Active Learning With Multiple Kernels

Songnam Hong, *Member, IEEE*, and Jeongmin Chae, *Student Member, IEEE*

***Abstract*— Online multiple kernel learning (OMKL) has provided an attractive performance in nonlinear function learning tasks. Leveraging a random feature (RF) approximation, the major drawback of OMKL, known as the curse of dimensionality, has been recently alleviated. These advantages enable RF-based OMKL to be considered in practice. In this article, we introduce a new research problem, named stream-based *active MKL* (AMKL), in which a learner is allowed to label some selected data from an oracle according to a selection criterion. This is necessary for many real-world applications as acquiring a true label is costly or time consuming. We theoretically prove that the proposed AMKL achieves an optimal sublinear regret $\mathcal{O}(\sqrt{T})$ as in OMKL with little labeled data, implying that the proposed selection criterion indeed avoids unnecessary label requests. Furthermore, we present AMKL with an adaptive kernel selection (named AMKL-AKS) in which irrelevant kernels can be excluded from a kernel dictionary "on the fly." This approach improves the efficiency of active learning and the accuracy of function learning. Via numerical tests with real data sets, we verify the superiority of AMKL-AKS, yielding a similar accuracy performance with OMKL counterpart using a fewer number of labeled data.**

***Index Terms*— Active learning (AL), multiple kernel learning (MKL), online learning, reproducing kernel Hilbert space (RKHS).**

I. Introduction

LEARNING a nonlinear function is of great interest in various machine learning tasks, such as classification, regression, clustering, dimensionality reduction, and reinforcement learning [1]–[4]. In particular, supervised functional learning tasks, which are closely related to the subject of this article, are formulated as follows. Given data $\{(\mathbf{x}_t, y_t) : t = 1, \ldots, T\}$ with features $\mathbf{x}_t \in \mathbb{R}^d$ and labels $y_t \in \mathbb{R}$, the objective of a function learning is to learn (or estimate) a function $f : \mathbb{R}^d \to \mathbb{R}$ which minimizes the accumulate loss $(1)/(T)\sum_{t=1}^{T} \mathcal{L}(f(\mathbf{x}_t), y_t)$, where $f(\mathbf{x}_t)$ and $\mathcal{L}(\cdot, \cdot)$ represent an estimated label and a loss function, respectively. This challenging problem can be tractable with the restriction that $f(\cdot)$ belongs to a well-structured function class [e.g., reproducing kernel Hilbert space (RKHS)] [1]. The accuracy of the kernel-based learning fully relies on a preselected kernel, which is chosen manually either by task-specific *a priori* knowledge or by some intensive cross-validation process. Multiple kernel learning (MKL), using a predetermined set of kernels (called a kernel dictionary), is more powerful. This is because it can enable a data-driven kernel selection from a given dictionary, i.e., a linear or nonlinear combination of multiple kernels is optimized as the part of a learning process [2], [5]–[8].

In many real-world applications, functional learning tasks are expected to be performed in an *online* fashion. For example, online learning is required when data arrive sequentially, such as online spam detection [9] and time series prediction [10], and when a large number of data makes, it impossible to carry out data analytic in batch form [11]. For such cases, online MKL (OMKL) has been proposed, which seeks the optimal combination of pools of single-kernel functions in an online fashion. Two popular methods to learn the best kernel combination are called the Hedge algorithm and the online gradient descent (OGD) algorithm [12], [13]. It was shown in [11], [12], and [14] that OMKL can provide superior accuracy and enjoy great flexibility compared with single-kernel online learning. In contrast, OMKL generally suffers from a high computational complexity as the dimension of optimization variables grow with time (i.e., the number of data T) [2], [15]. Recently in [14], this problem has been alleviated by applying a random feature (RF) approximation [16] to OMKL. In the resulting method, called RF-based OMKL (a.k.a., Raker), the dimension of the optimization variables can be determined irrespective of the number of data. Another advantage of RF-based OMKL is that function learning can be solved using the powerful toolboxes from online convex optimization and online learning developed under vector spaces [14]. More related works of OMKL have been well-summarized in [14].

Unlabeled data may be abundant but acquiring labels is difficult, time-consuming, or expensive, in particular, when only experts whose time is precious can generate reliable labels [17]–[19]. One motivating example is the labeling process of medical data [e.g., magnetic resonance imaging (MRI) data]. In this case, label acquisition requires the data analysis by a well-trained expert, such as an electroencephalographer, a cardiologist, or a medical imaging expert. In addition, it would need an invasive or expensive medical procedure (e.g., an angiogram). Because of them, the labeling of medical data would be expensive and time consuming. Active learning (AL), a subfield of machine learning, aims at overcoming the labeling bottleneck by allowing the learner to actively decide whether or not to acquire the label of an incoming data from the oracle (e.g., a human annotator,

Manuscript received May 6, 2020; revised August 25, 2020 and November 26, 2020; accepted December 25, 2020. Date of publication January 14, 2021; date of current version July 7, 2022. This work was supported by the National Research Foundation of Korea (NRF) Grant, Korea government (MSIT), under Grant NRF-2020R1A2C1099836. *(Corresponding author: Songnam Hong.)*

Songnam Hong is with the Department of Electronic Engineering, Hanyang University, Seoul 04763, South Korea (e-mail: snhong@hanyang.ac.kr).

Jeongmin Chae is with the Department of Electrical Engineering, University of Southern California, CA 90089 USA (e-mail: chaej@usc.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2020.3047953.

Digital Object Identifier 10.1109/TNNLS.2020.3047953

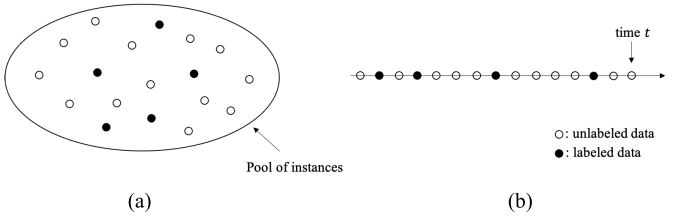


Fig. 1. Description of two types of AL. (a) Pool-based setting and (b) Stream-based setting.

such as a medical expert) [20]. From AL (or sampling), the desired accuracy of functional learning can be attained using little labeled data. Motivated by this, we introduce a new research problem to construct an AL framework for (RF-based) OMKL. To the best of our knowledge, this problem has not been investigated yet, regardless of its necessity in various real-world applications.

A. Related Works

Many AL approaches have been proposed in the literature [20]–[24]. According to the different ways of incoming unlabeled data, AL can be categorized into *pool-based* AL [21], [22] and *stream-based* AL [20], [23], [24], as shown in Fig. 1. In pool-based AL, a pool of unlabeled data is given and the goal is to optimally choose some data to label so that a learned function from them can generate the best label for the remaining data. Whereas, in stream-based AL, each unlabeled data is drawn one at a time from the data source and the learner must decide whether to query or discard it [20]. The latter is the subject of this article as it is most relevant to OMKL frameworks. The stream-based AL has been investigated in several functional learning tasks, such as speech tagging [25], sensor scheduling [26], information retrieval [27], drifting streaming data [28], and expert advice [29]. However, under the (RF-based) OMKL framework, none of the above-mentioned methods yield a performance guarantee, which is the main subject of this article.

B. Contributions

We propose a novel stream-based AL framework suitable for OMKL. The proposed method is referred to as *active MKL* (AMKL). Focusing on streaming (or sequential) data, at every time, a learner in AMKL decides whether to query or discard an incoming data according to a selection criterion. On the other hand, all the incoming data in OMKL are assumed to be labeled. We contribute to the subject in the following ways.

1) We propose a *selection criterion* with an analytic performance guarantee. The proposed selection criterion ensures that skipping the labeling only causes a η_c-bounded loss, where $\eta_c > 0$ can control the tradeoff of AL efficiency and function-learning accuracy.
2) Theoretically, it is proved that AMKL with $\eta_c = \mathcal{O}(1/\sqrt{T})$ achieves an *optimal* sublinear regret as in OMKL, implying that the proposed selection criterion indeed avoids unnecessary label requests.
3) In addition, we present AMKL with adaptive kernel selection (termed AMKL-AKS). The proposed AKS can exclude irrelevant kernels from a kernel dictionary "on the fly," in which they are determined on the basis of accumulated losses (i.e., kernel reliabilities). AMKL-AKS can considerably enhance AL efficiency as the selection criterion in this method is evaluated only with the refined kernels having accurate estimates. In contrast, the irrelevant kernels in AMKL can produce inaccurate estimates, which makes the evaluation of the selection criterion imprecise, regardless of the informativeness of labeling.
4) Via numerical tests with real data sets, we demonstrate that AMKL-AKS achieves almost the same performance with RF-based OMKL (a.k.a., Raker), using a fewer number of labeled data. Therefore, AMKL-AKS can provide an elegant accuracy–efficiency tradeoff.

Finally, we provide some discussions about the proposed AMKL-AKS. Regarding the hyper-parameter η_c, it is proved in Section IV that $\eta_c = \mathcal{O}(1/\sqrt{T})$ is asymptotically optimal, in the sense of achieving the best AL efficiency (denoted by $\mathrm{AL}_{\mathrm{eff}}$) by maintaining the learning accuracy of OMKL counterpart. Throughout this article, AL efficiency is formally defined as

$$\mathrm{AL}_{\mathrm{eff}} \stackrel{\Delta}{=} (\text{number of labeled data})/T. \tag{1}$$

In nonasymptotic cases, however, the performance of AMKL-AKS can be further enhanced by carefully optimizing the parameter η_c in a data-dependent way. For example, a time-varying $\eta_c(t)$ can be optimized 'on the fly' as similarly in [14] and [30]–[32]. Such hyper-parameter optimization is beyond the scope of this article and is left for interesting future work. For the numerical tests of this article, $\eta_c = 0.0005$ is used for all the data sets without the data-dependent optimization. This value is chosen from the theoretical analysis of η_c-bounded loss and with the assumption that about 10^{-4} loss is acceptable. We next emphasize that the proposed AKS can enhance the accuracy performance of OMKL as using a large kernel dictionary may deteriorate the accuracy of a function learning or cause a slower convergence to an optimal function if too many irrelevant kernels are included. In addition, it is a randomized algorithm that can provide robustness to potential adversarial attacks. Via martingale argument, we theoretically prove that both AMKL-AKS and OMKL-AKS (i.e., AMKL-AKS with $\mathrm{AL}_{\mathrm{eff}} = 1$) also achieve the optimal sublinear regret with high probability.

C. Outline and Notations

The remainder of this article is organized as follows. In Section II, we briefly review RF-based MKL which is the underlying method of the proposed algorithms. In Section III, we describe the proposed AL frameworks, named AMKL and AMKL-AKS. Regret analysis is provided in Section IV to verify the asymptotic optimality of the proposed methods. In Section V, beyond the asymptotic analysis, we verify the superiority of the proposed AMKL-AKS via numerical tests with real data sets. Some concluding remarks are provided in Section VI.

Notations: Bold lowercase letters will denote column vectors. For any vector $\mathbf{x}$, $\mathbf{x}^{\mathsf{T}}$ stands for the transpose of $\mathbf{x}$ and $\|\mathbf{x}\|$

denotes the ℓ_2-norm of $\mathbf{x}$. $\mathbb{E}[\cdot]$ denotes the expectation and $\langle\cdot,\cdot\rangle$ denotes the inner product in Euclidean space. To simplify notations, we let $[N] \triangleq \{1, 2, \ldots, N\}$ for any positive integer N.

II. Preliminaries

We briefly review MKL based on RF approximation as it is the baseline method for the proposed AMKL and AMKL-AKS. Given the training data $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)\}$, where $\mathbf{x}_t \in \mathcal{X} \subseteq \mathbb{R}^d$ and $y_t \in \mathcal{Y} \subseteq \mathbb{R}$, the objective is to learn a (nonlinear) function $f : \mathcal{X} \to \mathcal{Y}$ which minimizes the accumulate loss $(1)/(T)\sum_{t=1}^{T}\mathcal{L}(f(\mathbf{x}_t), y_t)$, where $f(\mathbf{x}_t)$ and $\mathcal{L}(\cdot,\cdot)$ represent an estimated label and a loss function, respectively. In kernel-based learning [7], [8], [33], it is assumed that a target function $f(\mathbf{x})$ belongs to a reproducing Hilbert kernel space (RKHS), defined as $\mathcal{H} \triangleq \{f : f(\mathbf{x}) = \sum_{t=1}^{\infty}\alpha_t\kappa(\mathbf{x}, \mathbf{x}_t)\}$, where $\kappa(\mathbf{x}, \mathbf{x}_t) : \mathcal{X} \times \mathcal{X} \to \mathcal{Y}$ is a symmetric positive semidefinite basis function (called kernel), which measures the similarity between $\mathbf{x}$ and $\mathbf{x}_t$. Among various kernels, one representative example is the Gaussian kernel with a parameter σ^2, given as

$$\kappa(\mathbf{x}, \mathbf{x}_t) = \exp(-\|\mathbf{x} - \mathbf{x}_t\|^2/2\sigma^2). \tag{2}$$

In addition, a kernel is said to be reproducing if

$$\langle\kappa(\mathbf{x}, \mathbf{x}_t), \kappa(\mathbf{x}, \mathbf{x}_{t'})\rangle_{\mathcal{H}} = \kappa(\mathbf{x}_t, \mathbf{x}_{t'}) \tag{3}$$

where $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ denotes an inner product defined in the Hilbert space $\mathcal{H}$. The associated RKHS norm is defined as $\|f\|_{\mathcal{H}}^2 \triangleq \sum_t\sum_{t'}\alpha_t\alpha_{t'}\kappa(\mathbf{x}_t, \mathbf{x}_{t'})$. The function learning problem over RKHS can be formulated as

$$\min_{f\in\mathcal{H}}\frac{1}{T}\sum_{t=1}^{T}\mathcal{L}(f(\mathbf{x}_t), y_t). \tag{4}$$

We remark that loss function can be chosen in a task-specific way, e.g., least-square cost for regression and logistic cost for classification. Especially when the number of data is finite (e.g., T training data), the representer theorem in [15] shows that the optimal solution of (4) is represented as

$$\hat{f}(\mathbf{x}) = \sum_{t=1}^{T}\alpha_t\kappa(\mathbf{x}, \mathbf{x}_t). \tag{5}$$

The major drawback of this approach is the curse of dimensionality as the number of parameters α_t's (to be optimized) grows with the number of data T.

In [16], it has been addressed by introducing RF approximation for kernels. As in [16], the kernel κ is assumed to be shift-invariant, i.e., $\kappa(\mathbf{x}_t, \mathbf{x}_{t'}) = \kappa(\mathbf{x}_t - \mathbf{x}_{t'})$ for any $t, t' \in [T]$. Note that Gaussian, Laplacian, and Cauchy kernels satisfy the shift-invariance [16]. For $\kappa(\mathbf{x}_t - \mathbf{x}_{t'})$ absolutely integrable, its Fourier transform $\pi_k(\mathbf{v})$ exists and represents the power spectral density. In addition, when $\kappa(\mathbf{0}) = 1$ it can also be viewed as a probability density function (PDF). For a Gaussian kernel in (2), we have $\pi_\kappa(\mathbf{v}) = \mathcal{N}(0, \sigma^{-2}\mathbf{I})$. Then, the kernel function can be rewritten as $\kappa(\mathbf{x}_t - \mathbf{x}_{t'}) = \mathbb{E}[\exp(j\mathbf{v}^{\mathsf{T}}(\mathbf{x}_t - \mathbf{x}_{t'}))]$. Having a sufficient number of independent and identically distributed (i.i.d.) samples $\{\mathbf{v}_i : i \in [D]\}$ from $\pi_\kappa(\mathbf{v})$, $\kappa(\mathbf{x}_t - \mathbf{x}_{t'})$ can be well-approximated by the sample mean such as

$$\kappa(\mathbf{x}_t - \mathbf{x}_{t'}) \approx \frac{1}{D}\sum_{i=1}^{D}\mathrm{Re}\big(\exp\big(j\mathbf{v}_i^{\mathsf{T}}(\mathbf{x}_t - \mathbf{x}_{t'})\big)\big) \tag{6}$$

where $\mathrm{Re}(a)$ denotes the real part of a complex value a. Clearly, the accuracy of this approximation grows as the number of samples D increases. In numerical tests, a proper D will be chosen by considering the accuracy-complexity tradeoff. The approximation in (6) can be rewritten as a vector form $\kappa(\mathbf{x}_t - \mathbf{x}_{t'}) = \mathbf{z}^{\mathsf{T}}(\mathbf{x}_t)\mathbf{z}(\mathbf{x}_{t'})$, where

$$\mathbf{z}(\mathbf{x}) = \frac{1}{\sqrt{D}}\big[\sin\mathbf{v}_1^{\mathsf{T}}\mathbf{x}, \ldots, \sin\mathbf{v}_D^{\mathsf{T}}(\mathbf{x}), \cos\mathbf{v}_1^{\mathsf{T}}\mathbf{x}, \ldots, \cos\mathbf{v}_D^{\mathsf{T}}\mathbf{x}\big]^{\mathsf{T}}. \tag{7}$$

Based on this, the optimal solution $\hat{f}(\mathbf{x})$ in (5) can be well-approximated as

$$\hat{f}(\mathbf{x}) = \sum_{t=1}^{T}\alpha_t\mathbf{z}^{\mathsf{T}}(\mathbf{x}_t)\mathbf{z}(\mathbf{x}) \triangleq \hat{\boldsymbol{\theta}}^{\mathsf{T}}\mathbf{z}(\mathbf{x}) \tag{8}$$

where the optimization variable $\hat{\boldsymbol{\theta}}$ is a $2D$-vector. Note that its dimension $2D$ can be determined irrespective of the number of data T.

RF-based kernel learning can be naturally extended into MKL framework, where a target function is formed as a linear (or convex) combination of multiple preselected kernels $\{\kappa_i : i \in [P]\}$. From [34], the function approximation can be represented as

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{P}\hat{p}_i\hat{f}_i(\mathbf{x}) \in \bar{\mathcal{H}} \tag{9}$$

where $\bar{\mathcal{H}} \triangleq \mathcal{H}_1 \bigotimes \mathcal{H}_2 \bigotimes \cdots \bigotimes \mathcal{H}_P$ and $\hat{f}_i(\mathbf{x}) \in \mathcal{H}_i$ which is an RKHS induced by the kernel κ_i, and $\hat{p}_i \in [0, 1]$ denotes the combination weight of the associated kernel function $\hat{f}_i$. In addition, under RF approximation, the kernel functions in (9) can be further simplified as $\hat{f}_i(\mathbf{x}) = \hat{\boldsymbol{\theta}}_i^{\mathsf{T}}\mathbf{z}_i(\mathbf{x})$ for $i \in [P]$, where $\mathbf{z}_i(\mathbf{x})$ is defined in (7) with D number of i.i.d. samples from $\pi_{\kappa_i}(\mathbf{v})$. Assuming RF-based MKL, kernel functions in the above are considered in the following section.

III. Methods

We first define the problem setting of OMKL framework. The main purpose of OMKL is to learn a sequence of functions $\hat{f}_{t+1}(\mathbf{x})$, $t \in [T]$, in an *online fashion*: at each time t, a function $\hat{f}_{t+1} : \mathcal{X} \times \mathcal{Y}$ is estimated from the observed training data $\{(\mathbf{x}_\tau, y_\tau) : \tau \in [t]\}$, where $\mathbf{x}_\tau \in \mathcal{X} \subseteq \mathbb{R}^d$ and $y_\tau \in \mathcal{Y} \subseteq \mathbb{R}$ represent the feature and the label, respectively. To evaluate the accuracy of learned functions, we let $\mathcal{L} : \mathcal{Y}\times\mathcal{Y} \to \mathbb{R}$ be a loss function. Throughout this article, it is assumed that there are P kernels in a kernel dictionary. Then, OMKL framework consists of the following two steps.

1) *Local Step:* Each kernel function $\hat{f}_{t+1,i}(\mathbf{x})$ is optimized independently of the other kernel functions.
2) *Global Step:* The learner seeks the best function approximation $\hat{f}_{t+1}(\mathbf{x})$ by combining the kernel functions

$\{\hat{f}_{t+1,i}(\mathbf{x}), i \in [P]\}$ with proper weights $\{\hat{p}_{t+1}(i), i \in [P]\}$:

$$\hat{f}_{t+1}(\mathbf{x}) = \sum_{i=1}^{P} \hat{p}_{t+1}(i)\hat{f}_{t+1,i}(\mathbf{x}). \tag{10}$$

Then, the objective of OMKL is to optimize the local functions $\{\hat{f}_{t+1,i}(\mathbf{x}) : i \in [P]\}$ and the weights $\{\hat{p}_{t+1}(i), i \in [P]\}$ such that the following (cumulative) *regret* is minimized:

$$\text{regret}_T = \sum_{t=1}^{T} \mathcal{L}(\hat{f}_t(\mathbf{x}_t), y_t) - \min_{1\le i\le P} \sum_{t=1}^{T} \mathcal{L}(f_i^\star(\mathbf{x}_t), y_t)$$

where the regret compares the cumulative loss of the learner to the cumulative loss of the best kernel in hindsight. Note that $f_i^\star(\cdot)$ denotes the best function in each kernel $\mathcal{H}_i$.

In the following subsections, RF-based OMKL is assumed as the baseline method of the proposed AMKL framework, due to the advantages of attractive learning accuracy and scalability [14]. Accordingly, a learned function $\hat{f}_t(\mathbf{x})$ is assumed to have the form of $\hat{f}_{t,i}(\mathbf{x}) = \hat{\boldsymbol{\theta}}_{t,i}^{\mathsf{T}}\mathbf{z}_i(\mathbf{x})$, where $\mathbf{z}_i(\mathbf{x})$ is defined in (7) with D number of i.i.d. samples from $\pi_{\kappa_i}(\mathbf{v})$. It is remarkable that the proposed AMKL framework can be directly applied to OMKL frameworks without RF approximation in [11] and [12]. In Section III-A, we propose a novel stream-based AL (termed AMKL) suitable for RF-based OMKL framework. We then improve AMKL in Section III-B by introducing an AKS. The resulting method is named AMKL-AKS.

A. Proposed AL Framework

We propose AMKL in which the label of an incoming data is revealed only when the learner has made a request to acquire the label from an oracle. Whereas, in OMKL, the label of every incoming data is always revealed to the learner. The active labeling in AMKL can be necessary for many real-world applications as the label acquisition is expensive or time consuming. Our major contribution is to construct an AMKL algorithm that can achieve almost the same accuracy as the OMKL counterpart with a fewer number of labeled data. Namely, it can reduce the labeling cost of OMKL without sacrificing the learning accuracy. In this extension, the key challenge is to decide when the learner should or should not request the label of incoming data to the oracle. This decision can be efficiently made by introducing a *selection criterion*. To be specific, the proposed AMKL performs in the following way: the learner skips the label request for an incoming data if the selection criterion is satisfied; otherwise, the learner directly follows the OMKL algorithm (e.g., RF-based OMKL (a.k.a., Raker) in [14]). Definitely, the selection criterion plays a crucial role in determining the AL efficiency and learning accuracy of the proposed AMKL, where AL efficiency is defined in (1). We, in this article, propose a new selection criterion by leveraging the structure of MKL. In addition, in Section IV, we theoretically prove that the proposed selection criterion indeed avoids unnecessary label requests, meaning that AMKL achieves the same asymptotic performance

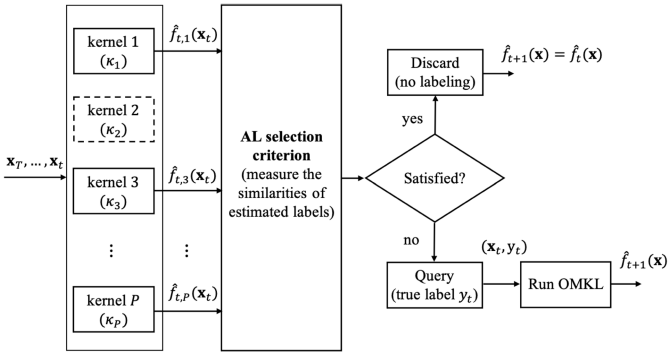


Fig. 2. Description of the proposed stream-based AL. The solid-line and dashed-line boxes denote the *active* and *inactive* kernels, respectively.

of OMKL counterpart with little labeled data. We further improve AMKL by introducing an AKS in which irrelevant kernels are excluded from a kernel dictionary "on the fly." Specifically, this can improve AL efficiency considerably as the selection criterion, which is evaluated only with refined kernels, generates a precise output. Without using AKS, some irrelevant kernels may produce inaccurate outputs, which makes the output of the selection criterion imprecise, regardless of the informativeness of labeling. The proposed AMKL with AKS is referred to as AMKL-AKS. In the remaining part of this section, we will provide the detailed procedures of AMKL-AKS, as shown in Fig. 2.

We first introduce a binary sequence $\{a_t \in \{0, 1\} : t \in [T]\}$ to indicate whether the label y_t is revealed or not, i.e., $a_t = 1$ if the learner requested the label of an incoming data $\mathbf{x}_t$ (i.e., the selection criterion is not satisfied), and $a_t = 0$, otherwise. Let $\mathcal{A}_t = \{\tau : a_\tau = 1, \tau \in [t]\}$ denote the index subset of revealed labels until time t. Focusing on time t, the procedures of AMKL-AKS are explained as follows. At time t, the learner observes an incoming *unlabeled* data $\mathbf{x}_t$, and from the previous times, has the knowledge of the following.

1) $\{\hat{f}_{t,i} : i \in [P]\}$: the estimated kernel functions (equivalently, $\{\hat{\boldsymbol{\theta}}_{t,i} : i \in [P]\}$).
2) $\mathcal{A}_{t-1}$: the index subset of revealed labels until time $t-1$.
3) the losses $\{\mathcal{L}(\hat{f}_{\tau,i}(\mathbf{x}_\tau), y_\tau) : \tau \in \mathcal{A}_{t-1}\}$.
4) the kernel subset $\mathcal{V}_{s_t} \subseteq [P]$ containing the indices of active kernels at time t.

Differently from OMKL [14], losses in AMKL-AKS are defined only when the labels are revealed. The construction of $\mathcal{V}_{s_t} \subseteq [P]$ will be explained in Section III-B. As in OMKL [12], [14], the choice of the weight vector (i.e., the weight distribution) $\hat{\mathbf{p}}_t = (\hat{p}_t(1), \ldots, \hat{p}_t(P))^{\mathsf{T}}$ in (10) should be well-optimized. Following the online learning framework [13], we optimize the weight vector using the so-called *exponential strategy* (EXP strategy). In this strategy, the weights are determined on the basis of the past losses as

$$\hat{p}_t(i) = \frac{\hat{w}_t(i)}{\sum_{i=1}^{P} \hat{w}_t(i)} \tag{11}$$

where the initial values are $\hat{w}_1(i) = 1$ and

$$\hat{w}_t(i) = \exp\left(-\eta_g \sum_{\tau\in\mathcal{A}_{t-1}} \mathcal{L}(\hat{f}_{i,\tau}(\mathbf{x}_\tau), y_\tau)\right) \tag{12}$$

for some parameter $\eta_g > 0$.

We are now ready to describe the procedures of AMKL-AKS, which consists of three steps: active labeling, active local, and active global.

1) Active Labeling Step: This step decides whether or not to acquire the label of an incoming data $\mathbf{x}_t$ from the oracle. The decision is quickly made by the proposed selection criterion. To define the selection criterion, we first introduce a *confidence condition*, such as

$$\max_{j\in\mathcal{V}_{s_t}} \sum_{i\in\mathcal{V}_{s_t}} \hat{p}_t(i)\mathcal{L}(\hat{f}_{t,i}(\mathbf{x}_t), \hat{f}_{t,j}(\mathbf{x}_t)) \le \eta_c \tag{13}$$

for some parameter $\eta_c > 0$. From (13), we observe that the condition is simply evaluated with the up-to-date estimated functions, without knowing the true label y_t. The proposed confidence condition ensures that skipping the labeling only causes a η_c-bounded loss, where the parameter η_c can control the tradeoff of AL efficiency and learning accuracy. The theoretical evidence is provided in Lemma 4. Intuitively, the confidence condition can measure the similarities of the learned kernel functions: if the condition holds, all the active kernels generate the estimates having similar losses with the true label y_t, i.e., acquiring the label y_t has little impact on the updates of the weights (i.e., reliabilities of the kernels). Thus, in this case, it would be better to skip the label-request in terms of the efficiency-accuracy tradeoff. We notice that avoiding label-requests has an also impact on the updates of local kernel functions (see an active local step in the following). However, the confidence condition in (13) only focuses on the updates of weights. In order to take the local updates into account, we introduce the parameter M to ensure sufficient local updates, where M indicates the maximum number of consecutive unlabeling. Then, the proposed AMKL-AKS has the parameters η_c and M. In asymptotic case, it is proved in Section IV that choosing $\eta_c = \mathcal{O}(1/\sqrt{T})$ and M as any constant can achieve an optimal sublinear regret. In the nonasymptotic case, however, the parameters should be carefully determined by considering the tradeoff of AL efficiency and learning accuracy. As noticed in Section I-B, such optimization is beyond the scope of this article. Instead, for numerical tests of this article, the constant values (e.g., $\eta_c = 0.0005$ and $M = 1$) are selected without careful optimization.

Given the parameters η_c and M, the selection criterion of the proposed AMKL-AKS is obtained as follows.

[Selection Criterion] Given $M \ge 1$ and $\eta_c > 0$, the label of an incoming data $\mathbf{x}_t$ is not requested (i.e., $a_t = 0$) only when $\sum_{\tau=1}^{M} a_{t-\tau} \ne 0$ and the confidence condition in (13) is satisfied. For ease of exposition, $M = 0$ is assumed that all the labels are revealed as in OMKL, i.e, $a_t = 1$ for all $t \in [T]$.

From the sequence of indicator variables $\{a_t : t \in [T]\}$, AL efficiency is computed as $\mathrm{AL}_{\mathrm{eff}} = (1)/(T)\sum_{t=1}^{T} a_t$. Unfortunately, this value cannot be obtained theoretically since it is data-dependent. We can only prove the lower bound as a function of M, such as $\mathrm{AL}_{\mathrm{eff}} \ge 1 - (M)/(M+1)$.

2) Active Local Step: This step learns a set of single-kernel functions $\hat{f}_{t+1,i}(\mathbf{x}) \in \mathcal{H}_i$ for $i \in [P]$. When $a_t = 0$ (i.e., the label y_t is not revealed), local functions cannot be updated as, in this case, the loss function $\mathcal{L}(\cdot, y_t)$ is undefined. Namely, we have

$$\hat{f}_{t+1,i}(\mathbf{x}) = \hat{f}_{t,i}(\mathbf{x}) \quad \forall i \in [P], \text{ if } a_t = 0. \tag{14}$$

When $a_t = 1$, the learner can observe the labeled data $(\mathbf{x}_t, y_t)$ and optimize the local functions $\hat{f}_{t+1,i}(\mathbf{x})$ for $i \in [P]$, via online optimization. Following the RF approximation in (8), each kernel function is determined by 2D-vector $\hat{\boldsymbol{\theta}}_{t+1,i}$ as

$$\hat{f}_{t+1,i}(\mathbf{x}) = \hat{\boldsymbol{\theta}}_{t+1,i}^{\mathsf{T}} \mathbf{z}_i(\mathbf{x}) \tag{15}$$

where $\mathbf{z}_i(\mathbf{x})$ is defined in (7). In this article, the parameter vector $\hat{\boldsymbol{\theta}}_{t+1,i}$ is optimized via the well-known OGD [35] as

$$\hat{\boldsymbol{\theta}}_{t+1,i} = \hat{\boldsymbol{\theta}}_{t,i} - \eta_l \nabla \mathcal{L}\big(\hat{\boldsymbol{\theta}}_{t,i}^{\mathsf{T}} \mathbf{z}_i(\mathbf{x}_t), y_t\big) \tag{16}$$

where $\nabla\mathcal{L}(\hat{\boldsymbol{\theta}}_{t,i}^{\mathsf{T}}\mathbf{z}_i(\mathbf{x}_t), y_t)$ denotes the gradient at $\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}_{t,i}$. To sum up, each kernel function i in AMKL-AKS is optimized, such as

$$\hat{\boldsymbol{\theta}}_{t+1,i} = \begin{cases} \hat{\boldsymbol{\theta}}_{t,i}, & \text{if } a_t = 0 \\ \hat{\boldsymbol{\theta}}_{t,i} - \eta_l \nabla \mathcal{L}\big(\hat{\boldsymbol{\theta}}_{t,i}^{\mathsf{T}} \mathbf{z}_i(\mathbf{x}_t, y_t)\big), & \text{if } a_t = 1. \end{cases} \tag{17}$$

3) Active Global Step: This step learns a target function $\hat{f}_{t+1}(\mathbf{x})$ by properly combining the single kernel functions $\{\hat{f}_{t+1,i}(\mathbf{x}) : i \in [P]\}$. This step consists of AKS and function combining.

1) Adaptive kernel selection: This step finds the subset of P kernels (denoted by $\mathcal{V}_{s_{t+1}}$) which contain the kernels having higher accuracy local functions. Toward this, the weights of P kernels are updated as

$$\hat{p}_{t+1}(i) = \frac{\hat{w}_{t+1}(i)}{\sum_{i=1}^{P} \hat{w}_{t+1}(i)} \tag{18}$$

where

$$\hat{w}_{t+1}(i) = \exp\left(-\eta_g \sum_{\tau\in\mathcal{A}_t} a_\tau \mathcal{L}(\hat{f}_{i,\tau}(\mathbf{x}_\tau), y_\tau)\right). \tag{19}$$

Then, the subset $\mathcal{V}_{s_{t+1}} \subset [P]$ is determined on the basis of the updated weights (see Section III-B for detailed procedures).

2) Function combining: Given the selected subset $\mathcal{V}_{s_{t+1}}$, AMKL-ASK learns a target function $\hat{f}_{t+1}(\mathbf{x})$ as

$$\hat{f}_{t+1}(\mathbf{x}) = \sum_{i\in\mathcal{V}_{s_{t+1}}} \hat{q}_{t+1}(i)\hat{f}_{t+1,i}(\mathbf{x}) \tag{20}$$

where the weight distribution is refined as

$$\hat{q}_{t+1}(i) = \frac{\hat{w}_{t+1}(i)}{\sum_{\ell\in\mathcal{V}_{s_{t+1}}} \hat{w}_{t+1}(\ell)}. \tag{21}$$

Then, the learned function can generate the estimated label of incoming data $\mathbf{x}_{t+1}$ as $\hat{y}_{t+1} = \hat{f}_{t+1}(\mathbf{x}_{t+1})$.

Remark 1: We would like to mention that AMKL-AKS can encompass various OMKL algorithms as its special cases. When AKS is not used (i.e., $\mathcal{V}_{s_t} = [P], \forall t \in [T]$), AMKL-AKS is reduced to AMKL. In addition, when all incoming data are labeled (i.e., $\mathrm{AL}_{\mathrm{eff}} = 1$), the resulting algorithm (AMKL-AKS with $\mathrm{AL}_{\mathrm{eff}} = 1$) is named OMKL-AKS since, in this case, active labeling is not used. Finally when $\mathcal{V}_{s_t} = [P], \forall t \in [T]$, and $\mathrm{AL}_{\mathrm{eff}} = 1$, AMKL-AKS is reduced to RF-base OMKL (a.k.a., Raker [14]). Both OMKL-AKS and AMKL-AKS with $\mathrm{AL}_{\mathrm{eff}} = 1$ will be used interchangeably.

B. Proposed AKS

We describe the proposed AKS in the active global step of AMKL-AKS. At every time t, the subset of P kernels, denoted by $\mathcal{V}_{s_{t+1}} \subseteq [P]$ is determined on the basis of the weights (i.e., the accumulated losses) $\hat{p}_t(i), i \in [P]$. Here, the weight $\hat{p}_t(i)$ can be thought of as the accuracy of the local function $\hat{f}_{t,i}$ at time t (i.e., the reliability of the information provided by the kernel i at the current time). As mentioned before, the goal of AKS is to select the kernel functions with higher accuracy since it can improve the AL efficiency and learning accuracy. To explain the proposed AKS, we introduce a design parameter $K_{t+1} \in [P]$ which indicates the number of kernels to be used for the construction of a function $\hat{f}_{t+1}(\mathbf{x})$. One reasonable way is to choose the parameter K_{t+1}, such as

$$K_{t+1} = |\{i \in [P] : \hat{p}_{t+1}(i)/\hat{p}^{\star}_{t+1} > \delta_{t+1}\}| \tag{22}$$

for some parameter $\delta_{t+1} > 0$, where $\hat{p}^{\star}_{t+1} = \max_{j\in[P]} \hat{p}_{t+1}(j)$. This approach will be used for our numerical tests in Section V. We remark that the proposed AMKL-AKS with any choice of K_{t+1} can guarantee the optimal asymptotic performance. Given K_{t+1}, the collection of all size-K_{t+1} subsets of $[P]$ is defined as

$$\Omega(K_{t+1}) = \{\mathcal{V} : \mathcal{V} \subseteq [P], |\mathcal{V}| = K_{t+1}\} \tag{23}$$

where $|\Omega(K_{t+1})| = \binom{P}{K_{t+1}}$. In addition, the $|\Omega(K_{t+1})|$ elements in $\Omega(K_{t+1})$ are denoted as $\{\mathcal{V}_1, \ldots, \mathcal{V}_{|\Omega(K_{t+1})|}\}$. This construction ensures the so-called *uniform frequency* property such that each kernel index occurs uniformly in the collection $\Omega(K_{t+1})$. The corresponding frequency, denoted by J_{t+1}, is computed as

$$J_{t+1} = K_{t+1}\binom{P}{K_{t+1}} \Big/ P \tag{24}$$

since $P \cdot J_{t+1} = |\Omega(K_{t+1})| \cdot K_{t+1}$. By construction, J_{t+1} in (24) should be an integer. In the example of $P = 4$ and $K_{t+1} = 2$, we have

$$\Omega(K_{t+1} = 2) = \{(1,2), (1,3), (1,4), (2,3), (2,4), (3,4)\}$$

where each kernel index occurs exactly $J_{t+1} = 3$ times, thereby satisfying the uniform frequency.

Then, a size-K_{t+1} subset is chosen randomly from $\Omega(K_{t+1})$ according to a certain probability distribution. The specific selection procedure will be explained at the bottom of this section. One may concern the complexity problem to generate all the subsets belonging to $\Omega(K_{t+1})$, especially for a large P. To address this problem, we choose $J_{t+1} = \gamma_{t+1}K_{t+1}$ with a parameter γ_{t+1} such that J_{t+1} is an integer, where γ_{t+1} is chosen by considering the size of the collection. Given J_{t+1} and K_{t+1}, define a collection $\Omega(J_{t+1}, K_{t+1})$ whose size is determined as

$$|\Omega(J_{t+1}, K_{t+1})| \triangleq \lfloor J_{t+1} \cdot P/K_{t+1} \rfloor = \lfloor \gamma_{t+1} P \rfloor \tag{25}$$

where $\lfloor x \rfloor$ denotes a floor function which produces the greatest integer less than or equal to x. Although there might be various methods to construct the elements (i.e., the subsets of $[P]$) of $\Omega(J_{t+1}, K_{t+1})$, the experiments in this article use a simple balls-bins random construction in Remark 2.

Remark 2 (Balls-Bins Construction): Given J_{t+1} and K_{t+1}, the elements of $\Omega(J_{t+1}, K_{t+1})$ are determined via Balls-Bins construction. Here, kernels and subsets (i.e., elements of $\Omega(J_{t+1}, K_{t+1})$) correspond to balls and bins, respectively. Then, there are P balls and $|\Omega(J_{t+1}, K_{t+1})|$ bins. As in well-known balls and bins problem, consider the process of tossing P balls into $|\Omega(J_{t+1}, K_{t+1})|$ bins. The tosses are uniformly at random and independent of each other. Repeat this process J_{t+1} times so that each ball i belongs to J_{t+1} distinct bins. Definitely, each bin contains K_{t+1} balls on average. Once these balls and bins processes are completed, the collection of the corresponding subsets, that is,

$$\Omega(J_{t+1}, K_{t+1}) \triangleq \{\mathcal{V}_i \subseteq [P] : i = 1, \ldots, \lfloor \gamma_{t+1} P \rfloor\} \tag{26}$$

is formed such that $\mathcal{V}_i$ takes the balls' indices belong to the bin i as elements. The notation in (26) can be rewritten as

$$\Omega(K_{t+1}) = \Omega\left(J_{t+1} = \binom{P}{K_{t+1}}, K_{t+1}\right). \tag{27}$$

That is, with the particular choice of J_{t+1}, the above-mentioned collection contains the all subsets of size K_{t+1} as before. In addition, we remark that the proposed collection $\Omega(J_{t+1}, K_{t+1})$ satisfies the uniform frequency, i.e., each kernel occurs exactly J_{t+1} times.

We finally propose a *randomized* algorithm to choose a subset of kernels from $\Omega(J_{t+1}, K_{t+1})$. Define a discrete random variable S_{t+1} with the probability mass function (PMF):

$$\hat{\alpha}_{t+1}(j) = \frac{\sum_{i\in\mathcal{V}_j} \hat{w}_{t+1}(i)}{J_{t+1}\sum_{i=1}^{P} \hat{w}_{t+1}(i)} \tag{28}$$

for $j \in [|\Omega(J_{t+1}, K_{t+1})|]$, where $\hat{w}_{t+1}(i)$ is defined in (12). Due to the uniform frequency (see Remark 2), we can easily verify that (28) is a valid PMF. Letting

$$\hat{\boldsymbol{\alpha}}_{t+1} = (\hat{\alpha}_{t+1}(1), \ldots, \hat{\alpha}_{t+1}(|\Omega(J_{t+1}, K_{t+1})|)). \tag{29}$$

AMKL-AKS chooses a subset in the following way.

1) Sampling S_{t+1} according to $\hat{\boldsymbol{\alpha}}_{t+1}$ in (29). The corresponding sample is denoted as s_{t+1}.
2) Accordingly, the selected subset is denoted as $\mathcal{V}_{s_{t+1}} \in \Omega(J_{t+1}, K_{t+1})$.

IV. Regret Analysis

We analyze the cumulative regrets of the proposed online and AL algorithms. For the regret analysis of this section, the following conditions are assumed.

1) **(a1)** For any fixed $\mathbf{z}_i(\mathbf{x}_t)$ and y_t, the loss function $\mathcal{L}(\boldsymbol{\theta}^{\mathsf{T}}\mathbf{z}_i(\mathbf{x}_t), y_t) = \mathcal{L}(y_t, \boldsymbol{\theta}^{\mathsf{T}}\mathbf{z}_i(\mathbf{x}_t))$ is convex with respect to $\boldsymbol{\theta}$ and is bounded as $\mathcal{L}(\boldsymbol{\theta}^{\mathsf{T}}\mathbf{z}_i(\mathbf{x}_t), y_t) \in [0, \ell_u]$.
2) **(a2)** For any kernel i, $\boldsymbol{\theta}_{t,i}$ belongs to a bounded set $\Theta_i \subseteq \mathbb{R}^{2D}$, i.e., $\|\boldsymbol{\theta}_{t,i}\| \leq C$ for any $t \in [T]$.
3) **(a3)** The loss function is L-Lipschitz continuous, i.e., $\|\nabla\mathcal{L}(\boldsymbol{\theta}^{\mathsf{T}}\mathbf{z}_i(\mathbf{x}_t), y_t)\| \leq L$.

It is remarkable that (a1)–(a3) are usually assumed for the analysis of online convex optimizations and online learning frameworks [13], [14], [16]. In addition, let $f_i^{\star}(\mathbf{x}) = (\boldsymbol{\theta}_i^{\star})^{\mathsf{T}}\mathbf{z}_i(\mathbf{x})$ denote the optimal RF approximation function at the kernel i

$$\boldsymbol{\theta}_i^{\star} \triangleq \arg\min_{\boldsymbol{\theta}\in\Theta_i} \sum_{t=1}^{T} \mathcal{L}(\boldsymbol{\theta}^{\mathsf{T}}\mathbf{z}_i(\mathbf{x}_t), y_t), \quad i \in [P]. \tag{30}$$

Algorithm 1 Proposed AMKL-AKS

Input: Kernels κ_i, $i \in [P]$, parameters $\eta_l, \eta_g, \eta_c, \gamma_t > 0, M \geq 1$, the number of random features D (for RF approximation).
Output: A sequence of functions $\hat{f}_t(\mathbf{x})$, $t \in [T+1]$.
Initialization: $\hat{\boldsymbol{\theta}}_{1,i} = \mathbf{0}$ (i.e., $\hat{f}_{1,i}(\mathbf{x}) = 0$), and $\hat{w}_1(i) = 1$ for all $i \in [P]$.
Iteration: $t = 1, \ldots, T$
- Receive a streaming data $\mathbf{x}_t$.
- Construct $\mathbf{z}_i(\mathbf{x}_t)$ via (7) using the kernel κ_i for $i \in [P]$.
- **Active labeling step:**
  - If $\sum_{\tau=1}^{M} a_{t-\tau} \neq 0$ and the confidence condition in (13) is satisfied:
    - Set $a_t = 0$ and $\hat{f}_{t+1}(\mathbf{x}) = \hat{f}_t(\mathbf{x})$.
    - Skip the local and global steps.
  - Otherwise, set $a_t = 1$ and receive y_t from the oracle.
- **Active local step** (when $a_t = 1$):
  - Update $\hat{\boldsymbol{\theta}}_{t+1,i}$ via OGD in (16).
  - Set $\hat{f}_{t+1,i}(\mathbf{x}) = \hat{\boldsymbol{\theta}}_{t+1,i}\mathbf{z}_i(\mathbf{x})$ for $i \in [P]$.
- **Active global step** ($a_t = 1$):
  - Adaptive kernel selection:
    - Obtain K_{t+1} via (22) and $J_{t+1} = \gamma_{t+1}K_{t+1}$.
    - Construct $\boldsymbol{\Omega}(J_{t+1}, K_{t+1})$ from Remark 2.
    - Obtain $\hat{\boldsymbol{\alpha}}_{t+1}$ via (29).
    - Choose a subset $\mathcal{V}_{s_{t+1}} \in \boldsymbol{\Omega}(J_{t+1}, K_{t+1})$ according to PMF $S_{t+1} \sim \hat{\boldsymbol{\alpha}}_{t+1}$.
  - Function combining:
    - Update $\hat{w}_{t+1}(i)$ via (19).
    - Obtain $\hat{q}_{t+1}(i)$ from (21), for $i \in [P]$.
    - Update $\hat{f}_{t+1}(\mathbf{x}) = \sum_{i \in \mathcal{V}_{s_{t+1}}} \hat{q}_{t+1}(i)\hat{f}_{t+1,i}(\mathbf{x})$.

$\divideontimes$ AMKL performs with $\mathcal{V}_{s_t} = [P]$ for all $t \in [T]$, i.e, the adaptive kernel selection in global step is skipped.

We state the main results of this section, i.e., the regret analysis of the proposed OMKL-AKS, AMKL, and AMKL-AKS.

Theorem 1: For any small $\delta > 0$, OMKL-AKS (or AMKL-AKS with $\mathrm{AL}_{\mathrm{eff}} = 1$) with parameters $\eta_l = \eta_g = \mathcal{O}(1/\sqrt{T})$ guarantees the following regret bound with probability 1-δ:

$$\begin{aligned}&\mathrm{regret}_T^{\mathrm{OL-A}}\\&= \sum_{t=1}^{T}\mathcal{L}\left(\sum_{i\in\mathcal{V}_{S_t}}\hat{q}_t(i)\hat{f}_{t,i}(\mathbf{x}_t), y_t\right) - \min_{1\leq i\leq P}\sum_{t=1}^{T}\mathcal{L}(f_i^\star(\mathbf{x}_t), y_t)\\&\leq \mathcal{O}(\sqrt{T})\end{aligned}$$

where a randomness is from an internal random kernel selection.

Remark 3: We emphasize that Theorem 1 is valid with any choices of J_t and K_t as long as the uniform frequency in the construction of collection (i.e., set of subsets of P kernels) is satisfied, i.e., each kernel i occurs exactly J_t times in the collection. Note that OMKL-AKS with $K_t = P$ for all $t \in [T]$ is equivalent to OMKL (a.k.a., Raker). In this case, the analysis in Theorem 1 holds with $\delta = 0$ as the randomness for a random subset selection disappears. Thus, Theorem 1 encompasses the regret analysis in [14].

For the analysis of AMKL and AMKL-AKS, we further assume that

1) **(a4)** If $\mathcal{L}(\hat{\boldsymbol{\theta}}_{t,i}^{\mathsf{T}}\mathbf{u}_t, \hat{\boldsymbol{\theta}}_{t,j}^{\mathsf{T}}\mathbf{u}_t) \leq \epsilon$ for an input $\mathbf{u}_t$ with $\|\mathbf{u}_t\| = 1$, then there exists a small $B > 0$ such that $\mathcal{L}(\hat{\boldsymbol{\theta}}_{t,i}^{\mathsf{T}}\mathbf{u}, \hat{\boldsymbol{\theta}}_{t,j}^{\mathsf{T}}\mathbf{u}) \leq \epsilon B$ for any $\mathbf{u}$ with $\|\mathbf{u}\| = 1$.
2) **(a5)** $\mathcal{L}(\cdot,\cdot)$ obeys the triangle inequality.

For example, 0–1 loss for classification and ℓ_1/ℓ_2-norm loss in regression satisfy the triangle inequality.

Theorem 2: AMKL with the parameters $\eta_l = \eta_g = \eta_c = \mathcal{O}(1/\sqrt{T})$ guarantees the sublinear regret as

$$\begin{aligned}&\mathrm{regret}_T^{\mathrm{AL}}\\&= \sum_{t=1}^{T}\mathcal{L}(\hat{f}_t(\mathbf{x}_t), y_t) - \min_{1\leq i\leq P}\mathcal{L}(f_i^\star(\mathbf{x}_t), y_t) \leq \mathcal{O}(\sqrt{T}).\end{aligned}$$

Theorem 3: For any $\delta > 0$, AMKL-AKS with the parameters $\eta_l = \eta_g = \eta_c = \mathcal{O}(1/\sqrt{T})$ guarantees the sublinear regret with probability $1-\delta$ as

$$\begin{aligned}&\mathrm{regret}_T^{\mathrm{AL-A}}\\&= \sum_{t=1}^{T}\mathcal{L}\left(\sum_{i\in\mathcal{V}_{S_t}}\hat{q}_t(i)\hat{f}_{t,i}(\mathbf{x}_t), y_t\right) - \min_{1\leq i\leq P}\mathcal{L}(f_i^\star(\mathbf{x}_t), y_t)\\&\leq \mathcal{O}(\sqrt{T}).\end{aligned}$$

The proofs of the main theorems will be provided in the following Sections IV-A and IV-B.

A. Proof of Theorem 1

We prove that the proposed OMKL-AKS achieves the sublinear regret with high probability. We provide key lemmas for the proof of Theorem 1. Lemma 1 in the following states that OGD in the local update can guarantee the sublinear regret. Lemmas 1 and 2 in the following are immediately obtained from [35, Th. 3.1] and [13, Th. 2.1], respectively.

Lemma 1: For any kernel i, OGD in (16) with step size η_l guarantees the following:

$$\begin{aligned}\mathrm{regret}_T^{\mathrm{l}} &= \sum_{t=1}^{T}\mathcal{L}(\hat{f}_{t,i}(\mathbf{x}_t), y_t) - \sum_{t=1}^{T}\mathcal{L}(f_i^\star(\mathbf{x}_t), y_t)\\&\leq \frac{C^2}{2\eta_l} + \frac{\eta_l L^2 T}{2}.\end{aligned}$$

Lemma 2: For any fixed $\eta_g > 0$, OMKL with the EXP strategy in (11) satisfies

$$\begin{aligned}\mathrm{regret}_T^{\mathrm{g}} &= \sum_{t=1}^{T}\sum_{i=1}^{P}\hat{p}_t(i)\mathcal{L}(\hat{f}_{t,i}(\mathbf{x}_t), y_t)\\&\quad - \min_{1\leq i\leq P}\sum_{t=1}^{T}\mathcal{L}(\hat{f}_{t,i}(\mathbf{x}_t), y_t) \leq \frac{\log P}{\eta_g} + \frac{\eta_g T\ell_u^2}{8}.\end{aligned}$$

We are now ready to prove Theorem 1. Note that Lemma 1 holds for any kernel i. Thus, from Lemma 1, Lemma 2,

and the convexity of the loss function $\mathcal{L}(\cdot, y_t)$ for any fixed label y_t, we can get

$$\begin{aligned}&\text{regret}_T^{\text{OL}}\\&= \sum_{t=1}^{T}\mathcal{L}\left(\sum_{i=1}^{P}\hat{p}_t(i)\hat{f}_{t,i}(\mathbf{x}_t), y_t\right) - \min_{1\le i\le P}\sum_{t=1}^{T}\mathcal{L}(f_i^\star(\mathbf{x}_t), y_t)\\&\le \frac{C^2}{2\eta_l} + \frac{\eta_l L^2 T}{2} + \frac{\log P}{\eta_g} + \frac{\eta_g T\ell_u^2}{8}. \end{aligned}\tag{31}$$

Setting $\eta_l = (C)/(\sqrt{T})$ and $\eta_g = 2((2\log P)/(T))^{1/2}$, OMKL (or Raker) guarantees the sublinear regret $\mathcal{O}(\sqrt{T})$. This proves the special case of Theorem 1 with $K_t = P$ for all $t \in [T]$. Then, the general case will be proved using Azuma–Hoeffding's ineqality (i.e., the concentration bound for a martingale difference sequence). Define a random variable X_t as

$$X_t = \sum_{i\in\mathcal{V}_{S_t}}\frac{\hat{w}_t(i)}{\sum_{\ell\in\mathcal{V}_{S_t}}\hat{w}_t(l)}\mathcal{L}(\hat{f}_{t,i}(\mathbf{x}_t), y_t) - U_t$$

where $U_t = \sum_{i=1}^{P}\hat{p}_t(i)\mathcal{L}(\hat{f}_{t,i}(\mathbf{x}_t), y_t)$. Let $\mathcal{F}_t = \sigma(S_1, S_2, \ldots, S_t)$ be the smallest signal algebra such that $S_1, S_2, \ldots, S_t$ is measurable. Then, $\{\mathcal{F}_t : t = 1, \ldots, T\}$ is filtration and X_t is $\mathcal{F}_t$ measurable.Note that condition on $\mathcal{F}_{t-1}$, the $\hat{w}_t(i)$ in (12), $\hat{p}_{t-1}(i)$ in (11), and $q_{S_t}(i)$ in (28) are fixed, and S_t is only random variable. Using this fact, we first show that $\{X_1, \ldots, X_T\}$ is a martingale difference sequence with respect to filtration $\mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \cdots \subseteq \mathcal{F}_T$, by showing that $\mathbb{E}[X_t|\mathcal{F}_{t-1}] = 0$. Then, this claim is proved as follows:

$$\begin{aligned}&\mathbb{E}[X_t|\mathcal{F}_{t-1}]\\&= \mathbb{E}\left[\sum_{i\in\mathcal{V}_{S_t}}\frac{\hat{w}_t(i)}{\sum_{\ell\in\mathcal{V}_{S_t}}\hat{w}_t(\ell)}\mathcal{L}(\hat{f}_{t,i}(\mathbf{x}_t), y_t) - U_t\middle|\mathcal{F}_{t-1}\right]\\&\overset{(a)}{=} \mathbb{E}\left[\sum_{i\in\mathcal{V}_{S_t}}\frac{\hat{w}_t(i)}{\sum_{\ell\in\mathcal{V}_{S_t}}\hat{w}_t(\ell)}\mathcal{L}(\hat{f}_{t,i}(\mathbf{x}_t), y_t)\middle|\mathcal{F}_{t-1}\right] - U_t\\&\overset{(b)}{=} \sum_{j=1}^{|\Omega(J_t,K_t)|}q_{S_t}(j)\left(\sum_{i\in\mathcal{V}_j}\frac{\hat{w}_t(i)}{\sum_{\ell\in\mathcal{V}_j}\hat{w}_t(\ell)}\mathcal{L}(\hat{f}_{t,i}(\mathbf{x}_t), y_t)\right) - U_t\\&= \sum_{j=1}^{|\Omega(J_t,K_t)|}\frac{\sum_{i\in\mathcal{V}_j}\hat{w}_t(i)}{J_t\sum_{i=1}^{P}\hat{w}_t(i)}\mathcal{L}(\hat{f}_{t,i}(\mathbf{x}_t), y_t) - U_t \overset{(c)}{=} 0\end{aligned}$$

where (a) and (b) follow from the fact that $\hat{w}_t(i)$, $\hat{p}_t(i)$, and $q_{S_t}(i)$ are functions of random variables $S_1, \ldots, S_{t-1}$, and (c) follows:

$$\sum_{j=1}^{|\Omega(J_t,K_t)|}\sum_{i\in\mathcal{V}_j}\hat{w}_t(i)\mathcal{L}(\hat{f}_{t,i}(\mathbf{x}_t), y_t) = J_t\sum_{i=1}^{P}\hat{w}_t(i)\mathcal{L}(\hat{f}_{t,i}(\mathbf{x}_t), y_t).$$

Since $\{X_t : t \in [T]\}$ is a martingale difference sequence and $X_t \in [A_t, A_t + c_t]$ is bounded, where $A_t = -\sum_{i=1}^{P}\hat{p}_t(i)\mathcal{L}(\hat{f}_{t,i}(\mathbf{x}_t), y_t)$ is a random variable and $\mathcal{F}_{t-1}$ measurable, and $c_t = \ell_u$. From Azuma–Hoeffding's inequality [36], the following bound holds for some $\delta > 0$ with high probability $1 - \delta$:

$$\sum_{t=1}^{T}X_t = \sum_{t=1}^{T}\sum_{i\in\mathcal{V}_{S_t}}\frac{\hat{w}_t(i)}{\sum_{l\in\mathcal{V}_{S_t}}\hat{w}_t(l)}\mathcal{L}(\hat{f}_{t,i}(\mathbf{x}_t), y_t) - \sum_{t=1}^{T}\sum_{i=1}^{P}\hat{p}_t(i)\mathcal{L}(\hat{f}_{t,i}(\mathbf{x}_t), y_t) \le \sqrt{\frac{\log(\delta^{-1})}{2}T\ell_u^2}.\tag{32}$$

From (32), the following bound holds with probability $1 - \delta$:

$$\begin{aligned}&\sum_{t=1}^{T}\sum_{i\in\mathcal{V}_{S_t}}\frac{\hat{w}_t(i)}{\sum_{\ell\in\mathcal{V}_{S_t}}\hat{w}_t(\ell)}\mathcal{L}(\hat{f}_{t,i}(\mathbf{x}_t), y_t)\\&\quad - \min_{1\le i\le P}\sum_{t=1}^{T}\mathcal{L}(f_i^\star(\mathbf{x}_t), y_t) \le \sum_{t=1}^{T}\sum_{i=1}^{P}\hat{p}_t(i)\mathcal{L}(\hat{f}_{t,i}(\mathbf{x}_t), y_t)\\&\quad - \min_{1\le i\le P}\sum_{t=1}^{T}\mathcal{L}(f_i^\star(\mathbf{x}_t), y_t) + \sqrt{\frac{T\ell_u^2\log(\delta^{-1})}{2}}\\&\overset{(a)}{\le} \frac{C^2}{2\eta_l} + \frac{\eta_l L^2 T}{2} + \frac{\log P}{\eta_g} + \frac{\eta_g T\ell_u^2}{8} + \ell_u\sqrt{\frac{T\log(\delta^{-1})}{2}}\end{aligned}$$

where (a) directly follows from Lemma 1 and Lemma 2. The proof is completed from the convexity of the loss function and by setting $\eta_l = \eta_g = \mathcal{O}(1\sqrt{T})$.

B. Proofs of Theorem 2 and Theorem 3

We prove the optimal sublinear regrets of the proposed AMKL and AMKL-AKS. We first derive the regret analysis of OGD in the active local step. This analysis is different from Lemma 1 since, as shown in (17), kernel functions cannot be updated at sometimes. The following lemma shows that the active local step can still guarantee the sublinear regret as long as the number of consecutive unlabeled data is a certain constant (i.e., not grow with T).

Lemma 3: Let M denote the maximum consecutive zeros (i.e., unlabeling) in $\{a_t : t \in [T]\}$. For any kernel i, OGD in (17) (i.e., in active local step) with step size η_ℓ guarantees the following:

$$\begin{aligned}\text{regret}_T^{\text{al}} &= \sum_{t=1}^{T}\mathcal{L}(\hat{f}_{t,i}(\mathbf{x}_t), y_t) - \sum_{t=1}^{T}\mathcal{L}(f_i^\star(\mathbf{x}_t), y_t)\\&\le \frac{M+1}{2}\left(\frac{C^2}{\eta_l} + \eta_l L^2 T\right).\end{aligned}$$

Setting $\eta_l = \mathcal{O}(1/\sqrt{T})$, OGD in the active local step can achieve the sublinear regret.

Proof: The proof is provided in Appendix A. ■

For the purpose of AMKL analysis, we introduce a *virtual* OMKL. This method employs the same kernel functions with AMKL, i.e., both AMKL and virtual OMKL use the kernel functions $\hat{f}_{t,i}, i \in [P], t \in [T]$ in active local update. Whereas, in virtual OMKL, the weights are updated as if all labels $\{y_t : t \in [T]\}$ are revealed, namely

$$\tilde{p}_t(i) \triangleq \frac{\tilde{w}_t(i)}{\sum_{i=1}^{P}\tilde{w}_t(i)}\tag{33}$$

where $\tilde{w}_t(i) = \exp(-\eta_g\sum_{\ell=1}^{t-1}\mathcal{L}(\hat{f}_{i,\ell}(\mathbf{x}_\ell), y_\ell))$ for some parameter $\eta_g > 0$ and with the initial values $\tilde{w}_1(i) = 1$, $i \in [P]$. Then, virtual OMKL learns a target function $\tilde{f}_t$ as

$$\tilde{f}_t(\mathbf{x}) = \sum_{i=1}^{P}\tilde{p}_t(i)\hat{f}_{t,i}(\mathbf{x}).\tag{34}$$

Comparing virtual OMKL and AMKL, we derive the following key lemmas.

Lemma 4: For a small constant $\eta_c > 0$, the confidence condition in (13) for AMKL (i.e., $\mathcal{V}_{s_t} = [P], \forall t \in [T]$) implies $\mathcal{L}(\hat{f}_t(\mathbf{x}_t), \tilde{f}_t(\mathbf{x}_t)) \leq \eta_c$.

Proof: The proof is provided in Appendix B. ∎

Lemma 5: Letting $a_t = 0$ and $a_{t+1} \neq 0$, we have $\mathcal{L}(\hat{f}_{t+1}(\mathbf{x}_{t+1}), \tilde{f}_{t+1}(\mathbf{x}_{t+1})) \leq \eta_c B$.

Proof: The proof is provided in Appendix B. ∎

Lemma 6: Setting $\eta_c = \mathcal{O}(1/\sqrt{T})$, the following sublinear regret holds:

$$\text{regret}_T^{\text{a}} = \sum_{t=1}^{T} \mathcal{L}(\hat{f}_t(\mathbf{x}_t), y_t) - \sum_{t=1}^{T} \mathcal{L}(\tilde{f}_t(\mathbf{x}_t), y_t) \leq \mathcal{O}(\sqrt{T}).$$

Proof: The proof is provided in Appendix C. ∎

From now on, we will prove the main theorem using the above-mentioned key lemmas. From Lemma 2 and the convexity of the loss function, we can obtain the regret bound of vOMKL with $\eta_g = \mathcal{O}(1/\sqrt{T})$, which is given as

$$\begin{aligned}\text{regret}_T^{\text{l}} &= \sum_{t=1}^{T} \mathcal{L}(\tilde{f}_t(\mathbf{x}_t), y_t) - \min_{1\leq i\leq P} \sum_{t=1}^{T} \mathcal{L}(\hat{f}_{i,t}(\mathbf{x}_t), y_t) \\ &\leq \mathcal{O}(\sqrt{T}).\end{aligned}$$

Then, the proof is completed as

$$\text{regret}_T^{\text{AL}} = \text{regret}_T^{\text{a}} + \text{regret}_T^{\text{l}} + \text{regret}_T^{\text{al}} \leq \mathcal{O}(\sqrt{T})$$

where $\text{regret}_T^{\text{a}}$ from Lemma 5 and $\text{regret}_T^{\text{al}} \leq \mathcal{O}(\sqrt{T})$ from Lemma 3 with $\eta_l = \mathcal{O}(1/\sqrt{T})$. This completes the proof of Theorem 2. In addition, in the proof of Theorem 1, it was shown that the proposed kernel selection can keep the sublinear regret with high probability, as long as the underlying OMKL can do it. The same argument can be applied for the case of AMKL and AMKL-AKS. From Theorem 1 and Theorem 2, thus, the proof of Theorem 3 is completed.

V. Experiments

In the following experiments, we evaluate the performances of the proposed AMKL-AKS, and its special cases AMKL and OMKL-AKS (or AMKL-AKS with $\text{AL}_{\text{eff}} = 1$) for various online learning tasks, such as online regressions, online classifications, and time-series predictions. Regarding loss functions, the regularized least-square function is used for online regressions and time-series predictions, and the regularized logistic function is used for online classifications. Due to the randomness of the above-mentioned algorithms caused by $\mathbf{z}_i(\mathbf{x})$ in (7), the averaged performances over 30 trials are evaluated. MATLAB is employed as the programming language. We consider the following performance measures to evaluate the learning accuracy, AL efficiency, and computational complexity of various learning algorithms. Let $\hat{y}_\tau$ and y_τ denote an estimated label and a true label, respectively.

1) *Learning Accuracy:* For online regressions and time-series predictions, the accuracy of a function learning is measured by the mean-square error (MSE), defined as $\text{MSE}(t) = (1)/(t)\sum_{\tau=1}^{t}(\hat{y}_\tau - y_\tau)^2$. In addition, for online classifications, it is measured by an error probability, defined as $\text{Error}(t) = (1)/(t)\sum_{\tau=1}^{t}\max\{0, \text{sign}(-y_\tau \hat{y}_\tau)\}$.
2) *AL Efficiency:* The AL efficiency, denoted by AL_{eff}, is measured as in (1).
3) *Computational Complexity:* We consider the CPU running time [i.e., execution time (second)] to compare the computational complexities of various learning algorithms.

We remark that $\text{AL}_{\text{eff}} = 1$ for OMKL and OMKL-AKS as all the incoming data are labeled. In AMKL and AMKL-AKS, however, AL_{eff} could be less than 1, and note that a fewer number of labeled data is used as AL_{eff} decreases. In addition, it is remarkable that AL_{eff} cannot be chosen in advance and it is determined as a consequence of the AL process. Thus, in our experimental results, AL_{eff} is data dependent. For comparisons, the following benchmark methods are considered.

1) *RBF:* The online *single*-kernel learning method using Gaussian kernels with the parameters $\sigma^2 = [0.1, 1, 10]$ [e.g., KL-RBF(σ^2)].
2) *POLY:* The online *single*-kernel learning method using polynomial kernels with degree $d = \{2,3\}$ (e.g., POLY2 and POLY3).
3) *LINEAR:* The online *single*-kernel learning method using a linear kernel.
4) *OMKR:* The famous OMKL algorithm without RF approximation [12].
5) *OMKL-B:* The OMKL algorithm on a budget [11].
6) *RAKER:* The OMKL algorithm based on RF approximation [14].

Regarding the above-mentioned online algorithms, the following parameters will be used throughout the experiments. The parameter settings closely follow the most relevant work in [14] for fair comparisons. For all MKL algorithms as OMKR, OMKL-B, Raker, OMKL-AKS, AMKL, and AMKL-AKS, we use the kernel dictionary consisting of 17 Gaussian kernels, whose parameters are given as $\sigma_i^2 = 10^{(i-9)/(2)}$, $i = 1, \ldots, 17$. In addition, for RF-based OMKL algorithms, such as Raker, OMKL-AKS, AMKL, and AMKL-AKS, the associated parameters are set by

$$\eta_l = \eta_g = \frac{1}{\sqrt{T}}, \quad D = 50, \quad \text{and } \lambda = 0.01. \tag{35}$$

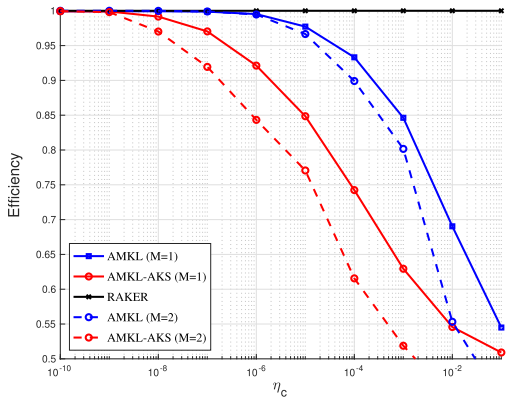
The budget size of OMKL-B is chosen as $B = 50$. In OMKL-AKS and AMKL-AKS, the size-K_t subset from the kernel dictionary is selected at every time t, where K_t is determined from (22) for OMKL-AKS and AMKL-AKS, respectively, with $\delta_t = 0.8$ for all $t \in [T]$ and $\gamma_t = \min\{\binom{P}{K_t}/P, 2\}$. In this way, the size of a collection is manageable during experiments as it is always less than or equal to $\gamma_t P = 34$. Finally, for AMKL and AMKL-AKS, the following parameters are chosen for the proposed selection criterion:

$$\eta_c = 0.0005 \text{ and } M = 1. \tag{36}$$

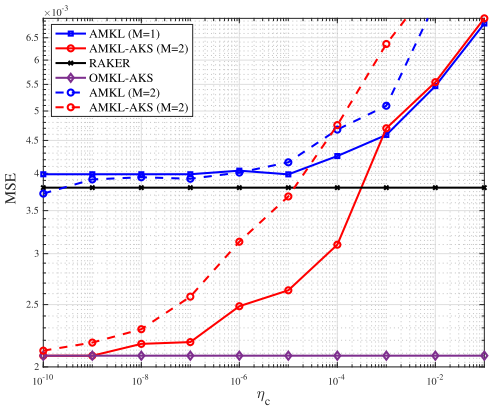
Obviously, these parameters can control the tradeoff between AL efficiency and learning accuracy of AMKL-AKS, as shown in Fig. 3. Instead of optimizing them for each data set, one pair of the parameters in (36) is used for all test data sets. As explained in Section I-B, such optimization is left for

TABLE I
SUMMARY OF REAL DATA SETS FOR EXPERIMENTS

Datasets	# of features	# of data	feature type
Regression task			
Twitter	77	13818	real & integer
Twitter (Large)	77	98704	real& integer
Tom's hardware	96	9725	real& integer
Air quality	13	7322	real
Appliance energy	25	18604	real
Naval propulsion plants	16	11934	real
Classification task			
Movement	4	13197	real
Electronic Device	60	3600	real
Human Activity	30	7352	real
Time series prediction task			
Traffic	5,10	6500	real
Temperature	5,10	5500	real



(a)



(b)

Fig. 3. Tradeoff between AL efficiency and learning accuracy of AMKL and AMKL-AKS for Tom's hardware data set (a) Active-learning efficiency. (b) MSE performance.

future work. We remark that, in this section, we specifically focus on revealing the superiority of AMKL-AKS compared with Raker, OMKL-AKS, and AMKL since Raker has already proved its superiority over (O)MKL methods in terms of both error performance and CPU time on various online learning tasks in [14].

A. Real-Data Tests for Online Regressions

For the experiments of online regressions, the following real data sets from UCI Machine Learning Repository are considered, which are also summarized in Table I.

1) *Twitter [37]:* Data contains buzz events from Twitter, where each attribute are used to predict the popularity of a topic. A higher value indicates more popularity. The larger data set with $T = 98704$ [termed *Twitter(L)*] is included to test algorithms.
2) *Tom's Hardware [37]:* Data consist of samples acquired from a forum, where each feature represents, such as the number of times content is displayed to visitors. The task is to predict the average number of displays on a certain topic.
3) *Air Quality [38]:* Data include samples of which features include an hourly response from an array of chemical sensors embedded in a city of Italy. The goal is to predict the concentration of polluting chemicals in the air.
4) *Appliances Energy [39]:* This data set contains samples describing appliances energy use, such as temperature and pressure in houses. The goal is to predict energy use in a low-energy building. A higher value denotes higher energy consumption.
5) *Naval Propulsion Plants [40]:* This data set has been obtained from the Gas Turbine plant. The data set contains samples with 16 features, such as ship speed and fuel flow. The goal is to determine the turbine decay state coefficient.

Performance Evaluation: In particular, we consider two types of AMKL-AKS, one of which uses all the labeled data (called OMKL-AKS) and the other follows the proposed selection criterion, in order to show the tradeoff of AL efficiency and learning accuracy. Fig. 4 shows the MSE performances of various online and AL algorithms. The numerical results of AMKL and AMKL-AKS are also summarized in Table II, where MSE and AL_{eff} are measured at the end of time. In Fig. 4, as observed in [14], RF-based algorithms as Raker, OMKL-AKS, AMKL, and AMKL-AKS, significantly outperform the famous (O)MKL methods and single-kernel methods (e.g., Gaussian, POLY, and Linear methods). Here, we more focus on the accuracy–efficiency tradeoff of the proposed AMKL-AKS in Table II. In stream-based AL frameworks, the following two factors play a key role in determining performances: one is to set a sharp selection criterion which enables an algorithm to choose essential data for labeling, and the other is to exploit the most relevant kernels for predicting a label precisely. In Table II, AMKL-AKS provides notable performances, showing its solidity in terms of the above-mentioned two factors. Namely, AMKL-AKS shows comparable MSE performances with OMKL-AKS, ensuring that the proposed selection criterion is accurate enough to avoid unnecessary label requests. Furthermore, we observe that AMKL-AKS performs better than AMKL using a fewer number of labeled data in most of the test data sets. These results imply that the elimination of irrelevant kernels (in a data-driven way) has a positive impact on AMKL, thereby improving the learning accuracy. In the comparisons of Raker and OMKL-AKS, a similar impact is observed.

Regarding the AL efficiency, Table II clearly demonstrates that the objective of AMKL-AKS is attained since it yields highly close performances as Raker and OMKL-AKS, only using $60\% \sim 70\%$ of labeled data (i.e., $\text{AL}_{\text{eff}} = 0.6 \sim 0.7$).

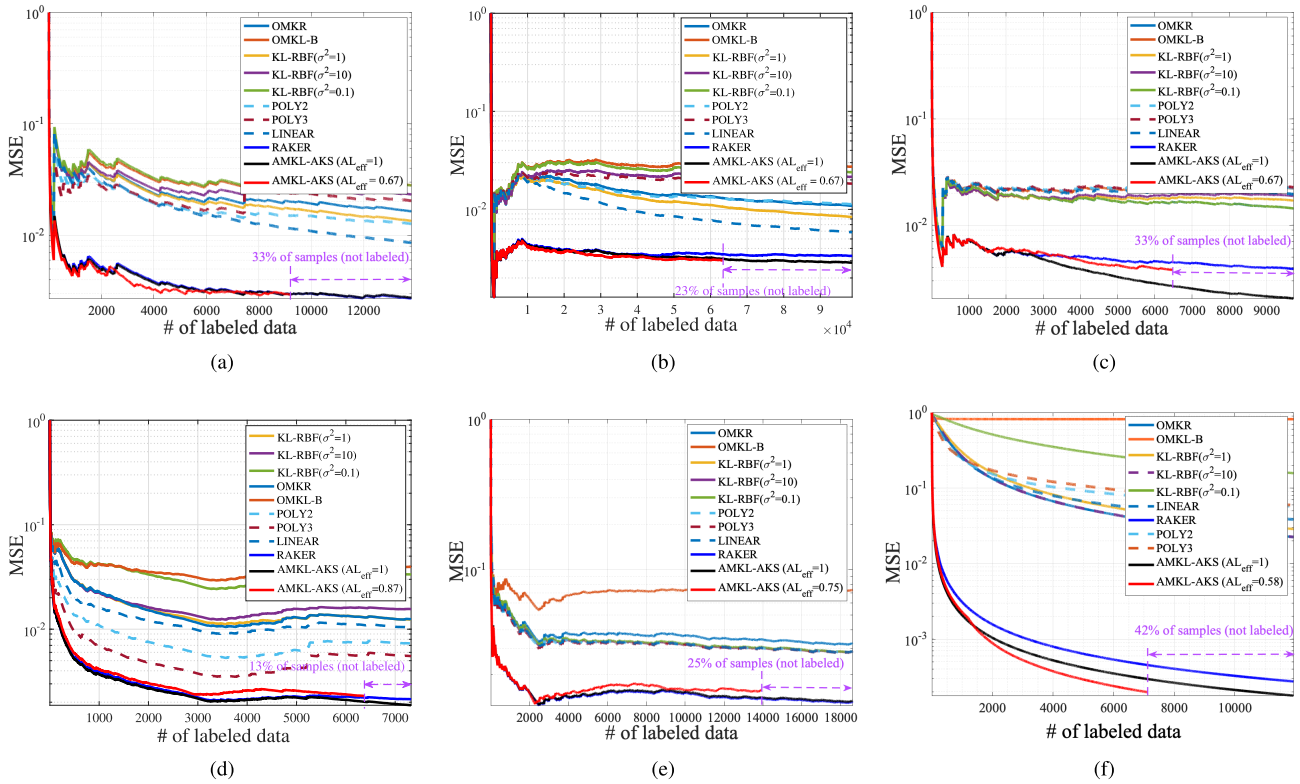


Fig. 4. Comparisons of MSE performances of various learning methods on online regression tasks. Note that to highlight the effectiveness of AMKL-AKS, the number of labeled data rather than time t is used as x-axis (a) Twitter data. (b) Twitter data (large). (c) Tom's hardware data. (d) Air quality data. (e) Appliances energy data. (f) Naval propulsion plant data.

TABLE II

COMPARISONS OF MSE ($\times 10^{-3}$) AND AL EFFICIENCY ON ONLINE REGRESSIONS

	Twitter		Twitter(L)		Tom's		Air		Energy		Plant	
	MSE	AL$_{\text{eff}}$	MSE	AL$_{\text{eff}}$	MSE	AL$_{\text{eff}}$	MSE	AL$_{\text{eff}}$	MSE	AL$_{\text{eff}}$	MSE	AL$_{\text{eff}}$
RAKER (D=50)	2.66	1	3.30	1	3.91	1	2.25	1	13.22	1	0.26	1
AMKL	2.70	0.95	3.32	0.98	4.31	0.88	2.23	0.99	13.25	0.98	0.27	1
AMKL-AKS (AL$_{\text{eff}}$ = 1)	2.71	1	2.84	1	2.05	1	1.96	1	13.43	1	0.19	1
AMKL-AKS	2.93	0.67	2.94	0.73	3.87	0.67	2.26	0.87	15.75	0.75	0.19	0.58

Remarkable, in the Plant data set, AMKL-AKS achieves the almost same performance as OMKL-AKS with 60% of labeled data. Moreover, it shows the outstanding performance over Raker with a fewer number of labeled data, which proves the superiority of AMKL-AKS. From Fig. 4, it is clearly shown that AMKL-AKS significantly outperforms the other methods [e.g. (O)MKL and POLY, and KL-RBF] with a fewer number of labeled data. Based on these results, we emphasize that AMKL-AKS can have a significant impact on the economical aspect having 30% $\sim$ 40% cost reduction for label acquisitions. As mentioned in Section I, this cost reduction would be crucial for learning tasks with medical data.

In the aspect of computational complexity, it is noticeable that AMKL-AKS yields attractive time efficiency. In Table III, the execution times (or CUP running times) of various online learning algorithms are provided. From Table III, we observe that AMKL-AKS achieves superb competitiveness toward Raker in regards to time-complexity and the MSE performance. Especially, in Tom and Plant data sets, AMKL-AKS shows better MSE performances than Raker with 20% and 15% of CPU time saving, respectively. The computational efficiency of AMKL-AKS mainly comes from two factors. One is to employ the RF approximation as in Raker [14], where the computational complexity does not grow with T, while the other MKL methods suffer from it. On top of this, due to the proposed selection criterion, AMKL-AKS enjoys its solid computational efficiency since it can skip the unnecessary active local and global steps. Although the active labeling step may seem to require some additional computation time, it may be negligible compared with the reduction of unnecessary function learning steps.

TABLE III

COMPARISONS OF CPU TIME (S) ON ONLINE REGRESSIONS

	Twitter	Twitter(L)	Tom's	Air	Energy	Plant
RAKER (D=50)	4.08	56.5	3.03	1.32	3.41	2.2
AMKL	4.69	73.6	2.81	1.44	4.02	2.36
AMKL-AKS	4.46	58.8	3.08	1.26	3.61	2.29
AMKL-AKS	3.76	37.7	2.45	1.42	3.66	1.89

TABLE IV
COMPARISONS OF ERROR (%), AL EFFICIENCY, AND CPU TIME (S) ON ONLINE CLASSIFICATIONS

	Movement			Device			Activity		
	Error	AL_{eff}	Time(s)	Error	AL_{eff}	Time(s)	Error	AL_{eff}	Time(s)
RAKER	11.37	1	3.43	2.45	1	0.75	0.956	1	1.54
AMKL	11.37	1	3.45	2.45	0.96	0.83	0.987	1	1.71
AMKL-AKS	11.34	1	3.15	2.45	1	0.70	0.677	1	1.57
AMKL-AKS	13.20	0.83	2.59	2.53	0.96	0.76	1.0	0.63	1.19

Remark 4: We shed light on an important relationship between AKS and AL efficiency. From the experimental results, we observe that the proposed AKS has a crucial role in enhancing the AL efficiency. Table II shows that AMKL-AKS attains a similar or better MSE performance than AMKL (using the entire 17 kernels) with higher efficiency in all test data sets. This interesting observation leads us to conclude that AMKL-AKS indeed enjoys the advantage of refined kernels. Specifically, the kernel selection enables to improve the accuracy of the proposed selection criterion (for active labeling) as inaccurate information from irrelevant kernels can be excluded at every time. In other words, AMKL-AKS can predict a function with higher accuracy by removing irrelevant kernels, so that it is in need of just a few labeled data compared with AMKL.

Remark 5 (Comparisons with uniform sampling): We notice the preciseness of the proposed selection criterion for all the online learning tasks in our experiments. To verify its effectiveness, we consider OMKL-AKS with *uniform sample selection*, where a learner decides whether to query or discard an incoming data randomly and uniformly. That is, the incoming data can be labeled with a predetermined probability of p_s. We call this method rOMKL-AKS. We compare the performances of AMKL-AKS and rOMKL-AKS to manifest the superiority of the proposed selection criterion. Note that in rOMKL-AKS, the parameter p_s, which determines AL efficiency, should be chosen in advance, and this may not be practical. Whereas, in AMKL-AKS, AL efficiency is determined as the consequence of the AL process. For fair comparisons, p_s is chosen according to AL_{eff} obtained from AMKL-AKS, so that rOMKL-AKS yields a similar AL efficiency with AMKL-AKS. From the experiments, we observe that for most of the data sets, rOMKL-AKS performs worse in accuracy than AMKL-AKS, even with a larger number of labeled data. For example, rOMKL-AKS demonstrates the MSE $= 3.46 \times 10^{-3}$ with $AL_{eff} = 0.76$ on Twitter data, MSE $= 4.44 \times 10^{-3}$ with $AL_{eff} = 0.76$ on Tom data and MSE $= 2.51 \times 10^{-3}$ with $AL_{eff} = 0.92$ on Air data. A similar tendency has been also observed in the classification and time series prediction tasks. For example, rOMKL-AKS shows the 1% classification error with $AL_{eff} = 0.74$ on activity data, and MSE $= 0.221 \times 10^{-3}$ with $AL_{eff} = 0.74$ on temperature data. By comparing with the performances of AMKL-AKS, we can conclude that the proposed selection criterion is indeed meaningful.

B. Real-Data Tests for Online Classifications

We more deeply investigate the performances of Raker, OMKL-AKS, AMKL, and AMKL-AKS on online classification tasks. For consistency with the related work, we test the above-mentioned algorithms on the same data sets in [14].

1) *Movement [41]:* This data set contains received signal strength measured between the nodes of a sensor network comprising four anchor nodes where each attribute. Data are collected during user movement at the frequency of 8 Hz. The binary label $\mathbf{y}_t$ indicates whether the user's trajectory will change into the spatial context or not.
2) *Electronic Device [42]:* This data set contains samples of which each feature vector represents electricity readings from different households. Binary label $\mathbf{y}_t$ indicates the type of electronic devices used at a certain interval of time: dishwasher or kettle.
3) *Human Activity [43]:* These data are collected from a group of 30 volunteers wearing a smartphone on their waist. Feature vectors measure body movements. Binary label $\mathbf{y}_t$ represents the activity during a certain period: walking or not walking.

Performance Evaluation: As similarly in online regression tasks, AMKL-AKS shows the notable efficiency and comparable performance in online classification tasks in Table IV. On the Activity data set, AMKL-AKS performs accurate enough compared with Raker while having higher efficiency. Specifically, with 63% of entire samples, AMKL-AKS shows only about 0.05% of accuracy difference with Raker. In addition, the error probability of OMKL-AKS is notable as it even performs 0.3% better than Raker, which implies the effectiveness of the proposed AKS. From this, we can conclude that the proposed AKS and selection criterion have indeed shown its effect on the process of the function learning task.

In the perspective of computational complexity, AMKL-AKS enjoys the advantage of the proposed selection criterion in online classification tasks as well. On Movement and Activity data set, each proves about 25% and 23% time efficiency with comparable error probability to Raker. This remarkable observation implies that the proposed AMKL-AKS demonstrates its universal applicability on various learning tasks.

C. Real-Data Tests for Time Series Predictions

We discuss the natural extensions of the proposed AMKL-AKS into time series prediction tasks which predict the future values in online fashion. Toward this, the famous time series prediction method called Autoregressive (AR) model is considered. An AR(p) model predicts the future value $\mathbf{y}_t$ assuming the linear dependence on its past p values, that is,

$$y_t = c + \sum_{i=1}^{p} \alpha_i y_{t-i} + \epsilon_t \tag{37}$$

TABLE V
COMPARISON OF MSE, AL EFFICIENCY, AND CPU TIME (S) ON TIME-SERIES PREDICTIONS

	Traffic			Temperature		
	MSE	AL_{eff}	Time(s)	MSE	AL_{eff}	Time(s)
$p = 5$						
RAKER	10.02	1	1.06	0.22	1	0.77
AMKL	10.37	1	1.17	0.23	0.88	1.03
AMKL-AKS	10.51	1	1.2	0.21	1	0.9
AMKL-AKS	13.08	0.92	1.08	0.22	0.67	0.86
$p = 10$						
RAKER	10.5	1	1.25	0.23	1	1.05
AMKL	10.45	1	1.38	0.23	0.91	1.03
AMKL-AKS	10.89	1	1.15	0.21	1	1.09
AMKL-AKS	14.68	0.90	1.16	0.22	0.68	0.92

where c is a constant, α_i denotes the weight associated with y_{t-i}, and ϵ_t denotes a Gaussian noise at time t. Based on this, the kernelized AR(p) model, which can explore a nonlinear dependence, is introduced in [12], where it is formulated as

$$y_t = c + f(y_{t-1}, y_{t-2}, \ldots, y_{t-p}) + \epsilon_t \tag{38}$$
$$= c + f(\mathbf{x}_t) + \epsilon_t \tag{39}$$

where $\mathbf{x}_t \triangleq [y_{t-1}, y_{t-2}, \ldots, y_{t-p}]^\mathsf{T}$ and $f(\mathbf{x}_t)$ belongs to kernel space. In this section, we further consider its natural extension to *RF-based* kernelized AR(p). Based on RF approximation, $f(\mathbf{x}_t)$ can be well approximated as in (8). Finally, this can be directly plugged into AMKL framework to solve time series prediction tasks. The proposed algorithms are tested with the following univariate time series data sets from UCI Machine Learning Repository.

1) *Traffic [44]:* $T = 5600$ time series traffic data obtained from Department of Transportation in the U.S. Data are collected from hourly traffic volume at the street in Minneapolis.
2) *Temperature [44]:* $T = 5500$ time series temperature data obtained from the same source as earlier.

Performance Evaluation: We consider the AR(p) model with $p = 5$ and 10. For experiments, the data sets in the earlier are normalized to $[0, 1]$. The performances of AMKL and AMKL-AKS on time series data sets are provided in Table V. In both $p = 5$ and $p = 10$, the proposed AMKL-AKS shows notable performance and high AL efficiency compared with Raker and OMKL-AKS. Specifically, in temperature data with $p = 5$, AMKL-AKS shows the outstanding performance over Raker only using 67% of entire samples, which naturally results in its time efficiency (i.e., lower execution time). In addition, among the algorithms, AMKL-AKS shows the almost same performance by maintaining the shortest execution time.

VI. CONCLUSION

In this article, we proposed a stream-based (or sequential) AL for OMKL frameworks. The proposed method is named AMKL. We further improved the learning accuracy and AL efficiency of AMKL by introducing an AKS, which is named AMKL-AKS. Theoretically, it was proved that AMKL-AKS achieves an optimal sublinear regret as in OMKL, implying that the proposed selection criterion indeed avoids unnecessary label requests. Beyond the asymptotic analysis, numerical tests with real data sets verified that AMKL-AKS yields a similar or better accuracy than the existing OMKL (termed Raker) using a fewer number of labeled data. Therefore, the proposed AMKL-AKS can provide an elegant accuracy–efficiency tradeoff. One interesting extension is to improve AMKL-AKS by exploiting *a priori* knowledge on a kernel dictionary. For example, in addition to the accumulated loss information, kernel dependencies can be also used for an AKS. Another extension is to develop an AL for online graph learning frameworks, in which the graph dependencies of data samples can be exploited for active labeling.

APPENDIX A
PROOF OF LEMMA 3

Let $\mathcal{A}$ be the index set of revealed labels. Then, the regret can be decomposed as

$$\underbrace{\sum_{t\in\mathcal{A}} \mathcal{L}(\hat{f}_{t,i}(\mathbf{x}_t), y_t) - \sum_{t\in\mathcal{A}} \mathcal{L}(f_i^\star(\mathbf{x}_t), y_t)}_{\triangleq(\star)} + \underbrace{\sum_{t\in\mathcal{A}^c} \mathcal{L}(\hat{f}_{t,i}(\mathbf{x}_t), y_t) - \sum_{t\in\mathcal{A}^c} \mathcal{L}(f_i^\star(\mathbf{x}_t), y_t)}_{\triangleq(\star\star)}.$$

Clearly, the part $(\star)$ is the regret of the usual online gradient descent (OGD) with time indices belong to $\mathcal{A}$. Then, from Lemma 1, we can get

$$(\star) \le \frac{C^2}{2\eta_1} + \frac{\eta_l L^2 |\mathcal{A}|}{2}. \tag{40}$$

Now, we focus on the part $(\star\star)$, which is different from the usual OGD. Let $\mathcal{A}^c \triangleq \{t_1, \ldots, t_{|\mathcal{A}^c|}\}$ with $t_1 < t_2 < \cdots < t_{|\mathcal{A}^c|}$. Let $\mathcal{A}_n^c$ be the subset of $\mathcal{A}^c$ only containing nonconsecutive indices, where among consecutive indices, the maximum index is only included. For example, if $\mathcal{A}^c = \{3, 4, 5, 9, 11, 12, 15\}$, then we have $\mathcal{A}_n^c = \{5, 9, 12, 15\}$. Following this notation, we let $\mathcal{A}_n^c = \{t_{\ell_1}, \ldots, t_{\ell_{|\mathcal{A}_n^c|}}\}$ with $t_{\ell_1} < t_{\ell_2} < \cdots < t_{\ell_{|\mathcal{A}_n^c|}}$. To simplify the notation, we let $\nabla_t \triangleq \nabla\mathcal{L}(\hat{\boldsymbol{\theta}}_{i,t}^\mathsf{T}\mathbf{z}_i(\mathbf{x}_t), y_t)$. For any $t_{\ell_j}, t_{\ell_{j+1}} \in \mathcal{A}_n^c$, we define the index set as $\mathcal{T}_j = \{t_{\ell_j} + 1, \ldots, t_{\ell_{j+1}} - 1\} \cap \mathcal{A}$. Using this, we have the following bound:

$$\begin{aligned}
\|\hat{\boldsymbol{\theta}}_{i,t_{\ell_{j+1}}} - \boldsymbol{\theta}_i^\star\|^2 &= \Big\|\hat{\boldsymbol{\theta}}_{i,t_{\ell_j}} - \eta_l \nabla_{t_{\ell_j}} - \eta_l \sum_{t\in\mathcal{T}_j} \nabla_t - \boldsymbol{\theta}_i^\star\Big\|^2 \\
&\le \|\hat{\boldsymbol{\theta}}_{i,t_{\ell_j}} - \eta_l \nabla_{t_{\ell_j}} - \boldsymbol{\theta}^\star\|^2 + \eta_l^2 \sum_{t\in\mathcal{T}_j} \|\nabla_t\|^2 \\
&\overset{(a)}{\le} \|\hat{\boldsymbol{\theta}}_{i,t_{\ell_j}} - \eta_l \nabla_{t_{\ell_j}} - \boldsymbol{\theta}_i^\star\|^2 + |\mathcal{T}_j| \eta_l^2 L^2 \\
&= \|\hat{\boldsymbol{\theta}}_{i,t_{\ell_j}} - \boldsymbol{\theta}_i^\star\|^2 + \eta_l^2 \|\nabla_{t_{\ell_j}}\|^2 \\
&\quad - 2\eta_l \nabla_{t_{\ell_j}}^\mathsf{T} (\hat{\boldsymbol{\theta}}_{i,t_{\ell_j}} - \boldsymbol{\theta}_i^\star) + |\mathcal{T}_j| \eta_l^2 L^2
\end{aligned} \tag{41}$$

where (a) is due to the fact that loss functions are L-Lipschitz. Note that the above-mentioned inequality always

holds since it is ensured that t_{ℓ_j} and $t_{\ell_{j+1}}$ are not consecutive. By rearranging (41), we obtain the following bound:

$$\begin{aligned}&\nabla_{t_{\ell_j}}^{\mathsf{T}}(\hat{\boldsymbol{\theta}}_{i,t_{\ell_j}}-\boldsymbol{\theta}_i^\star)\\&\le\frac{\|\hat{\boldsymbol{\theta}}_{i,t_{\ell_j}}-\boldsymbol{\theta}_i^\star\|^2-\|\hat{\boldsymbol{\theta}}_{i,t_{\ell_{j+1}}}-\boldsymbol{\theta}_i^\star\|^2}{2\eta_l}+\frac{\eta_l\|\nabla_{t_{\ell_j}}^{\mathsf{T}}\|^2}{2}\\&\quad+|\mathcal{T}_j|\eta_l^2L^2.\end{aligned}\tag{42}$$

In addition, the convexity of the loss function implies that

$$\begin{aligned}&\mathcal{L}(\hat{\boldsymbol{\theta}}_{i,t_{\ell_j}}^{\mathsf{T}}\mathbf{z}_i(\mathbf{x}_{t_{\ell_j}}),y_{t_{\ell_j}})-\mathcal{L}((\boldsymbol{\theta}_i^\star)^{\mathsf{T}}\mathbf{z}_i(\mathbf{x}_{t_{\ell_j}}),y_{t_{\ell_j}})\\&\le\nabla_{t_{\ell_j}}^{\mathsf{T}}(\hat{\boldsymbol{\theta}}_{i,t_{\ell_j}}-\boldsymbol{\theta}_i^\star)\\&\le\frac{\|\hat{\boldsymbol{\theta}}_{i,t_{\ell_j}}-\boldsymbol{\theta}_i^\star\|^2-\|\hat{\boldsymbol{\theta}}_{i,t_{\ell_{j+1}}}-\boldsymbol{\theta}_i^\star\|^2}{2\eta_l}+\frac{\eta_l\|\nabla_{t_{\ell_j}}^{\mathsf{T}}\|^2}{2}\\&\quad+|\mathcal{T}_j|\eta_l^2L^2\end{aligned}$$

where the second inequality follows from (42). By telescoping sum over $t\in\mathcal{A}_n^c$, we can get

$$\begin{aligned}&\sum_{t\in\mathcal{A}_n^c}\mathcal{L}(\hat{\boldsymbol{\theta}}_{i,t}^{\mathsf{T}}\mathbf{z}_i(\mathbf{x}_t),y_t)-\sum_{t\in\mathcal{A}_n^c}\mathcal{L}((\boldsymbol{\theta}_i^\star)^{\mathsf{T}}\mathbf{z}_i(\mathbf{x}_t),y_t)\\&\le\frac{\|\hat{\boldsymbol{\theta}}_{i,t_{\ell_1}}-\boldsymbol{\theta}_i^\star\|^2-\|\hat{\boldsymbol{\theta}}_{i,t_{\ell_{|\mathcal{A}_n^c|}}}-\boldsymbol{\theta}_i^\star\|^2}{2\eta_1}+\frac{\eta_1L^2|\mathcal{A}_n^c|}{2}\\&\quad+\frac{\eta_1L^2}{2}\sum_{j=1}^{|\mathcal{A}_n^c|}|\mathcal{T}_j|\le\frac{C^2}{2\eta_1}+\frac{\eta_l^2L^2T}{2}\end{aligned}\tag{43}$$

where the last inequality is due to the fact that $|\mathcal{A}_n^c|+\sum_{j=1}^{|\mathcal{A}_n^c|}|\mathcal{T}_j|\le T$. By following the above-mentioned procedures with the indices belong to $\mathcal{A}^c\setminus\mathcal{A}_n^c$, we have a similar bound. Since the number of consecutive unlabeling data is less than or equal to M, the following bound holds:

$$\begin{aligned}&\sum_{t\in\mathcal{A}^c}\mathcal{L}\big(\hat{\boldsymbol{\theta}}_{i,t}^{\mathsf{T}}\mathbf{z}_i(\mathbf{x}_t),y_t\big)-\sum_{t\in\mathcal{A}^c}\mathcal{L}\big((\boldsymbol{\theta}_i^\star)^{\mathsf{T}}\mathbf{z}_i(\mathbf{x}_t),y_t\big)\\&\qquad\le M\left(\frac{C^2}{2\eta_1}+\frac{\eta_l^2L^2T}{2}\right).\end{aligned}\tag{44}$$

From (40) and (44), the proof is completed.

Appendix B
Proofs of Lemma 4 and Lemma 5

We first prove Lemma 4. Leveraging the convexity of the loss function, we have

$$\begin{aligned}&\mathcal{L}(\hat{f}_t(\mathbf{x}_t),\tilde{f}_t(\mathbf{x}_t))\\&=\mathcal{L}\left(\sum_{i=1}^P\hat{p}_t(i)\hat{f}_{t,i}(\mathbf{x}_t),\tilde{f}_t(\mathbf{x}_t)\right)\\&\le\sum_{i=1}^P\hat{p}_t(i)\mathcal{L}(\hat{f}_{t,i}(\mathbf{x}_t),\tilde{f}_t(\mathbf{x}_t))\\&\le\sum_{j=1}^P\tilde{p}_t(j)\left(\sum_{i=1}^P\hat{p}_t(i)\mathcal{L}(\hat{f}_{t,i}(\mathbf{x}_t),\hat{f}_{t,j}(\mathbf{x}_t))\right)\le\eta_c\end{aligned}$$

where the last inequality follows from the confidence condition in (13). This completes the proof.

We next prove Lemma 5. Since $a_t=0$, we have $\hat{p}_{t+1}(i)=\hat{p}_t(i)$ and $\hat{f}_{t+1,i}(\mathbf{x})=\hat{f}_{t,i}(\mathbf{x})$ for all $i\in[P]$. From them, we have

$$\begin{aligned}&\mathcal{L}(\hat{f}_{t+1}(\mathbf{x}_{t+1}),\tilde{f}_{t+1}(\mathbf{x}_{t+1}))\\&=\mathcal{L}\left(\sum_{i=1}^P\hat{p}_t(i)\hat{f}_{t,i}(\mathbf{x}_{t+1}),\tilde{f}_{t+1}(\mathbf{x}_{t+1})\right)\\&\le\sum_{j=1}^P\tilde{p}_{t+1}(j)\left(\sum_{i=1}^P\hat{p}_t(i)\mathcal{L}(\hat{f}_{t,i}(\mathbf{x}_{t+1}),\hat{f}_{t,j}(\mathbf{x}_{t+1}))\right)\le\eta_cB\end{aligned}$$

where the last inequality is due to $a_t=0$ and the assumption (a4).

Appendix C
Proof of Lemma 6

Let $\mathcal{A}=\{t\in[T]:a_t=1\}$ be the index set of the revealed labels. Then, we have

$$\begin{aligned}\text{regret}_T^{\text{a}}&=\sum_{t=1}^T\mathcal{L}(\hat{f}_t(\mathbf{x}_t),y_t)-\mathcal{L}(\tilde{f}_t(\mathbf{x}_t),y_t)\\&\overset{(a)}{\le}\sum_{t=1}^T\mathcal{L}(\hat{f}_t(\mathbf{x}_t),\tilde{f}_t(\mathbf{x}_t))\\&=\sum_{t\in\mathcal{A}}\mathcal{L}(\hat{f}_t(\mathbf{x}_t),\tilde{f}_t(\mathbf{x}_t))+\sum_{t\in\mathcal{A}^c}\mathcal{L}(\hat{f}_t(\mathbf{x}_t),\tilde{f}_t(\mathbf{x}_t))\\&\overset{(b)}{\le}\sum_{t\in\mathcal{A}}\mathcal{L}(\hat{f}_t(\mathbf{x}_t),\tilde{f}_t(\mathbf{x}_t))+\eta_c|\mathcal{A}^c|\end{aligned}\tag{45}$$

where (a) is due to the assumption (a5) (i.e., triangle inequality) and (b) follows from Lemma 4. In the remaining part of this proof, we will show that he first term in (45) is also bounded by η_cB. Consider an arbitrary time index $t\in\mathcal{A}$ with $t_1<t<t_2$ for $t_1,t_2\in\mathcal{A}^c$. From Lemma 5, for $t=t_1+1$, we obtain the following upper bound as $\mathcal{L}(\hat{f}_{t_1+1}(\mathbf{x}_{t_1+1}),\tilde{f}_{t_1+1}(\mathbf{x}_{t_1+1}))\le\eta_cB$. In addition, for $t_1+1<t<t_2$ and any fixed value $\mathbf{x}_t$, we have

$$\mathcal{L}(\hat{f}_t(\mathbf{x}_t),\tilde{f}_t(\mathbf{x}_t))\overset{(a)}{\le}\mathcal{L}(\hat{f}_{t_1+1}(\mathbf{x}_t),\tilde{f}_{t_1+1}(\mathbf{x}_t))\overset{(b)}{\le}\eta_cB\tag{46}$$

where (a) is because the difference between $\hat{f}_t(\mathbf{x})$ and $\tilde{f}_t(\mathbf{x})$ is smaller as $t_1+1<t<t_2$ increases, i.e., the labeling makes them closer, and (b) is from Lemma 5. From this analysis, we have

$$\sum_{t\in\mathcal{A}}\mathcal{L}(\hat{f}_t(\mathbf{x}_t),\tilde{f}_t(\mathbf{x}_t))\le\eta_cB|\mathcal{A}|.\tag{47}$$

From (45) and (47), we can get $\text{regret}_T^{\text{a}}\le T\eta_cB$ and setting $\eta_c=\mathcal{O}(1/\sqrt{T})$, the proof is completed.

Acknowledgment

The authors would like to thank Prof. Y. Shen for encouraging feedback and sharing multiple kernel learning simulations.

References

[1] B. Scholkopf and A. J. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[2] J. Shawe-Taylor *et al.*, *Kernel Methods for Pattern Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[3] Y.-Y. Lin, T.-L. Liu, and C.-S. Fuh, "Multiple kernel learning for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 6, pp. 1147–1160, Jun. 2011.
[4] B. Dai, N. He, Y. Pan, B. Boots, and L. Song, "Learning from conditional distributions via dual embeddings," 2016, *arXiv:1607.04579*. [Online]. Available: http://arxiv.org/abs/1607.04579
[5] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet, "SimpleMKL," *J. Mach. Learn. Res.*, vol. 9, pp. 2491–2521, Nov. 2008.
[6] C. Cortes, M. Mohri, and A. Rostamizadeh, "L2 regularization for learning kernels," 2012, *arXiv:1205.2653*. [Online]. Available: http://arxiv.org/abs/1205.2653
[7] M. Gönen and E. Alpaydın, "Multiple kernel learning algorithms," *J. Mach. Learn. Res.*, vol. 12, pp. 2211–2268, Jul. 2011.
[8] J. A. Bazerque and G. B. Giannakis, "Nonparametric basis pursuit via sparse kernel-based learning: A unifying view with advances in blind methods," *IEEE Signal Process. Mag.*, vol. 30, no. 4, pp. 112–125, Jul. 2013.
[9] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Identifying suspicious URLs: An application of large-scale online learning," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 681–688.
[10] C. Richard, J. C. M. Bermudez, and P. Honeine, "Online prediction of time series data with kernels," *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 1058–1067, Mar. 2009.
[11] J. Kivinen, A. J. Smola, and R. C. Williamson, "Online learning with kernels," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2165–2176, Aug. 2004.
[12] D. Sahoo, S. C. H. Hoi, and B. Li, "Online multiple kernel regression," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 293–302.
[13] S. Bubeck, "Introduction to online optimization," Lecture Notes, 2011, vol. 2.
[14] Y. Shen, T. Chen, and G. B. Giannakis, "Random feature-based online multi-kernel learning in environments with unknown dynamics," *J. Mach. Learn. Res.*, vol. 20, no. 1, pp. 773–808, 2019.
[15] G. Wahba, *Spline Models for Observational Data*, vol. 59. Philadelphia, PA, USA: SIAM, 1990.
[16] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1177–1184.
[17] X. Zhu, J. Lafferty, and R. Rosenfeld, "Semi-supervised learning with graphs," Ph.D. dissertation, School Lang. Technol. Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, 2005.
[18] B. Settles, M. Craven, and L. Friedland, "Active learning with real annotation costs," in *Proc. NIPS Workshop Cost-Sensitive Learn.*, Vancouver, BC, Canada, 2008, pp. 1–10.
[19] A. Bordes, S. Ertekin, J. Weston, and L. Bottou, "Fast kernel classifiers with online and active learning," *J. Mach. Learn. Res.*, vol. 6, pp. 1579–1619, Oct. 2005.
[20] B. Settles, "Active learning literature survey," Dept. Comput. Sci., Univ. Wisconsin-Madison, Madison, WI, USA, Tech. Rep. 1648, Jan. 2010.
[21] M. Sugiyama and S. Nakajima, "Pool-based active learning in approximate linear regression," *Mach. Learn.*, vol. 75, no. 3, pp. 249–274, Jun. 2009.
[22] A. K. McCallumzy and K. Nigamy, "Employing EM and pool-based active learning for text classification," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 1998, pp. 359–367.
[23] J. Smailović, M. Grčar, N. Lavrač, and M. Žnidaršič, "Stream-based active learning for sentiment analysis in the financial domain," *Inf. Sci.*, vol. 285, pp. 181–203, Nov. 2014.
[24] D. Wu, "Pool-based sequential active learning for regression," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1348–1359, May 2019.
[25] I. Dagan and S. P. Engelson, "Committee-based sampling for training probabilistic classifiers," in *Machine Learning Proceedings*. Amsterdam, The Netherlands: Elsevier, 1995, pp. 150–157.
[26] V. Krishnamurthy, "Algorithms for optimal scheduling and management of hidden Markov model sensors," *IEEE Trans. Signal Process.*, vol. 50, no. 6, pp. 1382–1397, Jun. 2002.
[27] H. Yu, "SVM selective sampling for ranking with application to data retrieval," in *Proc. 11th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2005, pp. 354–363.
[28] I. Zliobaite, A. Bifet, B. Pfahringer, and G. Holmes, "Active learning with drifting streaming data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 1, pp. 27–39, Jan. 2014.
[29] S. Hao, P. Hu, P. Zhao, S. C. H. Hoi, and C. Miao, "Online active learning with expert advice," *ACM Trans. Knowl. Discovery Data*, vol. 12, no. 5, pp. 1–22, Jul. 2018.
[30] W. M. Koolen, T. Van Erven, and P. Grünwald, "Learning the learning rate for prediction with expert advice," in *Adv. Neural Inf. Process. Syst.*, 2014, pp. 2294–2302.
[31] S. Wassermann, T. Cuvelier, and P. Casas, "RAL—Improving stream-based active learning by reinforcement learning," in *Proc. Eur. Conf. Mach. Learn. Princ. Pract. Knowl. Discovery Database, Workshop Iterative Adapt. Learn.*, vol. 2444, 2019, pp. 32–47. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02265426
[32] H. S. Jomaa, J. Grabocka, and L. Schmidt-Thieme, "Hyp-RL: Hyperparameter optimization by reinforcement learning," 2019, *arXiv:1906.11527*. [Online]. Available: http://arxiv.org/abs/1906.11527
[33] B. Schoelkopf and A. J. Smola, *Learning With Kernels*. Cambridge, MA, USA: MIT Press, 2002.
[34] C. A. Micchelli and M. Pontil, "Learning the kernel function via regularization," *J. Mach. Learn. Res.*, vol. 6, pp. 1099–1125, Jul. 2005.
[35] E. Hazan, "Introduction to online convex optimization," *Found. Trends Optim.*, vol. 2, nos. 3–4, pp. 157–325, 2016.
[36] M. J. Wainwright, *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*, vol. 48. Cambridge, U.K.: Cambridge Univ. Press, 2019.
[37] F. Kawala, A. Douzal-Chouakria, E. Gaussier, and E. Dimert, "Prédictions d'activité dans les réseaux sociaux en ligne," in *Proc. 4th Conf. Sur Les Modèles l'Analyse Réseaux Approches Math. Inform.*, 2013, pp. 16–28.
[38] S. De Vito, E. Massera, M. Piga, L. Martinotto, and G. Di Francia, "On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario," *Sens. Actuators B, Chem.*, vol. 129, no. 2, pp. 750–757, Feb. 2008.
[39] L. M. Candanedo, V. Feldheim, and D. Deramaix, "Data driven prediction models of energy use of appliances in a low-energy house," *Energy Buildings*, vol. 140, pp. 81–97, Apr. 2017.
[40] A. Coraddu, L. Oneto, A. Ghio, S. Savio, D. Anguita, and M. Figari, "Machine learning approaches for improving condition-based maintenance of naval propulsion plants," *Inst. Mech. Eng. M, J. Eng. Maritime Environ.*, vol. 230, no. 1, pp. 136–153, Jul. 2014.
[41] D. Bacciu, P. Barsocchi, S. Chessa, C. Gallicchio, and A. Micheli, "An experimental characterization of reservoir computing in ambient assisted living applications," *Neural Comput. Appl.*, vol. 24, no. 6, pp. 1451–1464, May 2014.
[42] P. C.-S. Jason Lines, A. Bagnall, and S. Anderson, "Classification of household devices by electricity usage profiles," in *Proc. Int. Conf. Intell. Data Eng. Automated Learn.*, 2011, pp. 403–412.
[43] L. O.-X. P. Davide Anguita, A. Ghio, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," in *Proc. Eur. Symp. Artif. Neural Netw.*, 2013, pp. 1–3.
[44] *Metro Interstate Traffic Volume Data Set*. Accessed: Jul. 5, 2019. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Metro+Interstate+Traffic+Volume

Songnam Hong (Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical and computer engineering from Hanyang University, Seoul, South Korea, in 2003 and 2005, respectively, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 2014.

From 2005 to 2009, he was a Research Engineer with the Telecommunication Research and Development Center, Samsung Electronics, Suwon, South Korea, and from 2014 to 2016, he was a Senior Research Engineer with the Ericsson Research, San Jose, CA, USA. From 2016 to 2020, he was an Assistant/Associate Professor with Ajou University, Suwon. He is currently an Associate Professor with Hanyang University. His main research interests include the areas of large-scale optimization, statistical signal processing, and machine/deep learning.

Jeongmin Chae (Student Member, IEEE) received the M.Sc. degrees in electrical and computer engineering from Ajou University, Suwon, South Korea, in 2020. She is currently pursuing the Ph.D. degree in electrical engineering with the University of Southern California, Los Angeles, CA, USA.

Her main research interests include the areas of machine learning algorithms and statistical signal processing.