Adaptive Ensemble Learning With Confidence Bounds

Cem Tekin, *Member, IEEE*, Jinsung Yoon, and Mihaela van der Schaar, *Fellow, IEEE*

***Abstract*—Extracting actionable intelligence from distributed, heterogeneous, correlated, and high-dimensional data sources requires run-time processing and learning both locally and globally. In the last decade, a large number of meta-learning techniques have been proposed in which local learners make online predictions based on their locally collected data instances, and feed these predictions to an ensemble learner, which fuses them and issues a global prediction. However, most of these works do not provide performance guarantees or, when they do, these guarantees are asymptotic. None of these existing works provide confidence estimates about the issued predictions or rate of learning guarantees for the ensemble learner. In this paper, we provide a systematic ensemble learning method called Hedged Bandits, which comes with both long-run (asymptotic) and short-run (rate of learning) performance guarantees. Moreover, our approach yields performance guarantees with respect to the optimal local prediction strategy, and is also able to adapt its predictions in a data-driven manner. We illustrate the performance of Hedged Bandits in the context of medical informatics and show that it outperforms numerous online and offline ensemble learning methods.**

***Index Terms*—Ensemble learning, meta-learning, online learning, regret, confidence bound, multi-armed bandits, contextual bandits, medical informatics.**

I. Introduction

HUGE amounts of data streams are now being produced by more and more sources and in increasingly diverse formats: sensor readings, physiological measurements, GPS events, network traffic information, documents, emails, transactions, tweets, audio files, videos etc. These streams are then mined in real-time to provide actionable intelligence for a variety of applications: patient monitoring [2], recommendation systems [3], social networks [4], targeted advertisement [5], network security [6], [7], medical diagnosis [8] etc. Hence, online data mining algorithms have emerged that analyze the correlated, high-dimensional and dynamic data instances captured by one or multiple heterogeneous data sources, extract actionable intelligence from these instances and make decisions in real-time. To mine these data streams, the following questions need to be answered online, for each data instance: Which processing/prediction/decision rule should a *local learner* (LL) select? How should the LLs adapt and learn their rules to maximize their performance? How should the processing/predictions/decisions of the LLs be combined/fused by a meta-learner to maximize the overall performance?

Existing works on meta-learning [6], [9]–[11] have aimed to provide solutions to these questions by designing *ensemble learners* (ELs) that fuse the predictions[1] made by the LLs into global predictions. A majority of the literature treats the LLs as black box algorithms, and proposes various fusion algorithms for the EL with the goal of issuing predictions that are at least as good as the best LL in terms of prediction accuracy. In some of these works, the obtained result holds for any arbitrary sequence of data instance-label pairs, including the ones generated by an adaptive adversary. However, the performance bounds proved for the EL in these papers depend on the performance of the LLs. In this work, we go one step further and study the joint design of learning algorithms for both the LLs and the EL. Our approach also differs from *empirical risk minimization* (ERM) based approaches [12], [13]. Firstly, most of the literature on ERM is concerned with finding the best prediction rule on average. We depart from this approach and seek to find the best context-dependent prediction rule. Secondly, data is not available a priori in our model. Predictions are made on-the-fly based on the prediction rules chosen by the learning algorithm. This results in a trade-off between exploration and exploitation, which is not present in ERM.

In this paper, we present a novel learning method which continuously learns and adapts the parameters of both the LLs and the EL, after each data instance, in order to achieve strong performance guarantees - both confidence bounds and regret bounds. We call the proposed method *Hedged Bandits* (HB). The proposed system consists of a new contextual bandit algorithm for the LLs and two new variants of the Hedge algorithm [11] for the EL. The proposed method is able to exploit the adversarial regret guarantees of Hedge and the data-dependent regret guarantees of the contextual bandit algorithm to derive regret bounds for the EL. One proposed variant of the Hedge algorithm does not require the knowledge of time horizon T and achieves the $O(\sqrt{T \log M})$ on regret uniformly over time, where M is the number of LLs. The other variant uses the

Manuscript received December 22, 2015; revised July 1, 2016 and October 19, 2016; accepted October 20, 2016. Date of publication November 8, 2016; date of current version December 5, 2016. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Gonzalo Mateos. The work of C. Tekin was supported by the Scientific and Technological Research Council of Turkey (TUBITAK) under 2232 Fellowship 116C043. The work of J. Yoon and M. van der Schaar was supported by the NSF EECS 1407712. This paper was presented in part at the 2016 AAAI Workshop on Expanding the Boundaries of Health Informatics Using AI, Phoenix, AZ, USA, Feb. 2016 [1].

C. Tekin is with the Department of Electrical and Electronics Engineering, Bilkent University, Ankara 06800, Turkey (e-mail: cemtekin@ee.bilkent.edu.tr).

J. Yoon and M. van der Schaar are with the Department of Electrical Engineering, University of California, Los Angeles, CA 90095 USA (e-mail: jsyoon0823@gmail.com; mihaela@ee.ucla.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2016.2626250

[1] Throughout this paper the term prediction is used to denote a variety of tasks from making predictions to taking actions.

context/side information provided to the EL to fuse the predictions of the LLs.

The contributions of this paper are:

- We propose two variants of the Hedge algorithm [11]. The first variant, which is called Anytime Hedge (AH), is a parameter-free Hedge algorithm [14]–[17]. We prove that AH enjoys the same order of regret as the original Hedge [11]. The second variant, which is called Contextual Hedge (CH), is novel and uses the context information provided to the EL when fusing the LLs' predictions. Since the sequence of context arrivals to the EL are not known in advance, CH utilizes AH to learn the best LL for each context.
- We propose a new index-based learning rule for each LL, called Instance-based Uniform Partitioning (IUP). We prove an optimal regret bound for IUP, which holds for any sequence of data instance arrivals to the LL, and hence, also in expectation.
- We prove confidence bounds for each LL with respect to the optimal data-dependent prediction rule of that LL.
- Using the regret bounds proven for each LL and the EL, we prove a regret bound for the EL with respect to the optimal data-dependent prediction rule.
- We numerically compare IUP, AH and CH with state-of-the-art machine learning methods in the context of medical informatics and show the superiority of the proposed methods.

II. Problem Description

This section describes the system model and introduces the notation. $\mathrm{I}(\cdot)$ is the indicator function, $\mathrm{E}[\cdot]$ is the expectation operator. $\mathrm{E}_P[\cdot]$ denotes the expectation of a random variable with respect to distribution P. Given a set $\mathcal{S}$, $\Delta(\mathcal{S})$ denotes the set of probability distributions over $\mathcal{S}$ and $|\mathcal{S}|$ denotes the cardinality of $\mathcal{S}$. For a scalar or vector $z(t)$ indexed by $t \in \mathbb{N}^+ := \{1, 2, \ldots\}$, $\boldsymbol{z}^T := (z(1), \ldots, z(T))$. Given a vector $\boldsymbol{v}$, $\boldsymbol{v}_{-i}$ is the vector formed by the components of $\boldsymbol{v}$ except the ith component. Random variables are denoted by uppercase letters. Realizations of random variables are denoted by lowercase letters.

The system model is given in Fig. 1. There are M LLs indexed by the set $\mathcal{M} := \{1, 2, \ldots, M\}$. Each LL receives streams of data instances, sequentially, over discrete time steps $t \in \{1, 2, \ldots\}$. The instance received by LL i at time t is denoted by $X_i(t)$. Without loss of generality, we assume that $X_i(t)$ is a d_i-dimensional vector in $\mathcal{X}_i := [0, 1]^{d_i}$.[2] Let $\mathcal{X} := \prod_{i \in \mathcal{M}} \mathcal{X}_i$ denote the *joint data instance set*.

The collection of data instances at time t is denoted by $\boldsymbol{X}(t) = \{X_i(t)\}_{i \in \mathcal{M}}$. For example, $\boldsymbol{X}(t)$ can include in a medical diagnosis application real-valued features such as lab test results; discrete features such as age and number of previous conditions; and categorical features such as gender, smoker/non-smoker, etc. In this example each LL corresponds to a (different) medical expert. The true label at time t is denoted by $Y(t)$, which is a random variable that takes values in the finite label set $\mathcal{Y}$. Let J denote the joint distribution of $(\boldsymbol{X}(t), Y(t))$. It is assumed that $\{(\boldsymbol{X}(t), Y(t))\}_{t=1}^T$ is i.i.d. Let $J^i_{x_i}$ denote the conditional distribution of $(\boldsymbol{X}_{-i}(t), Y(t))$ given $X_i(t) = x_i$.

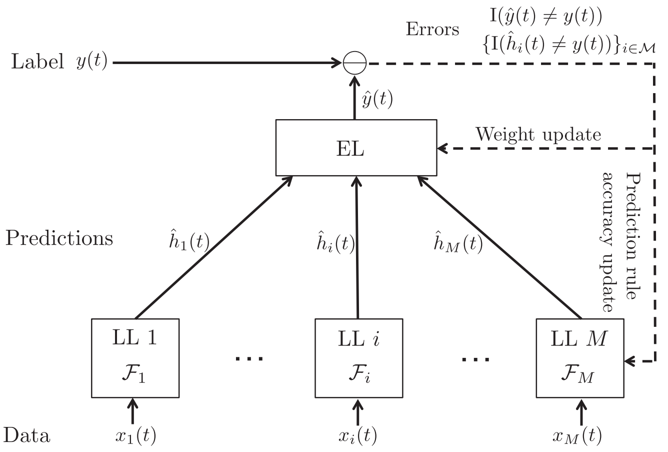

Fig. 1. Block diagram of the HB. The flow of information towards the EL is illustrated via a tree graph, where the LLs are the leaf nodes. After observing the instance, each LL selects one of its prediction rules to produce a prediction, and sends its prediction to the EL which makes the final prediction. Then, both the LLs and the EL update their prediction policies based on the received feedback $y(t)$. Note that the EL only observes the predictions $\hat{\boldsymbol{h}}(t)$ of the LLs but not their instances $\boldsymbol{x}(t)$.

The set of prediction rules of LL i is denoted by $\mathcal{F}_i$. For instance, a prediction rule can be a classifier such as an SVM with polynomial kernel, a neural network or a decision tree. Let $\mathcal{F} := \cup_{i \in \mathcal{M}} \mathcal{F}_i$ denote the set of all prediction rules. The prediction produced by $f \in \mathcal{F}_i$ given context $x_i \in \mathcal{X}_i$ is denoted by $Y_f(x_i)$. $Y_f(x_i)$ is a random variable whose distribution is given by $Q_f(x_i)$, where $Q_f : \mathcal{X}_i \to \Delta(\mathcal{Y})$. Prediction of $f \in \mathcal{F}_i$ at time t is denoted by $\hat{Y}_f(t) := Y_f(X_i(t))$. Let $\boldsymbol{Z}(t) := (\boldsymbol{X}(t), Y(t), \{\hat{Y}_f(t)\}_{f \in \mathcal{F}})$. Then, $\{\boldsymbol{Z}(t)\}_{t=1}^T$ is an i.i.d. sequence. Realizations of the random variables $X_i(t), Y(t)$ and $\hat{Y}_f(t)$ are denoted by $x_i(t)$, $y(t)$ and $\hat{y}_f(t)$, respectively. The accuracy of prediction rule $f \in \mathcal{F}_i$ for a data instance $x \in \mathcal{X}_i$ is given as

$$\pi_f(x) := \mathrm{E}\left[\mathrm{I}(\hat{Y}_f(t) = Y(t)) | X_i(t) = x\right].$$

LL i operates as follows: It first observes $x_i(t)$, and then selects a prediction rule $a_i(t) \in \mathcal{F}_i$. The selected prediction rule produces a prediction $\hat{h}_i(t) = \hat{y}_{a_i(t)}(t)$.[3] Then, all LLs send their predictions $\hat{\boldsymbol{h}}(t) := \{\hat{h}_i(t)\}_{i \in \mathcal{M}}$ to the EL, which combines them to produce a final prediction $\hat{y}(t)$. We assume that the true label $y(t)$ is revealed after the final prediction, by which the LLs and the EL can update their prediction rule selection

[2] The unit hypercube is just used for notational simplicity. Our methods can easily be generalized to arbitrary bounded, finite dimensional data spaces, including spaces of categorical variables.

[3] Without loss of generality we assume that only the selected prediction rule produces a prediction. For instance, in big data stream mining, the LL may be resource constrained and require to make timely predictions. The LL in this setting is constrained to activate only one of its prediction rules for each data instance. Moreover, observing the predictions of more than one prediction rule will result in faster learning. Hence, all our performance bounds will still hold when the LL observes the predictions of all of its prediction rules.

strategy, which is a mapping from the history of past observations, decisions, and the current instance to the set of prediction rules. We call $r_f(t) = \mathrm{I}(\hat{y}_f(t) = y(t))$ the reward of prediction rule f, $v_i(t) := \mathrm{I}(\hat{h}_i(t) = y(t))$ the reward of LL i and $r_{\mathrm{EL}}(t) = \mathrm{I}(\hat{y}(t) = y(t))$ the reward of the EL at time t. Random variables that correspond to the realizations $a_i(t)$, $\hat{h}_i(t)$, $r_f(t)$, $v_i(t)$ and $r_{\mathrm{EL}}(t)$ are denoted by $A_i(t)$, $\hat{H}_i(t)$, $R_f(t)$, $V_i(t)$ and $R_{\mathrm{EL}}(t)$, respectively.

In our setup each LL is only required to observe its own data instance and know its own prediction rules. However, the accuracy of the prediction rules is unknown and data dependent. The EL does not know anything about the instances and prediction rules of the LLs.[4] We assume that the accuracy of a prediction rule obeys the following Hölder rule, which represents a similarity measure between different data instances.

Assumption 1: There exists $L > 0$, $\alpha > 0$ such that for all $i \in \mathcal{M}$, $f \in \mathcal{F}_i$, and $x, x' \in \mathcal{X}_i$, we have

$$|\pi_f(x) - \pi_f(x')| \leq L||x - x'||^\alpha.$$

We assume that α is known by the LLs. Going back to our medical informatics example, we can interpret Assumption 1 as follows. If the lab tests, symptoms and demographic information of two patients are similar, it is expected that they have the same underlying medical condition, and hence, the (diagnosis) prediction should be similar for these two patients.

III. Performance Metrics: Regret

In this section, we introduce several performance metrics to assess the performance of the learning algorithms of the LLs and the EL. First, we define the performance measures for the LLs. We start by defining the optimal prediction rules and *local oracles* (LOs) that implement these prediction rules. Let $f_i^*(x)$ be the optimal prediction rule of LL i for an instance $x \in \mathcal{X}_i$, which is given by $f_i^*(x) \in \arg\max_{f \in \mathcal{F}_i} \pi_f(x)$. The accuracy of $f_i^*(x)$ is denoted by $\pi_i^*(x) := \pi_{f_i^*(x)}(x)$.

LO i knows $\{\pi_f(\cdot)\}_{f \in \mathcal{F}_i}$ perfectly. At each time step t it observes $x_i(t)$ and then selects $f_i^*(x_i(t))$ to make a prediction. Since LL i does not know $\{\pi_f(\cdot)\}_{f \in \mathcal{F}_i}$ a priori, we would like to measure how well it performs with respect to LO i. For this, we define the *data-dependent regret* of LL i with respect to LO i as

$$\mathrm{Reg}_i(T) := \sum_{t=1}^{T} R_{f_i^*(X_i(t))}(t) - \sum_{t=1}^{T} R_{A_i(t)}(t).$$

The strategy of LO i only depends on $\boldsymbol{X}_i^T = (X_i(1), \ldots, X_i(T))$. Thus, we would like to measure how well LL i performs given $\boldsymbol{X}_i^T$. For this, we define the *conditional regret* of LL i as

$$\mathrm{Reg}_i(T|\boldsymbol{X}_i^T) := \mathrm{E}\left[\mathrm{Reg}_i(T)|\boldsymbol{X}_i^T\right] \tag{1}$$

The algorithm we propose in Section IV *almost surely* (a.s.) upper bounds the conditional regret with a deterministic sublinear function of time. The *expected regret* of LL i is defined as

$$\begin{aligned}\overline{\mathrm{Reg}}_i(T) &:= \mathrm{E}\left[\mathrm{Reg}_i(T)\right] \\ &= \mathrm{E}\left[\mathrm{E}\left[\mathrm{Reg}_i(T)|\boldsymbol{X}_i^T\right]\right] = \mathrm{E}\left[\mathrm{Reg}_i(T|\boldsymbol{X}_i^T)\right].\end{aligned}$$

This implies that a deterministic upper bound on $\mathrm{Reg}_i(T|\boldsymbol{X}_i^T)$ that holds a.s. also holds for $\overline{\mathrm{Reg}}_i(T)$.

Next, we define the performance measures for the EL. Consider any realization $\{\boldsymbol{v}_i^T\}_{i \in \mathcal{M}}$ of the random reward sequence $\{\boldsymbol{V}_i^T\}_{i \in \mathcal{M}}$ of the LLs. The best LL for this realization is defined as I_b, where $I_b \in \arg\max_{i \in \mathcal{M}} \sum_{t=1}^{T} v_i(t)$. In Section V, we propose a learning algorithm for the EL, whose total reward is close to the total reward of I_b for any realization $\{\boldsymbol{v}_i^T\}_{i \in \mathcal{M}}$. To measure the distance between total rewards, we define the *pseudo-regret* of the EL given $\{\boldsymbol{v}_i^T\}_{i \in \mathcal{M}}$ as

$$\mathrm{Reg}_{\mathrm{EL}}(T) := \sum_{t=1}^{T} v_{I_b}(t) - \mathrm{E}\left[\sum_{t=1}^{T} R_{\mathrm{EL}}(t)\right] \tag{2}$$

where the expectation is taken with respect to the randomization of the EL. In Section V we bound $\mathrm{Reg}_{\mathrm{EL}}(T)$ by a sublinear function of T, which implies that $\lim_{T\to\infty} \mathrm{Reg}_{\mathrm{EL}}(T)/T = 0$. $\mathrm{Reg}_{\mathrm{EL}}(T)$ compares the performance of the EL with the best LL, which makes it a relative performance measure. This is the standard approach taken in prior works in ensemble learning [11], [18]. Since LLs themselves are learning agents, $\mathrm{Reg}_{\mathrm{EL}}(T)$ depends on the learning algorithms used by the LLs. Next, we propose a benchmark for the performance measure of the EL that is independent of the learning algorithms used by the LLs.

The optimal LO denoted by i^*, is given as $i^* \in \arg\max_{i \in \mathcal{M}} \mathrm{E}[\sum_{t=1}^{T} R_{f_i^*(X_i(t))}(t)]$. LO i^*'s total predictive accuracy is greatest among all LOs. On the other hand, the best LL in expectation is defined as $i_b^* \in \arg\max_{i \in \mathcal{M}} \mathrm{E}[\sum_{t=1}^{T} R_{A_i(t)}(t)]$. We would like to emphasize the fact that the expected reward of LL i depends on the learning algorithm used by the LL, while the expected reward of LO i is the optimal that can be achieved given the prediction rules in $\mathcal{F}_i$. Hence, the latter upper bounds the former. This implies that $\mathrm{E}[\sum_{t=1}^{T} R_{A_{i_b^*}(t)}(t)] \leq \mathrm{E}[\sum_{t=1}^{T} R_{f_{i^*}^*(X_{i^*}(t))}(t)]$. As an absolute measure of performance we define the *expected regret* of the EL as

$$\overline{\mathrm{Reg}}_{\mathrm{EL}}(T) := \mathrm{E}\left[\sum_{t=1}^{T} R_{f_{i^*}^*(X_{i^*}(t))}(t)\right] - \mathrm{E}\left[\sum_{t=1}^{T} R_{\mathrm{EL}}(t)\right] \tag{3}$$

which compares the EL with the best LO in terms of the expected reward.

Our goal is to jointly design algorithms for the LLs and the EL that minimize the learning loss (i.e. the growth rate of $\overline{\mathrm{Reg}}_{\mathrm{EL}}(T)$). This can be viewed equivalently as maximizing the learning speed/rate of the LL and the EL algorithms. We will prove in the Section VI a sublinear upper bound on $\overline{\mathrm{Reg}}_{\mathrm{EL}}(T)$, meaning that the proposed algorithms have a provably fast rate of learning, and the *average regret* $\overline{\mathrm{Reg}}_{\mathrm{EL}}(T)/T$ of the proposed algorithms converges asymptotically to 0. A learning algorithm that achieves sublinear regret guarantees that (in expectation) the number of prediction errors it makes is in the order of that of

[4] We consider the case when the EL has access to a subset of the features of the instances in Section VII, and propose a learning algorithm for this case.

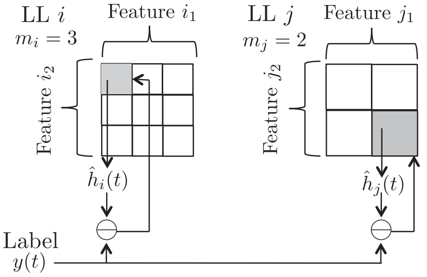

Fig. 2. Illustration of different partitions used by IUP for LLs i and j. Accuracy parameter is updated for the shaded sets in the partitions, which contains the current feature vector.

```
IUP for LL $i$:
  Input: $T$, $m_i$, $d_i$
  Initialize sets: Create partition $\mathcal{P}_i$ of $[0,1]^{d_i}$ into $m_i^{d_i}$
  identical hypercubes
  Initialize counters: $N^i_{f,p} = 0, \forall f \in \mathcal{F}_i, p \in \mathcal{P}_i$, $t = 1$
  Initialize estimates: $\hat{\pi}^i_{f,p} = 0$, $\forall f \in \mathcal{F}_i$, $p \in \mathcal{P}_i$
  while $t \geq 1$ do
    Find the set $p^* = p_i(t) \in \mathcal{P}_i$ that $x_i(t)$ belongs to
    Compute the index for each $f \in \mathcal{F}_i$:
    for $f \in \mathcal{F}_i$ do
      if $N^i_{f,p^*} > 0$ then
        $g^i_{f,p^*} = \hat{\pi}^i_{f,p^*} + \sqrt{\frac{2}{N^i_{f,p^*}}(1 + 2\log(2|\mathcal{F}_i| m_i^{d_i} T^{\frac{3}{2}}))}$
      else
        $g^i_{f,p^*} = +\infty$
      end if
    end for
    Select $a_i = \arg\max_{f \in \mathcal{F}_i} g^i_{f,p^*}$ (break ties randomly)
    Predict $\hat{h}_i(t) = \hat{y}_{a_i}(t)$
    Observe the true label $y(t)$ and the reward
    $v_i(t) = \mathrm{I}(\hat{h}_i(t) = y(t))$
    $\hat{\pi}^i_{a_i,p^*} \leftarrow (\hat{\pi}^i_{a_i,p^*} N^i_{a_i,p^*} + v_i(t))/(N^i_{a_i,p^*} + 1)$
    $N^i_{a_i,p^*} \leftarrow N^i_{a_i,p^*} + 1$
    $t \leftarrow t + 1$
  end while
```

Fig. 3. Pseudocode of IUP for LL i.

the optimal LO, which knows the accuracies of the prediction rules for each instance in advance.

IV. An Instance-Based Uniform Partitioning Algorithm for the LLs

Each LL uses the *Instance-based Uniform Partitioning* (IUP) algorithm given in Fig. 3. IUP is designed to exploit the similarity measure given in Assumption 1 when learning the accuracies of the prediction rules. Basically, IUP partitions $\mathcal{X}_i$ into a finite number of equal sized, identically shaped, non-overlapping sets, whose granularities determine the balance between approximation accuracy and estimation accuracy: increasing the size of a set in the partition results in more past instances falling within that set, which positively affects the estimation accuracy, but also allows more dissimilar instances to lie in the same set, which negatively affects the approximation accuracy. IUP strikes this balance by adjusting the granularity of the data space partition based on the information contained within the similarity measure (Assumption 1) and the time horizon T.[5]

Let m_i be the *partitioning parameter* of LL i, which is used to partition $[0,1]^{d_i}$ into $m_i^{d_i}$ identical hypercubes. This partition is denoted by $\mathcal{P}_i$.[6] IUP estimates the accuracy of each prediction rule for each set (hypercube) $p \in \mathcal{P}_i$, separately, by only using the past history from instance arrivals that fall into hypercube p. For each LL i, IUP keeps and updates the following parameters during its operation:

- $N^i_{f,p}(t)$: Number of times an instance arrived to hypercube $p \in \mathcal{P}_i$ and prediction rule f of LL i is used to make the prediction prior to time t.
- $\hat{\pi}^i_{f,p}(t)$: Sample mean accuracy of prediction rule $f \in \mathcal{F}_i$ at time t.

An illustration of the partitions used by IUP for each LL is given in Fig. 2. IUP strikes the balance between exploration and exploitation by keeping the following set of indices for each $p \in \mathcal{P}_i$ and $f \in \mathcal{F}_i$:[7]

$$g^i_{f,p}(t) = \hat{\pi}^i_{f,p}(t) + \sqrt{\frac{2}{N^i_{f,p}(t)}(1 + 2\log(2|\mathcal{F}_i| m_i^{d_i} T^{\frac{3}{2}}))}. \quad (4)$$

The second term in (4) is an inflation term that decreases with the square root of $N^i_{f,p}(t)$. The $(1 + 2\log(2|\mathcal{F}_i| m_i^{d_i} T^{3/2})$ term is a normalization constant that is required for the regret analysis in Theorem 1. These types of indices are commonly used in online learning [20] to tradeoff exploration and exploitation.

At the beginning of time step t, LL i observes $x_i(t)$, and identifies the hypercube $p_i(t) \in \mathcal{P}_i$ that contains $x_i(t)$. Then, it selects $a_i(t) \in \arg\max_{f \in \mathcal{F}_i} g^i_{f,p_i(t)}(t)$ and predicts $\hat{h}_i(t) = \hat{y}_{a_i(t)}(t)$. The second term of the index reflects the uncertainty in the estimated value $\hat{\pi}^i_{f,p}(t)$. It decreases as more observations are gathered from prediction rule f for data instances that lie in p. Hence, $g^i_{f,p}(t)$ serves as an optimistic estimate of the accuracy of f for data instances in p. LL i explores when $a_i(t) \notin \arg\max_{f \in \mathcal{F}_i} \hat{\pi}^i_{f,p_i(t)}(t)$, and exploits when $a_i(t) \in \arg\max_{f \in \mathcal{F}_i} \hat{\pi}^i_{f,p_i(t)}(t)$. In exploration, it chooses a prediction rule with suboptimal estimated accuracy and high uncertainty, while in exploitation it chooses the prediction rule with the highest estimated accuracy. In Section VI, we will show that the choice of the index in (4) results in optimal learning.

V. Anytime Hedge Algorithm for the EL

In this section, we consider a parameter-free variant of the Hedge algorithm, called the *Anytime Hedge* (AH), whose

[5] The doubling trick [19] allows any learning algorithm Γ that requires the time horizon as an input to run efficiently (with the same time order of regret) without the knowledge of the time horizon. With the doubling trick, time is partitioned into multiple phases ($j = 1, 2, \ldots$) with doubling lengths ($T_1, T_2, \ldots$). For instance, if the first phase is set to last for $\hat{T}$ time steps, then the length of the jth phase is equal to $2^{j-1}\hat{T}$ time steps. In each phase j, an independent instance of the original learning algorithm Γ, denoted by Γ_j, is run from scratch, without using any information available from the previous phases. With the doubling trick, Γ_j's time horizon input is set to $2^{j-1}\hat{T}$. When we run IUP for LL i with the doubling trick, the only modification that is needed is to set the partitioning parameter of phase j to $m_i = \lceil (2^{j-1}\hat{T})^{1/(2\alpha + d_i)} \rceil$.

[6] Instances laying at the edges of the hypercubes can be assigned to one of the hypercubes in a random fashion without affecting the derived performance bounds.

[7] When $N^i_{f,p}(t) = 0$, we set $g^i_{f,p}(t)$ to $+\infty$.

```
Anytime Hedge (AH)
  Input: A non-increasing sequence of positive real numbers
  ${\eta(t)}_{t\in\mathbb{N}+}$
  Initialization: $L_i(0) = 0$ for $i \in \mathcal{M}$, $t = 1$
  while $t \geq 1$ do
    Receive predictions of LLs: $\hat{\boldsymbol{h}}(t)$
    Choose the LL $I(t)$ to follow according to the distribution
    $\boldsymbol{q}(t) := (q_1(t), \ldots, q_M(t))$ where

          $q_i(t) = \frac{\exp(-\eta(t)L_i(t-1))}{\sum_{j=1}^{M}\exp(-\eta(t)L_j(t-1))}$

    Predict $\hat{y}(t) = \hat{h}_{I(t)}(t)$
    Observe the true label $y(t)$
    Receive the reward $r_{\mathrm{EL}}(t) = \mathrm{I}(\hat{y}(t) = y(t))$ and observe
    losses of all LLs: $l_i(t) := \mathrm{I}(\hat{h}_i(t) \neq y(t))$ for $i \in \mathcal{M}$
    Set $L_i(t) = L_i(t-1) + l_i(t)$
    $t \leftarrow t + 1$
  end while
```

Fig. 4. Pseudocode of AH.

pseudocode is given in Fig. 4. Hedge [11] is an algorithm that uses the exponential weights update rule. It achieves $O(\sqrt{T})$ regret under the prediction with expert advice model. In this model, the goal is to compete with the best expert given a pool of experts. Hedge takes as input a parameter η, that is called the *learning rate*. The regret of Hedge is minimized when η is carefully selected according to the time horizon T.

Unlike the original Hedge, AH does not require a priori knowledge of the time horizon. The EL uses AH to produce the final prediction $\hat{y}(t)$.[8] Although, numerous parameter-free variants of Hedge are introduced in prior works [14]–[17], to the best of our knowledge the regret analysis for AH is new. Specifically, in Theorem 2.3 of [17], regret bound for a parameter-free Exponentially Weighted Average Forecaster is derived. However, it is assumed that (i) the prediction of the EL is a deterministic weighted average of the predictions of the LLs, and (ii) the space of predictions and the loss functions are convex. In contrast to this, in our setting (i) the prediction of the EL is probabilistic, and (ii) the space of prediction is a finite set $\mathcal{Y}$ and the loss functions $\mathrm{I}(\hat{h}_i(t) \neq y(t))$ and $\mathrm{I}(\hat{y}(t) \neq y(t))$ are indicator functions.

AH keeps a cumulative loss/error vector $\boldsymbol{L}(t) = (L_1(t), \ldots, L_M(t))$, where $L_i(t)$ denotes the number of prediction errors made by LL i by the end of time step t. After observing $\hat{\boldsymbol{h}}(t)$, AH samples its final prediction $\hat{Y}(t)$ from this set according to probability distribution $\boldsymbol{q}(t) = (q_1(t), \ldots, q_M(t))$, where

$$\Pr(\hat{Y}(t) = \hat{h}_i(t)) = q_i(t) = \frac{\exp(-\eta(t)L_i(t-1))}{\sum_{j=1}^{M}\exp(-\eta(t)L_j(t-1))}$$

where $\{\eta(t)\}_{t\in\mathbb{N}^+}$ is a positive non-increasing sequence. This implies that AH will choose the LLs with smaller cumulative error with higher probability.

VI. Analysis of the Regret

In this section we prove bounds on the regrets given in (1) and (3), when the LLs use IUP and the EL uses AH as their algorithms. The following theorem bounds the regret of each LL.

Theorem 1: Regret bounds for LL i. When LL i uses IUP with the partitioning parameter $m_i \in \mathbb{N}^+$, given $\boldsymbol{X}_i^T = \boldsymbol{x}_i^T$ we have

$$\mathrm{Reg}_i(T|\boldsymbol{X}_i^T = \boldsymbol{x}_i^T) \leq 1 + 2Ld_i^{\alpha/2}m_i^{-\alpha}T + |\mathcal{F}_i|m_i^{d_i} + 2A_{m_i}\sqrt{|\mathcal{F}_i|m_i^{d_i}T}. \quad (5)$$

Specifically, when $m_i = \lceil T^{1/(2\alpha+d_i)} \rceil$, we have

$$\mathrm{Reg}_i(T|\boldsymbol{X}_i^T = \boldsymbol{x}_i^T) \leq T^{\frac{\alpha+d_i}{2\alpha+d_i}}C_i + T^{\frac{d_i}{2\alpha+d_i}}2^{d_i}|\mathcal{F}_i| + 1 \quad (6)$$

where $C_i = 2A_{m_i}|\mathcal{F}_i|^{1/2}2^{d_i/2} + 2Ld_i^{\alpha/2}$ and $A_{m_i} = 2\sqrt{2(1 + 2\log(2|\mathcal{F}_i|m_i^{d_i}T^{\frac{3}{2}}))}$. From (6), it immediately follows that

$$\mathrm{Reg}_i(T|\boldsymbol{X}_i^T) \leq T^{\frac{\alpha+d_i}{2\alpha+d_i}}C_i + T^{\frac{d_i}{2\alpha+d_i}}2^{d_i}|\mathcal{F}_i| + 1 \text{ a.s.}$$

$$\overline{\mathrm{Reg}}_i(T) \leq T^{\frac{\alpha+d_i}{2\alpha+d_i}}C_i + T^{\frac{d_i}{2\alpha+d_i}}2^{d_i}|\mathcal{F}_i| + 1.$$

Proof: See Appendix B. ■

Theorem 1 states that the difference between the expected number of correct predictions made by LO i and IUP increases as a sublinear function of the sample size T. Time order of the terms that appear in (5) are balanced when $m_i = \lceil T^{1/(2\alpha+d_i)} \rceil$. This means that the average excess prediction error of IUP compared to the optimal policy converges to zero as the number of data instances grows (approaches infinity). The regret bound enables us to exactly calculate how far IUP is from the optimal strategy for any finite T, in terms of the average number of correct predictions. Basically, we have $\overline{\mathrm{Reg}}_i(T)/T = \tilde{O}(T^{-\frac{\alpha}{2\alpha+d_i}})$. Moreover the rate of growth of the regret, which is $\tilde{O}(T^{\frac{\alpha+d_i}{2\alpha+d_i}})$ is optimal [21] (up to a logarithmic factor), i.e., there exists no other learning algorithm that can achieve a smaller rate of growth of the regret.

Remark 1: The memory complexity of IUP is $O(|\mathcal{F}_i|m_i^{d_i})$. For $m_i = \lceil T^{1/(2\alpha+d_i)} \rceil$, it becomes $O(|\mathcal{F}_i|T^{d_i/(2\alpha+d_i)})$. For memory bounded LLs, with a bound $M_i \in \mathbb{N}^+$ on the partitioning parameter, we can set $m_i = \min\{\lceil T^{1/(2\alpha+d_i)} \rceil, M_i\}$. In this case, LL i will incur sublinear regret when $\lceil T^{1/(2\alpha+d_i)} \rceil \leq M_i$. Otherwise, the regret may not be sublinear. However, we can still obtain an approximation guarantee for IUP, since $\lim_{T\to\infty} \mathrm{Reg}_i(T|\boldsymbol{X}_i^T)/T = 2Ld_i^{\alpha/2}m_i^{-\alpha}$. This implies that IUP's average reward will be within $2Ld_i^{\alpha/2}M_i^{-\alpha}$ of the average reward of LO i.

Remark 2: Time order of the regret decreases as α increases (given that $T > d_i^{\alpha+d_i/2}$ holds. Otherwise, the bound given in Theorem 1 becomes trivial). This can be observed by investigating Assumption 1. Given two instances x and x' and a prediction rule f, as α increases, difference between the prediction accuracies of f for two instances x and x' that lie in the same set of the partition decreases. The constant that multiplies the time order of the regret increases as L increases. This holds because

[8] We decided to use AH as the ensemble learning algorithm due to its simplicity and regret guarantees. In practice, AH can be replaced with other ensemble learning algorithms. For instance, we also evaluate the performance when LLs use IUP and the EL uses Weighted Majority (WM) algorithm [10] in the numerical results section. Unlike AH, WM uses $q_i(t)$ as the weight of the prediction of LL i. It sets the weight of $y \in \mathcal{Y}$ to be $w_y(t) = \sum_{i:\hat{h}_i(t)=y} q_i(t)$ and predicts $\hat{y}(t) \in \arg\max_{y\in\mathcal{Y}} w_y(t)$.

as L increases, the difference between prediction accuracies of f for x and x' may become larger.

As a corollary of the above theorem, we have the following *confidence bound* on the accuracy of the predictions of LL i made by using IUP.

Corollary 1: Confidence bound for LL i. Assume that LL i uses IUP with the value of the partitioning parameter m_i given in Theorem 1. Let $\mathrm{ACC}_{i,\epsilon}(t)$ be the event that the prediction rule chosen by IUP for LL i at time t has accuracy greater than or equal to $\pi^*_i(x_i(t)) - \epsilon$. For any time t, we have $\Pr(\mathrm{ACC}_{i,\epsilon_t}(t)) \geq 1 - 1/T$, where $\epsilon_t = \sqrt{\frac{8}{N^i_{a_i(t),p_i(t)}(t)}(1 + 2\log(2|\mathcal{F}_i|m_i^{d_i}T^{\frac{3}{2}}))} + 2Ld_i^{\alpha/2}T^{\frac{-\alpha}{2\alpha+d_i}}$.

Proof: See Appendix C. ■

Corollary 1 gives a confidence bound on the predictions made by IUP for each LL. This guarantees that the prediction made by IUP is very close to the prediction of the best prediction rule that can be selected given the instance. For instance, in medical informatics, the result of this corollary can be used to calculate the patient sample size required to achieve a desired level of confidence in the predictions of the LLs. For instance, for every (ϵ, δ) pair, we can calculate the minimum number of patients N^* such that, for every new patient $n > N^*$, IUP will not choose any prediction rule that has suboptimality greater than $\epsilon > 0$ with probability at least $1 - \delta$, when it exploits (To achieve this we need to set the second term in (4) appropriately). Moreover, Corollary 1 can also be used to determine the number of patients that need to be enrolled in a clinical trial to achieve a desired level of confidence on the effectiveness of a drug.

The theorem below bounds the pseudo-regret of AH for any realization of LLs' rewards, hence almost surely.

Theorem 2: When AH is run with learning parameter $\eta(t) = \sqrt{\log M/t}$, for any reward sequence $\{\boldsymbol{v}_i^T\}_{i\in\mathcal{M}}$, the pseudo-regret of the EL with respect to the best LL is bounded by $\mathrm{Reg}_{\mathrm{EL}}(T) \leq 2\sqrt{T\log M}$. Hence, we have $\max_{i\in\mathcal{M}} \sum_{t=1}^{T} V_i(t) - \mathrm{E}\left[\sum_{t=1}^{T} R_{\mathrm{EL}}(t)\right] \leq 2\sqrt{T\log M}$ a.s., where the expectation is taken with respect to the randomization of the EL.

Proof: The proof is given in the online appendix [22]. ■

The next theorem shows that the expected regret of the EL given in (3) grows sublinearly over time and the term with the highest regret order scales with $F_{\max} = \max_{i\in\mathcal{M}} |\mathcal{F}_i|$ rather than $|\mathcal{F}|$, which is the sum of the number of prediction rules of all the LLs.

Theorem 3: Regret bound for the EL. When the EL runs AH with learning parameter $\eta(t) = \sqrt{\log M/t}$ and all LLs run IUP with the partitioning parameter given in Theorem 1, the expected regret of the EL with respect to the best LO i^* is bounded by

$$\mathrm{Reg}_{\mathrm{EL}}(T) \leq T^{\frac{\alpha+d_{i^*}}{2\alpha+d_{i^*}}} C_{i^*} + T^{\frac{d_{i^*}}{2\alpha+d_{i^*}}} 2^{d_{i^*}} |\mathcal{F}_{i^*}|$$
$$+ 2\sqrt{T\log M} + 1$$

where the definition of C_{i^*} is given in Theorem 1.

Proof: See Appendix D. ■

Theorem 3 proves that the highest time order of the regret does not depend on M, since C_{i^*} only depends on $|\mathcal{F}_{i^*}| \leq F_{\max}$ but not on $|\cup_{i\in\mathcal{M}} \mathcal{F}_i|$. This implies that the effect of the number of LLs to the learning rate is negligible. Since regret is measured with respect to the optimal data-dependent prediction strategy of the best LL (identical to the best LO), the benchmark will generally improve as LLs with higher performances are added to the system. Moreover, the learning loss with respect to the benchmark is only slightly affected by introducing new LLs to the system. Therefore, the performance of the EL will generally improve as LLs with higher performances are added to the system.

VII. Extensions

Active EL: Since IUP selects a prediction rule with high uncertainty when it explores, the prediction accuracy of an LL can be low when it explores. Since the EL combines the predictions of the LLs, taking into account the prediction of an LL which explores can reduce the prediction accuracy of the EL. In order to overcome this limitation, we propose the following modification: Let $\mathcal{A}(t) \subset \mathcal{M}$ be the set of LLs that exploit at time t. If $\mathcal{A}(t) = \emptyset$, the EL will randomly choose one of the LLs' prediction as its final prediction. Otherwise, the EL will apply an ensemble learning algorithm (such as AH or WM) using only the LLs in $\mathcal{A}(t)$. This means that only the predictions of the LLs in $\mathcal{A}(t)$ will be used by the EL and only the weights of the LLs in $\mathcal{A}(t)$ will be updated by the EL. Our numerical results illustrate that such a modification can indeed result in an accuracy that is much higher than the accuracy of the best LL.

Contextual EL (CEL): The predictive accuracy of the EL can be further improved if it can observe a set of contexts that yields additional information about the accuracies of LLs' prediction rules. For instance, these contexts can be a subset of the data instances that LLs observe, or some other side observation about the instance that the EL currently examines.

We assume that CEL can observe d_{EL}-dimensional context in addition to the predictions of the LLs. Let $x_{\mathrm{EL}}(t)$ be the context observed at time t by the EL, which is an element of $\mathcal{X}_{\mathrm{EL}} = [0,1]^{d_{\mathrm{EL}}}$. The learning algorithm we propose for CEL is called *Contextual Hedge* (CH). Similar to IUP, CH partitions the context space into equal sized, identically shaped, non-overlapping sets, and learns a different LL selection rule for each set in the partition. With this modification, the EL can learn the best LL for each set in the partition, which will yield a higher predictive accuracy than learning the best LL only based on the number of correct predictions.

The pseudocode of the CH is given in Fig. 5. CH runs a different instance of the AH in each set p of its context space partition $\mathcal{P}_{\mathrm{EL}}$. The cumulative loss vector it keeps for p at time t is denoted by $\boldsymbol{L}_p(t) = (L_{p,1}(t), \ldots, L_{p,M}(t))$, where $L_{p,i}(t)$ denotes the number of prediction errors made by LL i by the end of time step t for contexts that arrived to p. $N_{\mathrm{EL},p}(t)$ denotes the number of context arrivals to p by the end of time t. At the beginning of time step t, CH identifies the set in $\mathcal{P}$ that $x_{\mathrm{EL}}(t)$ belongs to, which is denoted by $p_{\mathrm{EL}}(t)$. After CH receives the set of predictions $\hat{\boldsymbol{h}}(t)$ of the LLs, it samples its final prediction from this set according to probability distribution

```
Contextual Hedge (CH)
  Input: A non-increasing sequence of positive real numbers
  $\{\eta(t)\}_{t\in\mathbb{N}+}$, $m_{\mathrm{EL}}$ and $d_{\mathrm{EL}}$
  Initialize sets: Create partition $\mathcal{P}_{\mathrm{EL}}$ of $[0,1]^{d_{\mathrm{EL}}}$ into $(m_{\mathrm{EL}})^{d_{\mathrm{EL}}}$
  identical hypercubes
  Initialize counters: $N_{\mathrm{EL},p} = 0, \forall p \in \mathcal{P}_{\mathrm{EL}}$, $t = 1$
  Initialize losses: $L_{p,i} = 0$, $\forall i \in \mathcal{M}$, $p \in \mathcal{P}_{\mathrm{EL}}$
  while $t \geq 1$ do
    Find the set in $\mathcal{P}_{\mathrm{EL}}$ that $x_{\mathrm{EL}}(t)$ belongs to, i.e., $p_{\mathrm{EL}}(t)$. Let
    $p = p_{\mathrm{EL}}(t)$
    Set $N_{\mathrm{EL},p} \leftarrow N_{\mathrm{EL},p} + 1$
    Receive predictions of LLs: $\hat{\boldsymbol{h}}(t)$
    Choose the LL $I(t)$ to follow according to the distribution
    $\boldsymbol{q}(t) := (q_1(t), \ldots, q_M(t))$ where

        $q_i(t) = \frac{\exp(-\eta(N_{\mathrm{EL},p})L_{p,i})}{\sum_{j=1}^{M} \exp(-\eta(N_{\mathrm{EL},p})L_{p,j})}$

    Predict $\hat{y}(t) = \hat{h}_{I(t)}(t)$
    Observe the true label $y(t)$
    Receive the reward $r_{\mathrm{EL}}(t) = \mathrm{I}(\hat{y}(t) = y(t))$ and observe
    losses of all LLs: $l_i(t) = \mathrm{I}(\hat{h}_i(t) \neq y(t))$ for $i \in \mathcal{M}$
    Set $L_{p,i} \leftarrow L_{p,i} + l_i(t)$ for $i \in \mathcal{M}$
    $t \leftarrow t + 1$
  end while
```

Fig. 5. Pseudocode of CH.

$\boldsymbol{q}(t) = (q_1(t), \ldots, q_M(t))$, where

$$\begin{aligned}\Pr(\hat{Y}(t) = \hat{h}_i(t)) &= q_i(t)\\ &= \frac{\exp(-\eta(N_{\mathrm{EL},p_{\mathrm{EL}}(t)}(t))L_{p_{\mathrm{EL}}(t),i}(t-1))}{\sum_{j=1}^{M} \exp(-\eta(N_{\mathrm{EL},p_{\mathrm{EL}}(t)}(t))L_{p_{\mathrm{EL}}(t),j}(t-1))}.\end{aligned}$$

Standard Hedge algorithm is not suitable in this setting because it requires the knowledge of $N_{\mathrm{EL},p}(T)$ beforehand for each $p \in \mathcal{P}_{\mathrm{EL}}$. However, AH works properly because it can update its learning parameter $\eta(\cdot)$ on-the-fly for each $p \in \mathcal{P}_{\mathrm{EL}}$, using the most recent value of $N_{\mathrm{EL},p}(t)$. Let $\mathcal{Z}_p(t) := \{l \leq t : x_{\mathrm{EL}}(l) \in p\}$ denote the set of times in which the context is in p by time t. For a given sequence of LL rewards $\{\boldsymbol{v}_i^T\}_{i\in\mathcal{M}}$ and context arrivals $\boldsymbol{x}_{\mathrm{EL}}^T$ we define the best LL for set $p \in \mathcal{P}_{\mathrm{EL}}$ of the EL as

$$i_p^* \in \arg\max_{i\in\mathcal{M}} \sum_{l\in\mathcal{Z}_p(T)} v_i(l).$$

The contextual pseudo-regret of CEL is defined as

$$\mathrm{Reg}_{\mathrm{CEL}}(T) := \sum_{t=1}^{T} v_{i^*_{p_{\mathrm{EL}}(t)}}(t) - \mathrm{E}\left[\sum_{t=1}^{T} R_{\mathrm{EL}}(t)\right] \qquad (7)$$

where the expectation is taken with respect to the randomization of CH. The following theorem bounds the regret of CH based on the granularity of the partition it creates.

Theorem 4: Regret bound for CH. When CEL runs CH with learning parameter $\eta(t) = \sqrt{\log M/t}$ and partitioning parameter m_{EL}, the contextual pseudo-regret of the CEL is bounded by

$$\mathrm{Reg}_{\mathrm{CEL}}(T) \leq 2\sqrt{T(m_{\mathrm{EL}})^{d_{\mathrm{EL}}} \log M}$$

for any $(\{\boldsymbol{v}_i^T\}_{i\in\mathcal{M}}, \boldsymbol{x}_{\mathrm{EL}}^T)$.

Proof: See Appendix E. ■

The regret bound given in Theorem 4 is obtained without making any distributional assumptions on data instance and context arrivals. Given a fixed time horizon T, this regret bound increases at rate $m_{\mathrm{EL}}^{d_{\mathrm{EL}}/2}$. Since the trivial regret bound $\mathrm{Reg}_{\mathrm{CEL}}(T) \leq T$ always holds, the bound in Theorem 4 guarantees that the regret is sublinear only if $m_{\mathrm{EL}} < (T/(4\log M))^{1/d_{\mathrm{EL}}}$. It might seem counter-intuitive that the regret is minimized when $m_{\mathrm{EL}} = 1$. The reason for this is that our benchmark $\sum_{t=1}^{T} v_{i^*_{p_{\mathrm{EL}}(t)}}(t)$ given in the left-hand side of (7) reduces to the benchmark $\max_{i\in\mathcal{M}} \sum_{t=1}^{T} v_i(t)$ given in (2) when $m_{\mathrm{EL}} = 1$. The next lemma shows that the reward of the benchmark in (7) is non-decreasing in m_{EL} when m_{EL} is chosen from $\{1, 2, 4, 8, \ldots\}$.

Lemma 1: Consider m' and m in $\{1, 2, 4, 8, \ldots\}$ such that $m' > m$. Let $\mathcal{P}'$ ($\mathcal{P}$) be the partition of $\mathcal{X}_{\mathrm{EL}}$ formed by m' (m). Let $p'(t)$ ($p(t)$) denote the set in $\mathcal{P}'$ ($\mathcal{P}$) that $x_{\mathrm{EL}}(t)$ belongs to. For any $(\{\boldsymbol{v}_i^T\}_{i\in\mathcal{M}}, \boldsymbol{x}_{\mathrm{EL}}^T)$, we have

$$\sum_{t=1}^{T} v_{i^*_{p'(t)}}(t) \geq \sum_{t=1}^{T} v_{i^*_{p(t)}}(t).$$

Proof: Due to the fact that m' and m are chosen from $\{1, 2, 4, 8, \ldots\}$, each $p' \in \mathcal{P}'$ is included in exactly one $p \in \mathcal{P}$[9] Moreover, each $p \in \mathcal{P}$ includes exactly $(m'/m)^{d_{\mathrm{EL}}}$ sets in $\mathcal{P}'$. Let $\mathcal{S}_p$ denote the set of $p' \in \mathcal{P}'$ such that $p' \subset p$. For any $p \in \mathcal{P}$ we have

$$\max_{i\in\mathcal{M}} \sum_{l\in\mathcal{Z}_p(T)} v_i(l) \leq \sum_{p'\in\mathcal{S}_p} \max_{i\in\mathcal{M}} \sum_{l\in\mathcal{Z}_{p'}(T)} v_i(l).$$

Hence,

$$\begin{aligned}\sum_{t=1}^{T} v_{i^*_{p(t)}}(t) &= \sum_{p\in\mathcal{P}} \max_{i\in\mathcal{M}} \sum_{l\in\mathcal{Z}_p(T)} v_i(l)\\ &\leq \sum_{p\in\mathcal{P}} \sum_{p'\in\mathcal{S}_p} \max_{i\in\mathcal{M}} \sum_{l\in\mathcal{Z}_{p'}(T)} v_i(l)\\ &= \sum_{p'\in\mathcal{P}'} \max_{i\in\mathcal{M}} \sum_{l\in\mathcal{Z}_{p'}(T)} v_i(l) = \sum_{t=1}^{T} v_{i^*_{p'(t)}}(t).\end{aligned}$$

■

Theorem 4 and Lemma 1 shows the tradeoff between approximation and estimation errors. The benchmark we compare CH against improves (never gets worse) as m_{EL} increases. Ideally, we would like CH to compete with $\sum_{t=1}^{T} v_{i^*_{\{x_{\mathrm{EL}}(t)\}}}(t)$, i.e., with respect to the best LL given context $x_{\mathrm{EL}}(t)$. For $x_{\mathrm{EL}}(t) \in p$, CH approximates $i^*_{\{x_{\mathrm{EL}}(t)\}}$ with i^*_p. Learning (estimating) $i^*_{\{x_{\mathrm{EL}}(t)\}}$ is harder than learning i^*_p because the past observations that CH can use to learn $i^*_{\{x_{\mathrm{EL}}(t)\}}$ is less than or equal to (usually less than) that it can use to learn i^*_p. This is the reason why the regret increases with m_{EL}. The optimal value for m_{EL} can be found by pre-training CH before its online deployment.

[9] Assignment of the contexts that lie on the boundary to one of the adjacent sets can be done in any predetermined way without affecting the result.

VIII. Illustrative Results

In this section, we evaluate the performance of several HB-based methods and compare them with numerous other state-of-the-art machine learning methods on a breast cancer diagnosis dataset from the UCI archive [23].

A. Simulation Setup

Description of the dataset: The original dataset contains 569 instances and 32 attributes, of which one attribute is the ID number of the patient and one attribute is the label. Each instance contains features extracted from the images of fine needle aspirate (FNA) of breast mass. There are 30 clinically relevant attributes. The diagnosis outcome (label) is whether the tumor of the patient is malignant or benign.

Benchmarks: We compare HB with several state-of-the-art centralized and decentralized benchmarks. A centralized benchmark is a machine learning algorithm that has access to all the features of an instance. A decentralized benchmark on the other hand, applies the same LL and EL structure as the HB. Hence, each LL has access to a subset of features. However, the algorithms used to train the LL and the EL are different from the HB.

In the first set of experiments, we compare the HB methods with centralized benchmarks such as Support Vector Machine (SVM) and Logistic Regression (LR). In the second set of experiments, we study the performance of various ensemble learning methods for the EL, by fixing the learning algorithm of the LLs as IUP. In the third set of experiments, we evaluate the impact of system variables such as the number of LLs and past history on the performance of the HB methods. In the fourth set of experiments, we consider the extensions described in Section VII.

The list of the algorithms used by the EL in this section is given below.

- Adaptive Boosting (AdaBoost) [11].
- Perceptron Weighted Majority (PWM) [6], [24].
- Blum's variant of Weighted Majority (Blum) [25].
- Herbster's variant of Weighted Majority (TrackExp) [26].

We also compare performance of the HB with standard benchmarks that are widely used in learning theory, which are listed below.

- Best LL: LL with the highest accuracy over the dataset.
- Worst LL: LL with the lowest accuracy over the dataset.
- Average LL: Accuracy averaged over all the LLs.

When IUP is used, we assume that each LL has two prediction rules: rule 1 always predicts malignant, and rule 2 always predicts benign. Hence, using IUP, each LL is learning the best prediction for each set in its feature space partition.

General setup: For all the simulations, each algorithm is run 50 times. The reported results correspond to the averages taken over these runs.

For the HB, we create 3 LLs, and randomly assign 10 attributes to each LL as its feature types for each run independently. The LLs do not have any common attributes. Hence, $d_i = 10$ for all $i \in \{1, 2, 3\}$. Each run of the HB is done over $T = 10000$ data instances that are drawn independently and uniformly at random from the 569 instances of the original dataset except Experiment 1 and 2, in which training and test samples are separated (for offline algorithms).

Performance metrics: We report three performance metrics for the above experiments: prediction error rate (PER), false positive rate (FPR) and false negative rate (FNR). PER is defined as the fraction of times the prediction is different from the true label. FPR and FNR are defined as the prediction error rate for benign cases and malignant cases, respectively. The main goal of diagnosis is to minimize the FPR, given a tolerable threshold for the FNR selected by the system user. In the simulations, the threshold for FNR is set to be 3%, which is considered to be a reasonable level in breast cancer diagnosis [27]. Using this threshold, we can re-characterize the performance metric as follows.

$$\text{minimize FPR subject to FNR} \leq 3\%.$$

FNR can be set below 3% by introducing a *hyper-parameter* which trade-offs FPR and FNR. The details are explained below.

For IUP, $\hat{\pi}^i_{1,p}$ ($\hat{\pi}^i_{2,p}$) denotes the estimated accuracy for malignant (benign) classifier for feature set p of LL i. Prediction is performed using the indices given in (4). LL i will predict malignant if $g^i_{1,p} \geq g^i_{2,p}$.[10] Otherwise, it will predict benign. Let h_{IUP} be the hyper-parameter for IUP. We can modify the prediction rule of IUP as follows: LL i predicts malignant if $h_{\text{IUP}} \times g^i_{1,p} \geq g^i_{2,p}$. Otherwise, it predicts benign. It is obvious that when $h_{\text{IUP}} > 1$, LL i classifies more cases as malignant, which yields a decrease FNR and an increase FPR.

For SVMs and logistic regression, the hyper-parameter is the decision boundary between the malignant and benign cases. Assume that we assign label 1 to the malignant case and 0 to the benign case. An unbiased decision boundary will classify every output that is greater than 0.5 as malignant and less than 0.5 as benign. If we perturb the decision boundary such that it lies below 0.5, then it is expected that SVM and LR classify more cases as malignant. This yields a decrease in FNR and an increase in FPR.

In order to set FNR just below 3%, we first randomly select a hyper-parameter value and run the corresponding algorithm 50 times, and then calculate FPR and FNR. After this step, the hyper-parameter is adjusted to minimize the distance between FNR and the threshold. The reported PER and FPR correspond to the ones that are obtained for the hyper-parameter value which makes FNR just below 3%.

To compare the performance of various algorithms, we introduce the concept of *improvement ratio* (IR). Let PM(A) denote the performance of algorithm A for metric PM. PM can be any loss metric such as PER, FPR, FNR. The IR of algorithm A with respect to algorithm B is defined as

$$(\text{PM(B)} - \text{PM(A)})/\text{PM(B)}.$$

B. Experiment 1 (Table I, Fig. 6)

This experiment compares HB against LR, SVM, AdaBoost (all trained offline); and Best LL, Average LL and Worst LL benchmarks. The training of the offline methods is performed

[10] Without loss of generality, we assume that the prediction of LL i is malignant when $g^i_{1,p} = g^i_{2,p}$.

TABLE I
COMPARISON OF HB WITH OFFLINE BENCHMARKS

Units(%)	Average			Standard Deviation		
Performance Metric	**PER**	**FPR**	**FNR**	**PER**	**FPR**	**FNR**
HB(IUP + WM)	2.96	2.61	2.99	0.73	1.26	0.88
HB(IUP + AH)	3.83	4.46	2.98	0.85	1.43	0.79
Logistic Regression	6.04	8.48	2.94	2.18	4.07	1.3
AdaBoost	6.91	9.55	2.99	2.58	4.82	1.83
SVMs	9.73	14.21	2.98	2.5	4.19	1.98
Best LL of IUP	3.39	3.57	2.99	0.85	1.43	0.79
Average LL of IUP	4.68	6.17	2.97	0.89	1.49	0.85
Worst LL of IUP	6.23	9.06	2.99	1.62	2.71	1.54

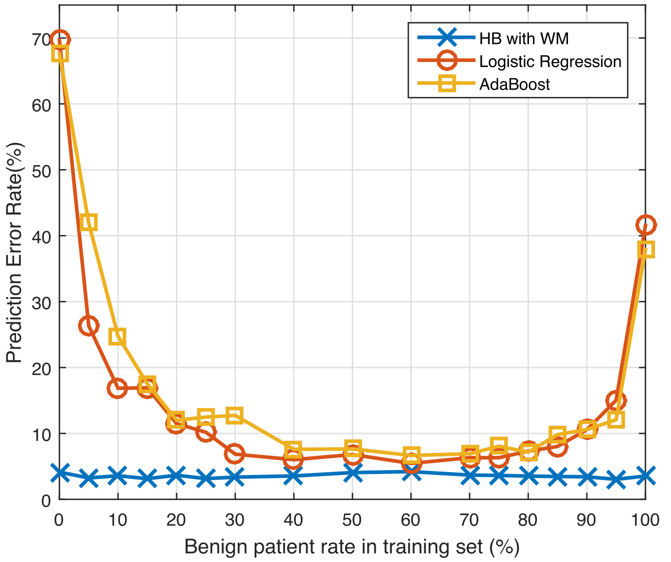

Fig. 6. PER of HB, LR and AdaBoost as a function of the composition of the training set.

as follows. LR and SVM are trained in a centralized way and have access to all 30 features. In the test phase, they observe all the 30 features of the new instance and make a prediction. AdaBoost is trained in a decentralized way. It has 3 weak learners (logistic regression with different parameters), which are randomly assigned to 10 of the 30 attributes. These weak learners do not share any common attributes.

For each run, offline methods are trained using different 285 (50$\%$) randomly drawn instances from the original 569 instances. Then, the performances of both the HB and benchmarks are evaluated on 10,000 instances drawn uniformly at random from the remaining 284 instances (excluding 285 training instances) for each run.

As Table I shows, HB (IUP + WM) has 2.96$\%$ PER and 2.61$\%$ FPR when the FNR is set to be just below 3$\%$. Hence, the PER IR of HB (IUP + WM) with respect to the best benchmark algorithm (LR) is 0.51. We also note that the PER IR of the best LL with respect to the second best algorithm is 0.44. This implies that the IUP used by the LLs yields high classification accuracy, because it is able to learn online the best prediction given the types of features seen by each LL.

HB with WM outperforms the best LL, because it takes a weighted majority of the predictions of LLs as its final prediction, rather than relying on the predictions of a single LL. As observed from Table I, all LLs have reasonably high accuracy, since PER of the worst LL is 6.23$\%$. In contrast to WM, AH puts a probability distribution over the LLs based on their weights, and follows the prediction of the chosen LL. With highly accurate LLs, the deterministic approach (WM) works better than the probabilistic approach (AH), because in almost all time steps, the majority of the LLs make correct predictions.

Another advantage of HB is that it has low standard deviation for PER, FPR and FNR, which is expected since IUP provides tight confidence bounds on the accuracy of the prediction rule chosen for any instance for which it exploits.

In Fig. 6, the performances of HB (with WM), LR and AdaBoost are compared as a function of the training set composition. Since both LLs and the EL learn online in HB, its performance does not depend on the training set composition. On the other hand, the performance of LR and AdaBoost highly depends on the composition of the training set. Although these benchmarks can be turned into an online algorithm by retraining them after every time step, the computational complexity of the online implementations for these algorithms will be high compared to that of the HB. Therefore, implementing the online versions of these benchmarks are not feasible when the dataset under consideration is large, and decisions have to be made on-the-fly.

C. Experiment 2 (Table II)

This experiment compares HB against four ensemble learning algorithms: AdaBoost, PWM, Blum and TrackExp. The goal of this experiment is to assess how the algorithm used by the EL impacts the performance. To isolate this effect, all the LLs use the same learning algorithm. The learning algorithms we use for the LLs are IUP (online), LR and SVM (offline). In this experiment, the performance metric is the accuracy for the 1001st patient. All of the other simulation details are exactly the same as in Experiment 1.

As seen in Table II, performance of the HB is better than the other ensemble learning methods when the FNR threshold is set to 3$\%$. More specifically, the performance improvement ratio of HB (with WM) in comparison with the second best algorithm (TrackExp) is 0.08 and 0.11 in terms of PER and FPR when IUP is used by the LLs.

D. Experiment 3 (Fig. 7)

This experiment analyzes the performance of the HB as a function of two system parameters: the number of LLs and the dataset size. Firstly, we analyze the performance using different numbers of LLs - from 2 to 30 -, over 10000 patients (as in Experiment 1). In this simulation, all the LLs have access to different types of attributes. Hence, as the number of LLs increase, the number of attributes per LL decreases. This can be viewed as increasing the amount of decentralization in the system. Secondly, we analyze the performance as a function of the total number of patients that have arrived so far. For this case, the number of LLs is fixed to 3.

TABLE II
COMPARISON OF HB WITH OTHER ENSEMBLE LEARNING METHODS

Unit(%)	**Local Learner Algorithm**								
	IUP			**Logistic Regression**			**SVMs**		
Ensemble Learning Algorithm	**PER**	**FPR**	**FNR**	**PER**	**FPR**	**FNR**	**PER**	**FPR**	**FNR**
HB(with WM)	**2.72**	**1.96**	2.94	**2.81**	**2.71**	2.97	**3.43**	**3.51**	2.97
HB(with AH)	4.35	5.05	**2.93**	3.61	3.81	2.95	4.63	5.11	2.97
AdaBoost	3.02	3.09	2.98	3.28	3.15	2.97	4.27	4.16	2.95
PWM	3.11	2.82	2.96	2.96	3.08	2.96	3.95	4.55	2.96
Blum	3.09	3.12	2.93	3.5	3.78	3.00	3.68	4.18	2.95
TrackExp	2.97	2.21	2.99	3.03	3.05	**2.93**	4.02	3.81	2.99
Best LL	3.96	4.69	2.96	3.22	3.3	2.99	3.96	4.26	2.96
Average LL	5.22	7.04	2.98	4.5	4.58	2.94	5.64	5.8	**2.92**
Worst LL	6.55	9.41	2.97	6.4	6.48	2.97	6.36	7.03	2.94

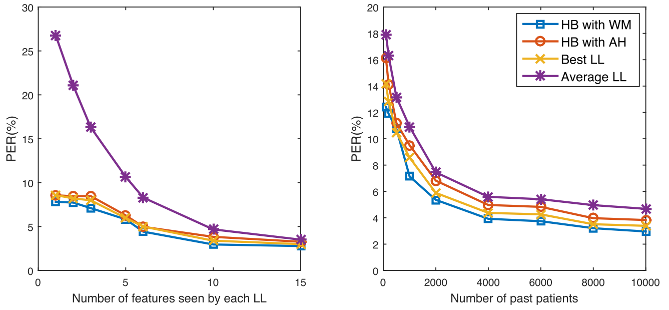

Fig. 7. Left: Number of features seen by each LL vs PER, Right: Number of past patients vs PER.

TABLE III
PERFORMANCE IN EXPLORATION AND EXPLOITATION STEPS

Units(%)	Exploration	Exploitation	Average
PER	50.34	3.79	4.51
FPR	42.88	2.51	2.94
FNR	55.88	5.93	7.11
Ratio	1.52	98.48	100.00

Effect of the number of LLs: The left Fig. 7 shows the performance of the HB with WM and AH as a function of the number of LLs. In this case, the number of features seen by each LL is roughly equal to $30/M$. As M increases both the performance of the LLs and the EL decreases. The decrease in the performance of the LLs is due to the fact that they see less features, and each LL has less information about the data. The decrease in the performance of the EL is due to the decrease in the performance of the LLs.

Effect of the number of previously diagnosed patients: The right Fig. 7 shows the performance of the HB as a function of the number past patients. As expected, the performance improves monotonically with the number of past patients, which is consistent with the regret results we have obtained.

E. Experiment 4

1) Extension 1: Active EL (Tables III and IV): Table III shows the percentage of times the LLs explore and exploit. The LLs are in exploration in 1.5% of the time steps, and the LLs' overall accuracy in these steps is around 50%.

TABLE IV
PERFORMANCE IMPROVEMENT WITH ACTIVE EL

Units(%)	**HB (with WM)**		
	Best LL	HB	HB with Active EL
PER	3.39	2.96	2.60
FPR	3.57	2.61	2.12
FNR	2.99	2.99	2.98
Units(%)	**HB (with AH)**		
	Best LL	HB	HB with Active EL
PER	3.39	3.83	3.30
FPR	3.57	4.46	3.46
FNR	2.99	2.98	2.98

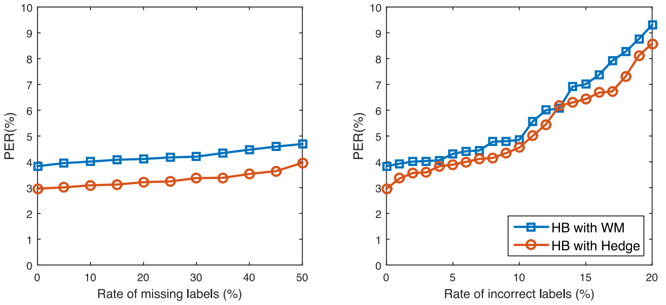

Fig. 8. Left: Performance degradation due to missing labels. Right: Performance degradation due to erroneous labels.

If the EL only considers the predictions of the LLs that exploit (Active EL), both HB with WM and AH have improved performance compared to the original HB, as shown in Table IV. Specifically, the PER IRs of Active EL (with WM or AH) with respect to the original HB (with WM or AH) are 0.12 and 0.14, respectively.

2) Extension 2: Missing and Erroneous Labels (Fig. 8): In this section, we illustrate the degradation in performance that results from randomly introducing missing or erroneous labels. When the label is missing, the LLs and the EL do not update their learning algorithms. Fig. 8 shows the affect of the missing label rate to the PER. It is observed that when 50% of the labels are missing, the PER degradation is only around 1% for both HB (with AH) and HB (with WM). This shows the robustness of HB to missing labels.

Next, we introduce erroneous labels (for binary labels, this correspond to flipped labels). Since the LLs and the EL update their learning algorithms when the label is incorrect, this results

TABLE V
COMPARISON OF CEL WITH ORIGINAL HB AND THE BEST LL IN TERMS OF PER, FPR AND FNR ($d_{EL} = 3$ FOR CEL(WM), $d_{EL} = 4$ FOR CEL(AH))

Units (%)	**PER**	**FPR**	**FNR**
CEL(WM)	2.31	1.48	2.98
CEL(AH)	3.49	4.01	2.95
HB(WM)	2.96	2.61	2.99
HB(AH)	3.83	4.46	2.98
Best LL of IUP	3.39	3.57	2.99

TABLE VI
PERFORMANCE OF CEL AS A FUNCTION OF THE NUMBER OF FEATURES THAT CEL CAN OBSERVE

	CEL +WM			**CEL +AH**		
d_{EL}	**PER**	**FPR**	**FNR**	**PER**	**FPR**	**FNR**
0 (Original HB)	2.96	2.61	2.99	3.83	4.46	2.98
1	2.71	2.13	2.99	4.16	5.07	2.97
2	2.33	1.47	3.00	3.74	4.31	2.98
3	**2.31**	**1.48**	**2.98**	3.64	4.16	2.96
4	2.42	1.61	2.92	**3.49**	**4.01**	**2.95**
5	2.46	1.54	2.97	4.03	4.95	2.98
6	2.58	1.71	2.93	4.35	5.34	2.99

in inaccurate accuracy estimates. Fig. 8 shows the affect of the erroneous label rate to the PER. For instance, when 10% of the labels are erroneous, the PER degradation is less than 2% for both HB (with AH) and HB (with WM).

3) Extension 3: Contextual EL (CEL): This experiment studies CEL introduced in Section VII. CEL is compared with the original HB and the best LL (each LL uses IUP). In addition to this, the predictive accuracy of the proposed method as a function of the number of features assigned to the CEL (d_{EL}) is computed. The simulation parameters are exactly the same as the parameters used in Experiment 1.

As Table V shows, CEL with WM has 2.31% PER, 1.48% FPR, and 2.98% FNR. Hence, the performance improvement ratios with respect to the original HB (IUP+WM) approach are 0.22 and 0.43 in terms of PER and FPR, respectively. In addition, the performance IRs with respect to the best LL are 0.32 and 0.59 in terms of PER and FPR, respectively. In other words, CEL with WM significantly outperforms the original HB (IUP+WM) and the best LL in terms of both PER and FPR. The reason for these improvements is that CEL learns the best LL for each feature set in its partition, rather than learning the best LL in overall.

Table VI shows the performance of CEL as a function of the number of features observed by the EL. When WM is used, the performance improves until $d_{\mathrm{EL}} = 3$, while when AH is used the performance improves until $d_{\mathrm{EL}} = 4$. The reason that the performance does not improve monotonically with d_{EL} is the tradeoff between estimation and approximation errors, which is described in detail in Section VII.

4) Effect of α on performance (Table VII): As α in Assumption 1 changes, optimized partitioning parameter $m_i = \lceil T^{1/(2\alpha + d_i)} \rceil$ changes. In illustrative results, we set $T = 10000$, $M = 3$ and $d_i = 10$ for all LLs. Thus, if $\alpha \geq 1.65$, $m_i = 2$. Otherwise, $m_i = 3$. Table VII shows that the optimal performance is achieved when $m_i = 2$.

TABLE VII
PERFORMANCE OF HB FOR DIFFERENT α VALUES

Units(%)		**HB (WM)**			**HB (AH)**		
α	m_i	PER	FPR	FNR	PER	FPR	FNR
≥ 1.65	2	2.96	2.61	2.99	3.83	4.46	2.98
< 1.65	3	5.56	6.81	2.98	6.46	9.02	2.97

IX. RELATED WORKS

In this section, we compare our proposed method with other online learning and ensemble learning methods in terms of the underlying assumptions and performance bounds.

Heterogeneous data observations: Most of the existing ensemble learning methods assume that the LLs make predictions by observing the same set of instances [10], [28]–[31]. Our methods allow the LLs to act based on heterogeneous data streams that are related to the same event. Moreover, we impose no statistical assumptions on the correlation between these data streams. This is achieved by isolating the decision making process of the EL from the data. Essentially, the EL acts solely based on the predictions it receives from the LLs.

Our proposed method can be viewed as attribute-distributed learning [9], [32]. In attribute-distributed learning, learners observe different features of the same instance and make local predictions. These local predictions are merged into a global prediction by a fusion center (EL). Numerous papers have considered the attribute-distributed learning model and proposed collaborative training algorithms to train the LLs [33], [34]. However, these algorithms require information exchange between the LLs. In contrast to these works, in our proposed work, information exchange is only possible between an LL and the EL. Hence concerns about data security and privacy are ruled out in our work.

There is a wide range of literature that develops distributed estimation techniques in which distributed LLs come up with consensus-based [35] or diffusion-based [36] parameter estimates by iteratively exchanging their local parameters computed based on the local observations. Unlike these works, in which the optimal parameter estimation problem is formulated as a distributed optimization problem, in our work the optimal prediction rule selection problem is formulated as a learning problem, and we explicitly focus on balancing the tradeoff between exploration and exploitation. Moreover, we do not make any restriction on the type of classifiers (prediction rules) used by LLs (except the similarity assumption), and do not require any message exchange between LLs.

Data-dependent oracle benchmark vs. empirical risk minimization: Our method can be viewed as online supervised learning with bandit feedback, because only the estimated accuracies of the prediction rules chosen by the LLs can be updated after the label is observed. Most of the prior works in this field use empirical risk minimization techniques [12], [13] to learn the optimal hypothesis. Let $H_i := \mathcal{X}_i \rightarrow \mathcal{F}_i$ denote a hypothesis for LL i, which is simply a mapping from the data instance that LL i observes to the set of prediction rules of LL i. Since the data instance space is taken as $[0, 1]^{d_i}$, there are infinitely many hypothesis. The optimal hypothesis for LL i is $H_i^*(x_i) = f_i^*(x_i)$.

As opposed to our work, ERM assumes access to N i.i.d. samples of the data instances, the label and the predictions (given as $\{(\boldsymbol{x}(t), y(t), \{\hat{y}_f(t)\}_{f\in\mathcal{F}})\}_{t=1}^{N}$) by which the loss of any hypothesis H_i can be evaluated. Using these i.i.d. samples, the empirical risk of H_i is calculated as $\overline{Risk}(H_i) = \frac{1}{N}\sum_{t=1}^{N}[R_{f_i^*(x_i(t))}(t) - R_{H_i(x_i(t))}(t)]$. For LL i, ERM seeks out to find a hypothesis $\hat{H}_i$ such that $\hat{H}_i \in \arg\min_{h\in\mathcal{H}_i} \overline{Risk}(h)$.

There are several important differences between ERM and our approach: In our approach the LLs and the EL update their hypothesis on-the-fly as more data and observations are gathered. IUP is an alternative to solving for the hypothesis that minimizes the empirical risk at each time step. Moreover, our algorithms are: (i) guaranteed to converge to the optimal hypothesis, and the convergence rate is explicitly characterized in terms of the regret bounds; (ii) work efficiently even when the hypothesis space $\mathcal{H}_i$ is infinite or very large by partitioning $\mathcal{X}_i$; (iii) work under partial feedback, i.e., only the prediction of the selected prediction rule is observed, hence the samples available at time step t are $(\boldsymbol{x}(t), y(t), \{\hat{y}_{a_i(t)}(t)\}_{i\in\mathcal{M}})$. Moreover, not all of these are observed by the same learner.

Reduced computational and memory complexity: Most ensemble learning methods require access to the entire dataset [11] or process the data in chunks [28]. For instance, [28] considers an online version of AdaBoost, in which weights are updated in a batch fashion after the processing of each data chunk is completed. Unlike this work, our method processes each instance only once upon its arrival, and do not need to store any past instances. Moreover, the LLs only learn from their own instances, and no data exchange between LLs are necessary. The above properties make our method efficient in terms of memory and computation, and suitable for distributed implementation.

Decentralized consensus optimization (DCO): The goal in DCO is to maximize a global objective function subject to numerous local constraints [37]–[40]. In this framework, distributed agents, which only have access to local information, exchange messages to cooperate with each other, in order to maximize the global payoff. The message exchange process continues until a predefined stopping criterion is satisfied. Unlike DCO, in our work, both local and global payoff functions are not known in advance. The LLs and the EL can only obtain noisy feedback about these payoffs, which is whether a prediction error happened or not. Moreover, the optimal actions (prediction rules) depend on the data instance (context), and hence are dynamically changing. In addition, the information only flows from the LLs to the EL, and there is only a single message exchange at each decision (time) step. Unlike maximizing the global objective function of a single-shot decision problem, our goal is to maximize the cumulative reward incurred over multiple decision steps.

X. Conclusion

In this paper we proposed a new online learning method that jointly considers the learning problem of the LLs and the EL. The proposed method comes with confidence and regret guarantees, which is very important in practice for many applications. Our theoretical results show that the time order of the regret for the EL is not affected by the number of LLs, which implies that the convergence speed of the EL to the optimal remains almost unchanged when the number of LLs in the system is increased. Our extensive numerical results show the superiority of the proposed approach in terms of its predictive accuracy. Specifically, Contextual EL performs significantly better than other ensemble learning methods, since it can utilize more information about the data features. We also proposed various other extensions to our proposed methods to deal with low confidence predictions during explorations and adaptation to missing labels.

Appendix A
Preliminaries for the Proof of Theorem 1

All the expressions used in the proofs below are related to LL i. To simplify the notation, we drop subscripts/superscripts related to LL i from the notation. For instance, we use $\hat{\pi}_{f,p}(t)$ instead of $\hat{\pi}^i_{f,p}(t)$, $N_{f,p}(t)$ instead of $N^i_{f,p}(t)$, $p(t)$ instead of $p_i(t)$ and f^* instead of $f_i^*(x)$ when the data instance we refer to is clear from the context.

The regret is computed by conditioning on $\boldsymbol{X}_i^T = \boldsymbol{x}_i^T$. Let $\tau_p^i(t)$ denote the time step in which the tth context arrives to $p \in \mathcal{P}_i$ of LL i. Let $\tilde{x}_p(t) = x_i(\tau_p^i(t))$, $\tilde{r}_{f,p}(t) = r_f(\tau_p^i(t))$, $\tilde{v}_p(t) = v_i(\tau_p^i(t))$, $\tilde{\pi}_{f,p}(t) = \hat{\pi}_{f,p}(\tau_p^i(t))$, $\tilde{N}_{f,p}(t) = N_{f,p}(\tau_p^i(t))$ and $\tilde{a}_p(t) = a_i(\tau_p^i(t))$. Let $N_p^i(T)$ (or simply $N_p(T)$) be the number of context arrivals to $p \in \mathcal{P}_i$ by the end of time T. Let

$$C_{f,p}(t) := \sqrt{\frac{2}{\tilde{N}_{f,p}(t)}(1 + 2\log(2|\mathcal{F}_i|(m_i)^{d_i}T^{\frac{3}{2}}))}.$$

For any $p \in \mathcal{P}_i$, $f \in \mathcal{F}_i$ and $t \in \{1, \ldots, N_p(T)\}$, we define the following lower confidence bound (LCB) and upper confidence bound (UCB):

$$L_{f,p}(t) := \max\{\tilde{\pi}_{f,p}(t) - C_{f,p}(t), 0\}$$
$$U_{f,p}(t) := \min\{\tilde{\pi}_{f,p}(t) + C_{f,p}(t), 1\}.$$

Let $\mathrm{UC}(f,p,v) := \cup_{t=1}^{N_p(T)}\{\pi_f(\tilde{x}_p(t)) \notin [L_{f,p}(t) - v, U_{f,p}(t) + v]\}$ denote the event that LL i is not confident about the accuracy of its prediction rule f at least once for instances in p by time T. Throughout our analysis we set $v = L(\sqrt{d_i}/m_i)^{\alpha}$. Let $\mathrm{UC}(p,v) := \bigcup_{f\in\mathcal{F}_i}\mathrm{UC}(f,p,v)$ and

$$\mathrm{UC}(v) := \cup_{p\in\mathcal{P}_i}\mathrm{UC}(p,v). \tag{8}$$

For each $p \in \mathcal{P}_i$ and $f \in \mathcal{F}_i$ let $\overline{\pi}_{f,p} := \sup_{x\in p}\pi_f(x)$ and $\underline{\pi}_{f,p} := \inf_{x\in p}\pi_f(x)$.

Appendix B
Proof of Theorem 1

We will bound the regret in each $p \in \mathcal{P}_i$ separately. Then, we will sum over all $p \in \mathcal{P}_i$ to bound the total regret. Preliminaries are given in Appendix A.

$$\begin{aligned}\mathrm{Reg}_i(T|\boldsymbol{X}_i^T) = {} & \sum_{t=1}^{T} \pi_{f_i^*(X_i(t))}(X_i(t)) \\ & - \mathrm{E}\left[\sum_{t=1}^{T} \pi_{A_i(t)}(X_i(t)) \middle| \boldsymbol{X}_i^T\right].\end{aligned} \tag{9}$$

The first term in (9) is obtained by observing that

$$\begin{aligned}& \mathrm{E}\left[\sum_{t=1}^{T} R_{f_i^*(X_i(t))}(t) \middle| \boldsymbol{X}_i^T\right] \\ & = \sum_{t=1}^{T} \sum_{f \in \mathcal{F}_i} \mathrm{E}\left[R_f(t) \mathrm{I}(f_i^*(X_i(t)) = f) | \boldsymbol{X}_i^T\right] \\ & = \sum_{t=1}^{T} \sum_{f \in \mathcal{F}_i} \mathrm{I}(f_i^*(X_i(t)) = f) \mathrm{E}\left[R_f(t) | \boldsymbol{X}_i^T\right] \\ & = \sum_{t=1}^{T} \pi_{f_i^*(X_i(t))}(X_i(t)).\end{aligned}$$

Let $\mathcal{F}_{t-1}$ be the sigma field generated by $\boldsymbol{X}_i^T$, $\boldsymbol{A}_i^{t-1}$, $\boldsymbol{Y}^{t-1}$. The second term in (9) is obtained by observing that

$$\mathrm{E}\left[\sum_{t=1}^{T} R_{A_i(t)}(t) \middle| \boldsymbol{X}_i^T\right]$$

$$= \sum_{t=1}^{T} \sum_{f \in \mathcal{F}_i} \mathrm{E}\left[\mathrm{E}\left[R_f(t) \mathrm{I}(A_i(t) = f) | \mathcal{F}_{t-1}\right] | \boldsymbol{X}_i^T\right] \tag{10}$$

$$= \sum_{t=1}^{T} \sum_{f \in \mathcal{F}_i} \mathrm{E}\left[\mathrm{I}(A_i(t) = f) \mathrm{E}\left[R_f(t) | \mathcal{F}_{t-1}\right] | \boldsymbol{X}_i^T\right] \tag{11}$$

$$= \sum_{t=1}^{T} \sum_{f \in \mathcal{F}_i} \mathrm{E}\left[\mathrm{I}(A_i(t) = f) \pi_f(X_i(t)) | \boldsymbol{X}_i^T\right] \tag{12}$$

$$= \mathrm{E}\left[\sum_{t=1}^{T} \pi_{A_i(t)}(X_i(t)) \middle| \boldsymbol{X}_i^T\right]$$

where (10) is by the law of iterated expectations, (11) is by the fact that $\mathrm{I}(A_i(t) = f)$ is $\mathcal{F}_{t-1}$ measurable, (12) is by definition of $\pi_f(\cdot)$ and the fact that $R_f(t)$ is independent of all random variables in $(\boldsymbol{X}_i^T, \boldsymbol{A}_i^{t-1}, \boldsymbol{Y}^{t-1})$ except $X_i(t)$.

For $p \in \mathcal{P}_i$, let

$$\begin{aligned}\mathrm{Reg}_{i,p}(T|\boldsymbol{X}_i^T = \boldsymbol{x}_i^T) := {} & \sum_{t=1}^{N_p(T)} \pi_{f^*(\tilde{x}_p(t))}(\tilde{x}_p(t)) \\ & - \mathrm{E}\left[\sum_{t=1}^{N_p(T)} \pi_{\tilde{A}_p(t)}(\tilde{x}_p(t)) \middle| \boldsymbol{X}_i^T = \boldsymbol{x}_i^T\right].\end{aligned} \tag{13}$$

Using (9) we obtain

$$\begin{aligned}\mathrm{Reg}_i(T|\boldsymbol{X}_i^T = \boldsymbol{x}_i^T) = {} & \sum_{p \in \mathcal{P}_i} \sum_{t=1}^{N_p(T)} \pi_{f^*(\tilde{x}_p(t))}(\tilde{x}_p(t)) \\ & - \mathrm{E}\left[\sum_{p \in \mathcal{P}_i} \sum_{t=1}^{N_p(T)} \pi_{\tilde{A}_p(t)}(\tilde{x}_p(t)) \middle| \boldsymbol{X}_i^T = \boldsymbol{x}_i^T\right] \\ = {} & \sum_{p \in \mathcal{P}_i} \mathrm{Reg}_{i,p}(T|\boldsymbol{X}_i^T = \boldsymbol{x}_i^T).\end{aligned} \tag{14}$$

The expectation in (13) is taken with respect to the randomness of $\tilde{A}_p(1), \ldots, \tilde{A}_p(N_p(T))$ given $\boldsymbol{X}_i^T = \boldsymbol{x}_i^T$. By the definition of IUP, conditioned on $\boldsymbol{X}_i^T = \boldsymbol{x}_i^T$, $\tilde{A}_p(t)$ only depends on random variables $\tilde{A}_p(1), \tilde{V}_p(1), \ldots, \tilde{A}_p(t-1), \tilde{V}_p(t-1)$. Since, $\tilde{V}_p(t) = \tilde{R}_{\tilde{A}_p(1),p}(t)$, we conclude that $\{\tilde{A}_p(t)\}_{t=1}^{N_p(T)}$ only depends on random variables $\boldsymbol{R}_p := \cup_{f \in \mathcal{F}_i} \{\tilde{R}_{f,p}(t)\}_{t=1}^{N_p(T)}$. Hence, the expectation in (13) is taken with respect to the conditional distribution of $\boldsymbol{R}_p$ given $\boldsymbol{x}_i^T$.

Since $\{(\boldsymbol{X}(t), Y(t), \{\hat{Y}_f(t)\}_{f \in \mathcal{F}})\}_{t=1}^T$ is an i.i.d. sequence, random variables $R_f(t)$, $t = 1, \ldots, T$ conditioned on $\boldsymbol{X}_i^T$ are independent. Since $R_f(t) \in \{0, 1\}$ and $\mathrm{E}\left[R_f(t) | \boldsymbol{X}_i^T = \boldsymbol{x}_i^T\right] = \pi_f(x_i(t))$, we can say that conditioned on $\boldsymbol{X}_i^T = \boldsymbol{x}_i^T$, $\{R_f(t)\}_{t=1}^T$ is a sequence of independent Bernoulli random variables with parameters $\{\pi_f(x_i(t))\}_{t=1}^T$ for $f \in \mathcal{F}_i$. With an abuse of notation, in the subsequent analysis in this section, $R_f(t)$ will denote the random reward of f conditioned on $X_i(t) = x_i(t)$, and all the expectations are taken with respect to the random variables defined above, unless otherwise stated. Hence, given $\boldsymbol{X}_i^T = \boldsymbol{x}_i^T$, we drop the conditioning on $\boldsymbol{X}_i^T$ from the notation and simply write

$$\mathrm{Reg}_{i,p}(T) = \sum_{t=1}^{N_p(T)} \pi_{f^*(\tilde{x}_p(t))}(\tilde{x}_p(t)) - \mathrm{E}_{\boldsymbol{R}_p}\left[\sum_{t=1}^{N_p(T)} \pi_{\tilde{A}_p(t)}(\tilde{x}_p(t))\right].$$

By the law of total expectation we have

$$\begin{aligned}& \mathrm{E}\left[\mathrm{Reg}_{i,p}(T)\right] = \mathrm{E}\left[\mathrm{Reg}_{i,p}(T) | \mathrm{UC}(p, v)\right] \Pr(\mathrm{UC}(p, v)) \\ & \quad + \mathrm{E}\left[\mathrm{Reg}_{i,p}(T) | \mathrm{UC}^C(p, v)\right] \Pr(\mathrm{UC}^C(p, v)) \\ & \quad \leq T \Pr(\mathrm{UC}(p, v)) + \mathrm{E}\left[\mathrm{Reg}_{i,p}(T) | \mathrm{UC}^C(p, v)\right].\end{aligned} \tag{15}$$

We will use the results of following lemmas to upper bound (15).

Lemma 2: (Bound on $\Pr(\mathit{UC}(f, p, v))$*):* $\Pr(\mathrm{UC}(f, p, v)) \leq 1/(|\mathcal{F}_i| m_i^{d_i} T)$.

Proof: Equivalently, we can define $\{\tilde{R}_{f,p}(t)\}_{t=1}^{N_p(T)}$ in the following way: Let $\{\eta_t\}_{t=1}^{N_p(T)}$ be a sequence of i.i.d. random variables uniformly distributed in $[0, 1]$. Then, $\tilde{R}_{f,p}(t) = \mathrm{I}(\eta_t \leq \pi_f(\tilde{x}_p(t)))$. We can express the sample mean reward (accuracy) of f as $\tilde{\pi}_{f,p}(t) = \sum_{l=1}^{t-1} \tilde{R}_{f,p}(l) \mathrm{I}(\tilde{a}_p(l) = f) / \tilde{N}_{f,p}(t)$. From the definitions of $L_{f,p}(t), U_{f,p}(t)$ and $\mathrm{UC}(f, p, v)$, it can be observed that the event $\mathrm{UC}(f, p, v)$ happens when $\tilde{\pi}_{f,p}(t)$ remains close to (or concentrates around) $\pi_f(\tilde{x}_p(t))$ for all $t \in \{1, \ldots, N_p(T)\}$.

This motivates us to use the concentration inequality given in Appendix F, which is derived in [41] from a similar concentration inequality in [42]. This inequality requires the expected reward from an action to be equal to the same constant at all time steps. This is clearly not the case for $\pi_f(\tilde{x}_p(t))$ since elements of $\{\tilde{x}_p(t)\}_{t=1}^{N_p(T)}$ are not identical which makes distributions of $\tilde{R}_{f,p}(t), t \in \{1, \ldots, N_p(T)\}$ different.

In order to overcome this issue, we propose a novel *sandwich technique*. Based on η_t, we define two new sequences of random variables, whose sample mean values will lower and upper bound $\tilde{\pi}_{\underline{f,p}}(t)$. The *best sequence* is defined as $\{\bar{R}_{f,p}(t)\}_{t=1}^{N_\rho(T)}$ where $\overline{R}_{f,p}(t) = \mathrm{I}(\eta_t \leq \overline{\pi}_{f,p})$, and the *worst sequence* is defined as $\{\underline{R}_{f,p}(t)\}_{t=1}^{N_\rho(T)}$ where $\underline{R}_{f,p}(t) = \mathrm{I}(\eta_t \leq \underline{\pi}_{f,p})$. Let $\overline{\pi}_{f,p}(t) := \sum_{l=1}^{t-1} \overline{R}_{f,p}(l)\mathrm{I}(\tilde{a}_p(l) = f)/\tilde{N}_{f,p}(t)$ and $\underline{\pi}_{f,p}(t) := \sum_{l=1}^{t-1} \underline{R}_{f,p}(l)\mathrm{I}(\tilde{a}_p(l) = f)/\tilde{N}_{f,p}(t)$. We have $\underline{\pi}_{f,p}(t) \leq \tilde{\pi}_{f,p}(t) \leq \overline{\pi}_{f,p}(t)$ $\forall t \in \{1, \ldots, N_p(T)\}$ almost surely. Let $\overline{L}_{f,p}(t) := \max\{\overline{\pi}_{f,p}(t) - C_{f,p}(t), 0\}$, $\overline{U}_{f,p}(t) := \min\{\overline{\pi}_{f,p}(t) + C_{f,p}(t), 1\}$, $\underline{L}_{f,p}(t) := \max\{\underline{\pi}_{f,p}(t) - C_{f,p}(t), 0\}$ and $\underline{U}_{f,p}(t) := \min\{\underline{\pi}_{f,p}(t) + C_{f,p}(t), 1\}$. It can be shown that

$$\begin{aligned}
&\{\pi_f(\tilde{x}_p(t)) \notin [L_{f,p}(t) - v, U_{f,p}(t) + v]\} \\
&\subset \{\pi_f(\tilde{x}_p(t)) \notin [\overline{L}_{f,p}(t) - v, \overline{U}_{f,p}(t) + v]\} \\
&\cup \{\pi_f(\tilde{x}_p(t)) \notin [\underline{L}_{f,p}(t) - v, \underline{U}_{f,p}(t) + v]\}.
\end{aligned}$$

The following inequalities can be obtained from Assumption 1.

$$\pi_f(\tilde{x}_p(t)) \leq \overline{\pi}_{f,p} \leq \pi_f(\tilde{x}_p(t)) + L\left(\frac{\sqrt{d_i}}{m_i}\right)^\alpha \tag{16}$$

$$\pi_f(\tilde{x}_p(t)) - L\left(\frac{\sqrt{d_i}}{m_i}\right)^\alpha \leq \underline{\pi}_{f,p} \leq \pi_f(\tilde{x}_p(t)). \tag{17}$$

Since $v = L\left(\sqrt{d_i}/m_i\right)^\alpha$, using (16) and (17) it can be shown that

$$\begin{aligned}
&\{\pi_f(\tilde{x}_p(t)) \notin [\overline{L}_{f,p}(t) - v, \overline{U}_{f,p}(t) + v]\} \\
&\subset \{\overline{\pi}_{f,p} \notin [\overline{L}_{f,p}(t), \overline{U}_{f,p}(t)]\}, \text{ and} \\
&\{\pi_f(\tilde{x}_p(t)) \notin [\underline{L}_{f,p}(t) - v, \underline{U}_{f,p}(t) + v]\} \\
&\subset \{\underline{\pi}_{f,p} \notin [\underline{L}_{f,p}(t), \underline{U}_{f,p}(t)]\}.
\end{aligned}$$

Using the equation above and the union bound we obtain

$$\begin{aligned}
\Pr(\mathrm{UC}(f,p,v)) \leq{}& \Pr\left(\cup_{t=1}^{N_p(T)}\{\overline{\pi}_{f,p} \notin [\overline{L}_{f,p}(t), \overline{U}_{f,p}(t)]\}\right) \\
&+ \Pr\left(\cup_{t=1}^{N_p(T)}\{\underline{\pi}_{f,p} \notin [\underline{L}_{f,p}(t), \underline{U}_{f,p}(t)]\}\right).
\end{aligned}$$

Both terms on the right-hand side of the inequality above can be bounded using the concentration inequality in Appendix F. Using $\delta = 1/(2|\mathcal{F}_i|(m_i)^{d_i}T)$ in Appendix F gives $\Pr(\mathrm{UC}(f,p,v)) \leq 1/(|\mathcal{F}_i|(m_i)^{d_i}T)$ since $1 + N_{f,p}(T) \leq T$. ■

Lemma 3: On event $\mathrm{UC}^C(p,v)$ we have $\pi_{f^*}(\tilde{x}_p(t)) - \pi_{\tilde{a}_p(t)}(\tilde{x}_p(t)) \leq U_{\tilde{a}_p(t),p}(t) - L_{\tilde{a}_p(t),p}(t) + 2v$ for all $t \in \{1, \ldots, N_p(T)\}$.

Proof: $U_{\tilde{a}_p(t),p}(t) + v \geq U_{f^*,p}(t) + v$ since IUP selects the decision rule with the highest index at each time step. On event $\mathrm{UC}^C(p,v)$ this implies $\pi_{f^*}(\tilde{x}_p(t)) \leq U_{\tilde{a}_p(t),p}(t) + v$. The proof concludes by observing that $\pi_{\tilde{a}_p(t)}(\tilde{x}_p(t)) \geq L_{\tilde{a}_p(t),p}(t) - v$ on event $\mathrm{UC}^C(p,v)$. ■

Lemma 4: (Bound on E $\left[Reg_{i,p}(T)|UC^C(p,v)\right]$):

$$\begin{aligned}
&\mathrm{E}\left[\mathrm{Reg}_{i,p}(T)|\mathrm{UC}^C(p,v)\right] \\
&\leq 2vN_p(T) + 2A_{m_i}\sqrt{|\mathcal{F}_i|N_p(T)} + |\mathcal{F}_i|
\end{aligned}$$

where $A_{m_i} := 2\sqrt{2(1 + 2\log(2|\mathcal{F}_i|(m_i)^{d_i}T^{\frac{3}{2}})}$.

Proof: Let $\mathcal{T}_{f,p} := \{t \leq N_p(T) : \tilde{a}_p(t) = f\}$. By Lemma 3,

$$\begin{aligned}
\mathrm{E}[\mathrm{Reg}_{i,p}(T)|\mathrm{UC}^C(p,v)] \leq{}& 2vN_p(T) \\
&+ \mathrm{E}\left[\sum_{f\in\mathcal{F}_i}\sum_{t\in\mathcal{T}_{f,p}}(U_{\tilde{A}_p(t),p}(t) - L_{\tilde{A}_p(t),p}(t))|\mathrm{UC}^C(p,v)\right].
\end{aligned} \tag{18}$$

Next, we show

$$\begin{aligned}
&\sum_{f\in\mathcal{F}_i}\sum_{t\in\mathcal{T}_{f,p}}(U_{\tilde{a}_p(t),p}(t) - L_{\tilde{a}_p(t),p}(t)) \\
&\leq \sum_{f\in\mathcal{F}_i}\left(1 + A_{m_i}\sum_{\{t\in\mathcal{T}_{f,p}:\tilde{N}_{f,p}(t)\geq 1\}}\sqrt{\frac{1}{\tilde{N}_{f,p}(t)}}\right) && (19) \\
&= |\mathcal{F}_i| + A_{m_i}\sum_{f\in\mathcal{F}_i}\sum_{l=0}^{N_{f,p}(T)-1}\sqrt{\frac{1}{1+l}} \\
&\leq |\mathcal{F}_i| + 2A_{m_i}\sum_{f\in\mathcal{F}_i}\sqrt{N_{f,p}(T)} && (20) \\
&\leq |\mathcal{F}_i| + 2A_{m_i}\sqrt{|\mathcal{F}_i|N_p(T)} && (21)
\end{aligned}$$

where (19) follows from the definition of $L_{f,p}(t)$ and $U_{f,p}(t)$, (20) follows from the fact that $\sum_{l=0}^{N_{f,p}(T)-1}\sqrt{\frac{1}{1+l}} \leq \int_{x=0}^{N_{f,p}(T)}(1/\sqrt{x})dx = 2\sqrt{N_{f,p}(T)}$ and (21) is obtained by applying the Cauchy-Schwarz inequality given in Appendix G and observing that $N_p(T) \geq \sum_{f\in\mathcal{F}_i} N_{f,p}(T)$. ■

Lemma 2 and the union bound yields

$$\Pr(\mathrm{UC}(p,v)) \leq 1/((m_i)^{d_i}T). \tag{22}$$

Upper bounding (15) by Lemma 4 and (22) gives

$$\begin{aligned}
\mathrm{E}[\mathrm{Reg}_{i,p}(T)] \leq{}& \frac{1}{(m_i)^{d_i}} + 2vN_p(T) + 2A_{m_i}\sqrt{|\mathcal{F}_i|N_p(T)} \\
&+ |\mathcal{F}_i|.
\end{aligned} \tag{23}$$

Using (14) together with (23) results in

$$\begin{aligned}
&\mathrm{Reg}_i(T|\boldsymbol{X}_i^T = \boldsymbol{x}_i^T) \\
&\leq \sum_{p\in\mathcal{P}_i}\left(\frac{1}{(m_i)^{d_i}} + 2vN_p(T) + |\mathcal{F}_i| + 2A_{m_i}\sqrt{|\mathcal{F}_i|N_p(T)}\right) \\
&\leq 1 + 2vT + |\mathcal{F}_i|(m_i)^{d_i} + 2A_{m_i}\sqrt{|\mathcal{F}_i|(m_i)^{d_i}T} && (24)
\end{aligned}$$

where the last inequality follows from the Cauchy-Schwarz inequality and $\sum_{p\in\mathcal{P}_i} N_p(T) = T$. The result of the theorem is obtained from (24) by setting $m_i = \lceil T^{1/(2\alpha+d_i)} \rceil$.

Appendix C
Proof of Corollary 1

Using (22), (8) and the union bound, we obtain (for any LL i) $\Pr\left(\mathrm{UC}^C\left(L(\frac{\sqrt{d_i}}{m_i})^\alpha\right)\right) \geq 1 - \frac{1}{T}$. Lemma 3 states that on event $\mathrm{UC}^C\left(L(\frac{\sqrt{d_i}}{m_i})^\alpha\right)$ we have

$$\begin{aligned}&\pi_{f_i^*(x_i(t))}(x_i(t)) - \pi_{a_i(t)}(x_i(t))\\ &\leq U_{a_i(t),p_i(t)}(N_{p_i(t)}(t)) - L_{a_i(t),p_i(t)}(N_{p_i(t)}(t))\\ &\quad + 2L\left(\frac{\sqrt{d_i}}{m_i}\right)^\alpha.\end{aligned}$$

The result follows from the definition of $U_{f,p}(t)$, $L_{f,p}(t)$ and the fact that $m_i = \lceil T^{1/(2\alpha+d_i)} \rceil$.

Appendix D
Proof of Theorem 3

Since the result of Theorem 2 holds for any realization $\{\boldsymbol{v}_i^T\}_{i\in\mathcal{M}}$ of the reward sequence, for any distribution over the reward sequence, we have

$$\mathrm{E}\left[\sum_{t=1}^T V_{i^*}(t)\right] - \mathrm{E}\left[\sum_{t=1}^T R_{\mathrm{EL}}(t)\right] \leq 2\sqrt{T\log M}. \quad (25)$$

The equation above holds since $\mathrm{E}[\max_{i\in\mathcal{M}} \sum_{t=1}^T V_i(t)] \geq \mathrm{E}[\sum_{t=1}^T V_{i^*}(t)]$ for any distribution over the reward sequence. $\overline{\mathrm{Reg}}_{\mathrm{EL}}(T)$ can be re-written as

$$\overline{\mathrm{Reg}}_{\mathrm{EL}}(T) = \mathrm{E}\left[\sum_{t=1}^T R_{f_{i^*}^*(X_{i^*}(t))}(t)\right] - \mathrm{E}\left[\sum_{t=1}^T R_{A_{i^*}(t)}(t)\right] \quad (26)$$

$$+\, \mathrm{E}\left[\sum_{t=1}^T V_{i^*}(t)\right] - \mathrm{E}\left[\sum_{t=1}^T R_{\mathrm{EL}}(t)\right] \quad (27)$$

since $\mathrm{E}[\sum_{t=1}^T R_{A_{i^*}(t)}(t)] = \mathrm{E}[\sum_{t=1}^T V_{i^*}(t)]$. The result follows from bounding (26) by using Theorem 1, and (27) by (25).

Appendix E
Proof of Theorem 4

Since CH keeps and updates a separate probability distribution over the LLs for each $p \in \mathcal{P}_{\mathrm{EL}}$, regret given in (7) can be re-written as

$$\mathrm{Reg}_{\mathrm{CEL}}(T) = \sum_{p\in\mathcal{P}_{\mathrm{EL}}}\left[\sum_{l\in\mathcal{Z}_p(T)} v_{i_p^*}(l) - \mathrm{E}\left[\sum_{l\in\mathcal{Z}_p(t)} R_{\mathrm{EL}}(l)\right]\right].$$

By Theorem 2 we obtain

$$\sum_{l\in\mathcal{Z}_p(T)} v_{i_p^*}(l) - \mathrm{E}\left[\sum_{l\in\mathcal{Z}_p(T)} R_{\mathrm{EL}}(l)\right] \leq 2\sqrt{N_{\mathrm{EL},p}(T)\log M}. \quad (28)$$

Using (28), the Cauchy-Schwarz inequality given in Appendix G and the fact that $\sum_{p\in\mathcal{P}_{\mathrm{EL}}} N_{\mathrm{EL},p}(T) = T$ we get

$$\begin{aligned}&\sum_{p\in\mathcal{P}_{\mathrm{EL}}}\left[\sum_{l\in\mathcal{Z}_p(T)} v_{i_p^*}(l) - \mathrm{E}\left[\sum_{l\in\mathcal{Z}_p(T)} R_{\mathrm{EL}}(l)\right]\right]\\ &\leq 2\sqrt{\log M}\sum_{p\in\mathcal{P}_{\mathrm{EL}}}\sqrt{N_{\mathrm{EL},p}(T)} \leq 2\sqrt{T(m_{\mathrm{EL}})^{d_{\mathrm{EL}}}\log M}.\end{aligned}$$

Appendix F
Concentration Inequality (Appendix A in [41])

Consider a prediction rule f of LL i for which the rewards are generated by an i.i.d. process $\{R(t)\}_{t=1}^T$ with $\pi_f = \mathrm{E}[R(t)]$, where the noise $R(t) - \pi_f$ is bounded in $[-1, 1]$. Let $N_f(T) \geq 1$ denote the number of times f is selected by LL i by the end of time T. Let $\hat{\pi}_f(T) = \sum_{t=1}^T \mathrm{I}(a_i(t) = f)R(t)/N_f(T)$. For any $\delta > 0$ with probability at least $1 - \delta$ we have

$$\begin{aligned}&|\hat{\pi}_f(T) - \pi_f|\\ &\leq \sqrt{\frac{2}{N_f(T)}\left(1 + 2\log\left(\frac{(1+N_f(T))^{1/2}}{\delta}\right)\right)} \quad \forall T \in \mathbb{N}.\end{aligned}$$

Appendix G
Cauchy-Schwarz Inequality

$| < \boldsymbol{x}, \boldsymbol{y} > | \leq \sqrt{< \boldsymbol{x}, \boldsymbol{x} >< \boldsymbol{y}, \boldsymbol{y} >}$, where $\boldsymbol{x}$ and $\boldsymbol{y}$ are D-dimensional real-valued vectors and $< \cdot, \cdot >$ denotes the standard inner product.

Appendix H
A bound on divergent series

For $\rho > 0$, $\rho \neq 1$, $\sum_{t=1}^T 1/(t^\rho) \leq 1 + (T^{1-\rho} - 1)/(1 - \rho)$.

Proof: See [43]. ■

References

[1] C. Tekin, J. Yoon, and M. van der Schaar, “Adaptive ensemble learning with confidence bounds for personalized diagnosis,” in *Proc. AAAI Workshop Expanding Boundaries Health Inf. Using AI*, 2016, pp. 462–468.

[2] D. Simons, “Consumer electronics opportunities in remote and home healthcare,” in *Proc. Int. Conf. Consum. Electron.*, 2008, pp. 1–2.

[3] Y. Cao and Y. Li, “An intelligent fuzzy-based recommendation system for consumer electronic products,” *Exp. Syst. Appl.*, vol. 33, no. 1, pp. 230–240, 2007.

[4] A. Beach *et al.*, “Whozthat? Evolving an ecosystem for context-aware mobile social networks,” *IEEE Netw.*, vol. 22, no. 4, pp. 50–55, Jul./Aug. 2008.

[5] S. R. Swix, J. R. Stefanik, and J. C. Batten, “Method and system for providing targeted advertisements,” U.S. Patent 6 718 551, Apr. 6, 2004.

[6] L. Canzian and M. van der Schaar, “Timely event detection by networked learners,” *IEEE Trans. Signal Process.*, vol. 63, no. 5, pp. 1282–1296, Mar. 2015.

[7] A. M. Eskicioglu and E. J. Delp, “An overview of multimedia content protection in consumer electronics devices,” *Signal Process.: Image Commun.*, vol. 16, no. 7, pp. 681–699, 2001.

[8] R. Arsanjani *et al.*, “Improved accuracy of myocardial perfusion spect for detection of coronary artery disease by machine learning in a large population,” *J. Nucl. Cardiol.*, vol. 20, no. 4, pp. 553–562, 2013.

[9] H. Zheng, S. R. Kulkarni, and H. Poor, “Attribute-distributed learning: models, limits, and algorithms,” *IEEE Trans. Signal Process.*, vol. 59, no. 1, pp. 386–398, Jan. 2011.

[10] N. Littlestone and M. K. Warmuth, “The weighted majority algorithm,” in *Proc. 30th Annu. Symp. Found. Comput. Sci.*, 1989, pp. 256–261.

[11] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of online learning and an application to boosting," in *Proc. Eur. Conf. Comput. Learn. Theory*, 1995, pp. 23–37.

[12] V. Vapnik, "Principles of risk minimization for learning theory," in *Proc. Advances Neural Inf. Process. Syst.*, 1992, pp. 831–838.

[13] 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.

[14] P. Auer, N. Cesa-Bianchi, and C. Gentile, "Adaptive and self-confident online learning algorithms," *J. Comput. Syst. Sci.*, vol. 64, no. 1, pp. 48–75, 2002.

[15] K. Chaudhuri, Y. Freund, and D. J. Hsu, "A parameter-free hedging algorithm," in *Proc. Adv.Neural Inf. Process. Syst.*, 2009, pp. 297–305.

[16] S. De Rooij, T. Van Erven, P. D. Grünwald, and W. M. Koolen, "Follow the leader if you can, hedge if you must," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1281–1316, 2014.

[17] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[18] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM J. Comput.*, vol. 32, pp. 48–77, 2002.

[19] N. Cesa-Bianchi, Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire, and M. K. Warmuth, "How to use expert advice," *J. ACM*, vol. 44, no. 3, pp. 427–485, 1997.

[20] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, pp. 235–256, 2002.

[21] T. Lu, D. Pál, and M. Pál, "Contextual multi-armed bandits," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2010, pp. 485–492.

[22] C. Tekin, J. Yoon, and M. van der Schaar, "Adaptive ensemble learning with confidence bounds," 2016. [Online]. Available: http://arxiv.org/abs/1512.07446.

[23] O. L. Mangasarian, W. N. Street, and W. H. Wolberg, "Breast cancer diagnosis and prognosis via linear programming," *Oper. Res.*, vol. 43, no. 4, pp. 570–577, 1995.

[24] L. Canzian, Y. Zhang, and M. van der Schaar, "Ensemble of distributed learners for online classification of dynamic data streams," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 1, no. 3, pp. 180–194, Sep. 2015.

[25] A. Blum, "Empirical support for winnow and weighted-majority algorithms: Results on a calendar scheduling domain," *Mach. Learn.*, vol. 26, no. 1, pp. 5–23, 1997.

[26] M. Herbster and M. K. Warmuth, "Tracking the best expert," *Mach. Learn.*, vol. 32, no. 2, pp. 151–178, 1998.

[27] L. Song, W. Hsu, J. Xu, and M. van der Schaar, "Using contextual learning to improve diagnostic accuracy: Application in breast cancer screening," *IEEE J. Biomed. Health Inf.*, vol. 20, no. 3, pp. 902–914, May 2016.

[28] W. Fan, S. J. Stolfo, and J. Zhang, "The application of AdaBoost for distributed, scalable and on-line learning," in *Proc. 5th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 1999, pp. 362–366.

[29] M. M. Masud, J. Gao, L. Khan, J. Han, and B. Thuraisingham, "Integrating novel class detection with classification for concept-drifting data streams," in *Machine Learning and Knowledge Discovery in Databases*. New York, NY, USA: Springer, 2009, pp. 79–94.

[30] W. N. Street and Y. Kim, "A streaming ensemble algorithm (SEA) for large-scale classification," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2001, pp. 377–382.

[31] L. L. Minku and X. Yao, "DDD: A new ensemble approach for dealing with concept drift," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 4, pp. 619–633, Apr. 2012.

[32] Y. Zhang, D. Sow, D. Turaga, and M. van der Schaar, "A fast online learning algorithm for distributed mining of bigdata," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 41, no. 4, pp. 90–93, 2014.

[33] D. Shutin, H. Zheng, B. H. Fleury, S. R. Kulkarni, and H. V. Poor, "Space-alternating attribute-distributed sparse learning," in *Proc. 2nd Int. Workshop Cognit. Inf. Process.*, 2010, pp. 209–214.

[34] D. E. Hershberger and H. Kargupta, "Distributed multivariate regression using wavelet-based collective data mining," *J. Parallel Distrib. Comput.*, vol. 61, no. 3, pp. 372–400, 2001.

[35] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5262–5276, Oct. 2010.

[36] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4289–4305, Aug. 2012.

[37] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[38] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Trans. Autom. Control*, vol. 55, no. 4, pp. 922–938, Apr. 2010.

[39] D. Yuan, S. Xu, and H. Zhao, "Distributed primal–dual subgradient method for multiagent optimization via consensus algorithms," *IEEE Trans. Syst. Man, Cybern. Part B (Cybern.)*, vol. 41, no. 6, pp. 1715–1724, Dec. 2011.

[40] T.-H. Chang, A. Nedić, and A. Scaglione, "Distributed constrained optimization by consensus-based primal-dual perturbation method," *IEEE Trans. Autom. Control*, vol. 59, no. 6, pp. 1524–1538, Jun. 2014.

[41] D. Russo and B. Van Roy, "Learning to optimize via posterior sampling," *Math. Oper. Res.*, vol. 39, no. 4, pp. 1221–1243, 2014.

[42] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári, "Improved algorithms for linear stochastic bandits," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 2312–2320.

[43] E. Chlebus, "An approximate formula for a partial sum of the divergent p-series," *Appl. Math. Lett.*, vol. 22, no. 5, pp. 732–737, 2009.

Cem Tekin (M'13) is an Assistant Professor in Electrical and Electronics Engineering Department at Bilkent University, Ankara, Turkey. He received the B.Sc. degree in electrical and electronics engineering from Middle East Technical University, Ankara, Turkey, in 2008, the M.S.E. degree in electrical engineering: systems, the M.S. degree in mathematics, and the Ph.D. degree in electrical engineering: systems from the University of Michigan, Ann Arbor, MI, USA, in 2010, 2011, and 2013, respectively. His research interests include machine learning, multi-armed bandit problems, data mining, and multi-agent systems. He received the University of Michigan Electrical Engineering Departmental Fellowship in 2008, and the Fred W. Ellersick award for the best paper in MILCOM 2009.

Jinsung Yoon received the B.S. degree in electrical engineering from Seoul National University, Seoul, Korea, in 2014, and the M.S. degree in electrical engineering from the University of California, Los Angeles, CA, USA, in 2016. He is currently working toward the Ph.D. degree in electrical engineering at University of California, Los Angeles. His research interests include machine learning algorithms, medical informatics, data mining, recommendation systems, and prediction systems. He received the Graduate Study Abroad Scholarship from the National Research Foundation of Korea in 2014.

Mihaela van der Schaar (F'10) is a Chancellor Professor of electrical engineering at the University of California, Los Angeles, CA, USA. Her research interests include machine learning, data science and decisions for medicine, education, and finance. She is an IEEE Fellow, a Distinguished Lecturer of the Communications Society for 2011–2012, the Editor in Chief of IEEE TRANSACTIONS ON MULTIMEDIA and a member of the Editorial Board of the IEEE JOURNAL ON SELECTED TOPICS IN SIGNAL PROCESSING. She received an NSF CAREER Award (2004), the Best Paper Award from IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (2005), the Okawa Foundation Award (2006), the IBM Faculty Award (2005, 2007, 2008), the Most Cited Paper Award from *EURASIP: Image Communications Journal* (2006), the Gamenets Conference Best Paper Award (2011), and the 2011 IEEE Circuits and Systems Society Darlington Award Best Paper Award. She received three ISO awards for her contributions to the MPEG video compression and streaming international standardization activities and holds 33 granted U.S. patents.