Fast Cross-Validation

Yong Liu[1], **Hailun Lin**[1*], **Lizhong Ding**[2], **Weiping Wang**[1], **Shizhong Liao**[3]
[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]King Abdullah University of Science and Technology (KAUST)
[3]Tianjin University
{liuyong,linhailun,wangweiping}@iie.ac.cn, lizhong.ding@kaust.edu.sa and szliao@tju.edu.cn

Abstract

Cross-validation (CV) is the most widely adopted approach for selecting the optimal model. However, the computation of CV has high complexity due to multiple times of learner training, making it disabled for large scale model selection. In this paper, we present an approximate approach to CV based on the theoretical notion of Bouligand influence function (BIF) and the Nyström method for kernel methods. We first establish the relationship between the theoretical notion of BIF and CV, and propose a method to approximate the CV via the Taylor expansion of BIF. Then, we provide a novel computing method to calculate the BIF for general distribution, and evaluate BIF for sample distribution. Finally, we use the Nyström method to accelerate the computation of the BIF matrix for giving the finally approximate CV criterion. The proposed approximate CV requires training only once and is suitable for a wide variety of kernel methods. Experimental results on lots of datasets show that our approximate CV has no statistical discrepancy with the original CV, but can significantly improve the efficiency.

1 Introduction

Kernel methods, such as SVM, least square SVM and kernel ridge regression (KRR), have been successfully solving various problems in machine learning community. The performance of these algorithms greatly depends on the selection of the hyper-parameters. Therefore, model selection is foundational to kernel methods and is also a challenging problem in kernel methods.

There have been many interesting attempts to derive the theoretical bounds of the generalization error or other techniques to select the hyper-parameters [Liu and Liao, 2015; Ding and Liao, 2014a; 2017; 2014b; Liu *et al.*, 2013; 2017; Li *et al.*, 2017; Liu and Liao, 2014], but the most widely accepted model selection method is still the t-fold cross-validation (t-CV). Unfortunately, t-CV requires training t times, making it disabled for large scale model selection.

*Corresponding author.

In this paper, we present an approach to approximating the CV based on the notion of Bouligand influence function (BIF) [Christmann and Messem, 2008] and Nyström method [Ding and Liao, 2012] for a variety of kernel methods, including LSSVM, KRR and SVM. Specifically, we first show how to approximating the CV via the Taylor expansion of BIF. Then, we provide a method to calculate the BIF for general distribution, and further evaluate BIF for sample distribution. Finally, we use the Nyström method to improve the efficiency of the computation of the BIF matrix, and give an approximate CV criterion for model selection of kernel methods. The proposed approximate CV requires training on the full data only once, hence can significantly improve the efficiency. Experimental results on 18 datasets show that our proposed CV can not only give the comparable results as the state-of-the-art methods, but also significantly improve the efficiency.

Related Work

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

Approximate CV of Kernel Methods

The extreme form of t-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 [Chapelle *et al.*, 2002; Vapnik and Chapelle, 2000] for SVM, [Cawley and Talbot, 2007; Ding and Liao, 2011; Ding *et al.*, 2018] for LSSVM, [Cawley and Talbot, 2004] for sparse LSSVM, [Cawley and Talbot, 2008] for kernel logistic regression, and [Debruyne *et al.*, 2008] 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-CV. In our previous [Liu *et al.*, 2014], we present a strategy for approximating the general CV based on the notion of Bouligand influence function (BIF) for some kernel-based algorithms. However, there are two limitations of this approximate method: 1) the loss function used in kernel-based algorithms must be differentiable, hence it can not be used for the non-differentiable case, such as the popular SVM; 2) we need to compute the inversion of the BIF matrix of time complexity $O(n^3)$, which is not suitable for large scale problem. To overcome these limitations, we propose a novel method to smooth the non-differentiable loss based

on the Huber function, and we use the Nyström method to improve the efficiency of the computation of the BIF matrix,

Bouligand Influence Function

In the field of robust statistics, the notion of influence function (IF) [Hampel *et al.*, 1986] is introduced in order to analyze the effects of outliers on the algorithm. Steinwart and Christmann [2008] showed that SVMs for classification and regression have a bounded influence function under some assumption on the loss function. Koh and Liang [2017] used the notion of influence functions to trace a model's prediction through the learning algorithm and back to its training data. Christmann and Messem [2008] generalized the notion of influence function, and introduced 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 also showed that SVMs have a bounded BIF with some assumptions on loss function. The focus of the above work lies in deriving theoretical analysis of robust statistics for some kernel methods, but little work aims at developing practical procedures for model assessment.

The rest of the paper is organized as follows. We introduce some notations and preliminaries in Section 2. In Section 3, we present an approximate CV method via BIF. In Section 4, we use the Nyström method to accelerate the computation of the BIF matrix, and give the final model section criterion. In Section 5, we analyze the performance of our proposed criterion compared with other state-of-the-art model selection criteria. We end in Section 6 with conclusion.

2 Notations and Preliminaries

We consider the 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 $\{\mathbf{x}_i\}_{i=1}^n$, the kernel matrix

$$\mathbf{K} = [\kappa(\mathbf{x}_i, \mathbf{x}_j)]_{i,j=1}^n$$

is positive semidefinite. The reproducing kernel Hilbert space (RKHS) $\mathcal{H}_\kappa$ 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

$$\langle \kappa(\mathbf{x}, \cdot), \kappa(\mathbf{x}', \cdot) \rangle_\kappa = \kappa(\mathbf{x}, \mathbf{x}').$$

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

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

where $V : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+ \cup \{0\}$ is a loss function, λ is the regularization parameter and $\|\cdot\|_\kappa$ is the norm in RKHS.

Let $\mathbb{P}_\mathcal{S}$ be the sample distribution associated with $\mathcal{S}$, that is $\mathbb{P}_\mathcal{S}(z) = \frac{1}{|\mathcal{S}|}$ if $z \in \mathcal{S}$, otherwise 0, where $|\mathcal{S}|$ is the size of $\mathcal{S}$. When $\mathbb{P}_\mathcal{S}$ is used,

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

KRR, LSSVM and SVM are the special cases of such regularized algorithms. For KRR and LSSVM, V is the square loss:

$$V(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2;$$

For SVM, V is the hinge loss:

$$V(y, f(\mathbf{x})) = \max(0, 1 - yf(\mathbf{x})).$$

Let $\mathcal{S}_1, \ldots, \mathcal{S}_t$ be a random equipartition of $\mathcal{S}$ into t parts, called folds. For simplicity, assume that n mod t, and hence,

$$|\mathcal{S}_i| = n/t =: m, i = 1, \ldots, t.$$

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 $\mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}(z) = \frac{1}{n-m}$ if $z \in \mathcal{S}\setminus\mathcal{S}_i$, otherwise 0. Let $f_{\kappa,\mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}}$ be the hypothesis learned on all of the data excluding $\mathcal{S}_i$. Then, the t-fold cross-validation (t-CV) can be written as

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

3 Approximate CV with BIF

In this section, we will introduce the notion of BIF, and show how to use BIF to approximate the CV.

3.1 Bouligand Influence Function (BIF)

Definition 1 ([Christmann and Messem, 2008]). *Let $\mathbb{P}$ be a distribution and f_κ be an operator $f_\kappa : \mathbb{P} \to f_{\kappa,\mathbb{P}}$, then the **Bouligand influence function (BIF)** of f_κ at $\mathbb{P}$ in the direction of a distribution $\mathbb{Q} \neq \mathbb{P}$ is defined as*

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

Denote

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

One can see that $\mathrm{BIF}(\mathbb{Q}; f_\kappa, \mathbb{P})$ is the first order derivative of $f_{\kappa,\mathbb{P}_{\epsilon,\mathbb{Q}}}$ at $\epsilon = 0$. If BIF exist, the following first Taylor expansion holds:

$$f_{\kappa,\mathbb{P}_{\epsilon,\mathbb{Q}}} \approx f_{\kappa,\mathbb{P}} + \epsilon \mathrm{BIF}(\mathbb{Q}; f_\kappa, \mathbb{P}). \tag{2}$$

Note 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 $\mathbb{P}_{\mathcal{S}_i}$ is the sample distribution corresponding to $\mathcal{S}_i$, that is, $\mathbb{P}_{\mathcal{S}_i}(\mathbf{x}) = \frac{1}{m}$ if $\mathbf{x} \in \mathcal{S}_i$, otherwise 0. Thus, if taking

$$\mathbb{Q} = \mathbb{P}_{\mathcal{S}_i}, \epsilon = \frac{-1}{t-1}, \mathbb{P}_{\epsilon,\mathbb{Q}} = \mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}, \mathbb{P} = \mathbb{P}_\mathcal{S}.$$

Equation (2) gives

$$f_{\kappa,\mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}} \approx f_{\kappa,\mathbb{P}_{\mathcal{S}}} + \frac{1}{1-t}\mathrm{BIF}(\mathbb{P}_{\mathcal{S}_i}; f_\kappa, \mathbb{P}_{\mathcal{S}}). \tag{3}$$

Thus, the approximation of t-CV can be written as t-BIF :=

$$\sum_{i=1}^{t}\sum_{z_j\in\mathcal{S}_i} V\left(y_j, f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_j) + \frac{\mathrm{BIF}(\mathbb{P}_{\mathcal{S}_i}; f_\kappa, \mathbb{P}_{\mathcal{S}})(\mathbf{x}_j)}{1-t}\right). \tag{4}$$

Note that t-BIF only depends on the calculation of $f_{\kappa,\mathbb{P}_{\mathcal{S}}}$ and $\mathrm{BIF}(\mathbb{P}_{\mathcal{S}_i}; f_\kappa, \mathbb{P}_{\mathcal{S}})$. Thus, if given the $\mathrm{BIF}(\mathbb{P}_{\mathcal{S}_i}; f_\kappa, \mathbb{P}_{\mathcal{S}})$, 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 $f_{\kappa,\mathbb{P}_{\mathcal{S}\setminus\mathcal{S}_i}}, i = 1,\ldots,t$.

The calculation of BIF at the continuous distribution $\mathbb{P}$ are given as follows:

Theorem 1 ([Liu *et al.*, 2014]). *Let $\mathcal{H}_\kappa$ be the RKHS of a bounded continuous kernel κ on $\mathcal{X}$, and $V(\cdot,\cdot)$ a loss function and $\mathbb{P}$ be a distribution on $\mathcal{Z}$, then the* BIF *of f_κ in the direction of a distribution $\mathbb{Q} \neq \mathbb{P}$ is*

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

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

$$L(f_\kappa) = 2\lambda f_\kappa + \mathbb{E}_{\mathbb{P}}\left[V''(y, f_{\kappa,\mathbb{P}}(\mathbf{x}))f_\kappa(\mathbf{x})\Phi(\mathbf{x})\right].$$

In the next, we will evaluate the BIF of the sample distribution to approximate CV for square loss (KRR and LSSVM) and Hinge loss (SVM).

3.2 Approximate CV of Square Loss

In the next, we will give an approximate CV for square loss. From Theorem 1, we know that the operator L at sample distribution $\mathbb{P}_{\mathcal{S}}$ maps $f_{\kappa,\mathbb{P}_{\mathcal{S}}} \in \mathcal{H}_\kappa$ 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

$$\begin{bmatrix} L(f_{\kappa,\mathbb{P}_{\mathcal{S}}})(\mathbf{x}_1) \\ \vdots \\ L(f_{\kappa,\mathbb{P}_{\mathcal{S}}})(\mathbf{x}_n) \end{bmatrix} = 2\left[\lambda\mathbf{I}_n + \frac{1}{n}\mathbf{K}\right]\begin{bmatrix} f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_1) \\ \vdots \\ f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_n) \end{bmatrix}, \tag{5}$$

where $\mathbf{I}_n = (1,\ldots,n)^{\mathrm{T}}$. Equation (5) indicates that the matrix

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

is the finite sample version of the operator L at $\mathbb{P}_{\mathcal{S}}$.

From Theorem 1, it is now clear that

$$\begin{bmatrix} \mathrm{BIF}(\mathbb{P}_{\mathcal{S}_i}; f_\kappa, \mathbb{P}_{\mathcal{S}})(\mathbf{x}_1) \\ \vdots \\ \mathrm{BIF}(\mathbb{P}_{\mathcal{S}_i}; f_\kappa, \mathbb{P}_{\mathcal{S}})(\mathbf{x}_n) \end{bmatrix} =$$

$$\mathbf{L}^{-1}\left[\frac{[\mathbf{K}\circ\mathbf{C}_i]}{m}\begin{bmatrix} y_1 - f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_1) \\ \vdots \\ y_n - f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_n) \end{bmatrix} - \lambda\begin{bmatrix} f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_1) \\ \vdots \\ f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_n) \end{bmatrix}\right],$$

where $\mathbf{C}_i$ is an $n \times n$ matrix with $[\mathbf{C}_i]_{j,k} = 1$ if $\mathbf{x}_k \in \mathcal{S}_i$, otherwise 0, $\circ$ is the entrywise matrix product.

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

$$[\mathbf{B}]_{j,i} = \mathrm{BIF}(\mathbb{P}_{\mathcal{S}_i}; f_\kappa, \mathbb{P}_{\mathcal{S}})(\mathbf{x}_j).$$

Therefore, according to Equation (3), we can obtain that

$$f_{\kappa,\mathbb{P}(\mathbf{x}_j)} \approx f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_j) + \frac{[\mathbf{B}]_{j,i}}{1-t}.$$

So, the t-CV for square loss can be approximated by

$$t\text{-BIF} := \sum_{i=1}^{t}\sum_{z_j\in\mathcal{S}_i} V\left(y_j, f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_j) + \frac{[\mathbf{B}]_{j,i}}{1-t}\right). \tag{6}$$

3.3 Approximate CV of Hinge Loss

Note that the hinge loss $V(y, f(\mathbf{x})) = \max(0, 1 - yf(\mathbf{x}))$ is not differentiable, but according to Theorem 1, we know that to obtain BIF we should compute the derivative of loss function. Thus, we propose to use a differentiable approximation of it, inspired by the Huber loss:

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

Note that if $h \to 0$, the Huber loss converges to the hinge loss. From the Huber loss, we know that

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

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

We say that $\mathbf{x}_i$ is a *support vector* if

$$|y_i(f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_i)) - 1| < h.$$

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. Similar with the square loss, it is easy to verity that

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

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

$$\begin{bmatrix} \mathrm{BIF}(\mathbb{P}_{\mathcal{S}_i}; f_\kappa, \mathbb{P}_{\mathcal{S}})(\mathbf{x}_1) \\ \vdots \\ \mathrm{BIF}(\mathbb{P}_{\mathcal{S}_i}; f_\kappa, \mathbb{P}_{\mathcal{S}})(\mathbf{x}_n) \end{bmatrix} =$$

$$\mathbf{L}_n^{-1}\left[\frac{\mathbf{K}\circ\mathbf{C}_i}{m}\begin{bmatrix} V'(y_i, f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_1)) \\ \vdots \\ V'(y_i, f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_n)) \end{bmatrix} - 2\lambda\begin{bmatrix} f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_1) \\ \vdots \\ f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_n) \end{bmatrix}\right].$$

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

$$[\mathbf{B}]_{j,i} = \mathrm{BIF}(\mathbb{P}_{\mathcal{S}_i}; f_\kappa, \mathbb{P}_{\mathcal{S}})(\mathbf{x}_j).$$

Therefore, the t-CV for hinge loss can be approximated by

$$t\text{-BIF} := \sum_{i=1}^{t} \sum_{z_j \in \mathcal{S}_i} V\left(y_j, f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_j) + \frac{[\mathbf{B}]_{j,i}}{1-t}\right). \quad (7)$$

Remark 1. *In this subsection, we only give an approximate CV for Hinge loss. In fact, we can use this strategy to approximate CV of other kernel-based algorithms of non-differentiable, such as support vector regression (SVR) and L1-SVM [Steinwart and Christmann, 2008].*

4 Model Selection

According to the above discussion, we know that

$$t\text{-BIF} := \sum_{i=1}^{t} \sum_{z_j \in \mathcal{S}_i} V\left(y_j, f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_j) + \frac{[\mathbf{B}]_{j,i}}{1-t}\right) \quad (8)$$

is an efficient approximation of CV, which only need to training once. However, to obtain t-BIF, we need $O(n^3)$ to calculate the inversion of $\mathbf{L}$ to obtain the BIF matrix $\mathbf{B}$.

To accelerate the computation of the inversion of $\mathbf{L}$, we consider the use of the popular Nyström method . Suppose we randomly sample c columns of the matrix $\mathbf{K}$ uniformly without replacement. Let $\mathbf{C}$ denote the $n \times c$ martix formed by theses columns. Let $\mathbf{W}$ be the $c \times c$ matrix consisting of the intersection of these c columns with the corresponding c rows of $\mathbf{K}$. Without loss of generality, we can rearrange the columns and rows of $\mathbf{K}$ based on this sampling such that:

$$\mathbf{K} = \begin{pmatrix} \mathbf{W}, \mathbf{K}_{21}^{\mathrm{T}} \\ \mathbf{K}_{21}, \mathbf{K}_{22} \end{pmatrix}, \mathbf{C} = \begin{pmatrix} \mathbf{W} \\ \mathbf{K}_{21} \end{pmatrix}.$$

The Nyström method uses $\mathbf{W}$ and $\mathbf{C}$ to construct an approximation $\widetilde{\mathbf{K}}$ of $\mathbf{K}$ defined by:

$$\widetilde{\mathbf{K}} = \mathbf{C}\mathbf{W}^{+}\mathbf{C}^{\mathrm{T}} \approx \mathbf{K}, \quad (9)$$

where $\mathbf{W}^{+}$ is the Moore-Penrose generalized inverse of $\mathbf{W}$.

If we write the SVD of $\mathbf{W} = \mathbf{U}_{\mathbf{W}}\Sigma_{\mathbf{W}}\mathbf{U}_{\mathbf{W}}^{\mathrm{T}}$, then

$$\mathbf{W}^{+} = \mathbf{U}_{\mathbf{W}}\mathbf{\Sigma}_{\mathbf{W}}^{+}\mathbf{U}_{\mathbf{W}}^{\mathrm{T}}, \quad (10)$$

where $\mathbf{U}_{\mathbf{W}}$ and $\mathbf{\Sigma}_{\mathbf{W}}$ is the singular values and singular vectors of $\mathbf{W}$.

Pluging Equation (10) into Equation (9), we can obtain that

$$\begin{aligned}\widetilde{\mathbf{K}} &= \mathbf{C}\mathbf{U}_{\mathbf{W}}\mathbf{\Sigma}_{\mathbf{W}}^{+}\mathbf{U}_{\mathbf{W}}^{\mathrm{T}}\mathbf{C}^{\mathrm{T}} \\ &= \underbrace{\mathbf{C}\mathbf{U}_{\mathbf{W}}\sqrt{\mathbf{\Sigma}_{\mathbf{W}}^{+}}}_{\mathbf{V}}\underbrace{\left(\mathbf{C}\mathbf{U}_{\mathbf{W}}\sqrt{\mathbf{\Sigma}_{\mathbf{W}}^{+}}\right)^{\mathrm{T}}}_{\mathbf{V}^{\mathrm{T}}},\end{aligned}$$

where we let $\mathbf{V} := \mathbf{C}\mathbf{U}_{\mathbf{W}}\sqrt{\mathbf{\Sigma}_{\mathbf{W}}^{+}} \in \mathbb{R}^{n\times c}$.

Note that we need to solve the inverse of $\mathbf{L}$:

$$\mathbf{L} = \begin{cases} \lambda\mathbf{I}_n + \dfrac{1}{n}\mathbf{K} & \text{for square loss} \\ 2\lambda\mathbf{I}_n + \dfrac{1}{2hn}\mathbf{K} & \text{for Hinge loss.} \end{cases}$$

To reduce the computational cost, we intend to use the inverse of $\widetilde{\mathbf{L}}$,

$$\widetilde{\mathbf{L}} = \begin{cases} \lambda\mathbf{I}_n + \dfrac{1}{n}\widetilde{\mathbf{K}} & \text{for square loss} \\ 2\lambda\mathbf{I}_n + \dfrac{1}{2hn}\widetilde{\mathbf{K}} & \text{for Hunge loss,} \end{cases}$$

as an approximation of the inverse of $\mathbf{L}$.

According to the Woodbury formula:

$$(\mathbf{A} + \mathbf{X}\mathbf{Y}\mathbf{Z})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{X}(\mathbf{Y}^{-1} + \mathbf{Z}\mathbf{A}^{-1}\mathbf{X})^{-1}\mathbf{Z}\mathbf{A},$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{X} \in \mathbb{R}^{c\times c}$, $\mathbf{Y} \in \mathbb{R}^{c\times c}$ and $\mathbf{Z} \in \mathbb{R}^{c\times n}$, it is easy to verity that

$$\widetilde{\mathbf{L}}^{-1} = \begin{cases} \dfrac{\mathbf{I}_n - \mathbf{V}\left(n\lambda\mathbf{I}_c + \mathbf{V}^{\mathrm{T}}\mathbf{V}\right)^{-1}\mathbf{V}^{\mathrm{T}}}{\lambda}, & \text{square loss} \\ \dfrac{\mathbf{I}_n - \mathbf{V}\left(4n\lambda h\mathbf{I}_c + \mathbf{V}^{\mathrm{T}}\mathbf{V}\right)^{-1}\mathbf{V}^{\mathrm{T}}}{2\lambda}, & \text{Hinge loss.} \end{cases}$$

Note that $\mathbf{V}^{\mathrm{T}}\mathbf{V} \in \mathbb{R}^{c\times c}$, so only need $\mathcal{O}\left(c^3 + nc^2\right)$ to compute the $\widetilde{\mathbf{L}}^{-1}$.

Therefore, in this paper, we consider the use of the following criterion for model selection:

$$t\text{-FBIF} := \sum_{i=1}^{t} \sum_{z_j \in \mathcal{S}_i} V\left(y_j, f_{\kappa,\mathbb{P}_{\mathcal{S}}}(\mathbf{x}_j) + \frac{[\widetilde{\mathbf{B}}]_{j,i}}{1-t}\right), \quad (11)$$

where $\widetilde{\mathbf{B}}$ is the approximation of $\mathbf{B}$ with $\widetilde{\mathbf{L}}$ replace of $\mathbf{L}$.

Time Complexity Analysis

To compute t-FBIF, we need $O(c^3 + nc^2)$ to calculate the inversion of $\widetilde{\mathbf{L}}$, and $O(n^2 + tnc)$ to calculate the $\mathbf{B}$. Since $f_{\kappa,\mathbb{P}_{\mathcal{S}}}$ has been obtained in the training process. Thus, the overall time complexity t-FBIF is

$$O(c^3 + n^2 + nc^2 + tnc).$$

For the traditional t-CV method, the algorithm under consideration need to be executed t times, hence the time complexities are $O(tn^3)$, which is much larger than $O(c^3 + n^2 + nc^2 + tnc)$.

5 Experiments

In this section, we will compare our proposed approximate t-CV (t−FBIF) with the popular t-CV (t−CV), the efficient leave-one-out CV (ELOO) [Cawley, 2006] and eigenvalue ratio (ER) [Liu and Liao, 2015], $t = 5, 10$. The data sets are 18 publicly available data sets from LIBSVM Data[1]: 9 data sets for classification and 9 data sets for regression. Experiments are performed on a PC of 3.1GHz CPU with 4GB memory. We use the Gaussian kernel

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

as our candidate kernel $\sigma \in \{2^i, i = -15, -14, \ldots, 14, 15\}$. The regularization parameter

$$\lambda \in \{2^i, i = -15, -13, \ldots, 13, 15\}.$$

[1] http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

Classification	5−FBIF	5−CV	10−FBIF	10−CV	ELOO	ER
Diabetes	20.87 ± 1.67	20.00 ± 1.62	21.30 ± 1.95	19.62 ± 1.73	21.75 ± 2.12	21.94 ± 1.89
Australian	13.04 ± 1.85	12.56 ± 1.43	13.01 ± 1.76	12.23 ± 1.83	13.75 ± 1.93	13.35 ± 1.57
Heart	13.58 ± 3.85	13.34 ± 3.08	12.35 ± 3.65	12.12 ± 3.12	14.81 ± 3.12	13.34 ± 2.56
Ionosphere	4.12 ± 1.86	3.87 ± 0.94	4.23 ± 1.23	3.64 ± 0.91	4.56 ± 1.12	5.76 ± 1.23
Breast	2.44 ± 0.43	1.95 ± 0.32	2.93 ± 0.21	2.44 ± 0.43	4.54 ± 0.35	3.74 ± 0.28
German	27.33 ± 2.84	27.00 ± 2.31	28.00 ± 2.34	27.45 ± 2.53	27.43 ± 2.41	28.76 ± 3.45
Liver	24.04 ± 3.23	23.53 ± 3.12	23.43 ± 2.35	23.34 ± 2.12	30.77 ± 2.84	26.45 ± 3.23
Sonar	17.74 ± 2.34	16.13 ± 2.21	16.74 ± 2.54	16.13 ± 2.74	14.52 ± 2.42	14.43 ± 2.45
A2a	20.47 ± 1.23	20.59 ± 0.83	22.09 ± 1.15	20.62 ± 1.01	20.43 ± 1.32	20.12 ± 1.97
Regression	5−FBIF	5−CV	10−FBIF	10−CV	ELOO	ER
Bodyfat	9.46 ± 1.24(e-6)	9.12 ± 1.04(e-6)	9.32 ± 1.32(e-6)	9.05 ± 0.98(e-6)	1.43 ± 0.21(e-6)	1.53 ± 0.32(e-5)
Housing	11.72 ± 3.82	11.09 ± 3.45	11.34 ± 2.94	11.02 ± 3.21	11.34 ± 2.98	11.12 ± 2.46
Triazines	1.62 ± 0.29(e-2)	1.43± 0.24(e-2)	1.31± 0.28(e-2)	1.23± 0.24(e-2)	1.76± 0.23(e-2)	1.77 ± 0.28(e-2)
Mpg	6.31 ± 0.73	5.96 ± 0.87	6.23 ± 0.76	5.84 ± 0.82	6.23 ± 0.92	5.93 ± 0.75
Pyrim	1.55 ± 0.45(e-2)	1.53± 0.35(e-2)	1.52± 0.29(e-2)	1.51± 0.27(e-2)	1.58± 0.26(e-2)	1.73± 0.23(e-2)
Eunite2001	372.34 ± 96.34	324.34 ± 63.34	364.43 ± 91.24	334.82 ± 78.45	364.43 ± 97.23	364.86 ± 86.34
Mg	1.39 ± 0.01(e-2)	1.86± 0.01(e-2)	1.39± 0.02(e-2)	1.86± 0.01(e-2)	1.39± 0.01(e-2)	1.39± 0.01(e-2)
Cpusmall	9.56 ± 1.34	9.32 ± 1.56	9.53 ± 1.67	9.34 ± 1.40	12.93 ±1.45	9.93 ±1.25
Abalone	4.82 ± 0.45	4.08 ±0.31	4.43 ± 0.34	4.05 ± 0.28	5.95 ± 0.44	5.82 ±0.30

Table 1: Test errors for classification and test mean square errors for regression. Our methods: t−FBIF, compared methods: t-CV (t−CV), efficient leave-one-out CV (ELOO) and eigenvalue ratio (ER), t=5,10.

Classification	5−FBIF	5−CV	10−BIF	10−CV	ELOO	ER
Australian	1.84	14.44	2.02	35.87	4.41	4.43
Heart	0.32	1.75	0.35	4.36	0.52	0.52
Ionosphere	0.45	2.93	0.50	6.62	0.84	0.84
Breast-cancer	2.45	13.07	2.65	32.09	3.89	3.90
Diabetes	2.87	17.19	3.01	42.24	5.24	5.27
German.numer	4.56	31.56	5.33	79.94	10.72	10.73
Liver-disorders	0.48	2.47	0.56	5.69	0.73	0.73
Sonar	0.23	1.16	0.30	2.68	0.32	0.33
A2a	45.43	256.77	54.54	761.95	140.72	144.63
Regression	5−FBIF	5−CV	10−BIF	10−CV	ELOO	ER
Bodyfat	0.21	1.59	0.23	3.77	0.42	0.42
Housing	0.83	5.87	0.98	14.99	1.77	1.76
Mpg	0.47	3.43	0.52	8.64	1.02	1.02
Pyrim	0.04	0.32	0.05	0.66	0.07	0.07
Triazines	0.56	0.91	0.67	2.19	0.25	0.25
Eunite2001	0.37	2.12	0.40	4.79	0.61	0.74
Mg	9.11	68.12	9.34	179.06	24.70	24.74
Cpusmall	80.12	486.24	87.08	1813.67	242.73	245.62
Abalone	205.67	2358.77	243.94	6632.53	974.92	987.11

Table 2: The computational time. Our methods: t−FBIF, compared methods: t-CV (t−CV), efficient leave-one-out CV (ELOO) and eigenvalue ratio (ER), t=5,10.

The learning algorithm used in our experiments for regression is KRR (square loss) and for classification is SVM (hinge loss). For each data set, we run all methods 50 times with randomly selected 70% of all data for training and the other 30% for testing. The use of multiple training/test partitions allows an estimate of the statistical significance of differences in performance between methods. Let A_i and B_i be the test errors of methods A and B in partition i, and $d_i = B_i - A_i$, $i = 1, \ldots, 50$. Let $\bar{d}$ and S_d be the mean and standard error of d_i. Then under t-test, with confidence level 95%, we claim that A is significantly better than B (or equivalently B significantly worse than A) if the t-statistic

$$\frac{\bar{d}}{S_d/\sqrt{50}} > 1.676.$$

All statements of statistical significance in the remainder refer to a 95% level of significance.

5.1 Accuracy

The test errors for classification and test mean square errors for regression are reported in Table 1. For our methods, we set $h = 0.05$ and $c = 0.1n$. The parameters (if have) for the compared algorithms follow the same experimental setting in their papers. The elements are obtained as follows: For each training set, we select the kernel parameter σ and the regularization parameter λ by each criterion on the training set, and evaluate the test error for the chosen parameters on the test set. The results in Table 1 can be summarized as follows:

(a) Neither of t−CV and t−FBIF is statistically superior at the 95% level of significance, t=5, 10. Thus, the quality of our approximation is quite good;

(b) t−FBIF is better than ELOO. In particular, t−FBIF is significantly better than ELOO on Heart, Breast, Liver, Bodyfat, Triazines and Cpusmall, but only significantly worse on Sonar.

(c) t−FBIF is better than ER. In particular, t−FBIF is significantly better than ER on Ionosphere, Liver, Bodyfat, Triazines and Abalone, but only significantly worse on Sonar.

5.2 Efficiency

The running time is reported in Table 2 that can be summarized as follows:

(a) The time cost of t−FBIF are much lower than that of tCV. In particular, 5−FBIF and 10−FBIF are 5 (or more) and 10 (or more) times faster than 5CV and 10CV on all datasets, respectively. For large dataset, such as Abalone and Cpusmall, 10−FBIF is 20 times faster than 10-CV. Thus, t−FBIF significantly improves the efficiency of tCV for model selection;

(b) t−FBIF is 2 (or more) faster than ELOO and ER on all datasets. For large dataset, t−FBIF is nearly 5 times than ELOO and ER.

6 Conclusion

In this paper, we present an approximate CV method based on the theoretical notion of BIF and Nyström method for a variety of kernel methods. The proposed approximate CV requires training on the full data only once, hence can significantly improve the efficiency. Experimental results on 18 data sets show that our approximate CV is 20 times more efficiency (for large scale datasets) and has no statistical discrepancy when compared to the original one. This is an interesting attempt to apply the theoretical notion of BIF for practical model selection.

Future work includes extending our criterion to other kernel-based algorithms, such as support vector regression, L1-SVM and kernel logistic regression.

Appendix: Proof of Theorem 1

Proof. From Theorem 2 in [Vito *et al.*, 2004], we have

$$-2\lambda f_{\kappa,\mathbb{P}} = \mathbb{E}_{\mathbb{P}}[V'(y, f_{\kappa,\mathbb{P}}(\mathbf{x}))\Phi(\mathbf{x})], \tag{12}$$

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

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

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

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

Setting $\epsilon = 0$ and according to Equation (12), we have

$$\begin{aligned} &2\lambda\frac{\partial}{\partial\epsilon}f_\epsilon|_{\epsilon=0} + \mathbb{E}_{\mathbb{P}}\left[\left(\frac{\partial}{\partial\epsilon}f_\epsilon(\mathbf{x})|_{\epsilon=0}\right)V''(y, f_{\mathbb{P}}(\mathbf{x}))\Phi(\mathbf{x})\right] \\ &= \mathbb{E}_{\mathbb{P}}\left[V'(y, f_{\mathbb{P}}(\mathbf{x}))\Phi(\mathbf{x})\right] - \mathbb{E}_{\mathbb{Q}}\left[V'(y, f_{\mathbb{P}}(\mathbf{x}))\Phi(\mathbf{x})\right] \\ &= -2\lambda f_{\mathbb{P}} - \mathbb{E}_{\mathbb{Q}}\left[V'(y, f_{\mathbb{P}}(\mathbf{x}))\Phi(\mathbf{x})\right]. \end{aligned} \tag{15}$$

By the definition of the operator L, we can obtain that

$$L\Big[\frac{\partial}{\partial\epsilon}f_\epsilon|_{\epsilon=0}\Big] = -2\lambda f_{\mathbb{P}} - \mathbb{E}_{\mathbb{Q}}\left[V'(y, f_{\mathbb{P}}(\mathbf{x}))\Phi(\mathbf{x})\right].$$

□

Acknowledgments

This work is supported in part by the National Key Research and Development Program of China (2016YFB1000604), the National Natural Science Foundation of China (No.6173396, No.61673293, No.61602467) and the Excellent Talent Introduction of Institute of Information Engineering of CAS (Y7Z0111107).

References

[Cawley and Talbot, 2004] Gavin Cawley and Nicola Talbot. Fast leave-one-out cross-validation of sparse least-squares support vector machines. *Neural Networks*, 17(10):1467–1475, 2004.

[Cawley and Talbot, 2007] Gavin Cawley and Nicola Talbot. Preventing over-fitting during model selection via Bayesian regularisation of the hyper-parameters. *Journal of Machine Learning Research*, 8:841–861, 2007.

[Cawley and Talbot, 2008] Gavin Cawley and Nicola Talbot. Efficient approximate leave-one-out cross-validation for kernel logistic regression. *Machine Learning*, 71(2-3):243–264, 2008.

[Cawley, 2006] Gavin Cawley. Leave-one-out cross-validation based model selection criteria for weighted LS-SVMs. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2006)*, pages 1661–1668, 2006.

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

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

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

[Ding and Liao, 2011] Lizhong Ding and Shizhong Liao. Approximate model selection for large scale LSSVM. *Journal of Machine Learning Research - Proceedings Track*, 20:165–180, 2011.

[Ding and Liao, 2012] Lizhong Ding and Shizhong Liao. Nyström approximate model selection for LSSVM. In *Proceedings of the 16th Pacific-Asia Conference (PAKDD 2012)*, pages 282–293, 2012.

[Ding and Liao, 2014a] Lizhong Ding and Shizhong Liao. Approximate consistency: Towards foundations of approximate kernel selection. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Database (ECML PKDD)*, pages 354–369. Springer, Berlin, 2014.

[Ding and Liao, 2014b] Lizhong Ding and Shizhong Liao. Model selection with the covering number of the ball of RKHS. In *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management (CIKM 2014)*, pages 1159–1168, 2014.

[Ding and Liao, 2017] Lizhong Ding and Shizhong Liao. An approximate approach to automatic kernel selection. *IEEE Transactions on Cybernetics*, 47(3):554–565, 2017.

[Ding *et al.*, 2018] Lizhong Ding, Shizhong Liao, Yong Liu, Peng Yang, and Xin Gao. Randomized kernel selection with spectra of multilevel circulant matrices. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI 2018)*, 2018.

[Hampel *et al.*, 1986] Frank Hampel, Elvezio Ronchetti, Peter Rousseeuw, and Werner Stahel. *Robust statistics: the approach based on influence functions*. Wiley, New York, 1986.

[Koh and Liang, 2017] PangWei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34 th International Conference on Machine Learning (ICML 2017)*, pages 1885–1894, 2017.

[Li *et al.*, 2017] Jian Li, Yong Liu, Hailun Lin, Yinliang Yue, and Weiping Wang. Efficient kernel selection via spectral analysis. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI 2017)*, pages 2124–2130, 2017.

[Liu and Liao, 2014] Yong Liu and Shizhong Liao. Preventing over-fitting of cross-validation with kernel stability. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2014)*, pages 290–305, 2014.

[Liu and Liao, 2015] Yong Liu and Shizhong Liao. Eigenvalues ratio for kernel selection of kernel methods. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI 2015)*, pages 2814–2820, 2015.

[Liu *et al.*, 2013] Yong Liu, Shali Jiang, and Shizhong Liao. Eigenvalues perturbation of integral operator for kernel selection. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (CIKM 2013)*, pages 2189–2198, 2013.

[Liu *et al.*, 2014] Yong Liu, Shali Jiang, and Shizhong Liao. Efficient approximation of cross-validation for kernel methods using Bouligand influence function. In *Proceedings of The 31st International Conference on Machine Learning (ICML 2014 (1))*, pages 324–332, 2014.

[Liu *et al.*, 2017] Yong Liu, Shizhong Liao, Hailun Lin, Yinliang Yue, and Weiping Wang. Infinite kernel learning: generalization bounds and algorithms. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI 2017)*, 2017.

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

[Steinwart and Christmann, 2008] Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Verlag, 2008.

[Vapnik and Chapelle, 2000] Vladimir Vapnik and Olivier Chapelle. Bounds on error expectation for support vector machines. *Neural Computation*, 12(9):2013–2036, 2000.

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