An Asymptotically Optimal Contextual Bandit Algorithm Using Hierarchical Structures

Mohammadreza Mohaghegh Neyshabouri, Kaan Gokcesu, Hakan Gokcesu, Huseyin Ozkan, and Suleyman S. Kozat, *Senior Member, IEEE*

***Abstract*—We propose an online algorithm for sequential learning in the contextual multi-armed bandit setting. Our approach is to partition the context space and then optimally combine all of the possible mappings between the partition regions and the set of bandit arms in a data driven manner. We show that in our approach, the best mapping is able to approximate the best arm selection policy to any desired degree under mild Lipschitz conditions. Therefore, we design our algorithm based on the optimal adaptive combination and asymptotically achieve the performance of the best mapping as well as the best arm selection policy. This optimality is also guaranteed to hold even in adversarial environments since we do not rely on any statistical assumptions regarding the contexts or the loss of the bandit arms. Moreover, we design an efficient implementation for our algorithm using various hierarchical partitioning structures such as lexicographical or arbitrary position splitting and binary trees (and several other partitioning examples). For instance, in the case of binary tree partitioning, the computational complexity is only log-linear in the number of regions in the finest partition. In conclusion, we provide significant performance improvements by introducing upper bounds (w.r.t. the best arm selection policy) that are mathematically proven to vanish in the average loss per round sense at a faster rate compared to the state-of-the-art. Our experimental work extensively covers various scenarios ranging from bandit settings to multi-class classification with real and synthetic data. In these experiments, we show that our algorithm is highly superior over the state-of-the-art techniques while maintaining the introduced mathematical guarantees and a computationally decent scalability.**

***Index Terms*—Contextual bandits, universal, online learning, adversarial, big data, multi-class classification.**

I. Introduction

We study online learning [1], [2] in the contextual multi-armed bandit setting [3]–[8]. In the classical formulation of the multi-armed bandit problem, one of the available M bandit arms (or *actions*) is chosen at each round to obtain a reward (or loss), and the reward (or loss) of all of the other unchosen $M-1$ arms stay oblivious. The objective is to maximize the cumulative reward of the selected arms in a series of rounds. Since the reward we would obtain from the other arms remain hidden, this setting can be considered as a limited feedback version of prediction with expert advice [9]–[14]. Additionally, the well-known fundamental trade-off between exploration and exploitation [15], [16] naturally appears in multi-armed bandits. One should balance exploitation of actions that gave the highest payoffs in the past and exploration of actions that might give higher payoffs in the future.

The multi-armed bandit problem has attracted significant attention due to the applicability of the bandit setting in a wide range of applications from online advertisement [17] and recommender systems [18]–[20] to clinical trials [21] and cognitive radio [22], [23]. For example, in the online advertisement application, different ads available to display to users are modeled as the bandit arms and the act of clicking by the user on the displayed ad is modeled as the reward [17].

In many instances of the bandit algorithms, additional information is available [24] such as the age or the gender of the patient in clinical trials [25], which is useful about the arm selection decision. However, most of the conventional bandit algorithms do not exploit or fail to fully exploit this information [26]–[28]. To remedy, contextual multi-armed bandit algorithms are introduced [16], [17], [29], where the additional information is represented as a context vector. For example, in the online advertisement applications, this context vector may contain certain information about the users such as historical activities or demographic/geographical information. Then the goal of the multi-armed bandit problem is extended to maximally exploit this additional information, i.e., the context, for optimizing the arm selection strategy and therefore gaining more rewards (or suffering less loss).

We consider the contextual extension in the online setting, where we operate sequentially on a stream of observations from a possibly non-stationary, chaotic or even adversarial environment [30]–[32]. Hence, we have no statistical assumptions on the context vectors and behavior of the bandit arms so that our results are guaranteed to hold in an individual sequence manner [16]. We follow a competitive algorithm perspective [16] and define the performance (total time accumulated reward or loss) with respect to a competition class of context dependent bandit arm selection policies. For this purpose, we design an exponentially large and parameterized competition class of predetermined mappings from the space of context vectors to the bandit arms such that the best arm

This work is supported in part by Turkish Academy of Sciences Outstanding Researcher Programme, TUBITAK Contract No. 113E517.

M. Mohaghegh N. and S. S. Kozat are with the Department of Electrical and Electronics Engineering, Bilkent University, Ankara, Turkey, e-mail: {mohammadreza, kozat}@ee.bilkent.edu.tr, tel: +90 (312) 290-2336.

K. Gokcesu is with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA, e-mail: gokcesu@mit.edu.

H. Gokcesu is with the School of Computer and Communication Sciences, Ecole Polytechnique Federale de Lausanne, Ecublens VD 1015 Switzerland, e-mail: hakan.gokcesu@epfl.ch.

H. Ozkan is with the Faculty of Engineering and Natural Sciences at Sabancı University, Istanbul 34956 Turkey, e-mail: hozkan@sabanciuniv.edu, tel: +90 (216) 483-9594.

selection policy[1] can be approximated arbitrarily well to a desired degree by the optimal mapping in the competition class. We point out that each mapping in our competition class partitions the space of context vectors into several disjoint regions and assigns each one of these regions to one of the bandit arms, i.e., each mapping selects the bandit arm corresponding to the region containing the observed context vector. Based on this competition class of such mappings, our goal is to asymptotically -at least- achieve[2] the performance of the optimal mapping as well as the performance of the best arm selection policy at a faster convergence (performance-wise or in terms of the convergence of the regret upper bound to zero) rate compared to the state-of-the-art as more data is observed.

In order to generate partitions of the context space and therefore a rich competition class, we use various hierarchical partitioning structures [33] such as the ones based on lexicographical or arbitrary position splitting, binary trees and several other partitioning examples, cf. Section IV. In our design, each of these structures leads to a different competition class but approximates (arbitrarily well, and even perfectly if desired) the same best arm selection policy by the optimal mapping in the corresponding competition class. However, each hierarchical structure encodes the best arm selection policy differently and one of them is the most efficient in the sense of the required number of partition regions (i.e. less number of regions means higher efficiency). Therefore, we explore various hierarchical structures and introduce an algorithm which covers each of such structures by using a carefully designed weighting over the corresponding competition class. The output of the introduced algorithm is the optimal data adaptive combination (w.r.t. the designed weighting) of the policies (aforementioned mappings) in the competition class. Our weighting/adaptive combination favors simpler models in the beginning of the data stream and gradually switches to more complex ones as the data overwhelms.

As a result, our algorithm is guaranteed to asymptotically perform -at least- as well as the best arm selection policy. We achieve this performance optimality at a faster convergence rate (for instance, at the rate $O(\sqrt{(RM\ln M\ln N)/T})$ in the case of binary tree partitioning after averaging the regret bound over T where R is the number of regions in the optimal partition, M is the number of bandit arms, N is the number of regions in the finest partition in the competition class and T is the number of rounds) compared to the state-of-the-art[3] rate $O(\sqrt{(MN\ln M)/T})$. Note that here, typically, $N >> R$ is the dominating factor. Our superior performance is due to exploiting the right hierarchical partitioning structure that encodes the best policy more efficiently and therefore assigns higher initial weights to the optimal partition. This exploitation of the right structure with the introduced weighting scheme also mitigates the overfitting issue as an additional merit.

We emphasize that our algorithm is designed to work for a generic class of hierarchical partitioning structures and our optimality results do hold for each type of structure in this generic class. Therefore, one can use the proposed algorithm with any type of partitioning that is appropriate for the target application with the corresponding performance guarantees. Such guarantees include upper bounds on the regret w.r.t. the best arm selection policy that are mathematically proven to vanish at $O(1/\sqrt{T})$ (after averaging over T) in a superior manner over the state-of-the-art, cf. the following Section I-A *Prior Art* and Section IV for detailed comparisons. We also present a computationally highly efficient implementation for the introduced algorithm that, for instance, combine M^N mappings with only computational complexity of $O(M\ln N)$ in the case of binary tree partitioning structure. Through an extensive set of experiments with real and synthetic data, we demonstrate the proposed approach in several scenarios such as multi-class classification, online advirtisement and multi-armed bandit along with various partitioning structures. In these experiments, our algorithm is shown to significantly outperform the state-of-the-art techniques with real-time data processing and strong modeling capabilities.

A. Prior Art

The contextual bandit problem is mostly studied in the stochastic setting [29], [34], [35], where context vectors and losses are assumed to be drawn randomly and independently from an unknown distribution. Additional assumptions regarding the relations between the context vectors and the arm losses are also used in other studies, e.g., a linear relation in [17] and [36], and more general ones in [37]. These algorithms essentially fail to hold their performance guarantees if the context vectors or the arm losses are chosen by an adversary rather than a prefixed distribution.

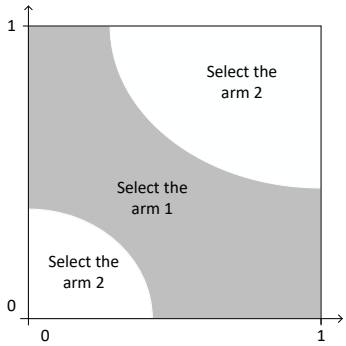
An alternative to the stochastic approaches is the adversarial setting, where algorithms do not use any assumptions on the behavior of the context vectors and bandit arms. The well-known *EXP3* algorithm [32] formulates the non-contextual bandit problem in an adversarial setting and achieves a regret upper bound[4] of $O(\sqrt{TM\ln M})$ against the best arm. *S-EXP3* algorithm [16] is a naive extension of *EXP3* in the contextual setting, which partitions the context space and runs independent *EXP3* algorithms over each one of the partition regions. *S-EXP3* achieves a regret upper bound of $O(\sqrt{TNM\ln M})$ against the best mapping from the regions to the bandit arms, where N is the number of regions in the partition of the context space. As implied by the regret bound, the *S-EXP3* algorithm works well only when the complexity (the granularity or the level of detailing/fine-ness) of the required partitioning to model the truly optimal selection policy is relatively small, otherwise it quickly overfits and suffer from insufficient data.

[1]This best arm selection policy is based on the fixed best partitioning of the context space and the best assignment of the arms to the regions of that best partition. It is not necessarily in our competition class. However, it can be approximated arbitrarily well by the optimal mapping in the class by varying the class parameter; and it can be determined only when the complete data stream is observed.

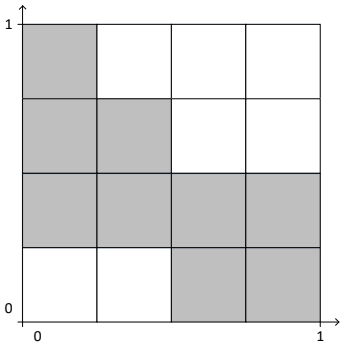
[2]In addition to achieving, we might well outperform since our approach is data driven and based on combination of partitions, i.e., we do not rely on a single fixed partition.

[3]The convergence rates given here samples our general regret results (after averaging over T) in the case of binary tree partitioning. Our rates for other partitionings in our generic class of hierarchical structures naturally vary but our superiority compared to the state-of-the-art stays valid in a similar manner, cf. Section IV for our complete regret results for all structures.

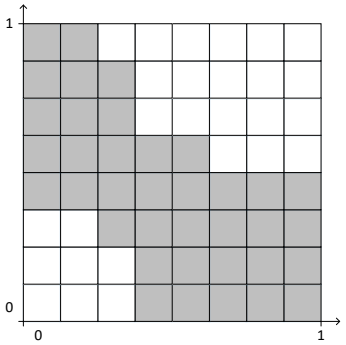
[4]We illustrate regret upper bounds without averaging over T here in this section; but with averaging in the previous section to demonstrate the convergence to 0 there.



(a) An example mapping from the context space $[0,1]^2$ to the set of bandit arms $\{1,2\}$.



(b) Closest mapping in the quantized competition class with 16 quantization levels to the mapping in Fig. 1a.



(c) Closest mapping in the quantized competition class with 64 quantization levels to the mapping in Fig. 1a.

Fig. 1: An example mapping from the context space to the set of bandit arms and its approximations in the quantized competition classes. In each mapping above, the dark and bright sections are mapped to the arms 1 and 2, respectively.

The *EXP4* algorithm [32] is another extension of *EXP3* in the contextual setting. In this algorithm, a set of K experts observe the context vectors and suggest distributions on the arms. Their suggestions are adaptively combined to select the arm to pull. It is shown that *EXP4* achieves a regret upper bound of $O(\sqrt{TM\ln K})$ against the best expert. Considering the M^N mappings from a partition of the context space to the arms as the K experts, *EXP4* achieves $O(\sqrt{TNM\ln M})$ against the optimal mapping. As we show in Section III, the *EXP4* algorithm can be improved by producing an initial tendency (in earlier times of the stream) toward the mappings of smaller complexity. In this case, although the finest partition has N regions (and hence there are M^N mappings in total), it suffices to run *EXP4* over $O((NM)^R)$ mappings with R regions resulting a regret bound of $O(\sqrt{TMR\ln(NM)})$, if the optimal partition consists of R regions. However, the main problem with this algorithm is its computational complexity of $O((NM)^R)$. On the other hand, the *CSB-FTPL* algorithm [38] achieves a regret upper bound of $O(T^{2/3}M\sqrt{\ln K})$ against the best expert among a set of K experts with a computational complexity that is polynomial in $\ln K$. Hence, running *CSB-FTPL* over $O((NM)^R)$ mappings with R disjoint regions yields a regret upper bound of $O(T^{2/3}M\sqrt{R\ln N})$ with a polynomial computational complexity in $\ln N$.

We emphasize that we seek to achieve a regret upper bound vanishing (w.r.t. rounds/time after averaging over T) faster than that of *EXP4* with a computational complexity linear in $\ln N$ which allows us to grow the hierarchical structure freely. To this end, our algorithm not only drastically reduces the computational complexity (e.g., down to $O(M\ln N)$ in the case of binary tree partitioning) compared to the discussed state-of-the-art techniques, but also achieves a regret upper bound of $O(\sqrt{TMR\ln M\ln N})$.

Finally, a simple instance of our hierarchical structures, the context trees, are widely used in various applications including but not limited to data compression [39], [40], estimation [41], [42], communications [43], regression [44], [45] and classification [46]. In all aforementioned applications, context trees are used to partition the context space in a nested structure, run an independent adaptive model over each one of the tree nodes and combine the models. On the other hand, in this paper, we use a generalized novel notion of hierarchical structures that is specifically designed for the completely different multi-armed contextual bandit problem.

B. Contributions

- We introduce a novel and efficient contextual bandit arm selection algorithm, which first quantizes the space of context vectors and then achieves the performance of the optimal mapping from the quantized regions to the bandit arms (in the average loss per round sense).
- We introduce an efficient quantization method and show that using this quantization method, our algorithm asymptotically achieves (not only the optimal mapping but also) the performance of the best arm selection policy (in the average loss per round sense) as the number of quantization levels increases.
- We introduce a novel and generalized notion of hierarchical context space partitioning structures for the contextual bandit setting and use such hierarchical structures to design an efficient implementation of our algorithm and achieve a faster convergence rate for the regret compared to the state-of-the-art.
- We demonstrate significant performance gains with the proposed algorithm in comparison to the state-of-the-art techniques through extensive experiments involving both synthetic and real data.

C. Organization of the Paper

In Section II, we describe the contextual multi-armed bandit framework. Next, we explain a first mixture of experts based approach and its challenges in Section III. In Section IV, we explain the notion of hierarchical structures and implement our algorithm using these structures. We introduce an efficient quantization method in Section V, and show that our algorithm is competitive against any mapping, including the best arm selection policy, from the context space to the bandit arms.

Section VI contains the experimental results over several synthetic and well known real life datasets followed by the concluding remarks in Section VII.

II. Problem Description

We study the contextual bandit problem in an adversarial setting[5]. Recall that the original multi-arm bandit problem is a sequential game. One of the available bandit arms $I_t \in \{1, ..., M\}$ is selected at each round t and then a related loss l_{t,I_t} is observed[6]. The objective is to minimize the accumulated loss $\sum_{t=1}^{T} l_{t,I_t}$ in a sequence of T rounds. In the contextual extension, a context vector $\mathbf{s}_t$ from a context space S is additionally provided at each round before selecting the arm. For example, S is $[0,1]^2$ in Fig. 1. Then the objective stays same but can be improved with the available context.

We consider this contextual bandit problem in adversarial setting [47], where at each round t, an adversary assigns a specific loss to each arm $i \in \{1, 2, \cdots, M\}$ simultaneously in parallel with the player who chooses an arm to pull. The adversary's goal is to maximize the player's loss, whereas the player tries to maximize her/his gain (here the loss maximization by the opponent give the name "adversary"). We emphasize that the adversary is provided with all the information from the previous rounds. It can even know the algorithm followed by the player. However, if the player's choice is randomized, then the adversary does not know the outcome of this randomization while assigning the losses to the arms, e.g, the adversary may know that the player tosses a coin to choose the arm to pull, but does not know the outcome of the toss. Namely, "adversarial setting" refers to the algorithmic framework or the game in which the data generation (assignment of losses in this case) or the adversary is acting against the player on purpose while the player tries to maximize her/his gain. In accordance with the nature of this adversarial setting, in designing the algorithm for the player to use, we make no statistical assumptions about the context vectors and the bandit arms [32], and our performance bounds are guaranteed to hold in an individual sequence manner. Hence, in designing our algorithm, we rigorously address such adversarial conditions and provide strong mathematical guarantees that hold for all possible data streams or for all possible moves of the adversary. Our algorithm is strictly sequential such that at each round t, it selects an arm I_t according to the information coming from the previous rounds including observed context vectors, selected arms and their losses, alongside the context vector we are currently observing, i.e.,

$$I_t = f_t(\mathbf{s}_t; \mathbf{s}_{t-1}, I_{t-1}, l_{t-1,I_{t-1}}; ...; \mathbf{s}_1, I_1, l_{1,I_1}). \quad (1)$$

In design of our algorithm, we aim at sequentially learning the optimal partitioning of the context space with the optimal assignment between the regions of the learned partition and the set of arms. For this purpose, we investigate a general framework of hierarchical structures to generate context space partitions and eventually learn the asymptotically optimal, time varying, context driven arm chooser f_t. We show that our approach, compared to the state-of-the-art techniques, yields a computationally highly superior algorithm with real time data processing capabilities while achieving a faster convergence rate to the optimal conditions (in terms of the convergence of the regret upper bounds to 0). The superiority of the proposed algorithm is due to that the set of all possible context space partitions considered here can theoretically achieve arbitrarily high degree of granularity (can be of arbitrarily high capacity) whereas the true complexity of the optimal partition is limited (cf. Section IV) in reality. Based on this observation, our approach additionally allows the regret analysis to incorporate an upper bound on the complexity of the optimal partition, which in turn significantly improves the convergence of the presented algorithm in almost all practical scenarios. This gain is essentially from $O(\sqrt{N})$ to $O(\sqrt{\ln N})$ (N is measuring the granularity, cf. Section IV). If the complexity of the optimal partition cannot be upper bounded, which would be a purely theoretical consideration as the true complexity is almost always limited and finite in real scenarios, our regret analysis then produces similar rates of convergence in that very worst theoretical scenario. Nevertheless, in any case, the proposed algorithm is computationally highly efficient and superior, and asymptotically optimal in the adversarial setting including the very worst scenario regardless of the stationary or non-stationary or perhaps chaotic source statistics.

To this end, we consider a large class $\mathcal{G}$ of deterministic mappings, i.e., $\forall g \in \mathcal{G}$, $g : S \to \{1, ..., M\}$. Each such mapping is composed of a fixed partition of the context space and an arm is assigned to each partition region. Depending on the partition region that a context s_t falls in, g chooses the assigned arm $g(s_t)$. An example is shown in Fig. 1a in the case of 2 dimensional context space $S = [0,1]^2$ with 2 bandit arms, where $g([0.5, 0.5]^T) = 1$. Note that for a given $g \in \mathcal{G}$, all of the other deterministic mappings resulting from all possible arm assignments to the regions of the partition of g are also included in $\mathcal{G}$. Since we work in the adversarial setting and therefore refrain from making any statistical assumptions about the context vectors and the loss of the bandit arms [32], we next define our performance w.r.t. the optimum (minimum loss) mapping in the "competition" class $\mathcal{G}$ based on the following regret:

$$\mathcal{R}(T, \mathcal{G}) \triangleq \max_{g \in \mathcal{G}} \mathbb{E}\left[\sum_{t=1}^{T} l_{t,I_t} - \sum_{t=1}^{T} l_{t,g(\mathbf{s}_t)}\right], \quad (2)$$

where the expectation is w.r.t. the internal randomization in our algorithm (the internal randomization here is not related to data statistics). Our goal is to upper bound the regret by a term that depends sublinearly in T, and hence asymptotically achieve -at least- the performance of the best g in $\mathcal{G}$ (in the averaged regret per round sense). Achieving this goal is equivalent to achieving the performance of the chooser of the optimal context space partition with the optimal assignment to the arms. Here, optimality of the context space partition

[5]All vectors are column vectors and denoted by boldface lower case letters. For a K-element vector $\mathbf{u}$, u_i represents the i^{th} element and $\|\mathbf{u}\| = \sqrt{\mathbf{u}^T\mathbf{u}}$ is the l^2-norm, where $\mathbf{u}^T$ is the transpose. Indicator function $\mathbf{1}_{\{\cdot\}} \in \{0, 1\}$ outputs 1 only if its argument condition holds. A function $f : \mathbb{R}^n \to \mathbb{R}$ is Lipschitz continuous over a region $W \subset \mathbb{R}^n$, if there exists a non-negative constant c such that $|f(\mathbf{x}_1) - f(\mathbf{x}_2)| \leq c\|\mathbf{x}_1 - \mathbf{x}_2\|$ for all $\mathbf{x}_1, \mathbf{x}_2 \in W$.

[6]We assume $l_{t,I_t} \in [0, 1]$ for simplicity, however, it can be straightforwardly shown that our results hold for any bounded loss after shifting and scaling in magnitude.

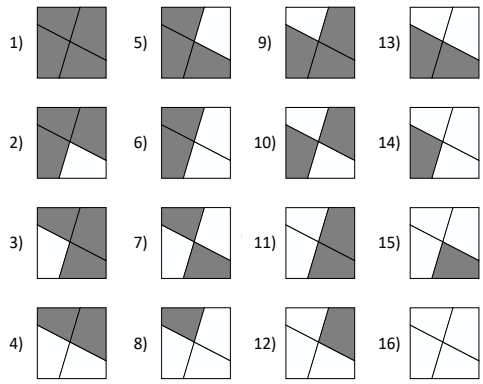


Fig. 2: All possible mappings in a 2-armed bandit problem with a predetermined quantization of the context space $S = [0,1]^2$ into 4 regions. In each mapping above, the dark and bright regions are mapped to the arms 1 and 2, respectively.

should be understood w.r.t. the class $\mathcal{G}$ which is certainly not restrictive, since it can be arbitrarily improved by generalizing (detailing) $\mathcal{G}$ to a desired degree, cf. Section III.

We next construct the class $\mathcal{G}$ and provide a mixture-of-experts based first solution to the introduced problem.

III. A Contextual Bandit Algorithm Based on Mixture of Experts

The ultimate goal in the contextual bandit problem is ideally to achieve the performance of the best mapping *in the set* $\mathcal{U}$[7] *of all arbitrary mappings* from the context space to the bandit arms. Since this set of all arbitrary mappings is too powerful to compete against in design of an algorithm, as the first step, we uniformly quantize the context space S into N disjoint regions $r_1, r_2, ..., r_N$, i.e., $\cup_{i=1}^{N} r_i = S$ and $r_i \cap r_j = \varnothing$ for $\forall i \neq j$. We use uniform quantization for simplicity, however, one can incorporate any arbitrary type of quantization into our framework straightforwardly. In our framework, we consider all possible assignments between the set of disjoint regions and the set of bandit arms, and call each context mapping resulting from one of those assignments an N-level quantized mapping. Therefore, each N-level quantized mapping is essentially a function from $\cup_{i=1}^{N} r_i = S$ to $\{1, ..., M\}$: a context $s \in r^* \subset S$ is mapped to the bandit arm that the region r^* is assigned to. Two examples of such quantized mappings of different levels for the case of 2-armed bandit with the context space $[0,1]^2$ are shown in Fig. 1b and Fig. 1c. Given a quantized context space $S = \cup_{i=1}^{N} r_i$, we define the class $\mathcal{G}^N$ of N-level quantized mappings as the "competition class" with N quantization levels consisting of all arbitrary assignments between the bandit arms and the given N regions $\{r_i\}_{i=1}^{N}$.

Remark: We seek to achieve the performance of the best quantized mapping in $\mathcal{G}^N$, which can get arbitrarily close (and N can be freely chosen in our framework) to the performance of the *best arbitrary mapping in* $\mathcal{U}$*, i.e., the best arm selection policy,* as N increases. For example, suppose that the mapping shown in Fig. 1a is the *best arbitrary mapping*. In this case, the mappings in Fig. 1b and Fig. 1c of improving optimalities will be the best mappings in $\mathcal{G}^{16}$ and $\mathcal{G}^{64}$, respectively.

Based on M^N different mappings in $\mathcal{G}^N$, we consider an expert chooser that is one-to-one-corresponding to each of those mappings such that $g_j(s)$ is the arm chosen by expert E_j for the context s, i.e., $E_j \leftrightarrow g_j, 1 \leq j \leq M^N$. An example of all 16 mappings followed by the experts for the case of $M = 2$ and $N = 4$ is shown in Fig. 2, where, unlike Fig. 1, we choose a nonuniform quantization to demonstrate the generality in our approach. One of these experts in Fig. 2 is $\mathcal{G}^4$-optimal for the underlying sequence of losses, however, naturally, we do not know which. Hence, instead of committing to a single expert, we next use a mixture of experts approach to learn the best one during rounds.

In order to achieve the performance of the best expert, we assign each expert E_j a weight $\alpha_{t,j}$ (showing our trust on the expert E_j at round t) and use exponentiated weights to adaptively combine them. After observing context $\mathbf{s}_t$ at each round t, we randomly select one of the experts using the probability simplex $\boldsymbol{\beta}_t = (\beta_{t,1}, ..., \beta_{t,M^N})$, where $\beta_{t,j} = \alpha_{t,j} / \sum_{k=1}^{M^n} \alpha_{t,k}$ is the normalized weight. Importantly, the probability of selecting each arm then follows the probability simplex $\mathbf{p}_t = (p_{t,1}, ..., p_{t,M})$, where

$$p_{t,i} = \sum_{j=1}^{M^N} \beta_{t,j} \mathbf{1}_{\{g_j(\mathbf{s}_t)=i\}}. \tag{3}$$

We initially set the weights $\alpha_{1,i}$ according to the complexity of the mappings of experts from $\mathcal{G}^N$, and use exponentiated losses to update during rounds: at each round $t \geq 2$, we have

$$\alpha_{t,i} = \alpha_{1,i} e^{-\eta \sum_{\tau=1}^{t-1} \tilde{l}_{\tau, g_i(\mathbf{s}_\tau)}}, \tag{4}$$

where $\eta \in \mathbb{R}^+$ is the (constant) learning rate and $\tilde{l}_{\tau, g_i(\mathbf{s}_\tau)}$ is the unbiased estimator of $l_{\tau, g_i(\mathbf{s}_\tau)}$. Since we do not observe the loss $l_{t,m}$ of the unchosen arms, we use the unbiased estimator

$$\tilde{l}_{t,m} = \begin{cases} \frac{l_{t,m}}{p_{t,m}} & m = I_t \\ 0 & m \neq I_t \end{cases}, \tag{5}$$

where $\mathbb{E}[\tilde{l}_{t,m}] = l_{t,m}$. Using this bandit arm selection probability assignment defined through (3), (4) and (5), we have the following regret result.

Theorem 1. *Consider an M-armed contextual bandit problem. If the context space is quantized into N disjoint regions, and experts E_j's are following the M^N possible mappings in $\mathcal{G}^N$ as described in Section III, then $\mathcal{R}(T, E_j)$ satisfies*

$$\mathcal{R}(T, E_j) \leq \frac{\ln(1/\beta_{1,j})}{\eta} + \frac{MT\eta}{2} \tag{6}$$

based on the probability assignments defined through (3), (4) *and* (5)*, where T is the number of rounds, $\eta \in \mathbb{R}^+$ is the learning rate parameter in* (4) *and $\beta_{1,j}$ is the normalized initial weight of the j^{th} expert E_j.*

Proof of Theorem 1 follows similar lines to the proof of Theorem 4.2 in [16] with certain variations due to our arbitrary initial weighting as opposed to uniform initial weights of the experts in [16]. The proof of our Theorem 1 is provided in Appendix A.

[7]This set $\mathcal{U}$ consists of *all possible arbitrary* context space partitions (not confined to $\mathcal{G}$) with all possible assignments of partition regions to the arms.

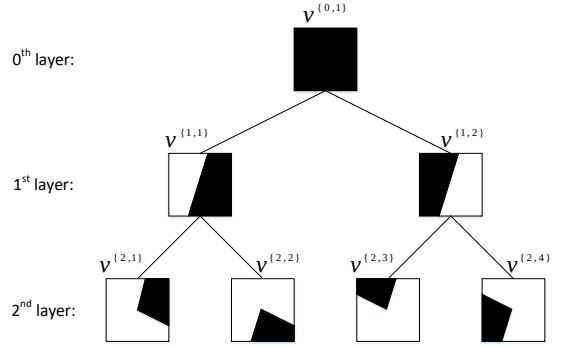


Fig. 3: A binary tree of depth $D = 2$ over the context space $[0, 1]^2$. The regions corresponding to each node are filled with black color.

We observe that the regret bound is logarithmically dependent on the reciprocal of the prior weight of the optimal partitioning in the competition class (i.e., its complexity cost). Hence, by using equal prior weights on the M^N experts, our regret bound will be in the order[8] of $O(\sqrt{NT})$ (after optimizing the learning rate). We point out that this result is similar to the *EXP4* algorithm [16], which achieves a regret upper bound of $O(\sqrt{NT})$ with optimum selection of the learning rate. Furthermore, *S-EXP3* algorithm [16] achieves a regret upper bound of the same order $O(\sqrt{NT})$ using an independent *EXP3* algorithm over each quantized region of the context space. This square root dependency of the regret bound on the quantization level is prohibitive and working against our motivation of approximating the performance of *the best arbitrary mapping* by freely increasing the number of quantization levels. Instead, we would like our regret bound to be dependent on the actual number R of disjoint regions that is needed and sufficient to model the actual complexity of the best arbitrary mapping whatever the quantization level N is. Hence, we want to achieve the order $O(\sqrt{RT})$. Moreover, working with these M^N parameters $\alpha_{t,1}, ..., \alpha_{t,M^N}$ has quite high space and computational complexities of $O(M^N)$.

To this end, we introduce hierarchical structures to generate context space partitions and exploit the level of complexity that is sufficient to model the best mapping over the introduced hierarchy. Thus, we achieve a regret upper bound with square-root dependency on the actual number of regions R in a computationally highly superior manner with significantly low space complexity.

IV. Hierarchical Structures

We use hierarchical structures to implement our contextual bandit algorithm *efficiently* in terms of both the regret upper bound convergence to 0 in average loss per round sense as well as computational and space complexities. Suppose that we have H nodes in a hierarchical structure labeled v_i, $i \in \{1, 2, ..., H\}$. We assign each node v_i a region r_i from the context space and there is hierarchical connection from each parent node to its child nodes. Let Φ_i be the set of child node groups of the node v_i, where each group $\phi \in \Phi_i$ consists of child nodes such that the union of their corresponding regions gives the region associated with the parent node v_i.

For instance, consider the binary tree of depth 2 in Fig. 3, which quantizes the 2-dimensional context space $S = [0, 1]^2$. Each node of such binary tree corresponds to a region of the context space, as shown in the figure. The region corresponding to each node is the union of the regions of its child nodes. Hence, for each node v_i in this tree (except for the leaf nodes), the set Φ_i is of size 1, which consists of only one group of cardinality 2 (which is the parent node's child pair). For the leaf nodes, Φ_i is the empty set and, hence, has a size of 0.

Next, we use this hierarchical structure to compactly represent our experts and combine them in an efficient manner.

A. A Weighted Mixture of Experts Algorithm Using Hierarchical Structures

In the following, we explain the details of our efficient implementation of the mixture of experts algorithm (described in Section III) by using hierarchical structures and present several examples. In addition to achieving computational scalability in our implementation, another goal of our work is to incorporate the model complexity of the best expert to improve the upper bound on the regret.

Here, each expert is composed of a partition of the context space and an arm assigned to each partition region. The partition corresponding to each expert can be represented using several nodes of the hierarchical structure. Hence, each expert can be represented using several nodes (showing the partition) and an arm corresponding to each one of them (showing the arm assignments). As an example, consider a 2-armed bandit problem. Suppose that we use a binary tree of depth 2 to quantize the context space into 4 regions. In this case, we define $2^4 = 16$ experts as in Fig. 2. We represent 4 samples among these 16 experts on our binary tree in Fig. 4. In this figure, the nodes representing the partition corresponding to the experts are marked using the circles and the arm selected by the expert at each one of these nodes is declared over the node. We seek to adaptively combine all of the experts to achieve the performance of the best one as explained in Section III.

In order to implement our mixture of experts, over each node v_i, we define M parameters $\alpha_{t,m,i}$ for $m = 1$ to M as the weight of m^{th} arm in the node v_i. This weight shows our trust on the m^{th} arm when the context vector falls into the region corresponding to the node v_i. We set $\alpha_{1,m,i} = 1$ for all m's and v_i's, and for $t \geq 2$,

$$\alpha_{t,m,i} = \exp\left(-\eta \sum_{\tau=1}^{t-1} \frac{l_{I_\tau}}{p_{\tau,m}} \mathbf{1}_{\{I_\tau = m\}} \mathbf{1}_{\{\mathbf{s}_\tau \in r_i\}}\right). \quad (7)$$

We can easily update these weights as follows. At each round t, after we receive $\mathbf{s}_t$, calculate $\mathbf{p}_t$, select I_t^{th} arm and observe the loss l_{t,I_t}, we calculate

$$\alpha_{t+1,m,i} = \alpha_{t,m,i} \exp\left(-\eta \frac{l_{I_t}}{p_{t,m}} \mathbf{1}_{\{I_t = m\}} \mathbf{1}_{\{\mathbf{s}_t \in r_i\}}\right). \quad (8)$$

[8] For ease of exposition and simplicity in our order notation here, we drop the variables, on which the dependency of order is similar or same or negligible across the compared algorithms.

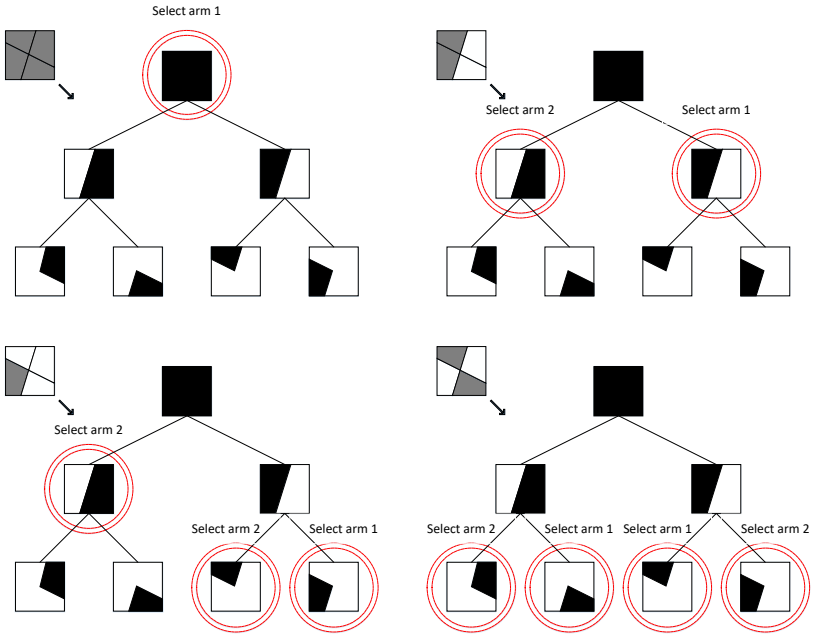


Fig. 4: Representation of 4 sample mappings in Fig. 2 over the binary tree in Fig. 3.

We point out that the weight of each expert $\alpha_{t,k}$ in (4) can be written as a multiplication of its initial weight and our weight parameters (i.e. $\alpha_{t,m,i}$'s) on the tree nodes corresponding to the mapping followed by the expert. To this end, in order to obtain the expert weights (cf. Theorem 2), we define another variable $w_{t,i}$ over each node v_i such that

$$w_{t,i} = \frac{1}{(|\Phi_i|+1)M}\sum_{m=1}^{M}\alpha_{t,m,i} + \frac{1}{|\Phi_i|+1}\sum_{\phi\in\Phi_i}\left(\prod_{j\in\phi} w_{t,j}\right). \tag{9}$$

Hence, if Φ_i is the empty set (i.e. $|\Phi_i| = 0$), then the equation simply becomes

$$w_{t,i} = \frac{1}{M}\sum_{m=1}^{M}\alpha_{t,m,i}. \tag{10}$$

The following proposition shows that using this recursion to calculate $w_{t,i}$ variables, the weight of the root node $w_{t,1}$ becomes equal to the sum of the expert weights, i.e., $\sum_k \alpha_{t,k}$ (as defined in (4)).

Proposition 1. *Using the recursive formula in* (9)*, at each node v_i, we have*

$$w_{t,i} = \sum_{k\in\Gamma_i}\alpha_{t,k}, \tag{11}$$

where Γ_i is the set of all experts defined over node v_i.

Proof of Proposition 1 is provided in Appendix B.

Now, in order to calculate the probability simplex in (3), we define M other variables to calculate $\sum_k \alpha_{t,k}\mathbf{1}_{\{g_k(\mathbf{s}_t)=i\}}$ for $i = 1, ..., M$. To this end, after we observe $\mathbf{s}_t$, we set

$$\gamma_{t,m,i} = \frac{1}{M}\alpha_{t,m,i}, \tag{12}$$

at the nodes v_i containing $\mathbf{s}_t$, where $|\Phi_i| = 0$ (i.e., leaf nodes). Then, we go up on the hierarchy using a recursive formula similar to the way we calculate $w_{t,i}$ variables in (9) as

$$\gamma_{t,m,i} = \frac{1}{(|\Phi_i|+1)M}\alpha_{t,m,i} + \frac{1}{|\Phi_i|+1}\sum_{\phi\in\Phi_i}\left(\prod_{j\in\phi} w_{t,j}\left(\frac{\gamma_{t,m,j}}{w_{t,j}}\right)^{\mathbf{1}_{\{\mathbf{s}_t\in r_j\}}}\right). \tag{13}$$

Using this recursion, we calculate $\gamma_{t,m,1}$ for $m = 1, ..., M$. The following proposition shows that using this recursion, $\gamma_{t,m,1}$ is the weighted sum of all experts, which select the m^{th} arm when they observe $\mathbf{s}_t$. Hence, we can build the probability simplex in (3) as

$$p_{t,m} = \gamma_{t,m,1}/w_{t,1}, \forall m \in \{1, ..., M\}. \tag{14}$$

Proposition 2. *Using the recursive formula in* (13)*, at each node v_i, for all $m \in \{1, ..., M\}$, we have*

$$\gamma_{t,m,i} = \sum_{k\in\Gamma_i}\alpha_{t,k}\mathbf{1}_{\{g_k(\mathbf{s}_t)=m\}}, \tag{15}$$

where Γ_i is the set of all experts defined over node v_i.

Proof of Proposition 2 is provided in Appendix C.

With the proposed implementation of the algorithm, at each round t, after observing $\mathbf{s}_t$, we first calculate $\gamma_{t,m,1}$ for $m = 1, ..., M$ and then divide by $w_{t,1}$ to form the probability simplex $\mathbf{p}_t = (p_{t,1}, ..., p_{t,m})$, using which we select an arm I_t. After we select our arm and suffer the loss according to the selected arm, we first update $\alpha_{t,I_t,i}$ parameters at the nodes containing $\mathbf{s}_t$. Then, we update $w_{t,i}$ variables at these affected nodes and go to the next round. The pseudo code of the explained procedure is provided in Algorithm 1.

Next, we show the regret bound of our hierarchical structure algorithm.

Algorithm 1 Hierarchical Structure based Bandits (*HSB*)

Parameter:
Set constant $\eta \in \mathbb{R}^+$
Initialization:
Initialize the structure including nodes v_i, the regions r_i and the hierarchical relations Φ_i.
Initialize $\alpha_{1,m,i} = 1$ for all m, i.
Initialize $w_{1,i}$ for all i using (9)
Algorithm:
for $t = 1$ **to** T **do**
  Observe $\mathbf{s}_t$
  for $m = 1$ **to** M **do**
    Calculate $\gamma_{t,m,i}$ according to (13)
  end for
  for $m = 1$ **to** M **do**
    $p_{t,m} = \gamma_{t,m,1}/w_{t,1}$
  end for
  Select a random arm I_t according to the probability simplex $\mathbf{p}_t = (p_{t,1}, ..., p_{t,M})$
  Set $\alpha_{t+1,m,i} = \alpha_{t,m,i}$ for all m, i
  Set $w_{t+1,i} = w_{t,i}$ for all i
  for the nodes v_i, where $\mathbf{s}_t \in r_i$ **do**
    Calculate $\alpha_{t+1,I_t,i}$ according to (8)
  end for
  for the nodes v_i, where $\mathbf{s}_t \in r_i$ **do**
    Calculate $w_{t+1,i}$ using (9)
  end for
end for

Theorem 2. *Algorithm 1 achieves the regret bound*

$$\mathcal{R}(T, \mathcal{G}^N) \leq \frac{\Psi(A_R+1)\ln((H_S+1)M)}{\eta} + \frac{MT\eta}{2}, \quad (16)$$

where Ψ is an upper bound on the cardinality of the child node groups ϕ, i.e., $\Psi \geq |\phi|$ for all ϕ, H_S is an upper bound on the cardinality of Φ_i, i.e., $H_S \geq |\Phi_i|$ for all i, and A_R is an upper bound on the minimum number of splittings needed in the hierarchical structure to model the optimal partition with R disjoint regions.

Proof of Theorem 2: If the optimal expert is defined over the root node, i.e., $A_R = 0$, its prior weight in the mixture is

$$\beta_{1,j} = \frac{1}{(|\Phi_i|+1)M} \geq \frac{1}{(H_S+1)M}. \quad (17)$$

With each split in the hierarchical structure (i.e., with each move down the hierarchy), the prior weights of the experts are divided by a factor which is at most $(H_S+1)^{\Psi}M^{\Psi-1}$. Thus, in case we need A_R splittings to model the partition corresponding to the optimal expert, its prior weight is

$$\beta_{1,j} \geq (H_S+1)^{-A_R\Psi-1}M^{A_R-A_R\Psi-1}. \quad (18)$$

Since $A_R \geq 1$ and $\Psi \geq 1$, we have

$$\beta_{1,j} \geq (H_S+1)^{-\Psi(A_R+1)}M^{-\Psi(A_R+1)}. \quad (19)$$

Hence,

$$\ln(1/\beta_{1,j}) \leq \Psi(A_R+1)\ln((H_S+1)M). \quad (20)$$

Putting (20) into (6) concludes the proof. ■

Corollary 1. *By setting*

$$\eta = \sqrt{\frac{2\Psi(A_R+1)\ln((H_S+1)M)}{MT}}, \quad (21)$$

we get the regret bound of

$$\mathcal{R}(T, \mathcal{G}^N) \leq \sqrt{0.5\Psi MT(A_R+1)\ln((H_S+1)M)}. \quad (22)$$

We next present several examples of hierarchical structures which can be employed by our algorithm with the introduced mathematical guarantees. Each structure has its own way of encoding the best arm selection policy, i.e., optimal arbitrary mapping. Hence, the proper selection of the hierarchical structure according to the target application leads to a smaller A_R and a better performance, i.e., a regret upper bound vanishing faster in the average loss per round sense, together with the introduced weighting over the corresponding competition class $\mathcal{G}^N$, cf. Section VI as well as the examples below.

B. Example 1: Arbitrary Splitting

If the hierarchical structure is an arbitrary splitting of N leaf nodes into 2 groups, then $\Psi = 2$, $H_S = 2^{N-1} - 1$ and $A_R = M - 1$. Hence, the regret is upper bounded as

$$\begin{aligned}\mathcal{R}(T, \mathcal{G}^N) &\leq \frac{2M\ln(2^{N-1}M)}{\eta} + \frac{MT\eta}{2} \\ &\leq \frac{2MN\ln(M)}{\eta} + \frac{MT\eta}{2}, \end{aligned} \quad (23)$$

where the last inequality uses $2 \leq M$.

C. Example 2: Binary Tree

In binary trees we have $\Psi = 2$ and $H_S = 1$. For a binary tree with N leaf nodes, we need at most $\log_2 N$ splitting to create each new region. Hence, $A_R = (R-1)\log_2 N$. Therefore,

$$\begin{aligned}\mathcal{R}(T, \mathcal{G}^N) &\leq \frac{2((R-1)\log_2 N + 1)\ln(2M)}{\eta} + \frac{MT\eta}{2} \\ &\leq \frac{2R\log_2 N\ln(2M)}{\eta} + \frac{MT\eta}{2}. \end{aligned} \quad (24)$$

D. Example 3: K-ary Tree

If the hierarchical structure is a K-ary tree (for $K = 2$ this becomes a binary tree) with N leaf nodes and depth $D = \log_K N$, then $\Psi = K$, $H_S = 1$ and $A_R = (R-1)\log_K N$. Therefore, we have

$$\begin{aligned}\mathcal{R}(T, \mathcal{G}^N) &\leq \frac{K(1+(R-1)\log_K N)\ln(2M)}{\eta} + \frac{MT\eta}{2} \\ &\leq \frac{KR\log_K N\ln(2M)}{\eta} + \frac{MT\eta}{2}. \end{aligned} \quad (25)$$

E. Example 4: Lexicographical Splitting Graph

In a lexicographal splitting graph with N leaf nodes, we have $\Psi = 2$, $H_S = N - 1$ and $A_R = R - 1$. Hence,

$$\mathcal{R}(T, \mathcal{G}^N) \leq \frac{2R\ln(NM)}{\eta} + \frac{MT\eta}{2}. \quad (26)$$

F. Example 5: K-group Lexicographical Splitting

If the hierarchical structure is a splitting of N sequentially ordered leaf nodes into K groups (when $K = 2$ this structure becomes the lexicographical splitting graph), then $\Psi = K$, $H_S = \binom{N-1}{K-1}$ and $A_R = \lceil \frac{R-1}{K-1} \rceil$. Therefore, the regret upper bound is

$$\begin{aligned}\mathcal{R}(T, \mathcal{G}^N) &\leq \frac{K(\lceil \frac{R-1}{K-1} \rceil + 1) \ln((1 + \binom{N-1}{K-1})M)}{\eta} + \frac{MT\eta}{2} \\ &\leq \frac{K(R+2K)\ln(NM)}{\eta} + \frac{MT\eta}{2}. \end{aligned} \tag{27}$$

G. Example 6: Arbitrary Position Splitting

In this case, for a d-dimensional context space, we have $\Psi = 2$, $H_S = d$ and $A_R = (R-1)\log_2 N$. Therefore,

$$\begin{aligned}\mathcal{R}(T, \mathcal{G}^N) &\leq \frac{2((R-1)\log_2 N + 1)\ln((d+1)M)}{\eta} + \frac{MT\eta}{2} \\ &\leq \frac{2R\log_2 N \ln((d+1)M)}{\eta} + \frac{MT\eta}{2}. \end{aligned} \tag{28}$$

We have successfully achieved a regret bound of $O(\sqrt{MTR \ln N \ln M})$ with proper selection of the learning rate. Note that typically, $N >> R$. Our regret bounds are only logarithmically dependent on N, hence, in soft-O notation, we achieve the minimax optimal regret bound $\tilde{O}(\sqrt{TR})$.

Next and finally, we address the goal of achieving the performance of the best arm selection policy, i.e., the performance of the optimal arbitrary mapping (in the ultimate set $\mathcal{U}$) from the context space to the bandit arms which is not necessarily in the competition class $\mathcal{G}^N$ but can be approximated arbitrarily well and almost perfectly, if desired, by the class by increasing N. The quantization process in our algorithm naturally produces an additive linear-in-time term in our regret against the truly optimal mapping in $\mathcal{U}$. In the following section, we assume that the arm losses are Lipschitz continuous in the context vectors at each specific round. With this assumption, we show that using a uniform quantization of the context space, we can diminish the linear-in-time term in our regret against the optimal mapping in $\mathcal{U}$ by increasing the number of quantization levels N. Hence, we can achieve a performance as close as desired to the performance of the optimal mapping in $\mathcal{U}$.

V. An Efficient Quantization Method to Asymptotically Achieve the Optimal Context Based Arm Selection

Suppose that the context space is the n-dimensional space $S = [0,1]^n$. Using a hierarchical structure with N leaf nodes, our quantization scheme is as follows. We split the context space into $2^{\lfloor (\log_2 N)/n \rfloor + 1}$ equal subspaces along the first $\log_2 N \pmod n$ dimensions (of the total n dimensions), and $2^{\lfloor (\log_2 N)/n \rfloor}$ equal subspaces along the remaining dimensions.

Theorem 3. *Using aforementioned quantization method for our algorithm, if the arm loss functions are Lipschitz continuous with the Lipschitzness constant c, then the difference between the loss corresponding to the best mapping in $\mathcal{G}^N$ and the loss corresponding to the truly optimal mapping (in the ultimate set[9] $\mathcal{U}$ of all possible arbitrary mappings from the context space to the set of bandit arms) is upper bounded by*

$$\frac{2c\sqrt{n}}{\sqrt[n]{N}}. \tag{29}$$

Proof of Theorem 3: Using this quantization method, the subspaces in the finest partition of the context space are n-dimensional cubes with the longest diagonal length equal to

$$\sqrt{\frac{n - (\log_2 N \pmod n)}{(2^{\lfloor \frac{\log_2 N}{n} \rfloor})^2} + \frac{\log_2 N \pmod n}{(2^{\lfloor \frac{\log_2 N}{n} + 1 \rfloor})^2}}. \tag{30}$$

Since $\log_2 N \pmod n \geq 0$, this upper bound is at most equal to

$$\sqrt{\frac{n}{2^{2\lfloor \frac{\log_2 N}{n} \rfloor}}} \leq \frac{2\sqrt{n}}{2^{\frac{\log_2 N}{n}}} = \frac{2\sqrt{n}}{\sqrt[n]{N}}. \tag{31}$$

Since the loss functions are Lipschitz continuous, the difference between the loss corresponding to the truly optimal mapping in $\mathcal{U}$ and the best mapping in $\mathcal{G}^N$ cannot exceed the Lipschitzness constant times the quantized cubes diagonal length, which concludes the proof. ∎

Note that the Lipschitzness assumption does not intervene with the adversarial setting. The loss functions can be quite different in different rounds and as long as they are Lipschitz continuous at each specific round, the assumption holds and our algorithm is competitive against the ultimate set of all possible arbitrary mappings $\mathcal{U}$. In this case, combining (29) with the regret bound in (22) directly concludes the following theorem.

Theorem 4. *Consider a contextual M-armed bandit problem with the context space $S = [0,1]^n$, where the loss functions of the arms are Lipschitz continuous with the constant c at all rounds. If we use a hierarchical structure with N leaf nodes following the quantization scheme described in Section V, the regret of Algorithm 1 against the truly optimal strategy in a T round trial is upper bounded as follows*

$$R(T, \mathcal{U}) \leq \sqrt{\frac{\Psi MT(A_R+1)\ln((H_S+1)M)}{2}} + \frac{2Tc\sqrt{n}}{\sqrt[n]{N}}. \tag{32}$$

We emphasize that we can make the linear-in-time term of the upper bound in (32) as small as desired by growing the hierarchical structure and increasing the number of leaf nodes N, which is equal to the number of quantization levels.

VI. Experiments

In this section, we demonstrate the performance of our algorithm in different scenarios involving both real and synthetic data. We demonstrate the performance of our main algorithm *HSB* with various hierarchical structures including binary tree (*HSB-BT*), lexicograph (*HSB-LG*) and arbitrary position splitting (*HSB-APS*) [33]. We compare the performance of our algorithm against the state-of-the-art adversarial bandit algorithms *EXP3* and *S-EXP3* [16]. In all of the experiments, the parameters of *EXP3* and *S-EXP3* algorithms are set to their optimal values according to their publication [16].

[9] This ultimate set can be non-rigorously considered as $\mathcal{G}^\infty$.

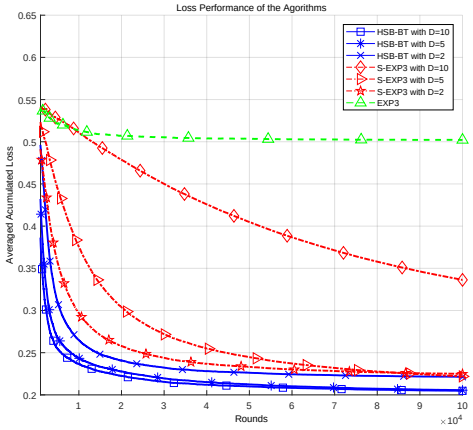


Fig. 5: The averaged accumulated loss of *HSB-BT*, *S-EXP3*, and *EXP3* on the datasets defined using (33).

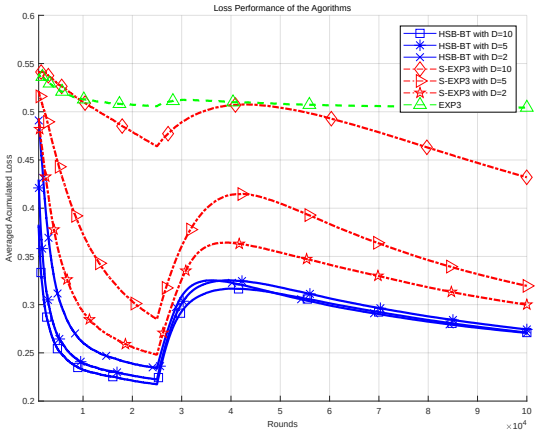


Fig. 6: The averaged accumulated loss of *HSB-BT*, *S-EXP3*, and *EXP3* on the datasets as described in Section VI-B, involving a rapid change in the behavior of the arms after 25% of the rounds.

A. Stationary Environment

We first construct a game with 3-armed bandit, where the context space is the 1-dimensional space $S = [0, 1]$. Each arm i generates its loss according to a Bernoulli distribution with parameter p_i, i.e., the loss is equal to 1 with probability equal to p_i. These parameters, i.e., p_1, p_2, p_3, depend on the context variable s_t as

$$\begin{aligned} p_1(s_t) &= 0.5 + 0.5\sin(2\pi s_t), \\ p_2(s_t) &= \sin(\pi s_t), \\ p_3(s_t) &= s_t. \end{aligned} \tag{33}$$

Here, the optimal strategy is defined as follows

$$g(s_t) = \begin{cases} 3, & s_t < 0.5 \\ 1, & 0.5 \leq s_t < 0.9182 \\ 2, & 0.9182 \leq s_t. \end{cases} \tag{34}$$

In this experiment, we generate the context variable s_t randomly with uniform distribution over the context space, i.e., $[0, 1]$, and compare the averaged cumulated loss performance, i.e., $(\sum_{\tau=1}^{t} l_{\tau,I_\tau})/t$, for our algorithm *HSB-BT* with various depth parameters equal to 2, 5, and 10, *S-EXP3* [16] with the same depth parameters, and *EXP3* [16].

To this end, we generate 10 synthetic datasets of length 10^5. To produce each dataset, first, 10^5 context variables s_t are drawn according to a uniform probability distribution over the interval $[0, 1]$. Then, the arm losses corresponding to different rounds are drawn from the Bernoulli distributions, parameters of which are determined according to (33). Each dataset is presented to the algorithms 10 times and the results are averaged. This process is repeated for all 10 datasets and the ensemble averages are plotted in Fig. 5. Two important results can be derived from the result of this experiment. First, our algorithm *HSB-BT* outperforms both of the *S-EXP3* and *EXP3* algorithms. Second, while increasing the depth uniformly improves the performance of our algorithm, it can degrade the performance of *S-EXP3* due to the overtraining. The superior performance of our algorithm in this experiment is because of its fast convergence to the optimal mapping. Here, *EXP3* has a fast convergence but it converges to a suboptimal mapping because it does not use the context information. On the other hand, *S-EXP3* converges to the optimal mapping, but needs a huge amount of data to get trained. Our algorithm uses an efficient adaptive combination of the experts with intelligent initial weights to obtain the advantages of both *EXP3* and *S-EXP3* algorithms, while mitigating their disadvantages.

B. Nonstationary Environment

In this part, we illustrate the averaged cumulated loss performance of the algorithms in a nonstationary environment. To this end, we construct 10 different datasets of length 10^5 as in Section VI-A. However, here the arm losses follow a model as in (33) in the first quarter of the rounds, and the following model in the rest of the rounds:

$$\begin{aligned} p_1(s_t) &= \sin(\pi s_t), \\ p_2(s_t) &= s_t, \\ p_3(s_t) &= 0.5 + 0.5\sin(2\pi s_t). \end{aligned} \tag{35}$$

Hence, we have an abrupt change in the model of the arms within the rounds. Each dataset is presented to the algorithms 10 times and the results are averaged. This process is repeated for all 10 datasets and the ensemble averages are plotted in Fig. 6. As shown in the figure, our algorithm *HSB-BT* not only outperforms its competitor before the rapid change in the model of the bandit arms but also adopts better to this rapid change in comparison to the competitors.

C. Real Life Online Advertisement Dataset

In this section, we demonstrate the superior performance of our algorithms *HSB-BT* and *HSB-LG* against their natural competitors *EXP3* and *S-EXP3* over the well known real life dataset provided by Yahoo! Research. This dataset contains a user click log for news articles displayed in the featured tab of the Today Module on Yahoo!'s front page, within October 2 to 16, 2011. The dataset contains 28041015 user visits. For each visit, the user is associated with a binary feature

Algorithm 2 The offline evaluation method used to test the competitor algorithms over the Yahoo! Today Module dataset

Input: Bandit algorithm $\mathcal{A}$, logged data for T rounds
Initialize: $L = 0$ and $R = 0$
for $t = 1$ **to** T **do**
  Get $s_t \in \{1, 2, ..., N\}$ from the log
  Run the algorithm $\mathcal{A}$.
  if the arm, selected by $\mathcal{A}$ is the arm which is shown to the user **then**
    Use the user feedback to update $\mathcal{A}$.
    Set $R = R + 1$.
    If the user has not clicked set $L = L + 1$.
  else
    Ignore this round.
  end if
end for
L and R show the total loss and the total rounds respectively.

vector of dimension 136 that contains information about the user like age, gender, behavior targeting features, etc. We used an unbiased offline evaluation method as in [48], to test the competitors over this dataset. A brief pseudo-code of this evaluation method is shown in Algorithm 2. In this experiment, we ran a PCA algorithm [49] over the first 5% of the data to get the principal components of the feature vectors. We mapped the feature vectors over the first principal component to form a set of $1-$dimensional context variables. We used these context variables for *S-EXP3*, *HSB-BT* and *HSB-LG* algorithms. We tested the *EXP3* and *S-EXP3* algorithms with several depth parameters, while their parameters were set to their optimum values [16]. However, since we do not have any information about the number of disjoint regions in the optimal mapping, i.e., R, the η parameter for the *HSB-BT* and *HSB-LG* algorithms cannot be tuned to the optimum value analytically. In this experiment, in order to have a fair comparison, we set the η parameter of the *HSB-BT* and *HSB-LG* algorithm with a specific depth equal to the η parameter of the *S-EXP3* algorithm with the same depth. We emphasize that no numerical optimization is done for the η parameter of our algorithms. The percentage of user clicks for different algorithms are shown in Fig. 7. As shown in this table, our algorithms outperform both of the *S-EXP3* and *EXP3* algorithms, even though the learning rate parameters of our algorithms are not tuned to the optimum values due to the lack of knowledge on the parameter R.

D. Real Life Classification Dataset

In this experiment, we use well-known LandSat dataset [50] to show how our algorithm can be employed for online multi-class classification in the Error Correcting Output Codes (ECOC) framework [51]. This dataset consists of 6435 samples from 6 classes. The feature vectors are 36-dimensional integer vectors.

In the ECOC framework, given a set of C classes, we assign a binary codeword of length N_C to each one of the classes. We arrange these codewords as rows of a coding

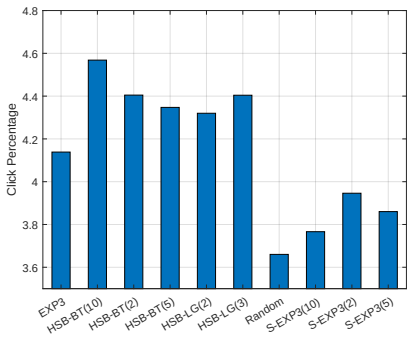


Fig. 7: Percentage of click in the Yahoo! Today Module dataset

matrix $M_C \in \{+1, -1\}^{C \times N_C}$. We consider each one of the N_C columns of M_C as a binary classification problem and run a binary classifier over each column. The i^{th} classifier is to learn whether the i^{th} bit of the codeword is $+1$ or -1. In order to label a new sample, the feature vector is fed to the binary classifiers to obtain a codeword based on their outputs. We then decide on the label of the sample based on its codeword.

In this experiment, we use the one-versus-all coding [51] to form our coding matrix as shown in table 2 and run 6 Online Perceptrons in parallel as our binary classifiers. We use the codewords obtained from the Perceptrons as our context vectors and the classes as our bandit arms. We provide our algorithm HSB with the context vectors and label the sample based on the arm suggested by the algorithm. Then, we observe the true label and suffer a loss equal to 1 in case of incorrect label. The competitors in this experiment are our algorithm HSB with two different hierarchical structures of "Arbitrary Position Splitting" (*HSB-APS*) and "Binary Tree" (*HSB-BT*), alongside *EXP3*, *S-EXP3* and *Hamming Decoding* [51]. The learning parameters of the algorithms are set to their optimal value.

We emphasize that while the *Hamming Decoder* knows the codewords corresponding the classes a priori, other competitors do not use this information and try to learn the best mapping from the context space, i.e., codewords space, to the classes. For presentation simplicity, we have splitted the samples into 9 consecutive epochs and averaged the number of errors over each epoch. As shown in Figure 8, the algorithms *S-EXP3*, *HSB-BT* and *HSB-APS* compensate their lack of information on the coding matrix (compared to the Hamming Decoder) as time goes on. Among them, *HSB-APS* outperforms the others and even *Hamming Decoder* in the last 3 epochs as expected.

VII. Concluding Remarks

We studied the contextual multi-armed bandit problem in an adversarial setting and introduced a truly online and low complexity algorithm that asymptotically achieves the performance of the best context dependent bandit arm selection policy. Our core algorithm quantizes the space of the context vectors into a large number of disjoint regions using an efficient quantization method and forms the class of all mappings from these regions to the bandit arms. Then, it adaptively combines these mappings in a mixture-of-experts setting and achieves

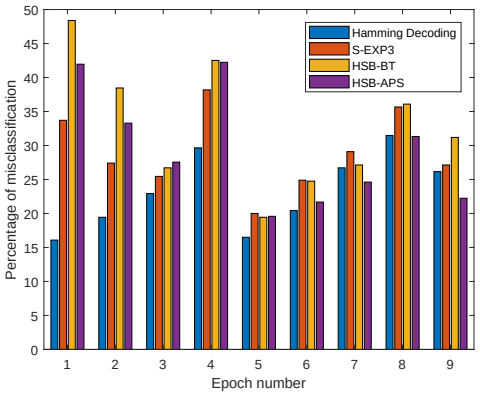


Fig. 8: The percentage of misclassification of the competitors over 9 consecutive epochs of length 715.

the performance of the best mapping in the class. We prove performance upper bounds for the introduced algorithm. These upper bounds show that we achieve the performance of the truly optimal mapping (which might be out of our class of mappings) by increasing the number of quantization levels. We use hierarchical structures to implement our algorithm in an efficient way such that the computational complexity is log-linear in the number of quantization levels. We have no statistical assumptions on the behavior of the context vectors and the bandit arms, hence our results are guaranteed to hold in an individual sequence manner. Through extensive set of experiments involving synthetic and real data, we demonstrate the significant performance gains achieved by the proposed algorithm in comparison to the state-of-the-art techniques.

Appendix A
Proof of Theorem 1

From the definition, denoting the mapping followed by the j^{th} expert by $g_j(.)$, we have

$$\mathcal{R}(T,E_j)=\mathbb{E}\left[\sum_{t=1}^{T} l_{t,I_t}-\sum_{t=1}^{T} l_{t,g_j(\mathbf{s}_t)}\right] \tag{36}$$

Here, l_{t,I_t} can be expanded as

$$\begin{aligned} l_{t,I_t} &= \mathbb{E}_{j\sim\boldsymbol{\beta}_t}\tilde{l}_{t,g_j(\mathbf{s}_t)} \\ &= \frac{1}{\eta}\left(\ln\left(\mathbb{E}_{j\sim\boldsymbol{\beta}_t}e^{-\eta\tilde{l}_{t,g_j(\mathbf{s}_t)}}\right)+\eta\mathbb{E}_{j\sim\boldsymbol{\beta}_t}\tilde{l}_{t,g_j(\mathbf{s}_t)}\right) \\ &\quad -\frac{1}{\eta}\ln\mathbb{E}_{j\sim\boldsymbol{\beta}_t}e^{-\eta\tilde{l}_{t,g_j(\mathbf{s}_t)}}. \end{aligned} \tag{37}$$

The first term in (37) can be bounded using the inequalities $\ln x\le x-1$ and $\exp(-x)-1+x\le x^2/2$, for all $x\ge 0$, as

$$\begin{aligned} &\ln\left(\mathbb{E}_{j\sim\boldsymbol{\beta}_t}e^{-\eta\tilde{l}_{t,g_j(\mathbf{s}_t)}}\right)+\eta\mathbb{E}_{j\sim\boldsymbol{\beta}_t}\tilde{l}_{t,g_j(\mathbf{s}_t)} \\ &\le \mathbb{E}_{j\sim\boldsymbol{\beta}_t}\left[e^{-\eta\tilde{l}_{t,g_j(\mathbf{s}_t)}}-1+\eta\tilde{l}_{t,g_j(\mathbf{s}_t)}\right] \\ &\le \mathbb{E}_{j\sim\boldsymbol{\beta}_t}\frac{\eta^2\tilde{l}^2_{t,g_j(\mathbf{s}_t)}}{2}=\frac{\eta^2 l^2_{t,I_t}}{2p_{t,I_t}}\le\frac{\eta^2}{2p_{t,I_t}}. \end{aligned} \tag{38}$$

In order to bound the second term in (37), we just rewrite the expectation using (4) as follows. For $t=1$, we have

$$-\frac{1}{\eta}\ln\mathbb{E}_{j\sim\boldsymbol{\beta}_1}e^{-\eta\tilde{l}_{1,g_j(\mathbf{s}_1)}}=-\frac{1}{\eta}\ln\frac{\sum_{j=1}^{M^N}\alpha_{1,j}e^{-\eta\tilde{l}_{1,g_j(\mathbf{s}_1)}}}{\sum_{j=1}^{M^N}\alpha_{1,j}}, \tag{39}$$

and for $t\ge 2$, we have

$$\begin{aligned} &-\frac{1}{\eta}\ln\mathbb{E}_{j\sim\boldsymbol{\beta}_t}e^{-\eta\tilde{l}_{t,g_j(\mathbf{s}_t)}} \\ &\qquad=-\frac{1}{\eta}\ln\frac{\sum_{j=1}^{M^N}\alpha_{1,j}e^{-\eta\sum_{\tau=1}^{t}\tilde{l}_{\tau,g_j(\mathbf{s}_\tau)}}}{\sum_{j=1}^{M^N}\alpha_{1,j}e^{-\eta\sum_{\tau=1}^{t-1}\tilde{l}_{\tau,g_j(\mathbf{s}_\tau)}}}. \end{aligned} \tag{40}$$

Putting the bounds in (38) and (40) into (37), we have

$$\begin{aligned} \sum_{t=1}^{T} l_{t,I_t} &\le -\frac{1}{\eta}\Big(\sum_{t=2}^{T}\ln\frac{\sum_{j=1}^{M^N}\alpha_{1,j}e^{-\eta\sum_{\tau=1}^{t}\tilde{l}_{\tau,g_j(\mathbf{s}_\tau)}}}{\sum_{j=1}^{M^N}\alpha_{1,j}e^{-\eta\sum_{\tau=1}^{t-1}\tilde{l}_{\tau,g_j(\mathbf{s}_\tau)}}} \\ &\quad+\ln\frac{\sum_{j=1}^{M^N}\alpha_{1,j}e^{-\eta\tilde{l}_{1,g_j(\mathbf{s}_1)}}}{\sum_{j=1}^{M^N}\alpha_{1,j}}\Big)+\frac{\eta T}{2p_{t,I_t}}. \end{aligned} \tag{41}$$

Opening the first two term in (41), we have

$$\begin{aligned} \sum_{t=1}^{T} l_{t,I_t} &\le -\frac{1}{\eta}\ln\sum_{j=1}^{M^N}\alpha_{1,j}e^{-\eta\sum_{\tau=1}^{T}\tilde{l}_{\tau,g_j(\mathbf{s}_\tau)}} \\ &\quad+\frac{1}{\eta}\ln\sum_{j=1}^{M^N}\alpha_{1,j}+\frac{\eta T}{2p_{t,I_t}}. \end{aligned} \tag{42}$$

Since $\sum_{j=1}^{M^N}\alpha_{1,j}e^{-\eta\sum_{\tau=1}^{T}\tilde{l}_{\tau,g_j(\mathbf{s}_\tau)}}\le\alpha_{1,j}e^{-\eta\sum_{\tau=1}^{T}\tilde{l}_{\tau,g_j(\mathbf{s}_\tau)}}$, we have

$$\begin{aligned} \sum_{t=1}^{T} l_{t,I_t} &\le -\frac{1}{\eta}\ln\alpha_{1,j}+\sum_{\tau=1}^{T}\tilde{l}_{\tau,g_j(\mathbf{s}_\tau)} \\ &\quad+\frac{1}{\eta}\ln\sum_{j=1}^{M^N}\alpha_{1,j}+\frac{\eta T}{2p_{t,I_t}} \\ &=\frac{\ln 1/\beta_{1,j}}{\eta}+\frac{\eta T}{2p_{t,I_t}}+\sum_{\tau=1}^{T}\tilde{l}_{\tau,g_j(\mathbf{s}_\tau)}. \end{aligned} \tag{43}$$

Taking expectation from both sides (with respect to $I_t\sim\mathbf{p}_t$) and substituting $\mathbb{E}[\tilde{l}_{\tau,g_j(\mathbf{s}_\tau)}]=l_{\tau,g_j(\mathbf{s}_\tau)}$ and $\mathbb{E}[\frac{1}{p_{t,I_t}}]=M$ into the result concludes the proof.

Appendix B
Proof of Proposition 1

We prove this proposition using induction. For leaf nodes where $\Phi_i=\emptyset$, we have

$$w_{t,i}=\frac{1}{M}\sum_{m=1}^{M}\alpha_{t,m,i}. \tag{44}$$

From the definition of $\alpha_{t,m,i}$ in (7) we have

$$w_{t,i}=\sum_{m=1}^{M}\frac{1}{M}\exp(-\eta\sum_{\substack{\tau<t\\ \mathbf{s}_\tau\in r_i}}\tilde{l}_{\tau,m})=\sum_{k\in\Gamma_i}\alpha_{t,k}, \tag{45}$$

where $\alpha_{1,k}=1/M$ for all $k\in\Gamma_i$.

Consider the node v_i. Suppose $\forall \phi \in \Phi_i, \forall j \in \phi$ we have

$$w_{t,j} = \sum_{k \in \Gamma_j} \alpha_{t,k}. \tag{46}$$

It suffices to show that

$$w_{t,i} = \sum_{k \in \Gamma_i} \alpha_{t,k}. \tag{47}$$

The set of experts defined over v_i, i.e., Γ_i, can be decomposed into the following subsets:

- Γ_i^o : The set of experts, which map the whole context space into a fixed arm. This set contains M experts.
- Γ_i^ϕ, $\phi \in \Phi_i$: The set of experts, which partition the context space into the regions r_j, $j \in \phi$, and follow a specific expert over each node $j \in \phi$, based on the observed $\mathbf{s}_t$. If $\mathbf{s}_t \in r_j$, the experts in Γ_i^ϕ follow the experts in Γ_j. This set contains $\prod_{j \in \phi} |\Gamma_j|$ experts. Each experts in Γ_i^ϕ can be represented by a vector of experts $\mathbf{k}_\phi \in \prod_{j \in \phi} \Gamma_j$, where $\mathbf{k}_\phi(j)$ is an expert defined over node j.

We emphasize that even though we have

$$\Gamma_i^o \cup (\bigcup_{\phi \in \Phi_i} \Gamma_i^\phi) = \Gamma_i, \tag{48}$$

the intersection of any two of these $|\Phi_i| + 1$ subsets is not empty necessarily. In particular, the M experts in Γ_i^o are also included among the elements of Γ_i^ϕ for all $\phi \in \Phi_i$. In fact, each expert in Γ_i^o can be seen as an expert which partitions the context space into r_j's for $j \in \phi$, and follows the experts which select a fixed arm m over all the nodes v_j's.

We have

$$\prod_{j \in \phi} w_{t,j} = \prod_{j \in \phi} \left(\sum_{k \in \Gamma_j} \alpha_{t,k} \right) = \sum_{\mathbf{k}_\phi \in \prod_{j \in \phi} \Gamma_j} \left(\prod_j \alpha_{t,\mathbf{k}_\phi(j)} \right). \tag{49}$$

We open the product term as

$$\begin{aligned} &\prod_j \alpha_{t,\mathbf{k}_\phi(j)} \\ &= \prod_j \alpha_{1,\mathbf{k}_\phi(j)} \exp\left(-\eta \sum_{\tau<t} \sum_j \tilde{l}_{\tau, g_{\mathbf{k}_{\phi(j)}}(\mathbf{s}_\tau)} \mathbf{1}_{\{\mathbf{s}_\tau \in r_j\}} \right) \\ &= \prod_j \alpha_{1,\mathbf{k}_\phi(j)} \exp\left(-\eta \sum_{\substack{\tau<t \\ \mathbf{s}_\tau \in r_i}} \tilde{l}_{\tau, g_{\mathbf{k}_\phi}(\mathbf{s}_\tau)} \right). \end{aligned} \tag{50}$$

Putting (50) into (9) we get

$$\begin{aligned} w_{t,i} &= \frac{1}{(|\Phi_i|+1)M} \sum_{k \in \Gamma_i^o} \alpha_{t,k} \\ &+ \frac{1}{|\Phi_i|+1} \sum_{\phi \in \Phi_i} \left(\sum_{\mathbf{k}_\phi \in \prod_{j \in \phi} \Gamma_j} \alpha_{1,\mathbf{k}_\phi} \exp\left(-\eta \sum_{\substack{\tau<t \\ \mathbf{s}_\tau \in r_i}} \tilde{l}_{\tau, g_{\mathbf{k}_\phi(\mathbf{s}_\tau)}} \right) \right) \\ &= \frac{1}{(|\Phi_i|+1)M} \sum_{k \in \Gamma_i^o} \alpha_{t,k} + \frac{1}{(|\Phi_i|+1)} \sum_{\phi \in \Phi_i} \sum_{k \in \Gamma_i^\phi} \alpha_{t,k} = \sum_{k \in \Gamma_i} \alpha_{t,k}, \end{aligned} \tag{51}$$

where

$$\begin{aligned} \alpha_{1,k} =& \frac{1}{(|\Phi_i|+1)M} \mathbf{1}_{\{k \in \Gamma_i^o\}} \\ &+ \frac{1}{|\Phi_i|+1} \sum_{\phi \in \Phi_i} \left(\mathbf{1}_{\{k = \mathbf{k}_\phi\}} \prod_{j \in \phi} \alpha_{1,\mathbf{k}_\phi(j)} \right). \end{aligned} \tag{52}$$

Appendix C
Proof of Proposition 2

Consider a specific bandit arm m^*. Given the context vector $\mathbf{s}_t$, for all $m \in \{1, 2, .., M\}$, for all nodes v_i in the hierarchy, we define the variables $\tilde{\alpha}_{t,m,i}$ as

$$\tilde{\alpha}_{t,m,i} = \begin{cases} 0, & \mathbf{s}_t \in r_i, m \neq m^* \\ \alpha_{t,m,i}, & \text{otherwise} \end{cases}. \tag{53}$$

Now, from the definition of $\gamma_{t,m,i}$ in (13), we have

$$\begin{aligned} \gamma_{t,m^*,i} =& \frac{1}{(|\Phi_i|+1)M} \sum_{m=1}^{M} \tilde{\alpha}_{t,m,i} \\ &+ \frac{1}{(|\Phi_i|+1)} \sum_{\phi \in \Phi_i} \left(\prod_{j \in \phi} \tilde{w}_{t,j} \right). \end{aligned} \tag{54}$$

The exact same lines of the proof of Theorem 1 hold to show that

$$\tilde{w}_{t,i} = \sum_{k \in \Gamma_i} \tilde{\alpha}_{t,k}, \tag{55}$$

where

$$\tilde{\alpha}_{t,k} = \begin{cases} \alpha_{t,k}, & g_k(\mathbf{s}_t) = m^* \\ 0, & \text{otherwise} \end{cases}. \tag{56}$$

Hence, (15) holds.

References

[1] J. Lin and D. X. Zhou, "Online learning algorithms can converge comparably fast as batch learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–12, 2017.

[2] L. Jian, S. Shen, J. Li, X. Liang, and L. Li, "Budget online learning algorithm for least squares svm," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 9, pp. 2076–2087, Sept 2017.

[3] A. Rakotomamonjy, S. Koo, and L. Ralaivola, "Greedy methods, randomization approaches, and multiarm bandit algorithms for efficient sparsity-constrained optimization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 11, pp. 2789–2802, Nov 2017.

[4] J. Peng, A. J. Aved, G. Seetharaman, and K. Palaniappan, "Multiview boosting with information propagation for classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–13, 2017.

[5] G. Ditzler, R. Polikar, and G. Rosen, "A sequential learning approach for scaling up filter-based feature subset selection," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–15, 2017.

[6] R. J. Meyer and Y. Shi, "Sequential choice under ambiguity: Intuitive solutions to the armed-bandit problem," *Management Science*, vol. 41, no. 5, pp. 817–834, 1995.

[7] S. Shalev-Shwartz, "Online learning and online convex optimization," *Found. Trends Mach. Learn.*, vol. 4, no. 2, pp. 107–194, Feb. 2012.

[8] N. Cesa-Bianchi and G. Lugosi, "Combinatorial bandits," *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1404 – 1422, 2012, {JCSS} Special Issue: Cloud Computing 2011.

[9] S. E. Yuksel, J. N. Wilson, and P. D. Gader, "Twenty years of mixture of experts," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1177–1193, Aug 2012.

[10] H. Ozkan, M. A. Donmez, S. Tunc, and S. S. Kozat, "A deterministic analysis of an online convex mixture of experts algorithm," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 7, pp. 1575–1580, July 2015.

[11] A. J. Bean and A. C. Singer, "Universal switching and side information portfolios under transaction costs using factor graphs," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 4, pp. 351–365, Aug 2012.

[12] A. C. Singer, S. S. Kozat, and M. Feder, "Universal linear least squares prediction: upper and lower bounds," *IEEE Transactions on Information Theory*, vol. 48, no. 8, pp. 2354–2362, Aug 2002.

[13] A. C. Singer and M. Feder, "Universal linear prediction by model order weighting," *IEEE Transactions on Signal Processing*, vol. 47, no. 10, pp. 2685–2699, Oct 1999.

[14] T. Moon and T. Weissman, "Universal fir mmse filtering," *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 1068–1083, March 2009.

[15] T. Mannucci, E. J. van Kampen, C. de Visser, and Q. Chu, "Safe exploration algorithms for reinforcement learning controllers," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–13, 2017.

[16] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *CoRR*, vol. abs/1204.5721, 2012.

[17] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proceedings of the 19th International Conference on World Wide Web*, New York, NY, USA, 2010, WWW '10, pp. 661–670, ACM.

[18] C. Tekin, S. Zhang, and M. van der Schaar, "Distributed online learning in social recommender systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 4, pp. 638–652, Aug 2014.

[19] L. Tang, Y. Jiang, L. Li, and T. Li, "Ensemble contextual bandits for personalized recommendation," in *Proceedings of the 8th ACM Conference on Recommender Systems*, New York, NY, USA, 2014, RecSys '14, pp. 73–80, ACM.

[20] X. Luo, M. Zhou, S. Li, Z. You, Y. Xia, and Q. Zhu, "A nonnegative latent factor model for large-scale sparse matrices in recommender systems via alternating direction method," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 3, pp. 579–592, March 2016.

[21] J. P. Hardwick and Q. F. Stout, "Bandit strategies for ethical sequential allocation," *Comp. Sci. and Statist*, pp. 421–424, 1991.

[22] Y. Gai, B. Krishnamachari, and R. Jain, "Learning multiuser channel allocations in cognitive radio networks: A combinatorial multi-armed bandit formulation," in *New Frontiers in Dynamic Spectrum, 2010 IEEE Symposium on*, April 2010, pp. 1–9.

[23] L. Lai, H. Jiang, and H. V. Poor, "Medium access in cognitive radio networks: A competitive multi-armed bandit framework," in *2008 42nd Asilomar Conference on Signals, Systems and Computers*, Oct 2008, pp. 98–102.

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

[25] T. L. Lai, P. W. Lavori, and K. W. Tsang, "Adaptive design of confirmatory trials: Advances and challenges," *Contemporary Clinical Trials*, vol. 45, Part A, pp. 93 – 102, 2015, 10th Anniversary Special Issue.

[26] K. Liu and Q. Zhao, "Distributed learning in multi-armed bandit with multiple players," *IEEE Transactions on Signal Processing*, vol. 58, no. 11, pp. 5667–5681, Nov 2010.

[27] M. Tokic, "Adaptive ϵ-greedy exploration in reinforcement learning based on value differences," in *Berlin / Heidelberg*. 2010, pp. 203–210, Springer.

[28] O. Chapelle and L. Li, "An empirical evaluation of thompson sampling," in *Advances in neural information processing systems*, 2011, pp. 2249–2257.

[29] J. Langford and T. Zhang, "The epoch-greedy algorithm for multi-armed bandits with side information," in *Advances in neural information processing systems*, 2008, pp. 817–824.

[30] S. Rota Bul, B. Biggio, I. Pillai, M. Pelillo, and F. Roli, "Randomized prediction games for adversarial machine learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 11, pp. 2466–2478, Nov 2017.

[31] L. Tang, R. Rosales, A. Singh, and D. Agarwal, "Automatic ad format selection via contextual bandits," in *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, New York, NY, USA, 2013, CIKM '13, pp. 1587–1594, ACM.

[32] P. Auer, N. Cesa-bianchi, Y. Freund, and R. E. Schapire, "The non-stochastic multiarmed bandit problem," *SIAM Journal on Computing*, vol. 32, pp. 2002, 2002.

[33] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens, "Context weighting for general finite-context sources," *IEEE Transactions on Information Theory*, vol. 42, no. 5, pp. 1514–1520, Sep 1996.

[34] A. Agarwal, D. Hsu, S. Kale, J. Langford, L. Li, and R. E. Schapire, "Taming the monster: A fast and simple algorithm for contextual bandits," *CoRR*, vol. abs/1402.0555, 2014.

[35] M. Dudik, D. Hsu, S. Kale, N. Karampatziakis, J. Langford, L. Reyzin, and T. Zhang, "Efficient optimal learning for contextual bandits," in *Proceedings of the Twenty-Seventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-11)*, Corvallis, Oregon, 2011, pp. 169–178, AUAI Press.

[36] P. Auer, "Using confidence bounds for exploitation-exploration trade-offs," *Journal of Machine Learning Research*, vol. 3, no. Nov, pp. 397–422, 2002.

[37] A. Agarwal, M. Dudik, S. Kale, J. Langford, and R. E. Schapire, "Contextual bandit learning with predictable rewards.," in *AISTATS*, 2012, pp. 19–26.

[38] V. Syrgkanis, A. Krishnamurthy, and R. E. Schapire, "Efficient algorithms for adversarial contextual learning," in *Proceedings of The 33rd International Conference on Machine Learning*, Maria Florina Balcan and Kilian Q. Weinberger, Eds., New York, New York, USA, 20–22 Jun 2016, vol. 48 of *Proceedings of Machine Learning Research*, pp. 2159–2168, PMLR.

[39] F. Willems, Y. M. Shtarkov, and T. J. Tjalkens, "The context-tree weighting method: basic properties," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 653–664, 1995.

[40] K. Sadakane, T. Okazaki, and H. Imai, "Implementing the context tree weighting method for text compression," in *Data Compression Conference, 2000. Proceedings. DCC 2000.* IEEE, 2000, pp. 123–132.

[41] I. Csiszár and Z. Talata, "Context tree estimation for not necessarily finite memory processes, via bic and mdl," *IEEE Transactions on Information theory*, vol. 52, no. 3, pp. 1007–1016, 2006.

[42] T. Dumont, "Context tree estimation in variable length hidden markov models," *IEEE Transactions on Information Theory*, vol. 60, no. 6, pp. 3196–3208, June 2014.

[43] F. Babich, O. E. Kelly, and G. Lombardi, "A context-tree based model for quantized fading," *IEEE communications letters*, vol. 3, no. 2, pp. 46–48, 1999.

[44] S. S. Kozat, A. C. Singer, and G. C. Zeitler, "Universal piecewise linear prediction via context trees," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3730–3745, 2007.

[45] N. D. Vanli and S. S. Kozat, "A comprehensive approach to universal piecewise nonlinear regression based on trees," *IEEE Transactions on Signal Processing*, vol. 62, no. 20, pp. 5471–5486, 2014.

[46] H. Ozkan, N. D. Vanli, and S. S. Kozat, "Online classification via self-organizing space partitioning," *IEEE Transactions on Signal Processing*, vol. 64, no. 15, pp. 3895–3908, Aug 2016.

[47] Ling Huang, Anthony D. Joseph, Blaine Nelson, Benjamin I.P. Rubinstein, and J. D. Tygar, "Adversarial machine learning," in *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*, New York, NY, USA, 2011, AISec '11, pp. 43–58, ACM.

[48] L. Li, W. Chu, J. Langford, and X. Wang, *Unbiased Offline Evaluation of Contextual-bandit-based News Article Recommendation Algorithms*, WSDM '11. ACM, New York, NY, USA, 2011.

[49] I. Jolliffe, *Principal component analysis*, Wiley Online Library, 2002.

[50] D. Michie, D. J. Spiegelhalter, C. C. Taylor, and J. Campbell, Eds., *Machine Learning, Neural and Statistical Classification*, Ellis Horwood, Upper Saddle River, NJ, USA, 1994.

[51] S. Escalera, O. Pujol, and P. Radeva, "Error-correcting ouput codes library," *J. Mach. Learn. Res.*, vol. 11, pp. 661–664, Mar. 2010.