Machine Learning (2022) 111:937–976
https://doi.org/10.1007/s10994-021-06082-8

Worst-case regret analysis of computationally budgeted online kernel selection

Junfan Li[1] · **Shizhong Liao**[1]

Received: 15 May 2021 / Revised: 14 August 2021 / Accepted: 22 September 2021 /
Published online: 22 January 2022

Abstract
We study the problem of online kernel selection under computational constraints, where the memory or time of kernel selection and online prediction procedures is restricted to a fixed budget. In this paper, we analyze the worst-case lower bounds on the regret of online kernel selection algorithm with a subset of the observed examples, and design algorithms enjoying corresponding upper bounds. We also identify the condition under which online kernel selection with time constraints is different from that with memory constraints. To design algorithms, we reduce the problems to two sequential decision problems, that is, the problem of prediction with expert advice and the multi-armed bandit problem with an additional observation. Our algorithms invent some new techniques, such as memory sharing, hypothesis space discretization and decoupled exploration-exploitation scheme. Numerical experiments on online regression and classification are conducted to verify our theoretical results.

Keywords Online learning · Kernel selection · Computational constraints · Regret analysis

1 Introduction

Kernel selection is a fundamental problem of online kernel learning, which focuses on how to select kernel functions for online kernel learning algorithms on the fly. This problem is also termed as online kernel selection, and related to the more general online model selection (Foster et al. 2017; Muthukumar et al. 2019). Different from offline kernel selection, where we first execute kernel selection on a training set and then learn a predictor for the subsequent prediction tasks, the kernel selection and online prediction procedures are integrated and form a sequential prediction procedure. Given a collection of kernel functions $\{\kappa_i\}_{i=1}^K$, which induce K reproducing kernel Hilbert spaces (RKHSs) $\{\mathcal{H}_i\}_{i=1}^K$, an adversary sequentially sends the learner an example $(\mathbf{x}_t, y_t) \in \mathbb{R}^d \times \mathbb{R}, t = 1, \ldots, T$. The learner will

Editors: Yu-Feng Li, Mehmet Gönen, Kee-Eung Kim.

✉ Shizhong Liao
szliao@tju.edu.cn

[1] College of Intelligence and Computing, Tianjin University, Tianjin 300350, China

choose a sequence of kernels $\{\kappa_{I_t}\}_{t=1}^T$ and a sequence of hypotheses $\{f_t\}_{t=1}^T$. At each round t, the learner suffers a loss $\ell(f_t(\mathbf{x}_t), y_t)$. General performance measurement is the regret. The regret with respect to (w.r.t.) $\mathcal{H}_i, i \in [K]$ is defined as follows

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

Since the best kernel function for the current learning task is unknown, the learner hopes to adapt to any $\mathcal{H}_i$ up to a small cost.

A major challenge of online kernel selection is the high computational complexity of evaluating kernel functions which requires to operate on the observed examples and thus incurs a $O(T)$ per-round time complexity and space complexity. We can solve this problem from two computational perspectives. The first computational perspective aims at reducing the computational complexity. Most of previous work followed this line. The random feature based online kernel selection approach (Nguyen et al. 2017) embedded the implicit RKHSs to relatively low-dimensional explicit feature spaces, in which the time and space complexity of evaluating kernel functions are linear with the dimension of random feature spaces. The sketch based online kernel selection approach (Zhang and Liao 2018, 2020) maintained a budget and incrementally constructed sketched hypothesis spaces, in which the time and space complexity are linear with the budget size. Another approach reduces online kernel selection to a problem of prediction with expert advice, and uses some master algorithm to wrap computationally efficient online kernel learning algorithms, including budgeted online kernel learning (Crammer et al. 2003; Dekel et al. 2008; Orabona et al. 2009; Koppel et al. 2019), low-rank matrix approximation based online kernel learning and projection to a low-dimension space (Lu et al. 2016; Jézéquel et al. 2019). For instance, Foster et al. (2017) studied online model selection in Banach space and developed a multi-scale expert advice algorithm, which can adapt to the loss range of different hypothesis set.

The second computational perspective limits the usable computational resources and is more practical for online learning problem. Previous work did not consider this new computational perspective, or only indirectly considered the memory constraints (Nguyen et al. 2017; Zhang and Liao 2018). Thus many fundamental problems induced by computational constraints have been omitted. The first fundamental problem is that how the regret depends on the computational constraints, T and K, where K is the number of candidate kernel functions. For instance, given a memory budge B, it is still unclear how the lower bound on the regret depends on B, T and K. The second problem is what the differences between memory constraints and time constraints are. The main obstacle induced by the computational constraints is how to avoid allocating the available computational resources over K RKHSs. Existing approaches allocate the computational resources, and thus may not be optimal.

In this paper, we study online kernel selection under computational constraints, where the kernel selection and online prediction procedures are restricted by a memory budget or a time budget of $\mathcal{T}$ quanta. We focus on the worst-case regret analysis[1] and solve the above two fundamental problems. To start with, we make mild assumptions that relate the memory budget and time budget to the example budget. Thus we only consider such online kernel selection approaches that operate on a subset of observed examples. For

[1] The worst-case regret is the regret that holds on any examples, also defined by $\max_{(\mathbf{x}_t, y_t)_{t=1,\ldots,T}} \mathrm{Reg}(\mathcal{H}_i)$. We aims at proving the lower bound on $\max_{(\mathbf{x}_t, y_t)_{t=1,\ldots,T}} \mathrm{Reg}(\mathcal{H}_i)$ defined on any algorithm, that is, $\min_{\{f_t\}_{t=1,\ldots,T}} \max_{(\mathbf{x}_t, y_t)_{t=1,\ldots,T}} \mathrm{Reg}(\mathcal{H}_i)$, and designing algorithms enjoying corresponding upper bound.

Table 1 Summary of main results

Constraint	Upper bound	Lower bound
Memory	$\sqrt{T \ln K} + (\|f_i^*\|_{\mathcal{H}_i}^2 + 1) \max\left\{\sqrt{T}, \frac{T}{\sqrt{\alpha T}}\right\}$	$\|f_i^*\|_{\mathcal{H}_i} \max\left\{\sqrt{T}, \frac{T}{\sqrt{\alpha T}}\right\}$
	$\text{Pen}_1 + \|f_i^*\|_{\mathcal{H}_i} \max\left\{\sqrt{T}, \frac{T}{\sqrt{\alpha T}}\right\} \sqrt{\ln(KT)}, K < \frac{d}{M}$	
Time	Equivalent to memory constraints, $K \leq d$	$\|f_i^*\|_{\mathcal{H}_i} \max\left\{\sqrt{T}, \frac{T}{\sqrt{\beta T}}\right\}$
	$\text{Pen}_{i,2} + (\|f_i^*\|_{\mathcal{H}_i}^2 + 1) \max\left\{\sqrt{TK}, \frac{T}{\sqrt{\beta T}}\right\}, K > d$	

$\text{Pen}_1 = \sqrt{T \ln(KT)}$, $M = \ln \sqrt{T}$ and $\text{Pen}_{i,2} = \sqrt{(U+1) L_T(f_i^*) K \ln K}$, where $L_T(f_i^*) = \min_{f \in \mathcal{H}_i} \sum_{t=1}^T \ell(f(\mathbf{x}_t), y_t), i = 1, \ldots, K$, and $U = \Theta(\sqrt{\beta T})$. α and β are two constants defined in Assumption 3

unconstrained RKHSs and convex loss functions, we separately prove a lower bound on the regret under a memory budget and time budget. Our proof technique is novelty, which relies on a sequence of equi-distant instances and does not require the orthogonality or approximate orthogonality in RKHSs. For online kernel selection with memory constraints, we reduce it to the problem of prediction with expert advice, and establish two nearly optimal algorithms with different regret bounds. The keys include a memory sharing and a hypothesis space discretization scheme. For online kernel selection with time constraints, we consider two cases. If $K \leq d$, the number of features, this problem is equivalent to the case of memory constraints. For the case of $K > d$, the two problems are different. We reduce it to the multi-armed bandit problem with an additional observation, and establish a nearly optimal algorithm. The key is a decoupled exploration-exploitation scheme. Table 1 gives a summary of the main results.

1.1 Related work

Online kernel learning with a memory budget has been studied for years (Crammer et al. 2003; Dekel et al. 2008; Orabona et al. 2009). The bounded online gradient descent algorithm (Zhao et al. 2012) enjoys a $O((\|f\|_{\mathcal{H}}^2 + 1)T/\sqrt{B})$ expected regret bound for the hinge loss. However, the matching lower bound is still unknown. Dekel et al. (2008) proved an incomplete hardness result. There exists a sequence of examples and a fixed hypothesis that makes no mistakes, but any online kernel learning algorithm with limited memory always makes mistakes. How the lower bound depends on the memory budget is still unclear. For smooth loss functions, Zhang et al. (2013) proved a $\Omega(T/B)$ lower bound on the regret in the case of $B = O(\sqrt{T})$. Cesa-Bianchi et al. (2015) studied the complexity of offline kernel learning with memory constraints, and proved several lower bounds on the optimization error, which is different from regret. Our work studies the lower bounds for online kernel selection with computational constraints and is suitable for online kernel learning.

Agarwal et al. (2011) initiated the study of computationally budgeted model selection, where the model selection procedure is restricted to a time budget. For a collection of finite number of model classes, by reducing the problems to a stochastic bandit problem, an upper-confidence bound algorithm was established, which can achieve the model selection oracle inequality. The algorithm is not suitable for online kernel selection, since the environments may not be i.i.d.. Our work is also related to online multiple kernel learning

(Jin et al. 2010; Hoi et al. 2013). Given K candidate RKHSs, at each round t, the goal is to learn a linear combination of K predictions. Sahoo et al. (2014) proposed budgeted online multi-kernel regression algorithms, which use a budget B to limit the number of support vectors. However, they did not prove how the regret upper bound depends on B. Besides, the per-round time complexity of such algorithms is linear with K. Within time constraints, such algorithms allocate the time resources to K RKHSs which would not be optimal. Our work revels how the upper bound depends on the computational constraints, T and K, and can make up the omitted regret analysis.

There are some other related work, including parameter-free online learning (McMahan and Abernethy 2013; McMahan and Orabona 2014; Cutkosky and Boahen 2016), and model selection for the multi-armed bandit problems (Agarwal et al. 2017; Foster et al. 2019), where the CORRAL algorithm (Agarwal et al. 2017) was proposed for selecting bandit algorithms on the fly. For our focused problems, the sub-algorithms are online kernel learning algorithms rather than bandit algorithms, thus CORRAL is not the best candidate. Parameter-free online learning aims at making regret bounds depend on $\|f\|_{\mathcal{H}}$ rather than $(\|f\|_{\mathcal{H}}^2 + 1)$. Previous work did not consider computational constraints. Our work can achieve this goal within memory constraints.

1.2 Contributions

We study online kernel selection in the regime of memory constraints or time constraints, and analyze the regret in the worst case. Our contributions can be summarized as follows.

- We prove the worst-case lower bounds on the regret of budgeted online kernel selection algorithm with memory constraints or time constraints. The lower bounds on the regret reveal the lower bounds on the computational constraints that are necessary for achieving a given upper bound on the regret. As a byproduct, our results are suitable for online kernel learning with memory constraints and make up the incomplete result established by Dekel et al. (2008).
- We identify the condition for the first time under which online kernel selection with time constraints is different from memory constraints.
- We separately propose nearly optimal algorithms for the two computational constraints which invent some new techniques, such as memory sharing, hypothesis space discretization and decoupled exploration-exploitation scheme.

2 Problem setup

Let $\mathcal{I}_T := \{(\mathbf{x}_t, y_t)\}_{t\in[T]}$ be a sequence of examples, where $\mathbf{x}_t \in \mathcal{X} \subset \mathbb{R}^d$ is an instance, $y_t \in [-Y, Y]$ is the output and $[T] = \{1, \ldots, T\}$. Let $\kappa(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a positive semidefinite kernel function, and $\mathcal{H}$ be the RKHS associated with κ, such that, for any $f \in \mathcal{H}$, (i) $\langle f, \kappa(\mathbf{x}, \cdot)\rangle_{\mathcal{H}} = f(\mathbf{x}), \forall \mathbf{x} \in \mathcal{X}$, and (ii) $\mathcal{H} = \overline{\text{span}(\kappa(\mathbf{x}, \cdot)|\mathbf{x} \in \mathcal{X})}$. We define $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ as the inner product in $\mathcal{H}$, which induces the norm $\|f\|_{\mathcal{H}} = \sqrt{\langle f, f\rangle_{\mathcal{H}}}$. Assuming the loss function $\ell : \mathbb{R} \times [-Y, Y] \to \mathbb{R}_+$ is convex in its first parameter.

Given a collection of kernel functions $\mathcal{K} = \{\kappa_i\}_{i=1}^K$, which induce K RKHSs $\mathcal{H} = \{\mathcal{H}_i\}_{i=1}^K$. If an oracle gives the best kernel κ^* for $\mathcal{I}_T$, then we just need to learn a sequence of hypotheses in $\mathcal{H}^*$. Lacking the prior of $\mathcal{H}^*$, the learner hopes to develop some 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 $\mathcal{H}^*$ solely. The regret of the algorithm w.r.t. $\mathcal{H}_i \in \mathcal{H}$ is defined in (1). For the sake of clarity, we restate it as follows,

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

To adapt to the unknown $\mathcal{H}^*$, a feasible approach is to keep sub-linear regret w.r.t. any $\mathcal{H}_i$.

To achieve this goal, the main challenge is the high time and space complexity. If we do not limit the model size, then the per-round time complexity and the space complexity would be $O(T)$. In this paper, we consider online kernel selection under computational constraints, including a memory budget or a time budget, and analyze the worst-case regret. Next, we define the two kinds of computational constraints.

Definition 1 (Memory Budget) Define a memory budget of $\mathcal{T}$ quanta as the maximal memory that any online kernel selection algorithm can use.

Definition 2 (Time Budget) Let the interval of arrival time between $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$, $t = 1, \ldots, T$ be less than $\mathcal{T}$ quanta. Define a time budget of $\mathcal{T}$ quanta as the maximal time interval that any online kernel selection algorithm outputs the prediction of $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$.

In Definition 1, the term "quanta" is the unit of memory, such as " Byte". In Definition 2, the term "quanta" is the unit of time, such as "millisecond" or "second". We further assume that the base kernels satisfy the following property.

Assumption 1 For all $\kappa_i \in \mathcal{K}$ and $\mathbf{u}, \mathbf{v} \in \mathcal{X}$, let $\kappa_i(\mathbf{u}, \mathbf{v})$ be a function of $\langle \mathbf{u}, \mathbf{v} \rangle$, $\|\mathbf{u}\|_2$ and $\|\mathbf{v}\|_2$, and $\kappa_i(\mathbf{u}, \mathbf{u}) \in [0, D_i]$.

Such kernels are also called Euclidean kernel (Kothari and Livni 2020). For simplicity, let $D := \max_i D_i$. Usual kernel functions, such as shift-invariant kernel and polynomial kernel with bounded degree, satisfy the assumption. We further give three key assumptions, which reduce the memory budget and time budget to example budget.

Assumption 2 Let the memory budget be linear with the space complexity of algorithm, and the time budget be linear with the time complexity of algorithm.

The space complexity is defined as the memory required by algorithm. Thus it is intuitive to assume that the memory budget is linear with the space complexity of algorithm. Similarly, assuming that m multiply operations can be executed within a unit time. For a given time budget of $\mathcal{T}$ quanta, the algorithm can execute $m\mathcal{T}$ multiply operations. The time complexity of algorithm is defined as the total number of multiply operations. Thus we can also assume that the time budget is linear with the time complexity.

Assumption 3 Under the condition of Assumptions 1 and 2, for any kernel $\kappa \in \mathcal{K}$, there exist positive integers α and β, such that any budgeted online kernel leaning algorithm running in $\mathcal{H}_\kappa$ can maintain a budget storing $B \leq \alpha\mathcal{T}$ examples within a memory budget of $\mathcal{T}$ quanta, or can execute $B \leq \beta\mathcal{T}$ kernel evaluations at each round within a time budget of $\mathcal{T}$ quanta. If the space complexity and time complexity of algorithm are linear with B, then "=" holds.

Assumption 4 Under the condition of Assumption 3, let the maximal memory budget $\mathcal{T}$ satisfy $B = T$, and the maximal time budget $\mathcal{T}$ satisfy $B = T$.

Assumption 4 means that there is no need to assume an infinite $\mathcal{T}$ unless T is infinite. The reason is that any algorithm can store T examples at most. In practice, $\mathcal{T}$ may be very small. In Assumption 3, the *budgeted online kernel learning algorithms* are such algorithms that operate on a subset of the observed examples, such as, Forgetron (Dekel et al. 2008), BOGD (Zhao et al. 2012), BSGD (Wang et al. 2012) to name but a few. We claim that α and β are independent of kernel function. It is reasonable, since the memory cost is used to store the support vectors and coefficient vectors, and the time cost of computing $\kappa(\mathbf{u}, \mathbf{v})$ is to compute $\langle \mathbf{u}, \mathbf{v} \rangle$, $\|\mathbf{u}\|_2$ and $\|\mathbf{v}\|_2$. We only focus on convex loss functions. Online gradient descent has the lowest space and time complexity, which is $O(dB)$, where B is the budget size. For algorithms whose time complexities are $O(dB^\gamma), \gamma > 1$, then "=" doest not hold in Assumption 3. Based on the above three assumptions, we only consider such online kernel selection algorithms that work in implicit RKHSs and operate on finite examples. For the sake of clarity, we denote such algorithms as *budgeted online kernel selection algorithms*.

Next we restate the main questions.

Q 1 How does the regret depend on $\mathcal{T}$, T and K in the worst case?
Q 2 What are the differences between memory constraints and time constraints?

To answer the two questions, we need to solve the following two problems, (i) proving the lower bounds on the regret under memory constraints or time constraints and, (ii) establishing algorithms achieving the lower bounds. Our main contributions are providing nearly complete answers to the questions.

3 Online kernel selection with memory constraints

In this section, we give both a lower bound on the regret for online kernel selection with a memory budget and two simple algorithms nearly achieving the lower bound.

3.1 Lower bound

We select K Gaussian kernel functions $\kappa_i(\mathbf{x}, \mathbf{z}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{z}\|_2^2}{\sigma_i}\right), i \in [K]$ as the candidates. Without loss of generality, let $0 < \sigma_1 < \ldots < \sigma_K$, where σ_K is a bounded constant. We can also create candidates from other kernel functions, such as polynomial kernels, or the mixture of polynomial kernels and Gaussian kernels.

Theorem 1 *Let $\ell(\cdot, \cdot)$ be the hinge loss or the absolute loss. There exist K kernel functions $\{\kappa_i\}_{i=1}^K$ selected by the learner, and a sequence of examples $\{(\mathbf{x}_t, y_t)\}_{t=1}^T$ selected by an oblivious adversary, where $y_t \in \{-1, 1\}$, such that, for a memory budget of $\mathcal{T}$ quanta, under the condition of Assumption 3, for all κ_i, the expected regret of any budgeted online kernel selection algorithm satisfies*

$$\mathbb{E}\left[\sum_{t=1}^{T}\ell(f_t(\mathbf{x}_t),y_t)\right]-\sum_{t=1}^{T}\ell(f_i^*(\mathbf{x}_t),y_t)=\begin{cases}\Omega\left(\left\|f_i^*\right\|_{\mathcal{H}_i}L\sqrt{T}\right) & \text{if } T=O(\alpha\mathcal{T}),\\ \Omega\left(\left\|f_i^*\right\|_{\mathcal{H}_i}L\frac{T}{\sqrt{\alpha\mathcal{T}}}\right) & \text{otherwise,}\end{cases}\tag{2}$$

where L is the Lipschitz constant of ℓ, and $f_i^*=\operatorname{argmin}_{f\in\mathcal{H}_i}\sum_{t=1}^{T}\ell(f(\mathbf{x}_t),y_t)$.

According to the lower bound, we can infer the relation between the upper bound on the regret and the lower bound on the required memory budget. In the case of $T=O(\alpha\mathcal{T})$, the optimal upper bound on the regret is $O\left(\left\|f_i^*\right\|_{\mathcal{H}_i}L\sqrt{T}\right)$. In the case of $T=\Omega(\alpha\mathcal{T})$, the optimal upper bound is $O\left(\left\|f_i^*\right\|_{\mathcal{H}_i}L\frac{T}{\sqrt{\alpha\mathcal{T}}}\right)$. Let $T(\alpha\mathcal{T})^{-\frac{1}{2}}\le CT^{\upsilon}$, $\frac{1}{2}\le\upsilon<1$, where C is a constant. Solving the inequality yields that the required lower bound on the memory budget satisfies $\mathcal{T}\ge C^{-2}\alpha^{-1}T^{2(1-\upsilon)}$. In the worst case, achieving a $O(T^{\upsilon})$, $\frac{1}{2}\le\upsilon<1$ regret bound requires a memory budget of order $\Omega(T^{2-2\upsilon})$. The lower bound on the regret seems surprising and may not be a strong result, since it is independent of K. We will show that it is optimal up to an additional penalty term.

If $K=1$, then Theorem 1 reveals the lower bound of budgeted online kernel learning algorithms. We can not provide a $O(\|f_1^*\|_{\mathcal{H}_1}L\sqrt{T})$ regret bound unless the memory budget $\mathcal{T}=\Omega(T/\alpha)$. The BOGD algorithm (Zhao et al. 2012) enjoys a $O((\|f_1^*\|_{\mathcal{H}_1}^2+1)LT/\sqrt{\alpha\mathcal{T}})$ expected regret bound which is optimal w.r.t. T, but sub-optimal w.r.t. $\|f_1^*\|_{\mathcal{H}_1}$. Dekel et al. (2008) proved an incomplete hardness result for online kernel learning under a memory budget B. There always exists $B+1$ examples, such that any algorithm only storing B examples will make $T=B+1$ mistakes. Besides, there is a hypothesis $f_1^*\in\mathcal{H}_1$ satisfying $\|f_1^*\|_{\mathcal{H}_1}=\sqrt{B+1}$ that never makes mistakes and attains a hinge loss of 0. Actually, their lower bound on the mistakes equals the lower bound on the regret for the hinge loss, or rather, the lower bound on the regret is $B+1=\|f_1^*\|_{\mathcal{H}_1}\sqrt{T}$, where we use the specific identity $T=B+1$. The weakness of this lower bound is that it can not be extended to the case $B=o(T)$. Our result in Theorem 1 provides a complete answer to the question.

3.2 A nearly optimal algorithm for any *K*

An intuitive approach is to allocate the memory budget to the K base kernels. According to the lower bound (2), such an approach will increase the regret by a factor of order $O(\sqrt{K})$. Recalling that any hypothesis $f_i\in\mathcal{H}_i$ can be represented by $f_i=\sum_{t=1}^{T}a_{t,i}\kappa_i(\mathbf{x}_t,\cdot)$. Thus the memory cost is used to store the support vectors $\{(\mathbf{x}_t,y_t)_{t=1}^{T}:a_{t,i}\neq0\}$, and the coefficients $\{(a_{t,i})_{t=1}^{T}:a_{t,i}\neq0\}$. According to this observation, we will present an algorithm that shares the support vectors and a coefficient vector among K different hypotheses $\{f_i\}_{i=1}^{K}$.

Instead of selecting kernels from a finite collection $\{\kappa_1,\ldots,\kappa_K\}$, we will select kernels from an infinite kernel space $\mathcal{K}$ defined as follows,

$$\mathcal{K}=\left\{\kappa=\sum_{i=1}^{K}p_i\kappa_i:\sum_{i=1}^{K}p_i=1,p_i\ge0\right\}.$$

The learning of the weight vector $\mathbf{p}$ will be clarified later. At the beginning of round t, assuming that there is a weight vector $\mathbf{p}_t$. We learn a new kernel $\kappa_{\mathbf{p}_t} = \sum_{i=1}^{K} p_{t,i}\kappa_i$, which induces a RKHS $\mathcal{H}_{\mathbf{p}_t}$ with embedding $\phi_{\mathbf{p}_t} : \mathcal{X} \to \mathcal{H}_{\mathbf{p}_t}$ defined as follows

$$\phi_{\mathbf{p}_t}(\mathbf{x}) = \left(\sqrt{p_{t,1}}\phi_{\kappa_1}^{\top}(\mathbf{x}), \ldots, \sqrt{p_{t,K}}\phi_{\kappa_K}^{\top}(\mathbf{x})\right)^{\top}, \forall \mathbf{x} \in \mathcal{X}, \tag{3}$$

where ϕ_{κ_i} is the embedding induced by κ_i. We select a hypothesis $f_t \in \mathcal{H}_{\mathbf{p}_t}$, defined by

$$\begin{aligned} f_t = \sum_{\tau=1}^{t-1} a_\tau \phi_{\mathbf{p}_t}(\mathbf{x}_\tau) &= \left(\sqrt{p_{t,1}}\sum_{\tau=1}^{t-1} a_\tau \phi_{\kappa_1}^{\top}(\mathbf{x}_\tau), \ldots, \sqrt{p_{t,K}}\sum_{\tau=1}^{t-1} a_\tau \phi_{\kappa_K}^{\top}(\mathbf{x}_\tau)\right)^{\top} \\ &= \left(\sqrt{p_{t,1}}f_{t,1}, \ldots, \sqrt{p_{t,K}}f_{t,K}\right). \end{aligned} \tag{4}$$

The prediction is given by $f_t(\mathbf{x}_t) = \langle f_t, \phi_{\mathbf{p}_t}(\mathbf{x}_t)\rangle_{\mathcal{H}_{\mathbf{p}_t}} = \sum_{i=1}^{K} p_{t,i} f_{t,i}(\mathbf{x}_t)$, or $\mathrm{sign}(f_t(\mathbf{x}_t))$ for classification. Although there are K hypotheses $\{f_{t,i}\}_{i=1}^{K}$, we just need to maintain a single set of support vectors and a single coefficient vector $(a_1, \ldots, a_{t-1})$.

To keep the memory constraints, we propose a simple example adding strategy. At any round t, let $\nabla_{f_t} := \ell'(f_t(\mathbf{x}_t), y_t)\phi_{\mathbf{p}_t}(\mathbf{x}_t)$ be the (sub-)gradient of $\ell(f_t(\mathbf{x}_t), y_t)$ w.r.t. f_t. We define a Bernoulli random variable $\rho_t \in \{0, 1\}$ satisfying

$$\mathbb{P}[\rho_t = 1] = \min\left\{1, \frac{C}{z_t}\right\} \cdot \mathbb{I}_{\nabla_{f_t} \neq 0}, \tag{5}$$

where $C > 0$ is a constant and $z_t > 0$ depends on t. The definition of C and z_t will be given in Theorem 2. Let S be a buffer storing the support vectors. We sample $\rho_t \sim \mathrm{Ber}(\mathbb{P}[\rho_t = 1], 1)$. If $\rho_t = 1$, then we update f_t and add the current example into the buffer, i.e., $S = S \cup (\mathbf{x}_t, y_t)$. Let $\tilde{\nabla}_{f_t}$ be an estimator of ∇_{f_t}, which is defined as follows,

$$\tilde{\nabla}_{f_t} := \frac{\nabla_{f_t}}{\mathbb{P}[\rho_t = 1]}\mathbb{I}_{\rho_t=1} = \tilde{\ell}'(f_t(\mathbf{x}_t), y_t)\phi_{\mathbf{p}_t}(\mathbf{x}_t), \quad \tilde{\ell}'(f_t(\mathbf{x}_t), y_t) := \frac{\ell'(f_t(\mathbf{x}_t), y_t)}{\mathbb{P}[\rho_t = 1]}\mathbb{I}_{\rho_t=1}.$$

We update the hypothesis by online gradient descent

$$f_{t+1} = f_t - \lambda\tilde{\ell}'(f_t(\mathbf{x}_t), y_t)\phi_{\mathbf{p}_t}(\mathbf{x}_t),$$

where λ is the learning rate (or stepsize) of gradient descent. According to (3) and the definition of f_t (4), the above updating can be rewritten by

$$f_{t+1,i} = f_{t,i} - \lambda\tilde{\ell}'(f_t(\mathbf{x}_t), y_t)\phi_{\kappa_i}(\mathbf{x}_t), \quad \forall i = 1, \ldots, K.$$

For simplicity, we define $\nabla_{t,i} := \ell'(f_t(\mathbf{x}_t), y_t)\phi_{\kappa_i}(\mathbf{x}_t)$.

To update $\mathbf{p}_t$, we reduce this problem to a problem of prediction with expert advice. Let $c_{t,i}$ be a criterion evaluating base κ_i, $i = 1, \ldots, K$, which serves as the loss of the i-th action.

$$c_{t,i} = \begin{cases} \frac{\ell'(f_t(\mathbf{x}_t), y_t)\left(f_{t,i}(\mathbf{x}_t) - \min_{j=1,\ldots,K} f_{t,j}(\mathbf{x}_t)\right)}{\max\{\ell_m, 1\}} & \text{if } \ell'(f_t(\mathbf{x}_t), y_t) > 0, \\ \frac{\ell'(f_t(\mathbf{x}_t), y_t)\left(f_{t,i}(\mathbf{x}_t) - \max_{j=1,\ldots,K} f_{t,j}(\mathbf{x}_t)\right)}{\max\{\ell_m, 1\}} & \text{otherwise}, \end{cases} \tag{6}$$

where $\ell_m = \max_t\{|\ell'(f_t(\mathbf{x}_t), y_t)| \cdot \max_{i,j}\left(f_{t,i}(\mathbf{x}_t) - f_{t,j}(\mathbf{x}_t)\right)\}$ and can be tuned by the doubling trick. Let $\mathcal{E}(K)$ be the exponential weights algorithm in (Cesa-Bianchi and Lugosi 2006) (see Sect. 4.2). Then $\mathbf{p}_{t+1} = (p_{t+1,1}, \ldots, p_{t+1,K})$ can be computed as follows,

$$p_{t+1,i} = \frac{p_{t,i}\exp(-\eta c_{t,i})}{\sum_{j=1}^{K} p_{t,j}\exp(-\eta c_{t,j})},$$

where η is the learning rate.

We name the algorithm LKMBooks (Learning Kernel for Memory BOunded Online Kernel Selection). The algorithm description is shown in Algorithm 1.

Algorithm 1 LKMBooks

Input: $\lambda, \eta, \ell_m, C, z_t, B$.
Initialization: $f_{1,i} = 0, p_{1,i} = \frac{1}{K}, i = 1, \ldots, K, S = \emptyset$.
for $t = 1, \ldots, T$ **do**
  Receive instance $\mathbf{x}_t$;
  Output prediction $f_t(\mathbf{x}_t) = \sum_{i=1}^{K} p_{t,i} f_{t,i}(\mathbf{x}_t)$ or $\text{sign}(f_t(\mathbf{x}_t))$;
  Receive the true output y_t and compute loss $\ell(f_t(\mathbf{x}_t), y_t)$;
  Compute $\mathbb{P}[\rho_t = 1]$ according to (5), and sample $\rho_t \sim \text{Ber}(\mathbb{P}[\rho_t = 1], 1)$;
  for $i = 1, \ldots, K$ **do**
    Update hypothesis $f_{t+1,i} = f_{t,i} - \lambda \frac{\nabla_{t,i}}{\mathbb{P}[\rho_t=1]} \mathbb{I}_{\rho_t=1}$;
    Compute criterion $c_{t,i}$ according to (6);
  end for
  Compute $\mathbf{p}_{t+1}$ by $p_{t+1,i} = \frac{p_{t,i}\exp(-\eta c_{t,i})}{\sum_{j=1}^{K} p_{t,j}\exp(-\eta c_{t,j})}$;
end for

Theorem 2 *Let* $E_t = \{\tau < t : \nabla_{f_\tau} \neq 0\}$, $B = \alpha\mathcal{T}$ *and* $C = B$. *Let* $z_t = (1-\upsilon)T^{1-\upsilon}(|E_t|+1)^{\upsilon}$, *where* $0 \leq \upsilon < 1$. *If there exists a* $\upsilon \in [0,1)$ *satisfying* $(1-\upsilon)T^{1-\upsilon} > B$, *then for any sequence* $\mathcal{I}_T$, *with probability at least* $1-\delta$, *LKMBooks guarantees that*

$$|S| \leq B + \frac{2}{3}\ln\frac{1}{\delta} + \sqrt{2B\ln\frac{1}{\delta}}.$$

Otherwise, $|S| \leq B$.

Theorem 2 shows that our algorithm will not excess the memory constraint in a high probability. z_t gives the probability that any support vector is added into the budget. It is worth noting that the key of z_t is the value of υ. If $\upsilon = 0$. then each support vector is added into the budget with a same probability. We can also use a non-uniform probability distribution, i.e., $\upsilon > 0$. In this case, the probability decreases with the increasing of support vectors. In experiments, we always set $\upsilon > 0$ and empirically find that the non-uniform probability distribution performs better. In theory, the two kinds of probability distributions are equivalent in the sense that they induce the same budget size and regret bounds.

Theorem 3 *Given a memory budget of* $\mathcal{T}$ *quanta*, *under the condition of Assumption 3*, *let* $B = \alpha\mathcal{T}$. *Assuming that* ℓ *satisfies* $|\ell'(f(\mathbf{x}), y)| \leq L$. *Let* $\mathcal{K} = \{\kappa_i\}_{i=1}^{K}$ *be a collection of kernel functions*, *and* $\eta = \sqrt{8\ln(K)/T}$. *If* $B < T$, *then let* $\lambda = \sqrt{(1+\upsilon)B}/(\sqrt{(1-\upsilon)DLT})$. *Otherwise*, *let* $\lambda = 1/(\sqrt{DTL})$. *For any* $\kappa_i \in \mathcal{K}$, *the expected regret of LKMBooks satisfies*

$$\mathbb{E}\big[\mathrm{Reg}(\mathcal{H}_i)\big] \le O\left(\max\{\ell_m, 1\}\sqrt{T\ln K} + (\|f_i^*\|_{\mathcal{H}_i}^2 + 1)L\max\left\{\sqrt{T}, \frac{T}{\sqrt{\alpha\mathcal{T}}}\right\}\right).$$

Remark 1 LKMBooks is similar with the online multi-kernel learning algorithm in (Jin et al. 2010) (Algorithm 5, denoted by DA-OMKL-O for simplicity), and the budgeted online multi-kernel regression algorithm in (Sahoo et al. 2014) (denoted by BOKMR for simplicity), since the three algorithms use a convex combination of K outputs $\{f_{t,i}(\mathbf{x}_t)\}_{i=1}^K$. The difference is that, DA-OMKL-O and BOKMR make $\{f_{t,i}\}_{i=1}^K$ possess different coefficient vectors. However, LKMBooks makes $\{f_{t,i}\}_{i=1}^K$ share a single coefficient vector. Besides, DA-OMKL-O does not limit the support vectors, and one of the two versions of BOKMR can also not share the support vectors. The space complexity of LKMBooks is $O(dB + K)$. The two versions of BOKMR suffer a $O(dB + KB + K)$ and $O(KBd)$ space complexity, respectively. For the case of $K \gg d$, LKMBooks suffers the lowest space complexity. What's more, BOKMR did not provide a regret bound.

We consider the optimality w.r.t. $\mathcal{T}, T$ and K. Compared with the lower bound (2), LKMBooks is optimal up to an additional penalty term of order $O(\max\{\ell_m, 1\}\sqrt{T\ln K})$, which comes from the intrinsic complexity of prediction with expert advice. The penalty term is a lower order term. Thus LKMBooks avoids the dependence on $O(\sqrt{K})$. However, LKMBooks depends on $(\|f_i^*\|_{\mathcal{H}}^2 + 1)$, which is much worse than $\|f_i^*\|_{\mathcal{H}}$. The reason is that LKMBooks uses online gradient descent (OGD) to update hypothesis. The standard regret bound of OGD depends on $(\|f_i^*\|_{\mathcal{H}}^2 + 1)$ (Orabona 2013). Using OGD aims at sharing a single coefficient vector. Next we show an optimal algorithm for the case of $K < d/\ln\sqrt{T}$.

3.3 Adapt to the norm of competitor for $K < d/\ln\sqrt{T}$

To adapt to $\|f_i^*\|_{\mathcal{H}}$, we propose a hypothesis space discretization scheme. For each κ_i, $i = 1, \ldots, K$, we define the feasible hypothesis space by $\mathbb{H}_i = \{f \in \mathcal{H}_i : \|f\|_{\mathcal{H}_i} \le U\}$. We discretize $(0, U]$ as follows

$$(0, U] = \left(0, \mathrm{e}^{\lceil \ln U_{\min}\rceil}\right] \bigcup_{j=\lceil \ln U_{\min}\rceil}^{\lceil \ln U\rceil - 1} \left(\mathrm{e}^j, \mathrm{e}^{j+1}\right]. \tag{7}$$

This technique is also known as the peeling technique. The key is the choice of U and $U_{\min}$, which depends on the memory budget $\mathcal{T}$ and will be determined later. For any $f \in \mathbb{H}_i$, there exists some j such that $\|f\|_{\mathcal{H}_i} \in (0, \mathrm{e}^{\lceil \ln U_{\min}\rceil}]$ or $(\mathrm{e}^j, \mathrm{e}^{j+1}]$. Let $M = \lceil \ln U\rceil - \lceil \ln U_{\min}\rceil + 1$. We construct $K' := KM$ nested hypothesis spaces

$$\mathbb{H}_{i,j} = \{f \in \mathcal{H}_i : \|f\|_{\mathcal{H}_i} \le U_j\}, \quad i = 1, \ldots, K, \quad j = 1, \ldots, M,$$

where $U_j = \mathrm{e}^{j+\lceil \ln U_{\min}\rceil - 1}$. Thus $\mathbb{H}_{i,1} \subset \ldots \subset \mathbb{H}_{i,M} \subset \mathcal{H}_i$. For the sake of clarity, we define two index functions $h : [K] \times [M] \to [K']$ and $h^* : [K'] \to [K] \times [M]$. Specifically, $h(i, j)$ maps (i, j) to the $h(i, j)$-th element in $[K']$. Similarly, $h^*(k)$ maps $k \in [K']$ to $(h^*(k)_1, h^*(k)_2)$, where $h^*(k)_1 = \lfloor (k-1)/M \rfloor + 1$ and $h^*(k)_2 = k - (h^*(k)_1 - 1)M$.

To share the support vectors, we use an oblivious example adding strategy. The term "oblivious" means that the strategy is independent of algorithms. At any round t, let $\rho_t \in \{0,1\}$ be a Bernoulli random variable satisfying

$$\mathbb{P}[\rho_t = 1] = \min\left\{1, \frac{C}{z_t}\right\}.$$

Let $\{f_{t,i,j}\}_{t=1}^{T}$ be a sequence of hypotheses in $\mathbb{H}_{i,j}$ and $\nabla_{t,i,j} =: \nabla_{f_{t,i,j}} \ell(f_{t,i,j}(\mathbf{x}_t), y_t)$ be the (sub-) gradient w.r.t. $f_{t,i,j}, i \in [K], j \in [M]$. At the end of round t, we sample $\rho_t \sim \mathrm{Ber}(\mathbb{P}[\rho_t = 1], 1)$. If $\rho_t = 1$, then we update the hypothesis $f_{t,i,j}$ and add the current example into the buffer, i.e., $S = S \cup (\mathbf{x}_t, y_t)$. Let $\tilde{\nabla}_{t,i,j}$ be an estimator of $\nabla_{t,i,j}$, which is defined as follows,

$$\tilde{\nabla}_{t,i,j} = \frac{\nabla_{t,i,j}}{\mathbb{P}[\rho_t = 1]} \mathbb{I}_{\rho_t = 1}.$$

We update the hypothesis by online gradient descent

$$\bar{f}_{t+1,i,j} = f_{t,i,j} - \lambda_{i,j} \tilde{\nabla}_{t,i,j}, \quad f_{t+1,i,j} = \arg\min_{f \in \mathbb{H}_{i,j}} \|f - \bar{f}_{t+1,i,j}\|_{\mathcal{H}_i}^2. \tag{8}$$

The projection of any $f \in \mathcal{H}_i$ onto $\mathbb{H}_{i,j}$ is defined by $g = \min\{1, \frac{U_j}{\|f\|_{\mathcal{H}_i}}\} f$.

Next we show the kernel selection procedure. Let $\mathcal{E}(K')$ be an algorithm for prediction with expert advice. We select a hypothesis space $\mathbb{H}_{h^*(I_t)_1, h^*(I_t)_2}$, where $I_t \sim \mathbf{p}_t$, and make prediction $\hat{y}_t = f_{t,h^*(I_t)_1,h^*(I_t)_2}(\mathbf{x}_t)$ or $\mathrm{sign}(\hat{y}_t)$. For each action $h(i,j) \in [K']$, let the criterion be $c_{t,h(i,j)} = \ell(f_{t,i,j}(\mathbf{x}_t), y_t)$. For all $f \in \mathbb{H}_{i,j}$, assuming that there is a function $g(U_j, D_i, Y)$ satisfying $c_{t,h(i,j)} \le g(U_j, D_i, Y)$. At the end of round t, we send $\mathbf{c}_t = (c_{t,1}, \ldots, c_{t,K'})$ to $\mathcal{E}(K')$. To adapt to the norm of competitor, $\mathcal{E}(K')$ needs to achieve a multi-scale regret bound. Let $\mathcal{E}(K')$ be the MSMW algorithm in Bubeck et al. (2019). which is shown in Algorithm 3.

We name this algorithm PFMBooks (Parameter-Free for Memory BOunded Online Kernel Selection).

Algorithm 2 PFMBooks

Input: $C, z_t, B, U, \eta, \lambda_{i,j}, i \in [K], j \in [M], A_{\min} = \{k_{\min} \in [K'], k_{\min} = \mathrm{argmin}_{k \in [K']} g(U_{h^*(k)_2}, D_{h^*(k)_1}, Y)\}$.

Initialization: $S = \emptyset$, $p_{1,k} = (1 - \frac{1}{U\sqrt{T}}) \frac{1}{|A_{\min}|} + \frac{1}{K'U\sqrt{T}}$ for $k \in A_{\min}$, and $p_{t,k} = \frac{1}{K'U\sqrt{T}}$ for all $k \notin A_{\min}$.

for $t = 1, \ldots, T$ **do**
  Receive instance $\mathbf{x}_t$;
  Select a hypothesis space $\mathbb{H}_{h^*(I_t)_1, h^*(I_t)_2}$, where $I_t \sim \mathbf{p}_t$;
  Output prediction $\hat{y}_t = f_{t,h^*(I_t)_1,h^*(I_t)_2}(\mathbf{x}_t)$;
  Receive the true label y_t and compute loss $\ell(\hat{y}_t, y_t)$;
  Sample $\rho_t \sim \mathrm{Ber}(\mathbb{P}[\rho_t = 1], 1)$;
  for $i = 1, \ldots, K$ **do**
    for $j = 1, \ldots, M$ **do**
      Compute gradient estimator $\tilde{\nabla}_{t,i,j} = \frac{\nabla_{t,i,j}}{\mathbb{P}[\rho_t = 1]} \mathbb{I}_{\rho_t = 1}$;
      Update hypothesis $f_{t+1,i,j} = \mathrm{Proj}_{\mathbb{H}_{i,j}}(f_{t,i,j} - \lambda_{i,j} \tilde{\nabla}_{t,i,j})$;
      Compute loss $c_{t,h(i,j)} = \ell(f_{t,i,j}(\mathbf{x}_t), y_t)$;
    end for
  end for
  Send $c_t = (c_{t,1}, \ldots, c_{t,K'})$ and $\{g(U_j, D_i, Y)\}_{i \in [K], j \in [M]}$ to $\mathscr{E}(K')$, and receive $\mathbf{p}_{t+1}$;
end for

Algorithm 3 $\mathscr{E}(K')$

Input: $\{g(U_j, D_i, Y)\}_{i\in[K], j\in[M]}$, $\{c_{t,k}\}_{k\in[K']}$, $A_{\min} = \{k_{\min} \in [K'], k_{\min} = \operatorname{argmin}_{k\in[K']} g(U_{h^*(k)_2}, D_{h^*(k)_1}, Y)\}$
Initialization: $p_{1,k} = (1 - \frac{1}{U\sqrt{T}})\frac{1}{|A_{\min}|} + \frac{1}{K'U\sqrt{T}}$ for $k \in A_{\min}$, and $p_{t,k} = \frac{1}{K'U\sqrt{T}}$ for all $k \notin A_{\min}$;
1: $\forall k \in [K']$, $\bar{p}_{t+1,k} = p_{t,k} \exp\left(-\eta c_{t,k}/g(U_{h^*(k)_1}, D_{h^*(k)_2})\right)$;
2: Find λ^* s.t. $\sum_{k=1}^{K'} \bar{p}_{t+1,k} \exp\left(-\lambda^*/g(U_{h^*(k)_2}, D_{h^*(k)_1})\right) = 1$ by binary search;
3: Return $\forall k \in [K']$, $p_{t+1,k} = \bar{p}_{t+1,k} \exp\left(-\lambda^*/g(U_{h^*(k)_2}, D_{h^*(k)_1})\right)$;

Theorem 4 *Let $B = \alpha\mathcal{T}$, $C = B$ and $z_t = 2(1-\upsilon)T^{1-\upsilon}t^{\upsilon}$, where $0 \le \upsilon < 1$. Under the condition of Assumption 4, there exists a $\upsilon \in [0, 1)$ such that $2(1-\upsilon)T^{1-\upsilon} > B$. For any sequence $\mathcal{I}_T$, with probability at least $1-\delta$, PFMBooks guarantees that*

$$|S| \le \frac{B}{2} + \frac{2}{3}\ln\frac{1}{\delta} + \sqrt{B\ln\frac{1}{\delta}}.$$

The proof is same with that of Theorem 2. PFMBooks ensures $|S| = O(B/2)$ with a high probability and maintains KM coefficient vectors. The total space complexity is $O(\frac{dB}{2} + \frac{BKM}{2}) = O(dB) = O(d\alpha\mathcal{T})$ in the case of $K < d/M$. We will set $U_{\min} = U/\sqrt{T}$ in Theorem 6, and thus $M < 1 + \ln\sqrt{T}$. PFMBooks will not exceed the total memory constraints in a high-probability. Next we state an important assumption, which is easily satisfied and forms the bases of obtaining the final regret bound.

Assumption 5 For any sequence of examples $\mathcal{I}_T := \{(\mathbf{x}_t, y_t)\}_{t\in[T]}$, let $|y_t| \le Y$. For any hypothesis $f \in \mathcal{H}_i, i = 1, \ldots, K$ and $(\mathbf{x}, y) \in \mathcal{I}_T$, there always exists a function $g(\|f\|_{\mathcal{H}_i}, D_i, Y) : \mathbb{R}^3 \to \mathbb{R}$ such that $\ell(f(\mathbf{x}), y) \le g(\|f\|_{\mathcal{H}_i}, D_i, Y)$ and $g(\|f\|_{\mathcal{H}_i}, D_i, Y) = \Theta(1 + \|f\|_{\mathcal{H}_i})$.

Many loss functions satisfy Assumption 5, such as the ε-insensitive hinge loss, and the ε-insensitive absolute loss. For instance, if $\ell(f(\mathbf{x}), y) = |f(\mathbf{x}) - y|$, then we can define $g(\|f\|_{\mathcal{H}_i}, D_i, Y) = \|f\|_{\mathcal{H}_i}\sqrt{D_i} + Y$. If $\ell(f(\mathbf{x}), y) = \max\{0, 1 - yf(\mathbf{x})\}$, then we can define $g(\|f\|_{\mathcal{H}_i}, D_i, Y) = 1 + Y\|f\|_{\mathcal{H}_i}\sqrt{D_i}$. Next we show the multi-scale regret bound of $\mathcal{E}(K')$.

Theorem 5 *Let $\eta = \sqrt{2\ln(K'T)/T}$ and $U = \Theta(B)$. Under the condition of Assumption 5, $\forall k \in [K']$, the expected regret of $\mathcal{E}(K')$ satisfies*

$$\sum_{t=1}^{T}\langle c_t, \mathbf{p}_t\rangle - \sum_{t=1}^{T} c_{t,k} = O\Big(g(U_{h^*(k)_2}, D_{h^*(k)_1})\sqrt{T\ln(K'T)}\Big).$$

Remark 2 $\mathcal{E}(K')$ is slightly different from the original MSMW algorithm in Bubeck et al. (2019), including: (i) MSMW uses "reward" as the feedback, but $\mathcal{E}(K')$ uses "loss" as the feedback; (ii) the initial distribution of MSMW and $\mathcal{E}(K')$ are different. Although we can transform "loss" to "reward" by $r_{t,k} = g(U_{h^*(k)_2}, D_{h^*(k)_1}, Y) - c_{t,k}$, where $r_{t,k}$ is the reward of the k-th action, the regret bound will increase a term $\sum_{t=1}^{T}[\sum_{k=1}^{k} p_{t,k}g(U_{h^*(k)_2}, D_{h^*(k)_1}, Y) - g(U_{h^*(k)_2}, D_{h^*(k)_1}, Y)]$, which can not adapt to the scale of individual action. Thus we need a different proof. We present a simpler proof in the Appendix. One of the key is using a different initial distribution.

Theorem 6 *Given a memory budget of $\mathcal{T}$ quanta, under the condition of Assumption 3, let $B = \alpha\mathcal{T}$. Let $U = \Theta(\sqrt{B})$, $U_{\min} = U/\sqrt{T}$ and $\lambda_{i,j} = \frac{U_j\sqrt{(1+\upsilon)B}}{\sqrt{2(1-\upsilon)D_i}LT}$. The expected regret of PFMBooks w.r.t. any $\mathcal{H}_i, i = 1, \ldots, K$ satisfies*

$$\mathbb{E}\big[\mathrm{Reg}(\mathcal{H}_i)\big] = O\left(\|f_i^*\|_{\mathcal{H}_i} L \max\left\{\sqrt{T\ln(K'T)}, \frac{T}{\sqrt{\alpha\mathcal{T}}}\right\}\sqrt{\ln(KT)} + \sqrt{T\ln(K'T)}\right).$$

Remark 3 In Theorem 1, the lower bound does not limit $\|f_i^*\|_{\mathcal{H}_i}$. Our upper bound may be invalid if $U < \|f_i^*\|_{\mathcal{H}_i}$. Inspecting the hard examples in the proof of Theorem 1, we find that $\|f_i^*\|_{\mathcal{H}_i} = \Theta(\sqrt{B})$. Thus our upper bound is still valid if $U = \Theta(\sqrt{B})$.

The expectation is w.r.t. the randomness of $\mathcal{E}(K')$ and the randomness of $\{\rho_t\}_{t=1}^{T-1}$. Compared with the upper bound in Theorem 3, PFMBooks improves the dependence on $\|f_i^*\|_{\mathcal{H}_i}$. Compared with the lower bound (2), PFMBooks is optimal up to a factor of order $O(\sqrt{\ln(K'T)})$ and a small penalty term of order $O\Big(\sqrt{T\ln(K'T)}\Big)$.

4 Online kernel selection with time constraints

In this section, we give both a lower bound on the regret for online kernel selection with a time budget and a simple algorithm nearly achieving the lower bound.

4.1 Lower bound

For the sake of clarity, we introduce a natation of resource allocation. Any kernel selection algorithm needs to assign a kernel selection strategy and a resource allocation strategy simultaneously. In this work, we consider the static resource allocation defined as follows.

Definition 3 (Static Resource Allocation) Define a static resource allocation $R(\mathcal{T}_1, \ldots, \mathcal{T}_K)$ as a strategy that allocates a time budget of $0 < \mathcal{T}_i \le \mathcal{T}$ quanta to kernel function κ_i before the game, and does not change later.

For any budgeted kernel selection algorithm with static resource allocation $R(\mathcal{T}_1, \ldots, \mathcal{T}_K)$, the following theorem gives a lower bound on the regret.

Theorem 7 *Let $\ell(\cdot,\cdot)$ be the hinge loss or the absolute loss. There exist K kernel functions $\{\kappa_i\}_{i=1}^K$ chosen by the learner, and a sequence of examples $\{(\mathbf{x}_t, y_t)\}_{t=1}^T$ chosen by an oblivious adversary, where $y_t \in \{-1, 1\}$, such that for a time budget of $\mathcal{T}$ quanta, under the condition of Assumption 3, for all κ_i, the expected regret of any budgeted online kernel selection algorithm with static resource allocation $R(\mathcal{T}_1, \ldots, \mathcal{T}_K)$ satisfies*

$$\mathbb{E}[L_T(f_t)] - L_T(f_i^*) = \begin{cases} \Omega\left(\left\|f_i^*\right\|_{\mathcal{H}_i} L\sqrt{T}\right) & \text{if} \quad T = O(\beta \max_{j\in[K]} \mathcal{T}_j), \\ \Omega\left(\left\|f_i^*\right\|_{\mathcal{H}_i} L\frac{T}{\sqrt{\beta \max_{j\in[K]} \mathcal{T}_j}}\right) & \text{otherwise}, \end{cases} \tag{9}$$

where L is the Lipschitz constant of ℓ, and $f_i^* \in \mathcal{H}_i = \overline{\mathrm{span}(\kappa_i(\mathbf{x}_1, \cdot), \ldots, \kappa_i(\mathbf{x}_t, \cdot))}$.

The lower bound also reveals that, in the worst case, achieving a $O(T^v)$, $\frac{1}{2} \leq v < 1$ regret bound requires a time budget of order $\Omega(T^{2-2v})$. To design algorithms achieving the lower bound (9), it is necessary to adopt the $R(\mathcal{T}, \ldots, \mathcal{T})$ resource allocation.

We first highlight the difference between memory constraints and time constraints. Recalling that the space complexity of LKMBooks is $O(dB + K)$. The time complexity of LKMBooks is $O(dB + KB + K)$, but not $O(KdB + K)$. The reason is that, under Assumption 1, the main time cost of computing $\kappa_i(\mathbf{x}_t, \mathbf{x}_\tau)$ for all $\mathbf{x}_\tau \in S$ is to compute the norm $\|\mathbf{x}_t - \mathbf{x}_\tau\|_2$ or the inner product $\langle \mathbf{x}_t, \mathbf{x}_\tau \rangle$. Since LKMBooks only maintains a single S, we can first compute the norm or inner between $\mathbf{x}_t$ and the support vectors in S. Thus the time complexity of computing $f_{t,i}(\mathbf{x}_t)$ for all $i = 1, \ldots, K$, is of order $O(dB + KB)$. If $K \leq d$, the two constraints are equivalent and LKMBooks can also be a nearly optimal algorithm for the case of time constraints. Thing is different for the case of $K > d$. Assuming that $K = d^\nu, \nu > 1$. If an algorithm achieves the lower bound (9), then it would adopt the $R(\mathcal{T}, \ldots, \mathcal{T})$ resource allocation. Let the available budget of such an algorithm be B_1, and B_2 be the available budget of LKMBooks. According to Assumptions 3, we have the two identities $dB_1 = \mathcal{T}$ and $(d + K)B_2 = \mathcal{T}$, which imply $B_2 = O(K^{\frac{1-\nu}{\nu}} B_1)$. Substituting into Theorem 3, LKMBooks will increase the regret by a factor of order $O(K^{\frac{\nu-1}{2\nu}})$.

Thus for the case of $K < d$, we can directly use LKMBooks or PFMBooks. Next we propose a nearly optimal algorithm for the case of $K > d$. The algorithm adapts the $R(\mathcal{T}/2, \ldots, \mathcal{T}/2)$ resource allocation.

4.2 A nearly optimal algorithm for $K > d$

A simply observation is that we need not to evaluate all of the base kernels at each round. An intuitive approach is to select a single kernel function, κ_{I_t}, and use the hypothesis f_{t,I_t} to make prediction. Such an approach has been adopted in (Yang et al. 2012), where the kernel selection problem is reduced to a K-armed bandit problem. However, the regret bound is far from optimal for online kernel selection. At each round, the approach constructs estimated gradient $\tilde{\nabla}_{t,i} = \nabla_{t,i}/p_{t,i}$. The second moment is of order $\max_t \nabla_{t,i}/p_{t,i}$, which may be a large term. To address this issue, we will propose a simple exploration-exploitation scheme.

For each κ_i, we define the feasible hypothesis space by $\mathbb{H}_i = \{f \in \mathcal{H}_i : \|f\|_{\mathcal{H}_i} \leq U\}$. We slightly modify Algorithm 1. The key difference is that we randomly evaluate two kernel functions at each round. The two kernel functions are selected by a decoupled exploration-exploitation scheme, which is defined as follows

- Exploitation: select a kernel function $\kappa_{I_t} \sim \mathbf{p}_t$,
- Exploration: select another kernel function $\kappa_{J_t} \sim \mathcal{K}$ uniformly.

Note that it is possible that $\kappa_{I_t} = \kappa_{J_t}$. The exploration procedure makes each kernel be selected with a high probability.

Let $S_i, i = 1, \ldots, K$ be K buffers storing the support vectors. At each round t, we output the prediction $\hat{y}_t = f_{t,I_t}(\mathbf{x}_t)$ or $\text{sign}(\hat{y}_t)$. However, we do not update f_{t,I_t} unless $I_t = J_t$. The goal is to make $(\mathbf{x}_t, y_t)$ be added into each S_i with equal probability. After receiving y_t, we compute the gradient $\nabla_{f_{t,J_t}} \ell(f_{t,J_t}(\mathbf{x}_t), y_t)$. If $\nabla_{f_{t,J_t}} \ell(f_{t,J_t}(\mathbf{x}_t), y_t) \neq 0$, then we decide whether to update f_{t,J_t}. Let $\rho_{t,i} \in \{0, 1\}$ be a Bernoulli random variable satisfying

$$\mathbb{P}[\rho_{t,i}=1]=\min\left\{1,\frac{C}{z_{t,i}}\right\}\cdot\mathbb{I}_{\nabla_{t,i}\neq 0},\quad i=1,\ldots,K,$$

If $\rho_{t,J_t}=1$, then we update f_{t,J_t} and add the current example into the budget, i.e., $S_{J_t}=S_{J_t}\cup(\mathbf{x}_t,y_t)$. Let $\tilde{\nabla}_{t,i}$ be an estimator of $\nabla_{t,i}$, defined as follows,

$$\tilde{\nabla}_{t,i}=\frac{\nabla_{t,i}}{\mathbb{P}[i=J_t]\cdot\mathbb{P}[\rho_{t,i}=1]}\mathbb{I}_{i=J_t}\mathbb{I}_{\rho_{t,i}=1}.$$

We update the hypothesis $f_{t,i}$ follows (8), where the projection can be computed incrementally in time $O(1)$.

To update $\mathbf{p}_t$, we define a K-armed adversarial bandit problem with an additional observation in which the algorithm may obtain two losses. $\forall i\in[K]$, let $c_{t,i}=\ell(f_{t,i}(\mathbf{x}_t),y_t)/\ell_m$, where $\ell_m=\max_{t,i}\{\ell(f_{t,i}(\mathbf{x}_t),y_t)\}$ is a normalizing constant and can be tuned by the doubling trick. The key is the estimated loss $\tilde{c}_{t,i}$ defined as follows,

$$\tilde{c}_{t,i}=\frac{c_{t,i}}{\mathbb{P}[i\in\{I_t,J_t\}]}\mathbb{I}_{i\in\{I_t,J_t\}},\quad \mathbb{P}[i\in\{I_t,J_t\}]=\frac{K-1}{K}p_{t,i}+\frac{1}{K}. \tag{10}$$

We update $\mathbf{p}_t$ by online stochastic mirror descent (OSMD) with the negative entropy regularizer (Bubeck and Cesa-Bianchi 2012),

$$p_{t+1}=\arg\min_{\mathbf{p}\in\Delta_{K-1}}\left\{\langle\mathbf{p},\tilde{c}_t\rangle+\mathcal{D}_{\psi_t}(\mathbf{p},\mathbf{p}_t)\right\}, \tag{11}$$

where $\psi_t(\mathbf{p})=\sum_{i=1}^K\eta_t p_i\ln p_t$ and $\mathcal{D}_{\psi_t}$ is Bregman divergence.

We name the algorithm BATBooks (Bandit with Additional observation for Time BOunded Online Kernel Selection). The algorithm description is shown in Algorithm 4.

Algorithm 4 BATBooks

Input: $C, z_{t,i}, i\in[K], B, U, \eta, \lambda_i, i\in[K], \ell_m$.
Initialization: $S_i=\emptyset, f_{1,i}=0, i=1,\ldots,K$
for $t=1,2,\ldots,T$ **do**
  Receive instance $\mathbf{x}_t$;
  Select a kernel function κ_{I_t}, where $I_t\sim\mathbf{p}_t$;
  Output prediction $\hat{y}_t=f_{t,I_t}(\mathbf{x}_t)$;
  Receive the true label y_t and compute loss $c_{t,I_t}=\ell(\hat{y}_t,y_t)/\ell_m$;
  Explore another kernel function κ_{J_t} uniformly;
  Compute $c_{t,J_t}=\ell(f_{t,J_t}(\mathbf{x}_t),y_t)/\ell_m$;
  if $\nabla_{t,J_t}\neq 0$ **then**
    Sample $\rho_{t,J_t}\sim\mathrm{Ber}(\mathbb{P}[\rho_{t,J_t}=1],1)$;
    Construct $\tilde{\nabla}_{t,J_t}=\frac{\nabla_{t,J_t}}{\mathbb{P}[i=J_t]\cdot\mathbb{P}[\rho_{t,J_t}=1]}\mathbb{I}_{\rho_{t,J_t}=1}$;
    Update hypothesis $f_{t+1,J_t}=\mathrm{Proj}_{\mathbb{H}_{J_t}}(f_{t,J_t}-\lambda_{J_t}\tilde{\nabla}_{t,J_t})$;
  end if
  for $i=1,\ldots,K$ **do**
    Compute estimated criterion $\tilde{c}_{t,i}=\frac{c_{t,i}}{\mathbb{P}[i\in\{I_t,J_t\}]}\mathbb{I}_{i\in\{I_t,J_t\}}$;
  end for
  Compute $\mathbf{p}_{t+1}$ by $p_{t+1,i}=\frac{p_{t,i}\exp(-\eta\tilde{c}_{t,i})}{\sum_{j=1}^K p_{t,j}\exp(-\eta\tilde{c}_{t,j})}$;
end for

Theorem 8 *Let* $B = \beta\mathcal{T}$, $C = KB$ *and* $z_{t,i} = 2(1-\upsilon)T^{1-\upsilon}t^{\upsilon}$, *where* $0 \le \upsilon < 1$. *For any sequence* $\mathcal{I}_T$, *with probability at least* $1-\delta$, *BATBooks guarantees that*

$$|S_i| \le \frac{B}{2} + \frac{2}{3}\ln\frac{K}{\delta} + \sqrt{B\ln\frac{K}{\delta}}.$$

For all $i = 1, \ldots, K$, we have $|S_i| = O(B/2)$. BATBooks evaluates two hypotheses at each round. The total time complexity is $O(dB) = O(d\beta\mathcal{T})$. Thus BATBooks will not excess the total time budget in a high-probability.

Theorem 9 *Let* $c_t \in [0,1]^K$ *be any loss vector, and* $\tilde{C}_{T,*} = \min_{i\in[K]}\sum_{t=1}^{T}\tilde{c}_{t,i}$, *where* $\tilde{c}_{t,i}$ *is the estimator of* $c_{t,i}$ *defined in* (10). *Let* $\eta = \min\{\sqrt{2\ln K/(K\tilde{C}_{T,*})}, \frac{1}{K}\}$. *BATBooks guarantees*

$$\mathbb{E}\left[\sum_{t=1}^{T}[\langle \mathbf{p}_t, c_t\rangle - c_{t,i}]\right] \le 2\sqrt{2\mathbb{E}\left[\sum_{t=1}^{T}c_{t,i}\right]K\ln K}.$$

We can obtain an expected small-loss regret bound for bandit with an additional observation, which may be of independent interest. Seldin et al. (2014) proved the worst-case expected regret bound for this problem. Thus we improve the previous result. Note that if $\{c_t\}_{t=1}^{T}$ are fixed loss vectors, then we can remove the expectation operation.

Theorem 10 *Given a time budget of* $\mathcal{T}$ *quanta, under the condition of Assumption* 3, *let* $B =: \beta\mathcal{T}$. *Let* $U = \Theta(\sqrt{B})$ *and* ℓ *satisfy* $|\ell'(f(\mathbf{x}), y)| \le L$. *If there exists a* $\upsilon \in [0,1)$ *satisfying*

$$2(1-\upsilon)T^{1-\upsilon} > KB, \tag{12}$$

then for any $\mathbb{H}_i, i \in [K]$, *let* $\lambda_i = \frac{\sqrt{(1+\upsilon)B}}{\sqrt{2(1-\upsilon)D_iLT}}$, *the expected regret of BATBooks satisfies*,

$$\mathbb{E}\left[\mathrm{Reg}(\mathbb{H}_i)\right] = O\left(\sqrt{(U+1)L_T(f_i^*)K\ln K} + (\|f_i^*\|_{\mathcal{H}_i}^2 + 1)L\sqrt{D_i}\frac{T}{\sqrt{\beta\mathcal{T}}}\right).$$

If condition (12) *can not be satisfied, then let* $\lambda_i = \frac{1}{\sqrt{KD_iTL}}$. *The expected regret satisfies*,

$$\mathbb{E}\left[\mathrm{Reg}(\mathbb{H}_i)\right] = O\left(\sqrt{(U+1)L_T(f_i^*)K\ln K} + (\|f_i^*\|_{\mathcal{H}_i}^2 + 1)L\sqrt{D_iTK}\right).$$

Remark 4 We show for the first time, that online kernel selection with time constraints is different from memory constraints only in the case of $K > d$, which answers our second question, **Q** 2. Thus for the case of $K \le d$, we can just use Algorithm 1 or Algorithm 2. All of previous work does not find such a condition. The online multi-kernel learning algorithms in (Hoi et al. 2013; Sahoo et al. 2014) and the online kernel selection algorithm in

(Yang et al. 2012) randomly update a hypothesis for reducing time complexity. We prove that such an approach is unnecessary unless $K > d$.

We analyze the optimality w.r.t. $\mathcal{T}$, T and K. First we consider a small time budget, i.e., $B < 2T/K$ (condition (12) is satisfied). Compared with the lower bound (9), BATBooks has an additional cost of order $O(\sqrt{UL_T(f_i^*)K \ln K})$. Then we consider a large time budget, i.e, $2T/K \leq B \leq T$ (condition (12) is not satisfied). BATBooks is sub-optimal by a multiplicative factor of order $O(\sqrt{K})$ and the same additional cost. Although $U = \Theta(\sqrt{B})$, we have $L_T(f_i^*) = 0$ for the hard examples in the proof of Theorem 7. In this case, our upper bounds are nearly optimal w.r.t. T, K and $\mathcal{T}$.

Next we consider the the dependence on $\|f_i^*\|_{\mathcal{H}_i}$. Note that $L_T(f_i^*)$ and U could not be large simultaneously. If $L_T(f_i^*)$ is much large, then $\|f_i^*\|_{\mathcal{H}_i}$ would be small, and we can ensure U being small. Using Assumption 5, we have $L_T(f_i^*) = O(\|f_i^*\|_{\mathcal{H}_i} T)$. Thus the additional cost would be $O(\sqrt{U\|f_i^*\|_{\mathcal{H}_i} TK \ln K})$. Our bounds depend on $O(\sqrt{U\|f_i^*\|_{\mathcal{H}_i}})$ and $O(\|f_i^*\|_{\mathcal{H}_i}^2)$, which are worse than the lower bound in Theorem 7. Improving the dependence on $\|f_i^*\|_{\mathcal{H}_i}$ is left to further work.

5 Experiments

In this section, we conduct numerical experiments to verify our theoretical results. As a whole, our goal is to verify the following results,

(**G** 1) Online kernel selection improves the learning performance relative to online single kernel learning with an empirical preset kernel.
(**G** 2) The superior of memory sharing scheme. Within a same memory constraint, our algorithms are better than such algorithms that do not share the memory.
(**G** 3) In the worst case, the time constraints is same with the memory constraints for the case of $K < d$. Thus Algorithm 1 is also nearly optimal for online kernel selection with time constraints.
(**G** 4) In the worst case, the time constraints is different from the memory constraints for the case of $K \geq d$, that is, Algorithm 4 is better than Algorithm 1 for the case of $K > d$.

We first state the experimental setting, and then show the experimental results for online kernel selection with memory constraints and time constraints, respectively.

5.1 Experimental setting

We compare our algorithms with the following baseline algorithms,

- NORMA (Budgeted online kernel learning algorithm) (Kivinen et al. 2004)
- BOGD (Budget online kernel learning algorithm) (Zhao et al. 2012)
- OKS (Online Kernel Selection) (Yang et al. 2012)
- OMKC (Online multi-kernel classification) (Hoi et al. 2013)
- ISKA (Incremental sketched kernel alignment) (Zhang and Liao 2018)
- BOMKR (Budget online multi-kernel regression) (Sahoo et al. 2014)
- BOMKR-V (Variant of BOMKR).

The baseline algorithms for online classification include BOGD, OKS, OMKC and ISKA. The other algorithms including OKS are used for online regression.

We set 9 Gaussian kernels, $\kappa(\mathbf{u}, \mathbf{v}) = \exp(-\|\mathbf{u} - \mathbf{v}\|^2/(2\sigma^2))$, of kernel width σ chosen from $2^{-4:1:4}$. We adopt the best kernel function in hindsight for NORMA and BOGD. BOMKR-V is a variant of BOMKR by changing the loss function. We test the algorithms on online regression and online classification tasks. The datasets are shown in Table 2, which are downloaded from WEKA and UCI machine learning repository.[2] *ailerons-v*, *Hardware-v*, *Twitter-v* and *Adv-SUSY-v* are constructed from *ailerons*, *Hardware*, *Twitter* and *Adv-SUSY*, respectively. For instance, we extract the first 6 features of *ailerons* and form *ailerons-v*. Our goal is to make $d < K$ ($K = 9$). We preprocess *Hardware* and *Twitter* by dividing the standard deviation. Note that we convert *magic04*, *a9a* and *SUSY* to adversarial datasets, denoted by *Adv-magic04*, *Adv-a9a* and *Adv-SUSY*. Our approach of constructing adversarial datasets is as follows: At each round $t = 1, \ldots, T$,

- If $t \leq \lceil T/20 \rceil$, let *Adv-magic04* equal to *magic04*.
- If $t \geq \lceil T/20 \rceil + 1$, we multiply the features of *magic04* by 2^{-3}.

The same operation is used to *Adv-a9a* and *Adv-SUSY*. There are two reasons that we construct adversarial datasets, i.e., (i) for online learning, the data may not be i.i.d., and may be provided by a malicious adversary; (ii) our theoretical results hold in the worst-case. The three adversarial datasets essentially yield hard learning tasks.

For online regression, we adopt the absolute loss $\ell(\hat{y}_t, y) = |\hat{y}_t - y|$ except for NORMA and BOKMR. NORMA adopts the ϵ-insensitive absolute loss $\ell(\hat{y}_t, y) = \max(0, |\hat{y}_t - y| - \varepsilon_t) + \nu\varepsilon_t$, and updates ε_t on the fly. For BOKMR, we adopt the version that uses NORMA as a sub-algorithm (Sahoo et al. 2014). We set $\nu = 0.5$ and $\varepsilon_1 = 0.001$. For online classification, we adopt the hinge loss $\ell(\hat{y}_t, y) = \max\{0, 1 - \hat{y}_t y\}$. We measure the Average Absolute Loss (AAL) defined by $\text{AAL} = \frac{1}{T}\sum_{t=1}^{T} |\hat{y}_t - y_t|$ for online regression, and measure the Average Mistake Rate (AMR) defined by $\text{AMR} = \frac{1}{T}\sum_{t=1}^{T} \mathbb{I}_{\hat{y}_t \neq y_t}$ for online classification. For OKS, we choose the smoothing parameter $\delta \in \{0.2, 0.02, 0.002\}$. For all of the baseline algorithms, we set the stepsize of gradient descent to $5/\sqrt{T}$. The other hyper-parameters are set to the recommended value in original papers. For PFMBooks, we set $g(U_j, D_i) = U_j + 0.1$ where $D_i = 1$ for Gaussian kernel and set $\eta = \sqrt{8\ln(KMT)/T}$. For LKMBooks, we set $\eta = \sqrt{8\ln(K)/T}$. All algorithms are implemented in R on a Windows machine with 2.5 GHz Core(TM) i5-7200U CPU. To weaken the randomization, we execute each experiment 20 times with random permutation of all datasets and average all the results.

5.2 Memory constraints

5.2.1 Online regression

Let $\mathcal{T}$ be a given memory budget. According to Assumptions 2 and 3, we can reduce $\mathcal{T}$ to an example budget of size B. We must ensure that all algorithms have the same space complexity. Table 3 shows the results. Since OKS does not control the number of support vectors, we use a heuristic variant, called BOKS, which stops updating hypothesis if the number of support vectors equals B. We use NORMA as the baseline, that is, for a memory

[2] http://archive.ics.uci.edu/ml/index.php

Table 2 Basic information of datasets

Dataset	Number of examples	Number of feature	Type
Housing	14,000	8	Regression
Ailerons	13,750	40	Regression
Ailerons-v	13,750	6	Regression
Elevators	16,599	18	Regression
Hardware	28,179	96	Regression
Hardware-v	28,179	3	Regression
Twitter	50,000	77	Regression
Twitter-v	50,000	3	Regression
Slice	53,500	384	Regression
Mushrooms	8,124	112	Binary classification
Magic04	19,020	10	Binary classification
Adv-magic04	19,020	10	Binary classification
Adv-a9a	16,281	123	Binary classification
Adv-SUSY	50,000	18	Binary classification
Adv-SUSY-v	50,000	6	Binary classification
cod-rna	59,535	8	Binary classification

budget $\mathcal{T}$, NORMA can use an example budget of size B_0. The third row of Table 3 is the available budget of each algorithm, which depends on the relation between d and K. BOKS and BOMKR do not share the memory and maintain K different sets of support vectors. For LKMBooks and PFMBooks, we set $\upsilon = \frac{1}{3}$ for satisfying $2(1-\upsilon)T^{1-\upsilon} > B$ (see Theorems 2 and 4), and set the stepsize to the values in Theorems 3 and 6. For PFMBooks, we set $U = \sqrt{B}$, $U_{\min} = U/\sqrt{T}$ as stated in Theorem 6. Since LKMBooks and PFMBooks can only achieve the memory constraints in high-probability, we stop updating hypotheses when the actual budget exceeds the available budget in Table 3.

Table 4 shows the empirical results. The bold in each column indicates the algorithm enjoying the best performance. It can be found that NORMA performs well on some datasets. There are two reasons: (i) we select the best kernel width in hindsight for NORMA, that is, we test all of the candidate kernel widths and select the one with minimal ALL; (ii) NORMA uses a good learning rate on those datasets. Tuning the learning rate is another problem of online learning algorithms. To avoid this issue, we set a fixed learning rate for baseline algorithms and use the theoretical values for our algorithms. In the first column of Table 4, we give the optimal kernel width of NORMA on each dataset. For instance, NORMA-2 means that the optimal kernel width is $\sigma = 2$ on *housing* dataset. For different datasets, the optimal kernel width is also different. Thus if we empirically set a fixed kernel for all datasets, then NORMA will perform badly on some datasets. On the contrary, the online kernel selection algorithms and online multi-kernel learning algorithms can perform well on all datasets (except for BOKS). The results verify the first goal, **G** 1.

Next we analyze BOMKR. Since BOMKR does not share the support vectors, $\forall i \in [K]$, the available budget for constructing $\{f_{t,i}\}_{t=1}^{T}$ is $\frac{B_0}{K} \ll B_0$. Thus BOMKR performs bad. LKMBooks, PFMBooks and BOMKR-V can share the support vectors, whose available budget is B_0, $\frac{dB_0}{(d+K')}$ and $\frac{dB_0}{(d+K)}$, respectively. Thus they perform well on all of the datasets.

Table 3 Space complexity and the available budget of individual algorithm

Algorithm	NORMA	BOKS	BOMKR	BOMKR-V	LKMBooks	PFMBooks
Space complexity	$O(Bd)$	$O(B(d+K))$	$O(KBd)$	$O(B(d+K))$	$O(Bd+K)$	$O(B(d+K'))$
Available budget	B_0	$\frac{dB_0}{d+K}$	$\frac{B_0}{K}$	$\frac{dB_0}{d+K}$	B_0	$\frac{dB_0}{d+K'}$

$K' = K\lceil \ln \sqrt{T} \rceil$

Besides, we also find that BOMKR-V performs worse than NORMA on some datasets. The main reason is that the learning rate of BOMKR-V is not well tuned. Since PFMBooks is applicable for the case of $K < d/\lceil \ln T \rceil$, we do not run it on the two low dimensional datasets, *housing* and *elevators*. PFMBooks performs much better than all of the other algorithms on *Slice* dataset. The reason is that PFMBooks is parameter-free and uses a suitable learning rate. For all of the other algorithms including LKMBooks, we actually do not set a suitable learning rate for individual dataset. The results verify the second goal, **G** 2.

5.2.2 Online classification

The overall parameter setting is same with that of online regression, except that LKMBooks uses the same learning rate with the baseline algorithms, i.e., $\lambda = 5/\sqrt{T}$. Let $U_{\min} = 5$ for PFMBooks. For the hinge loss, if f satisfies $\|f\|_{\mathcal{H}} < 1$, then $L_T(f) = \sum_{t=1}^{T}(1 - y_t f(\mathbf{x}_t)) = \Theta(T)$. Thus we set $U_{\min} > 1$. OMKC is an algorithm framework, based on which four algorithms are derived (Hoi et al. 2013). In the case of memory constraints, algorithms can suffer more time cost. Thus we adopt $\text{OMKC}_{D,D}$ which has the best prediction performance, but also suffers the highest time cost among the four algorithms. We set the hyper-parameters of $\text{OMKC}_{D,D}$ to the recommended values in original paper.

We still reduce $\mathcal{T}$ to an example budget of size B and ensure all algorithms have the same space complexity. If the number of support vectors of $\text{OMKC}_{D,D}$ equals B, then we stop updating hypotheses. We use BOGD as the baseline, whose space complexity is $O(Bd)$. Given $\mathcal{T}$ memory budget, BOGD can use an example budget of size B_0. The space complexity of $\text{OMKC}_{D,D}$ is $O(B(d+K))$. Thus $B = \frac{dB_0}{d+K}$. The space complexity of ISKA is $O(Bd+K)$. Thus $B = B_0$. Table 3 gives the size of example budget of other algorithms.

Table 5 shows the empirical results. It can be find that BOGD performs well on all datasets, since we select the optimal kernel width in hindsight. The first column shows the optimal kernel width on different datasets can be different, which is same with the result of Table 4. Thus we conclude that, if BOGD is equipped with a fixed kernel function for all datasets, then it will perform worse than the other algorithms. The results verify **G** 1.

Next we analyze $\text{OMKC}_{D,D}$, which performs bad on the last three datasets. We call the last three datasets *hard dataset* and call *mushrooms easy dataset*, since the mistake rates are very small on *mushrooms*. Recalling that $\text{OMKC}_{D,D}$ can use a budget of size $\frac{dB_0}{d+K}$. $\text{OMKC}_{D,D}$ does not share the memory, and thus it allocates the budget over K hypothesis sequences, i.e., $\{f_{t,i}\}_{t=1}^{T}, i \in [K]$. In this way, each hypothesis sequence approximately obtains a budget of size $\frac{1}{K} \cdot \frac{dB_0}{d+K}$. Thus it would perform bad on *hard dataset*. For *mushrooms*, since the number of mistakes is very small, thus a small budget is enough. For instance, for the case of $B_0 = 200$, the number of mistakes of $\text{OMKC}_{D,D}$ is roughly

Table 4 AAL (Average Absolute Loss) comparison within memory constraints

Algorithm	Housing ($B_0 = 50$)		($B_0 = 200$)		($B_0 = 400$)	
	AAL	Time (s)	AAL	Time (s)	AAL	Time (s)
NORMA-2	$\mathbf{0.1790 \pm 0.0003}$	0.27	0.1651 ± 0.0003	0.50	0.1615 ± 0.0003	0.77
BOKS	0.2078 ± 0.0137	0.30	0.2016 ± 0.0064	0.32	0.1982 ± 0.0138	0.38
BOMKR	0.2182 ± 0.0003	1.45	0.1876 ± 0.0003	1.80	0.1768 ± 0.0002	2.11
BOMKR-V	0.1896 ± 0.0004	0.96	$\mathbf{0.1613 \pm 0.0002}$	2.48	0.1512 ± 0.0002	4.29
LKMBooks	0.1832 ± 0.0075	0.80	0.1636 ± 0.0096	1.90	$\mathbf{0.1461 \pm 0.0055}$	3.52
PFMBooks	–	–	–	–	–	–
Algorithm	Elevators ($B_0 = 50$)		($B_0 = 200$)		($B_0 = 400$)	
	AAL	Time (s)	AAL	Time (s)	AAL	Time (s)
NORMA-1	0.0630 ± 0.0001	0.37	0.0554 ± 0.0001	0.73	0.0539 ± 0.0001	1.11
BOKS	0.1286 ± 0.0145	0.35	0.1254 ± 0.0175	0.41	0.1245 ± 0.0143	0.45
BOMKR	0.0714 ± 0.0001	1.87	0.0651 ± 0.0001	2.29	0.0621 ± 0.0001	2.94
BOMKR-V	0.0671 ± 0.0002	1.22	0.0596 ± 0.0001	2.95	0.0567 ± 0.0001	5.27
LKMBooks	$\mathbf{0.0602 \pm 0.0025}$	0.99	$\mathbf{0.0539 \pm 0.0012}$	2.41	$\mathbf{0.0515 \pm 0.0008}$	4.26
PFMBooks	–	–	–	–	–	–
Algorithm	Hardware ($B_0 = 50$)		($B_0 = 200$)		($B_0 = 400$)	
	AAL	Time (s)	AAL	Time (s)	AAL	Time (s)
NORMA-8	$\mathbf{0.2442 \pm 0.0001}$	1.07	$\mathbf{0.2303 \pm 0.0001}$	2.82	0.2261 ± 0.0001	5.73
BOKS	0.2755 ± 0.0082	0.77	0.2747 ± 0.0136	0.91	0.2673 ± 0.0157	1.16
BOMKR	0.2564 ± 0.0001	3.99	0.2515 ± 0.0001	5.70	0.2462 ± 0.0001	8.78
BOMKR-V	0.2516 ± 0.0001	2.31	0.2377 ± 0.0001	5.94	0.2308 ± 0.0001	12.05
LKMBooks	0.2440 ± 0.0089	1.95	0.2298 ± 0.0034	5.08	$\mathbf{0.2244 \pm 0.0032}$	9.28
PFMBooks	0.2534 ± 0.0018	6.95	0.2432 ± 0.0040	9.35	0.2388 ± 0.0047	13.13
Algorithm	Twitter ($B_0 = 50$)		($B_0 = 200$)		($B_0 = 400$)	
	AAL	Time (s)	AAL	Time (s)	AAL	Time (s)
NORMA-4	$\mathbf{0.1875 \pm 0.0001}$	1.72	$\mathbf{0.1584 \pm 0.0001}$	4.27	0.1519 ± 0.0001	9.12
BOKS	0.2228 ± 0.0127	1.25	0.2025 ± 0.0213	1.60	0.1947 ± 0.0218	2.35
BOMKR	0.2115 ± 0.0001	6.82	0.2019 ± 0.0001	9.37	0.2019 ± 0.0001	13.83
BOMKR-V	0.2004 ± 0.0001	4.19	0.1718 ± 0.0001	10.44	0.1599 ± 0.0001	20.41
LKMBooks	0.1847 ± 0.0114	3.18	0.1592 ± 0.0059	8.86	$\mathbf{0.1503 \pm 0.0024}$	16.38
PFMBooks	0.2020 ± 0.0061	11.36	0.1767 ± 0.0081	15.80	0.1655 ± 0.0065	22.47
Algorithm	Slice ($B_0 = 50$)		($B_0 = 200$)		($B_0 = 400$)	
	AAL	Time (s)	AAL	Time (s)	AAL	Time (s)
NORMA-4	0.3818 ± 0.0001	5.56	0.3504 ± 0.0001	18.51	0.3317 ± 0.0001	42.20
BOKS	0.3866 ± 0.0087	2.27	0.3448 ± 0.0122	3.64	0.3073 ± 0.0193	6.42
BOMKR	0.3972 ± 0.0001	14.34	0.3924 ± 0.0001	27.46	0.3857 ± 0.0001	48.97
BOMKR-V	0.3737 ± 0.0001	8.85	0.3308 ± 0.0001	27.35	0.3052 ± 0.0001	58.00
LKMBooks	0.3439 ± 0.0070	6.84	0.3095 ± 0.0042	21.87	0.2926 ± 0.0043	41.69
PFMBooks	$\mathbf{0.3392 \pm 0.0110}$	14.50	$\mathbf{0.2819 \pm 0.0163}$	24.11	$\mathbf{0.2467 \pm 0.0185}$	37.02

$0.62 * T \approx 50$, where $T = 8124$. Thus the optimal hypothesis sequence $\{f_{t,i^*}\}_{t=1}^{T}$ only needs a budget of size about 50. LKMBooks shares the memory and performs well on hard dataset. The experimental results do not match our theoretical results well, since we focus on the mistake rates not the average cumulative losses. Our theoretical results are the regret bounds, not the mistake bounds. Even so, the experimental results on the *hard datasets* still verify **G** 2.

ISKA also shares the memory and performs better than our algorithms on *mushrooms* and *magic04*, since it employs an elaborate removing strategy, while our algorithms just use simple randomized adding strategies. However, the regret bounds of ISKA does not reveal the superiority. We conjecture that data-dependent regret bounds can explain the superiority. Besides, ISKA performs worse than our algorithms on the two adversarial datasets. The kernel selection procedure of ISKA consists of two phases. During the first phase, ISKA converges to an empirically optimal kernel. During the second phase, ISKA always chooses the empirically optimal kernel. The adversary can easily change the optimal kernel by scaling the feature of instances and make ISKA converge to a bad kernel. Our algorithms randomly choose kernels and can converge to the optimal kernel defined on the whole datasets. Thus our algorithms are more robust than ISKA in adversarial environments.

5.3 Time constraints

5.3.1 Online regression

Let $\mathcal{T}$ be a given time budget. We also achieve the time constraints by fixing the budget size. To be specific, we choose BOMKR as baseline, where the budget is set to B_0. Denote the average per-round running time of BOMKR by t_{p}. We tune the budget of other algorithms for ensuring the same running time with t_{p}. For BATBooks, we set the learning rate $\eta = 4\sqrt{\ln K/(K\tilde{C}_{T,*})}$, where $\tilde{C}_{T,*}$ is tuned by the doubling trick, $U = B_0^{\frac{1}{3}}$ and $\ell_{\max} = 1$. For the parameter υ, we choose the maximal value from $\{1/i\}_{i=3,4,\ldots,12}$ for satisfying the condition (12). For the other algorithms, the parameter setting keeps unchanged.

Table 6 shows the empirical results. First, we consider the results on four high dimensional datasets, *elevator*, *ailerons*, *Hardware* and *Twitter*. In this case, we have $K < d$. Within a same time budget, LKMBooks shows the best performance except for NORMA. Although LKMBooks is designed for memory constraints, it is still nearly optimal for time constraints. In the second and fifth columns, the available budgets of all algorithms are different, since the per-round time complexities are different. It seems strange that BOKS has the maximal available budget. The reason is that BOKS allocates the available budget B_0 to K hypotheses $\{f_{t,i}\}_{i=1}^{K}$. Thus the available budget of each $f_{t,i}$ is less than B_0. The results verify the third goal, **G** 3.

Next we consider the four low dimensional datasets, *housing*, *ailerons-v*, *Hardware-v* and *Twitter-v*. In this case, we have $K > d$. Within a same time budget, BATBooks shows the best performance on all datasets except for NORMA. NORMA performs well, since it has the lowest time complexity and we set the optimal kernel width in hindsight. It is interesting to find that, the available budget of BATBooks is similar with that of NORMA. The reason is that the two algorithms have same per-round time complexity, which is $O(dB + K)$ and $O(dB)$, respectively. BATBooks performs better than LKMBooks for the case of $d < K$, which verifies the fourth goal, **G** 4.

Table 5 AMR (Average Mistake Rate) comparison within memory constraints

Algorithm	Mushrooms ($B_0 = 200$)		($B_0 = 400$)		($B_0 = 600$)	
	AMR (%)	Time (s)	AMR (%)	Time (s)	AMR (%)	Time (s)
BOGD-1	1.13 ± 0.11	2.05	0.48 ± 0.08	3.89	$\mathbf{0.31 \pm 0.04}$	5.90
BOKS	3.98 ± 0.29	0.38	3.25 ± 0.09	0.40	3.24 ± 0.10	0.38
$\text{OMKC}_{D,D}$	$\mathbf{0.62 \pm 0.19}$	3.54	$\mathbf{0.33 \pm 0.04}$	4.93	0.34 ± 0.04	5.10
ISKA	3.68 ± 0.34	5.14	3.39 ± 0.19	12.01	2.03 ± 0.12	21.64
LKMBooks	3.70 ± 0.56	1.51	3.23 ± 0.45	2.45	3.03 ± 0.28	3.30
PFMBooks	6.07 ± 0.79	3.31	4.46 ± 0.66	4.83	3.88 ± 0.57	6.60
Algorithm	Magic04 ($B_0 = 200$)		($B_0 = 400$)		($B_0 = 600$)	
	AMR (%)	Time (s)	AMR (%)	Time (s)	AMR (%)	Time (s)
BOGD-2^4	25.75 ± 0.27	1.34	23.96 ± 0.19	2.35	23.09 ± 0.18	2.66
BOKS	35.02 ± 1.03	0.66	34.81 ± 1.20	0.72	34.38 ± 1.34	0.74
$\text{OMKC}_{D,D}$	34.35 ± 1.88	4.82	33.19 ± 1.31	6.38	31.46 ± 1.22	7.67
ISKA	$\mathbf{23.79 \pm 0.59}$	4.03	$\mathbf{21.87 \pm 0.43}$	6.96	$\mathbf{21.19 \pm 0.30}$	9.79
LKMBooks	26.66 ± 0.87	2.91	24.48 ± 0.77	3.83	23.44 ± 0.56	4.80
PFMBooks	–	–	–	–	–	–
Algorithm	Adv-magic04 ($B_0 = 200$)		($B_0 = 400$)		($B_0 = 600$)	
	AMR (%)	Time (s)	AMR (%)	Time (s)	AMR (%)	Time (s)
BOGD-2	$\mathbf{26.35 \pm 0.26}$	1.35	24.48 ± 0.21	1.95	23.64 ± 0.18	2.60
BOKS	35.15 ± 0.74	0.68	33.72 ± 2.23	0.70	31.21 ± 1.99	0.78
$\text{OMKC}_{D,D}$	34.76 ± 0.96	4.90	34.38 ± 1.02	6.43	33.31 ± 2.50	7.71
ISKA	28.50 ± 3.66	3.44	27.87 ± 2.45	5.41	26.88 ± 2.52	5.93
LKMBooks	26.81 ± 2.19	2.85	$\mathbf{24.40 \pm 1.08}$	3.91	$\mathbf{23.38 \pm 0.09}$	5.10
PFMBooks	–	–	–	–	–	–
Algorithm	Adv-a9a ($B_0 = 200$)		($B_0 = 400$)		($B_0 = 600$)	
	AMR (%)	Time (s)	AMR (%)	Time (s)	AMR (%)	Time (s)
BOGD-2^{-3}	$\mathbf{19.33 \pm 0.24}$	4.50	$\mathbf{18.69 \pm 0.21}$	8.91	$\mathbf{18.44 \pm 0.22}$	13.91
BOKS	24.29 ± 0.56	0.80	23.73 ± 1.38	1.06	22.97 ± 0.99	1.30
$\text{OMKC}_{D,D}$	21.93 ± 1.96	9.06	21.64 ± 2.29	15.97	21.04 ± 2.78	23.05
ISKA	23.63 ± 0.02	12.27	23.63 ± 0.01	29.72	23.54 ± 0.27	43.57
LKMBooks	20.40 ± 1.52	4.08	19.37 ± 0.55	6.87	18.88 ± 0.52	9.01
PFMBooks	22.37 ± 1.18	7.50	21.49 ± 0.72	9.35	21.00 ± 0.51	15.50

The bold in each column of the tables indicates the algorithm enjoying the best performance

5.3.2 Online classification

For LKMBooks, the parameters follow the setting in Sect. 5.2.2. For BATBooks, the parameters follow the setting in Sect. 5.3.1, except that the stepsize is set to $\lambda = \frac{U\sqrt{(1+\upsilon)B}}{\sqrt{2(1-\upsilon)LT}}$ which is slightly different from that of Theorem 10. We choose $\text{OMKC}_{D,D}$ as baseline,

where the budget is set to B_0. Let t_p be the average per-round running time of $\text{OMKC}_{D,D}$. We tune the budget of other algorithms for ensuring the same running time with t_p.

Table 7 shows the empirical results. We first consider the results on two high-dimensional datasets, *mushrooms* and *Adv-a9a* in which $K \ll d$. Within a same time budget, LKMBooks performs better than BATBooks. For *Adv-SUSY*, we have $K \approx d$ ($K = 9, d = 18$). LKMBooks shows similar performance with BATBooks. The same result holds for *Adv-magic04*, in which $K = 9$ and $d = 10$. Besides, $\text{OMKC}_{D,D}$ performs much better than other algorithms on *mushrooms*. The reason is same with the analysis on *mushrooms* in Sect. 5.2.2. As a whole, for the case of $K \geq d$, LKMBooks performs well on most of dataset. The results verify **G** 3.

Next we consider the two low-dimensional datasets, *cod-rna* and *Adv-SUSY-v* in which $d < K$. We find that LKMBooks performs slightly better than BATBooks on *cod-rna*, and performs worse than BATBooks on *Adv-SUSY-v*. The results does not fully verify **G** 4. There may be two reasons: (i) for *cod-rna*, we have $d \approx K$ ($d = 8, K = 9$); (ii) the performance measure is the mistakes rate, not the average cumulative losses. Even so, our algorithms still perform better than $\text{OMKC}_{D,D}$ and ISKA.

6 Conclusion and discussion

In this paper, we studied the computationally budgeted online kernel selection, where the kernel selection and online prediction procedures face memory constraints or time constraints. We separately proved a lower bound on the regret under the two kinds of computational constraints, and developed several simple algorithms that nearly achieve the lower bounds. We also identified the condition under which online kernel selection with a time constraint is different from that with a memory constraint.

This work will open up many directions for future research. One of the most important research is to identify the sufficient conditions under which a constant computational constraint can achieve a sub-linear regret bound. Model selection aims at choosing the inductive bias that matches the data and improving the learning performance of algorithms. Thus the worst-case regret guarantees do not reveal the essence of model selection. The sufficient conditions play the role of inductive bias. To this end, it is necessary to establish some kind of data-dependent regret bounds. Although many work has focus on achieving data-dependent regret bounds for general online learning problem, such as prediction with expert advice, multi-armed bandit problems, online convex optimization and so on, few of them considers the computational constraints.

We need further study the worst-case regret analysis. For the case of memory constraints and $K > d/\ln\sqrt{T}$, our algorithm can not adapt to the norm of competitor. Thus the regret bound is far from optimality in terms of $\|f_i^*\|_{\mathcal{H}_i}$. For the case of time constraints and $K > d$, if $\mathcal{T} = \omega(T/K)$, then there is a gap of order $\sqrt{K}$ between the lower bound and upper bound. It is necessary to study whether this gap can be removed. Besides, the algorithm can also not adapt to the norm of competitor.

Table 6 AAL (Average Absolute Loss) comparison within time constraints

Algorithm	Elevators			Housing		
	B_0	AAL	$t_p \times 10^{-4}$(s)	B_0	AAL	$t_p \times 10^{-4}$(s)
NORMA-(1,2)	1000	0.0536 ± 0.0001	1.73 ± 0.12	1250	0.1447 ± 0.0003	1.40 ± 0.04
BOKS	6100	0.1734 ± 0.0184	1.71 ± 0.13	5800	0.1891 ± 0.0252	1.44 ± 0.18
BOMKR	50	0.0617 ± 0.0001	1.74 ± 0.07	50	0.1749 ± 0.0002	1.47 ± 0.02
BOMKR-V	190	0.0599 ± 0.0001	1.72 ± 0.02	160	0.1657 ± 0.0004	1.44 ± 0.02
LKMBooks	260	$\mathbf{0.0531 \pm 0.0014}$	1.76 ± 0.02	220	0.1592 ± 0.0082	1.47 ± 0.02
BATBooks	1250	0.0581 ± 0.0020	1.73 ± 0.01	1300	$\mathbf{0.1424 \pm 0.0020}$	1.43 ± 0.05
Algorithm	Ailerons			Ailerons-v		
	B_0	AAL	$t_p \times 10^{-4}$(s)	B_0	AAL	$t_p \times 10^{-4}$(s)
NORMA-8	830	0.0774 ± 0.0001	2.15 ± 0.06	1400	$\mathbf{0.0839 \pm 0.0002}$	1.42 ± 0.02
BOKS	3900	0.0952 ± 0.0076	2.18 ± 0.23	5500	0.1296 ± 0.0126	1.41 ± 0.19
BOMKR	50	0.0842 ± 0.0003	2.13 ± 0.07	50	0.1030 ± 0.0003	1.47 ± 0.08
BOMKR-V	220	0.0791 ± 0.0002	2.11 ± 0.02	150	0.0934 ± 0.0001	1.40 ± 0.04
LKMBooks	300	$\mathbf{0.0685 \pm 0.0034}$	2.17 ± 0.05	220	0.0966 ± 0.0089	1.43 ± 0.02
BATBooks	1000	0.0732 ± 0.0040	2.12 ± 0.03	1300	0.0925 ± 0.0042	1.49 ± 0.06
Algorithm	Hardware			Hardware-v		
	B_0	AAL	$t_p \times 10^{-4}$(s)	B_0	AAL	$t_p \times 10^{-4}$(s)
NORMA-8	650	0.2250 ± 0.0001	3.12 ± 0.20	1750	0.2426 ± 0.0003	1.27 ± 0.02
BOKS	3900	0.2435 ± 0.0288	3.10 ± 0.49	6720	0.2510 ± 0.0214	1.26 ± 0.31
BOMKR	50	0.2452 ± 0.0001	3.18 ± 0.19	50	0.2556 ± 0.0001	1.24 ± 0.02
BOMKR-V	310	0.2332 ± 0.0001	3.12 ± 0.01	200	0.2497 ± 0.0001	1.25 ± 0.01
LKMBooks	400	$\mathbf{0.2233 \pm 0.0022}$	3.13 ± 0.07	280	0.2394 ± 0.0064	1.26 ± 0.02
BATBooks	900	0.2363 ± 0.0056	3.15 ± 0.09	1700	$\mathbf{0.2327 \pm 0.0033}$	1.24 ± 0.01
Algorithm	Twitter			Twitter-v		
	B_0	AAL	$t_p \times 10^{-4}$(s)	B_0	AAL	$t_p \times 10^{-4}$(s)
NORMA-4	640	$\mathbf{0.1499 \pm 0.0001}$	2.69 ± 0.05	1300	$\mathbf{0.1482 \pm 0.0001}$	1.33 ± 0.11
BOKS	4450	0.1659 ± 0.0159	2.68 ± 0.48	6500	0.1637 ± 0.0097	1.30 ± 0.21
BOMKR	50	0.1885 ± 0.0001	2.65 ± 0.03	50	0.1868 ± 0.0001	1.35 ± 0.03
BOMKR-V	260	0.1667 ± 0.0001	2.65 ± 0.01	130	0.1805 ± 0.0001	1.30 ± 0.02
LKMBooks	330	0.1535 ± 0.0044	2.62 ± 0.03	180	0.1611 ± 0.0071	1.35 ± 0.12
BATBooks	930	0.1603 ± 0.0043	2.64 ± 0.03	1500	0.1533 ± 0.0032	1.37 ± 0.01

The bold in each column of the tables indicates the algorithm enjoying the best performance

Appendix

Proof of Theorem 1

Table 7 AMR (Average Mistake Rate) comparison within time constraints

Algorithm	Mushrooms			cod-rna		
	B_0	AMR	$t_p \times 10^{-4}$(s)	B_0	AMR	$t_p \times 10^{-4}$(s)
BOGD-$(1, 2^4)$	360	$\mathbf{0.50 \pm 0.07}$	4.52 ± 0.03	1250	12.90 ± 0.04	2.48 ± 0.14
BOKS	∞	3.26 ± 0.10	0.52 ± 0.03	∞	17.86 ± 0.16	2.30 ± 0.44
OMKC$_{D,D}$	200	0.62 ± 0.19	4.35 ± 0.09	200	20.71 ± 2.84	2.46 ± 0.20
ISKA	150	3.82 ± 0.78	4.81 ± 0.15	550	$\mathbf{12.84 \pm 0.05}$	2.50 ± 0.19
LKMBooks	600	3.03 ± 0.28	4.06 ± 0.00	950	13.28 ± 0.20	2.44 ± 0.22
BATBooks	850	5.75 ± 0.87	4.35 ± 0.07	2800	13.68 ± 0.18	2.41 ± 0.17
Algorithm	Adv-a9a			Adv-magic04		
	B_0	AMR	$t_p \times 10^{-4}$(s)	B_0	AMR	$t_p \times 10^{-4}$(s)
BOGD-$(2^{-3}, 2)$	430	$\mathbf{18.67 \pm 0.22}$	5.87 ± 0.43	1300	$\mathbf{22.35 \pm 0.18}$	2.57 ± 0.22
BOKS	∞	23.40 ± 0.40	4.82 ± 0.74	∞	27.89 ± 0.59	1.26 ± 0.24
OMKC$_{D,D}$	200	21.93 ± 1.96	5.67 ± 0.31	200	34.76 ± 0.96	2.58 ± 0.01
ISKA	170	23.63 ± 0.02	5.73 ± 0.16	380	26.88 ± 3.23	2.66 ± 0.23
LKMBooks	600	18.88 ± 0.52	5.54 ± 0.12	520	23.58 ± 0.66	2.53 ± 0.10
BATBooks	1800	19.95 ± 0.41	5.35 ± 0.25	2300	23.63 ± 0.44	2.51 ± 0.11
Algorithm	Adv-SUSY			Adv-SUSY-v		
	B_0	AMR	$t_p \times 10^{-4}$(s)	B_0	AMR	$t_p \times 10^{-4}$(s)
BOGD-2^{-3}	1100	28.02 ± 0.13	3.34 ± 0.24	1800	$\mathbf{32.72 \pm 0.18}$	2.68 ± 0.19
BOKS	7500	29.40 ± 1.28	3.37 ± 0.93	12000	37.08 ± 0.76	2.83 ± 0.55
OMKC$_{D,D}$	200	43.61 ± 1.80	3.38 ± 0.25	200	44.23 ± 1.78	2.72 ± 0.09
ISKA	360	43.44 ± 2.37	3.19 ± 0.23	600	46.22 ± 3.24	2.82 ± 0.23
LKMBooks	450	27.70 ± 0.58	3.34 ± 0.06	300	35.46 ± 1.56	2.84 ± 0.06
BATBooks	2350	$\mathbf{27.24 \pm 0.39}$	3.12 ± 0.08	2500	33.09 ± 0.36	2.67 ± 0.20

The bold in each column of the tables indicates the algorithm enjoying the best performance
∞ means $B_0 = T$

Proof We use the hinge loss $\ell(u, y) = \max\{0, 1 - yu\}$ as an example. Our analysis is also applicable to the absolute loss. We select K Gaussian kernel functions $\kappa_i(\mathbf{x}, \mathbf{z}) = \exp(-\frac{\|\mathbf{x}-\mathbf{z}\|_2^2}{\sigma_i})$, $i = 1, \ldots, K$ as the candidates. Without loss of generality, we assume that $0 < \sigma_1 < \sigma_2 < \ldots < \sigma_K$. Our proof is based on a sequence of instances $\mathcal{S} = \{\mathbf{x}_t\}_{t=1}^T$, such that

$$\forall \mathbf{x}_i \neq \mathbf{x}_j \in \mathcal{S}, \|\mathbf{x}_i - \mathbf{x}_j\|_2 = D \neq 0,$$

where D is a constant. For $r = 1, \ldots, K$ and $i \neq j$, we have

$$\kappa_r(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma_r^2}\right) = \exp\left(-\frac{D^2}{2\sigma_r^2}\right) = c_r,$$

where $c_1 < \ldots < c_K < 1$. For an Euclid space $\mathbb{R}^d$, we can always find $d + 1$ points satisfying the property. Note that we do not require the instances are orthogonal or approximately

orthogonal in RKHSs, which is different from the techniques adopted by (Dekel et al. 2008; Zhang et al. 2013; Cesa-Bianchi et al. 2015). We assume that $T \leq d+1$ and T is even. Next we will design a strategy for the adversary, based on which the adversary sends examples to the learner.

Before the game, the adversary assigns a label y_t for each instance $\mathbf{x}_t$, satisfying $y_t = 1$ if t is odd, otherwise, $y_t = -1$. Define a sequence of example pairs $s_i = \{(\mathbf{x}_i, y_i), (\mathbf{x}_{i+1}, y_{i+1})\}$, where $i = 1, 3, 5, \ldots$. The adversary assigns the examples $\{(\mathbf{z}_t, y'_t)\}_{t=1}^T$ as follows,

- *Case 1* $T \leq 2\mathrm{e}^{\frac{1}{4}} B$

  If t is odd, the adversary selects $(\mathbf{z}_t, y'_t) \in s_t$ uniformly. Otherwise, the adversary assigns $(\mathbf{z}_t, y'_t) \in s_{t-1} \backslash \{(\mathbf{z}_{t-1}, y'_{t-1})\}$.
- *Case 2* $T \geq 2\mathrm{e}^{\frac{1}{4}} B + 1$

  If $t \leq 2B$ and t is odd, the adversary selects $(\mathbf{z}_t, y'_t) \in s_t$ uniformly. If $t \leq 2B$ and t is even, the adversary assigns $(\mathbf{z}_t, y'_t) \in s_{t-1} \setminus \{(\mathbf{z}_{t-1}, y'_{t-1})\}$. If $t \geq 2B+1$, the adversary divides the time horizon $\{2B+1, \ldots, T\}$ into continuous epochs with length m, except for the last epoch. We require that m is even. Assuming there are $\Delta + 1$ epoches. Let $m = \lceil \frac{T-2B_i}{(2\mathrm{e}^{\frac{1}{4}}-2)B_i+1} \rceil$. If m is odd, then let $m = m+1$. Thus $\Delta = \lfloor \frac{T-2B}{m} \rfloor$. If the length of the last epoch is odd, then we add one more new example. For the r-th epoch, denote the start point as $s_r = (r-1)m + 2B + 1$ and the end point as $e_r = rm + 2B$. If $t = s_r$, then the adversary selects $(\mathbf{z}_t, y'_t) \in s_t$ uniformly, and assigns $(\mathbf{z}_{t+1}, y'_{t+1}) \in s_t \setminus \{(\mathbf{z}_t, y'_t)\}$. If $t = s_r + 2n, n = 1, 2, \ldots, \frac{m}{2} - 1$, the adversary first constructs an example pair $\bar{s}_t = \{(\bar{\mathbf{x}}_t, \bar{y}_t), (\bar{\mathbf{x}}_{t+1}, \bar{y}_{t+1})\}$. The adversary samples $(\bar{\mathbf{x}}_t, \bar{y}_t)$ from S_r uniformly, where S_r is the set of examples selected at the end of the s_r-th round, and then samples $(\bar{\mathbf{x}}_{t+1}, \bar{y}_{t+1})$ from $S_r \setminus \{(\mathbf{x}, y) \in S_r : y = \bar{y}_t\}$ uniformly. After that, the adversary selects $(\mathbf{z}_t, y'_t) \in \bar{s}_t$ uniformly, and assigns $(\mathbf{z}_{t+1}, y'_{t+1}) \in \bar{s}_t \setminus \{(\mathbf{z}_t, y'_t)\}$.

Let S_t be the budget maintained by the learner at the beginning of round t, satisfying $|S_t| \leq B$. The hypothesis f_t used by the learner has the form $f_t = \sum_{\mathbf{x}_\tau \in S_{t,I_t}} a_\tau \kappa_{I_t}(\mathbf{x}_\tau, \cdot)$, where $I_t \in [K]$ is the index of kernel function selected by the learner, and S_{t,I_t} is the budget allocated for κ_{I_t}, satisfying $\bigcup_{i=1}^K S_{t,i} = S_t$. Note that it is possible that $S_{t,1} = \ldots = S_{t,K}$.

Case 1 $T \leq 2\mathrm{e}^{\frac{1}{4}} B$. If t is odd, then it is easy to verify that $f_t(\mathbf{z}_t) = f_t(\mathbf{x}_t) = f_t(\mathbf{x}_{t+1})$. Thus the expected loss of the learner is

$$\ell(f_t(\mathbf{z}_t), y_t) = \frac{1}{2}\max\{0, 1 - f_t(\mathbf{x}_t)\} + \frac{1}{2}\max\{0, 1 + f_t(\mathbf{x}_{t+1})\} \geq 1.$$

If t is even, we have $\ell_t(f_t(\mathbf{z}_t), y_t) \geq 0$. Thus the cumulative loss of the learner is larger than $\frac{T}{2}$. For each $\mathcal{H}_i$, let the optimal hypothesis be $f_i^* = \sum_{\tau=1}^T a_\tau \kappa(\mathbf{x}_\tau, \cdot)$. Next we need to solve the coefficients $a_1, \ldots, a_T$.

First we require f_i^* satisfying condition (13),

$$f_i^*(\mathbf{x}_t) = \sum_{\tau=1}^T a_\tau \kappa_i(\mathbf{x}_\tau, \mathbf{x}_t) = y_t, \ \forall\, t = 1, 2, \ldots, T. \tag{13}$$

From the above condition, we can obtain the relation $f_i^*(\mathbf{x}_t) = f_i^*(\mathbf{x}_{t+2})$ for any t, i.e.,

$$a_t + a_{t+2}c_i + \sum_{\tau \neq t, \tau \neq t+2}^T \alpha_\tau c_i = a_{t+2} + a_t c_i + \sum_{\tau \neq t, \tau \neq t+2}^T \alpha_\tau c_i.$$

Since $c_i \neq 1$, we have $a_t = a_{t+2}$. Thus f_i^* has the form

$$f_i^*(\mathbf{x}_t) = \sum_{\tau=2n+1} a_1 \kappa_i(\mathbf{x}_\tau, \mathbf{x}_t) + \sum_{\tau=2n+2} a_2 \kappa_i(\mathbf{x}_\tau, \mathbf{x}_t), n = 0, 1, 2, \ldots .$$

Furthermore, taking into $f_i^*(\mathbf{x}_1) = 1$ and $f_i^*(\mathbf{x}_2) = -1$ yields condition (14) and (15),

$$a_1 + \sum_{\tau=2n+1, \tau\neq 1} a_1 c_i + a_2 c_i + \sum_{\tau=2n+2, \tau\neq 2} a_2 c_i = 1, \tag{14}$$

$$a_1 c_i + \sum_{\tau=2n+1, \tau\neq 1} a_1 c_i + a_2 + \sum_{\tau=2n+2, \tau\neq 2} a_2 c_i = -1. \tag{15}$$

Since we assume that T is even, solving the above two equations produces

$$a_1 = \frac{1}{1-c_i}, \quad a_2 = -\frac{1}{1-c_i}.$$

Thus the optimal hypothesis f_i^* is

$$f_i^* = \frac{1}{1-c_i} \sum_{\tau=2n+1} \kappa_i(\mathbf{x}_\tau, \cdot) - \frac{1}{1-c_i} \sum_{\tau=2n+2} \kappa_i(\mathbf{x}_\tau, \cdot),$$

which satisfies $\sum_{t=1}^T \ell(f_i^*(\mathbf{x}_t), y_t) = 0$, and

$$\|f_i^*\|_{\mathcal{H}_i} = \sqrt{\frac{\sum_{t,\tau=1}^T (-1)^{t+\tau} \kappa(\mathbf{x}_t, \mathbf{x}_\tau)}{(1-c_i)^2}} = \frac{\sqrt{T + \sum_{t\neq\tau=1}^T (-1)^{t+\tau} c_i}}{1-c_i} = \frac{\sqrt{T - Tc_i}}{1-c_i} = \frac{\sqrt{T}}{\sqrt{1-c_i}}.$$

Then the regret of any budgeted online kernel selection algorithm can be bounded as follows

$$\sum_{t=1}^T \mathbb{E}[\ell(f_t(\mathbf{x}_t), y_t)] - \sum_{t=1}^T \ell(f_i^*(\mathbf{x}_t), y_t) \geq \frac{T}{2} = \frac{\sqrt{1-c_i}}{2} \|f_i^*\|_{\mathcal{H}_i} L\sqrt{T},$$

where $L = \max_t \| - y_t \kappa_i(\mathbf{x}_t, \cdot)\|_{\mathcal{H}_i} = 1$.

Case 2 $T \geq 2\mathrm{e}^{\frac{1}{4}} B + 1$. For the first $2B$ rounds, the expected cumulative losses of any algorithm is larger than B. For $t \geq 2B+1, \ldots, T$, we first analyze the expected loss in a fixed epoch. At the r-th epoch, $r = 1, \ldots, \Delta$, if $t = s_r$, then the expected instantaneous loss is larger than 1. If $t = s_r + 2n, n = 1, 2, \ldots, \frac{m}{2} - 1$, the probability that $\mathbf{z}_t$ and $\mathbf{z}_{t+1}$ are not in S_{t,I_t} is

$$\frac{|S_r| - B_{I_t, y_t'}}{|S_r|} \cdot \frac{\frac{1}{2}|S_r| - B_{I_t, -y_t'}}{\frac{1}{2}|S_r|} \geq 1 - \frac{2B}{|S_r|}.$$

where $B_{I_t, y_t'}$ is the number of examples in S_{t,I_t}, whose label are y_t'. In this case, we still have $f_t(\mathbf{z}_t) = f_t(\bar{\mathbf{x}}_t) = f_t(\bar{\mathbf{x}}_{t+1})$. Thus at round $t = s_r + 2n$, the expected instantaneous loss is larger than $1 - \frac{2B}{|S_r|}$. The expected loss in the r-th epoch satisfies

$$\sum_{t=s_r}^{e_r} \mathbb{E}[\ell(f_t(\mathbf{z}_t), y_t')] \geq \frac{1}{2} + \frac{m-2}{2}\left(1 - \frac{2B}{|S_r|}\right) \geq \frac{m-1}{2} - (m-2)\frac{B}{|S_r|}.$$

Summing over $t = 1, 2, \ldots, T$ gives

$$\begin{aligned}\sum_{t=1}^{T}\mathbb{E}[\ell(f_t(\mathbf{x}_t), y_t)] &= \sum_{t=1}^{2B}\mathbb{E}[\ell(f_t(\mathbf{x}_t), y_t)] + \sum_{t=2B+1}^{T}\mathbb{E}[\ell(f_t(\mathbf{x}_t), y_t)] \\ &\geq B + \sum_{r=1}^{\Delta}\left[\frac{m-1}{2} - (m-2)\frac{B}{|S_r|}\right] + \sum_{t=s_\Delta}^{T}\mathbb{E}[\ell(f_t(\mathbf{x}_t), y_t)] \\ &\geq \frac{1}{2}\left[T - \Delta - 2(m-2)\sum_{r=1}^{\Delta}\frac{B}{2B+r+1}\right] \quad (|S_r| = 2B + r + 1) \\ &\geq \frac{1}{2}\left[T - \Delta - 2(m-2)B\ln\frac{2B+\Delta-1}{2B}\right] \\ &\geq \frac{1}{2}\left[T - \Delta - (m-2)\frac{B}{2}\right],\end{aligned}$$

where we use the fact $\Delta = \left\lfloor \frac{T-2B}{m} \right\rfloor \leq (2\mathrm{e}^{\frac{1}{4}} - 2)B + 1$. The optimal hypothesis f_i^* is

$$f_i^* = \sum_{t=1}^{2B} a_t \kappa(\mathbf{x}_t, \cdot) + \sum_{r=1}^{\Delta+1}(a_{s_r}\kappa(\mathbf{x}_{s_r}, \cdot) + a_{s_r+1}\kappa(\mathbf{x}_{s_r+1}, \cdot)).$$

According to the analysis in **Case 1**, we have $\sum_{t=1}^{T}\ell_t(f_i^*(\mathbf{x}_t), y_t) = 0$, and

$$\|f_i^*\|_{\mathcal{H}_i} = \frac{\sqrt{2B + 2\Delta + 2}}{\sqrt{1-c_i}} \leq \frac{\sqrt{2B + 2(2\mathrm{e}^{\frac{1}{4}} - 2)B + 4}}{\sqrt{1-c_i}} \leq \frac{\sqrt{3.2B}}{\sqrt{1-c_i}},$$

where we omit the constant 4 in the square root. A lower bound on the expected cumulative expected loss of any budgeted online kernel selection algorithm is as follows,

$$\begin{aligned}\sum_{t=1}^{T}\mathbb{E}[\ell(f_t(\mathbf{x}_t), y_t)] &\geq \frac{1}{2}\left[T - (2\mathrm{e}^{\frac{1}{4}} - 2)B - 1 - (m-2)\frac{B}{2}\right] \\ &\geq \frac{1}{2}\left[T - \left\lceil\frac{T-2B}{(2\mathrm{e}^{\frac{1}{4}} - 2)B + 1}\right\rceil\frac{B}{2} + \left(\frac{5}{2} - 2\mathrm{e}^{\frac{1}{4}}\right)B - 1\right] \quad (m \text{ is even}) \\ &\geq \frac{1}{2}\left[T - \frac{T-2B}{(2\mathrm{e}^{\frac{1}{4}} - 2)B + 1}\frac{B_i}{2} + (2 - 2\mathrm{e}^{\frac{1}{4}})B - 1\right].\end{aligned}$$

We can verify that

$$\frac{T-2B}{(2\mathrm{e}^{\frac{1}{4}} - 2)B + 1}\frac{B}{2} - (2 - 2\mathrm{e}^{\frac{1}{4}})B + 1 < \frac{11}{12}T.$$

Thus the expected regret can be lower bounded as follows

$$\mathbb{E}\left[\sum_{t=1}^{T}\ell(f_t(\mathbf{x}_t), y_t)\right] - \sum_{t=1}^{T}\ell(f_i^*(\mathbf{x}_t), y_t) \geq \frac{\sqrt{1-c_i}}{12}\|f_i^*\|_{\mathcal{H}_i}L\frac{T}{\sqrt{3.2B}},$$

Combining with the two cases gives the desired lower bound. □

Proof of Theorem 2

Before giving the detailed proof, we state an important lemma.

Lemma 1 (Bernstein's inequality for martingales)

Let $X_1, \ldots, X_n$ be a bounded martingale difference with respect to the filtration $\mathcal{F} = (\mathcal{F}_i)_{1 \le i \le n}$ and with $|X_i| \le a$. Let $S_i = \sum_{j=1}^{i} X_j$ be the associated martingale. Denote the sum of the conditional variances by

$$\Sigma_n^2 = \sum_{t=1}^{n} \mathbb{E}\left[X_t^2 | \mathcal{F}_{t-1}\right] \le v.$$

Then for all constants $a, v > 0$, with probability at least $1 - \delta$,

$$\max_{i=1,\ldots,n} S_i < \frac{2}{3} a \ln \frac{1}{\delta} + \sqrt{2v \ln \frac{1}{\delta}}.$$

Lemma 1 is derived from Lemma 1.8 in (Cesa-Bianchi and Lugosi 2006).

Proof First, assuming that $B < T$. In this case, there exists a $\upsilon \in [0, 1)$ such that $B < (1 - \upsilon) T^{(1-\upsilon)}$. Thus $\mathbb{P}[\rho_t = 1] = \frac{B}{(1-\upsilon)T^{(1-\upsilon)}(|E_t|+1)^{\upsilon}}$. After the $(T-1)$-th round, the number of support vectors in S satisfies $|S| = \sum_{t=1}^{T-1} \mathbb{I}_{\rho_t = 1}$. Define a random variable X_t as follows

$$X_t = \mathbb{I}_{\rho_t = 1} - \mathbb{P}[\rho_t = 1].$$

Under the condition of $\rho_1, \ldots, \rho_{t-1}$, it can be verified that $\mathbb{E}[X_t] = 0$ and $|X_t| \le 1$. Thus $X_1, \ldots, X_{T-1}$ forms bounded martingale sequence. The sum of conditional variances satisfies

$$\begin{aligned}\Sigma_T^2 = \sum_{t=1}^{T-1} \mathbb{E}[(X_t)^2] \le \sum_{t \in E_T} \mathbb{P}[\rho_t = 1] \le \sum_{t \in E_T} \frac{B}{(1-\upsilon)T^{1-\upsilon}(|E_t|+1)^{\upsilon}} \\ \le \frac{B}{T^{1-\upsilon}} \int_{t \in E_T} \frac{1}{(1-\upsilon)t^{\upsilon}} \mathrm{d}\, t \le \frac{B}{T^{1-\upsilon}} T^{1-\upsilon} \le B.\end{aligned}$$

Using Lemma 1, with probability at least $1 - \delta$,

$$|S| \le B + \frac{2}{3} \ln \frac{1}{\delta} + \sqrt{2B \ln \frac{1}{\delta}}.$$

Then we consider $B = T$. In this case, there is no υ satisfying $B < (1 - \upsilon) T^{1-\upsilon}$. Thus $\mathbb{P}[\rho_t = 1] = 1, t \in E_t$. We have $|S| \le T = B$. Combining with the two cases concludes the proof. □

Proof of Theorem 3

Proof Let $\mathbf{r} \in \Delta_{K-1}$. We consider the regret w.r.t. any $f \in \mathcal{H}_{\kappa_\mathbf{r}}$. We split the regret into two components,

$$\begin{aligned}\sum_{t=1}^{T} \ell(f_t(\mathbf{x}_t), y_t) - \sum_{t=1}^{T} \ell(f(\mathbf{x}_t), y_t) &\le \sum_{t=1}^{T} \ell'(f_t(\mathrm{x}_t), y_t) \cdot (f_t(\mathbf{x}_t) - f(\mathbf{x}_t)) \\ &= \sum_{t=1}^{T} \ell'(f_t(\mathrm{x}_t), y_t) \cdot (f_t(\mathbf{x}_t) - f_{t,i}(\mathbf{x}_t)) + \sum_{t=1}^{T} \ell'(f_t(\mathrm{x}_t), y_t) \cdot (f_{t,i}(\mathbf{x}_t) - f(\mathbf{x}_t)) \\ &= \max\{\ell_m, 1\} \sum_{t=1}^{T} \left[\langle \mathbf{p}_t, c_t\rangle - c_{t,i}\right] + \underbrace{\sum_{t=1}^{T} \ell'(f_t(\mathrm{x}_t), y_t) \cdot (f_{t,i}(\mathbf{x}_t) - f(\mathbf{x}_t))}_{\Xi_2},\end{aligned}$$

where the last inequality is derived from (6). According to Theorem 2.2 in Cesa-Bianchi and Lugosi (2006), let $\eta = \sqrt{8\ln(K)/T}$, the first term can be rewritten as follows,

$$\max\{\ell_m, 1\} \sum_{t=1}^{T} [\langle \mathbf{p}_t, c_t\rangle - c_{t,i}] = O\Big(\max\{\ell_m, 1\}\sqrt{T \ln K}\Big).$$

Next we analyze Ξ_2. Recalling that any $f \in \mathcal{H}_{\kappa_\mathbf{r}}$ can be represented as follows

$$f = \sum_{t=1}^{T} a_t \phi_{\kappa_\mathbf{r}}(\mathbf{x}_t) = \sum_{t=1}^{T} a_t (\sqrt{r_1}\phi_{\kappa_1}^\top(\mathbf{x}_t), \ldots, \sqrt{r_K}\phi_{\kappa_K}^\top(\mathbf{x}_t))^\top = (\sqrt{r_1}f_1, \ldots, \sqrt{r_K}f_K)^\top,$$

where $f_i = \sum_{t=1}^{T} \alpha_t \phi_{\kappa_i}^\top(\mathbf{x}_t)$. Thus Ξ_2 can be rewritten as follows,

$$\Xi_2 := \sum_{t=1}^{T}\sum_{i=1}^{K} r_i \langle \tilde{\nabla}_{t,i}, f_{t,i} - f_i\rangle + \sum_{t=1}^{T}\sum_{i=1}^{K} r_i \langle \nabla_{t,i} - \tilde{\nabla}_{t,i}, f_{t,i} - f_i\rangle.$$

If $B < T$, then using the standard analysis technique of online gradient descent and a constant learning rate, i.e. $\lambda_t = \lambda$ yields

$$\begin{aligned}\mathbb{E}\left[\Xi_2\right] = \mathbb{E}\left[\sum_{t=1}^{T} \langle \tilde{\nabla}_{t,i}, f_{t,i} - f_i\rangle\right] &\le \frac{\|f\|_{\mathcal{H}_{\kappa_\mathbf{r}}}^2}{2\lambda} + \frac{\lambda}{2}\mathbb{E}\left[\sum_{t=1}^{T} \|\tilde{\nabla}_{t,i}\|_{\mathcal{H}_i}^2\right] \\ &\le \frac{\|f\|_{\mathcal{H}_{\kappa_\mathbf{r}}}^2}{2\lambda} + \frac{\lambda D L^2}{2} \cdot \sum_{t \in E_T} \frac{(1-\upsilon)T^{1-\upsilon}(|E_t|+1)^\upsilon}{B} \\ &\le \frac{\|f\|_{\mathcal{H}_{\kappa_\mathbf{r}}}^2}{2\lambda} + \frac{\lambda D L^2}{2}\frac{(1-\upsilon)T^2}{(1+\upsilon)B} \le (\|f\|_{\mathcal{H}_{\kappa_\mathbf{r}}}^2 + 1)\frac{L\sqrt{(1-\upsilon)DT}}{\sqrt{(1+\upsilon)B}},\end{aligned}$$

where $\lambda = \sqrt{(1+\upsilon)B}/(\sqrt{(1-\upsilon)DLT})$.

If $B = T$, which implies $\mathbb{P}[\rho_t = 1] = 1$ for $t \in E_t$, then

$$\Xi_2 = \sum_{t=1}^{T}\sum_{i=1}^{K} r_i\langle \nabla_{t,i}, f_{t,i} - f_i\rangle \le \frac{\|f\|^2_{\mathcal{H}_{\kappa_{\mathbf{r}}}}}{2\lambda} + \frac{\lambda D L^2}{2} T \le (\|f\|^2_{\mathcal{H}_{\kappa_{\mathbf{r}}}} + 1) L\sqrt{DT},$$

where $\lambda = 1/(\sqrt{DT}L)$. Let $\mathbf{r}$ satisfy $r_i = 1$. Combining with Ξ_1 and Ξ_2 yields

$$\mathbb{E}\big[\mathrm{Reg}(\mathcal{H}_i)\big] \le O\left(\max\{\ell_m, 1\}\sqrt{T\ln K} + (\|f\|^2_{\mathcal{H}_i} + 1) L\sqrt{D}\max\{\frac{T}{\sqrt{B}}, \sqrt{T}\}\right).$$

Replacing f with f_i^* and $B = \alpha\mathcal{T}$ concludes the proof. □

Proof of Theorem 5

Proof According to the analysis in (Bubeck et al. 2019), the probability updating of $\mathcal{E}(K')$ is equivalent to the following online mirror descent

$$\bar{\mathbf{p}}_{t+1} = \arg\min_{\mathbf{p}\in\mathbb{R}^{K'}} \big\{\langle \mathbf{p}, c_t\rangle + \mathcal{D}_{\psi_t}(\mathbf{p}, \mathbf{p}_t)\big\}, \qquad \mathbf{p}_{t+1} = \arg\min_{\mathbf{p}\in\Delta_{K'-1}} \mathcal{D}_{\psi_t}(\mathbf{p}, \bar{\mathbf{p}}_{t+1})$$

where $\psi_t(\mathbf{p}) = \frac{1}{\eta_t}\sum_{k=1}^{K'} g(U_{h^*(k)_2}, D_{h^*(k)_1}) p_k \ln p_k$ is the weighted negative entropy regularizer, and $\mathcal{D}_{\psi_t}(\mathbf{u}, \mathbf{v}) = \psi_t(\mathbf{u}) - \psi_t(\mathbf{v}) - \langle \nabla\psi_t(\mathbf{v}), \mathbf{u} - \mathbf{v}\rangle$ is Bregman divergence. Let $\mathbf{u} \in \Delta_{K'-1}$. The expected regret w.r.t. any competitor $\mathbf{u} \in \Delta_{K'-1}$ is as follows

$$\begin{aligned}
\sum_{t=1}^{T}\langle \mathbf{p}_t - \mathbf{u}, c_t\rangle &= \sum_{t=1}^{T}\langle \bar{\mathbf{p}}_{t+1} - \mathbf{u}, c_t\rangle + \sum_{t=1}^{T}\langle \mathbf{p}_t - \bar{\mathbf{p}}_{t+1}, c_t\rangle \\
&\le \sum_{t=1}^{T}\big[\mathcal{D}_{\psi_t}(\mathbf{u}, \mathbf{p}_t) - \mathcal{D}_{\psi_t}(\mathbf{u}, \bar{\mathbf{p}}_{t+1}) - \mathcal{D}_{\psi_t}(\bar{\mathbf{p}}_{t+1}, \mathbf{p}_t)\big] + \sum_{t=1}^{T}\langle \mathbf{p}_t - \bar{\mathbf{p}}_{t+1}, c_t\rangle \\
&\le \sum_{t=1}^{T}\big[\mathcal{D}_{\psi_t}(\mathbf{u}, \mathbf{p}_t) - \mathcal{D}_{\psi_t}(\mathbf{u}, \mathbf{p}_{t+1}) - \mathcal{D}_{\psi_t}(\bar{\mathbf{p}}_{t+1}, \mathbf{p}_t)\big] + \sum_{t=1}^{T}\langle \mathbf{p}_t - \bar{\mathbf{p}}_{t+1}, c_t\rangle \\
&= \mathcal{D}_{\psi_1}(\mathbf{u}, \mathbf{p}_1) + \sum_{t=1}^{T}\big[\langle \mathbf{p}_t - \bar{\mathbf{p}}_{t+1}, c_t\rangle - \mathcal{D}_{\psi}(\bar{\mathbf{p}}_{t+1}, \mathbf{p}_t)\big],
\end{aligned}$$

where we use a constant learning rate i.e, $\eta_t = \eta$. Next we separately analyze the two terms.

The first derivative of the regularizer w.r.t. p_k is

$$\nabla_k\psi(\mathbf{p}) = \frac{1}{\eta} g(U_{h^*(k)_2}, D_{h^*(k)_1})(\ln p_k + 1), \quad k = 1, \ldots, K'.$$

The Bregman divergence between any $\mathbf{u}, \mathbf{v} \in \Delta_{K'-1}$ is

$$\mathcal{D}_{\psi}(\mathbf{u}, \mathbf{v}) = \frac{1}{\eta}\sum_{k=1}^{K'} g(U_{h^*(k)_2}, D_{h^*(k)_1})\left[u_k \ln\frac{u_k}{v_k} - (u_k - v_k)\right].$$

Thus the first term can be rewritten as follows

$$\mathcal{D}_{\psi}(\mathbf{u}, \mathbf{p}_1) = \frac{1}{\eta} \sum_{k=1}^{K'} g(U_{h^*(k)_2}, D_{h^*(k)_1}) \left[u_k \ln \frac{u_k}{p_{1,k}} - u_k + p_{1,k} \right].$$

Next we analyze the second term. We use the updating rule of $\mathcal{E}(K')$ (see Algorithm 3).

$$\begin{aligned}
&\sum_{t=1}^{T} [\langle \mathbf{p}_t - \bar{\mathbf{p}}_{t+1}, c_t \rangle - \mathcal{D}_{\psi_t}(\bar{\mathbf{p}}_{t+1}, \mathbf{p}_t)] \\
&= \sum_{t=1}^{T} \left[\langle \mathbf{p}_t - \bar{\mathbf{p}}_{t+1}, c_t \rangle - \frac{1}{\eta} \sum_{k=1}^{K'} g(U_{h^*(k)_2}, D_{h^*(k)_1}) \left[\bar{p}_{t+1,k} \ln \frac{\bar{p}_{t+1,k}}{p_{t,k}} - \bar{p}_{t+1,i} + p_{t,k} \right] \right] \\
&= \sum_{t=1}^{T} \left[\langle \mathbf{p}_t - \bar{\mathbf{p}}_{t+1}, c_t \rangle - \frac{1}{\eta} \sum_{k=1}^{K'} g(U_{h^*(k)_2}, D_{h^*(k)_1}) \left[-\frac{\eta \bar{p}_{t+1,k} c_{t,k}}{g(U_{h^*(k)_2}), D_{h^*(k)_1}} - \bar{p}_{t+1,k} + p_{t,k} \right] \right] \\
&= \sum_{t=1}^{T} \left[\langle \mathbf{p}_t, c_t \rangle + \frac{1}{\eta} \sum_{k=1}^{K'} g(U_{h^*(k)_2}, D_{h^*(k)_1}) \left[p_{t,k} \exp\left(-\eta \frac{c_{t,k}}{g(U_{h^*(k)_2})} \right) - p_{t,k} \right] \right] \\
&\le \sum_{t=1}^{T} \left[\langle \mathbf{p}_t, c_t \rangle + \frac{1}{\eta} \sum_{k=1}^{K'} g(U_{h^*(k)_2}, D_{h^*(k)_1}) p_{t,k} \left[-\frac{\eta c_{t,k}}{g(U_{h^*(k)_2}, D_{h^*(k)_1})} + \frac{\eta^2 c_{t,k}^2/2}{g^2(U_{h^*(k)_2}, D_{h^*(k)_1})} \right] \right] \\
&\le \frac{\eta}{2} \sum_{t=1}^{T} \sum_{k=1}^{K'} p_{t,k} \frac{c_{t,k}^2}{g(U_{h^*(k)_2}, D_{h^*(k)_1})} \le \frac{\eta}{2} \sum_{t=1}^{T} \langle \mathbf{p}_t, c_t \rangle.
\end{aligned} \tag{16}$$

where we use the fact $\exp(-x) \le 1 - x + \frac{x^2}{2}$ for $x \ge 0$ and the definition of $\bar{p}_{t+1,i}$. Combining with the two terms, we obtain

$$\sum_{t=1}^{T} \langle \mathbf{p}_t - \mathbf{u}, c_t \rangle \le \frac{\eta}{2} \sum_{t=1}^{T} \langle \mathbf{p}_t, c_t \rangle + \frac{1}{\eta} \sum_{k=1}^{K'} g(U_{h^*(k)_2}, D_{h^*(k)_1}) \left[u_k \ln \frac{u_k}{p_{1,k}} - u_k + p_{1,k} \right].$$

Denote $A_{\min} = \{k_{\min} \in [K'], k_{\min} = \operatorname{argmin}_{k \in [K']} g(U_{h^*(k)_2}, D_{h^*(k)_1}, Y)\}$. Let the initial distribution $\mathbf{p}_1$ satisfy $p_{1,k} = (1 - \frac{1}{U\sqrt{T}}) \frac{1}{|A|} + \frac{1}{K'U\sqrt{T}}$ for $k \in A_{\min}$, and $p_{1,k} = \frac{1}{K'U\sqrt{T}}$ for $k \notin A_{\min}$. We compare with the i-th action. Let $u_i = 1$ and $u_k = 0$ for $k \ne i$. Then we have

$$\begin{aligned}
\sum_{t=1}^{T} \langle \mathbf{p}_t, c_t \rangle \le \frac{1}{1 - \frac{\eta}{2}} \Bigg[&\sum_{t=1}^{T} c_{t,i} + \frac{g(U_{h^*(i)_2}, D_{h^*(i)_1}, Y) \ln(K'U\sqrt{T})}{\eta} \\
&+ \sum_{k \notin A_{\min}} \frac{g(U_{h^*(k)_2}, D_{h^*(k)_1}, Y)}{\eta K'U\sqrt{T}} + \left(1 - \frac{1}{U\sqrt{T}} + \frac{|A_{\min}|}{K'U\sqrt{T}} \right) \frac{1}{\eta} \min_{k \in [K']} g(U_{h^*(k)_2}, D_{h^*(k)_1}, Y) \Bigg].
\end{aligned}$$

Let $C_{T,i} := \sum_{t=1}^{T} c_{t,i}$. Subtracting $C_{T,i}$ on both sides yields

$$\begin{aligned}\sum_{t=1}^{T}\langle \mathbf{p}_t, c_t\rangle - C_{T,i} \leq & \frac{\eta C_{T,i}}{2-\eta} + \frac{2g(U_{h^*(i)_2}, D_{h^*(i)_1}, Y)\ln(K'T)}{(2-\eta)\eta} + \frac{2g_{\max}}{(2-\eta)\eta U\sqrt{T}} + \frac{2g_{\min}}{(2-\eta)\eta} \\ \leq & 2g(U_{h^*(i)_2}, D_{h^*(i)_1}, Y)\sqrt{2T\ln(K'T)} + \frac{\sqrt{2}g_{\max}}{U\sqrt{\ln(K'T)}} + \frac{g_{\min}\sqrt{2T}}{\sqrt{\ln(K'T)}} \\ = & O\Big(g(U_{h^*(i)_2}, D_{h^*(i)_1}, Y)\sqrt{T\ln(K'T)}\Big), \qquad (\eta = \sqrt{2\ln(K'T)/T} < 1)\end{aligned}$$

where $g_{\min} = \min_{k\in[K']} g(U_{h^*(k)_2}, D_{h^*(k)_1}, Y)$ and $g_{\max} = \max_{k\in[K']} g(U_{h^*(k)_2}, D_{h^*(k)_1}, Y)$. Using Assumption 5, we have $g_{\max} = \max_{k\in[K']} g(U_{h^*(k)_2}, D_{h^*(k)_1}, Y) = \Theta(\max_j U_j + 1)$. Besides, $\max_j U_j = U = \Theta(\sqrt{B})$ and $B \leq T$ (see Assumption 4) and $g_{\min} = U_1 = \Theta(U/\sqrt{T})$. Omitting the lower order terms, we complete the proof. □

Proof of Theorem 6

Proof For any $f \in \mathbb{H}_i$, let $\mathbb{H}_{i,j}$ be the smallest hypothesis space that contains f. If $j = 1$, then $\|f\|_{\mathcal{H}_i} \leq U_1$. Otherwise, we have $\mathrm{e}^{-1}U_j < \|f\|_{\mathcal{H}_i} \leq U_j$. We analyze the regret w.r.t. f.

$$\begin{aligned}\mathbb{E}[\mathrm{Reg}(\mathcal{H}_i)] = & \mathbb{E}\left[\sum_{t=1}^{T}\left[\sum_{k=1}^{K'} p_{t,k}\ell(f_{t,h^*(k)_1,h^*(k)_2}(\mathbf{x}_t), y_t) - \ell(f_{t,i,j}(\mathbf{x}_t), y_t)\right]\right] \\ & + \mathbb{E}\left[\sum_{t=1}^{T}\ell(f_{t,i,j}(\mathbf{x}_t), y_t)\right] - \sum_{t=1}^{T}\ell(f(\mathbf{x}_t), y_t) \\ = & \mathbb{E}\left[\sum_{t=1}^{T}[\langle \mathbf{p}_t, c_t\rangle - c_{t,h(i,j)}]\right] + \mathbb{E}\left[\sum_{t=1}^{T}[\ell(f_{t,i,j}(\mathbf{x}_t), y_t) - \ell(f(\mathbf{x}_t), y_t)]\right] \\ = & \underbrace{O\Big(g(U_j, D_i, Y)\sqrt{T\ln(K'T)}\Big)}_{\Xi_1} + \underbrace{\mathbb{E}\sum_{t=1}^{T}[\ell(f_{t,i,j}(\mathbf{x}_t), y_t) - \ell(f(\mathbf{x}_t), y_t)]}_{\Xi_2},\end{aligned} \tag{17}$$

where Ξ_1 comes from Theorem 5. Next we analyze Ξ_2.

Using the convexity of loss function, we have

$$\Xi_2 \leq \sum_{t=1}^{T}\langle \tilde{\nabla}_{t,i,j}, f_{t,i,j} - f\rangle + \sum_{t=1}^{T}\langle \nabla_{t,i,j} - \tilde{\nabla}_{t,i,j}, f_{t,i,j} - f\rangle.$$

Let $\lambda_{t,i,i} = \lambda_{i,j}$. Using the property of projection, we have

$$\left\|f_{t+1,i,j} - f\right\|_{\mathcal{H}_i}^2 \leq \left\|\bar{f}_{t+1,i,j} - f\right\|_{\mathcal{H}_i}^2 = \left\|f_{t,i,j} - \lambda_{i,j}\tilde{\nabla}_{t,i,j} - f\right\|_{\mathcal{H}_i}^2,$$

Rearranging terms and summing over $t = 1, \ldots, T$ yields

$$\sum_{t=1}^{T}\langle \tilde{\nabla}_{t,i,j}, f_{t,i,j} - f\rangle \leq \sum_{t=1}^{T}\frac{\|f_{t,i,j} - f\|_{\mathcal{H}_i}^2 - \|f_{t+1,i,j} - f\|_{\mathcal{H}_i}^2}{2\lambda_{i,j}} + \sum_{t=1}^{T}\frac{\lambda_{i,j}}{2}\|\tilde{\nabla}_{t,i,j}\|_{\mathcal{H}}^2.$$

Let $\mathbb{E}_t$ be the condition expectation w.r.t. ρ_t. Taking expectation w.r.t. $\{\rho_t\}_{t=1}^T$ yields

$$\begin{aligned}\mathbb{E}\big[\Xi_2\big] =&\mathbb{E}\left[\sum_{t=1}^{T}\langle\tilde{\nabla}_{t,i,j}, f_{t,i,j}-f\rangle\right]+\sum_{t=1}^{T}\mathbb{E}\big[\langle\nabla_{t,i,j}-\mathbb{E}_t[\tilde{\nabla}_{t,i,j}], f_{t,i,j}-f\rangle\big]\\ =&\frac{\|f\|_{\mathcal{H}_i}^2}{2\lambda_{i,j}}+\frac{\lambda_{i,j}}{2}\sum_{t=1}^{T}\mathbb{E}_t\big[\|\tilde{\nabla}_{t,i,j}\|_{\mathcal{H}}^2\big]\\ \le&\frac{U_j^2}{2\lambda_{i,j}}+\frac{\lambda_{i,j}}{2}\sum_{t=1}^{T}\frac{L^2D_i}{B}2(1-\upsilon)T^{1-\upsilon}t^{\upsilon}\\ \le&\frac{U_j^2}{2\lambda_{i,j}}+\lambda_{i,j}L^2D_i\frac{(1-\upsilon)T^2}{(1+\upsilon)B}\le U_jL\frac{\sqrt{2(1-\upsilon)D_iT}}{\sqrt{(1+\upsilon)B}},\end{aligned}$$

where we set $\lambda_{i,j}=\frac{U_j\sqrt{(1+\upsilon)B}}{\sqrt{2(1-\upsilon)D_iLT}}$. Next we further consider two cases: (i) $j>1$, (ii) $j=1$.

- *Case (i)* $j>1$
  Using the fact $\mathrm{e}^{-1}U_j\le\|f\|_{\mathcal{H}_i}\le U_j$, we have $\mathbb{E}[\Xi_2]\le\mathrm{e}\|f\|_{\mathcal{H}_i}LT\sqrt{\frac{2D_i}{B}}$.
- *Case (ii)* $j=1$
  Recalling that $U_{\min}=U/\sqrt{T}$ and $U=\Theta(B)$. Then $U_1\le\mathrm{e}\sqrt{B/T}$ (see (7)), and we obtain $\mathbb{E}[\Xi_2]\le\mathrm{e}L\sqrt{2D_iT}$.

Combining with the results of Case (i) and Case (ii), we obtain,

$$\mathbb{E}\big[\Xi_2\big]=O\left(\|f\|_{\mathcal{H}_i}LT\sqrt{D_i}\frac{1}{\sqrt{B}}+L\sqrt{D_iT}\right).$$

Next we show the final regret. Using Assumption 5, we can rewrite Ξ_1 as follows

$$\begin{aligned}\Xi_1=&O\Big(g(U_j,D_i,Y)\sqrt{T\ln(K'T)}\Big)=O\Big((U_j+1)\sqrt{T\ln(K'T)}\Big)\\ =&O\Big(\|f\|_{\mathcal{H}_i}L\sqrt{T\ln(K'T)}+\sqrt{T\ln(K'T)}\Big).\end{aligned}$$

Combining with Ξ_2 and Ξ_1 yields

$$\mathbb{E}\big[\mathrm{Reg}(\mathbb{H}_{i,j})\big]=\mathbb{E}\big[\Xi_2\big]+\Xi_1=O\left(\|f\|_{\mathcal{H}_i}L\frac{T}{\sqrt{B}}+\sqrt{T\ln(K'T)}\right),$$

where $K'=K(\lceil\ln U\rceil-\lceil\ln(U/\sqrt{T})\rceil+1)$.

According to Assumption 4, we have $B\le T$. If $B=T$, then the expected regret becomes

$$\mathbb{E}\big[\mathrm{Reg}(\mathbb{H}_{i,j})\big]=O\Big(\|f\|_{\mathcal{H}_i}L\sqrt{T}+\sqrt{T\ln(K'T)}\Big).$$

Combining with the two cases concludes the proof. □

Proof of Theorem 7

Proof The proof is same with that of Theorem 1. Thus we omit the details. For a static resource allocation $\mathcal{R}(\mathcal{T}_1, \ldots, \mathcal{T}_K)$, let $j^* = \max_{j\in[K]} \mathcal{T}_j$. According to Assumption 3, we have $B_{j^*} = \beta\mathcal{T}_{j^*}$. We also choose K Gaussian kernel functions $\kappa_i(\mathbf{x}, \mathbf{z}) = \exp(-\frac{\|\mathbf{x}-\mathbf{z}\|_2^2}{\sigma_i})$, $i = 1, \ldots, K$ as the candidates. The strategy that the adversary sends examples to the learner is same with that in the proof of Theorem 1, except that we replace B with B_{j^*}. Therefore, for all κ_i, the expected regret of any budgeted online kernel selection algorithm satisfies

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell(f_t(\mathbf{x}_t), y_t)\right] - \sum_{t=1}^{T} \ell(f_i^*(\mathbf{x}_t), y_t) = \begin{cases} \Omega\left(\left\|f_i^*\right\|_{\mathcal{H}_i} L\sqrt{T}\right) & \text{if} \quad T = O(B_{j^*}), \\ \Omega\left(\left\|f_i^*\right\|_{\mathcal{H}_i} L\frac{T}{\sqrt{B_{j^*}}}\right) & \text{otherwise}, \end{cases}$$

which recoveries the desired result. □

Proof of Theorem 8

Proof First, assuming that $B < \frac{2T}{K}$. In this case, there exists υ such that $B < \frac{2(1-\upsilon)T}{K}$. We just consider a fixed $i \in [K]$. After the $(T-1)$-th round, the number of support vectors in S_i is $|S_i| = \sum_{t=1}^{T-1} \mathbb{I}_{\rho_{t,i}=1} \cdot \mathbb{I}_{i=J_t}$. Define a random variable X_t as follows

$$X_t = \mathbb{I}_{\rho_{t,i}=1} \cdot \mathbb{I}_{i=J_t} - \mathbb{P}[\rho_{t,i} = 1] \cdot \mathbb{P}[i = J_t].$$

Under the condition of $(\rho_\tau, J_\tau)_{\tau<t}$, we can obtain $\mathbb{E}_t[X_t] = 0$ and $|X_t| \le 1$. Thus $X_1, \ldots, X_{T-1}$ forms bounded martingale difference. The sum of conditional variances satisfies

$$\Sigma_T^2 = \sum_{t=1}^{T-1} \mathbb{E}_t[(X_t)^2] \le \sum_{t=1}^{T-1} \mathbb{P}[\rho_{t,i} = 1] \cdot \mathbb{P}[i = J_t] \le \sum_{t\in E_{T,i}} \frac{KB}{2(1-\upsilon)T^{1-\upsilon}t^{\upsilon}} \cdot \frac{1}{K} \le \frac{B}{2},$$

where $E_{T,i} = \{t < T, \nabla_{t,i} \neq 0\}$. Using Lemma 1, with probability at least $1-\delta$,

$$|S_i| \le \frac{B}{2} + \frac{2}{3}\ln\frac{1}{\delta} + \sqrt{B\ln\frac{1}{\delta}}.$$

Then we consider $\frac{2T}{K} \le B \le T$. In this case, there is no υ satisfying $B < \frac{2(1-\upsilon)T}{K}$. Thus $\mathbb{P}[\rho_{t,i} = 1] = 1$ for $t \in E_{T,i}$. The same proof technique yields, with probability at least $1-\delta$,

$$\begin{aligned} |S_i| &\le \frac{T}{K} + \frac{2}{3}\ln\frac{1}{\delta} + \sqrt{\frac{2T}{K}\ln\frac{1}{\delta}} \\ &\le \frac{B}{2} + \frac{2}{3}\ln\frac{1}{\delta} + \sqrt{B\ln\frac{1}{\delta}}. \end{aligned}$$

Combining with the two cases and using the union of events bound to $i = 1, \ldots, K$ concludes the proof. □

Proof of Theorem 9

Proof Some of analysis is same with that of Theorem 5. We start with (16). Replacing $g(U_{h^*(k)_2}, D_{h^*(k)_1}, Y)$ with 1 yields

$$\sum_{t=1}^{T}[\langle \mathbf{p}_t - \bar{\mathbf{p}}_{t+1}, \tilde{c}_t\rangle - \mathcal{D}_{\psi_t}(\bar{\mathbf{p}}_{t+1}, \mathbf{p}_t)] \le \frac{\eta}{2}\sum_{t=1}^{T}\sum_{i=1}^{K} p_{t,i}\tilde{c}_{t,i}^2 \le \frac{K\eta}{2}\sum_{t=1}^{T}\langle \mathbf{p}_t, \tilde{c}_t\rangle,$$

in which we use the fact $\tilde{c}_{t,i} = \frac{c_{t,i}}{\mathbb{P}[i\in\{I_t,J_t\}]}\mathbb{I}_{i\in\{I_t,J_t\}} \le Kc_{t,i}$. Combining with $\mathcal{D}_{\psi}(\mathbf{u}, \mathbf{p}_1)$ yields

$$\sum_{t=1}^{T}\langle \mathbf{p}_t - \mathbf{u}, \tilde{c}_t\rangle \le \frac{K\eta}{2}\sum_{t=1}^{T}\langle \mathbf{p}_t, \tilde{c}_t\rangle + \frac{1}{\eta}\sum_{i=1}^{K}\left[u_i \ln\frac{u_i}{p_{1,i}} - u_i + p_{1,i}\right].$$

Let the initial distribution $\mathbf{p}_1$ satisfy $p_{1,i} = \frac{1}{K}$ for all $i = 1, \ldots, K$. We compare with the i-th action. Let $u_i = 1$ and $u_k = 0$ for $k \neq i$. Then we have

$$\sum_{t=1}^{T}\langle \mathbf{p}_t, \tilde{c}_t\rangle \le \frac{1}{1-\frac{K\eta}{2}}\left[\sum_{t=1}^{T}\tilde{c}_{t,i} + \frac{\ln K}{\eta}\right].$$

Now we replace i with $i^* = \text{argmin}_{i\in[K]}\sum_{t=1}^{T}\tilde{c}_{t,i}$. For simplicity, let $\sum_{t=1}^{T} c_{t,i^*} = \tilde{C}_{T,*}$. Subtracting $\tilde{C}_{T,*}$ on both sides yields

$$\sum_{t=1}^{T}[\langle \mathbf{p}_t, \tilde{c}_t\rangle - \tilde{c}_{t,i^*}] \le \frac{K\eta}{2-K\eta}\tilde{C}_{T,*} + \frac{2\ln K}{(2-K\eta)\eta} \le 2\sqrt{2\tilde{C}_{T,*}K\ln(K)},$$

where $\eta = \min\{\sqrt{2\ln K/(K\tilde{C}_{T,*})}, \frac{1}{K}\}$. Thus, for any $i \in [K]$, we have

$$\sum_{t=1}^{T}[\langle \mathbf{p}_t, \tilde{c}_t\rangle - \tilde{c}_{t,i}] \le \sum_{t=1}^{T}[\langle \mathbf{p}_t, \tilde{c}_t\rangle - \tilde{c}_{t,i^*}] \le 2\sqrt{2\tilde{C}_{T,*}K\ln(K)} \le 2\sqrt{2\tilde{C}_{T,i}K\ln(K)}.$$

Taking expectation yields the desired result. □

Proof of Theorem 10

The proof is similar with that of Theorem 6. We also consider two cases: Case 1: $B < \frac{2T}{K}$ and Case 2: $\frac{2T}{K} \le B \le T$.

Case 1 $B < \frac{2T}{K}$

We analyze the regret w.r.t. f. Recalling the regret decomposition (17),

$$\mathbb{E}\left[\text{Reg}(\mathbb{H}_l)\right] = \underbrace{g(U,D)\cdot\mathbb{E}\left[\sum_{t=1}^{T}\left[\langle \mathbf{p}_t, c_t\rangle - c_{t,i}\right]\right]}_{\Xi_1} + \underbrace{\mathbb{E}\sum_{t=1}^{T}[\ell(f_{t,i}(\mathbf{x}_t), y_t) - \ell(f(\mathbf{x}_t), y_t)]}_{\Xi_2}.$$

Next we separately analyze Ξ_1 and Ξ_2. Similarly with the proof of Theorem 6, we have

$$\Xi_2 \leq \sum_{t=1}^{T} \frac{\|f_{t,i} - f\|_{\mathcal{H}_i}^2 - \|f_{t+1,i} - f\|_{\mathcal{H}_i}^2}{2\lambda_i} + \sum_{t=1}^{T} \frac{\lambda_i}{2} \|\tilde{\nabla}_{t,i}\|_{\mathcal{H}}^2 + \sum_{t=1}^{T} \langle \nabla_{t,i} - \tilde{\nabla}_{t,i}, f_{t,i} - f \rangle.$$

Let $\mathbb{E}_t$ be the conditional expectation w.r.t. J_t and ρ_{t,J_t}. Taking expectation w.r.t. $\{J_\tau, \rho_{\tau,J_\tau}\}_{\tau=1}^{T}$, we can obtain

$$\begin{aligned}
\mathbb{E}\big[\Xi_2\big] \leq & \frac{\|f\|_{\mathcal{H}_i}^2}{2\lambda_i} + \sum_{t=1}^{T} \frac{\lambda_i}{2} \mathbb{E}\big[\mathbb{E}_t \|\tilde{\nabla}_{t,i}\|_{\mathcal{H}}^2\big] + \sum_{t=1}^{T} \mathbb{E}\big[\langle \nabla_{t,i} - \mathbb{E}_t \tilde{\nabla}_{t,i}, f_{t,i} - f \rangle\big] \\
\leq & \frac{\|f\|_{\mathcal{H}_i}^2}{2\lambda_i} + \frac{\lambda_i}{2} \sum_{t=1}^{T} \frac{L^2 D_i}{\mathbb{P}[i = J_t] \cdot \mathbb{P}[\rho_{t,i} = 1]} \\
= & \frac{\|f\|_{\mathcal{H}_i}^2}{2\lambda_{i,j}} + \frac{\lambda_{i,j}}{2} \sum_{t=1}^{T} \frac{L^2 D_i}{B} 2(1-\upsilon) T^{1-\upsilon} t^{\upsilon} = O\left(\left(\|f\|_{\mathcal{H}_i}^2 + 1 \right) \frac{L\sqrt{D_i} T}{\sqrt{B}} \right),
\end{aligned}$$

where we set $\lambda_{i,j} = \frac{\sqrt{(1+\upsilon)B}}{\sqrt{2(1-\upsilon)D_i LT}}$. Next we give the final regret.

Using Theorem 9 and the fact $\mathbb{E}\left[\sum_{t=1}^{T} \ell(f_{t,i}(\mathbf{x}_t), y_t)\right] \leq \sum_{t=1}^{T} \ell(f(\mathbf{x}_t), y_t) + \mathbb{E}[\Xi_2]$, we have

$$\Xi_1 = O\left(\sqrt{g(U, D)\big[L_T(f) + \mathbb{E}[\Xi_2]\big] K \ln \frac{K}{\delta}} \right).$$

where $L_T(f) := \sum_{t=1}^{T} \ell(f(\mathbf{x}_t), y_t)$. Using Assumption 5, we have,

$$\Xi_1 = O\left(\sqrt{(U+1) L_T(f) K \ln \frac{K}{\delta}} + \sqrt{(U+1)\mathbb{E}[\Xi_2] K \ln \frac{K}{\delta}} \right).$$

Combining with Ξ_1 and Ξ_2 yields

$$\mathbb{E}\big[\mathrm{Reg}(\mathbb{H}_i)\big] = O\left(\sqrt{(U+1) L_T(f) K \ln K} + (\|f\|_{\mathcal{H}_i}^2 + 1) L\sqrt{D_i} \frac{T}{\sqrt{B}} \right).$$

Replacing f with f_i^* yields the desired result.

Case 2 $\frac{2T}{K} \leq B \leq T$

In this case, $\mathbb{P}[\rho_{t,i} = 1] = 1$ for $i = J_t$.

$$\mathbb{E}\big[\Xi_2\big] \leq \frac{\|f\|_{\mathcal{H}_i}^2}{2\lambda_i} + \frac{\lambda_i}{2} \sum_{t=1}^{T} \frac{L^2 D_i}{\mathbb{P}[i = J_t]} = \frac{\|f\|_{\mathcal{H}_i}^2}{2\lambda_i} + \frac{\lambda_i}{2} L^2 D_i K T = O\Big(\big(\|f\|_{\mathcal{H}_i}^2 + 1 \big) L\sqrt{D_i T K} \Big),$$

where we set $\lambda_{i,j} = \frac{1}{\sqrt{K D_i T L}}$. Combining with Ξ_1 and Ξ_2, we obtain the regret,

$$\mathbb{E}\big[\mathrm{Reg}(\mathbb{H}_i)\big] = O\Big(\sqrt{(U+1) L_T(f) K \ln K} + (\|f\|_{\mathcal{H}_i}^2 + 1) L\sqrt{D_i T K} \Big).$$

Combining with the results of Case 1 and Case 2, we conclude the proof.

Author Contributions The two authors have the same contributions to the study conception and design. The first draft of the manuscript was written by [Junfan Li] and the second author commented on previous versions of the manuscript. The two authors read and approved the final manuscript.

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

Availability of data and material All data and materials as well as custom code support our claims and comply with field standards.

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

Consent for publication Not applicable.

Ethical approval Not applicable.

Code availability custom code.

Consent to participate Not applicable.

References

Agarwal, A., Duchi, J.C., Bartlett, P.L., & Levrard, C. (2011). Oracle inequalities for computationally budgeted model selection. In Proceedings of the 24th Annual Conference on Learning Theory (pp. 69–86).

Agarwal, A., Luo, H., Neyshabur, B., & Schapire, R.E. (2017). Corralling a band of bandit algorithms. In Proceedings of the 30th Annual Conference on Learning Theory (pp. 12–38).

Bubeck, S., & Cesa-Bianchi, N. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends®in Machine Learning, 5*(1), 1–122.

Bubeck, S., Devanur, N. R., Huang, Z., & Niazadeh, R. (2019). Multi-scale online learning: Theory and applications to online auctions and pricing. *Journal of Machine Learning Research, 20*(62), 1–37.

Cesa-Bianchi, N., & Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge University Press.

Cesa-Bianchi, N., Mansour, Y., & Shamir, O. (2015). On the complexity of learning with kernels. In Proceedings of the 28th Annual Conference on Learning Theory (pp. 297–325).

Crammer, K., Kandola, J. S., & Singer, Y. (2003). Online classification on a budget. *Advances in Neural Information Processing Systems, 16*, 225–232.

Cutkosky, A., & Boahen, K. (2016). Online convex optimization with unconstrained domains and losses. *Advances in Neural Information Processing Systems, 29*, 748–756.

Dekel, O., Shalev-Shwartz, S., & Singer, Y. (2008). The forgetron: A kernel-based perceptron on a budget. *SIAM Journal on Computing, 37*(5), 1342–1372.

Foster, D. J., Kale, S., Mohri, M., & Sridharan, K. (2017). Parameter-free online learning via model selection. *Advances in Neural Information Processing Systems, 30*, 6022–6032.

Foster, D. J., Krishnamurthy, A., & Luo, H. (2019). Model selection for contextual bandits. *Advances in Neural Information Processing Systems, 32*, 14741–14752.

Hoi, S. C. H., Jin, R., Zhao, P., & Yang, T. (2013). Online multiple kernel classification. *Machine Learning, 90*(2), 289–316.

Jézéquel, R., Gaillard, P., & Rudi, A. (2019). Efficient online learning with kernels for adversarial large scale problems. *Advances in Neural Information Processing Systems, 32*, 9427–9436.

Jin, R., Hoi, S.C.H., & Yang, T. (2010). Online multiple kernel learning: Algorithms and mistake bounds. In Proceedings of the 21st International Conference on Algorithmic Learning Theory (pp. 390–404)

Kivinen, J., Smola, A. J., & Williamson, R. C. (2004). Online learning with kernels. *IEEE Transactions on Signal Processing, 52*(8), 2165–2176.

Koppel, A., Warnell, G., Stump, E., & Ribeiro, A. (2019). Parsimonious online learning with kernels via sparse projections in function space. *Journal of Machine Learning Research, 20*(3), 1–44.

Kothari, P.K., & Livni, R. (2020). On the expressive power of kernel methods and the efficiency of kernel learning by association schemes. In Proceedings of the31st International Conferences on Algorithmic Learning Theory (pp 422–450).

Lu, J., Hoi, S. C. H., Wang, J., Zhao, P., & Liu, Z. (2016). Large scale online kernel learning. *Journal of Machine Learning Research, 17*(47), 1–43.

McMahan, B., & Abernethy, J. (2013). Minimax optimal algorithms for unconstrained linear optimization. *Advances in Neural Information Processing Systems, 26*, 2724–2732.

McMahan, H.B., & Orabona, F. (2014). Unconstrained online linear learning in hilbert spaces: Minimax algorithms and normal approximations. In Proceedings of The 27th Conference on Learning Theory (pp. 1020–1039).

Muthukumar, V., Ray, M., Sahai, A., & Bartlett, P. (2019). 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).

Nguyen, T.D., Le, T., Bui, H., & Phung, D. (2017). Large-scale online kernel learning with random feature reparameterization. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (pp. 2543–2549).

Orabona, F. (2013). Dimension-free exponentiated gradient. *Advances in Neural Information Processing Systems, 26*, 1806–1814.

Orabona, F., Keshet, J., & Caputo, B. (2009). Bounded kernel-based online learning. *Journal of Machine Learning Research, 10*, 2643–2666.

Sahoo, D., Hoi, S.C.H., & Li, B. (2014). Online multiple kernel regression. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD (pp. 293–302).

Seldin, Y., Bartlett, P.L., Crammer, K., & Abbasi-Yadkori, Y. (2014). Prediction with limited advice and multiarmed bandits with paid observations. In Proceedings of the 31st International Conference on Machine Learning (pp. 280–287).

Wang, Z., Crammer, K., & Vucetic, S. (2012). Breaking the curse of kernelization: Budgeted stochastic gradient descent for large-scale SVM training. *Journal of Machine Learning Research, 13*(1), 3103–3131.

Yang, T., Mahdavi, M., Jin, R., Yi, J., & Hoi, S.C.H. (2012). Online kernel selection: Algorithms and evaluations. In Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (pp 1197–1202).

Zhang, L., Yi, J., Jin, R., Lin, M., & He, X. (2013). Online kernel learning with a near optimal sparsity bound. In Proceedings of the 30th International Conference on Machine Learning (pp. 621–629).

Zhang, X., & Liao, S. (2018). Online kernel selection via incremental sketched kernel alignment. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (pp. 3118–3124).

Zhang, X., & Liao, S. (2020). Hypothesis sketching for online kernel selection in continuous kernel space. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (pp. 2498–2504).

Zhao, P., Wang, J., Wu, P., Jin, R., & Hoi, SCH. (2012). 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).