

# Improved Regret Bounds for Online Kernel Selection under Bandit Feedback <sup>★</sup>

Junfan Li<sup>[0000–0003–1027–4251]</sup> and Shizhong Liao<sup>[0000–0003–0594–7116]</sup> (✉)

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

**Abstract.** In this paper, we improve the regret bound for online kernel selection under bandit feedback. Previous algorithm enjoys a  $O((\|f\|_{\mathcal{H}_i}^2 + 1)K^{\frac{1}{3}}T^{\frac{2}{3}})$  expected bound for Lipschitz loss functions. We prove two types of regret bounds improving the previous bound. For smooth loss functions, we propose an algorithm with a  $O(U^{\frac{2}{3}}K^{-\frac{1}{3}}(\sum_{i=1}^K L_T(f_i^*))^{\frac{2}{3}})$  expected bound where  $L_T(f_i^*)$  is the cumulative losses of optimal hypothesis in  $\mathbb{H}_i = \{f \in \mathcal{H}_i : \|f\|_{\mathcal{H}_i} \leq U\}$ . The data-dependent bound keeps the previous worst-case bound and is smaller if most of candidate kernels match well with the data. For Lipschitz loss functions, we propose an algorithm with a  $O(U\sqrt{KT} \ln^{\frac{2}{3}} T)$  expected bound asymptotically improving the previous bound. We apply the two algorithms to online kernel selection with time constraint and prove new regret bounds matching or improving the previous  $O(\sqrt{T \ln K} + \|f\|_{\mathcal{H}_i}^2 \max\{\sqrt{T}, \frac{T}{\sqrt{\mathcal{R}}}\})$  expected bound where  $\mathcal{R}$  is the time budget. Finally, we empirically verify our algorithms on online regression and classification tasks.

**Keywords:** Model selection · Online learning · Bandit · Kernel method.

## 1 Introduction

Selecting a suitable kernel function is critical for online kernel learning algorithms, and is more challenge than offline kernel selection since the data are provided sequentially and may not be i.i.d.. Such kernel selection problems are named online kernel selection [22]. To address those challenges, many online kernel selection algorithms reduce it to a sequential decision problem, and then randomly select a kernel function or use a convex combination of multiple kernel functions on the fly [19, 7, 22, 16, 11]. Let  $\mathcal{K} = \{\kappa_i\}_{i=1}^K$  be predefined base kernels. An adversary sequentially sends the learner instances  $\{\mathbf{x}_t\}_{t=1}^T$ . The learner will choose a sequence of hypotheses  $\{f_t\}_{t=1}^T$  from the  $K$  reproducing kernel Hilbert spaces (RKHSs) induced by kernels in  $\mathcal{K}$ . At each round  $t$ , the learner suffers a prediction loss  $\ell(f_t(\mathbf{x}_t), y_t)$ . The goal is to minimize the regret defined as follows,

$$\forall \kappa_i \in \mathcal{K}, \forall f \in \mathcal{H}_i, \quad \text{Reg}_T(f) = \sum_{t=1}^T \ell(f_t(\mathbf{x}_t), y_t) - \sum_{t=1}^T \ell(f(\mathbf{x}_t), y_t). \quad (1)$$

---

<sup>★</sup> This work was supported in part by the National Natural Science Foundation of China under grants No. 62076181.

Effective online kernel selection algorithms must keep sublinear regret bounds w.r.t. the unknown optimal RKHS  $\mathcal{H}_{i^*}$  induced by  $\kappa_{i^*} \in \mathcal{K}$ .

Previous work reduces online kernel selection to a sequential decision problem, including (i) prediction with expert advice [4], (ii)  $K$ -armed bandit problem [3], (iii) prediction with limited advice [15]. The online multi-kernel learning algorithms [14, 5] which reduce the problem to prediction with expert advice, use a convex combination of  $K$  hypotheses and enjoy a  $O(\text{poly}(\|f\|_{\mathcal{H}_i})\sqrt{T \ln K})$  regret bound. Combining  $K$  hypotheses induces a  $O(Kt)$  per-round time complexity which is linear with  $K$ . To reduce the time complexity, the OKS algorithm (Online Kernel Selection) [19] reduces the problem to an adversarial  $K$ -armed bandit problem. OKS randomly selects a hypothesis per-round and only provides a  $O(\text{poly}(\|f\|_{\mathcal{H}_i})K^{\frac{1}{3}}T^{\frac{2}{3}})$ <sup>1</sup> expected bound. The per-round time complexity of OKS is  $O(t)$ . The B(AO)<sub>2</sub>KS algorithm [11] reduces the problem to predict with limited advice and randomly selects two hypotheses per-round. B(AO)<sub>2</sub>KS can provide a  $\tilde{O}(\text{poly}(\|f\|_{\mathcal{H}_i})\sqrt{KT})$  high-probability bound and suffers a  $O(t/K)$  per-round time complexity. From the perspective of algorithm design, an important question arises: does there exist some algorithm only selecting a hypothesis (or under bandit feedback) improving the  $O(\text{poly}(\|f\|_{\mathcal{H}_i})K^{\frac{1}{3}}T^{\frac{2}{3}})$  expected bound? The significances of answering the question include (i) explaining the information-theoretic cost induced by only selecting a hypothesis (or observing a loss); (ii) designing better algorithms for online kernel selection with time constraint. In this paper, we will answer the question affirmatively.

We consider Lipschitz loss functions and smooth loss functions (Assumption 1). For Lipschitz loss functions, we propose an algorithm whose expected regret bound is  $O(U\sqrt{KT} \ln^{\frac{2}{3}} T)$  asymptotically improving the  $O(\text{poly}(\|f\|_{\mathcal{H}_i})K^{\frac{1}{3}}T^{\frac{2}{3}})$  expected bound. Our regret bound proves that selecting a or multiple hypotheses will not induce significant variation on the worst-case regret bound. For smooth loss functions, we propose an adaptive parameter tuning scheme for OKS and prove a  $O(U^{\frac{2}{3}}K^{-\frac{1}{3}}(\sum_{j=1}^K L_T(f_j^*))^{\frac{2}{3}})$  expected bound where  $L_T(f_j^*) = \min_{f \in \mathbb{H}_j} \sum_{t \in [T]} \ell(f(\mathbf{x}_t), y_t)$ . If most of base kernels in  $\mathcal{K}$  match well with the data, i.e.,  $L_T(f_j^*) \ll T$ , then the data-dependent regret bound significantly improves the previous worst-case bound. In the worst case, i.e.,  $L_T(f_j^*) = O(T)$ , the data-dependent bound is still same with the previous bound. Our new regret bounds answer the above question. We summary the results in Table 1.

We apply the two algorithms to online kernel selection with time constraint where the time of kernel selection and online prediction is limited to  $\mathcal{R}$  quanta [9]. It was proved that any budgeted algorithm must suffer an expected regret of order  $\Omega(\|f_i^*\|_{\mathcal{H}_i} \max\{\sqrt{T}, \frac{T}{\sqrt{\mathcal{R}}}\})$  and the LKMBooks algorithm enjoys a  $O(\sqrt{T \ln K} + \|f\|_{\mathcal{H}_i}^2 \max\{\sqrt{T}, \frac{T}{\sqrt{\mathcal{R}}}\})$  expected bound [9]. LKMBooks uses convex combination to aggregate  $K$  hypotheses. Raker uses random features to approximate kernel functions and also aggregates  $K$  hypotheses [16]. Raker enjoys a  $\tilde{O}((\sqrt{\ln K} + \|f\|_1^2)\sqrt{T} + \|f\|_1 \frac{T}{\sqrt{\mathcal{R}}})$  bound where  $f = \sum_{t=1}^T \alpha_t \kappa_i(\mathbf{x}_t, \cdot)$  and

<sup>1</sup>  $\text{poly}(\|f\|_{\mathcal{H}_i}) = \|f\|_{\mathcal{H}_i}^2 + 1$ . The original paper shows a  $O((\|f\|_{\mathcal{H}_i}^2 + 1)\sqrt{KT})$  expected regret bound. We will clarify the difference in Section 2.

**Table 1.** Expected regret bounds for online kernel selection under bandit feedback.  $\mathcal{R}$  is the time budget.  $\bar{L}_T = \sum_{j=1}^K L_T(f_j^*)$ .  $\nu$  is a parameter in the definition of smooth loss (see Assumption 1). There is no algorithm under bandit feedback in the case of a time budget. Thus we report the result produced under the expert advice model [9].

$\mathcal{R}$	Loss function	Previous results	Our results
No	Lipschitz loss	$O\left(\text{poly}(\ f\ _{\mathcal{H}_i})K^{\frac{1}{3}}T^{\frac{2}{3}}\right)$ [19]	$O(U\sqrt{KT}\ln^{\frac{2}{3}}T)$
	Smooth loss $\nu = 1$		$O(U^{\frac{2}{3}}K^{-\frac{1}{3}}\bar{L}_T^{\frac{2}{3}})$
	Smooth loss $\nu = 2$		$O(U^{\frac{2}{3}}K^{-\frac{1}{3}}\bar{L}_T^{\frac{2}{3}})$
Yes	Lipschitz loss	$O\left(\ f\ _{\mathcal{H}_i}^2 \max\{\sqrt{T}, \frac{T}{\sqrt{\mathcal{R}}}\}\right)$ [9]	$O(U\sqrt{KT}\ln^{\frac{2}{3}}T + \frac{UT\sqrt{\ln T}}{\sqrt{\mathcal{R}}})$
	Smooth loss $\nu = 1$		$\tilde{O}(U^{\frac{2}{3}}K^{-\frac{1}{3}}\bar{L}_T^{\frac{2}{3}} + \frac{UL_T(f_i^*)}{\sqrt{\mathcal{R}}})$
	Smooth loss $\nu = 2$		$\tilde{O}(U^{\frac{2}{3}}K^{-\frac{1}{3}}\bar{L}_T^{\frac{2}{3}} + \frac{U\sqrt{TL_T(f_i^*)}}{\sqrt{\mathcal{R}}})$

$\|f\|_1 = \|\alpha\|_1$  [16]. The two algorithms reduce the problem to prediction with expert advice, while our algorithms just use bandit feedback.

We also use random features and make a mild assumption that reduces the time budget  $\mathcal{R}$  to the number of features. For smooth loss functions, we prove two data-dependent regret bounds which can improve the previous worst-case bounds [16, 9] if there is a good kernel in  $\mathcal{K}$  that matches well with the data. For Lipschitz loss functions, our algorithm enjoys a similar upper bound with LKMBounds. We also summary the results in Table 1.

## 2 Problem Setting

Denote by  $\{(\mathbf{x}_t, y_t)\}_{t \in [T]}$  a sequence of examples, where  $\mathbf{x}_t \in \mathcal{X} \subseteq \mathbb{R}^d$ ,  $y \in [-1, 1]$  and  $[T] = \{1, 2, \dots, T\}$ . Let  $\kappa(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  be a positive definite kernel and  $\mathcal{K} = \{\kappa_1, \dots, \kappa_K\}$ . For each  $\kappa_i \in \mathcal{K}$ , let  $\mathcal{H}_i = \{f | f : \mathcal{X} \rightarrow \mathbb{R}\}$  be the associated RKHS satisfying  $\langle f, \kappa_i(\mathbf{x}, \cdot) \rangle_{\mathcal{H}_i} = f(\mathbf{x})$ ,  $\forall f \in \mathcal{H}_i$ . Let  $\|f\|_{\mathcal{H}_i}^2 = \langle f, f \rangle_{\mathcal{H}_i}$ . We assume that  $\kappa_i(\mathbf{x}, \mathbf{x}) \leq 1$ ,  $\forall \kappa_i \in \mathcal{K}$ . Let  $\ell(\cdot, \cdot) : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$  be the loss function.

### 2.1 Online Kernel Selection under Bandit Feedback

We formulate online kernel selection as a sequential decision problem. At any round  $t \in [T]$ , an adversary gives an instance  $\mathbf{x}_t$ . The learner maintains  $K$  hypotheses  $\{f_{t,i} \in \mathcal{H}_i\}_{i=1}^K$  and selects  $f_t \in \text{span}(f_{t,i} : i \in [K])$ , and outputs  $f_t(\mathbf{x}_t)$ . Then the adversary gives  $y_t$ . The learner suffers a prediction loss  $\ell(f_t(\mathbf{x}_t), y_t)$ . The learner aims to minimize the regret w.r.t. any  $f \in \cup_{i=1}^K \mathcal{H}_i$  which is defined in (1). If the learner only computes a loss  $\ell(f_{t,I_t}(\mathbf{x}_t), y_t)$ ,  $I_t \in [K]$ , then we call it *bandit feedback setting*. The learner can also compute  $N \in \{2, \dots, K\}$  losses, i.e.,  $\{\ell(f_{t,i_j}(\mathbf{x}_t), y_t)\}_{j=1}^N$ ,  $i_j \in [K]$ . The OKS algorithm [19] follows the bandit feedback setting. The online multi-kernel learning algorithms [14, 5, 16] correspond to  $N = K$ . The B(AO)<sub>2</sub>KS algorithm [11] corresponds to  $N = 2$ . From the perspective of computation, the per-round time complexity of computing  $N$  losses is

**Algorithm 1** OKS

---

**Input:**  $\mathcal{K} = \{\kappa_1, \dots, \kappa_K\}$ ,  $\delta \in (0, 1)$ ,  $\eta$ ,  $\lambda$   
**Initialization:**  $\{f_{1,i} = 0, w_{1,i} = 1\}_{i=1}^K$ ,  $\mathbf{p}_1 = \frac{1}{K} \mathbf{1}_K$   
1: **for**  $t=1, \dots, T$  **do**  
2:   Receive  $\mathbf{x}_t$ ;  
3:   Sample a kernel  $\kappa_{I_t}$  where  $I_t \sim \mathbf{p}_t$ ;  
4:   Update  $w_{t+1, I_t} = w_{t, I_t} \exp(-\eta \frac{\ell(f_{t, I_t}(\mathbf{x}_t), y_t)}{p_{t, I_t}})$ ;  
5:   Update  $f_{t+1, I_t} = f_{t, I_t} - \lambda \frac{\nabla_{f_{t, I_t}} \ell(f_{t, I_t}(\mathbf{x}_t), y_t)}{p_{t, I_t}}$ ;  
6:   Update  $\mathbf{q}_{t+1} = \frac{\mathbf{w}_{t+1}}{\sum_{j=1}^K w_{t+1, j}}$  and set  $\mathbf{p}_{t+1} = (1 - \delta)\mathbf{q}_{t+1} + \frac{\delta}{K} \mathbf{1}_K$ ;  
7: **end for**  
8: **Output:**  $\mathbf{q}_T$ .

---

$N$  times larger than the bandit feedback setting. From the perspective of regret bound, we aim to reveal the information-theoretic cost induced by observing a loss (or bandit feedback) not multiple losses (or  $N \geq 2$ ).

**2.2 Regret bound of OKS**

We first prove that the regret bound of OKS [19] is  $O((\|f\|_{\mathcal{H}_i}^2 + 1)K^{\frac{1}{3}}T^{\frac{2}{3}})$ , and then explain the technical weakness of OKS.

The pseudo-code of OKS is shown Algorithm 1. Let  $\Delta_K$  be the  $(K - 1)$ -dimensional simplex. At any round  $t$ , OKS maintains  $\mathbf{p}_t, \mathbf{q}_t \in \Delta_K$ . OKS samples  $f_{t, I_t}$  where  $I_t \sim \mathbf{p}_t$ , and outputs  $f_{t, I_t}(\mathbf{x}_t)$ . For simplicity, we define two notations,

$$L_T(f) := \sum_{t=1}^T \ell(f(\mathbf{x}_t), y_t), \quad \bar{L}_{\mathbf{q}_{1:T}} := \sum_{t=1}^T \sum_{i=1}^K q_{t,i} \ell(f_{t,i}(\mathbf{x}_t), y_t).$$

**Theorem 1 ([19]).** *Assuming that  $\ell(f_{t,i}(\mathbf{x}), y) \in [0, \ell_{\max}]$ ,  $\forall i \in [K], t \in [T]$ , and  $\|\nabla_f \ell(f(\mathbf{x}), y)\|_{\mathcal{H}_i} \leq G$ ,  $\forall f \in \mathcal{H}_i$ . The expected regret of OKS satisfies*

$$\forall i \in [K], f \in \mathcal{H}_i, \quad \mathbb{E}[\bar{L}_{\mathbf{q}_{1:T}}] \leq L_T(f) + \frac{\|f\|_{\mathcal{H}_i}^2}{2\lambda} + \frac{\lambda K T G^2}{2\delta} + \frac{\eta K T \ell_{\max}^2}{2(1-\delta)} + \frac{\ln K}{\eta}.$$

*In particular, let  $\delta \in (0, 1)$  be a constant and  $\eta, \lambda = \Theta((KT)^{-\frac{1}{2}})$ , then the expected regret bound is  $O((\|f\|_{\mathcal{H}_i}^2 + 1)\sqrt{KT})$ .*

*Remark 1.* Since  $I_t \sim \mathbf{p}_t$ , the expected cumulative losses of OKS should be  $\mathbb{E}[\bar{L}_{\mathbf{p}_{1:T}}]$  which is different from  $\mathbb{E}[\bar{L}_{\mathbf{q}_{1:T}}]$  as stated in Theorem 1. Since  $\mathbf{p}_t = (1 - \delta)\mathbf{q}_t + \frac{\delta}{K} \mathbf{1}_K$ , the expected regret of OKS should be redefined as follows

$$\begin{aligned} \forall i \in [K], f \in \mathcal{H}_i, \quad & \mathbb{E}[\bar{L}_{\mathbf{p}_{1:T}}] - L_T(f) \\ & \leq \delta \mathbb{E}[\bar{L}_{\frac{1}{K} \mathbf{1}}] - \delta L_T(f) + \frac{\|f\|_{\mathcal{H}_i}^2}{2\lambda} + \frac{\lambda K T G^2}{2\delta} + \frac{\eta K T \ell_{\max}^2}{2(1-\delta)} + \frac{\ln K}{\eta} \\ & \leq \delta T \ell_{\max} + \frac{\|f\|_{\mathcal{H}_i}^2}{2\lambda} + \frac{\lambda K T G^2}{2\delta} + \frac{\eta K T \ell_{\max}^2}{2(1-\delta)} + \frac{\ln K}{\eta}. \end{aligned}$$

To minimize the upper bound, let  $\delta = (G/\ell_{\max})^{\frac{2}{3}}K^{\frac{1}{3}}T^{-\frac{1}{3}}$ ,  $\lambda = \sqrt{\delta/(KTG^2)}$  and  $\eta = \sqrt{2(1-\delta)\ln K}/\sqrt{KT\ell_{\max}^2}$ . The upper bound is  $O((\|f\|_{\mathcal{H}_i}^2 + 1)K^{\frac{1}{3}}T^{\frac{2}{3}})$ .

*Remark 2.* OKS is essentially an offline kernel selection algorithm, since it aims to output a hypothesis following  $\mathbf{q}_T$  for test datasets (see line 8 in Algorithm 1). Thus Theorem 1 defines the expected regret using  $\{\mathbf{q}_1, \dots, \mathbf{q}_T\}$ , and the  $O((\|f\|_{\mathcal{H}_i}^2 + 1)\sqrt{KT})$  bound is reasonable. For online kernel selection, we focus on the online prediction performance. Since OKS selects  $f_{t,I_t}$  following  $\mathbf{p}_t$ , the expected regret should be defined using  $\{\mathbf{p}_1, \dots, \mathbf{p}_T\}$ .

We find that the dependence on  $O(K^{\frac{1}{3}}T^{\frac{2}{3}})$  comes from the term  $\frac{\lambda K T G^2}{2\delta}$  which upper bounds the cumulative variance of gradient estimators, i.e.,

$$\frac{\lambda}{2}\mathbb{E}\left[\sum_{t=1}^T\|\tilde{\nabla}_{t,i}\|_{\mathcal{H}_i}^2\right]\leq\frac{\lambda K T G^2}{2\delta},\tilde{\nabla}_{t,i}=\frac{\nabla_{t,i}}{p_{t,i}}\mathbb{I}_{i=I_t},\nabla_{t,i}=\nabla_{f_{t,i}}\ell(f_{t,i}(\mathbf{x}_t),y_t).$$

Next we give a simple analysis. To start with, it can be verified that

$$\mathbb{E}\left[\|\tilde{\nabla}_{t,i}\|_{\mathcal{H}_i}^2\right]=\mathbb{E}\left[p_{t,i}\frac{\|\nabla_{t,i}\|_{\mathcal{H}_i}^2}{p_{t,i}^2}+(1-p_{t,i})\cdot 0\right]\leq\mathbb{E}\left[\max_{t=1,\dots,T}\left(\frac{1}{p_{t,i}}\right)\|\nabla_{t,i}\|_{\mathcal{H}_i}^2\right].$$

Recalling that  $p_{t,i}\geq\frac{\delta}{K}$ ,  $\forall i\in[K],t\in[T]$ . Summing over  $t=1,\dots,T$  yields

$$\sum_{t=1}^T\mathbb{E}\left[\|\tilde{\nabla}_{t,i}\|_{\mathcal{H}_i}^2\right]\leq\frac{K}{\delta}\sum_{t=1}^T\mathbb{E}\left[\|\nabla_{t,i}\|_{\mathcal{H}_i}^2\right]\leq\frac{K T G^2}{\delta}.$$

The regret bound of online gradient descent (this can be found in our supplementary materials) depends on  $\frac{\lambda}{2}\mathbb{E}\left[\sum_{t=1}^T\|\tilde{\nabla}_{t,i}\|_{\mathcal{H}_i}^2\right]\leq\frac{\lambda K T G^2}{2\delta}$ . Thus it is the high variance of  $\tilde{\nabla}_{t,i}$  that causes the  $O(K^{\frac{1}{3}}T^{\frac{2}{3}})$  regret bound.

OKS selects a hypothesis per-round, reduces the time complexity to  $O(t)$  but damages the regret bound. It was proved selecting two hypotheses can improve the regret bound to  $\tilde{O}((\|f\|_{\mathcal{H}_i}^2 + 1)\sqrt{KT})$  [11]. A natural question arises: will selecting a hypothesis induce worse regret bound than selecting two hypotheses? From the perspective of algorithm design, we concentrate on the question:

- does there exist some algorithm selecting a hypothesis (or under bandit feedback) that can improve the  $O((\|f\|_{\mathcal{H}_i}^2 + 1)K^{\frac{1}{3}}T^{\frac{2}{3}})$  bound?

### 3 Improved Regret bounds for Smooth Loss Functions

In this section, we propose the OKS++ algorithm using an adaptive parameter tuning scheme for OKS. Specifically, we reset the value of  $\delta, \eta$  and  $\lambda$  in Theorem 1 and prove data-dependent regret bounds for smooth loss functions. Such regret bounds can improve the previous worst-case bound if most of candidate kernel functions match well with the data. Although OKS++ just resets the value of parameters, deriving the new regret bounds requires novel and non-trivial analysis. To start with, we define the smooth loss functions.

**Assumption 1 (Smoothness condition)**  $\ell(\cdot, \cdot)$  is convex w.r.t. the first parameter. Denote by  $\ell'(a, b) = \frac{d\ell(a, b)}{da}$ . For any  $f(\mathbf{x})$  and  $y$ , there is a constant  $C_0 > 0$  such that

$$|\ell'(f(\mathbf{x}), y)|^\nu \leq C_0 \ell(f(\mathbf{x}), y), \quad \nu \in \{1, 2\}.$$

Zhang et al. [21] considered online kernel learning under smooth loss functions with  $\nu = 1$ . The logistic loss  $\ell(f(\mathbf{x}), y) = \ln(1 + \exp(-yf(\mathbf{x})))$  satisfies Assumption 1 with  $\nu = 1$  and  $C_0 = 1$ . The square loss  $\ell(f(\mathbf{x}), y) = (f(\mathbf{x}) - y)^2$  and the squared hinge loss  $\ell(f(\mathbf{x}), y) = (\max\{0, 1 - yf(\mathbf{x})\})^2$  satisfy Assumption 1 with  $\nu = 2$  and  $C_0 = 4$ .

Let  $U > 0$  be a constant. We define  $K$  restricted hypothesis spaces.  $\forall i \in [K]$ , let  $\mathbb{H}_i = \{f \in \mathcal{H}_i : \|f\|_{\mathcal{H}_i} \leq U\}$ . Then it is natural to derive Assumption 2.

**Assumption 2**  $\forall \kappa_i \in \mathcal{K}$  and  $\forall f \in \mathbb{H}_i$ , there exists a constant  $G > 0$  such that  $\max_{t \in [T]} |\ell'(f(\mathbf{x}_t), y_t)| \leq G$ .

It can be verified that many loss functions satisfy the assumption and  $G$  may depend on  $U$ . For instance, if  $\ell$  is the square loss, then  $G \leq 2(U + 1)$ . For simplicity, denote by  $c_{t,i} = \ell(f_{t,i}(\mathbf{x}_t), y_t)$  for all  $i \in [K]$  and  $t \in [T]$ . It can be verified that  $\max_{t,i} c_{t,i}$  is bounded and depends on  $U$ . Then our algorithm updates  $\mathbf{q}_t$  using  $c_t$  (see line 4 and line 6 in Algorithm 1). Since we use restricted hypothesis spaces, our algorithm changes line 5 in Algorithm 1 as follows

$$f_{t+1, I_t} = \arg \min_{f \in \mathbb{H}_{I_t}} \left\| f - \left( f_{t, I_t} - \lambda_{t, I_t} \frac{\nabla_{f_{t, I_t}} \ell(f_{t, I_t}(\mathbf{x}_t), y_t)}{p_{t, I_t}} \right) \right\|_{\mathcal{H}_{I_t}}^2. \quad (2)$$

Except for  $\{\lambda_{t,i}\}_{i=1}^K$ , our algorithm also uses time-variant  $\delta_t$  and  $\eta_t$ . We omit the pseudo-code of OKS++ since it is similar with Algorithm 1.

Next we show the regret bound. For simplicity, let  $\tilde{C}_{t,K} = \sum_{\tau=1}^t \sum_{i=1}^K \tilde{c}_{\tau,i}$  where  $\tilde{c}_{\tau,i} = \frac{c_{\tau,i}}{p_{\tau,i}} \mathbb{1}_{I_\tau=i}$ , and  $\bar{L}_T = \sum_{j=1}^K L_T(f_j^*)$  where  $L_T(f_j^*) = \min_{f \in \mathbb{H}_j} L_T(f)$ .

**Theorem 2.** Let  $\ell$  satisfy Assumption 1 with  $\nu = 1$  and Assumption 2. Let

$$\delta_t = \frac{(GC_0)^{\frac{1}{3}} (UK)^{\frac{2}{3}}}{2 \max \left\{ (GC_0)^{\frac{1}{3}} (UK)^{\frac{2}{3}}, 2\tilde{C}_{t,K}^{\frac{1}{3}} \right\}}, \eta_t = \frac{\sqrt{2 \ln K}}{\sqrt{1 + \sum_{\tau=1}^t \sum_{i=1}^K q_{\tau,i} \tilde{c}_{\tau,i}^2}},$$

$$\forall i \in [K], \lambda_{t,i} = \frac{U^{\frac{4}{3}} (\max \{ GC_0 U^2 K^2, 8\tilde{C}_{t,K} \})^{-\frac{1}{6}}}{\sqrt{4/3} K^{\frac{1}{6}} (GC_0)^{\frac{1}{3}} \sqrt{1 + \Delta_{t,i}}}, \Delta_{t,i} = \sum_{\tau=1}^t \frac{\ell(f_{\tau,i}(\mathbf{x}_\tau), y_\tau)}{p_{\tau,i}} \mathbb{1}_{I_\tau=i}.$$

Then the expected regret of OKS++ satisfies,  $\forall i \in [K]$ ,

$$\mathbb{E} [\bar{L}_{\mathbf{p}_{1:T}}] - L_T(f_i^*) = O \left( U^{\frac{2}{3}} (GC_0)^{\frac{1}{3}} K^{-\frac{1}{3}} \bar{L}_T^{\frac{2}{3}} + U^{\frac{2}{3}} (GC_0)^{\frac{1}{3}} K^{\frac{1}{6}} \bar{L}_T^{\frac{1}{6}} L_T^{\frac{1}{2}}(f_i^*) \right).$$

Let  $\ell$  satisfy Assumption 1 with  $\nu = 2$ . Let  $G = 1$  in  $\delta_t$  and  $\lambda_{t,i}$ .  $\eta_t$  keeps unchanged. Then the expected regret of OKS++ satisfies

$$\forall i \in [K], \mathbb{E} [\bar{L}_{\mathbf{p}_{1:T}}] - L_T(f_i^*) = O \left( U^{\frac{2}{3}} C_0^{\frac{1}{3}} K^{-\frac{1}{3}} \bar{L}_T^{\frac{2}{3}} + U^{\frac{2}{3}} C_0^{\frac{1}{3}} K^{\frac{1}{6}} \bar{L}_T^{\frac{1}{6}} L_T^{\frac{1}{2}}(f_i^*) \right).$$

The values of  $\lambda_{t,i}$ ,  $\delta_t$  and  $\eta_t$  which depend on the observed losses, are important to obtain the data-dependent bounds. Beside, it is necessary to set different  $\lambda_{t,i}$  for each  $i \in [K]$ . OKS sets a same  $\lambda$ . Thus the changes on the values of  $\delta$ ,  $\eta$  and  $\lambda$  are non-trivial. Our analysis is also non-trivial. OKS++ sets time-variant parameters and does not require prior knowledge of the nature of the data.

Now we compare our results with the regret bound in Theorem 1. The main difference is that we replace  $KT$  with a data-dependent complexity  $\bar{L}_T$ . In the worst case,  $\bar{L}_T = O(KT)$  and our regret bound is  $O(K^{\frac{1}{3}}T^{\frac{2}{3}})$  which is same with the result in Theorem 1. In some benign environments, we expect that  $\bar{L}_T \ll KT$  and our regret bound would be smaller. For instance, if  $L_T(f_i^*) = o(T)$  for all  $i \in [K]$ , then our regret is  $o(T^{\frac{2}{3}})$  improving the result in Theorem 1. If there are only  $M < K$  hypothesis spaces such that  $L_T(f_i^*) = O(T)$ , where  $M$  is independent of  $K$ , then our regret bound is  $O((MT)^{\frac{2}{3}}K^{-\frac{1}{3}})$ . Such a result still improves the dependence on  $K$ . A more interesting result is that, if  $L_T(f_i^*) = O(T^{\frac{3}{4}})$  for all  $i \in [K]$ , then OKS++ achieves a  $O(K^{\frac{1}{3}}\sqrt{T})$  regret bound which is better than the  $\bar{O}(\text{poly}(\|f\|_{\mathcal{H}_i})\sqrt{KT})$  bound achieved by B(AO)<sub>2</sub>KS [11].

## 4 Improved Regret bound for Lipschitz Loss Functions

In this section, we consider Lipschitz loss functions and propose a new algorithm with improved worst-case regret bound.

### 4.1 Algorithm

For the sake of clarity, we decompose OKS into two levels. At the outer level, it uses a procedure similar with Exp3 [3] to update  $\mathbf{p}_t$  and  $\mathbf{q}_t$ . At the inner level, it updates  $f_{t,I_t}$  using online gradient descent. Exp3 can be derived from online mirror descent framework with negative entropy regularizer [1], i.e.,

$$\nabla_{\mathbf{q}'_{t+1}} \psi_t(\mathbf{q}'_{t+1}) = \nabla_{\mathbf{q}_t} \psi_t(\mathbf{q}_t) - \tilde{c}_t, \quad \mathbf{q}_{t+1} = \arg \min_{\mathbf{q} \in \Delta_K} \mathcal{D}_{\psi_t}(\mathbf{q}, \mathbf{q}'_{t+1}), \quad (3)$$

where  $\psi_t(\mathbf{p}) = \sum_{i=1}^K \frac{1}{\eta} p_i \ln p_i$  is the negative entropy and  $\mathcal{D}_{\psi_t}(\mathbf{p}, \mathbf{q}) = \psi_t(\mathbf{p}) - \psi_t(\mathbf{q}) - \langle \nabla \psi_t(\mathbf{q}), \mathbf{p} - \mathbf{q} \rangle$  is the Bregman divergence. Different regularizer yields different algorithm. We will use  $\psi_t(\mathbf{p}) = \sum_{i=1}^K \frac{-\alpha}{\eta_{t,i}} p_i^{\frac{1}{\alpha}}$ ,  $\alpha > 1$ , which slightly modifies the  $\alpha$ -Tsallis entropy [17, 23]. We also use the increasing learning rate scheme in [1], that is  $\eta_{t,i}$  is increasing. The reason is that if  $\eta_{t,i}$  is increasing, then there will be a negative term in the regret bound which can be used to control the large variance of gradient estimator, i.e.,  $\mathbb{E} \left[ \sum_{t=1}^T \|\tilde{\nabla}_{t,i}\|_{\mathcal{H}_i}^2 \right]$  (see Section 2.2). If we use the log-barrier [1] or  $\alpha$ -Tsallis entropy with  $\alpha = 2$  [2, 23], then the regret bound will increase a  $O(\ln T)$  factor. This factor can be reduced to  $O(\ln^{\frac{2}{3}} T)$  for  $\alpha \geq 3$ . We choose  $\alpha = 8$  for achieving a small regret bound.

At the beginning of round  $t$ , our algorithm first samples  $I_t \sim \mathbf{p}_t$  and outputs the prediction  $f_{t,I_t}(\mathbf{x}_t)$  or  $\text{sign}(f_{t,I_t}(\mathbf{x}_t))$ . Next our algorithm updates  $f_{t,I_t}$

**Algorithm 2** IOKS

---

**Input:**  $\mathcal{K} = \{\kappa_1, \dots, \kappa_K\}$ ,  $\alpha = 8$ ,  $\nu = e^{\frac{2}{3 \ln T}}$ ,  $\eta$   
**Initialization:**  $\{f_{1,i} = 0, \eta_{1,i} = \eta\}_{i=1}^K$ ,  $\mathbf{q}_1 = \mathbf{p}_1 = \frac{1}{K} \mathbf{1}_K$

- 1: **for**  $t = 1, \dots, T$  **do**
- 2:   Receive  $\mathbf{x}_t$
- 3:   Sample a kernel  $\kappa_{I_t}$  where  $I_t \sim \mathbf{p}_t$
- 4:   Output  $\hat{y}_t = f_{t,I_t}(\mathbf{x}_t)$  or  $\text{sign}(\hat{y}_t)$
- 5:   Compute  $f_{t+1,I_t}$  according to (2)
- 6:   Compute  $\tilde{c}_{t,I_t}$  according to (4)
- 7:    $\forall i \in [K]$ , compute  $q_{t+1,i}$  according to (5)
- 8:   Compute  $\mathbf{p}_{t+1} = (1 - \delta)\mathbf{q}_{t+1} + \frac{\delta}{K} \mathbf{1}_K$
- 9:   **for**  $i = 1, \dots, K$  **do**
- 10:     **if**  $\frac{1}{p_{t+1,i}} > \rho_{t,i}$  **then**
- 11:        $\rho_{t+1,i} = \frac{2}{p_{t+1,i}}$ ,  $\eta_{t+1,i} = \nu \eta_{t,i}$
- 12:     **else**
- 13:        $\rho_{t+1,i} = \rho_{t,i}$ ,  $\eta_{t+1,i} = \eta_{t,i}$
- 14:     **end if**
- 15:   **end for**
- 16: **end for**

---

following (2).  $\forall i \in [K]$ , let  $c_{t,i} = \ell(f_{t,i}(\mathbf{x}_t), y_t) / \ell_{\max} \in [0, 1]$ . We redefine  $\tilde{c}_t$  by

$$\text{if } p_{t,I_t} \geq \max_i \eta_{t,i}, \text{ then } \tilde{c}_{t,i} = \frac{c_{t,i} \mathbb{1}_{i=I_t}}{p_{t,i}}, \text{ otherwise } \tilde{c}_{t,i} = \frac{c_{t,i} \cdot \mathbb{1}_{i=I_t}}{p_{t,i} + \max_i \eta_{t,i}}. \quad (4)$$

It is worth mentioning that  $\tilde{c}_t$  is essentially different from that in OKS, and aims to ensure that (3) has a computationally efficient solution as follows

$$\forall i \in [K], q_{t+1,i} = \left( q_{t,i}^{-\frac{7}{8}} + \eta_{t,i}(\tilde{c}_{t,i} - \mu^*) \right)^{-\frac{8}{7}}, \quad (5)$$

where  $\mu^*$  can be solved using binary search. We show more details in the supplementary materials. We name this algorithm IOKS (Improved OKS).

## 4.2 Regret bound

**Assumption 3 (Lipschitz condition)**  $\ell(\cdot, \cdot)$  is convex w.r.t. the first parameter. There is a constant  $G_1$  such that  $\forall \kappa_i \in \mathcal{K}$ ,  $f \in \mathbb{H}_i$ ,  $\|\nabla_f \ell(f(\mathbf{x}), y)\|_{\mathcal{H}_i} \leq G_1$ .

**Theorem 3.** Let  $\ell$  satisfy Assumption 3. Let  $\delta = T^{-\frac{3}{4}}$ ,

$$\eta = \frac{3\ell_{\max} K^{\frac{3}{8}}}{2UG_1\sqrt{T \ln T}}, \quad \forall i \in [K], \lambda_{t,i} = \frac{U}{\sqrt{2}\sqrt{1 + \sum_{\tau=1}^t \|\tilde{\nabla}_{\tau,i}\|_{\mathcal{H}_i}^2}}. \quad (6)$$

Let  $T \geq 40$ . Then the expected regret of IOKS satisfies,

$$\forall i \in [K], f \in \mathbb{H}_i, \mathbb{E}[\bar{L}_{\mathbf{p}_{1:T}}] - L_T(f) = O\left(UG_1\sqrt{KT \ln^{\frac{2}{3}} T} + \frac{\ell_{\max}^3 K^{\frac{11}{4}}}{U^2 G_1^2 \ln T}\right).$$



$\ell_{\max}$  is a normalizing constant and can be computed given the loss function, such as  $\ell_{\max} \leq U + 1$  in the case of absolute loss. Next we compare our regret bound with previous results. On the positive side, IOKS gives a  $O(U\sqrt{KT} \ln^{\frac{2}{3}} T)$  bound which asymptotically improves the  $O(K^{\frac{1}{3}}T^{\frac{2}{3}})$  bound achieved by OKS. On the negative side, if  $T$  is small, then  $\sqrt{KT} \ln^{\frac{2}{3}} T > K^{\frac{1}{3}}T^{\frac{2}{3}}$  and thus IOKS is slightly worse than OKS. B(AO)<sub>2</sub>KS [11] which selects two hypotheses per-round, can provide a  $\tilde{O}(\text{poly}(\|f\|_{\mathcal{H}_i})\sqrt{KT})$  bound which is same with our result.

We further compare the implementation of IOKS and OKS. It is obvious that OKS is easier than IOKS, since IOKS uses binary search to compute  $\mathbf{q}_{t+1}$  (see (5)). The computational cost of binary search can be omitted since the main computational cost comes from the computing of  $f_{t,I_t}(\mathbf{x}_t)$  which is  $O(t)$ .

## 5 Application to Online Kernel Selection with Time Constraint

In practice, online algorithms must face time constraint. We assume that there is a time budget of  $\mathcal{R}$  quanta. Both OKS++ and IOKS suffer a  $O(t)$  per-round time complexity, and do not satisfy the time constraint. In this section, we will use random features [12] to approximate kernel functions and apply our two algorithms to online kernel selection with time constraint [9].

We consider kernel function  $\kappa(\mathbf{x}, \mathbf{v})$  that can be decomposed as follows

$$\kappa(\mathbf{x}, \mathbf{v}) = \int_{\Omega} \phi_{\kappa}(\mathbf{x}, \omega) \phi_{\kappa}(\mathbf{v}, \omega) d\mu_{\kappa}(\omega), \quad \forall \mathbf{x}, \mathbf{v} \in \mathcal{X} \quad (7)$$

where  $\phi_{\kappa} : \mathcal{X} \times \Omega \rightarrow \mathbb{R}$  is the eigenfunctions and  $\mu_{\kappa}(\cdot)$  is a distribution function on  $\Omega$ . Let  $p_{\kappa}(\cdot)$  be the density function of  $\mu_{\kappa}(\cdot)$ . We can approximate the integral via Monte-Carlo sampling. We sample  $\{\omega_j\}_{j=1}^D \sim p_{\kappa}(\omega)$  independently and compute  $\tilde{\kappa}(\mathbf{x}, \mathbf{v}) = \frac{1}{D} \sum_{j=1}^D \phi_{\kappa}(\mathbf{x}, \omega_j) \phi_{\kappa}(\mathbf{v}, \omega_j)$ . For any  $f \in \mathcal{H}_{\kappa}$ , let  $f(\mathbf{x}) = \int_{\Omega} \alpha(\omega) \phi_{\kappa}(\mathbf{x}, \omega) p_{\kappa}(\omega) d\omega$ . We can approximate  $f(\mathbf{x})$  by  $\hat{f}(\mathbf{x}) = \frac{1}{D} \sum_{j=1}^D \alpha(\omega_j) \phi_{\kappa}(\mathbf{x}, \omega_j)$ . It can be verified that  $\mathbb{E}[\hat{f}(\mathbf{x})] = f(\mathbf{x})$ . Such an approximation scheme also defines an explicit feature mapping denoted by  $z(\mathbf{x}) = \frac{1}{\sqrt{D}}(\phi_{\kappa}(\mathbf{x}, \omega_1), \dots, \phi_{\kappa}(\mathbf{x}, \omega_D))$ . The approximation scheme is the so called random features [12].  $\forall \kappa_i \in \mathcal{K}$ , we define two hypothesis spaces [13] as follows

$$\mathbb{H}_i = \left\{ f(\mathbf{x}) = \int_{\Omega} \alpha(\omega) \phi_{\kappa_i}(\mathbf{x}, \omega) p_{\kappa_i}(\omega) d\omega \mid |\alpha(\omega)| \leq U \right\},$$

$$\mathcal{F}_i = \left\{ \hat{f}(\mathbf{x}) = \sum_{j=1}^{D_i} \alpha_j \phi_{\kappa_i}(\mathbf{x}, \omega_j) \mid |\alpha_j| \leq \frac{U}{D_i} \right\}.$$

We can rewrite  $\hat{f}(\mathbf{x}) = \mathbf{w}^T z_i(\mathbf{x})$ , where  $\mathbf{w} = \sqrt{D_i}(\alpha_1, \dots, \alpha_{D_i}) \in \mathbb{R}^{D_i}$ . Let  $\mathcal{W}_i = \{\mathbf{w} \in \mathbb{R}^{D_i} \mid \|\mathbf{w}\|_{\infty} \leq \frac{U}{\sqrt{D_i}}\}$ . It can be verified that  $\|\mathbf{w}\|_2^2 \leq U^2$ . For all  $\kappa_i$  satisfying (7), there is a constant  $B_i$  such that  $|\phi_{\kappa_i}(\mathbf{x}, \omega_j)| \leq B_i$  for all  $\omega_j \in \Omega$  and  $\mathbf{x} \in \mathcal{X}$  [10]. Thus we have  $|f(\mathbf{x})| \leq UB_i$  for any  $f \in \mathbb{H}_i$  and  $f \in \mathcal{F}_i$ .

Next we define the time budget  $\mathcal{R}$  and then present an assumption that establishes a reduction from  $\mathcal{R}$  to  $D_i$ .

**Definition 1 (Time Budget [9]).** *Let the interval of arrival time between  $\mathbf{x}_t$  and  $\mathbf{x}_{t+1}$ ,  $t = 1, \dots, T$  be less than  $\mathcal{R}$  quanta. A time budget of  $\mathcal{R}$  quanta is the maximal time interval that any online kernel selection algorithm outputs the prediction of  $\mathbf{x}_t$  and  $\mathbf{x}_{t+1}$ .*

**Assumption 4** *For each  $\kappa_i \in \mathcal{K}$  satisfying (7), there exist online leaning algorithms that can run in some  $\mathcal{F}_i$  whose maximal dimension is  $D_i = \beta_{\kappa_i} \mathcal{R}$  within a time budget of  $\mathcal{R}$  quanta, where  $\beta_{\kappa_i} > 0$  is a constant depending on  $\kappa_i$ .*

The online gradient descent algorithm (OGD) satisfies Assumption 4. The main time cost of OGD comes from computing the feature mapping. For shift-invariant kernels, it requires  $O(D_i d)$  time complexity [12]. For the Gaussian kernel, it requires  $O(D_i \log(d))$  time complexity [8, 20]. Thus the per-round time complexity of OGD is linear with  $D_i$ . Since the running time of algorithm is linear with the time complexity, it natural to assume that  $\mathcal{R} = \Theta(D_i)$ .

## 5.1 Algorithm

At any round  $t$ , our algorithm evaluates a hypothesis and avoids allocating the time budget. Thus we can construct  $\mathcal{F}_i$  satisfying  $D_i = \beta_{\kappa_i} \mathcal{R}$ . Our algorithm is extremely simple, that is, we just need to run OKS++ or IOKS in  $\{\mathcal{F}_i\}_{i=1}^K$ . It is worth mentioning that, learning  $\{\hat{f}_{t,i} \in \mathcal{F}_i\}_{t=1}^T$  is equivalent to learn  $\{\mathbf{w}_t^i \in \mathcal{W}_i\}_{t=1}^T$ , where  $\hat{f}_{t,i}(\mathbf{x}_t) = (\mathbf{w}_t^i)^\top z_i(\mathbf{x}_t)$ . We replace the update (2) with (8),

$$\begin{aligned} \tilde{\mathbf{w}}_{t+1}^i &= \mathbf{w}_t^i - \lambda_{t,i} \nabla_{\mathbf{w}_t^i} \ell(\hat{f}_{t,i}(\mathbf{x}_t), y_t) \frac{1}{p_{t,i}} \mathbb{I}_{i=I_t}, \\ \mathbf{w}_{t+1}^i &= \arg \min_{\mathbf{w} \in \mathcal{W}_i} \|\mathbf{w} - \tilde{\mathbf{w}}_{t+1}^i\|_2^2. \end{aligned} \tag{8}$$

The solution of the projection operation in (8) is as follows,

$$\forall j = 1, \dots, D_i, \quad w_{t+1,j}^i = \min \left\{ 1, \frac{U}{|\tilde{w}_{t+1,j}^i| \sqrt{D_i}} \right\} \tilde{w}_{t+1,j}^i.$$

The time complexity of projection is  $O(D_i)$  and thus can be omitted relative to the time complexity of computing feature mapping. We separately name the two algorithms RF-OKS++ (Random Features for OKS++) and RF-IOKS (Random Features for IOKS). We show the pseudo-codes in the supplementary materials due to the space limit. The pseudo-codes are similar with OKS++ and IOKS.

*Remark 3.* The application of random features to online kernel algorithms is not a new idea [18, 16, 6]. Previous algorithms did not restrict hypothesis spaces, while our algorithms consider restricted hypothesis spaces, i.e.,  $\mathbb{H}_i$  and  $\mathcal{F}_i$ . This is one of the differences between our algorithms and previous algorithms. The restriction on the hypothesis spaces is necessary since we must require  $\|\mathbf{w}_t^i\|_2 \leq U$  for any  $i \in [K]$  and  $t \in [T]$ .

## 5.2 Regret Bound

**Theorem 4.** *Let  $\ell$  satisfy Assumption 1 with  $\nu = 1$  and Assumption 2. Let  $\delta_t$ ,  $\eta_t$  and  $\{\lambda_{t,i}\}_{i=1}^K$  follow Theorem 2. For a fixed  $\delta \in (0, 1)$ , let  $\mathcal{R}$  satisfy  $D_i > \frac{32}{9}C_0^2U^2B_i^2 \ln \frac{1}{\delta}$ ,  $\forall i \in [K]$ . Under Assumption 4, with probability at least  $1 - \delta$ , the expected regret of RF-OKS++ satisfies*

$$\forall i \in [K], \mathbb{E} [\bar{L}_{\mathbf{p}_{1:T}}] - L_T(f_i^*) = O \left( \frac{C_0UB_i}{\sqrt{\beta_{\kappa_i}\mathcal{R}}} L_T(f_i^*) \sqrt{\ln \frac{KT}{\delta}} \right. \\ \left. + U^{\frac{2}{3}}(GC_0)^{\frac{1}{3}}K^{-\frac{1}{3}}\bar{L}_T^{\frac{2}{3}} + U^{\frac{2}{3}}(GC_0)^{\frac{1}{3}}K^{\frac{1}{6}}\bar{L}_T^{\frac{1}{6}}L_T^{\frac{1}{2}}(f_i^*) \right).$$

*Let  $\ell$  satisfy Assumption 1 with  $\nu = 2$ . Let  $G = 1$  in  $\delta_t$  and  $\lambda_{t,i}$ .  $\eta_t$  keeps unchanged. For a fixed  $\delta \in (0, 1)$ , with probability at least  $1 - \delta$ , the expected regret of RF-OKS++ satisfies*

$$\forall i \in [K], \mathbb{E} [\bar{L}_{\mathbf{p}_{1:T}}] - L_T(f_i^*) = O \left( UB_i \frac{\sqrt{C_0TL_T(f_i^*)}}{\sqrt{\beta_{\kappa_i}\mathcal{R}}} \sqrt{\ln \frac{KT}{\delta}} \right. \\ \left. + \frac{C_0U^2B_i^2T}{\beta_{\kappa_i}\mathcal{R}} \ln \frac{KT}{\delta} + U^{\frac{2}{3}}C_0^{\frac{1}{3}}K^{-\frac{1}{3}}\bar{L}_T^{\frac{2}{3}} + U^{\frac{2}{3}}C_0^{\frac{1}{3}}K^{\frac{1}{6}}\bar{L}_T^{\frac{1}{6}}L_T^{\frac{1}{2}}(f_i^*) \right).$$

The regret bounds depend on  $\frac{L_T(f_i^*)}{\sqrt{\mathcal{R}}}$  or  $\frac{1}{\sqrt{\mathcal{R}}}\sqrt{TL_T(f_i^*)} + \frac{T}{\mathcal{R}}$ . The larger the time budget is, the smaller the regret bound will be, which proves a trade-off between regret bound and time constraint. If  $L_T(f_i^*) \ll T$ , then RF-OKS++ can achieve a sublinear regret bound under a small time budget.

**Theorem 5.** *Let  $\ell$  satisfy Assumption 2 and Assumption 3. Let  $\{\lambda_{t,i}\}_{i=1}^K$ ,  $\eta$  and  $\delta$  follow Theorem 3. Under Assumption 4, with probability at least  $1 - \delta$ , the expected regret of RF-IOKS satisfies,  $\forall i \in [K], \forall f \in \mathbb{H}_i$ ,*

$$\mathbb{E} [\bar{L}_{\mathbf{p}_{1:T}}] - L_T(f) = O \left( UG_1\sqrt{KT} \ln^{\frac{2}{3}}T + \frac{\ell_{\max}^3K^{\frac{11}{4}}}{U^2G_1^2\sqrt{\ln T}} + \frac{GB_iUT}{\sqrt{\beta_{\kappa_i}\mathcal{R}}} \sqrt{\ln \frac{KT}{\delta}} \right).$$

The regret bound depends on  $\frac{T}{\sqrt{\mathcal{R}}}$  which also proves a trade-off between regret bound and time constraint. Achieving a  $\tilde{O}(T^\alpha)$  bound requires  $\mathcal{R} = \Omega(T^{2(1-\alpha)})$ ,  $\alpha \in [\frac{1}{2}, 1)$ . The regret bounds in Theorem 4 depend on  $L_T(f_i^*)$ , while the regret bound in Theorem 5 depends on  $T$ . Under a same time budget  $\mathcal{R}$ , if  $L_T(f_i^*) \ll T$ , then RF-OKS++ enjoys better regret bounds than RF-IOKS.

## 5.3 Comparison With Previous Results

For online kernel selection with time constraint, if the loss function is Lipschitz continuous, then there is a  $\Omega(\|f_i^*\|_{\mathcal{H}_i} \max\{\sqrt{T}, \frac{T}{\sqrt{\mathcal{R}}}\})$  lower bound on expected regret [9]. Theorem 5 gives a nearly optimal upper bound. LKMBooks [9] gives a  $O(\sqrt{T} \ln K + \|f\|_{\mathcal{H}_i}^2 \max\{\sqrt{T}, \frac{T}{\sqrt{\mathcal{R}}}\})$  bound in the case of  $K \leq d$ , and thus

is slightly better than RF-IOKS. LKMBooks selects  $K$  hypotheses per-round. RF-IOKS just selects a hypothesis per-round and is suitable for  $K > d$ .

For smooth loss functions, the dominated terms in Theorem 4 are  $O(\frac{L_T(f_i^*)}{\sqrt{\mathcal{R}}})$  and  $O(\frac{1}{\sqrt{\mathcal{R}}} \sqrt{TL_T(f_i^*)} + \frac{T}{\mathcal{R}})$ . If the optimal kernel  $\kappa_{i^*}$  matches well with the data, that is,  $L_T(f_{i^*}^*) \ll T$ , then  $O(\frac{L_T(f_{i^*}^*)}{\sqrt{\mathcal{R}}})$  and  $O(\frac{1}{\sqrt{\mathcal{R}}} \sqrt{TL_T(f_{i^*}^*)})$  are much smaller than  $O(\frac{T}{\sqrt{\mathcal{R}}})$ . To be specific, in the case of  $L_T(f_{i^*}^*) = o(T)$ , RF-OKS++ is better than LKMBooks within a same time budget  $\mathcal{R}$ .

Our algorithms are similar with Raker [16] which also adopts random features. Raker selects  $K$  hypotheses and provides a  $\tilde{O}((\sqrt{\ln K} + \|f\|_1^2)\sqrt{T} + \|f\|_1 \frac{T}{\sqrt{\mathcal{R}}})$  bound, where  $f = \sum_{t=1}^T \alpha_t \kappa_i(\mathbf{x}_t, \cdot)$  and  $\|f\|_1 = \|\alpha\|_1$ . The regret bounds of RF-OKS++ are better, since (i) they depend on  $L_T(f_i^*)$  and  $\sum_{j=1}^K L_T(f_j^*)$  while the regret bound of Raker depends on  $T$ ; (ii) they depend on  $U$ , while the regret bound of Raker depends on  $\|f\|_1$  which is hard to bound and explain.

## 6 Experiments

We adopt the Gaussian kernel  $\kappa(\mathbf{x}, \mathbf{v}) = \exp(-\frac{\|\mathbf{x}-\mathbf{v}\|_2^2}{2\sigma^2})$  and select 6 kernel widths  $\sigma = 2^{-2:1:3}$ . We choose four classification datasets (*magic04:19,020*, *phishing:11,055*, *a9a:32,561*, *SUSY:20,000*) and four regression datasets (*bank:8,192*, *elevators:16,599*, *aileron:13,750*, *Hardware:28,179*). The datasets are downloaded from UCI <sup>2</sup>, LIBSVM website <sup>3</sup> and WEKA. The features of all datasets are rescaled to fit in  $[-1, 1]$ . The target variables are rescaled in  $[0, 1]$  for regression and  $\{-1, 1\}$  for classification. We randomly permute the instances in the datasets 10 times and report the average results. All algorithms are implemented with R on a Windows machine with 2.8 GHz Core(TM) i7-1165G7 CPU <sup>4</sup>. We separately consider online kernel selection without and with time constraint.

### 6.1 Online kernel selection without time constraint

We compare OKS++, IOKS with OKS and aim to verify Theorem 2 and Theorem 3. We consider three loss functions: (i) the logistic loss satisfying Assumption 1 with  $\nu = 1$  and  $C_0 = 1$ ; (ii) the square loss satisfying Assumption 1 with  $\nu = 2$  and  $C_0 = 4$ ; (iii) the absolute loss which is Lipschitz continuous. We do not compare with B(AO)<sub>2</sub>KS [11], since it is only used for the hinge loss. If  $\ell$  is logistic loss, then we use classification datasets and measure the average mistake rate, i.e.,  $\text{AMR} := \frac{1}{T} \sum_{t=1}^T \mathbb{I}_{\hat{y}_t \neq y_t}$ , and set  $U = 15$ . Otherwise, we use regression datasets and measure the average loss, i.e.,  $\text{AL} := \frac{1}{T} \sum_{t=1}^T \ell(f_{t, I_t}(\mathbf{x}_t), y_t)$ , and set  $U = 1$ . The parameters of OKS++ and IOKS follow Theorem 2 and Theorem 3 where we change  $\eta = \frac{8\ell_{\max} K^{3/8}}{UG_1 \sqrt{T \ln T}}$  in Theorem 3 and set  $\ell_{\max} = 1$ . For OKS, we

<sup>2</sup> <http://archive.ics.uci.edu/ml/datasets.php>

<sup>3</sup> <https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>

<sup>4</sup> The codes are available at <https://github.com/JunfLi-TJU/OKS-Bandit>

**Table 2.** Online kernel selection without time constraint in the regime of logistic loss

Algorithm	phishing		a9a	
	AMR (%)	Time (s)	AMR (%)	Time (s)
OKS	13.80 ± 0.34	17.34 ± 1.48	19.65 ± 0.12	208.84 ± 31.16
IOKS	13.25 ± 0.28	6.58 ± 0.18	17.46 ± 0.12	103.91 ± 13.89
OKS++	<b>7.80 ± 0.49</b>	32.31 ± 3.98	<b>16.57 ± 0.31</b>	474.65 ± 117.43
Algorithm	magic04		SUSY	
	AMR (%)	Time (s)	AMR (%)	Time (s)
OKS	22.23 ± 0.22	6.31 ± 0.95	32.98 ± 0.66	9.97 ± 1.85
IOKS	21.50 ± 0.18	4.02 ± 0.11	31.75 ± 0.30	6.68 ± 0.15
OKS++	<b>17.88 ± 0.57</b>	11.06 ± 3.08	<b>27.84 ± 0.70</b>	19.88 ± 5.28

**Table 3.** Online kernel selection without time constraint in the regime of square loss

Algorithm	elevators		bank	
	AL	Time (s)	AL	Time (s)
OKS	0.0068 ± 0.0001	3.23 ± 0.25	0.0240 ± 0.0002	1.51 ± 0.17
IOKS	0.0077 ± 0.0001	4.08 ± 0.05	0.0252 ± 0.0002	1.57 ± 0.11
OKS++	<b>0.0046 ± 0.0001</b>	12.75 ± 3.12	<b>0.0205 ± 0.0006</b>	4.24 ± 0.76
Algorithm	aileron		Hardware	
	AL	Time (s)	AL	Time (s)
OKS	<b>0.0176 ± 0.0060</b>	6.94 ± 0.82	0.0012 ± 0.0000	53.84 ± 1.80
IOKS	0.0351 ± 0.0003	5.59 ± 0.08	0.0010 ± 0.0001	49.36 ± 1.14
OKS++	<b>0.0166 ± 0.0006</b>	22.79 ± 3.41	<b>0.0008 ± 0.0001</b>	114.47 ± 23.42

**Table 4.** Online kernel selection without time constraint in the regime of absolute loss

Algorithm	elevators		bank	
	AL	Time (s)	AL	Time (s)
OKS	0.0507 ± 0.0001	4.76 ± 0.17	0.0961 ± 0.0009	1.55 ± 0.13
IOKS	<b>0.0492 ± 0.0004</b>	5.20 ± 0.54	0.0961 ± 0.0008	1.64 ± 0.20
Algorithm	aileron		Hardware	
	AL	Time (s)	AL	Time (s)
OKS	<b>0.0723 ± 0.0005</b>	8.20 ± 0.19	<b>0.0105 ± 0.0001</b>	56.14 ± 1.07
IOKS	0.0771 ± 0.0007	9.86 ± 0.68	0.0155 ± 0.0002	52.01 ± 3.72

set  $\delta, \lambda$  and  $\eta$  according to Remark 1, where  $\lambda \in \{1, 5, 10, 25\} \cdot \sqrt{\delta/(KT)}$  and  $\ell_{\max} = G = 1$ . The results are shown in Table 2, Table 3 and Table 4.

Table 2 and Table 3 prove that OKS++ performs better than OKS and IOKS for smooth loss functions. The reason is that OKS++ adaptively tunes the parameters using the observed losses, while OKS and IOKS do not use this information to tune the parameters. The experimental results coincide with Theorem 2. Besides IOKS performs similar with OKS, since IOKS is only asymptotically better than OKS. If  $T$  is small, then the regret bound of OKS is smaller. The theoretical significance of IOKS is that it proves that selecting a hypothesis does not produce high information-theoretic cost in the worst case.

**Table 5.** Online kernel selection with time constraint in the regime of logistic loss

Algorithm	B-D	phishing		B-D	a9a	
		AMR (%)	$t_p * 10^5 (s)$		AMR (%)	$t_p * 10^5 (s)$
RF-OKS	500	14.61 ± 0.65	9.63	450	21.25 ± 0.12	11.61
LKMBooks	250	12.50 ± 1.03	9.46	220	20.06 ± 0.54	11.53
Raker	70	13.60 ± 1.00	9.35	90	24.08 ± 0.00	11.30
RF-IOKS	380	15.59 ± 0.39	9.66	380	22.99 ± 0.20	11.95
RF-OKS++	400	<b>9.15 ± 0.56</b>	9.20	400	<b>17.28 ± 0.29</b>	11.19

**Table 6.** Online kernel selection with time constraint in the regime of square loss

Algorithm	B-D	elevators		B-D	Hardware	
		AL * 10 <sup>2</sup>	$t_p * 10^5 (s)$		AL * 10 <sup>2</sup>	$t_p * 10^5 (s)$
RF-OKS	450	0.72 ± 0.02	6.47	420	0.13 ± 0.00	10.48
LKMBooks	220	0.90 ± 0.04	6.72	200	0.21 ± 0.01	10.76
Raker	40	0.70 ± 0.04	6.57	80	0.20 ± 0.00	10.25
RF-IOKS	380	0.89 ± 0.01	6.83	400	0.12 ± 0.01	10.20
RF-OKS++	400	<b>0.51 ± 0.02</b>	6.45	400	<b>0.09 ± 0.01</b>	10.31

**Table 7.** Online kernel selection with time constraint in the regime of absolute loss

Algorithm	B-D	elevators		B-D	Hardware	
		AL	$t_p * 10^5$		AL	$t_p * 10^5$
RF-OKS	530	<b>0.0515 ± 0.0004</b>	7.13	400	<b>0.0108 ± 0.0001</b>	10.39
LKMBooks	230	0.0550 ± 0.0014	7.35	200	0.0203 ± 0.0020	10.41
Raker	50	0.0550 ± 0.0012	7.41	80	0.0154 ± 0.0001	10.37
RF-IOKS	400	<b>0.0515 ± 0.0007</b>	7.63	400	0.0164 ± 0.0002	10.97

## 6.2 Online kernel selection with time constraint

We compare RF-OKS++, RF-IOKS with Raker [16], LKMBooks [9] and RF-OKS [19]. We construct RF-OKS by combining random features with OKS. The parameter setting of Raker and LKMBooks follows original paper, except that the learning rate of Raker is chosen from  $\{1, 5, 10, 25\} \cdot 1/\sqrt{T}$ . The parameter setting of RF-OKS++, RF-IOKS and RF-OKS is same with that of OKS++, IOKS and OKS, respectively. We limit time budget  $\mathcal{R}$  by fixing the number of random features. To be specific, we choose RF-OKS++ as the baseline and set  $D_i = D = 400$  for all  $i \in [K]$  satisfying the condition in Theorem 4. Let the average per-round running time of RF-OKS++ be  $t_p$ . We tune  $D$  or  $B$  of other algorithms for ensuring the same running time with  $t_p$ . The results are shown in Table 5, Table 6 and Table 7. In Table 7, we use RF-IOKS as the baseline.

For smooth loss functions, RF-OKS++ still performs best under a same time budget. The reason is also that RF-OKS++ adaptively tunes the parameters using the observed losses, while the other algorithms do not use the observed losses. For the square loss function, Theorem 4 shows the regret bound depends on  $O(\frac{1}{\sqrt{\mathcal{R}}}\sqrt{TL_T(f_i^*)})$  which becomes  $O(\frac{T}{\sqrt{\mathcal{R}}})$  in the worst case and

thus is same with previous results. To explain the contradiction, we recorded the cumulative square losses of RF-OKS++, i.e.,  $\sum_{t=1}^T (f_{t,I_t}(\mathbf{x}_t) - y_t)^2$  and use it as a proxy for  $L_T(f_i^*)$ . In our experiments,  $L_T(f_i^*) \approx 88.6$  on the *elevators* dataset and  $L_T(f_i^*) \approx 23.8$  on the *Hardware* dataset. Thus  $L_T(f_i^*) \ll T$  and  $O(\frac{1}{\sqrt{\mathcal{R}}}\sqrt{TL_T(f_i^*)})$  is actually smaller than  $O(\frac{T}{\sqrt{\mathcal{R}}})$ . The above results coincide with Theorem 4.

RF-IOKS shows similar performance with the baseline algorithms, which is consistent with Theorem 5. The regret bound of RF-IOKS is slightly worse than that of LKMBooks and Raker, and is only asymptotically better than RF-OKS. All of the baseline algorithms tune the stepsize in hindsight, which is impossible in practice since the data can only be predicted once. RF-IOKS also proves that selecting a hypothesis does not damage the regret bound much in the worst case. More experiments are shown in the supplementary materials.

## 7 Conclusion

In this paper, we have proposed two algorithms for online kernel selection under bandit feedback and improved the previous worst-case regret bound. OKS++ which is applied for smooth loss functions, adaptively tunes parameters of OKS and achieves data-dependent regret bounds depending on the minimal cumulative losses. IOKS which is applied for Lipschitz loss functions, achieves a worst-case regret bound asymptotically better than previous result. We further apply the two algorithms to online kernel selection with time constraint and obtain better or similar regret bounds.

From the perspective of algorithm design, there is a trade-off between regret bound and the amount of observed information. IOKS proves that selecting a hypothesis or multiple hypotheses per-round will not induce significant variation on the worst-case regret bound. OKS++ which performs well both in theory and practice, implies that there may be differences in terms of data-dependent regret bounds. This question is left to future work.

## References

1. Agarwal, A., Luo, H., Neyshabur, B., Schapire, R.E.: Corraling a band of bandit algorithms. In: Proceedings of the 30th Conference on Learning Theory, pp. 12–38 (2017)
2. Audibert, J., Bubeck, S.: Minimax policies for adversarial and stochastic bandits. In: Proceedings of the 22nd Annual Conference on Learning Theory. pp. 217–226 (2009)
3. Auer, P., Cesa-Bianchi, N., Freund, Y., Schapire, R.E.: The non-stochastic multi-armed bandit problem. *SIAM Journal on Computing* **32**(1), 48–77 (2002)
4. Cesa-Bianchi, N., Lugosi, G.: Prediction, Learning, and Games. Cambridge University Press (2006)
5. 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)

6. Ghari, P.M., Shen, Y.: Online multi-kernel learning with graph-structured feedback. In: Proceedings of the 37th International Conference on Machine Learning. pp. 3474–3483 (2020)
7. Hoi, S.C.H., Jin, R., Zhao, P., Yang, T.: Online multiple kernel classification. *Machine Learning* **90**(2), 289–316 (2013)
8. Le, Q.V., Sarlós, T., Smola, A.J.: Fastfood-computing hilbert space expansions in loglinear time. In: Proceedings of the 30th International Conference on Machine Learning. pp. 244–252 (2013)
9. Li, J., Liao, S.: Worst-case regret analysis of computationally budgeted online kernel selection. *Machine Learning* **111**(3), 937–976 (2022)
10. Li, Z., Ton, J.F., Oglie, D., Sejdinovic, D.: Towards a unified analysis of random Fourier features. In: Proceedings of the 36th International Conference on Machine Learning. pp. 3905–3914 (2019)
11. Liao, S., Li, J.: High-probability kernel alignment regret bounds for online kernel selection. In: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 67–83 (2021)
12. Rahimi, A., Recht, B.: Random Features for Large-Scale Kernel Machines. *Advances in Neural Information Processing Systems* **20**, 1177–1184 (2007)
13. Rahimi, A., Recht, B.: Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. *Advances in Neural Information Processing Systems* **21**, 1313–1320 (2008)
14. Sahoo, D., Hoi, S.C.H., Li, B.: Online multiple kernel regression. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD. pp. 293–302 (2014)
15. Seldin, Y., Bartlett, P.L., Crammer, K., Abbasi-Yadkori, Y.: Prediction with Limited Advice and Multiarmed Bandits with Paid Observations. In: Proceedings of the 31st International Conference on Machine Learning. pp. 280–287 (2014)
16. 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)
17. Tsallis, C.: Possible generalization of boltzmann-gibbs statistics. *Journal of Statistical Physics* **52**(1), 479–487 (1988)
18. Wang, J., Hoi, S.C.H., Zhao, P., Zhuang, J., Liu, Z.: Large scale online kernel classification. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence. pp. 1750–1756 (2013)
19. 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)
20. Yu, F.X., Suresh, A.T., Choromanski, K.M., Holtmann-Rice, D.N., Kumar, S.: Orthogonal random features. *Advances in Neural Information Processing Systems* **29**, 1975–1983 (2016)
21. 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)
22. 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)
23. Zimmert, J., Seldin, Y.: An optimal algorithm for stochastic and adversarial bandits. In: Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics. pp. 467–475 (2019)