Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)

Online Kernel Selection via Incremental Sketched Kernel Alignment

Xiao Zhang and **Shizhong Liao***
School of Computer Science and Technology, Tianjin University, Tianjin 300350, China
{xiaozhang, szliao}@tju.edu.cn

Abstract

In contrast to offline kernel selection, online kernel selection must rise to the new challenges of passing the training set once, selecting optimal kernels and updating hypotheses at each round, enjoying a sublinear regret bound for online kernel learning, and requiring a constant maintenance time complexity at each round and an efficient overall time complexity integrated with online kernel learning. However, most of existing online kernel selection approaches can not meet the new challenges. To address this issue, we propose a novel online kernel selection approach via the incremental sketched kernel alignment criterion, which meets all the new challenges. We first define the incremental sketched kernel alignment (ISKA) criterion, which estimates the kernel alignment and can be computed incrementally and efficiently. When applying the proposed ISKA criterion to online kernel selection, we adopt the subclass coherence to maintain the hypothesis space, select the optimal kernel at each round using the median of the ISKA criterion estimates, and update the hypothesis following the online gradient decent method. We prove that the ISKA criterion is an unbiased estimate of the maximum mean discrepancy, enjoys the optimal logarithmic regret bound for online kernel learning, and has a constant maintenance time complexity at each round and a logarithmic overall time complexity integrated with online kernel learning. Empirical studies demonstrate that the proposed online kernel selection approach is computationally efficient while maintaining comparable accuracy for online kernel learning.

1 Introduction

Kernel selection is to select the kernel function corresponding to a reproducing kernel Hilbert space (RKHS) by a proper kernel selection criterion. Offline kernel selection criteria are usually defined via the estimate of the generalization error, such as k-fold cross validation [Anguita *et al.*, 2009], Rademacher complexity [Bartlett and Mendelson, 2002] and eigenvalues ratio [Liu and Liao, 2015].

In online setting, the kernels used significantly affect the performance of online kernel learning, which are usually preset empirically without theoretical guarantees. We refer to the kernel selection for online kernel learning as *online kernel selection*. In contrast to offline kernel selection performing separately training and testing on all instances with generalization guarantee, online kernel selection is a completely different problem that intermixes kernel selection and hypothesis updating on arrived instances at each round, which requires a sublinear regret bound, a constant maintenance time complexity at each round and at most a linear overall time complexity integrated with online kernel learning. Traditional offline kernel selection approaches do not apply to the online kernel learning scenario. First, offline kernel selection approaches need multiple passes over the data and have at least quadratic time complexity with respect to the data size, suffering from a low efficiency. Second, most of offline kernel selection approaches consist of the training phase and the testing phase, while online kernel learning can not split the training set and does not separate the two phases.

Much attention has been directed at online model selection. Reisinger *et al.* (2008) presented an online model selection approach for Gaussian process temporal difference using sequential Monte-Carlo methods. Foster *et al.* (2015) established elegant model selection inequalities for online learning problems via the sequential complexity measures, and presented a generic meta-algorithm framework for online model selection [Foster *et al.*, 2017], but efficient online model selection algorithms for online kernel learning need to be further explored. Recently, some online kernel selection approaches have been proposed. Yang *et al.* (2012) presented an online kernel selection approach that learns a probability distribution for each candidate kernel classifier, and selects the optimal kernel according to the distribution, which enjoys a regret bound of order $O(\sqrt{T})$, where T is the number of rounds. Kernel learning using the adaptive kernel was also introduced for online kernel selection, which updates the kernel widths using the gradient descent method for instantaneous loss directly [Chen *et al.*, 2016; Nguyen *et al.*, 2017]. A major limitation of most online kernel selection approaches is that they do not limit the model size at each round, leading to a linear space complexity and

*Corresponding author

a quadratic time complexity with respect to the number of rounds. Besides, the existing adaptive kernel approaches for online kernel selection do not enjoy a sublinear regret bound that is critical to online kernel learning, and are not robust to initial kernels that may lead to poor performance.

In this paper, we propose an online kernel selection approach that is theoretically guaranteed and computationally efficient. We define a novel online kernel selection criterion, which is an unbiased estimate of the maximum mean discrepancy and can be computed incrementally. When applying the proposed criterion to online kernel selection, there needs only logarithmic number of updates, where each update is in a linear time complexity with respect to the budget. In contrast to the existing regret analysis using a fixed kernel, the key challenge of the regret analysis for online kernel selection is to bound the average gradient error with varying kernels, and we derive the optimal logarithmic regret bound for online kernel selection using a novel definition of the weight degradation. Experimental results demonstrate the effectiveness and efficiency of our proposed online kernel selection approach that does not depend on the initially selected kernel.

2 Notations and Preliminary

Let $[T] = \{1, 2, \ldots, T\}$, $[a : b] = \{a, a+1, \ldots, b\}$ and $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^T \subseteq (\mathcal{X} \times \mathcal{Y})^T$ be the sequence of T instances, where $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} = \{-1, +1\}$. We define the numbers of the positive and negative examples as T_+, T_-, respectively. Let $\boldsymbol{y} = [y_1, y_2, \ldots, y_T]^\intercal$ be the label vector, and $\boldsymbol{Y} = \boldsymbol{y}\boldsymbol{y}^\intercal$ be the label matrix. We denote the loss function by $\ell(\cdot, \cdot)$, the kernel function by $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and its corresponding kernel matrix by $\boldsymbol{K} = (\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)) \in \mathbb{R}^{T \times T}$. The reproducing kernel Hilbert space (RKHS) associated with κ is defined as $\mathcal{H}_\kappa = \overline{\operatorname{span}\{\kappa(\cdot, \boldsymbol{x}) : \boldsymbol{x} \in \mathcal{X}\}}$. For kernel selection, we denote by $\mathcal{K} = \{\kappa_i : i \in [N]\}$ the candidate kernel set.

The alignment metric of kernels is a widely used class of kernel selection criteria,which measures the similarity between the kernel matrix and the label matrix. Kernel Alignment (KA) problem is to maximize the following criterion [Sinha and Duchi, 2016]

$$\mathcal{C}_{\mathrm{ka}}(\kappa) := \sum_{i,j \in [T]} \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) y_i y_j = \langle \boldsymbol{K}, \boldsymbol{Y} \rangle.$$

For $i \in [T]$, define the modified labels $\hat{y}_i = 1/T_+$ if $y_i = +1$ and $\hat{y}_i = -1/T_-$ if $y_i = -1$ as in [Kandola *et al.*, 2002]. The corresponding label vector and matrix are denoted by $\hat{\boldsymbol{y}}$ and $\widehat{\boldsymbol{Y}}$. Then the kernel alignment is translated into

$$\widehat{\mathcal{C}}_{\mathrm{ka}}(\kappa) = \langle \boldsymbol{K}, \widehat{\boldsymbol{Y}} \rangle = \left\| \sum_{i \in [T]} \hat{y}_i \kappa(\cdot, \boldsymbol{x}_i) \right\|_{\mathcal{H}_\kappa}^2, \tag{1}$$

which is a biased empirical estimate of the Maximum Mean Discrepancy (MMD) [Gretton *et al.*, 2012]

$$\mathcal{C}_{\mathrm{md}}(\kappa) := \left\| \mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}_1}[\kappa(\cdot, \boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x}' \sim \mathcal{D}_2}[\kappa(\cdot, \boldsymbol{x}')] \right\|_{\mathcal{H}_\kappa}^2,$$

where $\mathcal{D}_1$ ($\mathcal{D}_2$) is the distribution from which the positive (negative) examples are drawn. The hypothesis using the optimal kernel of MMD enjoys a generalization bound of order $O(1/\sqrt{T})$ [Sinha and Duchi, 2016] for the soft-margin SVM.

It is obvious that the time complexity of KA is in $O(NT^2)$[1], where N is the size of the candidate kernel set.

3 Incremental Sketched Kernel Alignment

In this section, we propose the incremental sketched kernel alignment to reduce the computational complexities, which can be applied to online kernel selection.

We consider the *kernel matrix sketch* $\boldsymbol{S}_t^\intercal \boldsymbol{K}_t \boldsymbol{S}_t$ of the kernel matrix $\boldsymbol{K}_t \in \mathbb{R}^{t \times t}$ at round t, where $\boldsymbol{S}_t \in \mathbb{R}^{t \times B_t}$ is the sketch matrix and $\lim_{t \to +\infty} B_t/t = 0$. We first define the *Incremental Sketched Kernel Alignment (ISKA)* as follows

$$\mathcal{C}_{\mathrm{sk}}^{(t)}(\kappa) := \langle \boldsymbol{S}_t^\intercal \boldsymbol{K}_t \boldsymbol{S}_t, \boldsymbol{S}_t^\intercal \widehat{\boldsymbol{Y}}_t \boldsymbol{S}_t \rangle.$$

We choose the sub-sampling sketch matrix as the matrix $\boldsymbol{S}_t$, and introduce two buffers to store the sampled examples of positive class and negative class, denoted by $\mathcal{V}_t^+$ and $\mathcal{V}_t^-$ respectively. Let $\mathcal{V}_t = \mathcal{V}_t^+ \cup \mathcal{V}_t^-$, $B_t^+ = |\mathcal{V}_t^+|$, $B_t^- = |\mathcal{V}_t^-|$, $|\mathcal{V}_t| = B_t$, and denote the examples in the buffers by $\tilde{\boldsymbol{x}} \in \mathcal{V}_t$ with label $\tilde{y}$. Then ISKA is equivalent to

$$\mathcal{C}_{\mathrm{sk}}^{(t)}(\kappa) = \left\| \sum_{\xi = +,-} \frac{1}{B_{t+1}^\xi} \sum_{\tilde{\boldsymbol{x}}_i \in \mathcal{V}_{t+1}^\xi} \tilde{y}_i \kappa(\cdot, \tilde{\boldsymbol{x}}_i) \right\|_{\mathcal{H}_\kappa}^2, \tag{2}$$

which is computed in the updated buffers.

For the incremental computation of ISKA, we transform (2) into

$$\mathcal{C}_{\mathrm{sk}}^{(t)}(\kappa) = \sum_{\xi = +,-} \frac{1}{(B_{t+1}^\xi)^2} \hat{\beta}_{t+1}^\xi - \frac{2}{B_{t+1}^+ B_{t+1}^-} \gamma_{t+1},$$

where

$$\beta_{t+1}^\xi = \sum_{\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}_j \in \mathcal{V}_{t+1}^\xi} \kappa(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}_j), \quad \hat{\beta}_{t+1}^\xi = \frac{B_{t+1}^\xi}{B_{t+1}^\xi - 1} \beta_{t+1}^\xi,$$

and $\gamma_{t+1} = \sum_{\tilde{\boldsymbol{x}}_i \in \mathcal{V}_{t+1}^+, \tilde{\boldsymbol{x}}_j \in \mathcal{V}_{t+1}^-} \kappa(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}_j)$, in which we replace the original β_{t+1}^ξ with $\hat{\beta}_{t+1}^\xi$ to obtain an unbiased estimate of MMD as shown in Theorem 1.

Then we update β_{t+1}^ξ and γ_{t+1} in the ISKA criterion incrementally at each round. When $y_t = +1$, let ξ denote $+$, $\bar{\xi}$ denote $-$ and vice versa. For the insert operation, i.e., $\mathcal{V}_{t+1}^\xi = \mathcal{V}_t^\xi \cup \{\boldsymbol{x}_t\}$, we perform

$$\begin{aligned} \beta_{t+1}^\xi &= \beta_t^\xi + 2 \sum_{\tilde{\boldsymbol{x}}_j \in \mathcal{V}_{t+1}^\xi} \kappa(\boldsymbol{x}_t, \tilde{\boldsymbol{x}}_j) - \kappa(\boldsymbol{x}_t, \boldsymbol{x}_t), \\ \gamma_{t+1} &= \gamma_t + \sum_{\tilde{\boldsymbol{x}}_j \in \mathcal{V}_{t+1}^{\bar{\xi}}} \kappa(\boldsymbol{x}_t, \tilde{\boldsymbol{x}}_j). \end{aligned} \tag{3}$$

For the delete operation, i.e., $\mathcal{V}_{t+1}^\xi = \mathcal{V}_t^\xi \setminus \{\tilde{\boldsymbol{x}}_{r_t}\}$, we use

$$\begin{aligned} \beta_{t+1}^\xi &= \beta_t^\xi - 2 \sum_{\tilde{\boldsymbol{x}}_j \in \mathcal{V}_t^\xi} \kappa(\tilde{\boldsymbol{x}}_{r_t}, \tilde{\boldsymbol{x}}_j) + \kappa(\tilde{\boldsymbol{x}}_{r_t}, \tilde{\boldsymbol{x}}_{r_t}), \\ \gamma_{t+1} &= \gamma_t - \sum_{\tilde{\boldsymbol{x}}_j \in \mathcal{V}_{t+1}^{\bar{\xi}}} \kappa(\tilde{\boldsymbol{x}}_{r_t}, \tilde{\boldsymbol{x}}_j). \end{aligned} \tag{4}$$

[1] For the complexity analysis, we focus on the number of rounds and omit the feature dimension d in this paper.

4 New Online Kernel Selection Approach

In this section, we present the new online kernel selection approach using ISKA and apply it to online kernel learning.

4.1 Integration with ISKA

We first formulate the *hypothesis space sketch* at round t using the basis functions corresponding to examples in the buffer as follows

$$\mathcal{H}_\kappa = \{f_t(\cdot) = \langle \boldsymbol{\omega}^{(t)}, \boldsymbol{\psi}_\kappa^{(t)}(\cdot)\rangle\},$$

where $\boldsymbol{\psi}_\kappa^{(t)}(\cdot)$ is the basis function vector

$$\boldsymbol{\psi}_\kappa^{(t)}(\cdot) = [\kappa(\cdot, \tilde{\boldsymbol{x}}_1), \kappa(\cdot, \tilde{\boldsymbol{x}}_2), \ldots, \kappa(\cdot, \tilde{\boldsymbol{x}}_{B_t})]^\intercal, \quad \tilde{\boldsymbol{x}}_i \in \mathcal{V}_t,$$

and $\boldsymbol{\omega}^{(t)}$ is the weight vector

$$\boldsymbol{\omega}^{(t)} = \left[\omega_1^{(t)}, \omega_2^{(t)}, \ldots, \omega_{B_t}^{(t)}\right]^\intercal \in \mathbb{R}^{B_t}.$$

For a new instance $(\boldsymbol{x}_t, y_t)$, we first give the predicted label $\hat{y}_t = \operatorname{sgn}(f_t(\boldsymbol{x}_t))$, receive the true label y_t, and maintain the hypothesis space sketch using the buffer updating strategy under a fixed budget B. Specifically, let $B_t^+ \leq B/2$ and $B_t^- \leq B/2$, inspired by the reservoir sampling policy [Vitter, 1985; Zhao *et al.*, 2011; Hu *et al.*, 2015], when the size of the buffer is smaller than the budget we insert the new example into the buffer, i.e., $\mathcal{V}_{t+1}^\xi = \mathcal{V}_t^\xi \cup \{\boldsymbol{x}_t\}$, otherwise invoke the BUFFERUPDATING process according to a Bernoulli distribution

$$\Pr[X_t = 1] = \min\left\{\frac{B}{2t}, 1\right\}, \tag{5}$$

where $X_t = 1$ indicates that we perform the BUFFERUPDATING process at round t, including the following two steps.

- INSERT

  Different from the existing notion of coherence [Tropp, 2004], we consider the correlation in positive and negative classes separately, which can reduce the computational cost with theoretical guarantee. At round t, we define the *Subclass Coherence (SC)* for $\kappa(\cdot, \boldsymbol{x}_t)$ as

  $$\tau_t^\xi := \max_{\tilde{\boldsymbol{x}}_j \in \mathcal{V}_t^\xi} \left|\langle \kappa(\cdot, \boldsymbol{x}_t), \kappa(\cdot, \tilde{\boldsymbol{x}}_j)\rangle_{\mathcal{H}_\kappa}\right|,$$

  where ξ denotes $+$ for $y_t = +1$ or $-$ for $y_t = -1$.

  Given the threshold μ^ξ, if the SC condition holds, i.e.,

  $$\tau_t^\xi < \mu^\xi, \tag{6}$$

  we insert $\boldsymbol{x}_t$ into the buffer after the delete operation.

- DELETE

  If the SC condition holds at the t-th round, we choose the example $\tilde{\boldsymbol{x}}_{r_t}$ in the buffer $\mathcal{V}_t^\xi$ to delete via

  $$r_t = \arg\min_{i \in [B_t^\xi]} \left|\omega_i^{(t)}\right|,$$

  and update the buffer as $\mathcal{V}_{t+1}^\xi = \mathcal{V}_t^\xi \setminus \{\tilde{\boldsymbol{x}}_{r_t}\} \cup \{\boldsymbol{x}_t\}$, while $\mathcal{V}_{t+1}^{\bar{\xi}} = \mathcal{V}_t^{\bar{\xi}}$.

Finally, when the buffers are changed at round t, we first compute the ISKA criterion incrementally using (3) and (4), called INCREMENTALCOMPUTATION process. Then we evaluate the performance of each kernel in the candidate kernel set $\mathcal{K} = \{\kappa_i : i \in [N]\}$ using the median of ISKA criterion estimates over the past q ($q \geq 1$) rounds

$$\overline{\mathcal{C}}_{\mathrm{sk}}^{(t)}(\kappa, q) = \operatorname*{median}_{i \in [t-q+1:t]} \mathcal{C}_{\mathrm{sk}}^{(i)}(\kappa),$$

and select the optimal kernel at round t by

$$\kappa_{t+1}^* = \arg\max_{\kappa \in \mathcal{K}} \overline{\mathcal{C}}_{\mathrm{sk}}^{(t)}(\kappa, q).$$

4.2 Application to Online Kernel Learning

For online kernel learning, we update the hypothesis using Online Gradient Descent (OGD) method [Kivinen *et al.*, 2001; Shalev-Shwartz, 2011] to minimize the following instantaneous loss at round t

$$\mathcal{L}_t(f) = \ell(f(\boldsymbol{x}_t), y_t) + \frac{\lambda}{2}\|f\|_{\mathcal{H}_{\tilde{\kappa}}}^2, \tag{7}$$

where $\tilde{\kappa} = \kappa_{t+1}^*$ and λ denotes the regularization parameter. Denote the hypothesis of round t after selecting the optimal kernel by

$$\tilde{f}_t(\cdot) = \langle \boldsymbol{\omega}^{(t)}, \boldsymbol{\psi}_{\tilde{\kappa}}^{(t+1)}(\cdot)\rangle,$$

and the hypothesis updating using OGD can be expressed as

$$f_{t+1}(\cdot) = (1 - \eta_t\lambda)\tilde{f}_t(\cdot) - \eta_t \nabla_{\tilde{f}_t} \ell(\tilde{f}_t(\boldsymbol{x}_t), y_t),$$

where η_t denotes the stepsize at round t and

$$\nabla_{\tilde{f}_t} \ell(\tilde{f}_t(\boldsymbol{x}_t), y_t) = \ell'(\tilde{f}_t(\boldsymbol{x}_t), y_t)\tilde{\kappa}(\cdot, \boldsymbol{x}_t).$$

Then we consider two cases depending on whether the budget has been reached or not. In the first case, $B_t^\xi < B/2$. If $\boldsymbol{x}_t$ is inserted into the buffer, we set $\tilde{\boldsymbol{x}}_{B_{t+1}} = \boldsymbol{x}_t$ and update the weight vector as follows

$$\omega_i^{(t+1)} = \begin{cases} -\eta_t \ell'(\tilde{f}_t(\boldsymbol{x}_t), y_t) & i = B_{t+1}, \\ (1 - \eta_t\lambda)\omega_i^{(t)} & i \neq B_{t+1}. \end{cases} \tag{8}$$

If the buffers are not changed, choose the *surrogate example*

$$\tilde{\boldsymbol{x}}_{s_t} := \arg\max_{\tilde{\boldsymbol{x}}_j \in \mathcal{V}_t^\xi} \left|\tilde{\kappa}(\boldsymbol{x}_t, \tilde{\boldsymbol{x}}_j)\right|,$$

and update the weight vector as follows

$$\omega_i^{(t+1)} = \begin{cases} (1 - \eta_t\lambda)\omega_i^{(t)} - \eta_t \ell'(\tilde{f}_t(\boldsymbol{x}_t), y_t) & i = s_t, \\ (1 - \eta_t\lambda)\omega_i^{(t)} & i \neq s_t. \end{cases} \tag{9}$$

Next we consider the second case when $B_t^\xi = B/2$. If the buffers change we replace $\tilde{\boldsymbol{x}}_{r_t}$ with $\boldsymbol{x}_t$ and perform

$$\omega_i^{(t+1)} = \begin{cases} -\eta_t \ell'(\tilde{f}_t(\boldsymbol{x}_t), y_t) & i = r_t, \\ (1 - \eta_t\lambda)\omega_i^{(t)} & i \neq r_t, \end{cases} \tag{10}$$

otherwise we update the weight vector as in (9).

For hinge loss $\ell(\cdot, \cdot)$, the value of $\ell'(\tilde{f}_t(\boldsymbol{x}_t), y_t)$ is 0 if $y_t\tilde{f}_t(\boldsymbol{x}_t) \geq 1$ and $-y_t$ otherwise. If the loss is zero, i.e., $y_t\tilde{f}_t(\boldsymbol{x}_t) \geq 1$, we just update the weight vector using weight decay as $\boldsymbol{\omega}^{(t+1)} = (1 - \eta_t\lambda)\boldsymbol{\omega}^{(t)}$. Finally, we summarize the above stages into Algorithm 1, called OKS-ISKA.

5 Theoretical Analysis

In this section, we analyze the statistical properties of ISKA, derive the regret bound of online kernel learning using ISKA, and analyze the computational complexities of ISKA.

Algorithm 1: OKS-ISKA Algorithm

Require: The budget B, regularization parameter λ, candidate kernel set $\mathcal{K}$, subclass coherence parameters μ^+, μ^-, parameter q for median estimate

Initialize $\boldsymbol{\omega}^{(1)} = \mathbf{0}, \mathcal{C}_{\mathrm{sk}}^{(0)} = 0$ and two buffers $\mathcal{V}_1^+ = \emptyset, \mathcal{V}_1^- = \emptyset$
for $t = 1, \ldots, T$ **do**
  Compute $\boldsymbol{\psi}_{\kappa_t^*}^{(t)}(\boldsymbol{x}_t) = [\kappa_t^*(\boldsymbol{x}_t, \tilde{\boldsymbol{x}}_1), \ldots, \kappa_t^*(\boldsymbol{x}_t, \tilde{\boldsymbol{x}}_{B_t})]^\intercal$
  Compute $f_t(\boldsymbol{x}_t) = \langle \boldsymbol{\omega}^{(t)}, \boldsymbol{\psi}_{\kappa_t^*}^{(t)}(\boldsymbol{x}_t) \rangle$
  Predict $\hat{y}_t = \mathrm{sgn}(f_t(\boldsymbol{x}_t))$
  if $y_t f_t(\boldsymbol{x}_t) < 1$ **then**
    Set ξ as $+$ for $y_t = +1$ or $-$ for $y_t = -1$
    if $B_t^\xi < B/2$ **then**
      $\mathcal{V}_{t+1}^\xi = \mathcal{V}_t^\xi \cup \{\boldsymbol{x}_t\}$
    else
      Sample X with $\Pr(X = 1) = B/(2t)$
      if $X = 1$ **then**
        $(\mathcal{V}_{t+1}^\xi, \tilde{\boldsymbol{x}}_{r_t}) = \textsc{BufferUpdating}(\mathcal{V}_t^\xi, \boldsymbol{x}_t)$
      end if
    end if
    if the buffers change **then**
      $\mathcal{C}_{\mathrm{sk}}^{(t)}(\kappa) = \textsc{IncrementalComputation}(\mathcal{C}_{\mathrm{sk}}^{(t-1)}, \boldsymbol{x}_t, \tilde{\boldsymbol{x}}_{r_t})$
      Select the optimal kernel $\kappa_{t+1}^* = \arg\max_{\kappa \in \mathcal{K}} \overline{\mathcal{C}}_{\mathrm{sk}}^{(t)}(\kappa, q)$
    end if
    Update $\boldsymbol{\omega}^{(t+1)}$ via OGD
  else
    $\boldsymbol{\omega}^{(t+1)} = (1 - \eta_t \lambda)\boldsymbol{\omega}^{(t)}$
  end if
end for

5.1 Unbiasedness and Regret Analysis

We demonstrate the expectation and probabilistic error bound of ISKA as shown in Theorem 1.

Theorem 1. *For Gaussian kernel, the median estimate of ISKA is an unbiased estimate of MMD* [2]*, i.e.,*

$$\mathbb{E}\left[\overline{\mathcal{C}}_{\mathrm{sk}}^{(t)}(\kappa, q)\right] = \mathcal{C}_{\mathrm{md}}(\kappa).$$

And with probability at least $1 - 2\exp\left(-(1-2\delta)^2 q/2\right)$

$$\left|\overline{\mathcal{C}}_{\mathrm{sk}}^{(t)}(\kappa, q) - \mathcal{C}_{\mathrm{md}}(\kappa)\right| \leq 4\sqrt{\ln(4/\delta^2)} B^{-\frac{1}{2}}. \quad (11)$$

Remark 1. *Theorem 1 indicates that* (11) *holds with probability at least* $1 - \bar{\delta}$ *provided* $q = \Theta(\ln(1/\bar{\delta}))$.

Due to the incremental computation of ISKA, we may obtain different optimal kernel at each round, and perform OGD in varying RKHS. For regret analysis, we construct a surrogate hypothesis space $\mathcal{H}_{\hat{\kappa}}$ that contains all the candidate kernel functions and derive the norm bound in Theorem 2.

Theorem 2. *For Gaussian kernel* $\kappa(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp(-\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2/(2\sigma^2))$*, given candidate kernel set* $\mathcal{K} = \{\kappa_i : i \in [N]\}$ *with kernel widths* $\sigma_1, \sigma_2, \ldots, \sigma_N$*, define*

$$\sigma_{\min} = \min_{1 \leq i \leq N} \{\sigma_i\}, \quad \sigma_{\max} = \max_{1 \leq i \leq N} \{\sigma_i\}.$$

Then the RKHS $\mathcal{H}_{\hat{\kappa}}$ *induced by* $\hat{\kappa}$ *with* $\hat{\sigma} = \sigma_{\min}$ *contains all the candidate kernel functions, and* $\|\kappa_i(\cdot, \boldsymbol{x})\|_{\mathcal{H}_{\hat{\kappa}}}^2 \leq \theta$*,*

$$\kappa_i(\boldsymbol{x}_1, \boldsymbol{x}_2) \leq \langle \kappa_i(\cdot, \boldsymbol{x}_1), \kappa_i(\cdot, \boldsymbol{x}_2) \rangle_{\mathcal{H}_{\hat{\kappa}}} \leq \theta,$$

where

$$\theta = \left[2\left(\sigma_{\min}/\sigma_{\max}\right)^2 - \left(\sigma_{\min}/\sigma_{\max}\right)^4\right]^{-\frac{d}{2}}. \quad (12)$$

[2] We omit the constant terms that do not affect the kernel selection results.

In contrast to the regret analysis for online kernel learning using a fixed kernel [Cesa-Bianchi *et al.*, 2010], we derive the regret bound for online kernel selection with varying kernels in $\mathcal{H}_{\hat{\kappa}}$ using a novel definition of weight degradation. We define the *weight degradation* caused by online kernel selection and hypothesis updating at round t as follows

$$\Lambda_t := (1 - \eta_t \lambda) f_t(\cdot) - \eta_t \nabla_{f_t} \ell(f_t(\boldsymbol{x}_t), y_t) - f_{t+1}(\cdot), \quad (13)$$

where $f_t(\cdot) = \langle \boldsymbol{\omega}^{(t)}, \boldsymbol{\psi}_{\kappa_t^*}^{(t)}(\cdot) \rangle$ uses the optimal kernel κ_t^* of round $t - 1$. Then we give the upper bound of the average gradient error

$$\overline{E} := \sum_{t=1}^{T} \|E_t\|_{\mathcal{H}_{\hat{\kappa}}}/T,$$

where $E_t = \Lambda_t/\eta_t$, and demonstrate the regret bound of online kernel learning as follows.

Theorem 3. *Let* $\nu = \max\{P/\lambda, 1/\sqrt{\lambda}\}$, $\eta_t = 1/(t\lambda)$, $\mu_{\min} = \min\{\mu^+, \mu^-\}$, θ *be the norm bound in* (12)*. Assume* $\langle \kappa_t^*(\cdot, \boldsymbol{x}_t), \kappa_t^*(\cdot, \tilde{\boldsymbol{x}}_{s_t}) \rangle_{\mathcal{H}_{\hat{\kappa}}} \geq Z$ *for any surrogate example* $\tilde{\boldsymbol{x}}_{s_t}$ *while not invoking the* BufferUpdating *process, and* $\|E_t\|_{\mathcal{H}_{\hat{\kappa}}} + 1 \leq P$*. For problem* (7) *with the hinge loss and the Gaussian kernel, let* $f^* = \arg\min_{f \in \mathcal{H}_{\hat{\kappa}}} \sum_{t=1}^{T} \mathcal{L}_t(f)$ *using the optimal kernel selected by MMD,* ζ *be the maximum size that the buffer may reach without budget, and* $f_t, t \in [T]$ *be the sequence of hypotheses generated by* (8)*,* (9) *and* (10)*. Then there is a constant* $C \geq 0$ *such that* $\mathbb{E}[\overline{E}] \leq \hat{E}/T$*, and*

$$\mathbb{E}\left[\sum_{t=1}^{T} \left(\mathcal{L}_t(f_t) - \mathcal{L}_t(f^*)\right)\right] \leq \frac{(\lambda\nu + P)^2}{2\lambda} O(\ln T) + 2\nu\hat{E},$$

where

$$\hat{E} := \frac{\lambda B C \zeta}{2} + \left[T - \frac{B \ln T + B}{2}\right]\sqrt{2\theta - 2Z} + \frac{(BT - B\zeta)[\ln T + 1]}{\sqrt{2}T}\sqrt{\theta - \mu_{\min}} + \frac{B\zeta}{2}\theta.$$

Proof Sketch. For the hinge loss, the weight degradation in (13) can be expressed as follows

$$\Lambda_t = (1 - \eta_t \lambda) f_t(\cdot) + \eta_t y_t \kappa_t^*(\cdot, \boldsymbol{x}_t) - f_{t+1}(\cdot).$$

We decompose the weight degradation Λ_t into two terms $\Lambda_t = \Gamma_t + \Delta_t$, where

$$\Gamma_t = (1 - \eta_t \lambda)(f_t(\cdot) - \tilde{f}_t(\cdot)) + \eta_t y_t (\kappa_t^*(\cdot, \boldsymbol{x}_t) - \tilde{\kappa}(\cdot, \boldsymbol{x}_t))$$

is the weight degradation of incremental computation for ISKA,

$$\Delta_t = (1 - \eta_t \lambda)\tilde{f}_t(\cdot) + \eta_t y_t \tilde{\kappa}(\cdot, \boldsymbol{x}_t) - f_{t+1}(\cdot)$$

is that of buffer updating and hypothesis updating. Then we derive the upper bound of the average gradient error and complete the proof by Theorem 1 and Theorem 2. □

Remark 2. *By Proposition 3 in [Richard* et al.*, 2009], our SC condition is a sufficient condition of the Approximate Linear Dependency (ALD) condition. Since the eigenvalues of Gaussian kernel matrix decay exponentially to* 0 *[Caponnetto and De Vito, 2007], the maximum size of the buffer without budget via ALD is of order* $O(\ln T)$ *[Sun* et al.*, 2012], which yields*

$\zeta = O(\ln T)$. *Then let* $Z = \theta - (\ln T/T)^2$*, we can obtain* $\hat{E} = O(\ln T)$*, and the regret bound of order* $O(\ln T)$ *in Theorem 3 that is the optimal regret bound for strongly convex objective function and OGD [Hazan, 2016].*

5.2 Computational Complexities of ISKA

When applying the proposed ISKA to online kernel selection with N candidate kernels, the maintenance of ISKA criterion includes two operations at each updating step: (a) the buffer updating is in $O(B)$ time complexity, which is a linear time complexity with respect to the budget; (b) the incremental computation is in $O(NB)$ time complexity[3]. Since the updating steps are according to the probability distribution in (5), the expectation of the number of the updates is $O(B \ln T)$ for ISKA. Thus, the overall time complexity of ISKA is $O\left(NB^2 \ln T\right)$.

Table 1 summarizes the time complexities of KA and our ISKA after T rounds. Since only β_t^+, β_t^- and γ_t need to be stored for each candidate kernel, the space complexity of ISKA is $O(B+N)$, where $B, N \ll T$ and both are constant.

KA	ISKA			
	# Updates	BU	IC	Total
$O(NT^2)$	$O(B \ln T)$	$O(B)$	$O(NB)$	$O\left(NB^2 \ln T\right)$

Table 1: Time complexities of kernel alignment criteria (BU: buffer updating per round; IC: incremental computation per round; N: size of the candidate kernel set; T: number of rounds; B: budget).

6 Experiments

This section empirically evaluates the proposed OKS-ISKA for online kernel learning.

6.1 Experimental Setups

Algorithms were implemented in R 3.3.2 on a machine with 4-core Intel Core i7 3.60 GHz CPU and 16GB memory. All the experiments were performed over 20 different random permutations of the datasets. We merged the training and testing data into a single dataset for each benchmark dataset[4], and compared the proposed OKS-ISKA with the following state-of-the-art online kernel selection algorithms.

- **Online Kernel Selection** (OKS) [Yang *et al.*, 2012]: OKS is a randomized kernel selection algorithm. It learns a probability distribution for each candidate kernel by which the optimal kernel is selected.
- **Online Kernel Learning with Adaptive Kernel** [Chen *et al.*, 2016]: It updates the hypothesis and the kernel width by OGD method to minimize the squared loss. We replace the squared loss with the hinge loss for fair comparison, which we refer to as **OKL-GD**[5].
- **Projectron++** [Orabona *et al.*, 2008]: Projectron++ is a budgeted online kernel learning algorithm using a projection strategy that is equivalent to ALD condition. For baseline, similar to the two-stage method [Cortes *et al.*, 2012], we first select a kernel via KA criterion in (1) and then run Projectron++ with the selected kernel, which we refer to as **Proj-KA**.[6]

A set of Gaussian kernels with kernel widths $\sigma \in \{2^{-(i+1)/2}, i = [-12 : +2 : 12]\}$ was adopted as the candidate kernel set and the kernel widths of OKL-GD were restricted to the same range. The initial parameter i of the kernel width was selected in $\{-12, -10, -8\}$ uniformly, since small kernel width may lead to the vanishing of the initial gradient of OKL-GD. We tuned the stepsize of OGD in a range $10^{[-5:+1:0]}$ and the regularization parameter λ in a range $10^{[-4:+1:1]}$. For our OKS-ISKA, we set $q = 5$ for the median estimate, $\mu^{\xi} = 0.3$, $B = 150$ for small datasets ($T < 10,000$) and $B = 200$ for other datasets according to the experimental analysis in "Parameter Influence." Besides, we set $\eta_t = 1/(t\lambda)$ as in Theorem 3. For fair comparison, we ran OKS with single pass over the data and set the smoothing parameter $\bar{\delta} = 0.2$, stepsize of weight updating $\eta = \sqrt{2(1-\bar{\delta})\ln N/(NT)}$ as in [Yang *et al.*, 2012]. For Projectron++, we use hinge loss for fair comparison, and set the projection parameter $U = 1/4\sqrt{(B+1)/\log(B+1)}$ as in [Orabona *et al.*, 2008]. The mistake rate is used to evaluate the accuracy of online kernel selection approaches for online kernel learning, which is computed by $\sum_{t=1}^{T} I(y_t f_t(\boldsymbol{x}_t) < 0)/T \times 100$, and the mean of mistake rates over the 20 runs is referred to as average mistake rate.

6.2 Performance Evaluation

Table 2 lists the experimental results in terms of the mistake rate and running time for online kernel learning. We can summarize the results as follows: (a) For small datasets, OKS-ISKA performs better than the other kernel selection algorithms in terms of the mistake rates, while having a slightly lower efficiency due to the frequent updates of the buffers for OKS-ISKA at the initial period; (b) For large datasets, OKS-ISKA is significantly more efficient than the other algorithms while preserving accuracy; (c) The offline computation of kernel alignment for Proj-KA consumes much more time than the incremental computation for our ISKA.

Figure 1 depicts the convergence behaviors in terms of the mistake rates and running time on `splice`. Since Proj-KA adopted the offline kernel selection, we do not include its running time curve. The results show that the OKS-ISKA is significantly more efficient after the initial period.

6.3 Parameter Influence

We further conducted experiments using different initial kernel widths, budgets and SC conditions.

Figure 2 shows the mistake rate and runtime on `spambase` in terms of initial kernel widths in a range $i \in$

[3]Since we set $q < N$ in the experiments, we omit q in the complexity analysis for ISKA.

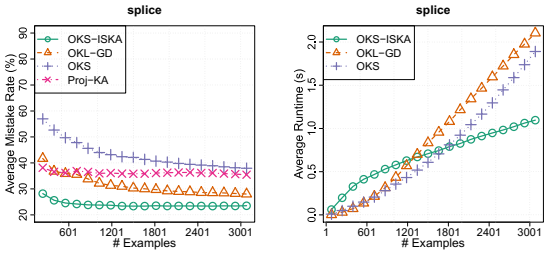
[4]http://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/ http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[5]To prevent the curse of kernelization, we keep at most 1000 examples in memory when learning using OKL-GD.

[6]Due to the high computational complexities, we randomly sample 10000 examples for Proj-KA to select the optimal kernel in the experiments on large datasets.

Dataset	OKS-ISKA		OKL-GD		OKS		Proj-KA	
	Mistake rate (%)	Time (s)	Mistake rate (%)	Time (s)	Mistake rate (%)	Time (s)	Mistake rate (%)	Time (s)
german	**30.012 ± 0.767**	0.373	35.202 ± 1.813	**0.223**	42.207 ± 1.377	0.303	39.280 ± 1.119	0.889
svmguide3	**22.684 ± 0.625**	0.368	23.263 ± 0.338	0.341	29.572 ± 0.963	**0.304**	26.641 ± 1.282	1.787
spambase	**28.876 ± 0.942**	**1.236**	33.136 ± 2.323	3.403	34.055 ± 0.263	3.641	29.723 ± 0.791	19.120
splice	**23.616 ± 1.855**	**1.112**	28.057 ± 5.097	2.182	37.814 ± 2.154	1.986	35.389 ± 6.478	11.262
mushrooms	**0.360 ± 0.059**	**2.293**	9.652 ± 1.725	21.430	8.003 ± 0.328	8.938	0.377 ± 0.014	92.592
a9a	**17.293 ± 0.420**	**26.023**	23.930 ± 0.001	172.202	23.132 ± 0.101	1033.891	20.893 ± 0.097	183.605
w7a	**2.582 ± 0.083**	**83.061**	2.973 ± 0.125	537.550	6.926 ± 0.073	1134.709	2.632 ± 0.078	364.350
w8a	2.570 ± 0.074	**98.530**	2.912 ± 0.111	633.350	6.911 ± 0.072	1453.720	**2.486 ± 0.075**	415.682
corrected	**3.737 ± 0.341**	**268.341**	4.452 ± 0.315	1256.462	5.505 ± 0.045	13930.106	4.323 ± 0.377	397.781
cod-rna	**11.615 ± 0.002**	**41.230**	12.603 ± 0.450	68.030	12.727 ± 0.124	1852.235	15.142 ± 0.255	115.877

Table 2: Performances of online kernel selection approaches for online kernel learning on benchmark datasets.



(a) average mistake rate (b) average runtime

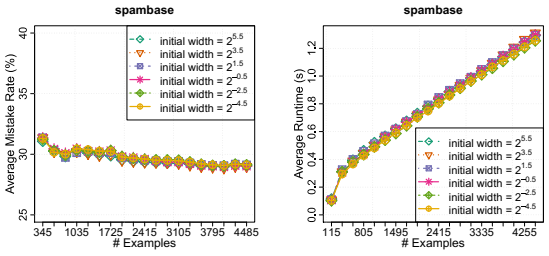
Figure 1: Convergence behaviors w.r.t. the mistake rate and runtime on benchmark datasets.

$\{-12, -8, -4, 0, 4, 8\}$. We can observe that the initial kernel width has almost no effect on the performance of OKS-ISKA. Thus, OKS-ISKA is robust to the initial kernel.

From Figure 3, we observe that, for our OKS-ISKA, the buffers with budget $B = 200$ perform as well as those with larger budget with respect to the mistake rate, while having less runtime. Thus, OKS-ISKA is suitable for large-scale online kernel learning. Besides, a smaller threshold μ^ξ of SC condition incurs a better performance in terms of both accuracy and efficiency, which demonstrates that our buffer updating strategy is effective for online kernel selection.

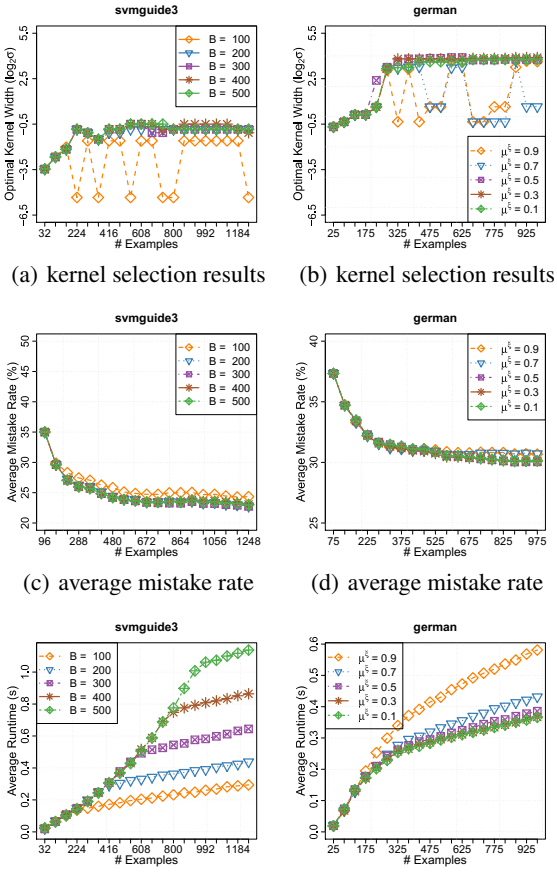
7 Conclusion

Kernel selection is critical to kernel methods, and online kernel selection is faced with some new challenges. In this paper, we have proposed the novel online kernel selection approach that is theoretically sound and computationally efficient. The proposed approach gives an unbiased estimate of the maximum mean discrepancy, enjoys the optimal regret bound for online kernel learning, and dynamically maintains the hypothesis space and selects the optimal kernel at each round, while the maintenance time complexity is constant and the overall time complexity is logarithmic. We conclude that the proposed online kernel selection approach meets the new challenges of online kernel selection and is promising for both online and offline large-scale model selection.



(a) average mistake rate (b) average runtime

Figure 2: The effect of initial kernel width on the mistake rate and runtime of OKS-ISKA.



(a) kernel selection results (b) kernel selection results

(c) average mistake rate (d) average mistake rate

(e) average runtime (f) average runtime

Figure 3: Kernel selection results, average mistake rate, average runtime w.r.t. budget B and μ^ξ in the SC condition (6).

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (No. 61673293 and No. 61703396), and the State Key Development Program of China (No. 2017YFE0111900).

References

[Anguita *et al.*, 2009] Davide Anguita, Alessandro Ghio, Sandro Ridella, and Dario Sterpi. k-fold cross validation for error rate estimate in support vector machines. In *Proceedings of the 2009 International Conference on Data Mining*, pages 291–297, 2009.

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

[Caponnetto and De Vito, 2007] A. Caponnetto and E. De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

[Cesa-Bianchi *et al.*, 2010] Nicolò Cesa-Bianchi, Shai Shalev-Shwartz, and Ohad Shamir. Online learning of noisy data with kernels. In *Proceedings of the 23rd Conference on Learning Theory*, pages 218–230, 2010.

[Chen *et al.*, 2016] Badong Chen, Junli Liang, Nanning Zheng, and José C Príncipe. Kernel least mean square with adaptive kernel size. *Neurocomputing*, 191:95–106, 2016.

[Cortes *et al.*, 2012] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *Journal of Machine Learning Research*, 13:795–828, 2012.

[Foster *et al.*, 2015] Dylan J Foster, Alexander Rakhlin, and Karthik Sridharan. Adaptive online learning. In *Advances in Neural Information Processing Systems 28*, pages 3375–3383, 2015.

[Foster *et al.*, 2017] Dylan J Foster, Satyen Kale, Mehryar Mohri, and Karthik Sridharan. Parameter-free online learning via model selection. In *Advances in Neural Information Processing Systems 30*, pages 6022–6032, 2017.

[Gretton *et al.*, 2012] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.

[Hazan, 2016] Elad Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

[Hu *et al.*, 2015] Junjie Hu, Haiqin Yang, Irwin King, Michael R Lyu, and Anthony Man-Cho So. Kernelized online imbalanced learning with fixed budgets. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 2666–2672, 2015.

[Kandola *et al.*, 2002] J. Kandola, J. Shawe-Taylor, and N. Cristianini. On the extensions of kernel alignment. Technical Report Technical report 120, University of London, 2002.

[Kivinen *et al.*, 2001] Jyrki Kivinen, Alex J Smola, and Robert C Williamson. Online learning with kernels. In *Advances in Neural Information Processing Systems 14*, pages 785–792, 2001.

[Liu and Liao, 2015] Yong Liu and Shizhong Liao. Eigenvalues ratio for kernel selection of kernel methods. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 2814–2820, 2015.

[Nguyen *et al.*, 2017] Tu Dinh Nguyen, Trung Le, Hung Bui, and Dinh Q. Phung. Large-scale online kernel learning with random feature reparameterization. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 2543–2549, 2017.

[Orabona *et al.*, 2008] Francesco Orabona, Joseph Keshet, and Barbara Caputo. The projectron: A bounded kernel-based perceptron. In *Proceedings of the 25th International Conference on Machine learning*, pages 720–727, 2008.

[Reisinger *et al.*, 2008] Joseph Reisinger, Peter Stone, and Risto Miikkulainen. Online kernel selection for Bayesian reinforcement learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 816–823, 2008.

[Richard *et al.*, 2009] Cédric Richard, José Carlos M Bermudez, and Paul Honeine. Online prediction of time series data with kernels. *IEEE Transactions on Signal Processing*, 57(3):1058–1067, 2009.

[Shalev-Shwartz, 2011] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2011.

[Sinha and Duchi, 2016] Aman Sinha and John C Duchi. Learning kernels with random features. In *Advances in Neural Information Processing Systems 29*, pages 1298–1306, 2016.

[Sun *et al.*, 2012] Yi Sun, Faustino Gomez, and Jürgen Schmidhuber. On the size of the online kernel sparsification dictionary. In *Proceedings of the 29th International Conference on Machine Learning*, pages 329–336, 2012.

[Tropp, 2004] Joel A Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004.

[Vitter, 1985] Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software*, 11(1):37–57, 1985.

[Yang *et al.*, 2012] Tianbao Yang, Mehrdad Mahdavi, Rong Jin, Jinfeng Yi, and Steven C.H. Hoi. Online kernel selection: Algorithms and evaluations. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, pages 22–26, 2012.

[Zhao *et al.*, 2011] Peilin Zhao, Rong Jin, Tianbao Yang, and Steven C.H. Hoi. Online AUC maximization. In *Proceedings of the 28th International Conference on Machine Learning*, pages 233–240, 2011.