac{\partial}{\partial^{k+1}\epsilon}\boldsymbol{f}_{P_{\epsilon,\Delta_{S_i}}}|_{\epsilon=0} \\ \frac{\partial}{\partial^{k+1}\epsilon} b_{P_{\epsilon,\Delta_{S_i}}}|_{\epsilon=0} \end{bmatrix} = \\ (k+1)\boldsymbol{L}_n^{-1}\begin{bmatrix} \frac{1}{n}\boldsymbol{K}\boldsymbol{b}_k - \frac{1}{M}\boldsymbol{K} \bullet \boldsymbol{S}_i\boldsymbol{b}_k \\ \frac{1}{n}\mathbf{1}^{\mathrm{T}}\boldsymbol{b}_k - \frac{1}{M}\mathbf{1}^{\mathrm{T}}\boldsymbol{b}_{k,S_i} \end{bmatrix},$$

where

$$\boldsymbol{b}_k = \Big(BIF_k(\Delta_{S_i}; f_{\lambda,K}+b_{\lambda,K}, P_n)(\boldsymbol{x}_1)), \ldots, \\ BIF_k(\Delta_{S_i}; f_{\lambda,K}+b_{\lambda,K}, P_n)(\boldsymbol{x}_n)\Big)^{\mathrm{T}}, \\ BIF_k(\Delta_{S_i}; f_{\lambda,K}+b_{\lambda,K}, P_n)(\boldsymbol{x}_j) = \\ \frac{\partial}{\partial^k\epsilon} f_{\epsilon,\Delta_{S_i}}(\boldsymbol{x}_j)|_{\epsilon=0} + \frac{\partial}{\partial^k\epsilon} b_{\epsilon,\Delta_{S_i}}(\boldsymbol{x}_j)|_{\epsilon=0},$$

$\boldsymbol{b}_{k,S_i} = (b_{k,S_i,1}, \ldots, b_{k,S_i,n})^{\mathrm{T}}$, $b_{k,S_i,j} = b_{k,j}$ if $\boldsymbol{x}_j \in S_i$, otherwise 0.

For the k-fold cross-validation, define $[BIFMLSSVM_t]$ the $k \times n$ matrix with

$$[BIFMLSSVM_t]_{i,j} = BIF_t(\Delta_{S_i}; f_{\lambda,K}+b_{\lambda,K}, P_n)(\boldsymbol{x}_j).$$

According to Equation (2), by cutting it off at some step r, we have

$$f_{P_n^{-S_i}}(\boldsymbol{x}_j) + b_{P_n^{-S_i}} \approx f_{P_n}(\boldsymbol{x}_j) + b_{P_n} + \sum_{s=1}^{r} \left(\frac{-1}{k-1}\right)^s \frac{1}{s!}[BIFMLSSVM_s]_{i,j}. \tag{3}$$

4.2.2. QUADRATIC ϵ-SVR

For the quadratic ϵ-insensitive loss we have that

$$V(r) = \begin{cases} 0, & \text{if } |r| \le \epsilon \\ (r-\epsilon)^2, & \text{if } |r| > \epsilon \end{cases}$$

and thus $V'(r) = \begin{cases} 0, & \text{if } |r| < \epsilon \\ 2(r-\epsilon), & \text{if } |r| > \epsilon \end{cases}, V''(r) = \begin{cases} 0, & \text{if } |r| < \epsilon \\ 2, & \text{if } |r| > \epsilon. \end{cases}$ Note that the derivatives in $r = \epsilon$ do not exist, but in practice the probability that $r = \epsilon$ is 0, so we can ignore this possibility.

Similar with the least squares loss, it is easy to verify that

$$\boldsymbol{L}_n := \begin{bmatrix} 2\lambda \boldsymbol{I}_n + \frac{1}{n}[\boldsymbol{K} \bullet \boldsymbol{B}] & [\boldsymbol{K} \bullet \boldsymbol{B}]\boldsymbol{1} \\ \frac{1}{n}\boldsymbol{v}^{\mathrm{T}} & \boldsymbol{v}^{\mathrm{T}}\boldsymbol{1} \end{bmatrix}$$

is the finite sample version of the operator L at sample P_n, where $\boldsymbol{B}$ denote the matrix containing $V''(y_i - f_{P_n}(\boldsymbol{x}_i) - b_{P_n})$ at every entry in the i-th column, and $\boldsymbol{v} = (v_1, \ldots, v_n)^{\mathrm{T}}, v_i = V''(y_i - f_{P_n}(\boldsymbol{x}_i) - b_{P_n})$.

From Theorem 1, we have

$$\begin{bmatrix} \frac{\partial}{\partial \epsilon} \boldsymbol{f}_{P_{\epsilon,\Delta_{S_i}}}|_{\epsilon=0} \\ \frac{\partial}{\partial \epsilon} b_{P_{\epsilon,\Delta_{S_i}}}|_{\epsilon=0} \end{bmatrix} = \boldsymbol{L}_n^{-1} \begin{bmatrix} \frac{1}{M}\boldsymbol{K} \bullet \boldsymbol{S}_i\boldsymbol{u} - \lambda \boldsymbol{f}_{P_n} \\ \frac{1}{M}\boldsymbol{u}_{S_i}^{\mathrm{T}}\boldsymbol{1} \end{bmatrix}$$

where $\boldsymbol{u} = (u_1, \ldots, u_n), u_i = V'(y_i - f_{P_n}(\boldsymbol{x}_i) - b_{P_n})$, $\boldsymbol{u}_{S_i} = (u_{S_i,1}, \ldots, u_{S_i,n}), u_{S_i,j} = u_j$ if $\boldsymbol{x}_j \in S_i$, otherwise 0. By Theorem 2, the higher order terms can be computed,

$$\begin{bmatrix} \frac{\partial}{\partial \epsilon^{k+1}} \boldsymbol{f}_{P_{\epsilon,\Delta_{S_i}}}|_{\epsilon=0} \\ \frac{\partial}{\partial \epsilon^{k+1}} b_{P_{\epsilon,\Delta_{S_i}}}|_{\epsilon=0} \end{bmatrix} = (k+1) S_n^{-1} \begin{bmatrix} \frac{1}{n}[\boldsymbol{K} \bullet \boldsymbol{B}]\boldsymbol{b}_k - \frac{1}{M}\boldsymbol{K} \bullet \boldsymbol{B} \bullet \boldsymbol{S_i}\boldsymbol{b}_k \\ \frac{1}{n}\boldsymbol{v}^{\mathrm{T}}\boldsymbol{b}_k - \frac{1}{M}\boldsymbol{v}_{S_i}^{\mathrm{T}}\boldsymbol{b}_k \end{bmatrix}.$$

where $\boldsymbol{v} = (v_1, \ldots, v_n)^{\mathrm{T}}, v_i = V''(y_i - f_{P_n}(\boldsymbol{x}_i) - b_{P_n})$, $v_{S_i,j} = v_j$ if $\boldsymbol{x} \in S_i$, otherwise 0.

For the k-fold cross-validation, let $[BIFMSVR_t]$ be the $k \times n$ matrix with

$$[BIFMSVR_t]_{i,j} = BIF_t(\Delta_{S_i}; f_{\lambda,K} + b_{\lambda,K}, P_n)(\boldsymbol{x}_j).$$

From Equation (2), we have

$$f_{P_n^{-S_i}}(\boldsymbol{x}_j) + b_{P_n^{-S_i}} \approx f_{P_n}(\boldsymbol{x}_j) + b_{P_n} + \sum_{s=1}^{r} \left(\frac{-1}{k-1}\right)^s \frac{1}{s!}[BIFMSVR_s]_{i,j}. \tag{4}$$

5. Approximate KCV Criteria

The traditional k-fold cross-validation error is given by

$$k\text{CV} = \frac{1}{n}\sum_{i=1}^{k}\sum_{\boldsymbol{x}_j \in S_i} \ell(y_j, f_{P_n^{-S_i}}(\boldsymbol{x}_j) + b_{P_n^{-S_i}}),$$

where $\ell(\cdot,\cdot)$ is an appropriate loss function. The idea we investigate is to replace the explicit k-fold cross-validation by the approximation in (3) for LSSVM and (4) for quadratic ϵ-SVR.

The t-th order BIF criterion of the approximate k-fold cross-validation error for LSSVM is defined as

$$BIF_k^t = \frac{1}{n}\sum_{i=1}^{k}\sum_{\boldsymbol{x}_j \in S_i} \ell\Big(y_j, f_{\lambda,K,P_n}(\boldsymbol{x}_j) + b_{\lambda,K,P_n} + \sum_{s=1}^{t} \left(\frac{-1}{k-1}\right)^s \frac{1}{s!}[BIFMLSSVM_s]_{i,j}\Big).$$

For quadratic ϵ-SVR:

$$\epsilon\text{-}BIF_k^t = \frac{1}{n}\sum_{i=1}^{k}\sum_{\boldsymbol{x}_j \in S_i} \ell\Big(y_j, f_{\lambda,K,P_n}(\boldsymbol{x}_j) + b_{\lambda,K,P_n} + \sum_{s=1}^{t} \left(\frac{-1}{k-1}\right)^s \frac{1}{s!}[BIFMSVR_s]_{i,j}\Big).$$

5.1. Time Complexity Analysis

To compute BIF_k^t and $\epsilon\text{-}BIF_k^t$, we need $O(n^3)$ to calculate the inversion of $\boldsymbol{L}_n$[1], and $O(kn^2 + tn^2)$ to calculate the BIF matrices, where n is size of the training set, k is the fold of cross-validation and t is the order of the Taylor expansion. Thus, the overall time complexity of BIF_k^t and $\epsilon\text{-}BIF_k^t$ are both $O(n^3 + kn^2 + tn^2)$.

For the traditional k-fold cross-validation method, the algorithm under consideration need to be executed k times, thus for LSSVM and quadratic ϵ-SVR the time complexity are both $O(kn^3)$.

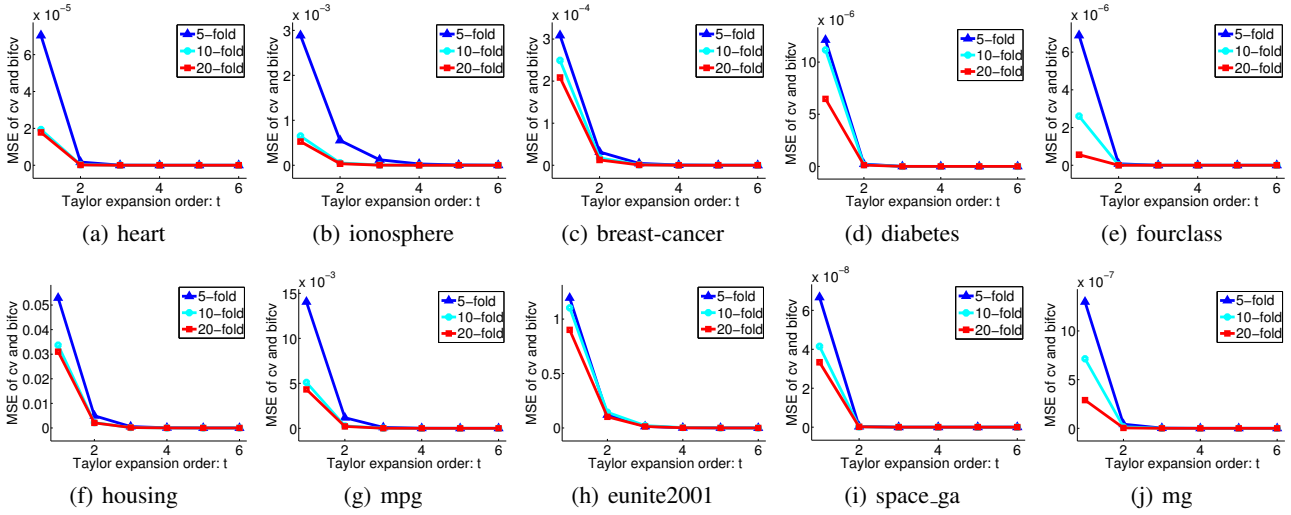
6. Experiments

In this section, we will empirically analyze the performance of our proposed approximate k-fold cross-validation criterion (BIF-kCV).

[1] If $\boldsymbol{L}_n$ is not invertible, we can use the pseudo-inverse of $\boldsymbol{L}_n$.

Table 1. The average testing errors (%) on the classification data sets and the testing mean square error (MSE) on regression data sets, the order of Taylor expansion $t = 3$.

Classification	EP	ELOO	5CV	BIF-5CV	10CV	BIF-10CV	20CV	BIF-20CV
ionosphere	14.74 ± 3.97	6.65 ± 1.47	7.16 ± 1.54	8.18 ± 1.54	7.16 ± 2.07	7.61 ± 1.69	8.18 ± 1.54	8.07 ± 1.68
breast	3.58 ± 0.38	3.07 ± 0.59	3.45 ± 0.81	3.45 ± 0.81	3.45 ± 0.81	3.45 ± 0.81	3.45 ± 0.81	3.45 ± 0.81
diabetes	24.22 ± 1.67	23.83 ± 1.69	22.24 ± 2.47	22.24 ± 2.47	22.66 ± 2.23	22.66 ± 2.23	22.50 ± 2.18	22.50 ± 2.18
fourclass	22.87 ± 0.98	19.49 ± 2.03	18.19 ± 3.32	18.19 ± 3.32	18.19 ± 3.32	18.19 ± 3.32	17.12 ± 2.28	17.12 ± 2.28
australian	13.51 ± 1.38	14.29 ± 1.81	15.19 ± 2.18	15.19 ± 2.18	14.09 ± 1.96	14.09 ± 1.96	14.49 ± 2.35	14.49 ± 2.35
heart	18.96 ± 3.08	19.70 ± 4.19	16.56 ± 3.35	17.41 ± 1.69	16.15 ± 3.33	17.85 ± 2.25	16.15 ± 2.98	17.59 ± 3.07
german	25.84 ± 2.84	26.38 ± 2.31	25.52 ± 1.45	25.52 ± 1.45	25.28 ± 1.38	25.28 ± 1.38	25.28 ± 1.38	25.28 ± 1.38
liver	39.42 ± 4.06	31.39 ± 3.71	29.71 ± 1.86	29.71 ± 1.86	29.25 ± 2.73	31.21 ± 1.29	31.10 ± 3.43	31.10 ± 3.43
sonar	17.12 ± 2.39	16.15 ± 3.65	16.92 ± 4.49	17.88 ± 2.08	17.12 ± 4.58	18.62 ± 2.45	16.92 ± 4.69	17.32 ± 2.45
a2a	20.38 ± 1.68	18.90 ± 1.01	18.98 ± 0.95	18.98 ± 0.95	19.10 ± 0.96	19.10 ± 0.96	19.10 ± 1.05	19.10 ± 1.05
Regression	**EP**	**ELOO**	**5CV**	**BIF-5CV**	**10CV**	**BIF-10CV**	**20CV**	**BIF-20CV**
bodyfat	5.1e-5± 3.1e-5	3.9e-5 ± 9.9e-6	4.5e-5 ± 1.4e-5	4.5e-5 ± 1.4e-5	4.5e-5 ± 1.4e-5	4.5e-5 ± 1.4e-5	4.5e-5 ± 1.4e-5	4.5e-5 ± 1.4e-5
housing	31.3 ± 6.4	24.3 ± 3.4	23.9 ± 3.8	23.9 ± 3.8	23.98 ± 3.8	23.9 ± 3.8	23.9 ± 3.8	23.9 ± 3.8
mpg	12.4 ± 2.2	9.6 ± 1.5	8.7 ± 0.8	8.7 ± 0.8	8.7 ± 0.8	8.6 ± 0.8	8.6 ± 0.8	8.6 ± 0.8
pyrim	1.2e-2 ± 4.0e-3	1.4e-2 ± 4.2e-3	1.0e-2 ± 2.9e-3	1.1e-2 ± 2.4e-3	1.0e-2 ± 2.9e-3	1.1e-2 ± 2.4e-3	1.0e-2 ± 2.9e-3	1.1e-2 ± 2.1e-3
triazines	2.0e-2 ± 2.9e-3	2.2e-2 ± 3.3e-3	2.3e-2 ± 3.6e-3	2.3e-2 ± 4.4e-3	2.2e-2 ± 3.2e-3	2.2e-2 ± 3.7e-3	2.3e-2 ± 3.1e-3	2.3e-2 ± 4.4e-3
eunite	700.4 ± 118.4	625.8 ± 62.1	593.1 ± 95.0	592.5 ± 95.0	596.9 ± 95.8	594.6 ± 96.3	596.9 ± 95.8	594.6 ± 96.2
space-ga	2.7e-2 ± 3.9e-3	1.9e-2 ± 2.0e-3	1.9e-2 ± 2.0e-3	1.9e-2 ± 2.0e-3	1.9e-2 ± 2.0e-3	1.9e-2 ± 2.0e-3	1.9e-2 ± 2.0e-3	1.9e-2 ± 2.0e-3
cpusmall	42.0 ± 13.1	44.5 ± 4.4	42.9 ± 5.9	42.9 ± 5.9	42.9 ± 5.9	42.91 ± 5.9	42.9 ± 5.9	42.9 ± 5.9
mg	1.6e-2 ± 3.3e-4	1.5e-2 ± 7.6e-4	1.5e-2 ± 9.7e-4	1.5e-2 ± 9.7e-4	1.5e-2 ± 9.7e-4	1.5e-2 ± 9.7e-4	1.5e-2 ± 9.7e-4	1.5e-2 ± 9.7e-4
abalone	6.4 ± 0.5	5.7 ± 0.5	5.5 ± 0.3	5.5 ± 0.3	5.5 ± 0.3	5.5 ± 0.3	5.5 ± 0.3	5.5 ± 0.3



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

(f) housing (g) mpg (h) eunite2001 (i) space_ga (j) mg

Figure 1. The mean square discrepancies between 5CV and BIF-5CV, 10CV and BIF-10CV, 20CV and BIF-20CV with different t, where t is the order of Taylor expansion.

The evaluation is made on 20 publicly available data sets from LIBSVM Data: 10 data sets for classification and 10 data sets for regression seen in Table 1. Experiments are performed on a Dell Vestro PC with 3.4-GHz 8-core CPU and 8-GB memory.

We use $K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2/2\tau)$ as our candidate kernels, $\tau \in \{2^i, i = -6, -5, \ldots, 7, 8\}$ [2]. The regularization parameter $\lambda \in \{2^i, i = -7, -6, \ldots, 2\}$. The learning algorithm considered in our experiments is LSSVM. For each data set, we have run all the methods 10 times with training and testing data sets be split randomly (50% of all the examples for training and the other 50% for testing).

Accuracy. We will compare our proposed BIF-kCV with the traditional k-fold cross-validation (kCV), the efficient leave-one-out cross-validation (ELOO) (Cawley, 2006; Cawley & Talbot, 2007) and the latest proposed eigenvalues perturbation criterion (EP) (Liu et al., 2013).

In our first experiment, we set the order of Taylor expansion $t = 3$. The average testing errors for classification and testing mean square error for regression are reported in Table 1. For each training set, we choose the kernel parameter τ and regularization parameter λ by each criterion on the training set, and evaluate the testing error for the chosen parameters on the testing set.

[2]Note for LSSVM, when τ is too small (e.g. 2^{-6}), our approximation would probably fail due to (near) identity kernel matrix. But we can easily exclude such small τ (which is unlikely to be an optimal parameter) by setting our approximate criterion to ∞ if the kernel matrix is near identity.

Table 2. The average computational time (second), the order of Taylor expansion $t = 3$

Classification	EP	ELOO	5CV	BIF-5CV	10CV	BIF-10CV	20CV	BIF-20CV
ionosphere	0.91 ± 0.01	0.43 ± 0.02	0.87 ± 0.02	0.47 ± 0.01	2.02 ± 0.03	0.66 ± 0.01	4.60 ± 0.03	1.02 ± 0.01
breast	2.99 ± 0.05	1.42 ± 0.11	2.88 ± 0.06	1.70 ± 0.04	6.83 ± 0.15	2.20 ± 0.05	14.03 ± 0.23	3.21 ± 0.08
diabetes	3.57 ± 0.04	2.10 ± 0.09	3.30 ± 0.04	2.57 ± 0.03	8.17 ± 0.21	3.46 ± 0.03	21.63 ± 0.11	5.15 ± 0.04
fourclass	4.23 ± 0.02	2.50 ± 0.08	4.40 ± 0.17	3.31 ± 0.09	11.64 ± 0.25	4.39 ± 0.18	26.05 ± 0.48	6.52 ± 0.27
australian	2.82 ± 0.17	1.45 ± 0.23	2.70 ± 0.09	1.71 ± 0.03	6.81 ± 0.04	2.19 ± 0.01	13.76 ± 0.04	3.17 ± 0.08
heart	0.58 ± 0.01	0.30 ± 0.01	0.58 ± 0.01	0.32 ± 0.02	1.31 ± 0.02	0.45 ± 0.01	2.79 ± 0.03	0.73 ± 0.02
german	7.02 ± 0.06	3.85 ± 0.13	6.78 ± 0.12	4.65 ± 0.10	16.89 ± 0.18	6.10 ± 0.11	38.88 ± 0.34	8.99 ± 0.07
liver	1.04 ± 0.04	0.42 ± 0.02	0.81 ± 0.01	0.46 ± 0.01	1.96 ± 0.02	0.62 ± 0.02	4.02 ± 0.08	0.97 ± 0.02
sonar	0.54 ± 0.01	0.25 ± 0.01	0.48 ± 0.01	0.23 ± 0.00	1.05 ± 0.02	0.36 ± 0.02	2.24 ± 0.03	0.57 ± 0.01
a2a	58.44 ± 0.21	31.87 ± 0.20	52.66 ± 0.15	37.50 ± 0.39	142.92 ± 0.70	46.57 ± 0.27	308.06 ± 0.96	64.85 ± 0.38
Regression	EP	ELOO	5CV	BIF-5CV	10CV	BIF-10CV	20CV	BIF-20CV
bodyfat	0.76 ± 0.01	0.30 ± 0.04	0.59 ± 0.03	0.32 ± 0.02	1.28 ± 0.06	0.45 ± 0.02	2.61 ± 0.16	0.72 ± 0.04
housing	1.78 ± 0.01	0.86 ± 0.03	1.60 ± 0.04	0.91 ± 0.02	3.63 ± 0.12	1.22 ± 0.02	8.38 ± 0.20	1.85 ± 0.01
mpg	1.01 ± 0.02	0.52 ± 0.04	0.99 ± 0.01	0.57 ± 0.01	2.31 ± 0.00	0.77 ± 0.01	4.94 ± 0.02	1.19 ± 0.01
pyrim	0.23 ± 0.01	0.09 ± 0.01	0.20 ± 0.01	0.09 ± 0.01	0.40 ± 0.01	0.15 ± 0.01	0.78 ± 0.01	0.24 ± 0.01
triazines	0.39 ± 0.03	0.22 ± 0.01	0.46 ± 0.01	0.21 ± 0.00	0.94 ± 0.02	0.30 ± 0.00	2.00 ± 0.03	0.50 ± 0.01
eunite	0.17 ± 0.07	0.42 ± 0.03	0.83 ± 0.07	0.43 ± 0.02	1.75 ± 0.08	0.61 ± 0.02	3.83 ± 0.20	0.94 ± 0.04
space-ga	97.77 ± 0.15	64.89 ± 6.29	93.65 ± 0.45	69.84 ± 0.62	252.3 ± 0.77	85.49 ± 0.28	600.1 ± 0.42	117.8 ± 0.2
cpusmall	73.65 ± 0.03	41.69 ± 0.25	68.49 ± 2.48	48.38 ± 1.28	172.2 ± 5.66	60.21 ± 0.82	395.8 ± 11.9	85.01 ± 1.33
mg	16.25± 0.05	8.49 ± 0.17	13.36 ± 0.47	8.99 ± 0.07	37.17 ± 0.46	13.00 ± 0.04	81.72 ± 0.73	19.15 ± 0.02
abalone	275.5± 3.52	152.8 ± 3.45	253.2 ± 2.66	168.7 ± 1.92	730.4 ± 3.62	196.7 ± 1.56	1760.9 ± 8.05	255.1 ± 3.41

The results in Table 1 can be summarized as follows: (a) On most data sets, BIF-kCV gives almost the same testing errors as the traditional kCV, $k = 5, 10, 20$. On breast, diabetes, australian, fourclass, german, a2a, bodyfat, housing, eunite, space-ga, mg and abalone, BIF-kCV gives the same testing errors as kCV. On the remaining data sets, both BIF-kCV and kCV give the similar results. Thus, it implicates that the quality of our approximation based on the Bouligand influence function is quite good. (b) BIF-kCV gives much better results than EP on most data sets. In particular, BIF-CV outperforms EP on 16 out of 20 data sets, and also give results close to results of EP on the remaining 4 sets. (c) For classification, BIF-kCV and ELOO give comparable results. However, for regression, BIF-kCV outperforms ELOO on 8 out of 10 data sets.

In the second experiment, we will explore the effect of the parameter t (the order of Taylor expansion). The discrepancies between kCV and BIF-kCV with different k are given in Figure 1 (due to space limit, we randomly select 5 classification data sets and 5 regression data sets). For each training set, we choose the τ and λ by cross validation on the training set. Plotted are the mean square error of the approximate $f_{P_n^{-S_i}}(\boldsymbol{x})$'s (computed by BIF-$k$CV) and $f_{P_n^{-S_i}}(\boldsymbol{x})$'s (computed by kCV). for the chosen parameters on the validation sample S_i, $\boldsymbol{x} \in S_i$, $i = 1, \ldots, k$. We can find that, on most data sets, the discrepancies between kCV and BIF-kCV is equal 0 when $t \geq 3$. Thus, we can select $t = 3$ in practice without sacrificing accuracy.

Efficiency. The running time are reported in Table 2. The results in Table 2 can be summarized as follows: (a) The time cost of BIF-kCV is much lower than that of kCV. Thus, BIF-kCV significantly improves the efficiency of kCV. (b) BIF-5CV and BIF-10CV are faster than EP, BIF-20CV and EP are comparable in computing time. (c) BIF-5CV and ELOO give the similar results.

7. Conclusion

We propose a novel strategy for approximating the k-fold cross-validation error based on the Bouligand influence function (BIF), which can be computed efficiently. Link between the concept of BIF and concept of cross-validation is considered. The calculation of the higher order BIFs and a recursive relation are proposed. It is shown that these theoretical results can be applied in practice to approximate the cross-validation error. Experiments indicate that our proposed criterion based on BIF is a good choice for model selection.

Future work will extend our method to other kernel based methods, such as kernel-based logistic regression and SVM.

Acknowledgments

The work is supported in part by the National Natural Science Foundation of China under grant No. 61170019, the Natural Science Foundation of Tianjin under grant No. 11JCYBJC00700, and Tianjin Key Laboratory of Cognitive Computing and Application.

Appendix A: Proof of Theorem 1

Proof. From Theorem 2 in (Vito et al., 2004), we have

$$2\lambda f_P = \mathbb{E}_P[V_P' \Phi(\boldsymbol{x})], 0 = \mathbb{E}_P V_P'. \tag{5}$$

Let $f_\epsilon = f_{P_{\epsilon,Q}}$ and $b_\epsilon = b_{P_{\epsilon,Q}}$. Note that $P_{\epsilon,Q} = (1-\epsilon)P + \epsilon Q$, thus we can obtain that

$$2\lambda f_\epsilon = (1-\epsilon)\mathbb{E}_P[V_\epsilon' \Phi(\boldsymbol{x})] + \epsilon \mathbb{E}_Q[V_\epsilon' \Phi(\boldsymbol{x})] \tag{6}$$

$$0 = (1-\epsilon)\mathbb{E}_P V_\epsilon' + \epsilon \mathbb{E}_Q V_\epsilon', \tag{7}$$

where $V_\epsilon = V(y - f_\epsilon(\boldsymbol{x}) - b_\epsilon)$.

Taking the first derivative on both sides of (6) with respect to ϵ yields

$$\begin{aligned}&2\lambda\frac{\partial}{\partial\epsilon}f_\epsilon = \\ &(1-\epsilon)\mathbb{E}_P[-(\frac{\partial}{\partial\epsilon}f_\epsilon(\boldsymbol{x})+\frac{\partial}{\partial\epsilon}b_\epsilon)V_\epsilon''\Phi(\boldsymbol{x})]-\mathbb{E}_P(V_\epsilon'\Phi(\boldsymbol{x})) \\ &+\epsilon\mathbb{E}_Q[-(\frac{\partial}{\partial\epsilon}f_\epsilon(\boldsymbol{x})+\frac{\partial}{\partial\epsilon}b_\epsilon)V_\epsilon''\Phi(\boldsymbol{x})]+\mathbb{E}_Q(V_\epsilon'\Phi(\boldsymbol{x})).\end{aligned} \tag{8}$$

Set $\epsilon=0$ and according to (5), we have

$$\begin{aligned}&2\lambda\frac{\partial}{\partial\epsilon}f_\epsilon|_{\epsilon=0}+\mathbb{E}_P[(\frac{\partial}{\partial\epsilon}f_\epsilon(\boldsymbol{x})|_{\epsilon=0}+\frac{\partial}{\partial\epsilon}b_\epsilon|_{\epsilon=0})V_P''\Phi(\boldsymbol{x})] \\ &=-2\lambda f_P+\mathbb{E}_Q(V_P'\Phi(\boldsymbol{x})).\end{aligned} \tag{9}$$

Taking the first derivative on both sides of (7) with respect to ϵ yields

$$\begin{aligned}0=&(1-\epsilon)E_P[(-\frac{\partial}{\partial\epsilon}f_\epsilon(\boldsymbol{x})-\frac{\partial}{\partial\epsilon}b_\epsilon)V_\epsilon'']-\mathbb{E}_PV_\epsilon' \\ &+\epsilon E_Q[(-\frac{\partial}{\partial\epsilon}f_\epsilon(\boldsymbol{x})-\frac{\partial}{\partial\epsilon}b_\epsilon)V_\epsilon'']+\mathbb{E}_QV_\epsilon'.\end{aligned} \tag{10}$$

Set $\epsilon=0$ and according to (5),

$$E_P[(\frac{\partial}{\partial\epsilon}f_\epsilon(\boldsymbol{x})|_{\epsilon=0}+\frac{\partial}{\partial\epsilon}b_\epsilon|_{\epsilon=0})V_P'']=\mathbb{E}_QV_P'. \tag{11}$$

By the definition of the operator L, the system of linear equations, (9) and (11), can be written as $L\Big[\frac{\partial}{\partial\epsilon}f_\epsilon|_{\epsilon=0},\frac{\partial}{\partial\epsilon}b_\epsilon|_{\epsilon=0}\Big] = \Big[-2\lambda f_P+\mathbb{E}_Q(V_P'\Phi(\boldsymbol{x})),\mathbb{E}_Q(V_P')\Big]$. □

Appendix B: Proof of the Theorem 2

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

$$\begin{aligned}2\lambda\frac{\partial}{\partial\epsilon^k}f_\epsilon =&(1-\epsilon)\mathbb{E}_P[-(\frac{\partial}{\partial\epsilon^k}f_\epsilon(\boldsymbol{x})+\frac{\partial}{\partial\epsilon^k}b_\epsilon)V_\epsilon''\Phi(\boldsymbol{x})]+ \\ &k\mathbb{E}_P[(\frac{\partial}{\partial\epsilon^{k-1}}f_\epsilon(\boldsymbol{x})+\frac{\partial}{\partial\epsilon^{k-1}}b_\epsilon)V_\epsilon''\Phi(\boldsymbol{x})]- \\ &k\mathbb{E}_Q[(\frac{\partial}{\partial\epsilon^{k-1}}f_\epsilon(\boldsymbol{x})+\frac{\partial}{\partial\epsilon^{k-1}}b_\epsilon)V_\epsilon''\Phi(\boldsymbol{x})]- \\ &\epsilon\mathbb{E}_Q[(\frac{\partial}{\partial\epsilon^k}f_\epsilon(\boldsymbol{x})+\frac{\partial}{\partial\epsilon^k}b_\epsilon)V_\epsilon''\Phi(\boldsymbol{x})].\end{aligned} \tag{12}$$

Taking the derivative on both sides of (8) with respect to ϵ yields $2\lambda\frac{\partial}{\partial^2\epsilon}f_\epsilon = (1-\epsilon)\mathbb{E}_P[-(\frac{\partial}{\partial^2\epsilon}f_\epsilon(\boldsymbol{x})+\frac{\partial}{\partial^2\epsilon}b_\epsilon)V_\epsilon''\Phi(\boldsymbol{x})] + 2\mathbb{E}_P\left[(\frac{\partial}{\partial\epsilon}f_\epsilon(\boldsymbol{x})+\frac{\partial}{\partial\epsilon}b_\epsilon)V_\epsilon''\Phi(\boldsymbol{x})\right] + \epsilon\mathbb{E}_Q[-(\frac{\partial}{\partial^2\epsilon}f_\epsilon(\boldsymbol{x})+\frac{\partial}{\partial^2\epsilon}b_\epsilon)V_\epsilon''\Phi(\boldsymbol{x})]+2\mathbb{E}_QV_\epsilon''(\frac{\partial}{\partial\epsilon}f_\epsilon(\boldsymbol{x})+\frac{\partial}{\partial\epsilon}b_\epsilon)\Phi(\boldsymbol{x})$. Thus for $k=2$, the Equation (12) is satisfied.

Taking the derivatives of both sides in (12),

$$\begin{aligned}&2\lambda\frac{\partial}{\partial\epsilon^{k+1}}f_\epsilon \\ &=(1-\epsilon)\mathbb{E}_P[-(\frac{\partial}{\partial\epsilon^{k+1}}f_\epsilon(\boldsymbol{x})+\frac{\partial}{\partial\epsilon^{k+1}}b_\epsilon)V_\epsilon''\Phi(\boldsymbol{x})] \\ &+(k+1)\mathbb{E}_P[(\frac{\partial}{\partial\epsilon^k}f_\epsilon(\boldsymbol{x})+\frac{\partial}{\partial\epsilon^k}b_\epsilon)V_\epsilon''\Phi(\boldsymbol{x})] \\ &-(k+1)\mathbb{E}_Q(\frac{\partial}{\partial\epsilon^k}f_\epsilon(\boldsymbol{x})+\frac{\partial}{\partial\epsilon^k}b_\epsilon)V_\epsilon''\Phi(\boldsymbol{x}) \\ &-\epsilon\mathbb{E}_Q(\frac{\partial}{\partial\epsilon^{k+1}}f_\epsilon(X)+\frac{\partial}{\partial\epsilon^{k+1}}b_\epsilon)V_\epsilon''\Phi(\boldsymbol{x})\end{aligned}$$

from which it follows that (12) holds for $k+1$ indeed. Set $\epsilon=0$:

$$\begin{aligned}&2\lambda\frac{\partial}{\partial\epsilon^{k+1}}f_\epsilon|_{\epsilon=0}+ \\ &\mathbb{E}_P[(\frac{\partial}{\partial\epsilon^{k+1}}f_\epsilon(\boldsymbol{x})|_{\epsilon=0}+\frac{\partial}{\partial\epsilon^{k+1}}b_\epsilon|_{\epsilon=0})V_P''\Phi(\boldsymbol{x})]= \\ &(k+1)\mathbb{E}_P[(\frac{\partial}{\partial\epsilon^k}f_\epsilon(\boldsymbol{x})|_{\epsilon=0}+\frac{\partial}{\partial\epsilon^k}b_\epsilon|_{\epsilon=0})V_P''\Phi(\boldsymbol{x})]- \\ &(k+1)\mathbb{E}_Q(\frac{\partial}{\partial\epsilon^k}f_\epsilon(\boldsymbol{x})|_{\epsilon=0}+\frac{\partial}{\partial\epsilon^k}b_\epsilon|_{\epsilon=0})V_P''\Phi(\boldsymbol{x}).\end{aligned}$$

Taking the derivative on both sides of (10) and setting $\epsilon=0$, we have

$$\begin{aligned}&\mathbb{E}_P[-(\frac{\partial}{\partial^2\epsilon}f_\epsilon(\boldsymbol{x})|_{\epsilon=0}-\frac{\partial}{\partial^2\epsilon}b_\epsilon|_{\epsilon=0})V_P'']= \\ &\mathbb{E}_P(\frac{\partial}{\partial\epsilon}f_\epsilon(\boldsymbol{x})|_{\epsilon=0}+\frac{\partial}{\partial\epsilon}b_\epsilon|_\epsilon)^2V_P'''+ \\ &2\mathbb{E}_P[(\frac{\partial}{\partial\epsilon}f_\epsilon(\boldsymbol{x})|_{\epsilon=0}+\frac{\partial}{\partial\epsilon}b_\epsilon|_{\epsilon=0})V_P'']- \\ &2\mathbb{E}_Q[(\frac{\partial}{\partial\epsilon}f_\epsilon(\boldsymbol{x})|_{\epsilon=0}+\frac{\partial}{\partial\epsilon}b_\epsilon|_{\epsilon=0})V_\epsilon''].\end{aligned} \tag{13}$$

Similar to the above proof, it is easy to verify that

$$\begin{aligned}&\mathbb{E}_P[(\frac{\partial}{\partial\epsilon^{k+1}}f_\epsilon(X)|_{\epsilon=0}+\frac{\partial}{\partial\epsilon^{k+1}}b_\epsilon|_{\epsilon=0})V_P'']= \\ &(k+1)\mathbb{E}_P[(\frac{\partial}{\partial\epsilon^k}f_\epsilon(X)|_{\epsilon=0}+\frac{\partial}{\partial\epsilon^k}b_\epsilon|_{\epsilon=0})V_P'']- \\ &(k+1)\mathbb{E}_Q(\frac{\partial}{\partial\epsilon^k}f_\epsilon(X)|_{\epsilon=0}+\frac{\partial}{\partial\epsilon^k}b_\epsilon|_{\epsilon=0})V_P''.\end{aligned}$$

Thus, we have

$$\begin{aligned}&L\left[\frac{\partial}{\partial\epsilon^{k+1}}f_\epsilon|_{\epsilon=0},\frac{\partial}{\partial\epsilon^{k+1}}b_\epsilon|_{\epsilon=0}\right] \\ &=(k+1)\Big[\mathbb{E}_P(BIF_k(Q;(f_{\lambda,K}),P))V_P''(\Phi(\boldsymbol{x})) \\ &\quad-\mathbb{E}_Q(BIF_k(Q;(f_{\lambda,K}),P)V_P'')\Phi(\boldsymbol{x}), \\ &\quad+E_P(BIF_k(Q;(f_{\lambda,K}),P))V_P'' \\ &\quad-\mathbb{E}_Q(BIF_k(Q;(f_{\lambda,K}),P)V_P'')\Big].\end{aligned}$$

□

References

Aronszajn, Nachman. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68: 337–404, 1950.

Bartlett, Peter L. and Mendelson, Shahar. Rademacher and gaussian complexities: Risk bounds and structural results. *The Journal of Machine Learning Research*, 3: 463–482, 2002.

Bartlett, Peter L., Boucheron, Stéphane, and Lugosi, Gábor. Model selection and error estimation. *Machine Learning*, 48:85–113, 2002.

Cawley, Gavin C. Leave-one-out cross-validation based model selection criteria for weighted ls-svms. In *Proceeding of the International Joint Conference on Neural Networks (IJCNN 2006)*, pp. 1661–1668, 2006.

Cawley, Gavin C. and Talbot, Nicola L. C. Preventing overfitting during model selection via bayesian regularisation of the hyper-parameters. *Journal of Machine Learning Research*, 8:841–861, 2007.

Chapelle, Olivier, Vapnik, Vladimir, Bousquet, Olivier, and Mukherjee, Sayan. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1-3): 131–159, 2002.

Christmann, Andreas and Messem, Arnout Van. Bouligand derivatives and robustness of support vector machines for regression. *Journal of Machine Learning Research*, 9:915–936, 2008.

Christmann, Andreas and Steinwart, Ingo. On robustness properties of convex risk minimization methods for pattern recognition. *Journal of Machine Learning Research*, 5:1007–1034, 2004.

Christmann, Andreas and Steinwart, Ingo. Consistency and robustness of kernel based regression. *Bernoulli*, 13: 799–819, 2007.

Christmann, Andreas, Messem, Arnout Van, and Steinwart, Ingo. On consistency and robustness properties of support vector machines for heavy-tailed distributions. *Statistics and Its Interface*, 2:311–327, 2009.

Debruyne, Michiel, Hubert, Mia, and Suykens, Johan A.K. Model selection in kernel based regression using the influence function. *Journal of Machine Learning Research*, 9:2377–2400, 2008.

Golub, Gene H., Heath, Michael, and Wahba, Grace. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.

Hampel, Frank R, Ronchetti, Elvezio M, Rousseeuw, Peter J, and Stahel, Werner A. *Robust Statistics: The Approach Based on Influence Functions*. Wiley, New York, 1986.

Liu, Yong, Jiang, Shali, and Liao, Shizhong. Eigenvalues perturbation of integral operator for kernel selection. In *Proceedings of the 22th ACM International Conference on Information and Knowledge Management (CIKM 2013)*, 2013.

Luxburg, Ulrike Von, Bousquet, Olivier, and Schölkopf, Bernhard. A compression approach to Support Vector model selection. *Journal of Machine Learning Research*, 5:293–323, 2004.

Messem, Arnout Van and Christmann, Andreas. A review on consistency and robustness properties of support vector machines for heavy-tailed distributions. *Advances in Data Analysis and Classification*, 4(2-3):199–220, 2010.

Robinson, Stephen M. An implicit-function theorem for a class of nonsmooth functions. *Mathematics of Operations Research*, 16:292–309, 1991.

Schölkopf, Bernhard and Smola, Alexander J. *Learning with kernels*. MIT Press, Cambridge, MA, 2002.

Shawe-Taylor, John and Cristianini, Nello. *An introduction to support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000. ISBN 0521780195.

Steinwart, Ingo and Christmann, Andreas. *Support vector machines*. Springer Verlag, New York, 2008.

Suykens, Johan A. K. and Vandewalle, Joos. Least squares Support Vector Machine classifiers. *Neural Processing Letters*, 9(3):293–300, 1999.

Vapnik, Vladimir. *The nature of statistical learning theory*. Springer Verlag, 2000.

Vito, Ernesto De, Rosasco, Lorenzo, Caponnetto, Andrea, Piana, Michele, and Verri, Alessandro. Some properties of regularized kernel methods. *Journal of Machine Learning Research*, 5:1363–1390, 2004.

Wahba, Grace. *Spline Models for Observational Data*. CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM, 1990.

Wahba, Grace, Lin, Yi, and Zhang, Hao. GACV for support vector machines. In *Advances in Large Margin Classifiers*. MIT Press, Cambridge,, 1999.