Multi-Feedback Bandit Learning with Probabilistic Contexts

Luting Yang* , **Jianyi Yang*** and **Shaolei Ren**
University of California, Riverside
{lyang029, jyang239, shaolei}@ucr.edu

Abstract

Contextual bandit is a classic multi-armed bandit setting, where side information (i.e., context) is available before arm selection. A standard assumption is that exact contexts are perfectly known prior to arm selection and only single feedback is returned. In this work, we focus on multi-feedback bandit learning with probabilistic contexts, where a bundle of contexts are revealed to the agent along with their corresponding probabilities at the beginning of each round. This models such scenarios as where contexts are drawn from the probability output of a neural network and the reward function is jointly determined by multiple feedback signals. We propose a kernelized learning algorithm based on upper confidence bound to choose the optimal arm in reproducing kernel Hilbert space for each context bundle. Moreover, we theoretically establish an upper bound on the cumulative regret with respect to an oracle that knows the optimal arm given probabilistic contexts, and show that the bound grows sublinearly with time. Our simulation on machine learning model recommendation further validates the sub-linearity of our cumulative regret and demonstrates that our algorithm outperforms the approach that selects arms based on the most probable context.

1 Introduction

Multi-armed bandit (MAB) is a crucial online learning problem to discover optimal decisions (a.k.a. arms) based on received feedback signals over time [Lai and Robbins, 1985]. Importantly, contextual bandit learning extends the standard MAB setting by allowing the learner/agent to access some side information (i.e., context) about the environment prior to arm selection [Lin *et al.*, 2018]. For contextual bandit, the context and selected arm jointly determine the distribution of reward received by the agent, and the goal of the agent is to maximize its cumulative reward by gradually identifying the optimal mapping of context information into actions based on the history of context-action-feedback.

Contextual bandits have found success in many applications, including online recommendation [Mary *et al.*, 2015], commercial advertising [Tang *et al.*, 2013] and medical experiment design [Villar *et al.*, 2015]. Subsequently, efficient learning algorithms like Lin-UCB [Li *et al.*, 2010], EXP4 [Auer *et al.*, 2002] and their variations [Li *et al.*, 2017] have drawn great attention. Nonetheless, most of the prior studies assume that the context information acquired by the agent before arm selection is perfect. While this assumption facilitates performance analysis of the proposed algorithms, it may fail in certain practical scenarios, where there is randomness and uncertainty about the context information.

To alleviate the uncertainty and randomness from environment, one can apply classification techniques, such as support vector machine (SVM) [Quinlan, 1986] and deep neural networks (DNN) [Wan, 1990], which yield a probability distribution over possible candidate category of contexts. For example, a recommendation system commonly recommends personalized items given user features (i.e., contexts) predicted by a neural network classifier. Thus, all the possible candidate contexts together form a context bundle with a probability distribution of different contexts, and the exact context is included in the bundle (possibly not having the greatest probability) but unknown to the agent. In this paper, we also use "context" and "context candidate" exchangeably.

In addition to the lack of exact context information, another practical consideration for contextual bandit learning is that the agent can receive multiple feedbacks instead of a single one. In this case, the goal of the agent is to maximize its reward modeled as a (possibly time-varying) utility function jointly determined by multiple feedbacks rather than any of the individual feedback. For example, when selecting an app for a mobile device, both energy and latency can be measured and reported to the learner/agent, and these metrics jointly affect the performance of the selected app.

Motivated by the aforementioned practical considerations, the focus of this work is to study a novel contextual bandit setting where the agent can only access to a probabilistic context bundle for arm selection and its goal is to maximize a time-varying utility function jointly determined by the multiple feedback signals received at the end of each round. In order to design an efficient learning algorithm, the key is how the agent leverages the available probabilistic context information to learn multiple feedback functions that jointly de-

*Equal contribution.

termine a reward. To study this problem, we consider a general setting where each individual feedback function can be nonlinear with respect to the selected arm and contexts, and apply the kernel method to transfer feedback function in the reproducing kernel Hilbert space (RKHS). We design a new algorithm by extending upper confidence bound (UCB) techniques to account for the probabilistic context information, using the expectation of reward over the probabilistic context distribution. For each feedback, we learn its relation with the selected arm given a probabilistic context bundle. Then, an arm is selected based on an estimated reward in terms of all the estimated feedback values. Importantly, we prove that our algorithm achieves a sub-linear regret upper bound $\mathcal{O}(\sqrt{T\log(T)})$ compared to an oracle that knows the optimal arm given any probabilistic context bundle.

We apply our learning algorithm to the problem of deep neural network (DNN) model recommendation for edge inference on mobile devices. Our experiments show that our proposed algorithm outperforms the alternative solution that selects arms based on the most probable context. More importantly, our algorithm yields a sub-linear regret with respect to the oracle, demonstrating the effectiveness of our algorithm and validating the regret analysis.

2 Problem Formulation

In a standard contextual bandit setting, the context $x_t \in \mathcal{R}^M$ is available to the agent at each round. In many cases, however, the agent only has the knowledge of a bundle of context candidates $\mathcal{X} = \left\{x^1, \cdots, x^N\right\}$, where $|\mathcal{X}| = N$, and the true context x_t is in the bundle. At each round t, with some prior knowledge, the agent can get a collection of probabilities for context candidates, i.e. $Pr_t(\mathcal{X}) = \left\{P_t(x^1), \cdots, P_t(x^N)\right\}$ and $\sum_{i=1}^{N} P_t(x^i) = 1$. This can be done by using, for example, a well-trained DNN classifier and extracting the softmax layer output of the classifier. The extracted probabilities define a probability space over the context bundle $\mathcal{X}$. Now, the context at round t is a random variable X_t in the probability space with the probability measure $Pr\left(X_t = x_t^i\right) = P_t\left(x_t^i\right)$. Note that, with a notational change, our model can also be extended to continuous context with a probability density function. Additionally, we consider a more general case where the agent receives multiple feedbacks. The jth feedback ($j = 1, \cdots, J$) with respect to action $a \in \mathcal{A}$ and the random context X_t can be expressed as

$$f_{a,t}^j = g_a^j(X_t) + \epsilon^j \tag{1}$$

where $g_a^j(\cdot)$ is a deterministic feedback function which can be linear or nonlinear, and ϵ^j is zero-mean Gaussian noise and ϵ^i and ϵ^j are mutually independent for $i \neq j$. Assume that for $j = 1, \cdots, J$, a kernel function $k^j : \mathbb{R}^M \times \mathbb{R}^M \to \mathbb{R}$ can be found to represent $g_a^j(\cdot)$ in RKHS $\mathcal{F}^j$. In other words, the kernel function k^j corresponds to a feature map $\phi^j : \mathbb{R}^n \to \mathcal{F}^j$ which satisfies $k^j(x, x') = \phi^j(x)^\top \phi^j(x'), \forall x, x' \in \mathbb{R}^M$, and $g_a^j(x) = \phi^j(x)^\top \theta_a^j$.

The agent's reward is evaluated by a utility function $U_t : \mathcal{R}^J \to \mathcal{R}$, which may change over time and is known to the agent. Assuming that the Lipschitz constant of the utility function U_t is L_t and $L = \max_t L_t$, we have

$$|U_t(\mathbf{f}_1 - \mathbf{f}_2)| \leq L \|\mathbf{f}_1 - \mathbf{f}_2\|. \tag{2}$$

If an action a is selected, then a reward $U_t(\mathbf{f}_{a,t})$ is obtained by the agent where $\mathbf{f}_{a,t} = \left[f_{a,t}^1, \cdots, f_{a,t}^J\right]$ is the feedback vector. By selecting actions, we seek to maximize the expected reward over both the probabilistic context space and the noise space, which is denoted as $\mathbb{E}[U_t(\mathbf{f}_{a,t})]$. For the convenience of analysis, we further assume that $\mathbb{E}_\epsilon[U_t(\mathbf{f}_{a,t})] = U_t(\mathbb{E}_\epsilon[\mathbf{f}_{a,t}])$ where the expectation $\mathbb{E}_\epsilon[\cdot]$ is taken over the noise space. Example utility functions include a linear form $U_t(\mathbf{f}) = \mathbf{u}_t^\top \mathbf{f}$ or a multiplication form $U_t(\mathbf{f}) = \prod_{j=1}^J f^j$, which are also common functions used in multi-objective bandits [Roijers *et al.*, 2017; Yahyaa and Manderick, 2015].

The best action at round t is defined as the action that leads to the highest expected reward, i.e.

$$a_t^* = \arg\max_{a \in \mathcal{A}} \mathbb{E}[U_t(\mathbf{f}_{a,t})] \tag{3}$$

where the expectation $\mathbb{E}[\cdot]$ is taken over both noise space and context space. This best action oracle is reasonable and common for the cases with context uncertainty, and also considered as a benchmark in [Kirschner and Krause, 2019; Yun *et al.*, 2017]. With this oracle, the expected instant regret reg_t at every round can be expressed as

$$reg_t = \mathbb{E}\left[U_t(\mathbf{f}_{a_t^*,t}) - U_t(\mathbf{f}_{a_t,t})\right] \tag{4}$$

where the expectation $\mathbb{E}[\cdot]$ is taken over both noise space and context space. The algorithm needs to be designed to find an arm selection policy based on the history information to minimize the cumulative regret $R_T = \sum_{t=1}^T reg_t$.

3 Algorithm

In this section, we first introduce the feedback prediction algorithm and then, given the predicted feedbacks and confidence widths, design a UCB-based algorithm with probabilistic contextual information.

3.1 Feedback Prediction

In order to select an action, the algorithm should be able to predict the feedbacks corresponding to each action, which then determines the resulting reward. Note that we cannot simply treat the overall utility function as a single feedback signal and directly predict it given incoming contextual information as in prior studies [Kirschner and Krause, 2019], because the utility function in terms of multiple feedbacks is changing over time in our setting. To accomplish feedback prediction, we can estimate the parameter θ_a^j in feedback functions by kernel-based empirical risk minimization based on the history $\mathcal{H}_{a,t}^j = \left\{\left(\mathcal{X}, Pr_\tau(\mathcal{X}), f_{a,\tau}^j\right), \tau = 1, \cdots, t\right\}, j = 1, \cdots, J$. Denote the set of rounds when arm a is selected before round t as $\mathcal{T}_{a,t} = \left\{\tau_a^1, \tau_a^2, \cdots, \tau_a^{n_{a,t}}\right\}$, where $n_{a,t}$ is the number of times that arm a has been selected prior to round t. The kernel based empirical risk minimization is to solve the following problem

$$\hat{\theta}_{a,t}^j = \arg\min_{\theta_a^j} \frac{1}{n_{a,t}} \sum_{\tau \in \mathcal{T}_{a,t}} (\mathbb{E}[\phi^j(X_t)]^\top \theta_a^j - f_{a,\tau}^j)^2 + \lambda \|\theta_a^j\|^2 \tag{5}$$

where $\lambda \geq 0$ is a hyper-parameter.

Denote $\mathbf{\Phi}_{a,t}^j = \left[\mathbb{E}\left[\phi^j(X_{\tau_a^1})\right], \cdots, \mathbb{E}\left[\phi^j(X_{\tau_a^{n_{a,t}}})\right]\right]$ and $\mathbf{y}_{a,t}^j = \left[f_{a,\tau_a^1}^j, f_{a,\tau_a^2}^j, \cdots, f_{a,\tau_a^{n_{a,t}}}^j\right]^\top$. By solving the optimization problem (5), the parameter $\theta_{a,t}^j$ is estimated as

$$\hat{\theta}_{a,t}^j = {\mathbf{C}_{a,t}^j}^{-1} \mathbf{\Phi}_{a,t}^j \mathbf{y}_{a,t}^j \tag{6}$$

where $\mathbf{C}_{a,t}^j = \mathbf{\Phi}_{a,t}^j {\mathbf{\Phi}_{a,t}^j}^\top + \lambda \mathbf{I}$. Then, the estimated feedback with respect to candidate x_t^i can be calculated as

$$\hat{f}_{a,t}^{i,j} = \phi^j(x_t^i)^\top \hat{\theta}_{a,t}^j \tag{7}$$

whose confidence width [Deshmukh *et al.*, 2017; Kirschner and Krause, 2019] is

$$w_{a,t}^{i,j} = \sqrt{\phi^j(x_t^i)^\top (\mathbf{C}_{a,t}^j)^{-1} \phi^j(x_t^i)}. \tag{8}$$

As the algorithm may not have access to the mapping function $\phi^j(x)$, we need to represent Eqn. (7) and Eqn. (8) by kernel function. By the Woodbury matrix identity, we have

$$\begin{aligned} \hat{f}_{a,t}^{i,j} &= \phi^j(x_t^i)^\top ({\mathbf{\Phi}_{a,t}^j}{\mathbf{\Phi}_{a,t}^j}^\top + \lambda \mathbf{I})^{-1} \mathbf{\Phi}_{a,t}^j \mathbf{y}_{a,t}^j \\ &= \phi^j(x_t^i)^\top \mathbf{\Phi}_{a,t}^j ({\mathbf{\Phi}_{a,t}^j}^\top \mathbf{\Phi}_{a,t}^j + \lambda \mathbf{I})^{-1} \mathbf{y}_{a,t}^j \end{aligned} \tag{9}$$

and

$$\begin{aligned} \lambda {w_{a,t}^{i,j}}^2 &= \phi^j(x_t^i)^\top \phi^j(x_t^i) - \\ &\phi^j(x_t^i)^\top \mathbf{\Phi}_{a,t}^j ({\mathbf{\Phi}_{a,t}^j}^\top \mathbf{\Phi}_{a,t}^j + \lambda \mathbf{I})^{-1} {\mathbf{\Phi}_{a,t}^j}^\top \phi^j(x_t^i). \end{aligned} \tag{10}$$

Denote $\mathbf{k}_{a,t}^{i,j} = {\mathbf{\Phi}_{a,t}^j}^\top \phi^j(x_t^i)$ and $\mathbf{K}_{a,t}^j = {\mathbf{\Phi}_{a,t}^j}^\top \mathbf{\Phi}_{a,t}^j$. The pth entry in $\mathbf{k}_{a,t}^j$ is $\mathbb{E}\left[\phi^j(X_{\tau_a^p})\right]^\top \phi^j(x_t^i) = \sum_{n=1}^{|\mathcal{X}|} P(x_{\tau_a^p}^i) k^j(x_{\tau_a^p}^n, x_t^i)$. Similarly, the entry of $\mathbf{K}_{a,t}^j$ in the pth row and qth column is $\mathbb{E}\left[\phi^j(X_{\tau_a^p})\right]^\top \mathbb{E}[\phi^j(X_{\tau_a^q})] = \sum_{n,m=1}^{|\mathcal{X}|} P(x_{\tau_a^p}^n) P(x_{\tau_a^q}^m) k^j(x_{\tau_a^p}^n, x_{\tau_a^q}^m)$. Now, the estimated feedback can be represented as

$$\hat{f}_{a,t}^{i,j} = {\mathbf{k}_{a,t}^{i,j}}^\top (\mathbf{D}_{a,t}^j)^{-1} \mathbf{y}_{a,t}^j \tag{11}$$

and the confidence width is

$$w_{a,t}^{i,j} = \sqrt{\frac{1}{\lambda} k^j(x_t^i, x_t^i) - \frac{1}{\lambda} {\mathbf{k}_{a,t}^{i,j}}^\top (\mathbf{D}_{a,t}^j)^{-1} \mathbf{k}_{a,t}^{i,j}} \tag{12}$$

where $\mathbf{D}_{a,t}^j = \mathbf{K}_{a,t}^j + \lambda \mathbf{I}$.

3.2 Multi-Feedback Probabilistic Contextual UCB

Based on the results of empirical risk minimization, the proposed algoritm, multi-feedback probabilistic contextual UCB, is given in Algorithm 1.

At each round, the algorithm needs to get the estimated expected reward and the corresponding expected confidence width. To do so, the algorithm first calculates estimated feedbacks and corresponding confidence widths according to Eqn. (11) and Eqn. (12), respectively. Then, given the utility function $U_t : \mathcal{R}^J \to \mathcal{R}$, the estimated reward with respect to the ith context candidate is predicted as $U_t(\hat{\mathbf{f}}_{a,t}^i)$ where

Algorithm 1 Multi-Feedback Probabilistic Contextual UCB

1: **Inputs** :
Arm set $\mathcal{A}$, a horizon T, kernel function $k^1, \cdots, k^J$ and parameter α and λ.
2: **for** $t = 1, \cdots, T$ **do**
3: Receive a set of probabilities $Pr_t(\mathcal{X})$ for the candidates in the context bundle $\mathcal{X}$
4: **for** $a \in \mathcal{A}$ **do**
5: Calculate $\mathbb{E}[U_t(\hat{\mathbf{f}}_{a,t})]$ and $\mathbb{E}[w_{a,t}]$ according to Eqn. (13) and Eqn. (14).
6: **end for**
7: $a_t = \arg\max_{a \in \mathcal{A}} (\mathbb{E}[U_t(\hat{\mathbf{f}}_{a,t})] + L\beta \mathbb{E}[w_{a,t}])$
8: Receive feedback $\mathbf{f}_{a,t} = \left[f_{a,t}^1, \cdots, f_{a,t}^J\right]$.
9: Update $\mathbf{y}_{a,t+1}^j$, $\mathbf{K}_{a,t+1}^j$ and $\mathbf{D}_{a,t+1}^j$
10: **end for**

$\hat{\mathbf{f}}_{a,t}^i = [\hat{f}_{a,t}^{i,1}, \hat{f}_{a,t}^{i,2}, \cdots \hat{f}_{a,t}^{i,j}, \cdots]^\top$. As the exact context is not given, the estimated feedback $\hat{\mathbf{f}}_{a,t}$ is a random vector over the probabilistic context space. Hence, the estimated expected reward over the probabilistic context space is written as

$$\mathbb{E}[U_t(\hat{\mathbf{f}}_{a,t})] = \sum_{i=1}^{|\mathcal{X}|} P_t(x_t^i) U_t(\hat{\mathbf{f}}_{a,t}^i). \tag{13}$$

The confidence width is important for arm exploration, but it is not trivial to get the expected confidence width. Here, we calculate the upper bound of the expected confidence width over the probabilistic context space by exploiting Lipschitz continuity of the utility function. Concretely, if L is the Lipschitz constant of the utility function, the upper bound of expected confidence width over the probabilistic context space is calculated as

$$\mathbb{E}[w_{a,t}] = \sum_{i=1}^{|\mathcal{X}|} P_t(x_t^i) \sum_{j=1}^{J} w_{a,t}^{i,j}. \tag{14}$$

The detailed derivation of Eqn. (14) will be given in Lemma 4.2.

With the estimated expected reward and the corresponding expected confidence width, the selected arm is $a_t = \arg\max_{a \in \mathcal{A}} (\mathbb{E}[U_t(\hat{\mathbf{f}}_{a,t})] + L\beta \mathbb{E}[w_{a,t}])$, where β is a hyper-parameter to balance the exploration and exploitation.

4 Regret Analysis

In this section, we analyze the regret with respect to an oracle that also has the probabilistic context information and establish an upper bound on cumulative regret of Algorithm 1 which shows the cumulative regret sub-linearly increases with $\mathcal{O}(\sqrt{T \log T})$, followed by the proof sketch.

4.1 Cumulative Regret Bound

The following theorem provides an upper bound on the cumulative regret of Algorithm 1.

Theorem 4.1. *Assume at each round t, the utility function $U_t(\mathbf{f}_{a,t}) \in [0,1]$ satisfies $\mathbb{E}_\epsilon[U_t(\mathbf{f}_{a,t})] = U_t(\mathbb{E}_\epsilon[\mathbf{f}_{a,t}])$ with*

Lipschitz constant L_t and $L = \max_t L_t$, and the kernel function is $k^j(x, x') \leq c_k$ such that $\phi^j(x) \succeq 0$. At each round t, the agent receives a probabilistic context set $\mathcal{X}$ and the corresponding probability set $Pr_t(\mathcal{X})$, selects arm from $\mathcal{A}$ by Algorithm 1 and get J different feedbacks. With probability $1-\delta$, the cumulative expected regret R_T of Algorithm 1 is bounded by

$$R_T \leq 2L\beta J|\mathcal{A}||\mathcal{X}|\sqrt{2q\gamma_m T\log(\frac{(T+1)c_k+\lambda}{d^{\frac{1}{\gamma_m}}\lambda})} = \mathcal{O}(\sqrt{T\log T}) \tag{15}$$

where γ_m is the maximum rank of $\mathbf{K}^j_{a,t}$, $q = \max(1, \frac{c_k}{\lambda})$ and $\beta = (\sqrt{\frac{\log(2TJ|\mathcal{A}|/\delta)}{2}} + c\sqrt{\lambda})$.

Remark 4.1. Theorem 4.1 shows that, for the bandit setting with probabilistic contexts and multiple feedbacks, our proposed algorithm can achieve a sub-linear cumulative expected regret bound $\mathcal{O}(\sqrt{T\log T})$. This demonstrates the effectiveness of our proposed algorithm.

Remark 4.2. Compared with the cumulative regret bound of kernel-UCB in the standard bandit setting [Chowdhury and Gopalan, 2017; Deshmukh *et al.*, 2017], the cumulative regret bound of the proposed algorithm is scaled by Lipschitz constant L, number of feedbacks J and size of context bundle $|\mathcal{X}|$. As a result, the regret in our setting is more difficult to be reduced than that in the standard setting. Nonetheless, by the proposed algorithm, the cumulative regret can still be guaranteed to be sub-linear.

4.2 Proof Sketch

The proof sketch of Theorem 4.1 is given below. Compared with other UCB algorithms [Abbasi-Yadkori *et al.*, 2011; Deshmukh *et al.*, 2017; Kirschner and Krause, 2019], the consideration of probabilistic context, multiple noisy feedbacks and Lipschitz utility function adds new challenges to the regret bound proof. First, since the algorithm predicts feedbacks instead of the reward, the predicted feedbacks can be guaranteed to converge to the expected feedbacks by Lemma 1 in [Deshmukh *et al.*, 2017], but we still need to bound the gap between the estimated reward and the true expected reward. Second, since the proposed algorithm only has probabilistic contexts, we can bound the expected reward estimation error by the expected confidence width, but it is still challenging to get the sum of the confidence width over time. Next, we show several important lemmas to address the aforementioned challenges.

First, by exploiting the Lipschitz continuity of utility function, the confidence width of estimated reward is bounded in Lemma 4.2, which also explains the setting of confidence width in Algorithm 1.

Lemma 4.2 (Concentration of Empirical Risk Minimization)**.** *Assume the utility function $U_t(\cdot)$ is in a linear form or multiplication form with Lipschitz constant L. With probability at least $1-\frac{\delta}{T}$, for $\forall a \in \mathcal{A}$, we have*

$$\left|\mathbb{E}\left[U_t(\hat{\mathbf{f}}_{a,t})\right] - \mathbb{E}\left[U_t(\mathbf{f}_{a,t})\right]\right| \leq L\beta\mathbb{E}[w_{a,t}] \tag{16}$$

where $\beta = (\sqrt{\frac{\log(2TJ|\mathcal{A}|/\delta)}{2}} + c\sqrt{\lambda})$.

Proof. Let $g^{i,j}_{a,t} = g^j_a\left(x^i_t\right)$ and $\mathbf{g}^i_{a,t} = \left[g^{i,1}_a, \cdots, g^{i,j}_a\right]^\top$.By the assumption $\mathbb{E}_\epsilon[U_t(\mathbf{f}_{a,t})] = U_t(\mathbb{E}_\epsilon[\mathbf{f}_{a,t}])$, we have

$$\begin{aligned}\left|\mathbb{E}\left[U_t(\hat{\mathbf{f}}_{a,t})\right] - \mathbb{E}\left[U_t(\mathbf{f}_{a,t})\right]\right| &= \left|\sum_{i=1}^{|\mathcal{X}|} P_t\left(x^i_t\right)\left(U_t\left(\hat{\mathbf{f}}^i_{a,t}\right) - U_t\left(\mathbf{g}^i_{a,t}\right)\right)\right| \\ \leq \sum_{i=1}^{|\mathcal{X}|} P_t\left(x^i_t\right) L\left\|\hat{\mathbf{f}}^i_{a,t} - \mathbf{g}^i_{a,t}\right\| &\leq L\sum_{i=1}^{|\mathcal{X}|} P_t\left(x^i_t\right)\sum_{j=1}^{J}\left|\hat{f}^{i,j}_{a,t} - g^{i,j}_{a,t}\right|.\end{aligned} \tag{17}$$

By Lemma 1 in [Deshmukh *et al.*, 2017], we have $\left|\hat{f}^{i,j}_{a,t} - g^{i,j}_{a,t}\right| \leq \beta w^{i,j}_{a,t}$ with probability at least $1-\frac{\delta}{JT}$. Thus, with probability at least $1-\frac{\delta}{T}$, we have $\left|\mathbb{E}\left[U_t(\hat{\mathbf{f}}_{a,t}) - U_t(\mathbf{f}_{a,t})\right]\right| \leq L\beta\sum_{i=1}^{|\mathcal{X}|} P_t(x^i_t)\sum_{j=1}^{J} w^{i,j}_{a,t} = L\beta\mathbb{E}[w_{a,t}]$. □

Then, by using Lemma 4.2, we will bound the regret by the expected confidence width in the next lemma.

Lemma 4.3 (Regret Bound by Confidence Width)**.** *Assume that the utility function $U_t(\cdot)$ satisfies $\mathbb{E}_\epsilon[U_t(\mathbf{f}_{a,t})] = U_t(\mathbb{E}_\epsilon[\mathbf{f}_{a,t}])$ with Lipschitz constant L. With probability at least $1-\frac{\delta}{T}$, the cumulative regret satisfies*

$$R_T = \sum_{t=1}^{T} reg_t \leq 2L\beta\sum_{t=1}^{T}\mathbb{E}[w_{a_t,t}] \tag{18}$$

where $\beta = (\sqrt{\frac{\log(2TJ|\mathcal{A}|/\delta)}{2}} + c\sqrt{\lambda})$.

Proof. By similar proof techniques for standard UCB [Abbasi-Yadkori *et al.*, 2011; Kirschner and Krause, 2019], with probability at least $1-\frac{\delta}{T}$, the instant regret for round t is bounded as

$$\begin{aligned} reg_t &= \mathbb{E}\left[U_t(\mathbf{f}_{a^*_t,t}) - U_t(\hat{\mathbf{f}}_{a^*_t,t}) + U_t(\hat{\mathbf{f}}_{a^*_t,t}) - U_t(\mathbf{f}_{a_t,t})\right] \\ &\leq L\beta\mathbb{E}[w_{a^*_t,t}] + \mathbb{E}\left[U_t(\hat{\mathbf{f}}_{a^*_t,t})\right] - \mathbb{E}\left[U_t(\mathbf{f}_{a_t,t})\right] \\ &\leq L\beta\mathbb{E}[w_{a_t,t}] + \mathbb{E}\left[U_t(\hat{\mathbf{f}}_{a_t,t})\right] - \mathbb{E}\left[U_t(\mathbf{f}_{a_t,t})\right] \\ &\leq 2L\beta\mathbb{E}[w_{a_t,t}] \end{aligned} \tag{19}$$

where the first and third inequalities hold by Lemma 4.2 and the second inequality holds by the arm selection policy in Algorithm 1. Thus, the cumulative regret is bounded as Eqn. (18). □

The next challenge is to bound the sum of confidence width, which is expressed as

$$\sum_{t=1}^{T}\mathbb{E}[w_{a_t,t}] = \sum_{t=1}^{T}\sum_{i=1}^{|\mathcal{X}|} P_t(x^i_t)\sum_{j=1}^{J}\sqrt{\phi^j(x^i_t)^\top(\mathbf{C}^j_{a,t})^{-1}\phi^j(x^i_t)}. \tag{20}$$

We cannot directly use Sylvester's determinant theorem or Schur's determinant identity like in the proofs of Lemma 11

in [Abbasi-Yadkori *et al.*, 2011] and Lemma 7 in [Deshmukh *et al.*, 2017]. Thus, we first derive Lemma 4.4 to get an upper bound of $\mathbb{E}[w_{a_t,t}]$, and then get the sum of the expected confidence width in Lemma 4.4

Lemma 4.4 (Sum of Confidence Width)**.** *Assume kernel function k^j is chosen such that mapping function $\phi^j(x) \succeq 0$, we have*

$$\sum_{t=1}^{T} \mathbb{E}[w_{a_t,t}] \leq J|\mathcal{X}|\sqrt{2q\gamma_m T \log(\frac{(T+1)c_k+\lambda}{d^{\frac{1}{\gamma_m}}\lambda})} \quad (21)$$

where γ_m is the maximum rank of $\mathbf{K}^j_{a,t}$, $q = \max(1, \frac{c_k}{\lambda})$.

Proof. First, we bound $\mathbb{E}[w^j_{a,t}]$ by $\bar{w}^j_{a,t}$ where $\bar{w}^j_{a,t} = \sqrt{\mathbb{E}[\phi^j(X_t)]^\top (\mathbf{C}^j_{a,t})^{-1}\mathbb{E}[\phi^j(X_t)]}$. Let $w^{i,j}_{a_t,t} = \sqrt{\phi^j(x^i_t)^\top (\mathbf{C}^j_{a,t})^{-1}\phi^j(x^i_t)}$. Then, we have

$$\begin{aligned}\left(P_t(x^i_t)w^{i,j}_{a_t,t}\right)^2 &= P_t(x^i_t)\phi^j(x^i_t)^\top(\mathbf{C}^j_{a,t})^{-1}P_t(x^i_t)\phi^j(x^i_t) \\ &\leq \mathbb{E}[\phi^j(X_t)]^\top(\mathbf{C}^j_{a,t})^{-1}\mathbb{E}[\phi^j(X_t)]\end{aligned} \quad (22)$$

where the inequality holds because $\phi^j(x) \succeq 0$ and thus $\mathbb{E}[\phi^j(X_t)] = \sum_{n=1}^{|\mathcal{X}|} P_t(x^n_t)\phi^j(x^n_t) \geq P_t(x^i_t)\phi^j(x^i_t)$. By taking squared root of both sides of Eqn. (22), we have $P_t(x^i_t)w^{i,j}_{a,t} \leq \bar{w}^j_{a,t}$, and thus $\mathbb{E}[w^j_{a,t}] = \sum_{i=1}^{|\mathcal{X}|} P_t(x^i_t)w^{i,j}_{a,t} \leq |\mathcal{X}|\bar{w}^j_{a,t}$. Since $\sum_{t=1}^{T}\bar{w}^j_{a_t,t}$ can be bounded by Lemma 8 in [Deshmukh *et al.*, 2017], i.e. $\sum_{t=1}^{T}\bar{w}^j_{a_t,t} \leq \sqrt{2q\gamma_m T\log(\frac{(T+1)c_k+\lambda}{d^{\frac{1}{\gamma_m}}\lambda})}$, the inequality (21) can be proved. □

By substituting Eqn. (21) into Eqn. (18), we can get the cumulative regret bound in of Algorithm 1 in Theorem 4.1.

5 Simulation Results

We now apply Algorithm 1 to the problem of DNN model selection for mobile devices and show its performance in terms of average reward and cumulative regret. Importantly, our result demonstrates that compared to the oracle that knows the optimal arm given a probabilistic context bundle, the cumulative regret achieved by our algorithm increases sub-linearly over time, validating our theoretical regret analysis.

5.1 Application to DNN Model Selection

The recent breakthrough in DNN model compression has made it possible to run DNN inference on edge devices (e.g., mobile phones and tablets). While they can have similar inference accuracies, different DNN models have different latencies and energy consumption under different system conditions. Thus, it is crucial to select an optimal DNN model for edge inference with the best user experience. This is challenged by the fact that, although the basic configuration of an edge device (e.g., CPU, OS, RAM) requesting a DNN model is exposed to the model provider, the device's actual resource management and system condition (e.g., available system resources) that decide the latency and energy of the deployed DNN model can only be known probabilistically.

	Phone 1	Phone 2	Tablet 1	Tablet 2
InceptionV2Q	0.45 J	0.07 J	6.18 J	1.41 J
InceptionV4Q	1.88 J	0.22 J	11.66 J	6.59 J
InceptionV4F	5.04 J	1.14 J	37.29 J	10.87 J
MobileNetV1Q	0.13 J	0.03 J	2.00 J	0.69 J
MobileNetV1F	0.18 J	0.04 J	2.00 J	0.60 J

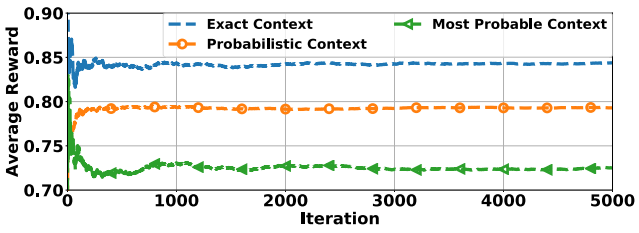
Table 1. Average Energy Consumption

	Phone 1	Phone 2	Tablet 1	Tablet 2
InceptionV2Q	$0.33s$	$0.11s$	$2.60s$	$0.57s$
InceptionV4Q	$1.40s$	$0.35s$	$4.45s$	$2.53s$
InceptionV4F	$2.01s$	$1.23s$	$18.95s$	$4.58s$
MobileNetV1Q	$0.10s$	$0.05s$	$0.83s$	$0.22s$
MobileNetV1F	$0.13s$	$0.07s$	$0.60s$	$0.25s$

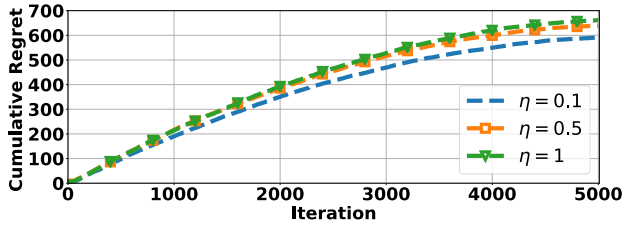
Table 2. Average Latency

Concretely, the available DNN models for selection constitute the set of arms, an edge device's actual system condition is the context, and we consider DNN inference latency l and energy consumption e as the two feedback signals. Our goal is to select optimal DNN models for edge devices that arrive sequentially. For evaluation purposes, we run experiments and collect measured data of five image classification DNN models from TensorFlow Hub running on two cellphones (Vivo V1838A and Google Pixel 3a) and two tablets (Samsung - Galaxy Tab A7 and Vankyo MatrixPad Z4). The energy consumption and latency measurement results are shown in Tables 1 and 2, respectively. We use these four devices to represent four types of actual system conditions (i.e., context in our study) in an edge device requesting a DNN model. In other words, when an edge device arrives, its actual system condition is assumed to fall into one of the conditions as specified by the four different devices in our evaluation. While we can further run experiments on these devices under different usage scenarios to have more fine-grained types of contexts, our current setup is enough to validate our theoretical analysis. Note that although an edge device's basic hardware configuration is accessible to the DNN model provider, its actual system condition (i.e., context in our problem) is only known probabilistically for DNN model selection.

We assume that the utility function for a DNN model selection decision is a weighted linear combination of energy consumption and latency, while noting that other utility functions can also be considered (e.g., the energy-delay product function) provided that they satisfy $\mathbb{E}_\epsilon[U_t(\mathbf{f}_{a,t})] = U_t(\mathbb{E}_\epsilon[\mathbf{f}_{a,t}])$ where the expectation $\mathbb{E}_\epsilon[\cdot]$ is taken over the noise space. In general, the weights in our linear utility function can change for different devices (e.g., energy consumption plays a more important role for devices with a small battery capacity). For illustration, we assume that the utility function can be either $U(e,l) = -0.36e - 0.54l + 1$ or $U(e,l) = -0.25e - 0.65l + 0.9$, which is randomly determined. We compare our algorithm with kernel contextual bandit algorithms that utilize the exact context and the most probable



(a) Reward



(b) Regret

Figure 1. Performance Comparison.

context, respectively. We randomly generate the $Pr_t(\mathcal{X})$ as input at each round. We use the radial basis function kernel $k(x, x') = \exp(-\rho\|x - x'\|_2^2)$.

5.2 Results

In Fig. 1(a), we show the average reward achieved by different algorithms. Naturally, the algorithm with the exact true context achieves the highest reward. Nonetheless, the reward of our algorithm is greater than that of the straightforward algorithm that utilizes the most probable context as if it were the true context (similar to a standard UCB algorithm). The reason for the low reward achieved by using the most probable context is that the stored possibly erroneous contexts can be uncorrelated with the received feedback, thus resulting in biased estimation of the feedback functions and hence inaccurate reward prediction further.

In Fig. 1(b), to validate the sub-linear regret, we compare the cumulative regret of our algorithm to the oracle that knows the optimal arm for any probabilistic context bundle. We use entropy of context's probability distribution $H(Pr_t(\mathcal{X}))$ as a measure for how random the provided context bundle is. We denote $H_{max} = \log_2 |\mathcal{X}|$ as the largest entropy and η $(0 \leq \eta \leq 1)$ as a threshold to bound randomness for different rounds, where $H(Pr_t(\mathcal{X})) \leq \eta H_{max}$. The smaller η, the more concentrated probabilistic distribution $Pr_t(\mathcal{X})$ of a context bundle (or, less randomness). If $\eta = 0$, then the distribution only reveals the exact context. The result shows that the regret of our algorithm is sub-linearly increasing, regardless of randomness of probabilistic bundle.

6 Related Work

Contextual bandits have been studied in various settings due to their wide applications [Langford and Zhang, 2008]. The study [Li *et al.*, 2010] proposes Lin-UCB algorithm, assuming a linear relationship between its context and expected reward, which applies ridge regression for estimated feedback. As for the nonlinear contextual bandits, [Valko *et al.*, 2013] and [Deshmukh *et al.*, 2017] both propose kernelized contextual bandit as a nonlinear version of Lin-UCB by finding linear members in RHKS. [Allesiardo *et al.*, 2014] utilizes neural networks to predict the rewards given the context and proposed a multi-expert approach to decide the parameters of networks. [Zhou *et al.*, 2019] provides a formal regret bound for neural network-based contextual UCB. [Badanidiyuru *et al.*, 2014] introduces the concept of contextual bandits with budget constraints, and proposes a resourceful contextual bandits algorithm that provably achieves $\mathcal{O}(\sqrt{T})$ regret bound. Another variant of contextual bandit considers that not all contextual information is accessible [Bouneffouf *et al.*, 2017]. Similarly, [Wang *et al.*, 2016] assumes the existence of hidden features and arm vectors from context together and proposes hLin-UCB algorithm.

Among the studies on probabilistic contextual bandits, a relevant one [Kirschner and Krause, 2019] considers that the agent only knows the probability distribution of context. The major difference is that we consider multiple feedbacks and a time-varying utility function. Another one is [Yun *et al.*, 2017], which studies contextual bandit with perturbation noise on observed context. The authors assume a linear reward function with a single feedback, and propose an algorithm called NLin-Rel that achieves $\mathcal{O}(T^{\frac{7}{8}})$ regret bound under the assumption of identical noise. Different from this work, we consider a utility function in terms of multiple nonlinear feedback functions with probabilistic contexts.

As for multiple feedbacks, [Lu *et al.*, 2019] proposes an algorithm based on Pareto optimality to solve a multi-objective problem under contextual settings, resulting in a regret bound that increases sub-linearly under the assumption of a linear feedback function. Another relevant work is [Wanigasekara *et al.*, 2019], which considers a multi-objective online contextual ranking system with the assumption that some parameters in both feedback and reward are unknown. By setting linear feedback and logistic utility, the proposed UCB-based algorithm is shown to significantly increase the click-through rate. In contrast, we use a more general time-varying utility function to combine multiple feedback signals and consider probabilistic contexts.

7 Conclusion

In this paper, we consider a new setting of bandit learning with multiple feedback signals, time-varying utility functions and probabilistic context information. For this setting, we propose a multi-feedback probabilistic kernelized UCB algorithm to choose the optimal arm in order to minimize the expected cumulative regret. We derive an upper bound of the expected cumulative regret incurred by our proposed algorithm, with respect to the best action that maximize the expected reward, and show that the bound grows sub-linearly with time. We apply the proposed algorithm to DNN model selection. The simulation results further validate the sub-linearity of the cumulative regret.

Acknowledgments

This work was supported in part by the U.S. NSF under grants CNS-1551661, ECCS-1610471, and CNS-1910208.

References

[Abbasi-Yadkori *et al.*, 2011] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

[Allesiardo *et al.*, 2014] Robin Allesiardo, Raphaël Féraud, and Djallel Bouneffouf. A neural networks committee for the contextual bandit problem. In *International Conference on Neural Information Processing*, pages 374–381, 2014.

[Auer *et al.*, 2002] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

[Badanidiyuru *et al.*, 2014] Ashwinkumar Badanidiyuru, John Langford, and Aleksandrs Slivkins. Resourceful contextual bandits. In *Conference on Learning Theory*, pages 1109–1134, 2014.

[Bouneffouf *et al.*, 2017] Djallel Bouneffouf, Irina Rish, Guillermo A Cecchi, and Raphaël Féraud. Context attentive bandits: contextual bandit with restricted context. In *International Joint Conference on Artificial Intelligence*, pages 1468–1475, 2017.

[Chowdhury and Gopalan, 2017] Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning*, pages 844–853, 2017.

[Deshmukh *et al.*, 2017] Aniket Anand Deshmukh, Urun Dogan, and Clay Scott. Multi-task learning for contextual bandits. In *Advances in Neural Information Processing Systems*, pages 4848–4856, 2017.

[Kirschner and Krause, 2019] Johannes Kirschner and Andreas Krause. Stochastic bandits with context distributions. In *Advances in Neural Information Processing Systems*, pages 14090–14099, 2019.

[Lai and Robbins, 1985] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

[Langford and Zhang, 2008] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in Neural Information Processing Systems*, pages 817–824, 2008.

[Li *et al.*, 2010] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *International Conference on World Wide Web*, pages 661–670, 2010.

[Li *et al.*, 2017] Lihong Li, Yu Lu, and Dengyong Zhou. Provably optimal algorithms for generalized linear contextual bandits. In *International Conference on Machine Learning*, pages 2071–2080, 2017.

[Lin *et al.*, 2018] Baihan Lin, Guillermo Cecchi, Djallel Bouneffouf, and Irina Rish. Adaptive representation selection in contextual bandit. *arXiv:1802.00981*, 2018.

[Lu *et al.*, 2019] Shiyin Lu, Guanghui Wang, Yao Hu, and Lijun Zhang. Multi-objective generalized linear bandits. In *International Joint Conference on Artificial Intelligence*, pages 3080–3086, 2019.

[Mary *et al.*, 2015] Jérémie Mary, Romaric Gaudel, and Philippe Preux. Bandits and recommender systems. In *International Workshop on Machine Learning, Optimization and Big Data*, pages 325–336, 2015.

[Quinlan, 1986] J. Ross Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[Roijers *et al.*, 2017] Diederik M Roijers, Luisa M Zintgraf, and Ann Nowé. Interactive thompson sampling for multi-objective multi-armed bandits. In *International Conference on Algorithmic Decision Theory*, pages 18–34, 2017.

[Tang *et al.*, 2013] Liang Tang, Romer Rosales, Ajit Singh, and Deepak Agarwal. Automatic ad format selection via contextual bandits. In *International Conference on Information & Knowledge Management*, pages 1587–1594, 2013.

[Valko *et al.*, 2013] Michal Valko, Nathan Korda, Rémi Munos, Ilias Flaounas, and Nello Cristianini. Finite-time analysis of kernelised contextual bandits. In *Conference on Uncertainty in Artificial Intelligence*, pages 654–663, 2013.

[Villar *et al.*, 2015] Sofía S Villar, Jack Bowden, and James Wason. Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges. *Statistical Science: A Review Journal of the Institute of Mathematical Statistics*, 30(2):199, 2015.

[Wan, 1990] Eric A. Wan. Neural network classification: A bayesian interpretation. *IEEE Transactions on Neural Networks*, 1(4):303–305, 1990.

[Wang *et al.*, 2016] Huazheng Wang, Qingyun Wu, and Hongning Wang. Learning hidden features for contextual bandits. In *International Conference on Information and Knowledge Management*, pages 1633–1642, 2016.

[Wanigasekara *et al.*, 2019] Nirandika Wanigasekara, Yuxuan Liang, Siong Thye Goh, Ye Liu, Joseph Jay Williams, and David S Rosenblum. Learning multi-objective rewards and user utility function in contextual bandits for personalized ranking. In *International Joint Conference on Artificial Intelligence*, pages 3835–3841, 2019.

[Yahyaa and Manderick, 2015] Saba Yahyaa and Bernard Manderick. Thompson sampling for multi-objective multi-armed bandits problem. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning.*, pages 47–52, 2015.

[Yun *et al.*, 2017] Se-Young Yun, Jun Hyun Nam, Sangwoo Mo, and Jinwoo Shin. Contextual multi-armed bandits under feature uncertainty. *arXiv:1703.01347*, 2017.

[Zhou *et al.*, 2019] Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural contextual bandits with UCB-based exploration. *arXiv:1911.04462*, 2019.