High-probability Kernel Alignment Regret Bounds for Online Kernel Selection ⋆

Shizhong Liao[0000−0003−0594−7116] and Junfan Li[0000−0003−1027−4251] (✉)

College of Intelligence and Computing, Tianjin University,
Tianjin 300350, China
{szliao,junfli}@tju.edu.cn

Abstract. In this paper, we study data-dependent regret bounds for online kernel selection in the regime online classification with the hinge loss. Existing work only achieves $O(\|f\|^2_{\mathcal{H}_\kappa}T^\alpha)$, $\frac{1}{2}\le\alpha<1$ regret bounds, where $\kappa\in\mathcal{K}$, a preset candidate set. The worst-case regret bounds can not reveal kernel selection improves the performance of single kernel leaning in some benign environment. We develop two adaptive online kernel selection algorithms and obtain the first high-probability regret bound depending on $\mathcal{A}(\mathcal{I}_T,\kappa)$, a variant of kernel alignment. If there is a kernel in the candidate set matching the data well, then our algorithms can improve the learning performance significantly and reduce the time complexity. Our results also justify using kernel alignment as a criterion for evaluating kernel function. The first algorithm has a $O(T/K)$ per-round time complexity and enjoys a $O(\|f\|^2_{\mathcal{H}_{i^*}}\sqrt{K\mathcal{A}(\mathcal{I}_T,\kappa_{i^*})})$ high-probability regret bound. The second algorithm enjoys a $\tilde{O}(\beta^{-1}\sqrt{T\mathcal{A}(\mathcal{I}_T,\kappa_{i^*})})$ per-round time complexity and achieves a $\tilde{O}(\|f\|^2_{\mathcal{H}_{i^*}}K^{\frac{1}{2}}\beta^{\frac{1}{2}}T^{\frac{1}{4}}\mathcal{A}(\mathcal{I}_T,\kappa_{i^*})^{\frac{1}{4}})$ high-probability regret bound, where $\beta\ge1$ is a balancing factor and $\kappa_{i^*}\in\mathcal{K}$ is the kernel with minimal $\mathcal{A}(\mathcal{I}_T,\kappa)$.

Keywords: Model Selection · Online learning · Kernel method.

1 Introduction

Model selection aims at choosing inductive bias that matches learning tasks, and thus is central to the learning performance of algorithms. For online kernel learning, one of the model selection problems is how to choose a suitable RKHS (or kernel function), in which the data are represented with a low complexity. A simple representation of the data makes algorithms enjoy superior learning performance. This problem is also termed as online kernel selection, related to the more general online model selection [7, 16]. An adversary sends a learner a sequence of examples $\{(\mathbf{x}_t,y_t)\}_{t=1}^T$. The learner chooses a sequence of kernels $\{\kappa_{I_t}\}_{t=1}^T$ from a preset kernel space $\mathcal{K}$, and a sequence of hypotheses $\{f_t\}_{t=1}^T$. At

⋆ This work was supported in part by the National Natural Science Foundation of China under grants No. 62076181.

each round t, the loss is $\ell(f_t(\mathbf{x}_t), y_t)$. The learner should be competitive with the unknown optimal RKHS, $\mathcal{H}_{i^*}$. We use the regret to measure the performance,

$$\mathrm{Reg}_T(\mathcal{H}_{i^*}) := \sum_{t=1}^{T} \ell(f_t(\mathbf{x}_t), y_t) - \min_{f\in\mathcal{H}_{i^*}} \sum_{t=1}^{T} \ell(f(\mathbf{x}_t), y_t). \tag{1}$$

$\kappa_{i^*} \in \mathcal{K}$ is the optimal kernel for the data and induces $\mathcal{H}_{i^*}$. To this end, a stronger guarantee is to adapt to any $\mathcal{H}_\kappa, \kappa \in \mathcal{K}$ up to a small cost.

To achieve a sub-linear regret bound with respect to (w.r.t) any $\mathcal{H}_\kappa$, the main challenge is the high time complexity. The per-round time complexity of evaluating kernel functions and making prediction would be $O(KT)$, if we do not limit the model size, where K is the number of base kernels. Most of existing online kernel selection researches focus on achieving a $O(\|f\|^2_{\mathcal{H}_\kappa} T^\alpha)$, $\alpha < 1$ regret bound, and keeping a constant per-round time complexity. One of approaches embeds implicit RKHSs to relatively low-dimensional random feature spaces [17, 13, 19], in which the time complexity of evaluating kernel functions and prediction is linear with D, the number of random features. The algorithm proposed in [13] has a $O(\|f\|^2_{\mathcal{H}_\kappa} K^{\frac{1}{3}} T^{\frac{2}{3}})$ expected regret bound and suffers a $O(D)$ time complexity. Similarly, an algorithm with a $O(\|f\|^2_{\mathcal{H}_\kappa}\sqrt{T})$ regret bound and a $O(KD)$ time complexity was proposed in [19]. The other approach maintains a fixed budget with size B [24]. An algorithm with a $O(B\ln T)$ regularized regret bound (or a $\tilde{O}(\|f\|^2_{\hat{\mathcal{H}}} T^{\frac{2}{3}} + BT^{\frac{1}{3}})$ standard regret bound by setting λ to the optimal value $O(T^{-\frac{1}{3}})$ in [24]) and a $O(B + KB^2/T)$ time complexity was proposed, where $\hat{\mathcal{H}}$ is a surrogate hypothesis space.

However, the $O(\|f\|^2_{\mathcal{H}_\kappa} T^\alpha)$ regret bound is pessimistic in the sense that (i) it can not distinguish the convergence rate w.r.t. T when choosing different kernel; (ii) it can not reveal kernel selection improves the learning performance in some benign environment. Recalling that the cumulative losses of algorithms are upper bounded by $\min_{f\in\mathcal{H}_\kappa}\sum_{t=1}^{T}\ell(f(\mathbf{x}_t), y_t) + O(\|f\|^2_{\mathcal{H}_\kappa} T^\alpha)$. If the minimal cumulative losses in $\mathcal{H}_\kappa$ are small, then the regret bound is the dominated term and is hard to compare among different base kernels. To resolve the two issues, we should require regret bounds adapting to the data complexity in each RKHS. Thus a fundamental problem of online kernel selection is how to provide data-dependent regret bounds. It would be easy to solve the problem without considering the computational constraints. Our question is whether it is possible to achieve the two goals simultaneously. In this paper, we answer the question affirmatively.

We define a variant of kernel alignment, denoted by $\mathcal{A}(\mathcal{I}_T, \kappa)$, for measuring the complexity of data represented in $\mathcal{H}_\kappa$, which reveals the matching between the label matrix and kernel matrix. Different kernel embeds the instances into different RKHS, and thus induces different data complexity. A good kernel should be the one that represents data simply. We establish two computationally efficient algorithms achieving high-probability kernel alignment regret bounds. The first algorithm achieves a $O(\|f\|^2_{\mathcal{H}_{i^*}}\sqrt{K\mathcal{A}(\mathcal{I}_T, \kappa_{i^*})})$ regret and suffers a $O(T/K)$ per-round time complexity. The second algorithm enjoys a favorable regret-performance trade-off, which can provide a $\tilde{O}(\|f\|^2_{\mathcal{H}_{i^*}} K^{\frac{1}{2}}\beta^{\frac{1}{2}} T^{\frac{1}{4}} \mathcal{A}(\mathcal{I}_T, \kappa_{i^*})^{\frac{1}{4}})$ regret bound and suffer a $\tilde{O}(\beta^{-1}\sqrt{T\mathcal{A}(\mathcal{I}_T,\kappa_{i^*})})$ time complexity, where $\beta \geq 1$ is a balancing factor. The algorithms are based on the adaptive and optimistic online mirror descent framework and two novel model evaluation strategies. We also reveal a new result: if there is a good kernel in the candidate set, then online kernel selection can improve the learning performance and computational efficiency significantly relative to online kernel learning using a bad kernel. Numerical experiments on benchmark datasets are conducted to verify our theoretical results.

1.1 Related work

Yang et al. [22] proposed the first online kernel selection algorithm, OKS, for alleviating the high time complexity of offline kernel selection and multi-kernel learning. For online kernel selection, OKS can provide a $O(\|f\|^2_{\mathcal{H}_\kappa}\sqrt{KT})$ regret bound and suffers a $O(T)$ per-round time complexity. Foster et al. [6] studied online model selection in Banach space, and proposed a multi-scale expert advice algorithm which achieves regret bounds scaling with the loss range of individual hypothesis space. The multi-scale algorithm can achieve data-dependent regret bounds, only if there are computationally efficient sub-algorithms. A related but different work is online multi-kernel learning [11, 19], where algorithms make a prediction $f_t(\mathbf{x}_t)$ by a convex combination of K base predictions $\{f_{t,i}(\mathbf{x}_t)\}_{i=1}^K$. Existing algorithms can also not achieve data-dependent regret bounds.

Another related research is achieving data-dependent regret bounds for online kernel learning. If the loss function satisfies specifical curvature property [23, 2, 10], such as the square loss and the logistic loss, then there exist computationally efficient online kernel learning algorithms that achieve regret bound depending on the smallest cumulative losses [23], or the effective dimension [2, 10]. However, the hinge loss does not enjoy the curvature property. Thus achieving data-dependent regret bounds is more difficult. Our algorithms can be applied to online kernel learning so long as $\mathcal{K}$ only contains a single kernel.

2 Problem Setting

Let $\mathcal{I}_T = \{(\mathbf{x}_t, y_t)\}_{t\in[T]}$ be a sequence of examples, where $\mathbf{x}_t \in \mathbb{R}^d$ is an instance, $y_t \in \{-1, 1\}$ and $[T] = \{1, \ldots, T\}$. Let $\kappa(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ be a positive semidefinite kernel function, and $\mathcal{K} = \{\kappa_1, \ldots, \kappa_K\}$. Assuming that $\kappa_i(\mathbf{x},\mathbf{x}) \in [1, D_i]$ for all $i \in [K]$ and $D = \max_i D_i$. Let $\mathcal{H}_i = \{f | f : \mathbb{R}^d \rightarrow \mathbb{R}\}$ be the RKHS associated with κ_i, such that (i) $\langle f, \kappa_i(\mathbf{x},\cdot)\rangle_{\mathcal{H}_i} = f(\mathbf{x})$; (ii) $\mathcal{H}_i = \mathrm{span}(\kappa_i(\mathbf{x}_t,\cdot) : t \in [T])$. We define $\langle\cdot,\cdot\rangle_{\mathcal{H}_i}$ as the inner product in $\mathcal{H}_i$, which induces the norm $\|f\|_{\mathcal{H}_i} = \sqrt{\langle f, f\rangle_{\mathcal{H}_i}}$. Let $\ell(f(\mathbf{x}), y) = \max\{0, 1 - yf(\mathbf{x})\}$ be the hinge loss, and $\mathcal{D}_{\psi_{t,i}}(\cdot,\cdot) : \mathcal{H}_i \times \mathcal{H}_i \rightarrow \mathbb{R}$ be the Bregman divergence induced by a strongly convex regularizer $\psi_{t,i}(\cdot) : \mathcal{H}_i \rightarrow \mathbb{R}$.

For a sequence $\mathcal{I}_T$, if an oracle gives the optimal kernel $\kappa_{i^*} \in \mathcal{K}$, then we can learn a sequence of hypotheses in $\mathcal{H}_{i^*}$. Lacking such prior, the learner hopes to develop a kernel selection algorithm and generate a sequence of hypotheses $\{f_t\}_{t=1}^T$, which is competitive to that generated by the same algorithm running in $\mathcal{H}_{i^*}$. The regret of the algorithm w.r.t. $\mathcal{H}_i, i \in [K]$ is defined by (1), where we replace $\mathcal{H}_{i^*}$ with $\mathcal{H}_i$. A general goal is to keep $\mathrm{Reg}_T(\mathcal{H}_i) = O(\mathrm{Poly}(\|f_i^*\|_{\mathcal{H}_i})T^{\alpha})$, $\alpha < 1$. Thus the average loss of the algorithm converges to that of the optimal hypothesis in $\mathcal{H}_{i^*}$. The worst-case optimal regret bound is obtained at $\alpha = \frac{1}{2}$.

If the minimal cumulative losses in all RKHSs are smaller than $O(\sqrt{T})$, then such a worst-case regret bound is unsatisfactory, since it can not reveal kernel selection improves the learning performance. Hence, it is necessary to establish some kind of regret bound adapting to the complexity of data represented in RKHS. The representation of $(\mathbf{x}_t, y_t)$ in $\mathcal{H}_i$ is $(\phi_i(\mathbf{x}_t), y_t)$, where ϕ_i is the feature mapping induced by κ_i. We can use the variance of the examples in $\mathcal{H}_i$, i.e, $\sum_{t=1}^{T} \|y_t\phi_i(\mathbf{x}_t, \cdot) - \mu_{T,i}\|^2_{\mathcal{H}_i}$, where $\mu_{T,i} = \frac{1}{T}\sum_{\tau=1}^{T} y_\tau \phi_i(\mathbf{x}_\tau)$, to measure the data complexity [9]. Next, we present a formal definition.

Definition 1 (Alignment). *For any sequence of examples $\mathcal{I}_T$ and kernel function κ_i, the alignment is defined as follows*

$$\mathcal{A}(\mathcal{I}_T, \kappa_i) := \sum_{t=1}^{T} \kappa_i(\mathbf{x}_t, \mathbf{x}_t) - \frac{1}{T}\mathbf{Y}_T^{\top}\mathbf{K}_{\kappa_i}\mathbf{Y}_T.$$

The alignment is an extension of *kernel polarization* [1], a classical kernel selection criterion. If $\mathbf{K}_{\kappa_i}$ is the ideal kernel matrix $\mathbf{Y}_T\mathbf{Y}_T^{\top}$, then $\mathcal{A}(\mathcal{I}_T, \kappa_i) = 0$. Thus the alignment can be a criterion for evaluating the goodness of kernel function κ_i on $\mathcal{I}_T$. Model selection aims at adapting to $\mathcal{H}_{i^*}$ induced by κ_{i^*}, the unknown optimal kernel. A natural question is that does there exist some computationally efficient algorithm that achieves high-probability regret bounds depending on $\mathcal{A}(\mathcal{I}_T, \kappa_{i^*})$? Our main contribution is to answer this question affirmatively.

3 A Nearly Optimal High-probability Regret Bound

We first show a simple algorithm achieving the kernel alignment regret bound without considering the computational complexity.

3.1 Warm-up

At a high level, our approach is based on the adaptive and optimistic online mirror descent framework (AO$_2$MD) [4, 18]. We explain AO$_2$MD in a fixed RKHS $\mathcal{H}_i$. Let $\mathbb{H}_i = \{f \in \mathcal{H}_i : \|f\|_{\mathcal{H}_i} \leq U, U \geq D\}$. At any round t, let $f_{t,i}, f'_{t-1,i} \in \mathbb{H}_i$ and $\nabla_{t,i} := \nabla_{f_{t,i}}\ell(f_{t,i}(\mathbf{x}_t), y_t)$. AO$_2$MD running in $\mathbb{H}_i$ is defined as follows,

$$f_{t,i} = \mathop{\arg\min}_{f \in \mathbb{H}_i} \left\{\langle f, \bar{\nabla}_{t,i}\rangle + \mathcal{D}_{\psi_{t,i}}(f, f'_{t-1,i})\right\}, \tag{2}$$

$$f'_{t,i} = \mathop{\arg\min}_{f \in \mathbb{H}_i} \left\{\langle f, \nabla_{t,i}\rangle + \mathcal{D}_{\psi_{t,i}}(f, f'_{t-1,i})\right\}, \tag{3}$$

where $\{f'_{t,i}\}_{t=0}^{T-1}$ is a sequence of auxiliary hypotheses. The solutions of (2) and (3) are shown in supplementary material. The main idea of AO$_2$MD is to select an optimistic estimator of $\nabla_{t,i}$, denoted by $\bar{\nabla}_{t,i}$, and execute the first mirror updating (2). After obtaining $f_{t,i}$, we output the prediction $\hat{y}_t = \mathrm{sign}(f_{t,i}(\mathbf{x}_t))$. When receiving the label y_t, we observe the true gradient $\nabla_{t,i}$ and execute the second mirror updating (3). If the data evolves slowly, then it is possible to find a good estimator $\bar{\nabla}_{t,i}$. The final regret bound depends on the cumulative difference $\sum_{t=1}^{T} \|\nabla_{t,i} - \bar{\nabla}_{t,i}\|^2_{\mathcal{H}_i}$, where we define $\bar{\nabla}_{1,i} = 0$.

A simple approach for obtaining a regret bound depending on the alignment is to reduce online kernel selection to a problem of prediction with expert advice. Let $\mathcal{E}(K)$ be an algorithm for prediction with expert advice. We can instantiate an AO$_2$MD algorithm in each $\mathbb{H}_i, i \in [K]$, and then aggregate the K algorithms with $\mathcal{E}(K)$. The following theorem shows the data-dependent regret bound.

Theorem 1. *Let $\bar{\nabla}_{t,i} = \nabla_{r_i(t),i}$ where $r_i(t) = \max_\tau\{\tau < t : y_\tau f_{\tau,i}(\mathbf{x}_t) < 1\}$ and $\mathcal{E}(K)$ be some algorithm for expert advice. There exists an online kernel selection algorithm such that, for all $\kappa_i \in \mathcal{K}$, with probability at least $1-\delta$,*

$$\mathrm{Reg}_T(\mathbb{H}_i) = O\left(\sqrt{L_T(f_i^\ast)\ln K \ln\frac{\ln(2T)}{\delta}} + (\|f_i^\ast\|^2_{\mathcal{H}_i} + 1)\sqrt{\mathcal{A}(\mathcal{I}_T, \kappa_i)}\right).$$

where $L_T(f_i^\ast) = \min_{f\in\mathbb{H}_i}\sum_{t=1}^{T}\ell(f(\mathbf{x}_t), y_t) \leq \mathcal{A}(\mathcal{I}_T, \kappa_i)$. The per-round time complexity of the algorithm is $O(TK)$.

To construct the algorithm, we just let $\mathcal{E}(K)$ be the weighted majority algorithm [3] that enjoys a high-probability small-loss regret bound (We give a proof in supplementary material). The algorithm description is presented in supplementary material due to the space limit. The $O(TK)$ per-round time complexity comes from the unbounded number of support vectors and running K AO$_2$MD algorithms. For a large number of base kernels, the time complexity is prohibitive. Thus such a simple algorithm is not practical. Next, we develop a more efficient algorithm enjoying a $O(T/K)$ per-round time complexity.

3.2 A More Efficient Algorithm

The simple algorithm in Theorem 1 evaluates all of the base kernels at each round. Thus the time complexity is linear with K. To resolve this issue, an intuitive approach is to reduce online kernel selection to a K-armed bandit problem [22]. However, such an approach induces two new technique challenges, i.e,

(i) If $\bar{\nabla}_{t,i} = \nabla_{r_i(t),i}$, then we can not obtain a regret bound depending on $\sum_{t=1}^{T}\|\nabla_{t,i} - \bar{\nabla}_{t,i}\|^2_{\mathcal{H}_i}$, which has been proved in [21].

(ii) The true gradient $\nabla_{t,i}$ can not be observed unless κ_i is selected. If we use an importance-weighted estimator, such as $\nabla_{t,i}/p_{t,i}$, then the second moment is linear with $\max_{t\in[T]}\frac{1}{p_{t,i}}$, which could be much large.

To solve the first challenge, we choose the optimistic estimator $\bar{\nabla}_{t,i} := \mu_{t-1,i}$, where $\mu_{t-1,i} = \sum_{\tau=1}^{t-1}\frac{-y_\tau}{t-1}\kappa_i(\mathbf{x}_\tau, \cdot)$, $t \geq 2$. However, computing $\mu_{t-1,i}$ requires to store all of the received examples. To avoid this issue, we use the "Reservoir Sampling (RS)" technique [20, 8] to construct an unbiased estimator of $\mu_{t-1,i}$, denoted by $\tilde{\mu}_{t-1,i}$. Let V be a fixed budget, which we call "Reservoir". At the beginning of round t, let $\tilde{\mu}_{t-1,i} = -\frac{1}{|V|}\sum_{(\mathbf{x},y)\in V} y\kappa_i(\mathbf{x},\cdot)$ and $\tilde{\mu}_{0,i} = 0$. Thus we define the optimistic estimator $\bar{\nabla}_{t,i} := \tilde{\mu}_{t-1,i}$. The RS technique constructs V as follows. At the end of round t, $(\mathbf{x}_t, y_t)$ is added into V with probability $\min\{1, \frac{M}{t}\}$, where $M > 1$ is the maximal size of V. If $|V| = M$ and we are to add the current example, then an old example should be removed uniformly. Note that we just need to maintain a single V, since $\bar{\nabla}_{t,i}, i = 1, \ldots, K$ can be computed by the same examples.

To solve the second challenge, assuming that we can pay additional computational cost for obtaining more information. In this way, we define a K-armed bandit problem with an additional observation. Next we propose a new decoupling exploration-exploitation scheme for obtaining more information. Let Δ_{K-1} be a $(K-1)$-dimensional probability simplex. At each round t,

- Exploitation: select a kernel κ_{I_t}, $I_t \sim \mathbf{p}_t \in \Delta_{K-1}$,
- Exploration: uniformly select a kernel $\kappa_{J_t} \in \mathcal{K}$.

Such an exploration procedure makes κ_{J_t} independent of κ_{I_t}. Based on the exploration procedure, we construct the following variance-reduced estimator,

$$\tilde{\nabla}_{t,i} = \frac{\nabla_{t,i} - \bar{\nabla}_{t,i}}{\mathbb{P}[i = J_t]}\mathbb{I}_{i=J_t} + \bar{\nabla}_{t,i},\ \forall i \in [K].$$

In this way, the second moment of $\tilde{\nabla}_{t,i}$ is linear with $1/\mathbb{P}[i = J_t] = K$. A more intuitive estimator should incorporate the information of κ_{I_t}. However, we abandon the gradient information ∇_{t,I_t} for the goal of keeping a $O(T/K)$ per-round time complexity. AO$_2$MD is as follows,

$$f_{t,i} = \mathop{\arg\min}_{f\in\mathbb{H}_i}\{\langle f, \bar{\nabla}_{t,i}\rangle + \mathcal{D}_{\psi_{t,i}}(f, f'_{t-1,i})\}, \tag{4}$$

$$f'_{t,i} = \mathop{\arg\min}_{f\in\mathbb{H}_i}\{\langle f, \tilde{\nabla}_{t,i}\rangle + \mathcal{D}_{\psi_{t,i}}(f, f'_{t-1,i})\}. \tag{5}$$

Let $\psi_{t,i}(f) = \frac{1}{2\lambda_{t,i}}\|f\|^2_{\mathcal{H}_i}$. Then the projection of any $g \in \mathcal{H}_i$ onto $\mathbb{H}_i$ is defined by $f = \min\{1, \frac{1}{\|g\|_{\mathcal{H}_i}}U\}g$.

Let $\mathcal{M}(K)$ be some algorithm for a K-armed bandit problem, which outputs $\mathbf{p}_t$ at the beginning of round t. We first select a kernel κ_{I_t}, $I_t \sim \mathbf{p}_t$, and compute f_{t,I_t} using the first mirror updating (4). After that, we output the prediction $\hat{y}_t = \mathrm{sign}(f_t(\mathbf{x}_t))$, where $f_t = f_{t,I_t}$. Then we explore another kernel κ_{J_t} for obtaining the gradient information ∇_{t,J_t}. The final step is to update $\mathbf{p}_t$. To this end, we define some criterion for evaluating each base kernel. Since $|f_{t,i}(\mathbf{x}_t)| = |\langle f_{t,i}, \kappa_i(\mathbf{x}_t,\cdot)\rangle| \leq U\sqrt{D}$, we have $\ell(f_{t,i}(\mathbf{x}_t), y_t) \leq 1+U\sqrt{D}$. Let $c_{t,i} = \frac{\ell(f_{t,i}(\mathbf{x}_t),y_t)}{1+U\sqrt{D}}$ be the criterion. The denominator scales the criterion to $[0,1]$. At the end of round t, we send $\mathbf{c}_t = (c_{t,1}\mathbb{I}_{I_t=1}, \ldots, c_{t,K}\mathbb{I}_{I_t=K})$ to $\mathcal{M}(K)$.

We name this approach B(AO)$_2$KS (Bandit with Additional Observations for Adaptive Online Kernel Selection) and present the pseudo-code in Algorithm 1.

Algorithm 1 B(AO)$_2$KS

```
Input: $\lambda_i, i = 1, \ldots, K$, $D$, $U$, $M$.
Initialization: $\forall \kappa_i \in \mathcal{K}$, $f'_{0,i} = 0$, $V = \emptyset$.
for $t = 1, 2, \ldots, T$ do
    Select a kernel $\kappa_{I_t} \sim \mathbf{p}_t$ ($\mathbf{p}_t$ is output by $\mathcal{M}(K)$),
    Compute $\bar{\nabla}_{t,I_t} = \frac{-1}{|V|} \sum_{(\mathbf{x},y)\in V} y\kappa_{I_t}(\mathbf{x}, \cdot)$,
    Update $f_{t,I_t}$ according to (4) and output prediction $\hat{y}_t = \mathrm{sign}(f_{t,I_t}(\mathbf{x}_t))$,
    Sample a kernel $\kappa_{J_t} \in \mathcal{K}$ uniformly,
    for $\kappa_i \in \mathcal{K}$ do
        if $\kappa_i = \kappa_{J_t}$ then
            Compute $\bar{\nabla}_{t,J_t} = \frac{-1}{|V|} \sum_{(\mathbf{x},y)\in V} y\kappa_{J_t}(\mathbf{x}, \cdot)$,
            Update $f_{t,J_t}$ according to (4) and compute $\nabla_{t,J_t}$,
        end if
        Compute estimator $\tilde{\nabla}_{t,i} = K(\nabla_{t,i} - \bar{\nabla}_{t,i}) \cdot \mathbb{I}_{i=J_t} + \bar{\nabla}_{t,i}$,
        Update $f'_{t,i}$ according to (5),
    end for
    Compute $c_{t,I_t} = \frac{1}{1+U\sqrt{D}} \max\{0, 1 - y_t f_{t,I_t}(\mathbf{x}_t)\}$,
    Send $\mathbf{c}_t = (c_{t,1}\mathbb{I}_{I_t=1}, \ldots, c_{t,K}\mathbb{I}_{I_t=K})$ to $\mathcal{M}(K)$,
    Sample a Bernoulli random variable $\delta_t \sim \mathrm{Ber}(1, M/t)$,
    if $\delta_t = 1$ and $t > M$, then sample $(\mathbf{x}_{j_t}, y_{j_t}) \in V$ and $V = V \cup (\mathbf{x}_t, y_t) \backslash (\mathbf{x}_{j_t}, y_{j_t})$,
    if $\delta_t = 1$ and $t \leq M$, then $V = V \cup (\mathbf{x}_t, y_t)$,
end for
```

3.3 Regret bound

We first establish an important technique lemma about the reservoir estimator.

Lemma 1. *Let $T > M$. For all $i = 1, \ldots, K$, with probability at least $1 - \delta$,*

$$\sum_{t=1}^{T} \|\tilde{\mu}_{t,i} - \mu_{t,i}\|^2_{\mathcal{H}_i} \leq \frac{\mathcal{A}(\mathcal{I}_T, \kappa_i)}{M} \ln \frac{T}{M} + \frac{8D_i}{3} \ln \frac{K}{\delta} + \sqrt{\frac{8D_i \mathcal{A}(\mathcal{I}_T, \kappa_i) \ln T \ln \frac{K}{\delta}}{M}}.$$

Lemma 1 is an extension of the statistic guarantee of reservoir sampling estimator in [8], where the expected unbiasedness of $\tilde{\mu}_{t,i}$ was proved. Next we present a sufficient condition for obtaining the data-dependent regret bound, which gives a strong constraint on the bandit algorithm $\mathcal{M}(K)$.

Assumption 1 *Let $c_t \in [0, 1]^K$ be any loss vector. For any K-armed adversarial bandit problem, with probability at least $1 - \delta$, the regret of $\mathcal{M}(K)$ satisfies*

$$\sum_{t=1}^{T} c_{t,I_t} - \min_{i\in[K]} \sum_{t=1}^{T} c_{t,i} = \tilde{O}\left(\sqrt{K\mathcal{C}_{T,*}}\right), \quad \mathcal{C}_{T,*} = \min_{i\in[K]} \sum_{t=1}^{T} c_{t,i}.$$

Assumption 1 requires $\mathcal{M}(K)$ achieving a high-probability small-loss bound. There are some superior bandit algorithms satisfying Assumption 1, such as GREEN-IX [15] and the online mirror descent based algorithm proposed in [12].

The following theorem gives the high-probability regret bound induced by the hypothesis sequence $\{f_{t,i}\}_{t=1}^{T} \subseteq \mathbb{H}_i$, $i = 1, \ldots, K$.

Theorem 2. *Let* $\psi_{t,i}(f)=\frac{1}{\lambda_i}\|f\|^2_{\mathcal{H}_i}$ *and* $\delta\in(0,1)$*. For all base kernel* $\kappa_i\in\mathcal{K}$ *and any* $f\in\mathbb{H}_i$*, with probability at least* $1-4\delta$*, the regret of the hypothesis sequence* $\{f_{t,i}\}_{t=1}^T$ *induced by* $\mathrm{B(AO)_2KS}$ *satisfies*

$$\begin{aligned}L_T(f_{1:T,i})-L_T(f)\leq&\frac{\|f\|^2_{\mathcal{H}_i}}{2\lambda_i}+11\lambda_i\sqrt{D_i}Kg_i(T,M)\mathcal{A}(\mathcal{I}_T,\kappa_i)\ln^{\frac{3}{4}}\frac{K}{\delta}\\&+20\lambda_iK^2UD_i\ln\frac{K}{\delta}+13K\sqrt{D_i}U\ln\frac{K}{\delta}+7U\sqrt{Kg_i(T,M)\mathcal{A}(\mathcal{I}_T,\kappa_i)}\ln^{\frac{3}{4}}\frac{K}{\delta},\end{aligned}$$

where $L_T(f_{1:T,i})=\sum_t\ell(f_{t,i}(\mathbf{x}_t),y_t)$*,* $L_T(f)=\sum_t\ell(f(\mathbf{x}_t),y_t)$ *and* $g_i(T,M)=\frac{M+D_i\ln\frac{T}{M}}{M}$*. Let* $\lambda_i=(22\sqrt{D_i}Kg_i(T,M)\mathcal{A}(\mathcal{I}_T,\kappa_i))^{-\frac{1}{2}}$*. The regret is*

$$L_T(f_{1:T,i})-L_T(f)=\tilde{O}\left((\|f\|^2_{\mathcal{H}_i}+U)\sqrt{K\mathcal{A}(\mathcal{I}_T,\kappa_i)}\ln^{\frac{3}{4}}\frac{K}{\delta}\right).$$

Combining Assumption 1 and Theorem 2, we obtain the regret induced by the hypothesis sequence $\{f_t\}_{t=1}^T$ w.r.t. any $f\in\mathbb{H}_i$, $i=1,\ldots,K$.

Theorem 3. *Under the condition of Assumption 1 and Theorem 2, for all* $\kappa_i\in\mathcal{K}$*, with probability at least* $1-5\delta$*, the regret of* $\mathrm{B(AO)_2KS}$ *w.r.t.* $\mathbb{H}_i$ *satisfies*

$$\mathrm{Reg}_T(\mathbb{H}_i)=\tilde{O}\left(\sqrt{L_T(f_i^*)K}+(\|f_i^*\|^2_{\mathcal{H}_i}+U)\sqrt{K\mathcal{A}(\mathcal{I}_T,\kappa_i)}\ln^{\frac{3}{4}}\frac{K}{\delta}\right),$$

where $f_i^*=\arg\min_{f\in\mathbb{H}_i}L_T(f)$*, and* $L_T(f_i^*)\leq\mathcal{A}(\mathcal{I}_T,\kappa_i)$*.*

Compared with Theorem 1, the regret of $\mathrm{B(AO)_2KS}$ only increases by a factor of order $O(\sqrt{K})$, which is nearly optimal. If we just consider online kernel learning using some non-optimal kernel κ_i, then $\mathrm{B(AO)_2KS}$ obtains a regret bound of order $O(\|f_i^*\|^2_{\mathcal{H}_i}\sqrt{\mathcal{A}(\mathcal{I}_T,\kappa_i)})$. Let κ_{i^*} be the optimal kernel. After executing online kernel selection, $\mathrm{B(AO)_2KS}$ achieves a $O(\|f_{i^*}^*\|^2_{\mathcal{H}_{i^*}}\sqrt{K\mathcal{A}(\mathcal{I}_T,\kappa_{i^*})})$ regret. If κ_{i^*} matches well with $\mathcal{I}_T$, i.e., $\mathcal{A}(\mathcal{I}_T,\kappa_{i^*})$ is small, then $\mathrm{B(AO)_2KS}$ improves the learning performance significantly relative to online kernel learning using κ_i. Existing $O(\|f_i^*\|^2_{\mathcal{H}_i}T^\alpha)$ regret bounds may not reveal that kernel selection improves the learning performance, since we can not distinguish $O(\|f_i^*\|^2_{\mathcal{H}_i}T^\alpha)$ from $O(\|f_{i^*}^*\|^2_{\mathcal{H}_{i^*}}T^\alpha)$. Besides, our result also reveals that the information-theoretic cost induced by executing online kernel selection rather than executing online kernel learning using κ_{i^*} is of order $\tilde{O}(\sqrt{L_T(f_i^*)K})$, which could be very small.

3.4 Time Complexity Analysis

The computational cost of $\mathrm{B(AO)_2KS}$ is dominated by computing $\nabla_{t,i}$ and the projection operation. Let S_i be the set of support vectors used to construct $\{f_{t,i}\}_{t=1}^T$. The time complexity of computing $\nabla_{t,i}$ depends on $|S_i|$. The support vectors in S_i comes from (i) reservoir updating, and (ii) the second mirror updating (5). After round $T-1$, the updating times of reservoir is of order $O(M\log T)$, which is proved by Lemma 7 in supplementary material. At any round t, $(\mathbf{x}_t, y_t)$ is added into S_i via the second mirror updating only if $\kappa_i = \kappa_{J_t}$. Let $S_{i,1} = \{(\mathbf{x}_t, y_t) \in S_i : \kappa_i = \kappa_{J_t}\}$. Since $\mathbb{P}[i = J_t] = 1/K$, it is easy to prove that $|S_{i,1}| = O(T/K)$ with a high probability. The projection operation can be executed incrementally, which only induces a $O(M)$ time complexity. The incremental computation procedure is presented in supplementary material. Thus the per-round time complexity of B(AO)$_2$KS is $O(T/K)$.

Remark 1. For online classification with the hinge loss, OKS [22] achieves a $O(\|f^*_i\|^2_{\mathcal{H}_i}\sqrt{KT})$ expected regret bound and suffers a $O(T)$ per-round time complexity. Recently, ISKA [24] provides a $\tilde{O}(\|f^*_i\|^2_{\hat{\mathcal{H}}}T^{\frac{2}{3}} + BT^{\frac{1}{3}})$ expected regret bound, where $\hat{\mathcal{H}} = \cup_{i=1}^K \mathcal{H}_i$, and enjoys a $O(B + KB^2/T)$ per-round time complexity. The online multi-kernel learning algorithm, Raker [19], uses random feature technique to approximate kernel function, which enjoys a $O(\|f^*_i\|^2_{\mathcal{H}_i}\sqrt{T})$ regret bound and suffers a $O(KD)$ per-round time complexity where D is the number of random features. Raker can provide a $O(\sqrt{T})$ regret bound only if $D = \Omega(T)$, which yields a $O(KT)$ per-round time complexity. The same weakness of the above three algorithms is that the $O(\|f^*_i\|^2_{\mathcal{H}_i}T^{\alpha})$, $\frac{1}{2} \leq \alpha < 1$ regret bound is worse than $O(\|f^*_i\|^2_{\mathcal{H}_i}\sqrt{K\mathcal{A}(\mathcal{I}_T, \kappa_i)})$ in the case of $\mathcal{A}(\mathcal{I}_T, \kappa_i) = o(T/K)$.

In the next section, we will propose another algorithm, which relates the time complexity with the alignment and could further reduce the time complexity.

4 Regret-Performance Trade-off

The computational cost of B(AO)$_2$KS comes from the unbounded number of support vectors. Although many effective approaches have been proposed to solve this issue, such as budgeted online kernel leaning [5, 25], Nyström method [2] and random feature technique [14]. However, existing approaches can not provide regret bounds relying on the alignment. To resolve the two issues, we will propose a novel budgeted AO$_2$MD for online kernel learning. The keys include (i) how to select the optimistic estimator, and (ii) how to maintain the budget, especially construct an adaptive example adding strategy.

To solve the first challenge, we still use the reservoir sampling technique to construct the optimistic estimator $\bar{\nabla}_{t,i} := \tilde{\mu}_{t-1,i}$. The key is the second challenge. Let S_i be the budget constructing the hypothesis sequence $\{f_{t,i}\}_{t=1}^T$. We propose an adaptive sampling strategy. At the beginning of round t, we still execute the first mirror updating (4) for obtaining $f_{t,i}$. If $y_t f_{t,i}(\mathbf{x}_t) < 1$, then we observe the gradient $\nabla_{t,i}$ and define a Bernoulli random variable $b_{t,i}$ satisfying

$$\mathbb{P}[b_{t,i} = 1] = \frac{\|\nabla_{t,i} - \bar{\nabla}_{t,i}\|_{\mathcal{H}_i}}{Z_{t,i}},\ Z_{t,i} = \beta_i\left(\|\nabla_{t,i} - \bar{\nabla}_{t,i}\|_{\mathcal{H}_i} + \|\bar{\nabla}_{t,i}\|_{\mathcal{H}_i}\right),$$

where $Z_{t,i}$ is a normalizing constant, and $\beta_i \geq 1$ is a *balancing factor* used to balance the regret and time complexity. If $b_{t,i} = 1$, then we add the current example into the budget, i.e. $S_i = S_i \cup \{(\mathbf{x}_t, y_t)\}$. Otherwise, S_i keeps unchanged.

Different from B(AO)$_2$KS, we reduce online kernel selection to a problem of prediction with expert advice. Let $\mathcal{E}(K)$ be the algorithm for expert advice in Theorem 1. Although evaluating all of the base kernels increases the time complexity by K times, the affection can be counteracted by tuning the balancing factor. At the beginning of round t, we first select κ_{I_t}, $I_t \sim \mathbf{p}_t$ and output $\hat{y}_t = \mathrm{sign}(f_t(\mathbf{x}_t))$, where $f_t = f_{t,I_t}$. Then we explore all of the unselected kernels. Similarly, we update $f_{t,j}$ and compute the gradient $\nabla_{t,j}$. To update the auxiliary hypothesis, we define the variance-reduced gradient estimator $\tilde{\nabla}_{t,i}$ as follows

$$\tilde{\nabla}_{t,i} = \nabla_{t,i}\mathbb{I}_{y_t f_{t,i}(\mathbf{x}_t)\geq 1} + \left[\frac{\nabla_{t,i} - \bar{\nabla}_{t,i}}{\mathbb{P}[b_{t,i}=1]}\mathbb{I}_{b_{t,i}=1} + \bar{\nabla}_{t,i}\right]\mathbb{I}_{y_t f_{t,i}(\mathbf{x}_t)<1}, \forall i = 1, \ldots, K.$$

To update the probability distribution $\mathbf{p}_t$, let $c_{t,i} = \frac{\max\{0, 1 - y_t f_{t,i}(\mathbf{x}_t)\}}{1 + U\sqrt{D}}$. At the end of round t, we send $\mathbf{c}_t = (c_{t,1}, \ldots, c_{t,K})$ to $\mathcal{E}(K)$.

We name this approach EA$_2$OKS (Expert Advice for Adaptive Online Kernel Selection) and present the pseudo-code in Algorithm 2.

Algorithm 2 EA$_2$OKS

```
Input: $\lambda_i, \beta_i, i = 1, \ldots, K$, $D$, $U$, $M$.
Initialization: $\forall \kappa_i \in \mathcal{K}$, $f'_{0,i} = 0, S_i = \emptyset, V = \emptyset$.
for $t = 1, 2, \ldots, T$ do
    Select a kernel $\kappa_{I_t} \sim \mathbf{p}_t$ ($\mathbf{p}_t$ is output by $\mathcal{E}(K)$),
    Compute $\bar{\nabla}_{t,I_t} = \frac{-1}{|V|}\sum_{(\mathbf{x},y)\in V} y\kappa_{I_t}(\mathbf{x}, \cdot)$,
    Update $f_{t,I_t}$ according to (4), and output prediction $\hat{y}_t = \mathrm{sign}(f_{t,I_t}(\mathbf{x}_t))$,
    for $\kappa_i \in \mathcal{K}$ do
        if $\kappa_i \neq \kappa_{I_t}$, then update $f_{t,i}$ according to (4),
        if $y_t f_{t,i}(\mathbf{x}_t) < 1$ then
            Compute $\mathbb{P}[b_{t,i} = 1] = \|\nabla_{t,i} - \bar{\nabla}_{t,i}\|_{\mathcal{H}_i}/Z_{t,i}$,
            Sample $b_{t,i} \sim \mathrm{Ber}(\mathbb{P}[b_{t,i} = 1], 1)$,
            if $b_{t,i} = 1$, then $S_i = S_i \cup (\mathbf{x}_t, y_t)$,
            Compute $\tilde{\nabla}_{t,i} = \frac{\nabla_{t,i}\mathbb{I}_{b_{t,i}=1}}{\mathbb{P}[b_{t,i}=1]} + \left(1 - \frac{\mathbb{I}_{b_{t,i}=1}}{\mathbb{P}[b_{t,i}=1]}\right)\bar{\nabla}_{t,i}$,
            Updating $f'_{t,i}$ according to (5),
            Compute $c_{t,i} = \frac{1}{1+U\sqrt{D}}\max\{0, 1 - y_t f_{t,i}(\mathbf{x}_t)\}$,
        end if
    end for
    Send $\mathbf{c}_t = (c_{t,1}, \ldots, c_{t,K})$ to $\mathcal{E}(K)$,
    Update Reservoir $V$ (line 16-18 in Algorithm 1),
end for
```

4.1 Regret Bound

Theorem 4 gives an upper bound on the number of support vectors in each S_i, which implies the per-round time complexity of EA$_2$OKS.

Theorem 4. *For all $i = 1, \ldots, K$, with probability at least $1 - 2\delta$,* EA$_2$OKS *guarantees that the number of support vectors in S_i satisfies*

$$|S_i| \leq 4M \ln \frac{T}{M}\sqrt{\ln \frac{K}{\delta}} + \frac{4}{3}\ln\frac{K}{\delta} + \frac{10}{\beta_i}\sqrt{\frac{M + D_i \ln \frac{T}{M}}{M} T\mathcal{A}(\mathcal{I}_T, \kappa_i)} \ln^{\frac{3}{4}}\left(\frac{K}{\delta}\right).$$

The time complexity of EA$_2$OKS depends on the alignment $\mathcal{A}(\mathcal{I}_T, \kappa_i)$ implying that selecting different kernel function not only has an impact on the learning performance of online kernel learning algorithms, but also the time complexity. More discussions are shown in Remark 2. Next we present the high-probability regret bound induced by the hypothesis sequence $\{f_{t,i}\}_{t=1}^T$, $i = 1, \ldots, K$.

Theorem 5. *Let $\psi_{t,i}(f) = \frac{1}{\lambda_i}\|f\|^2_{\mathcal{H}_i}$, $\beta_i \geq 1$ and $\delta \in (0, 1)$. For all base kernel $\kappa_i \in \mathcal{K}$ and any $f \in \mathbb{H}_i$, with probability at least $1-4\delta$, the regret of the hypothesis sequence $\{f_{t,i}\}_{t=1}^T$ induced by* EA$_2$OKS *satisfies*

$$L_T(f_{1:T,i}) - L_T(f) \leq \frac{\|f\|^2_{\mathcal{H}_i}}{2\lambda_i} + 18\lambda_i\beta_i\sqrt{D_i g_i(M,T) T\mathcal{A}(\mathcal{I}_T, \kappa_i)\ln\frac{K}{\delta}} +$$
$$(6\lambda_i\beta_i^2 + 7U\beta_i)D_i^{\theta}\ln\frac{K}{\delta} + 9(2\lambda_i\beta_i^{\frac{3}{2}} + U\beta_i^{\frac{1}{2}})D_i^{\frac{\theta}{2}} g_i(M,T)^{\frac{1}{4}} T^{\frac{1}{4}}\mathcal{A}(\mathcal{I}_T, \kappa_i)^{\frac{1}{4}}\ln^{\frac{3}{4}}\frac{K}{\delta},$$

where $g_i(T, M) = (M + D_i \ln \frac{T}{M})/M$ and $\theta \in \{1/2, 2\}$. Let $\beta_i < \sqrt{T\mathcal{A}(\mathcal{I}_T, \kappa_i)}$ and $\lambda_i = (36\beta_i\sqrt{Tg_i(M,T)\mathcal{A}(\mathcal{I}_T, \kappa_i)})^{-\frac{1}{2}}$. The regret is thus of order

$$L_T(f_{1:T,i}) - L_T(f) = \tilde{O}\left((\|f\|^2_{\mathcal{H}_i} + U)\sqrt{\beta_i}T^{\frac{1}{4}}\mathcal{A}(\mathcal{I}_T, \kappa_i)^{\frac{1}{4}}\ln^{\frac{3}{4}}\frac{K}{\delta}\right).$$

Now we can show the final high-probability regret bound.

Theorem 6. *Let $\mathcal{E}(K)$ be the algorithm in Theorem 1. Under the condition of Theorem 5, for all base kernel $\kappa_i \in \mathcal{K}$, with probability at least $1 - 5\delta$, the regret of* EA$_2$OKS *w.r.t. $\mathbb{H}_i$ satisfies*

$$\mathrm{Reg}_T(\mathbb{H}_i) = \tilde{O}\left(\sqrt{L_T(f_i^*)\ln\frac{K}{\delta}} + (\|f_i^*\|^2_{\mathcal{H}_i} + U)\sqrt{\beta_i}T^{\frac{1}{4}}\mathcal{A}(\mathcal{I}_T, \kappa_i)^{\frac{1}{4}}\ln^{\frac{3}{4}}\frac{K}{\delta}\right).$$

The per-round time complexity is of order $\tilde{O}\left(\sum_{i=1}^K \beta_i^{-1}\sqrt{T\mathcal{A}(\mathcal{I}_T, \kappa_i)}\right)$.

Remark 2 (regret-performance trade-off). Theorem 6 reveals that the per-round time complexity depends on $1/\beta_i$, and the regret only depends on $\sqrt{\beta_i}$. Thus β_i balances the regret and time complexity. If $\beta_i = K^{\epsilon}, \epsilon \geq 0$, a universal value, then the time complexity is $\tilde{O}\left(K^{-\epsilon}\sum_{i=1}^K \sqrt{T\mathcal{A}(\mathcal{I}_T, \kappa_i)}\right)$, while the regret only increases by a factor of $K^{\frac{\epsilon}{2}}$. For $\epsilon \geq 2$, EA$_2$OKS is more efficient than B(AO)$_2$KS, but also suffers larger regret. A better approach of setting β_i is to incorporate the information of $\mathcal{A}(\mathcal{I}_T, \kappa_i)$. A more interesting result is shown in Corollary 1.

Corollary 1. *Let the optimal kernel* $\kappa_{i^*} = \mathrm{argmin}_{i\in[K]}\mathcal{A}(\mathcal{I}_T, \kappa_i)$*, and the balance factor* β_i *satisfy the following condition*

$$\beta_i\sqrt{\mathcal{A}(\mathcal{I}_T, \kappa_{i^*})} = K\beta\sqrt{\mathcal{A}(\mathcal{I}_T, \kappa_i)}, \quad \beta \geq 1, \quad i = 1, \ldots, K.$$

The regret of EA$_2$OKS *satisfies, with probability at least* $1 - 5\delta$*,*

$$\mathrm{Reg}_T(\mathbb{H}_i) = \begin{cases} \tilde{O}\left(\sqrt{L_T(f^*_{i^*})K\ln\frac{K}{\delta}} + (\|f^*_{i^*}\|^2_{\mathcal{H}_{i^*}} + U)\beta_K^{\frac{1}{2}}T^{\frac{1}{4}}\mathcal{A}(\mathcal{I}_T, \kappa_{i^*})^{\frac{1}{4}}\right) & i = i^* \\ \tilde{O}\left(\sqrt{L_T(f^*_i)K\ln\frac{K}{\delta}} + (\|f^*_i\|^2_{\mathcal{H}_i} + U)\beta_K^{\frac{1}{2}}T^{\frac{1}{4}}\frac{\mathcal{A}(\mathcal{I}_T,\kappa_i)^{\frac{1}{2}}}{\mathcal{A}(\mathcal{I}_T,\kappa_{i^*})^{\frac{1}{4}}}\right) & i \neq i^* \end{cases}$$

where $\beta_K = K\beta$*. The per-round time complexity is of order* $O(\frac{1}{\beta}\sqrt{T\mathcal{A}(\mathcal{I}_T, \kappa_{i^*})})$*.*

For kernel selection, it is unnecessary to compare with all of the base kernels. Any algorithm just needs to be competitive with the case in which we know the optimal kernel κ_{i^*} in advance. Thus, we allow the algorithm to achieve a worse regret bound w.r.t. the non-optimal RKHS $\mathbb{H}_i, i \neq i^*$. The significance of Corollary 1 is that EA$_2$OKS can keep the same regret bound w.r.t. $\mathbb{H}_{i^*}$, and reduce the per-round time complexity to $O(\beta^{-1}\sqrt{T\mathcal{A}(\mathcal{I}_T, \kappa_{i^*})})$.

We analyze the time complexity. According to Theorem 4, the exact time complexity of EA$_2$OKS is of order $O\left(\sum_{i=1}^K |S_i|\right)$. Let $M = O(\ln T)$. Then the time complexity of EA$_2$OKS is the one claimed In Remark 2 or the comments after Corollary 1. We omit the time complexity of projection operation, since it can be executed incrementally in $O(M)$ time.

4.2 Budgeted EA$_2$OKS

Inspired by Corollary 1, we can set a same threshold for all S_i, i.e., $|S_i| \leq B$. At the end of round $t - 1$, if $|S_i| = B$, then we reset $S_i = \emptyset$ and $f'_{t-1,i} = 0$. We name the algorithm BEA$_2$OKS (Budgeted EA$_2$OKS). Due to BEA$_2$OKS is much similar with EA$_2$OKS and the space limit, the algorithm description is presented in supplementary material. Combining Theorem 4 and Corollary 1, we further obtain Corollary 2.

Corollary 2. *Let* $\kappa_{i^*} = \mathrm{argmin}_{i\in[K]}\mathcal{A}(\mathcal{I}_T, \kappa_i)$*. If* B *satisfies the condition*

$$B = 4M\ln\frac{T}{M}\sqrt{\ln\frac{K}{\delta} + \frac{4}{3}\ln\frac{K}{\delta} + \frac{10}{\beta_{i^*}}\sqrt{g_{i^*}(M,T)T\mathcal{A}(\mathcal{I}_T, \kappa_{i^*})}\ln^{\frac{3}{4}}\left(\frac{K}{\delta}\right)},$$

and $\beta_{i^*} = K$*, then with probability at least* $1 - 5\delta$*,* S_{i^*} *will not restart, and*

$$\mathrm{Reg}_T(\mathbb{H}_{i^*}) = \tilde{O}\left(\sqrt{L_T(f^*_{i^*})K\ln\frac{K}{\delta}} + (\|f^*_{i^*}\|^2_{\mathcal{H}_{i^*}} + U)K^{\frac{1}{2}}T^{\frac{1}{4}}\mathcal{A}(\mathcal{I}_T, \kappa_{i^*})^{\frac{1}{4}}\right).$$

The per-round time complexity of BEA$_2$OKS *is of order* $O(\sqrt{T\mathcal{A}(\mathcal{I}_T, \kappa_{i^*})})$*.*

According to Corollary 2 and 1, we claim that BEA$_2$OKS and EA$_2$OKS (let $\beta = 1$) are equivalent in the sense that they enjoy the same regret bound w.r.t. $\mathbb{H}_{i^*}$ and the same per-round time complexity. The superiority of BEA$_2$OKS is that it only needs to tune B, while EA$_2$OKS needs to tune $\beta_i, i = 1, \ldots, K$.

5 Experiments

In this section, we conduct experiments to verify our theoretical results.

5.1 Experimental Setting

We only verify B(AO)$_2$KS, EA$_2$OKS and BEA$_2$OKS, but do not run the algorithm in Theorem 1, since the $O(KT)$ time complexity is prohibitive. For B(AO)$_2$KS, let $\mathcal{M}(K)$ be GREEN-IX [15]. For EA$_2$OKS and BEA$_2$OKS, let $\mathcal{E}(K)$ be the exponentially weighted average algorithm (chapter 4.2 in [3]) whose learning rate (Corollary 2.4 in [3]) is tuned by doubling trick. We use four binary classification datasets downloaded from LIBSVM website [1], including *w7a* (Num: 24692, Fea: 300), *w8a* (Num: 49749, Fea: 300), *a9a* (Num: 48842, Fea: 123) and *ijcnn1* (Num: 141691, Fea: 22). Let $\mathcal{K} = \{\sigma_i\}_{i=1}^K$ contain K Gaussian kernels, where $\kappa_i(\mathbf{u}, \mathbf{v}) = \exp(-\|\mathbf{u} - \mathbf{v}\|^2/(2\sigma_i^2))$. We implement all algorithms with R on a Windows machine with 2.5 GHz Core i7 CPU, execute each experiment 10 times with random permutation of all datasets and average all of the results [2].

We will execute three experiments. For all experiments, let $M = \lceil \ln T \rceil$ and $U = 20$. $D = 1$ for Gaussian kernel. The first experiment aims at verifying the influence of K on B(AO)$_2$KS. We choose 3 groups of $\mathcal{K}$, denoted by $\mathcal{K}_{12} = \{2^{-2:0.5:3.5}\}$, $\mathcal{K}_8 = \{2^{-1:0.5:2.5}\}$ and $\mathcal{K}_4 = \{2^{-1:1:2}\}$, and name the corresponding algorithm B(AO)$_2$KS-12, B(AO)$_2$KS-8 and B(AO)$_2$KS-4. According to Theorem 2, the optimal learning rate λ_i should be $\tilde{O}(1/\sqrt{K\mathcal{A}(\mathcal{I}_T, \kappa_i)})$. However, B(AO)$_2$KS is not parameter-free. In this paper, we set $\lambda_i = 1/\sqrt{K\mathcal{A}(\mathcal{I}_T, \kappa_{i^*})}$ for all $i \in [K]$, and set $\mathcal{A}(\mathcal{I}_T, \kappa_{i^*}) = \sqrt{T}$ which is an optimistic estimator.

The second experiment aims at proving the advantage of the data-dependent regret bounds and time complexity. The baseline algorithms include two online kernel selection algorithms: OKS [22] and ISKA [24], and two online kernel learning algorithms: Forgetron-σ [5] and BOGD-σ [25]. For Forgetron-σ and BOGD-σ, we set σ to the best value in hindsight. The other hyper-parameters are set to the recommended value in original papers. For EA$_2$OKS and BEA$_2$OKS, we set $\beta_i = K^{\frac{3}{2}}$. Although the optimal B is unknown for BEA$_2$OKS, a feasible approach is to set a slightly large value, which ensures S_{i^*} will not restart and Corollary 2 holds on. We select $\mathcal{K} = \{2^{-2:1:3}\}$. For fair comparison, we set the stepsize of gradient descent (or λ_i in this paper) to $5/\sqrt{T}$ for all algorithms.

The third experiment shows the influence of the balancing factor β_i and budget B on EA$_2$OKS and BEA$_2$OKS, and compares the two algorithms further. We adopt the same $\mathcal{K}$ and λ_i used in the second experiment.

5.2 Experimental Results

Table 1 shows the results of the first experiment. Overall, the experimental results coincide with our theoretical analyses (Theorem 3). If $\mathcal{K}_{12}$, $\mathcal{K}_8$ and $\mathcal{K}_4$

[1] https://www.csie.ntu.edu.tw/\%7Ecjlin/libsvmtools/datasets/

[2] The codes are available at https://github.com/JunfLi-TJU/KARegret-OKS

Table 1. The influence of K on B(AO)$_2$KS

Algorithm	w8a		a9a	
	Mistake (%)	Time (s)	Mistake (%)	Time (s)
B(AO)$_2$KS-4	$\mathbf{2.15} \pm 0.09$	444.88 ± 23.81	$\mathbf{12.72} \pm 0.08$	320.55 ± 5.63
B(AO)$_2$KS-8	2.53 ± 0.04	271.27 ± 13.57	15.39 ± 0.12	216.96 ± 4.53
B(AO)$_2$KS-12	2.79 ± 0.03	245.85 ± 8.49	17.47 ± 0.18	226.98 ± 6.05

Table 2. Performance comparison among different online kernel selection algorithms

Algorithm	B-$\|S_{i^*}\|$	w7a		B-$\|S_{i^*}\|$	w8a	
		Mistake (%)	Time (s)		Mistake (%)	Time (s)
BOGD-σ	500	2.75 ± 0.03	44.66	800	3.26 ± 0.34	143.31
Forgetron-σ	500	5.29 ± 0.07	34.90	800	4.74 ± 0.06	119.92
OKS	-	2.55 ± 0.16	201.26	-	$\mathbf{2.38} \pm 0.10$	753.08
ISKA	250	4.72 ± 2.20	53.73	400	3.16 ± 0.24	181.27
B(AO)$_2$KS	-	$\mathbf{2.56} \pm 0.06$	100.57	-	2.48 ± 0.07	364.27
EA$_2$OKS	127	3.12 ± 0.04	112.21	201	3.03 ± 0.02	394.78
BEA$_2$OKS	200	3.17 ± 0.08	40.74	400	3.02 ± 0.02	116.56

Algorithm	B-$\|S_{i^*}\|$	a9a		B-$\|S_{i^*}\|$	jicnn1	
		Mistake (%)	Time (s)		Mistake (%)	Time (s)
BOGD-σ	1500	17.85 ± 0.06	114.31	3500	9.57 ± 0.00	149.93
Forgetron-σ	1700	24.38 ± 0.16	170.34	3500	13.04 ± 0.08	164.99
OKS	-	18.71 ± 0.22	715.39	-	8.90 ± 0.12	477.14
ISKA	1400	16.95 ± 0.05	296.29	1500	8.48 ± 0.05	135.79
B(AO)$_2$KS	-	$\mathbf{15.16} \pm 0.11$	308.75	-	$\mathbf{7.58} \pm 0.17$	391.48
EA$_2$OKS	1062	17.88 ± 0.23	267.68	1025	8.75 ± 0.11	227.43
BEA$_2$OKS	1200	17.83 ± 0.16	167.12	1500	8.79 ± 0.16	188.20

contain the same optimal kernel, then the smaller K is, the better learning performance and the longer running time is. The result of B(AO)$_2$KS-12 on *a9a* does not satisfy the rule. The reason is that $\mathcal{K}_{12}$ contains many kernels performing badly on *a9a* which leads to a large number of support vectors.

Table 2 reports the results of the second experiment. In the second column, B is the budget size and $|S_{i^*}| = \min_i |S_i|$ in EA$_2$OKS. Note that we use the kernel with minimal $|S_i|$ as a proxy for the optimal kernel κ_{i^*}. B(AO)$_2$KS enjoys the best learning performance, but also suffers a slightly larger time complexity. OKS has the longest running time on all datasets. For BOGD-σ and Forgetron-σ, we select the best σ in hindsight for constructing strong baseline algorithms, which is unprocurable in practice. Note that BEA$_2$OKS enjoys the same prediction performance with EA$_2$OKS, and has lower running time, since we set $B \approx |S_{i^*}|$ in EA$_2$OKS. Overall, BEA$_2$OKS provides the best regret-time complexity trade-off except for the *ijcnn1* dataset on which ISKA performs better.

Table 3 shows the results of the third experiment. #rs is the average restart times of S_{i^*} in BEA$_2$OKS, or the average $|S_{i^*}|$ in EA$_2$OKS. For a same β, BEA$_2$OKS keeps the same prediction accuracy with EA$_2$OKS, and improves the efficiency significantly. The reason is that #rs is 0. Thus BEA$_2$OKS just needs

Table 3. Parameter influence on EA$_2$OKS and BEA$_2$OKS on *w8a* dataset

Algorithm	Mistake (%)	Time (s)	#rs	Mistake (%)	Time (s)	#rs
EA$_2$OKS	$\beta = K$			$\beta = K^2$		
	2.99 ± 0.02	1060.17 ± 26.0	343	3.35 ± 0.15	215.28 ± 6.15	140
BEA$_2$OKS	$\beta = K$, $B = 400$			$\beta = K^2$, $B = 400$		
	2.99 ± 0.02	132.33 ± 2.28	0	3.43 ± 0.19	123.92 ± 5.96	0
BEA$_2$OKS	$\beta = K$, $B = 600$			$\beta = K^2$, $B = 600$		
	3.00 ± 0.03	174.77 ± 2.20	0	3.43 ± 0.22	144.39 ± 7.68	0

a small budget to keep the regret w.r.t. $\mathbb{H}_{i^*}$ achieved by EA$_2$OKS. In this case, there is no sense to increase B. For a same B, the smaller β is, the better learning performance and the longer running time is. The experimental results coincide with Corollary 2.

6 Conclusion

In this paper, we develop several computationally efficient online kernel selection algorithms, which achieve the first kernel alignment regret bound improving previous worst-case regret bounds. Theoretical analyses reveal that if there is a good kernel in the candidate set, then our algorithms can not only improve the learning performance relative to single kernel learning, but also suffer a low time complexity. Experimental results verify the effectiveness and efficiency of our algorithms. An important question is whether it is possible to achieve the $O(\|f\|^2_{\mathcal{H}_{i^*}}\sqrt{\mathcal{A}(\mathcal{I}_T, \kappa_{i^*})})$ regret bound with a $O(\mathcal{A}(\mathcal{I}_T, \kappa_{i^*}))$ time complexity.

References

1. Baram, Y.: Learning by kernel polarization. Neural Computation **17**(6), 1264–1275 (2005)
2. Calandriello, D., Lazaric, A., Valko, M.: Efficient second-order online kernel learning with adaptive embedding. Advances in Neural Information Processing Systems **30**, 6140–6150 (2017)
3. Cesa-Bianchi, N., Lugosi, G.: Prediction, Learning, and Games. Cambridge University Press (2006)
4. Chiang, C., Yang, T., Lee, C., Mahdavi, M., Lu, C., Jin, R., Zhu, S.: Online optimization with gradual variations. In: Proceedings of the 25th Annual Conference on Learning Theory. pp. 6.1–6.20 (2012)
5. Dekel, O., Shalev-Shwartz, S., Singer, Y.: The forgetron: A kernel-based perceptron on a budget. SIAM Journal on Computing **37**(5), 1342–1372 (2008)
6. Foster, D.J., Kale, S., Mohri, M., Sridharan, K.: Parameter-free online learning via model selection. Advances in Neural Information Processing Systems **30**, 6022–6032 (2017)
7. Foster, D.J., Rakhlin, A., Sridharan, K.: Adaptive online learning. Advances in Neural Information Processing Systems **28**, 3375–3383 (2015)

8. Hazan, E., Kale, S.: Better algorithms for benign bandits. In: Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 38–47 (2009)
9. Hazan, E., Kale, S.: Extracting certainty from uncertainty: regret bounded by variation in costs. Machine Learning **80**(2-3), 165–188 (2010)
10. Jézéquel, R., Gaillard, P., Rudi, A.: Efficient online learning with kernels for adversarial large scale problems. Advances in Neural Information Processing Systems **32**, 9427–9436 (2019)
11. Jin, R., Hoi, S.C.H., Yang, T.: Online multiple kernel learning: Algorithms and mistake bounds. In: Proceedings of the 21st International Conference on Algorithmic Learning Theory. pp. 390–404 (2010)
12. Lee, C., Luo, H., Wei, C., Zhang, M.: Bias no more: high-probability data-dependent regret bounds for adversarial bandits and mdps. Advances in Neural Information Processing Systems **33**, 15522–15533 (2020)
13. Li, J., Liao, S.: Online kernel selection with multiple bandit feedbacks in random feature space. In: Proceedings of the 11th International Conference on Knowledge Science, Engineering and Management. pp. 301–312 (2018)
14. Lu, J., Hoi, S.C.H., Wang, J., Zhao, P., Liu, Z.: Large scale online kernel learning. Journal of Machine Learning Research **17**(47), 1–43 (2016)
15. Lykouris, T., Sridharan, K., Tardos, É.: Small-loss bounds for online learning with partial information. In: Proceedings of the 31st Conference on Learning Theory. pp. 979–986 (2018)
16. Muthukumar, V., Ray, M., Sahai, A., Bartlett, P.: Best of many worlds: Robust model selection for online supervised learning. In: Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics. pp. 3177–3186 (2019)
17. Nguyen, T.D., Le, T., Bui, H., Phung, D.: Large-scale online kernel learning with random feature reparameterization. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence. pp. 2543–2549 (2017)
18. Rakhlin, A., Sridharan, K.: Online learning with predictable sequences. In: Proceedings of the 26th Annual Conference on Learning Theory. pp. 993–1019 (2013)
19. Shen, Y., Chen, T., Giannakis, G.B.: Random feature-based online multi-kernel learning in environments with unknown dynamics. Journal of Machine Learning Research **20**(22), 1–36 (2019)
20. Vitter, J.S.: Random sampling with a reservoir. ACM Transactions on Mathematical Software **11**(1), 37–57 (1985)
21. Wei, C., Luo, H.: More adaptive algorithms for adversarial bandits. In: Proceedings of the 31st Annual Conference on Learning Theory. pp. 1263–1291 (2018)
22. Yang, T., Mahdavi, M., Jin, R., Yi, J., Hoi, S.C.H.: Online kernel selection: Algorithms and evaluations. In: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence. pp. 1197–1202 (2012)
23. Zhang, L., Yi, J., Jin, R., Lin, M., He, X.: Online kernel learning with a near optimal sparsity bound. In: Proceedings of the 30th International Conference on Machine Learning. pp. 621–629 (2013)
24. Zhang, X., Liao, S.: Online kernel selection via incremental sketched kernel alignment. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence. pp. 3118–3124 (2018)
25. Zhao, P., Wang, J., Wu, P., Jin, R., Hoi, S.C.H.: Fast bounded online gradient descent algorithms for scalable kernel-based online learning. In: Proceedings of the 29th International Conference on Machine Learning. pp. 1075–1082 (2012)