Personalized Active Learning for Activity Classification in Wireless Health

Jie Xu, Student Member, IEEE, Linqi Song, Student Member, IEEE, Mihaela van der Schaar, Fellow, IEEE, James Y. Xu, Student Member, IEEE, and Gregory J. Pottie, Fellow, IEEE

Abstract

Abstract—Enabling accurate and low-cost classification of a range of motion activities is important for a variety of applications, ranging from treatment to rehabilitation to training. This paper proposes a novel contextual online learning method for activity classification based on data captured by low-cost, body-worn inertial sensors and smartphones. The proposed method is able to address the unique challenges arising in online monitoring and activity classification with requirements for personalization and online adaptation needing to be fulfilled without a training phase. Another key challenge of activity classification is that the labels may change over time, as the data as well as the activity to be monitored evolve continuously, and the true label is often costly and difficult to obtain. The proposed algorithm is able to actively learn when to ask for the true label by assessing the benefits and costs of obtaining them. We rigorously characterize the performance of the proposed learning algorithm and prove that the learning regret (i.e., performance loss due to learning as compared to an omniscient oracle) is sublinear in time, thereby ensuring fast convergence to the optimal reward as well as providing short-term performance guarantees. Our experiments show that the proposed algorithm outperforms existing algorithms (e.g., online weighted majority and online AdaBoost) in terms of both providing higher classification accuracy as well as lower energy consumption.

Index Terms

Wireless health, activity classification, activity monitoring, context-aware, online learning, active learning, multi-armed bandits

The authors are with the Department of Electrical Engineering, University of California, Los Angeles.
I. INTRODUCTION

One of the biggest problems faced by many nations this century is the staggering cost of health care and the ever growing number of people suffering from disabilities requiring treatment and continued rehabilitation. In just the United States, stroke alone produces around 650,000 survivors each year, most of whom require physical rehabilitation long after discharge from a hospital [1]. Across all diseases, roughly 40% of the elderly population experience one or more forms of disabilities requiring rehabilitation and this number grows at a significant rate [2]. The advent of microelectronics and powerful mobile devices has created a number of new opportunities in wireless health that allow us to address this serious challenge. In particular, inexpensive and pervasive remote activity monitoring through body worn inertial sensors and smartphones represents a low-cost solution to provide monitoring and ensure compliance to physical exercises prescribed to patients, regardless of their ability to directly access health care professionals [3][4][5][6]. Using activity monitoring, physicians can prescribe a set of recommended daily exercises to the patients, monitor them remotely and then provide feedback without a clinic revisit.

While much work done so far has focused on the accurate detection of physical activities [7][8][9][5][10][11], many challenges remain that prevent the technique from being widely adopted: 1) Domain experts such as clinicians come from diverse backgrounds with unique sets of activities of interest. As the number of potential types of activities to be monitored increase, traditional classifiers suffer from degraded performance and reliability; 2) Most activity classifiers’ performance directly relates to the amount of training data the intended individual is able to provide. Our in-field experience in conducting large scale international trials [7] has demonstrated that this data is very difficult, if not impossible to obtain due to lack of training from healthcare professionals and patient’s inability to physically perform the activities for the required training time; 3) State of the art activity classifiers are often deployed without the means to retrain and adapt to a user’s improving situation, significantly reducing their usefulness after a few months.

In this paper, we present a novel contextual multi-armed bandits (MAB) approach that enables efficient, personalized activity classification. This method is able to address the challenges highlighted above by melding context adaptation, continuous online learning and active learning.
First, we use the context information about the physical activities, such as location where the activity takes place, the user profile including his/her age, gender weight etc. to assist and personalize the activity classification. Since the context space can be also very large, our approach adaptively decides when to and how to partition the context space into smaller subspaces to enable faster learning and achieve more accurate classification. Second, the proposed classification algorithm is developed based on a continuous online learning method, requiring no a priori training data of the individual users. The use of continuous online learning allows the system to start with classifiers that are trained using generic data sets (easy to obtain) and gradually adapt to individual users. In this way, the classifiers can also adapt themselves to a user’s improving situation by selecting more suitable models and boot-strapping the model with newer data. Thirdly, the proposed approach enables active learning in activity classification: it actively and judiciously acquires the costly “ground-truth” labels of user activities by assessing the benefits and costs of obtaining them. The capability of active learning effectively reduces the labeling cost, which can be significant in a large-scale monitoring system with many users.

Last but not the least, we can provide performance guarantees for our proposed activity classification method. As a performance measure for our online activity classification approach, we use the regret, which is defined as the difference between the expected total reward (classification accuracy minus classification cost including sensing, wireless transmission, labeling cost) of the best classification scheme given complete knowledge about the accuracies of the available classifier for all possible contexts and the expected total reward of the proposed algorithm. We then show that our proposed classification algorithm achieves regret sublinear in the number of classification requests made 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 and the rate of learning is fast, which is vital for the deployment of the activity monitoring system.

The rest of this paper is organized as follows. Section II discusses related works and highlights the differences of this paper from them. Section III describes the system model and formulates the context-aware activity classification problem. Section IV proposes the online learning algorithm for activity classification and bounds its learning regret when there is no labeling cost. Section V proposes the active learning version of the proposed algorithm when labeling is costly. Section VI provides simulation results using real-world data. Section VII concludes the paper.
II. RELATED WORKS

A. Activity monitoring and classification

The benefits of activity monitoring through sensors have been demonstrated in many existing works [12][8][9][5][10][11][13][14][15][16]. For instance, one system for measuring home-based physical rehabilitation has been described in [8]. Using a signature detection algorithm and accelerometer’s signal vector magnitude as a feature, the system detects if a user has performed a set of rehabilitation exercises accurately, and provides appropriate feedback. In [9], a human activity classification system was developed for promoting exercises in an effort to reduce injuries. The system uses multiple on body accelerometers, and a large number of binary classifiers each trained to recognize specific activities. A series of optimizations link the individual classifiers to produce a final output. In [10], a small set of activities are recognized using a single triaxial accelerometer and machine learning techniques. Hierarchical classification is performed by preliminary clustering of motion into static, transitional and dynamic states, followed by refined classification of actual activities. Most methods confront the challenge of classifying a specific motion among many possibilities at any observation time [11]. As the number of potential motions increase, the classifier model complexity increases and classification performance and reliability are degraded. In addition, these systems do not address the issue of rapid adaptation to the demands of large heterogeneous user communities, where different activities are of interest, requiring separate models, classification methods and features. The focus of this paper is not on constructing classifiers but rather selecting the best classifier among a candidate classifier set in different context scenarios. We develop online algorithms for learning the best activity classifiers for specific contexts by exploiting the available contextual information.

B. Ensemble learning

In our model, the system has access to many classifiers but activates only one of them to perform the activity classification task for each request based on the context. When there are many classifiers, another approach is to use an ensemble of classifiers to produce a final result based on the results of the classifiers in the ensemble. There are many works that consider ensemble learning [17][18][19][20][21], using techniques such as bagging, boosting, stacked generalization and cascading. However, most of them provide 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. In other words, we prove regret bounds that hold uniformly over time.

Some other ensemble learning methods assign a weight to the prediction of each classifier and produce a final prediction based on the weighted sum of the prediction of individual classifiers \[22\][23][24][25][26]. These weights can be updated online [24] based on the past performance of the classifiers. In contrast, we focus on how contextual specialization of classifiers can be discovered over time to create a strong classifier from many weak classifiers. In addition, our method incurs less energy cost due to sensing since only one classifier is activated for each request while the ensemble learning techniques require all classifiers to be activated.

C. Multi-armed bandits and Active learning

The proposed algorithm is developed under the contextual multi-armed bandits (MAB) framework. Previously, MAB methods were applied to solve problems in clinical trials [27][28], multi-user communication networks [29], web advertising [30], recommender systems [31][32] and stream mining systems [33][34]. A key advantage of MAB methods as compared to other online learning methods is that they can provide a bound on the convergence speed as well as a bound on the loss due to learning compared to an oracle solution which requires knowledge of the stochastic model of the system, which is named regret. To the authors’ best knowledge, it is the first time that MAB methods are applied to solve activity classification problems in wireless health systems.

In the existing works of contextual multi-armed bandits works, the learner observes the reward of its selected action (classifier) every time it selects that action. In contrast, in the current paper the “ground-truth” activity label (hence the classification reward) may not always be available. The reward can only be actively observed, where there is a cost associated with acquiring the true label. This paper enables active learning in the MAB framework by carefully choosing when to acquire the “ground-truth” label. Several active learning methods were proposed in [35][36][37][38]. For example, in [38], the authors propose a pool based active learning method in which the data to be classified is actively selected from a database. In contrast, we do not have control over the data arrival process, and we do not need to store it in a database. Our
algorithm needs to learn when to perform active learning online, instead of which data instance to select from the database to perform active learning.

III. System Model

A. Activity Monitoring: System Overview

Both standalone and client-server architectures can be adopted for the activity monitoring system. In the standalone architecture, body-worn inertial sensors send raw data to the user’s smart mobile device (e.g. smart phones and tablets) through bluetooth transmission. The smart mobile device processes the raw data locally to obtain the activity classification results which can be consumed by third party applications. In the client-server architecture (see Figure 1), users’ smart mobile device sends the data to a remote server for processing and activity classification. The returned results can also be consumed by third party applications. Our proposed activity classification method can be applied to both architectures. For illustration purposes, we will focus on the client-server architecture which was designed in prior work [7].

In the system developed in [7], individual subsystems are modeled as objects, and the entire architecture is defined by a set of interfaces and relationships. Each subsystem can be developed independently without revealing specific implementations, so long as it implements the required interface. This enables any part of the proposed system to be overridden by custom realizations, allowing for rapid prototyping and evaluation of various algorithms. This architecture naturally conforms to the client-server paradigm. There are two clients: the end-user client allows a user to authenticate, collect required data for the context and activity classification systems; the domain expert client is used by experts to design and prescribe scenarios for end-users. Sever components include prescription and scenario management, context classification, context driven activity classification and sensor control. In this paper, we focus on the activity classification module (ACM) which takes the results outputted by the context classification module (CCM) as the context information.

B. Activity Classification Module

We consider an infinite-horizon discrete time system where time is divided into discrete slots $t = 1, 2, \ldots$. At the beginning of each time slot $t$, the ACM on the server receives one activity classification request from the one of the end-users. Note the discrete slots do not represent
Fig. 1. System architecture: Client-Server.

absolute time but rather the sequence of requests. The CCM first performs context classification and provides the context information to the ACM. The context can include information about the location in which unclassified activity takes place as well as information about the user profile such as age, gender and weight etc. Note that contexts could also be features extracted from the data rather than meta-data/side information. Extracting specific features is more costly, so we will use only the side information as the context in this paper. This is not a limitation of this paper since features can also be used as contexts. We abstract the context information at time \( t \) using the notation \( \theta^t \in \Theta \) with \( \Theta \) being a \( d \)-dimensional metric space.

The ACM maintains a finite set of \( K \) activity classifiers \( \mathcal{F} = \{ f_1, \ldots, f_K \} \). Each classifier \( f_k \) takes input using the activity sensing data \( x^t \) from a specific set of sensors with respect to \( f_k \) and outputs a classification result. That is, \( f_k \) is a function \( f_k : \mathcal{X}_{f_k} \to \mathcal{Y} \) where \( \mathcal{X}_{f_k} \) is the sensing data value space with respect to \( f_k \)-specific sensor set and \( \mathcal{Y} \) is the activity value space. Because different classifiers require the input data from different sets of sensors, invoking them incurs different costs due to sensing and wireless transmission energy consumption.

At each time slot \( t \), given the context information \( \theta^t \), the ACM selects one classifier from the classifier set, denoted by \( f^t \in \mathcal{S}' \), to perform the activity classification task. Depending on the classifier selected, a specific set of sensors is invoked via the sensor selection module, incurring a cost (i.e. sensing and transmission cost) of \( c^t \) which is reported to the ACM at the end of the time slot. Once the sensor data \( x^t \in \mathcal{X}_{S'} \) is collected and provided to the ACM, the ACM extracts
activity features from the raw sensing data and invokes the selected classifier $f^t$ to classify the activity. The classifier $f^t$ then outputs a classification result, denoted by $y^t \in \mathcal{Y}$. We assume for now that at the end of each time slot $t$, the ground-truth label of the activity $\hat{y}^t \in \mathcal{Y}$ is revealed without incurring any additional labeling cost. This label may be provided by the end-user himself/herself or by the physicians. Later in Section V, we will relax this assumption and study when to actively request the ground-truth label which is costly to obtain. Let $a^t = I(y^t = \hat{y}^t)$ be the classification accuracy where $I(\cdot)$ is the indicator function. In sum, the problem under consideration features a supervised online learning problem: after selecting a classifier $f_t$, a realized classification reward $r^t$ which jointly takes into account the classification accuracy $a^t$ and the associated classification cost $c^t$ is revealed to the ACM. Using the realized random rewards $r^t$, the ACM learns over time the best classifier to use for each context. For example, the classification reward can be a linear combination of accuracy and cost, i.e. $r^t = a^t - \beta c^t$ where $\beta$ is a trade-off parameter.

C. Classification Reward and Learning Regret

For each context $\theta \in \Theta$, selecting a classifier $f$ yields an (unknown) expected classification reward $\mu_\theta(f) = \mathbb{E}r_\theta(f)$. The expected rewards of different classifiers can be different for various reasons. For example, suppose we use location as the context, a simple classifier activates a small number of sensors could yield a already high classification accuracy when the location is gym while the same classifier performs poorly for an outdoor environment. Thus a more advanced classifier is required to achieve a sufficiently high accuracy but more sensing and transmission costs will be incurred due to a larger number of sensors activated and a larger amount of raw data transmitted from the sensors to the mobile device.

Notice that for each specific activity classification request with context information $\theta$, the realized reward $r_\theta(f)$ by selecting the classifier $f$ is a random variable drawn from an unknown distribution with mean $\mu_\theta(f)$, which is also initially unknown. However, a good estimate of the expected reward can be formed if many realizations have been observed. Let $f^*(\theta) := \arg \max_{f \in \mathcal{F}} \mu_\theta(f)$ and $\mu^*_\theta := \mu_\theta(f^*(\theta))$ denote the classifier that yields the highest expected reward given context $\theta$. We call $f^*(\theta)$ the best oracle classifier for context $\theta$. Notice that the best oracle classifiers are not known by the ACM but instead need to be learned.

A learning algorithm $\sigma$ selects for the context information $\theta^t$ in each time slot $t$ a classifier
time $t \in \mathcal{F}$ to perform the classification task. The regret (learning loss) of a learning algorithm used by the ACM with respect to the oracle benchmark by time $T$ is given by

$$\text{Reg}(T) = \sum_{t=1}^{T} \mu_{\theta_t}^* - E \left[ \sum_{t=1}^{T} r_{\theta_t}(\sigma_t) \right]$$

(1)

Regret gives the convergence rate of the total expected classification reward of the learning algorithm to the value of the optimal solution. Any algorithm whose regret is sublinear, i.e. $\text{Reg}(T) = O(T^\gamma)$ such that $\gamma < 1$, will converge to the optimal solution in terms of the average reward. The goal of the ACM is to minimize its regret, which is equivalent to maximizing the total reward by time $T$.

In the next section, we will propose an efficient learning algorithm that learns the optimal classifiers with sublinear regret bounds. To enable rigorous regret analysis, we assume that if context information is similar, then the expected reward obtained by selecting the same classifier is also similar. This is a widely adopted technical assumption [33][30], which can be formalized as follows:

**Assumption.** (Lipschitz) for each $f \in \mathcal{F}$, there exists $L > 0, \alpha > 0$ such that for all $\theta, \theta' \in \Theta$, we have $|\mu_{\theta}(f) - \mu_{\theta'}(f)| \leq L\|\theta - \theta'\|_\Theta^\alpha$.

**IV. PERSONALIZED ACTIVITY CLASSIFICATION**

The basic idea of our online learning algorithm is as follows. The algorithm alternates between two phases over time. In the exploration phases, different classifiers are selected to learn their expected classification reward. In the exploitation phases, the classifier with the best estimated classification reward is selected in order to maximize the classification rewards. Note that the exploration and exploitation phases are interleaved unlike in the conventional learning approaches where only a single training phase is executed followed by the exploitation phase. On the one hand, this learning problem would be simple if there was no context information. But without using the context information the performance of the learning algorithm can be poor because the best oracle classifiers can be very different for different context information. On the other hand, the context space $\Theta$ can be very large and even continuous. Thus learning the best oracle classifier for each individual context $\theta \in \Theta$ is extremely difficult, if not impossible, for a finite number $T$ of activity classification requests. To overcome this obstacle, our learning algorithm will exploit
the similarity information of contexts, adaptively and dynamically partition the context space into smaller subspaces and learn the best oracle classifier within each subspace.

A. Algorithm Description

In this subsection, we describe the proposed online learning algorithm for activity classification. For analysis simplicity, we normalize the context space to be $\Theta = [0, 1]^d$. For example, if the context information includes only the user age, then $d = 1$ and the context space will be normalized with respect to the maximum age and the minimum age. If the context information includes both the user age and weight, then $d = 2$ and the context space will be normalized with respect to the maximum/minimum age and weight. The key idea of the proposed algorithm is to adaptively partition the context space and learn the performance of classifiers in each subspace using sample mean techniques. The purpose of context space partitioning is to allow the system to learn the performance of different classifiers for a subspace of contexts rather than each individual context, thereby increasing the learning speed. Take the age context as an example, instead of learning the best classifier for age 25 directly, the algorithm first learns the best classifier for a larger range of ages, e.g. ages 10 - 40, initially and then based on the learned results, gradually partition this range into smaller ranges that contain age 25. In order to improve the learning for the performance of classifiers for each individual contexts, the partition will be dynamically updated according to the context arrival process.

The following notions are important for the proposed algorithm.

- **Context Space Partition.** By uniformly partitioning the context space on each dimension by $l - 1$, we create $2^{(l-1)d}$ context subspaces, each of which is a $d$-dimensional hypercube with side length being $2^{-(l-1)}$. We call this partition a level $l$ partition $\mathcal{P}_l$ and clearly $|\mathcal{P}_l| = 2^{(l-1)d}$. Each subspace in the partition $\mathcal{P}_l$ is called a level-$l$ subspace. Note that $\mathcal{P}_1$ contains only a single hypercube which is the entire context space $\Theta$. Let $\mathcal{P} := \bigcup_{l=1}^{\infty} \mathcal{P}_l$ denote the set of all possible such subspaces.

- **Active Context Subspace.** In each time slot $t$, the algorithm keeps a set of mutually exclusive context subspaces that cover the entire context space. We call these subspaces the active subspaces, and denote the set of active subspaces at time $t$ by $\mathcal{A}^t$. Clearly we have $\bigcup_{C \in \mathcal{A}^t} = \Theta, \forall t$. 
• **Activation, Partitioning and Deactivation.** Once a subspace \( C \in \mathcal{P} \) is activated, we maintain a counter \( N_C \) that records the number of times that context arrives to \( C \). A level \( l \) subspace \( C \) will stay active until the first time \( t \) such that \( N_C \geq A2^{pl} \) where \( p > 0 \) and \( A > 0 \) are algorithm design parameters. At this point, the level \( l \) subspace \( C \) is further partitioned into \( 2^d \) smaller \( l + 1 \) subspaces that constitutes \( C \). Then \( C \) becomes inactivate and these smaller subspaces become active and \( 2^d \) new counters are created.

• **Reward Estimates.** For each active context subspace \( C \), the algorithm maintains the estimated rewards \( \bar{r}_C(f) \) for all classifiers for the context arrivals to this subspace.

• **Counters.** For each active context subspace \( C \), the algorithm maintains several counters. The first counter \( M_C \) records the number of context arrivals to \( C \) which is used for context subspace partitioning. For each subspace \( C \), the algorithm also maintains for each classifier \( f \) a counter \( N_C(f) \) that records the number of times \( f \) is selected to classify the request. Clearly, \( M_C = \sum_{f \in \mathcal{F}} N_C(f) \) at any moment in time.

• **Control Function.** The algorithm uses a control function \( D(t) \) which takes time as the input and outputs a real number. The control function has the form of \( D(t) = t^z \ln t \) where \( z \) is the design parameter.

The algorithm works as follows. When an activity classification request with context information \( \theta^t \) comes at time \( t \), the algorithm checks to which active subspace \( C \in \mathcal{A}^t \) it belongs. Then it investigates counters \( N_C(f) \) for all classifiers to see if there exists any under-explored classifier \( f \) such that \( N_C(f) \leq D(t) \). There are two cases:

- If there exists such an under-explored classifier \( f \), then the algorithm selects this classifier for the current request, i.e. \( \sigma^t = f \). In this way, we obtain one more reward realization by selecting \( f \) and hence the reward estimate of \( f \) can be more accurate. We call such a time slot an **Exploration** slot.

- If there does not exist any under-explored classifier, then the algorithm selects the classifier with the highest reward estimate \( \sigma^t = \arg \max_{f \in \mathcal{F}} \bar{r}_C(f) \). In this way, we exploit the best classifier that has been learned so far. We call such a time slot an **Exploitation** slot.

At the end of time \( t \), the true label and the classification cost is revealed and hence, the reward estimate \( \bar{r}_C(f), \forall f \in \mathcal{S}^t \) of the activated classifiers can be updated. If the context arrival counter for this context subspace \( M_C \) exceeds \( A2^{pl} \) where \( A \) and \( p \) are the design parameters and \( l \) is
the level of the current subspace, then the context subspace is further partitioned. The formal description is presented in Algorithm 1. A pictorial illustration of the adaptive partition process is provided in Figure 2.

**Algorithm 1** Online Learning for Activity Classification

Initialize $A_1 = P_1$, $M_C = 0$, $\bar{r}_\theta(f) = 0, \forall f \in \mathcal{F}$.

for each activity classification request at time $t$ do

Determine $C \in A^t$ such that $\theta \in C$

Case 1: $\exists f \in \mathcal{F}$ such that $N_C(f) < D(t)$ (Exploration)

Select $\sigma^t = f$

Set $N_C(\sigma^t) \leftarrow N_C(\sigma^t) + 1$

(The activity classification reward $r^t$ is revealed.)

Update $\bar{r}_C(\sigma^t)$

Case 2: $\forall f \in \mathcal{F}$, $N_C(f) \geq D(t)$ (Exploitation)

Select $\sigma^t = \arg \max_{f \in \mathcal{F}} \bar{r}_C(f)$.

Set $N_C(\sigma^t) \leftarrow N_C(f^t) + 1$

(The activity classification reward $r^t$ is revealed.)

Update $\bar{r}_C(\sigma^t)$

Set $M_C \leftarrow M_C + 1$

if $M_C \geq 2^\rho$ then

Set $A^{t+1} = (A^t \setminus C) \cup P_{t+1}(C)$

end if

end for

B. Learning Regret Analysis

In this subsection, we bound the regret of the proposed learning algorithm for activity classification. Keep in mind that the design parameters are $z$ (which controls when to explore and when to exploit), $A$ and $p$ (which control when to partition a subspace). The regret of the proposed algorithm depends how context space partition is formed over time which further depends on the context arrival process. We investigate the regret for different context arrival scenarios.
Definition. We call the context arrival process the **worst case arrival process** if it is uniformly distributed inside the context space, with minimum distance between any two context samples being $T^{-1/d}$, and the **best case arrival process** if $x^t \in C, \forall t$ for some level $\lceil \log_2(T)/p \rceil + 1$ subspace $C$.

To have a better idea of the different arrival processes, suppose that the algorithm only uses the user age as the context information. Therefore the context space $\Theta$ is one-dimensional. In the worst case arrival process, requests with different age contexts arrive uniformly so that the context space will be partitioned uniformly as well. This creates the largest number of context subspaces and hence, learning is slow. In the best case arrival process, the context information of requests are concentrated to a small range of ages. Thus, the context partition will be refined in that range very fast, leading to a fast learning speed. Figure 3 provides an illustrative example in which we set the partitioning parameter $A = 1$ and $p = 1$. In the worst case arrival, the first partitioning occurs after two request arrivals (e.g. age 11, 52). The second partitioning is performed on the subspace $[10, 50)$ and the third partitioned is performed on the subspace $[50, 90)$, creating two level-2 context subspaces. However, in the best case arrival, the third partition is performed on $[10, 30)$, creating two level-3 context subspaces.

The following theorem determines the finite time, uniform regret bounds for the online activity classification algorithm.
Theorem 1. For the worst case arrival process, $\text{Reg}(T) = O(T^{d+\alpha})$ by choosing $p = \frac{3\alpha + \sqrt{9\alpha^2 + 8\alpha d}}{2}$ and $z = 2\alpha/p$. For the best case arrival process, $\text{Reg}(T) = O(T^{2/3})$ by choosing $p = 3\alpha$ and $z = 2\alpha/p$.

Proof: See Appendix A.

The regret bounds proved in Theorem 1 are sublinear in time $T$ which guarantee convergence in terms of the average classification rewards, i.e. $\lim_{T \to \infty} \text{Reg}(T)/T = 0$. Thus our online learning algorithm makes the optimal classification as sufficient classification requests have arrived. More importantly, the regret bound tells how much reward would have been lost by running our learning algorithm for any finite time $T$. Hence, it also provides a rigorous characterization on the short-term learning performance.

The regret bound for the worst case arrival process depends on both the context space dimension $d$ and the similarity metric $\alpha$ in the Lipschitz condition (Assumption 1). The regret bound is worse if the dimension $d$ is larger. When we have many context dimensions $d \to \infty$, the regret bound tends to be linear in $T$. This is intuitive since we need to learn the performance of the classifiers in infinitely many subspaces. The regret bound is better if the similarity is larger (i.e. larger $\alpha$). As $\alpha \to \infty$, the performance of a classifier is almost the same for all contexts in the entire context space, the regret bound approaches the same for the best case arrival process.

For a more specific example, if we use only the user age and weight as the context information, then $d = 2$. Suppose the performance difference of a classifier is at most linear in the difference of context distance, then $\alpha = 1$. In this case, by choosing the algorithm parameters $p = 4$ and $z = 1/2$, we can achieve a regret bound $\text{Reg}(T) = O(T^{6/7})$ for the worst case arrival process.
and, by choosing the algorithm parameters $p = 3$ and $z = 2/3$, we can achieve a regret bound $Reg(T) = (T^{2/3})$ for the best case arrival process.

V. PERSONALIZED ACTIVITY CLASSIFICATION WITH ACTIVE LEARNING

In the previous section, we considered a model in which the ground-truth label of activity $y^t$ is revealed at the end of each time slot $t$ without incurring any additional cost and developed an algorithm that achieves sublinear regret. However, in practice, individual users may be reluctant to report their true activities and labeling the activity may require healthcare professionals to perform. Moreover, more wireless resources and power will be consumed for the additional information exchange between user clients and servers about the activity labelling. As such, observing the true label is often costly and thus, labels should be judiciously acquired by assessing the benefits and costs of obtaining them. In this section, we propose a modified learning algorithm that proactively acquires labels based on a benefit-cost analysis.

A. Active Learning Cost and Learning Regret

Similar to the previous model, each time $t$ when an activity classification request arrives, the learning algorithm $\sigma$ selects for the associated context information $\theta^t$ a classifier $\sigma^t \in \mathcal{F}$ to perform the classification task. The difference in the active learning scenario is that the algorithm also determines an active learning action which decides whether to acquire the ground-truth label $y^t$. Let $\tau^t \in \{0, 1\}$ be the active learning strategy where $\tau^t = 1$ stands for asking for the label and $\tau^t = 0$ otherwise. If $\tau^t = 1$, then an additional active learning cost $\delta^t > 0$ is incurred. The active learning cost $\delta^t$ can be different in different slots due to, for example, different labeling methods (e.g. self-reporting or labeling by healthcare professionals) and different expertise levels of people who provide the label.

Due to the additional active learning cost, the regret of a learning algorithm $(\sigma, \tau)$ used by the ACM with respect to the oracle benchmark by time $T$ is given by

$$\text{Reg}(T) = \sum_{t=1}^{T} \mu^*_t - E \left[ \sum_{t=1}^{T} (r^t_{\hat{y}^t}(\sigma^t) - \delta^t \tau^t) \right]$$

(2)

Note that the difference of the above definition of the learning regret in equation (1) is that in (1) obtaining the ground-truth label is costly, i.e. $\delta^t = 0, \forall t$ and hence labels are acquired in each slot, i.e. $\tau^t = 1, \forall t$ while in (2) obtaining the label is costly and hence, asking for the label
becomes another decision parameter. We also point out that the oracle benchmark does not incur any labeling cost since the true classification accuracies are already known.

B. Algorithm Description

In this subsection, we describe the proposed online activity classification algorithm with active learning. Most notions described in Section IV.A will remain useful for the current algorithm. The major difference is in the notion of counters which we describe below.

- **Counters.** For each active context subspace \( C \), the algorithm maintains several counters. The first counter \( M_C \) records the number of context arrivals to \( C \) which is used for context subspace partitioning. For each subspace \( C \), the algorithm also maintains for each classifier \( f \) a counter \( N_C(f) \) that records the number of times when \( f \) is selected to classify the request and the true label is revealed.

With this new definition, the algorithm works as follows. When an activity classification request with context information \( \theta^t \) comes at time \( t \), the algorithm first checks to which active subspace \( C \in A^t \) it belongs. Then it investigates counters \( N_C(f) \) for all classifiers to see if there exists any under-explored classifier \( f \) such that \( N_C(f) \leq D(t) \). There are two cases:

- If there exists such an under-explored classifier \( f \), then the algorithm selects this classifier for the current request, i.e. \( \sigma^t = f \). At the end of this slot, the algorithm asks for the ground-truth label \( \hat{y}^t \), i.e. \( \tau^t = 1 \). In this way, we can observe one more reward realization by selecting \( f \) and hence the reward estimate of \( f \) can be more accurate. Such a time slot is called a **Exploration** slot.

- If there does not exist any under-explored classifier, then the algorithm selects the classifier with the highest reward estimate \( \sigma^t = \arg\max_{f \in \mathcal{F}} \bar{r}_C(f) \). At the end of this slot, the algorithm does not ask for the ground-truth label, i.e. \( \tau^t = 0 \). In this way, we exploit the best classifier that has been learned so far but no additional information is learned since the true label is not revealed. Such a time slot is called a **Exploitation** slot.
In this algorithm, the reward estimates $\bar{r}_C(f)$ are updated only in the first case, namely the exploration slots. If the context arrival counter for the current subspace $M_C$ exceeds $A2^{pl}$ where $A$ and $p$ are design parameters, then the context subspace is further partitioned. The formal description of the active learning algorithm is presented in Algorithm 2. Figure 4 shows the block diagram for the algorithm. Notice that the true labels are revealed only in the exploration slots.

**Algorithm 2 Online Active Learning for Activity Classification**

```
Initialize $A_1 = P_0$, $M_\theta = 0$, $\bar{r}_\theta(f) = 0, \forall f \in \mathcal{F}$.

for each activity classification request at time $t$ do
    Determine $C \in A^t$ such that $\theta \in C$

    Case 1: $\exists f \in \mathcal{F}$ such that $N_C(f) < D(t)$ (Exploration)
    Select $\sigma^t = f$
    Set $N_C(\sigma^t) \leftarrow N_C(\sigma^t) + 1$
    (The activity classification reward $r^t$ is revealed.)
    Update $\bar{r}_C(\sigma^t)$

    Case 2: $\forall f \in \mathcal{F}, N_C(f) \geq D(t)$ (Exploitation)
    Select $\sigma^t = \arg \max_{f \in \mathcal{F}} \bar{r}_C(f)$.
    Set $M_C \leftarrow M_C + 1$
    if $M_C \geq A2^{pl}$ then
        Set $A^{t+1} = (A^t \setminus C) \cup P_{t+1}(C)$
    end if
```

end for

**C. Learning Regret**

In the proposed algorithm with active learning, the ground-truth labels are acquired only in the exploration slots. Therefore the additional active learning cost is incurred only in the exploration slots. This leads to a larger regret in the exploration slots. However, since the algorithm ensures that only a sublinear number of slots are exploration slots, the order of regret due to exploration is sublinear in time. Hence, a similar result as Lemma 2 would still hold in the active learning
case by replacing $\Delta$ with $\Delta + E\delta$. Since labels are not required in the exploitation slots, the same result of Lemma 3 and Lemma 4 would hold for the case with active learning. By considering both the regret due to exploration and exploitation, we can achieve the same regret bound as the original learning algorithm determined in Theorem 1.

**Theorem 2.** The proposed online activity classification algorithm with active learning achieves the same regret bound as in Theorem 1.

The proposed active learning algorithm can also be easily extended for the scenario where the requested label is missing with small probability or is provided with (finite) delay. This could happen when the wireless communication channel fails to work or it takes longer than usual time for the healthcare professionals to label the activity if their expertise level is low. In this case, we can make a small modification to the algorithm in the exploration slots as follows: the counter $N_C(\sigma^t)$ and the reward estimate $\tilde{r}_C(\sigma^t)$ is updated only after the true reward $r^t$ is revealed. This ensures that the estimated rewards are accurate enough during the exploitation steps but prolongs the exploration phase by a constant factor. However, since the number of exploration slots by any time $\mathcal{T}$ is sublinear in $\mathcal{T}$ in the original algorithm, it is still sublinear in the modified algorithm. Therefore, the regret of the modified algorithm is still sublinear in the modified algorithm, thereby ensuring the convergence to the optimal performance.
VI. SIMULATIONS

A. Setup and Benchmarks

Our experiments are performed using real-world sensor data collected from end-users. Three major components of the system need to be deployed for data collect and system evaluation. The server responsible for user and scenario management, context and activity classification was deployed to the UCLA’s medical network servers. The domain expert client was given the collaborators at the UCLA’s Department of Neurology. The end-user component is a physical package containing four body worn inertial measurement units (IMUs) with Velcro attachments, a Nexus 7 table and the associated mobile application. A more detailed description of our system deployment can be found in [7]. Note, however, the proposed method can also be applied a standalone system which does not feature a client-server architecture.

The server continuously receives classification requests from end-users through wireless transmissions (wifi, blueooth or cellular). For each request, the CCM first detects the context information (such as location and user profile) associated with the request. Then the ACM selects a classifier to perform the classification using this context information. Due to experiment constraints, all requests are from the same location. Thus, the context information which we use in the experiments are the gender and age of the end-user. Nevertheless, the framework can be applied to any context information in general. We consider three possible activities: “Running”, “Walking Around” and “Walking Normal”. The activity “Walking Around” refers to non-sustained walking segments that are typical of walking in a confined spaces, while “Walking Normal” refers to sustained long distance walk typical of open spaces.

For the purpose of proof of concept, we implement four pre-trained generic classifiers for the candidate classifier set, each requiring different sensors to be activated and use different features extracted from the raw sensor data as the input. Table I shows the sensors required for these four classifiers. These classifiers are built using the Wireless Health Institute Sensor Fusion Toolkit (WHIST) [39][40] which is a suite of accurate classification methods for user activities that has undergone testing in diverse situations and clinical settings. It provides multimodal hierarchical classification based on a set of classifiers such as Naive Bayes and Support Vector Machine [40]. Features such as short time energy, mean, and variance are computed from the raw data, from which, hierarchical structures can be built to model the classification problem. Since different
TABLE I
SENSORS REQUIRED FOR DIFFERENT CLASSIFIERS

<table>
<thead>
<tr>
<th></th>
<th>$f_1$</th>
<th>$f_2$</th>
<th>$f_3$</th>
<th>$f_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upper right arm sensor</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lower right arm sensor</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Lower right leg sensor</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

classifiers require different numbers of sensors to be activated, they incur different sensing and wireless transmission cost. For example, classifier $f_1$ incurs a cost three times of $f_3$.

For performance evaluation, we compare the proposed algorithm against the oracle benchmark which has the complete knowledge of the classifiers’ accuracies for each context and two other widely adopted ensemble learning algorithms: weighted majority (WM) [25] and online AdaBoost [24]. We note that for the latter two benchmarks, since all classifiers are used for each request, all sensors need to be activated, thereby incurring a larger sensing and transmission cost. Moreover, both benchmarks require the true label to be obtained for all requests, thereby incurring a larger labeling cost.

B. Performance comparison without labeling cost

In this set of experiments, the reward function has the form of $r^t = a^t - \beta c^t$ where $a^t$ is the prediction accuracy, $c^t$ is the sensing and wireless transmission cost depending on the number of sensors activated for classification and $\beta$ is a tradeoff parameter. The larger $\beta$, the higher we value the classification cost.

Figures 5 and 6 show the obtained average per period reward (normalized to the optimal reward obtained by the oracle solution) over time for $\beta = 0.01, 0.1$, respectively. The proposed learning algorithms significantly outperform both benchmark solutions. Compared with WM and Adaboost, the proposed learning algorithms gain up to 15% more rewards. Since the context information used in our experiments is limited (i.e. only age and gender), the performance improvement by using the context information is moderate. However, a more significant performance improvement can be expected when more context information (e.g. location, user weight) is available. Moreover, the larger $\beta$, the higher the performance gain since our algorithm activates only one classifier in each slot, which requires much less sensing and transmission...
energy compared with WM and Adaboost. In Table II, we take a closer look at the classification results for each individual activities (\(\beta = 0.1\)). WM and AdaBoost perform moderately well for the “Running” activity but the rewards for the two similar activities, “Walking Around” and “Walking Normal”, are very low. Instead, the proposed algorithm is able to achieve high classification rewards for all three activities.

### C. Performance comparison with labeling cost

In this set of experiments, we evaluate the performance of the proposed algorithm with active learning in the presence of labeling cost. Thus, the reward function has the form of

\[
r^t = a^t - \beta c^t - \delta^t r^t.
\]

The active learning cost \(\delta^t\) is changing due to the varying level of
labeling expertise and the varying wireless channel condition for the label feedback transmission. Figure 7 and 8 show the obtained average per period reward (normalized to the optimal reward obtained by the oracle solution) over time for $\beta = 0.1$ and and expected active learning cost $E\delta = 0.1, 0.2$, respectively. Again, the proposed algorithm with active learning significantly outperforms the benchmark algorithms (about 30-40% improvement in classification reward) as well as our algorithm without active learning. When the labeling cost is higher, the performance improvement is more significant.

D. Accuracy and Energy tradeoff

Finally, we investigate the trade-off between energy consumption (i.e. classification cost) and classification accuracy. Figure 9 illustrates the accuracy and energy consumption trade-off curve of the proposed algorithm without active learning. The energy consumption is normalized to the maximum power consumption when all sensors are activated. Note that WM and AdaBoost use all sensors for all requests and hence, they are not able to make trade-off between energy consumption and accuracy. As can be seen from the figure, a higher accuracy can be obtained at a cost of higher energy consumption for all three activities as well the the overall performance.
Fig. 7. Normalized classification reward comparison for $\beta = 0.1, E\delta = 0.1$.

Fig. 8. Normalized classification reward comparison for $\beta = 0.1, E\delta = 0.2$. 


**VII. CONCLUSIONS**

In this paper, we proposed a novel online learning method for activity classification in wireless health systems using wearable inertial sensors. We have proposed a novel mobile computing system and associated algorithms and have shown that personalizing the system by incorporating the available contextual information can significantly improve the activity classification performance as compared to existing approaches. The proposed method does not require a priori training data, it self-adapts to a user’s improving situation and to diverse sets of users. It is also able to decide when to actively request the true labels of the activities, thereby reducing the labeling cost. We also have systematically proved that our online learning algorithms achieve sublinear regret, thereby providing both long-term and short-term performance guarantee for the activity classification system.

**REFERENCES**


[40] Chieh Chien and Gregory J Pottie, “A universal hybrid decision tree classifier design for human activity classification,” in
APPENDIX

PROOF OF THEOREM 1

The following notations will be useful for our analysis. Let \( \Delta = \max_{\theta \in \Theta} \{ \mu^*_\theta - \min_{f \in F} \mu_\theta(f) \} \) be the maximized reward difference between the best oracle classifier and a non-optimal classifier. Let \( \mathcal{E}_C^t(f) \) be the set of rewards collected by selecting classifier \( f \) by time \( t \) for active subspace \( C \).

For each subspace \( C \) let \( f^*(C) \) be the classifier which is optimal for the center context of that subspace, and let \( \bar{\mu}_C(f) := \sup_{\theta \in C} \mu_\theta(f) \) and \( \underline{\mu}_C(f) := \inf_{\theta \in C} \mu_\theta(f) \). For a level \( l \) subspace \( C \), we define the set of sub-optimal classifiers to be

\[
\mathcal{L}_C(B) := \{ f : \underline{\mu}_C(f^*) - \bar{\mu}_C(f) > BLd^{\alpha/2 - l\alpha} \}
\]

where \( B \) is a constant. Finally, let \( \beta_a := \sum_{t=1}^{\infty} 1/t^a \).

We decompose the regret of learning into three parts

\[
\text{Reg}(T) = \text{Reg}_e(T) + \text{Reg}_s(T) + \text{Reg}_n(T)
\]

where \( \text{Reg}_e(T) \) is the regret due to exploration, \( \text{Reg}_s(T) \) is the regret due to sub-optimal classifier selections in exploitation and \( \text{Reg}_n(T) \) is the regret due to near-optimal classifier selections in exploitation. The following series of lemmas bound each of these terms separately. We start with a simple lemma which gives an upper bound on the highest level subspace that is active at any time \( t \).

**Lemma 1.** All active subspaces in \( A_t \) at time \( t \) have a level of at most \( (\log_2 t)/p + 1 \).

**Proof:** Let \( l + 1 \) be the highest level. According to the partitioning process, we must have \( \sum_{j=1}^{l} A2^{pj} < k \), otherwise the highest level will be less than \( l + 1 \). From this, for \( t > A \), we have \( l < \log_2(t)/p \).

The next three lemmas bound the regret for any level \( l \) context subspace.

**Lemma 2.** If \( D(t) = t^z \ln t \), then for any level \( l \) subspace, the regret due to exploration by time \( t \) is bounded above by \( K(t^z \ln t + 1)\Delta \).
Proof: Time slot \( t \) is an exploration slot if and only if there exists \( f \) such that \( N_C^t(f) \leq D(t) \). Therefore, up to time \( T \), there can be at most \( t^z \ln t + 1 \) exploration time slots for each classifier \( f \). Since there are a total number of \( K \) classifiers, the number of exploration slots is upper bounded by \( K(t^z \ln t + 1) \).

Lemma 3. Let \( B = \frac{2}{L^p(d)^2 - \alpha} + 2 \). If \( p > 0, 2 \alpha p \leq z < 1 \), \( D(t) = t^z \ln t \), then for any level \( l \) subspace \( C \), the regret due to choosing sub-optimal classifiers in exploitation steps is bounded by \( 2K \beta_2 \Delta \).

Proof: Let \( \Omega \) denote the space of all possible outcomes, and \( w \) be a sample path. The event that the algorithm exploits in \( C \) at time \( t \) is given by

\[
W_C^t := \{ w : N_C(f) > D(t), \forall f; \theta^t \in C; C \in A^t \}
\]  

(5)

We will bound the probability that the algorithm chooses a sub-optimal classifier in an exploitation step in subspace \( C \), and then bound the expected number of times a sub-optimal classifier is chosen by the algorithm. Recall that the loss in every slot is at most \( \Delta = 1 \). Let \( V_C^t(f) \) be the event that a sub-optimal classifier \( f \) is chosen. Then

\[
\text{Reg}_{C,s}(T) \leq \sum_{t=1}^{T} \sum_{f \in L_C(B)} P(V_C^t(f), W_C^t)
\]

(6)

For any classifier \( f \), we have

\[
\{ V_C^t(f), W_C^t \}
\]

(7)

\[
\subset \{ \bar{r}_C(f) \geq \bar{\mu}_C(f) + H_t, W_C^t \}
\]

(8)

\[
\cup \{ \bar{r}_C(f^*) \leq \mu_C(f^*) - H_t, W_C^t \}
\]

(9)

\[
\cup \{ \bar{r}_C(f) \geq \bar{r}_C(f^*), \bar{r}_C(f) < \bar{\mu}_C(f) + H_t \}
\]

(10)

\[
\bar{r}_C(f^*) > \mu_C(f^*) - H_t, W_C^t \}
\]

(11)
for some $H_t > 0$. This implies

$$P(\mathcal{Y}_C^t(f), \mathcal{W}_C^t)$$

$$\leq P(\bar{r}^\text{best}_C(N_C(f)) \geq \tilde{\mu}_C(f) + H_t + Ld^{\alpha/2}2^{-t\alpha}, \mathcal{W}_C^t)$$

$$+ P(\bar{r}^\text{worst}_C(N_C(f^*)) \leq \mu_C(f^*) - H_t - Ld^{\alpha/2}2^{-t\alpha}, \mathcal{W}_C^t)$$

$$+ P(r^\text{best}_C(N_C(f)) \geq \bar{r}^\text{worst}_C(M_C(f^*)))$$

$$\bar{r}^\text{best}_C(N_C(f)) < \tilde{\mu}_C(f) + H_t,$$

$$\bar{r}^\text{worst}_C(N_C(f^*)) > \mu_C(f^*) - H_t, \mathcal{W}_C^t)$$

Consider the last term in the above equation. In order to make it 0, we need $2H_t \leq (B - 2)Ld^{\alpha/2}2^{-t\alpha}$. This holds when $2H_t \leq (B - 2)Ld^{\alpha/2}2^{-t\alpha}$. Therefore, for $H_t = t^{-z/2}, z \geq 2\alpha/p$ and $B = \frac{2}{Ld^{\alpha/2}2^{-\alpha}} + 2$, the last term is 0. Then, by using a Chernoff-bound, for any $f \in \mathcal{L}_C(B)$, since on the event $\mathcal{W}_C^t, N_C(f) \geq t^z \ln t$, the first two terms in the last line in the above equation are both bounded by

$$e^{-2(2t^z\ln t)} \leq t^{-2} \quad (12)$$

Therefore $\text{Reg}_{C,s}(T) \leq \sum_{t=1}^{T} \sum_{f \in \mathcal{L}_C(B)} t^{-2} \leq K\beta_2$.

Lemma 4. Let $B = \frac{2}{Ld^{\alpha/2}2^{-\alpha}} + 2$. If $p > 0, 2\alpha p \leq z < 1, D(t) = t^z \ln t$, then for any level $l$ subspace $C$, the regret due to choosing near-optimal classifiers in exploitation slots is bounded above by $2ABLd^{\alpha/2}2^{(p-\alpha)t}\Delta$.

Proof: The one-slot regret of any near-optimal classifier $f$ is bounded by $2BLd^{\alpha/2}2^{-t\alpha}$. Since $C$ remains active for at most $A2^d$ context arrivals, we obtain the desired bound.

In order to obtain the regret bound of the proposed online activity classification algorithm, we need to consider how many subspaces of each level is formed up to time $T$. The number of such subspaces explicitly depends on the context information arrival process. We are now ready to prove Theorem 1.

Proof of Theorem 1: (1) Let $B = \frac{2}{Ld^{\alpha/2}2^{-\alpha}} + 2$. We first consider the worst case. It can be shown that in the worst case the highest level subspace has level at most $1 + \log_{2^{d+1}} T$. The
total number of subspaces is bounded by
\[
\sum_{l=0}^{1+\log_{2p+d} T} 2^d l \leq 2^{2dT^{d/p}}
\]  
\tag{13}

According to Lemma 4, the regret from choosing a near optimal classifier is
\[
\text{Reg}_n(T) \leq 2ABLd^{\alpha/2} \sum_{l=0}^{1+\log_{2p+d} T} 2^{(p-\alpha)l}
\leq 2ABLd^{\alpha/2}2^{2(d+p-\alpha)}T^{d+p-\alpha \over d+p}
\]  
\tag{14}

\tag{15}

Hence, \text{Reg}_n(T) is on the order of \(T^{d+p-\alpha \over d+p}\). Moreover, since the number of activated subspaces is on the order of \(O(T^{d+p-\alpha \over d+p})\), according to Lemma 2, \text{Reg}_e(T) is on the order of \(O(T^{d+p-\alpha \over d+p} \ln T)\) and according to Lemma 3, \text{Reg}_s(T) is on the order of \(O(T^{d+p-\alpha \over d+p})\), for \(z \geq 2\alpha / p\). These three parts of the regret are balanced when \(z = 2\alpha / p\) and \(d+p-\alpha \leq d+p-\alpha \leq d+p + 2\alpha / p\). Solving for \(p\) we get
\[
p = 3\alpha + \sqrt{9\alpha^2 + 8\alpha d / 2}
\]  
\tag{16}

Therefore, the regret is \(\text{Reg}(T) = O(T^{2 \alpha / (d+1.5\alpha)})\). We can further simplify the expression by noticing that
\[
{d + \alpha / 2 + \sqrt{9\alpha^2 + 8\alpha d / 2} \over d + 3\alpha / 2 + \sqrt{9\alpha^2 + 8\alpha d / 2}} < \frac{d + \alpha / 2 + (3\alpha + 2d) / 2}{d + 3\alpha / 2 + (3\alpha + 2d) / 2} = \frac{d + \alpha}{d + 1.5\alpha}
\]  
\tag{17}

Hence, \(\text{Reg}(T) = O(T^{d+\alpha \over d+1.5\alpha})\).

(2)Now we consider the best case, the number of activated subspaces is upper bounded by \(\log_2 T / p + 1\), and by the property of context arrivals all activated subspaces have different levels. We calculate the regret from choosing near optimal classifiers as
\[
\text{Reg}_n(T) \leq 2ABLd^{\alpha / 2} \sum_{l=0}^{1+\log_2^{T/p}} 2^{p-\alpha} l
\leq 2ABLd^{\alpha / 2}2^{2(p-\alpha)}T^{p-\alpha \over 2p-\alpha}
\]  
\tag{18}

\tag{19}

The other regret parts are the same as the worst case. These three parts are balanced by setting \(z = 2\alpha / p, p = 3\alpha\). Substituting these parameters we obtain \(\text{Reg}(T) = O(T^{2/3})\).