Active Learning in Context-Driven Stream Mining With an Application to Image Mining

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

***Abstract*— We propose an image stream mining method in which images arrive with contexts (metadata) and need to be processed in real time by the image mining system (IMS), which needs to make predictions and derive actionable intelligence from these streams. After extracting the features of the image by preprocessing, IMS determines online the classifier to use on the extracted features to make a prediction using the context of the image. A key challenge associated with stream mining is that the prediction accuracy of the classifiers is unknown, since the image source is unknown; thus, these accuracies need to be learned online. Another key challenge of stream mining is that learning can only be done by observing the true label, but this is costly to obtain. To address these challenges, we model the image stream mining problem as an active, online contextual experts problem, where the context of the image is used to guide the classifier selection decision. We develop an active learning algorithm and show that it achieves regret sublinear in the number of images that have been observed so far. To further illustrate and assess the performance of our proposed methods, we apply them to diagnose breast cancer from the images of cellular samples obtained from the fine needle aspirate of breast mass. Our findings show that very high diagnosis accuracy can be achieved by actively obtaining only a small fraction of true labels through surgical biopsies. Other applications include video surveillance and video traffic monitoring.**

***Index Terms*— Image stream mining, active learning, online classification, online learning, contextual experts, breast cancer diagnosis.**

I. Introduction

IMAGE stream mining aims to extract relevant knowledge from a diverse set of images generated by medical or surveillance systems, or personal cameras [1]. In this paper, we introduce a novel image stream mining method for classification of streams generated by heterogeneous and unknown image sources. The images sequentially arrive to the IMS which is equipped with a heterogeneous set of classifiers.

The images are first pre-processed using any of a plethora of existing image processing methods (targeted towards the specific application) to extract a set of features. In addition to the extracted features, each image comes together with a context that may give additional information (metadata) about the image. For example, for medical images, some dimensions of the context may include information from the health record of the patient, while some other dimensions may include a subset of the features extracted from the image. Fig. 1 depicts the envisioned system for a specific image stream mining application. Note, however, that our method is applicable to a wide range of other image stream mining applications such as surveillance, traffic monitoring etc. The task of the IMS is to mine the images as soon as they arrive and make predictions about the contents of the images. To accomplish this task, the IMS is endowed with a set of classifiers that make predictions using the features extracted from the images, which are trained a-priori based on images obtained from different sources, using a variety of training methods (logistic regression, naive Bayes, SVM, etc.). The goal of the IMS is to utilize the context information that comes along with the image to choose a classifier and follow its prediction. A key challenge is that the image characteristics of the acquired image streams are unknown, and thus, the accuracy of the various classifiers when applied to these images is unknown; the accuracy of a classifier for certain image streams, for specific contexts, can only be learned by observing how well such a classifier performed in the past when mining images with similar contexts. We call the module of the IMS that performs this learning as the *learning algorithm*. The performance of a classifier is measured against the true class (label) of the images. Nevertheless, observing the true label is costly and thus, labels should be judiciously acquired by assessing the benefits and costs of obtaining them. We call the task of the IMS, where image streams are acquired and need to be mined online, by selecting among a set of classifiers of unknown performance, and whose performance is learned online by proactively acquiring labels based on a benefit-cost analysis, *active image stream mining*. In this paper we propose methods for performing active image stream mining.

A key application of the envisioned active image stream mining is related to medical image diagnosis. One field which has received a lot of attention recently is radiology. The healthcare industry started taking steps to use data driven techniques to improve diagnosis accuracy from radiological images due to the existence of high error rates in radiological interpretations [2] and high variability of interpretations made by different radiologist on the same image [3]. Thus, it is important to design automated learning algorithms to help radiologists reduce error rates and interpretation variance. In the illustration provided in Fig. 1, breast cancer diagnosis is performed by analyzing images of cells obtained from FNA

Manuscript received May 8, 2014; revised March 3, 2015 and June 13, 2015; accepted June 14, 2015. Date of publication June 17, 2015; date of current version July 17, 2015. This work was supported in part by the Air Force Office of Scientific Research through the DDDAS Program and in part by the Division of Computer and Network Systems through the National Science Foundation under Grant 1016081. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Deniz Erdogmus.

The authors are with the Department of Electrical Engineering, University of California at San Diego, La Jolla, CA 92093 USA (e-mail: cmtkn@ucla.edu; 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/TIP.2015.2446936

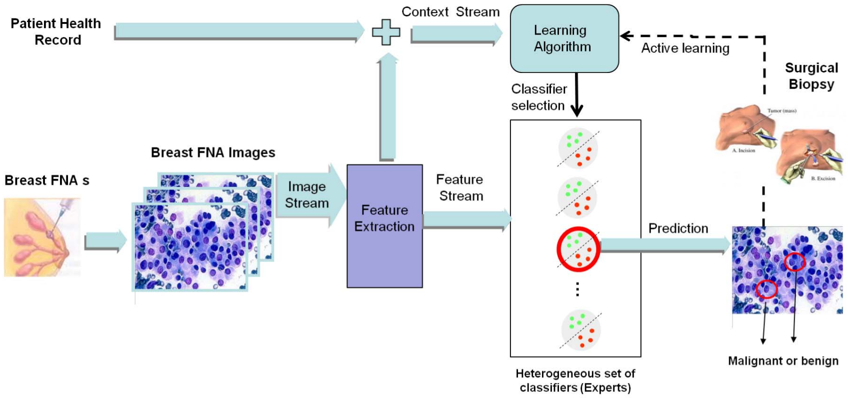

Fig. 1. Active image stream mining performed by the IMS by utilizing contextual information for classifier selection for breast cancer diagnosis.

of breast mass which is a minimally invasive procedure with a low cost [4]. For these images, features and contexts such as the number of cells, cell size, cell uniformity, etc. can be extracted by applying readily available feature extraction techniques [5], [6]. For example, [6] proposes a thresholding method to count the number of cells, while [5] uses a watershed based region growing approach to detect the cell nucleus and the features of nucleus (size shape, etc.) in a breast tissue image. However, such image analysis methods can only provide a prediction, and the true label (whether there is cancer or not), can only be obtained by a surgically invasive biopsy of the breast mass [7], which has a high cost. Then, a key task becomes learning when to ask for the true label such that the joint loss from misclassification and cost of asking for the true label is minimized. As we noted before, we call this task active image stream mining. It is different from most of the works in active learning [8] in the following sense. The focus of active image stream mining is to learn which classifier to choose among a set of pre-trained classifiers based on the context of the image, i.e., to learn the contextual specialization of classifiers, by inquiring minimum number of labels. In contrast, the focus of active learning is to selectively choose the training samples to design a classifier that works well on the remaining set of instances. Since we do not have control over the arriving images, most of the prior active learning methods [9]–[13] do not work in our problem.

According to our formulation, each classifier can be interpreted as an expert, that outputs a prediction about the image under consideration. Thus, in a more general instantiation of our proposed system, a classifier can be a software system or a radiologist. Since the learner follows the prediction of one of the experts based on the context of the image, we call this learning problem a *contextual experts problem*. As we mentioned, medical imaging is just one application of the proposed methodology.

As a performance measure for our active image stream mining method, we use the regret, which is defined as the difference between the expected total reward (number of correct predictions minus active learning costs) of the best mining scheme given complete knowledge about the accuracies of the available classifiers for all possible contexts and the expected total reward of the proposed algorithm. We then show that our proposed mining algorithms achieve regret sublinear in the number of images observed so far, which implies that the best classifier to choose for each possible context can be learned without any loss in terms of the average reward.

To summarize, the proposed active image stream mining methodology exhibits the following key features:

- Image streams are gathered and need to be mined continuously, online, rather then being stored in a database for offline processing.
- The IMS cannot control the sequence of arrivals.
- Our active stream mining algorithms are general and can be used in conjunction with any available set of classifiers.
- Classifier selection is based on the context of images, hence mining performance is maximized by learning contextual specialization of the classifiers.
- Learning speed is boosted by learning together for a group of similar contexts, instead of learning individually for each context.
- Our proposed algorithms achieve sublinear learning regret, which implies that the average loss due to learning and actively asking for the labels converges to zero.

Besides providing theoretical bounds for our proposed methods, we also illustrate the performance of our methods using a real world breast cancer diagnosis dataset obtained from UCI repository [14].

The remainder of the paper is organized as follows. In Section II, we describe the related work. In Section III, we formalize the active image stream mining problem, the benchmarks, and the regret. Then, in Sections IV and V, we propose active learning algorithms for the IMS, and prove sublinear regret bounds. Application of the proposed methods for breast cancer diagnosis is given in Section VI. Discussion and several extensions are proposed in Section VII. Concluding remarks are given in Section VIII.

II. Related Work

A. Related Work on Classifier Design

Previous works on image mining focus mainly on the design of classifiers [15]–[18] using supervised methods with training

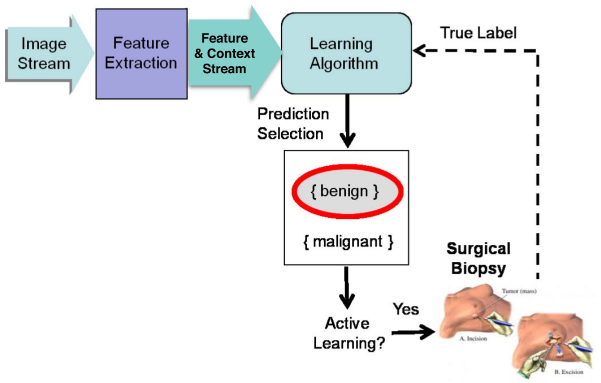

Fig. 2. Active image stream mining performed by the IMS by direct context based predictions.

images or unsupervised clustering methods [19], [20] by grouping images based on their features. Other works consider association rules [18], [21] or neural networks [22] to identify patterns and trends in large image sets.

For example, [19] considers an unsupervised learning problem in high dimensional data sets using generalized Dirichlet distribution to form clusters. The parameters of the distribution are estimated using a hybrid stochastic expectation maximization algorithm. This algorithm is offline and requires a batch of samples to form the clusters of data. In [15], an evolutionary artificial neural network is proposed to predict breast cancer, while in [23], a selective Bayesian classifier that chooses which features to use in trainings is designed. In [24], an artificial neural network is proposed to classify the fundus of the eye of a patient for detection of diabetic retinopathy.

All the abovementioned literature either requires a certain number of training images, or works in a completely unsupervised way [20], without requiring any labels.

Our active image stream mining system operates at a different level: it builds on the existing work by assuming that the system has access to many classifiers (with unknown accuracies), and learns online which classifier's prediction to follow based on the contexts of the images. The main motivation of this paper is to use the diversity of the base classifiers along with learning their contextual specializations to boost the prediction accuracy. Hence, all the proposed methods above can be used to design the base classifiers that are used by our learning algorithms. In addition to employing pretrained base classifiers that make predictions about the image using the features extracted from the image, our proposed mining methods can also directly use the extracted features as context information to make context based predictions about the image. This system is illustrated for the breast cancer diagnosis application in Fig. 2.

B. Related Work on Active Learning

Since the past performance can only be assessed through the labels, and since obtaining the labels is costly, actively learning when to ask for the label becomes an important challenge. The literature on active learning can be divided into three categories. In stream-based active learning [25]–[28], the learner is provided with a stream of unlabeled instances. When an instances arrives, the learner decides to obtain the label or not. In pool-based active learning [9]–[12], there is a pool of unlabeled instances that the learner can choose from. At each time slot the learner can pick an instance from the pool and obtain its label. In active learning with membership queries [13], the learner has access to every possible instance, and at each time slot chooses an instance and obtains its label. In all of these active learning problems, the goal is to obtain only the labels of the instances that have the highest label uncertainty based on the labels obtained so far.

Unlike pool-based and membership queries methods, in this paper the IMS does not have control over image arrival process, and we do not need to store the images in a database. Hence, the most closely related active learning category to our work is stream-based active learning.

C. Related Work on Ensemble Learning

In our model, the learner has access to many classifiers and follows the prediction of a single classifier based on the context. Our method can be seen as a deterministic ensemble learning method where the IMS learns to follow the best (*expert*) classifier for a given context. There are other types of ensemble learning methods which combine the predictions of classifiers (e.g., by weights), and produce a final prediction based on the predictions of the classifiers in the ensemble. For example, [29]–[35] use techniques such as bagging, boosting, stacked generalization and cascading. However, most of them provides algorithms which are asymptotically converging to an optimal or locally-optimal solution without providing any rates of convergence. On the contrary, we do not only prove convergence results, but we are also able to explicitly characterize the performance loss incurred at each time step with respect to the complete knowledge benchmark which knows the accuracies of all classifiers.

Some other ensemble learning methods use the weights assigned to the classifiers to build a randomized algorithm that chooses a prediction [36]–[40]. These weights can be updated online [38] based on the past performance of the classifiers. These papers also prove strong theoretical guarantees on the performance of the ensemble. Our difference is that, we focus on how contextual specializations of classifiers can be discovered over time to create a strong (high overall prediction accuracy) predictor from many weak (low overall prediction accuracy) classifiers.

D. Related Work on Experts and Contextual Bandits

The most closely related theoretical approaches to ours are the ones on prediction with expert advice [26]–[28], [41] and contextual bandits [42]–[47].

In the experts problem [41], the learner observes predictions of N experts at each time slot, and comes up with a final prediction using this information. The goal is to design algorithms that perform as good as the best expert for a sequence of labels generated by an adversary. To do this, the authors propose a randomized learning algorithm with exponential weights. [26] proposes the active learning version of the experts problem called *label efficient learning*.

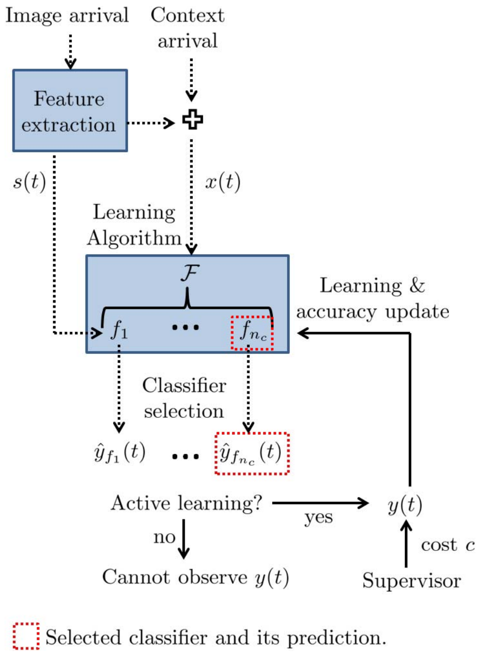

Fig. 3. Operation of the IMS during a time slot.

They derive conditions on the number of required label queries such that the regret of the learning algorithm is sublinear with respect to the best classifier in a given set of classifiers. The variation in [28] considers costs associated with obtaining the features as well as the label, while [27] studies a slightly different problem, where labels are generated by a set of *teachers* according to some unknown noisy linear function of the features. Instead of actively learning the ground truth, the learner learns actively from the labels generated by different teachers depending on their expertise. In contrast, in our paper labels are generated according to an arbitrary joint distribution over features, contexts and labels, and active stream mining reveals the ground truth. In all of the work described above, the benchmark of regret is the best fixed classifier in a given set of classifiers as opposed to our benchmark which is the best context-dependent classifier, which can be significantly better in terms of accuracy. Another difference is that we propose a deterministic learning approach as opposed to the randomized learning approach proposed in above works.

In the contextual bandit framework [42], [43], [47], the learner can only observe the reward of the selected action (classifier), but observes it every time that action is selected. This results in an exploration-exploitation tradeoff which needs to be carefully balanced to achieve good performance. In contrast, in this paper, reward observation is not always possible.

III. Problem Formulation

In this section we present the system model, define the data and context arrival process, classifier accuracies and the regret. Frequently used notations are given in Appendix A.

A. System Model

The system model is shown in Fig. 3. The IMS is equipped with n_c classifiers indexed by the set $\mathcal{F} := \{1, 2, \ldots, n_c\}$. The system operates in a discrete time setting $t = 1, 2, \ldots, T$, where the following events happen sequentially, in each time slot t: (i) An image arrives to the IMS and its features $s(t)$ are extracted by some preprocessing method. As we discussed in the Introduction Section, this extraction can be performed by applying readily available feature extraction techniques [5], [6]. The context $x(t)$ of the image is either given externally together with the image or includes some of the extracted features or both. (ii) Each classifier $f \in \mathcal{F}$ produces a prediction $\hat{y}_f(t)$ based on $s(t)$. (iii) The IMS follows the prediction of one of its classifiers, which is denoted by $\hat{y}(t)$. (iv) The true label is revealed only when it is asked for, by a supervisor such as a human operator, and there is a constant cost $c > 0$, i.e., *active learning cost*, associated with asking for the true label. (v) If the true label $y(t)$ is obtained, the IMS updates the estimated accuracy of its classifiers.

Let $a_{\text{pr}}(t) \in \mathcal{F}$ be the *prediction action* of the IMS at time t. It is the classifier whose prediction is followed by the IMS at time t. We also call $\mathcal{F}$ as the set of arms of the IMS. Hence we use the term classifier and arm interchangeably. Let $a_{\text{lr}}(t)$ be the *learning action* of the IMS at time t. For $a_{\text{lr}}(t) = 0$, the IMS does not ask for the label, while for $a_{\text{lr}}(t) = 1$, it asks for the label and pays cost c. Clearly, $\hat{y}(t) = \hat{y}_{a_{\text{pr}}(t)}(t)$.

B. Context, Label and Classifiers

Let $\mathcal{X} = [0, 1]^D$ be the set of contexts,[1] where D is the dimension of the context space, $\mathcal{S}$ be the set of images and $\mathcal{Y} = \{0, 1\}$ be the set of labels.[2] Each classifier f is endowed with a *prediction rule* $Q_f : \mathcal{S} \to \Delta(\mathcal{Y})$, where $\Delta(\mathcal{Y})$ denotes the set of probability distributions over $\mathcal{Y}$. Let $\hat{Y}_f(s)$ be the random variable which denotes the label produced by classifier f when input image is s.

At each time slot t, $s(t)$, $x(t)$ and $y(t)$ are drawn from an unknown but fixed joint distribution J over $\mathcal{S} \times \mathcal{X} \times \mathcal{Y}$. We call J the *image distribution*. Let J_x denote the conditional distribution of image and label given context x. Then, classifier f's accuracy for context x is given by $\pi_f(x) := \mathrm{E}_{J_x, Q_f}[\mathrm{I}(\hat{Y}_f(S) = Y)]$, where S and Y are the random variables corresponding to image and label whose conditional distribution is given by J_x. We assume that each classifier has similar accuracies for similar contexts; we formalize this in terms of a Hölder condition.

Assumption 1: There exists $L > 0$, $\alpha > 0$ such that for all $x, x' \in \mathcal{X}$ and $f \in \mathcal{F}$, we have $|\pi_f(x) - \pi_f(x')| \leq L||x - x'||^\alpha$.

Assumption 1 indicates that the accuracy of a classifier for similar contexts is similar to each other. We assume that the IMS knows α, while L is not required to be known. An unknown α can be estimated online using the sample mean estimates of accuracies for similar contexts, and our proposed algorithms can be modified to include the estimation of α.

The image input $s(t)$ is high dimensional and its dimension is greater than D (in most of the cases its much larger than D). For example, in the breast cancer dataset 10 features

[1] In general, our results will hold for any bounded subspace of $\mathbb{R}^D$.

[2] Considering only binary labels/classifiers is not restrictive since in general, ensembles of binary classifiers can be used to accomplish more complex classification tasks [48], [49].

are extracted from the image by preprocessing. However, in one of our simulations we only use one of the features as contexts. In such a setting, exploiting the low dimensional context information may significantly improve the learning speed.

The goal of the IMS is to minimize the number of incorrect predictions and costs of asking for the label. Hence, it should learn well the accuracies of the classifiers while minimizing the number of times it actively asks for the label. We model the IMS's problem as a contextual experts problem, where the accuracies translate into rewards.

C. The Complete Knowledge Benchmark

Our benchmark when evaluating the performance of the learning algorithms is the optimal solution in which the IMS follows the prediction of the best classifier in $\mathcal{F}$, i.e., the classifier with the highest accuracy for context $x(t)$, at time t. Given context x, the classifier followed by the complete knowledge benchmark is

$$f^*(x) := \arg\max_{f \in \mathcal{F}} \pi_f(x). \tag{1}$$

D. The Regret of Learning

Simply, the regret is the loss incurred due to the unknown system dynamics. The regret of the IMS by time T is

$$R(T) := \sum_{t=1}^{T} \pi_{f^*(x(t))}(x(t)) - \mathrm{E}\left[\sum_{t=1}^{T} (\mathrm{I}(\hat{y}(t) = y(t)) - c a_{\mathrm{lr}}(t))\right]$$

where the expectation is taken with respect to the randomness of the prediction, label and the actions. Regret gives the convergence rate of the total expected reward of the learning algorithm to the value of the optimal solution given in (1). Any algorithm whose regret is sublinear, i.e., $R(T) = O(T^\gamma)$ such that $\gamma < 1$, will converge to the optimal solution in terms of the average reward.

IV. Uniformly Partitioned Contextual Experts

In this section we propose a learning algorithm for the IMS which achieves sublinear regret for the active image stream mining problem that creates a uniform partition of the context space and learns the best classifier (expert) for each set in the partition. The algorithm is called *Uniformly Partitioned Contextual Experts* (UPCE) and its pseudocode is given in Fig. 4.

We would like to note that our contextual experts algorithm is significantly different from prior works [42]–[47], which design index-based learning algorithms for contextual bandits. The main difference is that UPCE must actively control when to ask for the true label, and hence, when to update the accuracy of the classifiers, while in prior work in contextual bandits the reward is always observed *after* an action is taken. However, in contextual bandits only the reward of the selected action is observed, while in contextual experts, reward of all the actions are observed when the label is obtained. At each time slot UPCE follows the prediction of the expert with the highest estimated accuracy, while in contextual bandits, exploration of suboptimal classifiers are needed occasionally. This difference between contextual experts and bandits is very important from the application point of view, since in many applications including the medical applications, explorations are not desirable to promote fairness and equally treat all patients.

```
Uniformly Partitioned Contextual Experts (UPCE):
Input: $D(t)$, $T$, $m_T$
Initialize: Form $\mathcal{P}_T$, the partition of $[0,1]^D$ into $(m_T)^D$ hypercubes. For each $p \in \mathcal{P}_T$ set $N_p = 0$, $\hat{\pi}_{f,p} = 0$, $\forall f \in \mathcal{F}$
while $t \geq 1$ do
    Find the set in $\mathcal{P}_T$ that $x(t)$ belongs to, i.e., $p(t)$
    Set $p = p(t)$
    Set $a_{\mathrm{pr}}(t) = \arg\max_{f \in \mathcal{F}} \hat{\pi}_{f,p}$
    Set $\hat{y}(t)$ be the prediction of classifier $a_{\mathrm{pr}}(t)$
    if $N_p \leq D(t)$ then
        Obtain the true label $y(t)$ by paying cost $c$, i.e., $a_{\mathrm{lr}}(t) = 1$
        for $f \in \mathcal{F}$ do
            $\hat{\pi}_{f,p} = \frac{\hat{\pi}_{f,p} N_p + \mathrm{I}(\hat{y}_f(t) = y(t))}{N_p + 1}$
            $N_p + +$
        end for
    else
        Do not ask for the true label, i.e., $a_{\mathrm{lr}}(t) = 0$
    end if
    $t = t + 1$
end while
```

Fig. 4. Pseudocode for UPCE.

Basically, UPCE forms a uniform partition $\mathcal{P}_T$ of the context space consisting of $(m_T)^D$, D dimensional hypercubes, and estimates the accuracy of each classifier for each hypercube based only on the history of observations in that hypercube. The essence behind UPCE is that if a set $p \in \mathcal{P}_T$ is small enough, then the variation of the classifier accuracies in this set is small due to Assumption 1, hence the average of the rewards observed in p at times when classifier f is selected approximates well the accuracy of classifier f. Thus, there is a tradeoff between the number of hypercubes and the approximation mentioned above, which needs to be carefully balanced. Moreover, since asking for the true label is costly, UPCE should also balance the tradeoff between the cost incurred due to active learning and reward loss due to inaccurate classifier accuracy estimates.

In order to balance this tradeoff, UPCE keeps a deterministic *control function* $D(t)$ that is a non-decreasing function of t. For each $p \in \mathcal{P}_T$ UPCE keeps a counter $N_p(t)$ which counts the number of times a context in set p arrived to the IMS by time t and the IMS obtained its true label. Also for each classifier f in this set, it keeps the estimated accuracy $\hat{\pi}_{f,p}(t)$. $\hat{\pi}_{f,p}(t)$ is the sample mean of the rewards (correct predictions) obtained from classifier f for contexts in set p at time slots for which the true label is obtained by time t.

The IMS does the following at time t. It first finds to which set in $\mathcal{P}_T$ the context $x(t)$ belongs to. Denote this set by $p(t)$. Then, it observes the predictions of the classifiers for $s(t)$, i.e., $\hat{y}_f(t)$, $f \in \mathcal{F}$. It follows the prediction of the (estimated) best classifier, i.e., $a_{\mathrm{pr}}(t) = \arg\max_{f \in \mathcal{F}} \hat{\pi}_{f,p}(t)$. If the classifier accuracies for set p are under-explored, i.e., if $N_p(t) \leq D(t)$, the IMS asks for the true label $y(t)$ and pays cost c. Otherwise it does not ask for $y(t)$. If $y(t)$ is obtained, the IMS updates the estimated accuracy of classifier $f \in \mathcal{F}$ as follows:

$$\hat{\pi}_{f,p}(t+1) = (\hat{\pi}_{f,p}(t) N_p(t) + \mathrm{I}(\hat{y}_f(t) = y(t)))/(N_p(t)+1).$$

In the following subsection we will derive the values of m_T and $D(t)$ that will lead to optimal tradeoff between active learning cost and prediction accuracy.

A. Regret Bound for UPCE

Let $\beta_a := \sum_{t=1}^{\infty} 1/t^a$, and let log(.) denote logarithm in base e. For each set (hypercube) $p \in \mathcal{P}_T$ let $\overline{\pi}_{f,p} := \sup_{x \in p} \pi_f(x)$ and $\underline{\pi}_{f,p} := \inf_{x \in p} \pi_f(x)$, for $f \in \mathcal{F}$. Let x^*_p be the context at the center (center of symmetry) of the hypercube p. We define the optimal classifier for set p as

$$f^*(p) := \arg\max_{f \in \mathcal{F}} \pi_f(x^*_p).$$

When the set p is clear from the context, we will simply denote the optimal classifier for set p with f^*. Let

$$\mathcal{L}_p(t) := \left\{ f \in \mathcal{F} \text{ such that } \underline{\pi}_{f^*(p),p} - \overline{\pi}_{f,p} > At^{\theta} \right\}$$

be the set of suboptimal classifiers at time t, where $\theta < 0$, $A > 0$ are parameters that are only used in the analysis of the regret and do not need to be known by the IMS. First, we will give regret bounds that depend on values of θ and A and then we will optimize over these values to find the best bound. Let $\mathcal{W}(t) := \{N_{p(t)}(t) > D(t)\}$ be the event that there is adequate number of samples to form accurate accuracy estimates for the set the context belongs to at time t. We call time t for which $N_{p(t)}(t) > D(t)$, a *good* time. All other times are *bad* times.

The regret given in (1) can be written as a sum of three components: $R(T) = \mathrm{E}[R^a(T)] + \mathrm{E}[R^s(T)] + \mathrm{E}[R^n(T)]$, where $R^a(T)$ is the *active learning regret*, which is the regret due to costs of obtaining the true label by time T plus the regret due to inaccurate estimates in bad times, $R^s(T)$ is the regret due to suboptimal classifier selections in good times by time T and $R^n(T)$ is the regret due to near optimal classifier selections in good times by time T, which are all random variables. In the following lemmas we will bound each of these terms separately. The following lemma bounds $\mathrm{E}[R^a(T)]$. Due to space limitations, the some of the proofs are given in our online technical report [50].

Lemma 1: When the IMS runs UPCE with parameters $D(t) = c^{\eta} t^z \log t$ and $m_T = \lceil T^{\gamma} \rceil$, where $0 < z < 1$, $\eta < 0$ and $0 < \gamma < 1/D$, we have

$$\begin{aligned}\mathrm{E}[R^a(T)] &\leq (c+1) \sum_{p=1}^{(m_T)^D} \lceil c^{\eta} T^z \log T \rceil \\ &\leq (c^{\eta+1} + c^{\eta}) 2^D T^{z+\gamma D} \log T + (c+1) 2^D T^{\gamma D}.\end{aligned}$$

Proof: See [50]. ■

We would like to note that this is the worst-case regret due to active learning. In practice, some regions of the context space (some hypercubes) may have only a few context arrivals, hence active learning is not required to be performed for those hypercubes for $\lceil c^{\eta} T^z \log T \rceil$ times. From Lemma 1, we see that the regret due to active learning is linear in the number of hypercubes $(m_T)^D$, hence exponential in parameter γ and z. We conclude that z and γ should be small enough to achieve sublinear regret in *active learning* steps. Moreover, since $\eta < 0$, this part of regret only sublinearly depends on c. We will show later that our algorithms can achieve regret that only scales with cubic rot of c, hence the performance scales well when the active learning cost is high.

Let $\mathcal{E}_{f,p}(t)$ denote the set of (*realized*) rewards (1 for correct prediction, 0 for incorrect prediction) obtained from classifier f for contexts in p for time slots the true label is obtained by time t. Clearly we have $\hat{\pi}_{f,p}(t) = \sum_{r \in \mathcal{E}_{f,p}(t)} r/|\mathcal{E}_{f,p}(t)|$. Each of the realized rewards are sampled from a context dependent distribution. Hence, those rewards are not identically distributed. In order to facilitate our analysis of the regret, we generate two different artificial i.i.d. processes to bound the deviation probability of $\hat{\pi}_{f,p}(t)$ from $\pi_f(x)$, $x \in p$. The first one is the *best* process in which rewards are generated according to a bounded i.i.d. process with expected reward $\overline{\pi}_{f,p}$, the other one is the *worst* process in which the rewards are generated according to a bounded i.i.d. process with expected reward $\underline{\pi}_{f,p}$. Let $\hat{\pi}^{\mathrm{b}}_{f,p}(z)$ denote the sample mean of the z samples from the best process and $\hat{\pi}^{\mathrm{w}}_{f,p}(z)$ denote the sample mean of the z samples from the worst process. We will bound the terms $\mathrm{E}[R^n(T)]$ and $\mathrm{E}[R^s(T)]$ by using these artificial processes along with the similarity information given in Assumption 1. Details are given in the proofs.

The following lemma bounds $\mathrm{E}[R^s(T)]$.

Lemma 2: When the IMS runs UPCE with parameters $D(t) = t^z \log t$ and $m_T = \lceil c^{\eta} T^{\gamma} \rceil$, where $0 < z < 1$, $\eta < 0$ and $0 < \gamma < 1/D$, given that $2L(\sqrt{D})^{\alpha} t^{-\gamma\alpha} + 2c^{-\eta/2} t^{-z/2} \leq At^{\theta}$ for $t = 1, \ldots, T$, we have $\mathrm{E}[R^s(T)] \leq n_c \beta_2 2^{D+1} T^{\gamma D}$.

Proof: See [50]. ■

From Lemma 2, we see that the regret increases exponentially with parameter γ. These two lemmas suggest that γ and z should be as small as possible, given the condition $2L(\sqrt{D})^{\alpha} t^{-\gamma\alpha} + 2c^{-\eta/2} t^{-z/2} \leq At^{\theta}$, is satisfied.

The following lemma bounds $\mathrm{E}[R^n(T)]$.

Lemma 3: When the IMS runs UPCE, we have $\mathrm{E}[R^n(T)] \leq \frac{AT^{1+\theta}}{1+\theta} + 3LD^{\alpha/2} T^{1-\alpha\gamma}$.

Proof: See [50]. ■

From Lemma 3, we see that the regret due to near optimal choices depends exponentially on θ which is related to negative of γ and z. Therefore γ and z should be chosen as large as possible to minimize the regret due to near optimal arms.

In the next theorem we bound the regret of the IMS by combining the above lemmas.

Theorem 1: When the IMS runs UPCE with parameters $D(t) = c^{-2/3} t^{2\alpha/(3\alpha+D)} \log t$ and $m_T = \lceil T^{1/(3\alpha+D)} \rceil$, we have

$$\begin{aligned}R(T) \leq\ & T^{\frac{D}{3\alpha+D}}\left(n_c\beta_2 2^{D+1} + (c+1)2^D\right)\\ &+ T^{\frac{2\alpha+D}{3\alpha+D}}\Bigg(\frac{(2LD^{\alpha/2}+2c^{1/3})}{(2\alpha+D)/(3\alpha+D)} + 3LD^{\alpha/2}\\ &\qquad + (c^{1/3}+c^{-2/3})2^D\log T\Bigg)\end{aligned}$$

i.e., $R(T) = \tilde{O}\left(c^{1/3}T^{\frac{2\alpha+D}{3\alpha+D}}\right)$.

Proof: The highest time orders of regret come from active learning and near optimal classifiers which are $\tilde{O}(T^{\gamma D+z})$, $O(T^{1-\alpha\gamma})$ and $O(T^{1+\theta})$ respectively. We need to optimize them with respect to the constraint $2LD^{\alpha/2}t^{-\gamma\alpha} + 2c^{-\eta/2}t^{-z/2} \leq At^{\theta}$, which is assumed in Lemma 2. The values that minimize the regret for which this constraint hold are $\theta = -z/2$, $\gamma = z/(2\alpha)$, $A = 2LD^{\alpha/2} + 2c^{-\eta/2}$ and $z = 2\alpha/(3\alpha + D)$. With these choices, the order of regret for near optimal classifier in c becomes $O(c^{-\eta/2})$. Since the order of regret in c is $O(c^{1+\eta})$ for active learning, these two terms are balanced for $\eta = -2/3$, making the order of total regret in c equal to $O(c^{1/3})$. Result follows from summing the bounds in Lemmas 1, 2 and 3. ∎

Remark 1: Although the parameter m_T of UPCE depends on T, we can make UPCE run independently of the final time T and achieve the same regret bound by using a well known doubling trick (see [43]). Consider phases $\tau \in \{1, 2, \ldots\}$, where each phase has length 2^{τ}. We run a new instance of algorithm UPCE at the beginning of each phase with time parameter 2^{τ}. Then, the regret of this algorithm up to any time T will be $\tilde{O}\left(T^{(2\alpha+D)/(3\alpha+D)}\right)$. Although doubling trick works well in theory, UPCE can suffer from cold-start problems. The algorithm we will define in the next section will not require T as an input parameter.

Remark 2: The learning algorithms proposed in this paper have the goal of minimizing the regret, which is defined in terms of prediction accuracies. However, in certain deployment scenarios, one might also be interested in minimizing false alarms, misdetections or a weighted sum of them. For example, in order to minimize misdetections, the IMS needs to learn the classifier with the lowest misdetection rate for each context. Since a misdetection can happen only if the prediction is 0 but the true label is 1, then the IMS does active learning only at time slots when it predicted 0. If the obtained true label is 1, then it updates the estimated misdetection probabilities of all classifiers. Note that, a misdetection in an active learning slot will not cause harm because the true label is recovered.

Remark 3: In this work we take the approach to have c as a design parameter which is set by the learner based on the tradeoff it assumes between the active learning cost and classification accuracy. For instance, such an approach is also taken in [28] in which the regret is written as a weighted sum of prediction accuracy and label observation cost. As can be seen from Theorem 1, although we write the regret due to active learning and regret due to incorrect predictions together as a single term, the active learning part of the regret only comes from Lemma 1. Since the costs due to active learning and near-optimal classifier selections are balanced in Theorem 1, UPCE achieves the optimal growth rate (in terms of the time order) both for the active learning regret and the regret due to near-optimal and suboptimal classifier selections. It is also possible to interpret c as the absolute cost of active learning with fixed budget. Recall that Lemma 1 gives the active learning cost of UPCE when using a control function $D(t) = c^{\eta}t^{z}\log t$. If the learner has a final time horizon T and a budget C with an absolute active learning cost c, then it can optimize the η and z parameters in order to satisfy the budget constraint. However, the regret bound given in Theorem 1 would be different since η or z are set according to the active learning budget in this case.

The regret bound proved in Theorem 1 is sublinear in time which guarantees convergence in terms of the average reward, i.e., $\lim_{T\to\infty} R(T)/T = 0$. For a fixed α, the regret becomes linear in the limit as D goes to infinity. On the contrary, when D is fixed, the regret decreases, and in the limit, as α goes to infinity, it becomes $O(T^{2/3})$. This is intuitive since increasing D means that the dimension of the context increases and therefore the number of hypercubes to explore increases. While increasing α means that the level of similarity between any two pairs of contexts increases, i.e., knowing the accuracy of classifier f in one context yields more information about its accuracy in another context. Also as for large c, we see that the number of times active learning is performed decreases. This changes the estimated accuracies, and the tradeoff is captured by choosing a larger A, i.e., defining a coarser near optimality.

V. Adaptively Partitioned Contextual Experts

In real-world image stream mining applications, based on the temporal correlations between the images, the image arrival patterns can be non-uniform. Intuitively it seems that the loss due to partitioning the context space into different sets and learning independently for each of them can be further minimized when the learning algorithm inspects the regions of the context space with large number of context arrivals more carefully. In this section we propose such an algorithm called *Adaptively Partitioned Contextual Experts* (APCE), whose pseudocode is given in Fig. 5. In the previous section the finite partition of hypercubes $\mathcal{P}_T$ is formed by UPCE at the beginning by choosing the slicing parameter m_T. Differently, APCE adaptively generates the partition by learning from the context arrivals. Similar to UPCE, APCE independently estimates the accuracies of the classifiers for each set in the partition.

Let $\mathcal{P}(t)$ be the IMS's partition of $\mathcal{X}$ at time t and $p(t)$ denote the set in $\mathcal{P}(t)$ that contains $x(t)$. Using APCE, the IMS starts with $\mathcal{P}(1) = \{\mathcal{X}\}$, then divides $\mathcal{X}$ into sets with smaller sizes as time goes on and more contexts arrive. Hence the cardinality of $\mathcal{P}(t)$ increases with t. This division is done in a systematic way to ensure that the tradeoff between the variation of classifier accuracies inside each set and the number of past observations that are used in accuracy estimation for each set is balanced. As a result, the regions of the context space with a lot of context arrivals are covered with sets of smaller sizes than regions of contexts space with few context arrivals. In other words, APCE *zooms* into the regions of context space with

```
Adaptively Partitioned Contextual Experts (APCE):
Input: $\alpha$, $\rho$, $\eta$, $B$
Initialization: $\mathcal{P}(1) = \{[0,1]^D\}$, Run **Initialize**($\mathcal{P}(1)$)
**while** $t \geq 1$ **do**
    Find the set in $\mathcal{P}(t)$ that $x(t)$ belongs to, i.e., $p(t)$
    Set $p = p(t)$
    Set $a_{\mathrm{pr}}(t) = \arg\max_{f \in \mathcal{F}} \hat{\pi}_{f,p}$
    Set $\hat{y}(t)$ be the prediction of classifier $a_{\mathrm{pr}}(t)$
    **if** $N_p \leq D(p,t)$ **then**
        Obtain the true label $y(t)$ by paying cost $c$, i.e., $a_{\mathrm{lr}}(t) = 1$
        **for** $f \in \mathcal{F}$ **do**
            $\hat{\pi}_{f,p} = \frac{\hat{\pi}_{f,p} N_p + \mathrm{I}(\hat{y}_f(t) = y(t))}{N_p + 1}$
            $N_p + +$
        **end for**
    **else**
        Do not ask for the true label, i.e., $a_{\mathrm{lr}}(t) = 0$
    **end if**
    $N_p^{\mathrm{ttl}} + +$
    **if** $N_p^{\mathrm{ttl}} \geq B 2^{\rho l(p)}$ **then**
        Create $2^D$ level $l(p)+1$ child hypercubes denoted by $\mathcal{A}_p^{l(p)+1}$
        Run **Initialize**($\mathcal{A}_p^{l(p)+1}$)
        $\mathcal{P}(t+1) = \mathcal{P}(t) \cup \mathcal{A}_p^{l(p)+1} - \{p\}$
    **end if**
    $t = t + 1$
**end while**

**Initialize**($\mathcal{B}$):
**for** $p \in \mathcal{B}$ **do**
    Set $N_p^{\mathrm{ttl}} = 0$, $N_p = 0$, $\hat{\pi}_{f,p} = 0$ for $f \in \mathcal{F}$
**end for**
```

Fig. 5. Pseudocode for APCE and its initialization module.

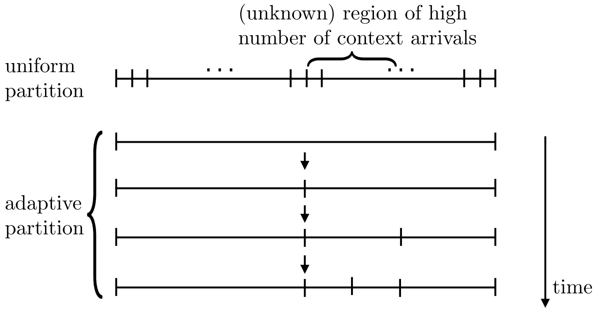

Fig. 6. An illustration showing how the partition of APCE differs from the partition of UPCE for $D = 1$. As contexts arrive, APCE *zooms* into regions of high number of context arrivals.

large number of arrivals. An illustration that shows partition of UPCE and APCE is given in Fig. 6 for $D = 1$. As we discussed in the Section II the zooming idea have been used in a variety of multi-armed bandit problems [43]–[46], [51]. However, the creation of hypercubes and the time spent in active learning in each hypercube is different from these works, which do not consider the problem of actively asking the labels. Instead, they use index-based policies in which the index of each arm is updated at the end of every time slot, since the reward feedback for the selected arm is always received at the end of the time slot.

The sets in the adaptive partition of the IMS are chosen from hypercubes with edge lengths coming from the set $\{1, 1/2, 1/2^2, \ldots\}$. We call a D dimensional hypercube which has edges of length 2^{-l} a level l hypercube (or level l set). For a hypercube p, let $l(p)$ denote its level. For $p \in \mathcal{P}(t)$ let $\tau_i(p)$ be the time p is activated and $\tau_f(p)$ be the time p is deactivated by the IMS. We will describe the activation and deactivation process of hypercubes after defining the counters of APCE which are initialized and updated differently than UPCE. For $p \in \mathcal{P}(t)$, $N_p(t)$ counts the number of context arrivals in set p from times $\{\tau_i(p), \ldots, t-1\}$ for which the IMS obtained the true label, and $N_p^{\mathrm{ttl}}(t)$ counts the number of all context arrivals in set p from times $\{\tau_i(p), \ldots, t-1\}$.

The IMS updates its partition $\mathcal{P}(t)$ as follows. At the end of each time slot t, the IMS checks if $N_p^{\mathrm{ttl}}(t+1)$ exceeds a threshold $B 2^{\rho l(p(t))}$, where $B > 0$ and $\rho > 0$ are parameters of APCE. If $N_p^{\mathrm{ttl}}(t+1) \geq B 2^{\rho l(p(t))}$, the IMS divides $p(t)$ into 2^D level $l(p(t))+1$ hypercubes, activates these hypercubes by initializing their counters to zero and adding them to $\mathcal{P}(t+1)$, and deactivates $p(t)$ by removing it from $\mathcal{P}(t+1)$.

The IMS keeps a control function $D(p,t)$ for each $p \in \mathcal{P}(t)$ to decide when to obtain the true label. We set $D(p,t) = c^{\eta} 2^{2\alpha l(p)} \log t$, $\eta < 0$ and will prove that it is the optimal value to balance the cost of active learning with estimation accuracy. At time t, if the number of times the IMS obtained the true label for contexts in $p(t)$ is less than or equal to $D(p(t), t)$, i.e., $N_{p(t)}(t) \leq c^{\eta} 2^{2\alpha l(p(t))} \log t$, then, the IMS asks for the true label, otherwise it does not ask for the true label. For $p \in \mathcal{P}(t)$, let $\mathcal{E}_{f,p}(t)$ denote the set of rewards (*realized* accuracy) obtained from classifier f for contexts in p at times in $\{\tau_i(p), \ldots, t-1\}$ when the true label is obtained. Clearly we have $\hat{\pi}_{f,p}(t) = \sum_{r \in \mathcal{E}_{f,p}(t)} r / |\mathcal{E}_{f,p}(t)|$. The classifier whose prediction is followed by the IMS at time t is $a_{\mathrm{pr}}(t) = \arg\max_{f \in \mathcal{F}} \hat{\pi}_{f,p(t)}(t)$. We will analyze the regret of APCE in the next subsection.

A. Analysis of the Regret of APCE

Our analysis for UPCE in Section IV was for worst-case context arrivals. In this section we analyze the regret of APCE under different types of context arrivals. To do this we will bound the regret of APCE in a level l hypercube, and then give the bound in terms of the total number of level l hypercubes activated by time T.

We start with a simple lemma which gives an upper bound on the highest level hypercube that is active at any time t.

Lemma 4: When the IMS runs APCE, all the active hypercubes $p \in \mathcal{P}(t)$ at time t have at most a level of $\lceil (\log_2 t)/\rho \rceil + 1$.

Proof: See [50]. ∎

For a set p, $\overline{\pi}_{f,p}$, $\underline{\pi}_{f,p}$, x_p^* and $f^*(p)$ are defined the same way as in Section IV-A. Let

$$\mathcal{L}_p := \left\{ f \in \mathcal{F} \text{ such that } \underline{\pi}_{f^*(p),p} - \overline{\pi}_{f,p} > A 2^{-\alpha l(p)} \right\}$$

be the set of suboptimal classifiers at time t, where $A > 0$ is a parameter that is just used in the analysis and not an input to the algorithm. Let $\mathcal{W}(t) := \{N_{p(t)} > c^{\eta} 2^{2\alpha l(p(t))} \log t\}$, be the event that there is adequate number of samples to form accurate accuracy estimates for the set $p(t)$ the context belongs to at time t. Similar to Section IV-A, we call time t for which $N_{p(t)} > c^{\eta} 2^{2\alpha l(p(t))}$, a *good* time. All other times are *bad* times.

For a hypercube p, the regret incurred from its activation to time T can be written as a sum of three components: $R_p(T) = \mathrm{E}[R_p^a(T)] + \mathrm{E}[R_p^s(T)] + \mathrm{E}[R_p^n(T)]$. where $R_p^a(T)$ is the regret due to costs of obtaining the true label plus the regret due to inaccurate estimates in bad times, $R_p^s(T)$ is the regret due to suboptimal classifier selections in good times and $R_p^n(T)$ is the regret due to near optimal classifier selections in good times, from the activation of hypercube p till time T. In the following lemmas we will bound each of these terms separately. The following lemma bounds $\mathrm{E}[R_p^a(T)]$.

Lemma 5: When the IMS runs APCE, for a level l hypercube p, we have $\mathrm{E}[R_p^a(T)] \leq (c^{1+\eta} + c^{\eta})2^{2\alpha l} \log T + (c+1)$.

Proof: See [50]. ■

The following lemma bounds $\mathrm{E}[R_p^s(T)]$.

Lemma 6: When the IMS runs APCE, for a level l hypercube p, given that, $2L\left(\sqrt{D}/2^{-l}\right)^{\alpha} + 2c^{-\eta/2}2^{-\alpha l} - A2^{-\alpha l} \leq 0$, we have $\mathrm{E}[R_p^s(T)] \leq 2n_c\beta_2$.

Proof: See [50]. ■

In the next lemma we bound $\mathrm{E}[R_p^n(T)]$.

Lemma 7: When the IMS runs APCE, for a level l hypercube p, we have $\mathrm{E}[R_p^n(T)] \leq (3LD^{\alpha/2} + A)B2^{l(\rho-\alpha)}$.

Proof: See [50]. ■

Next, we combine the results from Lemmas 5, 6 and 7 to obtain our regret bound. Let $K_l(T)$ be the number of level l hypercubes that are activated by time T. We know that $K_l(T) \leq 2^{Dl}$ for any l and T. Moreover, from the result of Lemma 4, we know that $K_l(T) = 0$ for $l > \lceil(\log_2 t)/\rho\rceil + 1$. Although, these bounds give an idea about the range of values that $K_l(T)$ can take, the actual values of $K_l(T)$ depends on the context arrival process, α and B, and can be exactly computed for a sample path of context arrivals.

Theorem 2: When the IMS runs APCE with parameters $\rho = 3\alpha$, $\eta = -2/3$ and $B = 1$, we have

$$R(T) \leq \sum_{l=1}^{\lceil(\log_2 t)/\rho\rceil+1} K_l(T)\left(2^{2\alpha l}(A^* + (c^{1/3} + c^{-2/3})\log T) + 2n_c\beta_2 + c + 1\right)$$

where $A^* = 5LD^{\alpha/2} + 2c^{1/3}$.

Proof: Consider a hypercube p. The highest orders of regret come from $\mathrm{E}[R_p^a(T)]$ and $\mathrm{E}[R_p^n(T)]$. The former is in the order of $\tilde{O}(2^{2\alpha l})$ and the latter is in the order of $O(2^{(\rho-\alpha)l})$. These two are balanced for $\rho = 3\alpha$. Although, choosing ρ smaller than 3α will not make the regret in hypercube p larger, it will increase the number of hypercubes activated by time T, causing an increase in the regret. Therefore, since we sum over all activated hypercubes, it is best to choose ρ as large as possible, while balancing the regrets due to $\mathrm{E}[R_p^a(T)]$ and $\mathrm{E}[R_p^n(T)]$. In order for condition in Lemma 6 to hold we set $A = 2LD^{\alpha/2} + 2c^{-\eta/2}$ and optimize over η. ■

The regret bound derived for APCE in Theorem 2 is quite different from the regret bound of UPCE in Theorem 1. APCE's bound is a more general form of bound whose exact value depends on how the contexts arrive, hence $K_l(T)$, $l = 1, \ldots, \lceil(\log_2 t)/\rho\rceil + 1$. We will show in the next corollary that for the worst-case context arrivals in which the arrivals are uniformly distributed over the context space, the time order of the regret bound reduces to the bound in Theorem 1.

Corollary 1: When APCE is run with parameters $\rho = 3\alpha$, $\eta = -2/3$ and $B = 1$, if the context arrivals by time T is uniformly distributed over the context space, we have

$$R(T) \leq T^{\frac{2\alpha+D}{3\alpha+D}} 2^{D+2\alpha}(A + (c^{1/3} + c^{-2/3})\log T) + T^{\frac{D}{3\alpha+D}} 2^{D}(2n_c\beta_2 + c + 1)$$

where $A^* = 5LD^{\alpha/2} + 2c^{1/3}$. Hence $R(T) = \tilde{O}\left(c^{1/3}T^{\frac{2\alpha+D}{3\alpha+D}}\right)$.

Proof: See [50]. ■

VI. Numerical Results

In this section we evaluate the performance of the proposed algorithms in a breast cancer diagnosis application. In general, our proposed algorithms can be used in any image stream mining application.

A. Description of the Dataset

The breast cancer dataset is taken from UCI repository [14]. The dataset consists of features extracted from the images of FNA of breast mass, that gives information about size, shape, uniformity, etc., of the cells. Each case is labeled either as malignant or benign. We assume that images arrive to the IMS in an online fashion. At each time slot, our learning algorithms operate on a subset of the features extracted from the images to make a prediction about the tumor type. We assume that the actual outcome can only be observed when the true label is asked (surgical biopsy) by paying an active learning cost $c > 0$. The number of instances is 50000. About 69% of the images represent benign cases while the rest represent malignant cases. We say that an error happens when the prediction is wrong, a misdetection happens when a malignant case is predicted as benign, and a false alarm happens when a benign case is predicted as malignant.

B. Simulations With Pre-Trained Base Classifiers

For the numerical results in this subsection, 6 logistic regression classifiers, each trained with a different set of 10 images are used as base classifiers both by UPCE and APCE. These trainings are done by using 6 features extracted from each image. The error rate of these classifiers on test data are 15.6, 10.8, 68.6, 31.5, 14 and 16.3 percent. It is obvious that none of these classifiers work well for all instances. Our goal in this subsection is to show how UPCE and APCE can achieve much higher prediction accuracy (lower error rate) than each individual classifier, by exploiting the contexts of the images when deciding the prediction of which classifier to follow. Essentially, UPCE and APCE learns the context dependent accuracies of the classifiers. For each image, we take one of the extracted features as context, hence $D = 1$. We use the same type of feature as context for all the images. This feature is also present in the training set of the logistic regression classifiers (it is one of the 6 features).

One of our benchmarks is the No Context Experts (NCE) algorithm which uses the control function of UPCE for active learning, but does not exploit the context information in

TABLE I

INPUT PARAMETERS FOR UPCE AND APCE USED IN THE SIMULATIONS

	c	α	ρ	$D(t)$ for UPCE $D(p,t)$ for APCE ($B=1$)	m_T
U1 (UPCE)	1	1	N/A	$t^{1/2} \log t/32$	$\lceil T^{1/4} \rceil$
U2 (UPCE)	5	1	N/A	$5^{-2/3} t^{1/2} \log t/32$	$\lceil T^{1/4} \rceil$
A1 (APCE)	1	1	3	$2^{2l(p)} \log t/64$	N/A
A2 (APCE)	5	1	3	$5^{-2/3} 2^{2l(p)} \log t/64$	N/A
S1 (UPCE)	1	1	N/A	$t^{\frac{2\alpha}{3\alpha+D}} \log t/16$	$\lceil T^{\frac{1}{3\alpha+D}} \rceil$
S2 (UPCE)	1	1	N/A	$t^{1/4} \log t/16$	$\lceil T^{1/8} \rceil$

selecting the classifier to follow. NCE learns the classifier accuracies by keeping and updating a single sample mean accuracy estimate for each of them, not taking into account the context provided along with an image.

Our other benchmarks are ensemble learning methods including Average Majority (AM) [36], Adaboost (Ada) [37], Fan's Online Adaboost (OnAda) [38], the Weighted Majority (WM) [39] and Blum's variant of WM (Blum) [40]. The goal of these methods is to create a strong (high accuracy) classifier by combining predictions of weak (low accuracy) classifiers, which are the base classifiers in our simulations. These are different than UPCE and APCE, which exploit contextual information to learn context based specialization of weak classifiers to create a strong predictor.

AM simply follows the prediction of the majority of the classifiers, hence it does not perform active learning. Ada is trained a priori with 1500 images, in which the labels of these images are used to update the weight vector. Its weight vector is fixed during the test phase (it is not learning online), hence no active learning is performed during the test phase. In contrast, OnAda always receives the true label at the end of each time slot. It uses a time window of 1000 past observations to retrain its weight vector. WM and Blum uses a control function similar to the control function of UPCE to decide when to ask for the label. The control function we use for other methods is $D(t) = t^{1/2} \log t$. Assuming $\alpha = 1$, this gives the optimal order of active learning in Theorem 1 for $D = 1$.

The parameter values used for UPCE and APCE for the simulations in this subsection are given in the first four rows of Table I. Simulation results are given in Table II. In order to have a fair comparison of our algorithms and other methods, we compare for active learning cost $c = 1$. For each simulation criteria, the first number in the parenthesis shows the rank of the algorithm over all algorithms. For UPCE and APCE, the second number in the parenthesis shows the percent improvement over the best algorithm among the other algorithms. We see that in terms of the error rate UPCE and APCE are significantly better than NCE (about at least 70% reduction in the error rate). They also outperform the best logistic regression classifier by at least 68% in terms of the reduction in the error rate. UPCE and APCE are also better than all the ensemble learning methods (about at least 25% reduction in error rate). Although Ada and online OnAda are better than UPCE and APCE in terms of the misdetection rates, they have significantly higher false alarm rates. The disadvantage of Ada is that it does not learn online, it performs active learning only for the samples at the beginning. Although it works well for this particular dataset, its performance will be poor when the initial samples do not represent the general population well. OnAda can deal with this, but it constantly retrains its weights by actively asking for the labels, hence its active learning rate cannot decrease over time. The number of times active learning is performed by UPCE and APCE is 1140 and 1341 respectively, which is lower than the number of true labels used by all ensemble learning methods to train their weights (2470 WM and Blum).

Finally we compare the performance of UPCE and APCE for different values of active learning costs, $c = 1$ (U1 and A1) and $c = 5$ (U2 and A2). The results in Table II show that UPCE and APCE adaptively decrease their active learning rate to compensate for the increase in the cost of obtaining the label. Although the cost of obtaining the label increases by 500%, the total cost of active learning for UPCE and APCE increases less than 100% due to this adaptation. This results in a significant reduction in the number of active learnings performed by UPCE and APCE, however, the increase in error rates due to this is less than 30% for both algorithms.

Remark 4 (On the Choice of Base Classifiers): Although we used logistic regression classifiers as our base classifiers in this section, our algorithms will work with any base classifier. In order to obtain theoretical performance guarantees, the only assumption we require on classifiers concerns their accuracies for images with different contexts, which is formalized in Assumption 1. Under this assumption, our algorithms are guaranteed to converge to the performance of the best context-specialized classifier $f^*(x)$ for all $x \in \mathcal{X}$. Hence, using our algorithms, the learner is guaranteed to perform as good as the best context-specialized classifier in the long run. As a rule of thumb, in order to get a good prediction accuracy for every context $x \in \mathcal{X}$, there must exist at least one base classifier (fixed or learned) which has a high prediction accuracy for x. Therefore, it is good to have a diverse set of classifiers with a diverse set of parameters, since our algorithms can adaptively learn their contextual specializations. Creating context-specialized classifiers can be done by learning without base classifiers (Section VII-A) and/or re-training base classifiers on different parts of the context space (Section VII-B).

C. Simulations for UPCE Without Base Classifiers

Different from the previous subsection, where UPCE learns which classifier to follow given a context, in this subsection UPCE directly learns which prediction to make given a context. Due to this, UPCE can be seen as an online learning classifier, which updates itself based on the context arrivals and the labels that have been obtained so far. Equivalently, we can view this scenario as UPCE having two base classifiers, one which always predicts benign and the other which always predicts malignant.

We simulate UPCE for two different sets of parameter values S1 and S2 that are given in Table I. In S1, the control function and the size of hypercubes are adjusted according to

TABLE II

COMPARISON OF UPCE AND APCE WITH ENSEMBLE LEARNING METHODS AND NCE FOR PARAMETER SETTINGS U1, U2, A1 AND A2 AT $T = 50000$

Abbreviation	Name of the Scheme	Reference	Performance				
			error %	missed %	false %	active learning cost	number of active learnings
AM	Average Majority	[36]	8.22	17.20	4.09	N/A	N/A
Ada	Adaboost	[37]	4.60 (3)	3,82 (1)	4.97	1500	1500
OnAda	Fan's Online Adaboost	[38]	4.68 (4)	4.07 (2)	4.95	N/A	N/A
WM	Weighted Majority algorithm	[39]	21.46	12.16 (5)	25.74	2470	2470
Blum	Blum's variant of WM	[40]	11.18	27.12	3.86 (4)	2470	2470
NCE	No Context Experts	benchmark	12.33	32.45	3.09 (3)	2470	2470
UPCE (U1)	Uniformly Partitioned Contextual Experts	our work	3.35 (1, 27%)	6.08 (4)	2.1 (1, 32%)	1140	1140
UPCE (U2)	”	our work	4.11	7.71	2.46	1950	390
APCE (A1)	Adaptively Partitioned Contextual Experts	our work	3.44 (2, 25%)	5.48 (3)	2.5 (2, 19%)	1341	1341
APCE (A2)	”	our work	4.46	8.21	2.74	2465	493

TABLE III

SIMULATION RESULTS FOR UPCE FOR PARAMETER SETS S1 AND S2 AT $T = 50000$. err = ERROR RATE, alrn = ACTIVE LEARNING RATE, mis = RATE OF MISSED DETECTIONS, false = RATE OF FALSE ALARMS, mis-na (false-na) = RATE OF MISDETECTIONS (FALSE ALARMS) AT TIME SLOTS EXCEPT ACTIVE LEARNING SLOTS, ncube = NUMBER OF HYPERCUBES

	err %	alrn %	mis %	false %	mis-na %	false-na %	ncube
S1 ($D = 3$)	2.96	7.95	5.40	1.84	4.70	1.31	7^3
S1 ($D = 6$)	0.77	5.19	0.66	0.82	0.60	0.45	2^{12}
S2 ($D = 3$)	4.14	1	6.12	3.23	6.03	3.13	2^6
S2 ($D = 6$)	0.82	7	0.74	0.85	0.64	0.46	2^{10}

the optimal values given in Theorem 1 for similarity exponent $\alpha = 1$. In S2, the control function and the size of hypercubes are chosen independently from the dimension of the context space and the similarity exponent. While APCE and UPCE takes the similarity exponent as given, the similarity constant L is not required to be known by the algorithms. Given any similarity metric with exponent $\alpha > 1$ and constant $L > 0$, it is possible to generate a *relaxed* similarity metric with exponent 1 and constant $\tilde{L} > 0$ such that

$$\begin{aligned} &|\pi_f(x) - \pi_f(x')| \leq L||x - x'||^\alpha \\ &\qquad \Rightarrow |\pi_f(x) - \pi_f(x')| \leq \tilde{L}||x - x'|| \end{aligned}$$

for all $x, x' \in \mathcal{X}$ and $f \in \mathcal{F}$. Therefore, if no prior information exists about the similarity metric both UPCE and APCE can set $\alpha = 1$.

As we discussed in Section IV and V, there is a tradeoff between active learning cost and prediction accuracy in setting $D(t)$ and m_T for UPCE and $D(p, t)$ and ρ for APCE. For instance, by choosing a larger $D(t)$ UPCE increases its probability of choosing the optimal classifier at time steps it exploits, while it incurs larger active learning cost. Similarly, by choosing a larger m_T it decreases the errors in accuracy estimates due to the variation of the classifier accuracies for different context values (due to Assumption 1), while the number of past context observations that can be used to form these estimates decreases since the size of each hypercube is inversely proportional to m_T. Recall that the parameter choice in Theorem 1 yields the optimal tradeoff between these events for an arbitrary context arrival process. In practice, if the time horizon of interest is large, it is better to choose $D(t)$ and m_T according to the theoretical values, since it guarantees the optimal tradeoff between the active learning cost and classification accuracy under any possible context arrival process. However, if the learner aims to maximize the performance at the very early stages, it may set $D(t)$ to a higher value (which lets it observe more labels) and m_T to a smaller value (which lets it use a larger set of past observations for each hypercube).

The active learning rate (percentage of time when the true label is asked), number of hypercubes, error, misdetection and false alarm rates for UPCE are given in Table III as a function of the dimension of the context D. It is observed that the computational complexity of UPCE increases exponentially with D, due to the increase in the number of hypercubes. We can see that the prediction accuracy significantly increases with D. This is due to the fact that the information UPCE gets about each image increases with D, hence UPCE learns to make better predictions. For S2, when 6 features are used as contexts, the error rate is 0.82%, which is significantly lower than using 3 features as contexts, that results in an error rate of 4.14%. However, the number of hypercubes for $D = 3$ is 1/64th of the number of hypercubes for $D = 6$, and the total cost of active learning for $D = 3$ is only about 13% of the total cost of active learning for $D = 6$. Hence, there is a clear tradeoff between active learning cost and prediction accuracy. Another observation is the fact that the misdetection and false alarm rates at time slots which are not active learning slots are lower than the total misdetection and false alarm rates. This means that UPCE is more accurate on time slots when it does not need to perform active learning compared to time slots that it needs to perform active learning. For this application, since the true label is observed at the time slots when active learning is performed (surgical biopsy), number of false alarms and misdetections in these slots do not have a negative consequence on the patient's health.

Comparing the results for S1 and S2, we see that for all types of contexts the error rate is lower for S1. Since the order of active learning constant is kept fixed in S2 independent of D, the total active learning cost increases with D. In contrast, for S1, the active learning cost has a non-uniform behavior as a function of D, and is much lower compared to S2 when the context dimension is high ($D = 6$).

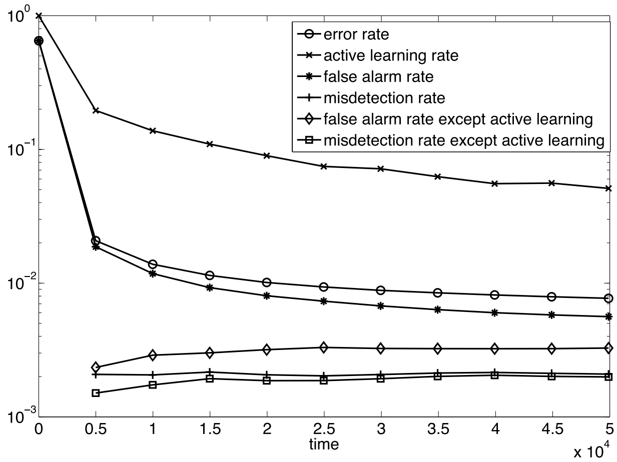

Fig. 7. The error, active learning, misdetection and false alarm rates of UPCE for $D = 6$ and for the parameter values given in S1, as a function of time.

This is due to the fact that the rate of active learning for each hypercube is in the order of $\tilde{O}(t^{\frac{2\alpha}{3\alpha+D}})$.

So far we have talked about the performance of UPCE at the final time. Fig. 7 shows the average active learning cost, error, misdetection and false alarm rates of UPCE over time. We see that the performance of UPCE improves over time, and the largest improvement is in the first 5000 time slots. As more images arrive, both the rate of actively asked labels and error decrease.

VII. Discussion

A. Learning Without Base Classifiers

Both UPCE and APCE can directly learn to make the best prediction corresponding to each set in the partition of the context space that they generate. In order to do this, they need to form two classifiers for each partition, one that always predicts 1 and the other that always predicts 0. Then, they will actively learn the accuracy of these classifiers in order to find out the best prediction to make for that region of the context space. In this case, the feature vector can be taken as the context vector to learn the best prediction for each *region* of the feature space (as shown in the numerical results of Section VI-C). One limitation of this approach is that, the dimension of the feature space can be large, which will result in slow but asymptotically optimal learning as shown in the regret bounds in Theorem 1 and Theorem 2. An interesting research direction is to learn a low dimensional set of features that are relevant to the prediction (given that such low dimensional representations exist) in order to increase the learning speed. This is discussed in Section VII-D.

B. Re-Training Classifiers

As opposed to UPCE, APCE generates its partition of the context space on-the-fly, based on the history of context arrivals. As described in the pseudocode of APCE in Fig. 5, whenever the number of arrivals to a particular level p in the context partition of APCE exceeds the specified threshold $(N_p^{\text{ttl}} \geq B2^{pl(p)})$, p is divided into 2^D level $l(p) + 1$ hypercubes. When this division is performed, classifier accuracy estimates $\hat{\pi}_{f,p}$ are destroyed and for every new hypercube p' in set $\mathcal{A}_p^{l(p)+1}$, accuracy of f needs to be re-estimated. This structure of APCE allows a classifier f to be re-trained within region p, without altering its operation in other parts of the context space. Recall from Section III-B that the prediction rule of classifier f is given by $Q_f(\cdot)$. With the new modification, we can denote the prediction rule of classifier f in the region $p \in \mathcal{P}(t)$ of the context space by $Q_{f,p}(\cdot)$. For instance, $Q_{f,p}(\cdot)$ can be determined by using the history of past context arrivals to p at the time p was created. The snapshots given in Fig. 8 show the prediction rules of classifier f at three different points in time. As can be seen from this figure, classifier f is re-trained at each newly created set in the partition, hence, it can *specialize* as more contexts arrive. This type of re-trainings will not change the regret bound derived in Theorem 2. Because, when deriving that regret bound, we first bounded the regret within each hypercube p that is generated by APCE, and then summed over all the possible hypercubes that can be generated by APCE. Due to this separation technique that is used in the proof, it also holds when the classifiers are re-trained.

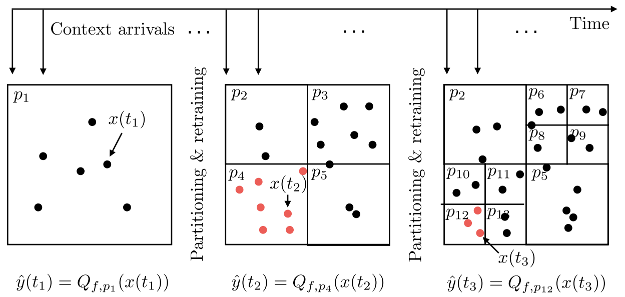
Context arrivals . . .
. . .
. . .
Time
Partitioning & retraining
Partitioning & retraining
p_1
$x(t_1)$
p_2
p_3
p_4
p_5
$x(t_2)$
p_2
p_6
p_7
p_8
p_9
p_{10}
p_{11}
p_5
p_{12}
p_{13}
$x(t_3)$
$\hat{y}(t_1) = Q_{f,p_1}(x(t_1))$
$\hat{y}(t_2) = Q_{f,p_4}(x(t_2))$
$\hat{y}(t_3) = Q_{f,p_{12}}(x(t_3))$

Fig. 8. APCE re-training classifier f based on the past arrivals to the newly created sets. Red dots indicate the contexts that are used in the re-training phase of classifier f for sets p_4 and p_{12} respectively.

C. Dynamic Control Functions

Recall that the control functions used by UPCE and APCE are deterministic, which implies that a prefixed maximum amount of active learning will be applied up to a certain point in time in each hypercube.

Intuitively, the number of active learning steps can be adjusted according to the estimated suboptimality gap between the classifier with the highest accuracy and the other classifiers. In this subsection we will show how this can be done for APCE. Let $\mathcal{A}_{\text{pr}}^*(t) := \arg\max_{f\in\mathcal{F}} \hat{\pi}_{f,p(t)}(t)$ be the set of *estimated optimal* classifier(s) at time t, where $p(t)$ is the set in learner i's partition that contains $x(t)$. Let

$$\hat{\mathcal{U}}_p(t) := \{f \in \mathcal{F} : \hat{\pi}_{f^*,p}(t) - \hat{\pi}_{f,p}(t) \leq A2^{-\alpha l(p)}\}$$

where $f^* \in \mathcal{A}_{\text{pr}}^*(t)$ and the value of A is given in Theorem 2. The estimated suboptimality of a classifier $f \in \mathcal{F} - \mathcal{A}_{\text{pr}}^*(t)$ at time t is defined as $\hat{\Delta}_{f,p(t)}(t) := \hat{\pi}_{f^*,p(t)}(t) - \hat{\pi}_{f,p(t)}(t)$. Let the estimated minimum suboptimality gap at time t be $\hat{\Delta}_{\min,p(t)}(t) := \min_{f\in\mathcal{F}-\hat{\mathcal{U}}_{p(t)}(t)} \hat{\Delta}_{f,p(t)}(t)$, when $\mathcal{F}-\hat{\mathcal{U}}_{p(t)}(t) \neq \emptyset$, and $\hat{\Delta}_{\min,p(t)}(t) := 2^{-\alpha l(p(t))}$ when $\mathcal{F}-\hat{\mathcal{U}}_{p(t)}(t) = \emptyset$. Instead of keeping a deterministic control function, APCE can be modified to keep separate control functions for each hypercube p, based on $\hat{\Delta}_{\min,p}(t)$. The idea is to adjust the number of times true label is obtained within each hypercube in a way that is inversely proportional to $\hat{\Delta}_{\min,p}(t)$ such that there will not be any suboptimal classifier selections at exploitations with a high probability (although near-optimal classifier selections are allowed). For instance, by letting $D(p,t) = O(\log t/\hat{\Delta}^2_{\min,p}(t))$, the number of active learning steps can be made dependent on the suboptimality gap between the optimal classifiers and suboptimal classifiers, instead of the original $D(p,t) = O(\log t/(2^{-\alpha l(p)})^2)$, which only depends on the variation of classifier accuracies within p due to Assumption 1. This will result in much smaller number of active learning steps when $\hat{\Delta}_{\min,p(t)}(t) \gg 2^{-\alpha l(p(t))}$ for most of the time steps, which will hold when the (true) difference between the accuracy of the optimal classifier and suboptimal classifiers is much larger than $2^{-\alpha l(p(t))}$.

D. Learning the Relevant Contexts

Since both UPCE and APCE form a partition of the context space, the number of sets in the partitions they form grows exponentially with the dimension of the context space. While this curse of dimensionality is unavoidable for an arbitrary (worst-case) context arrival process, it is possible to achieve much faster convergence when the dataset under consideration has additional structure such that a low dimensional representation is possible. In our recent work [52], we solved the curse of dimensionality problem for datasets in which the true label depends only on a subset of (unknown) *relevant* contexts. The method we developed solves a slightly different problem which falls under the class of *sequential decision making under uncertainty problems*, but can easily be adapted to work with UPCE and APCE.

VIII. Conclusion

In this paper we proposed online active image stream mining algorithms that use the context information that comes along with an image to guide its classifier selection. Our learning methods estimate classifier accuracies for similar contexts using the past labels for images with similar contexts. We prove sublinear regret bounds for our algorithms under mild assumptions on the images, classifiers and contexts. While our illustrative image stream mining application in this paper is a medical one, our framework can also be applied to other image stream mining problems such as video surveillance and video traffic monitoring.

Appendix A
Notation

Unless noted otherwise, sets are denoted by calligraphic letters. $\mathrm{P}(\cdot)$ is the probability operator. $\mathrm{E}_F(\cdot)$ is the expectation operator with respect to distribution F. The subscript is dropped whenever the distribution is clear from the context. $\mathrm{I}(\omega)$ is the indicator function which is equal to 1 if event ω happens and 0 else. For a set $\mathcal{A}$, $|\mathcal{A}|$ denotes its cardinality. For a real number r, $\lceil r \rceil$ denotes the smallest integer that is greater than or equal to r. Standard Euclidian norm is denoted by $||\cdot||$. Index f denotes a classifier. $O(\cdot)$ is the standard Big O notation and $\tilde{O}(\cdot)$ also hides terms that grow logarithmically.

References

[1] J. Zhang, W. Hsu, and M. L. Lee, "Image mining: Issues, frameworks and techniques," in *Proc. 2nd ACM SIGKDD Int. Workshop Multimedia Data Mining*, 2001, pp. 13–20.

[2] J. P. Borgstede, R. S. Lewis, M. Bhargavan, and J. H. Sunshine, "RADPEER quality assurance program: A multifacility study of interpretive disagreement rates," *J. Amer. College Radiol.*, vol. 1, no. 1, pp. 59–65, 2004.

[3] C. D. Johnson, K. N. Krecke, R. Miranda, C. C. Roberts, and C. Denham, "Developing a radiology quality and safety program: A primer," *Radiographics*, vol. 29, no. 4, pp. 951–959, 2009.

[4] 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.

[5] U. Adiga, R. Malladi, R. Fernandez-Gonzalez, and C. O. de Solorzano, "High-throughput analysis of multispectral images of breast cancer tissue," *IEEE Trans. Image Process.*, vol. 15, no. 8, pp. 2259–2268, Aug. 2006.

[6] F. Schnorrenberg, C. S. Pattichis, K. C. Kyriacou, and C. N. Schizas, "Computer-aided detection of breast cancer nuclei," *IEEE Trans. Inf. Technol. Biomed.*, vol. 1, no. 2, pp. 128–140, Jun. 1997.

[7] S. Ciatto *et al.*, "Benign breast surgical biopsies: Are they always justified?" *Tumori*, vol. 84, no. 5, pp. 521–524, 1997.

[8] B. Settles, "Active learning literature survey," Dept. Comput. Sci., Univ. Wisconsin–Madison, Madison, WI, USA, Tech. Rep. 1648, Jan. 2010.

[9] Y. Baram, R. El-Yaniv, and K. Luz, "Online choice of active learning algorithms," *J. Mach. Learn. Res.*, vol. 5, pp. 255–291, Dec. 2004.

[10] R. Gilad-Bachrach, A. Navot, and N. Tishby, "Query by committee made real," in *Proc. NIPS*, vol. 5. 2005, pp. 443–450.

[11] S. Dasgupta, A. T. Kalai, and C. Monteleoni, "Analysis of perceptron-based active learning," in *Learning Theory*. Berlin, Germany: Springer-Verlag, 2005, pp. 249–263.

[12] A. I. Schein and L. H. Ungar, "Active learning for logistic regression: An evaluation," *Mach. Learn.*, vol. 68, no. 3, pp. 235–265, 2007.

[13] D. Angluin, "Queries and concept learning," *Mach. Learn.*, vol. 2, no. 4, pp. 319–342, 1988.

[14] K. Bache and M. Lichman. (2013). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[15] H. A. Abbass, "An evolutionary artificial neural networks approach for breast cancer diagnosis," *Artif. Intell. Med.*, vol. 25, no. 3, pp. 265–281, 2002.

[16] A. Vailaya, M. A. T. Figueiredo, A. K. Jain, and H.-J. Zhang, "Image classification for content-based indexing," *IEEE Trans. Image Process.*, vol. 10, no. 1, pp. 117–130, Jan. 2001.

[17] J. Z. Wang, G. Wiederhold, and O. Firschein, "System for screening objectionable images using Daubechies' wavelets and color histograms," in *Interactive Distributed Multimedia Systems and Telecommunication Services*. Berlin, Germany: Springer-Verlag, 1997, pp. 20–30.

[18] O. R. Zaïane, J. Han, Z.-N. Li, and J. Hou, "Mining multimedia data," in *Proc. Conf. Centre Adv. Studies Collaborative Res.*, 1998, p. 24.

[19] N. Bouguila and D. Ziou, "A hybrid SEM algorithm for high-dimensional unsupervised learning using a finite generalized Dirichlet mixture," *IEEE Trans. Image Process.*, vol. 15, no. 9, pp. 2657–2668, Sep. 2006.

[20] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.

[21] V. Megalooikonomou, C. Davatzikos, and E. H. Herskovits, "Mining lesion-deficit associations in a brain image database," in *Proc. 5th KDD*, 1999, pp. 347–351.

[22] S. S. Haykin, *Neural Networks: A Comprehensive Foundation*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1994.

[23] R. Kohavi, "Scaling up the accuracy of Naive–Bayes classifiers: A decision-tree hybrid," in *Proc. KDD*, 1996, pp. 202–207.

[24] G. G. Gardner, D. Keating, T. H. Williamson, and A. T. Elliott, "Automatic detection of diabetic retinopathy using an artificial neural network: A screening tool," *Brit. J. Ophthalmol.*, vol. 80, no. 11, pp. 940–944, 1996.

[25] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby, "Selective sampling using the query by committee algorithm," *Mach. Learn.*, vol. 28, nos. 2–3, pp. 133–168, 1997.

[26] N. Cesa-Bianchi, G. Lugosi, and G. Stoltz, "Minimizing regret with label efficient prediction," *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 2152–2162, Jun. 2005.

[27] O. Dekel, C. Gentile, and K. Sridharan, "Selective sampling and active learning from single and multiple teachers," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 2655–2697, 2012.

[28] N. Zolghadr, G. Bartók, R. Greiner, A. György, and C. Szepesvári, "Online learning with costly features and labels," in *Proc. NIPS*, 2013, pp. 1241–1249.

[29] M. Sewell, "Ensemble learning," Dept. Comput. Sci., UCL, London, U.K., Tech. Rep. RN/11/02, 2011.

[30] E. Alpaydin, *Introduction to Machine Learning*. Cambridge, MA, USA: MIT Press, 2004.

[31] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[32] D. H. Wolpert, "Stacked generalization," *Neural Netw.*, vol. 5, no. 2, pp. 241–259, 1992.

[33] P. Bühlmann and B. Yu, "Boosting with the L_2 loss: Regression and classification," *J. Amer. Statist. Assoc.*, vol. 98, no. 462, pp. 324–339, 2003.

[34] A. Lazarevic and Z. Obradovic, "The distributed boosting algorithm," in *Proc. 7th KDD*, 2001, pp. 311–316.

[35] C. Perlich and G. Świrszcz, "On cross-validation and stacking: Building seemingly predictive models on random data," *ACM SIGKDD Explorations Newslett.*, vol. 12, no. 2, pp. 11–15, 2011.

[36] J. Gao, W. Fan, and J. Han, "On appropriate assumptions to mine data streams: Analysis and practice," in *Proc. 7th IEEE ICDM*, Oct. 2007, pp. 143–152.

[37] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *Computational Learning Theory*. Berlin, Germany: Springer-Verlag, 1995, pp. 23–37.

[38] W. Fan, S. J. Stolfo, and J. Zhang, "The application of AdaBoost for distributed, scalable and on-line learning," in *Proc. 5th KDD*, 1999, pp. 362–366.

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

[40] 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.

[41] 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.

[42] E. Hazan and N. Megiddo, "Online learning with prior knowledge," in *Learning Theory*. Berlin, Germany: Springer-Verlag, 2007, pp. 499–513.

[43] A. Slivkins, "Contextual bandits with similarity information," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 2533–2568, Jan. 2014.

[44] M. Dudik *et al.* (2011). "Efficient optimal learning for contextual bandits." [Online]. Available: http://arxiv.org/abs/1106.2369

[45] J. Langford and T. Zhang, "The Epoch–Greedy algorithm for contextual multi-armed bandits," in *Proc. NIPS*, 2007, pp. 1096–1103.

[46] W. Chu, L. Li, L. Reyzin, and R. E. Schapire, "Contextual bandits with linear payoff functions," in *Proc. AISTATS*, 2011, pp. 208–214.

[47] T. Lu, D. Pál, and M. Pál, "Contextual multi-armed bandits," in *Proc. AISTATS*, 2010, pp. 485–492.

[48] R. Lienhart, L. Liang, and A. Kuranov, "A detector tree of boosted classifiers for real-time object detection and tracking," in *Proc. ICME*, vol. 2. Jul. 2003, pp. II-277–II-280.

[49] Y. Mao, X. Zhou, D. Pi, Y. Sun, and S. T. Wong, "Multiclass cancer classification by using fuzzy support vector machine and binary decision tree with gene selection," *J. Biomed. Biotechnol.*, vol. 30, no. 2, pp. 160–171, 2005.

[50] C. Tekin and M. van der Schaar. (2015). *Online Appendix for: Active Learning in Context-Driven Stream Mining With an Application to Image Mining*. [Online]. Available: http://medianetlab.ee.ucla.edu/papers/TIPAppendixTekinvanderSchaar.pdf

[51] S. Bubeck, R. Munos, G. Stoltz, and C. Szepesvari, "X-armed bandits," *J. Mach. Learn. Res.*, vol. 12, pp. 1655–1695, Feb. 2011.

[52] C. Tekin and M. van der Schaar, "Discovering, learning and exploiting relevance," in *Proc. NIPS*, Dec. 2014, pp. 1233–1241.

Cem Tekin received the B.Sc. degree in electrical and electronics engineering from Middle East Technical University, Ankara, Turkey, in 2008, and 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, in 2010, 2011, and 2013, respectively. He was a Post-Doctoral Scholar with the University of California at Los Angeles, from 2013 to 2015. He is currently an Assistant Professor with the Electrical and Electronics Engineering Department, Bilkent University, Turkey. His research interests include machine learning, multiarmed bandit problems, data mining, multiagent systems, and game theory. 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.

Mihaela van der Schaar (F'10) is currently a Chancellor Professor of Electrical Engineering with the University of California at Los Angeles. She holds 33 granted U.S. patents. Her research interests include network economics and game theory, online learning, dynamic multiuser networking and communication, multimedia processing and systems, and real-time stream mining. She was a Distinguished Lecturer of the Communications Society from 2011 to 2012, the Editor-in-Chief of the 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 the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (2005), the Okawa Foundation Award (2006), the IBM Faculty Award (2005, 2007, and 2008), the Most Cited Paper Award from the *EURASIP: Image Communications Journal* (2006), the Gamenets Conference Best Paper Award (2011), and the IEEE Circuits and Systems Society Darlington Award Best Paper Award (2011). She received three ISO awards for her contributions to the MPEG video compression and streaming international standardization activities.