Multi-facet Contextual Bandits: A Neural Network Perspective

Yikun Ban  
University of Illinois at Urbana-Champaign  
yikunb2@illinois.edu

Jingrui He  
University of Illinois at Urbana-Champaign  
jingrui@illinois.edu

Curtiss B. Cook  
Mayo Clinic Arizona  
cook.curtiss@mayo.edu

ABSTRACT

Contextual multi-armed bandit has shown to be an effective tool in recommender systems. In this paper, we study a novel problem of multi-facet bandits involving a group of bandits, each characterizing the users’ needs from one unique aspect. In each round, for the given user, we need to select one arm from each bandit, such that the combination of all arms maximizes the final reward. This problem can find immediate applications in E-commerce, healthcare, etc. To address this problem, we propose a novel algorithm, named MuFasa, which utilizes an assembled neural network to jointly learn the underlying reward functions of multiple bandits. It estimates an Upper Confidence Bound (UCB) linked with the expected reward to balance between exploitation and exploration. Under mild assumptions, we provide the regret analysis of MuFasa. It can achieve the near-optimal $\tilde{O}\left((K + 1)\sqrt{T}\right)$ regret bound where $K$ is the number of bandits and $T$ is the number of played rounds. Furthermore, we conduct extensive experiments to show that MuFasa outperforms strong baselines on real-world data sets.

CCS CONCEPTS

• Information systems → Personalization; Display advertising  
• Theory of computation → Online learning algorithms.

KEYWORDS

Contextual Bandits; Neural Network; Regret Analysis

1 INTRODUCTION

The personalized recommendation is ubiquitous in web applications. Conventional approaches that rely on sufficient historical records, e.g., collaborative filtering [37, 47], have proven successful both theoretically and empirically. However, with the cold-start problem and the rapid change of the recommendation content, these methods might render sub-optimal performance [23, 28]. To solve the dilemma between the exploitation of historical data and the exploration of new information, Multi-Armed Bandit (MAB) [1, 7, 8, 10, 26] turns out to be an effective tool, which has been adapted to personalized recommendation [23, 28], online advertising [40], clinical trials [12, 20], etc.

In the conventional contextual bandit problem setting [28], i.e., single MAB, the learner is presented with a set of arms in each round, where each arm is represented by a feature vector. Then the learner needs to select and play one arm to receive the corresponding reward that is drawn from an unknown distribution with an unknown mean. To achieve the goal of maximizing the accumulated rewards, the learner needs to consider the arms with the best historical feedback as well as the new arms for potential gains. The single MAB problem has been well studied in various settings. With respect to the reward function, one research direction [1, 19, 23, 28, 29] assumes that the expected reward is linear with respect to the arm’s feature vector. However, in many real applications, this assumption fails to hold. Thus many exiting works turn to focus on the nonlinear or nonparametric bandits [13, 36] with mild assumptions such as the Lipschitz continuous property [13] or embedding in Reproducing Kernel Hilbert Space [18, 39]. Furthermore, the single MAB problem has been extended to best arm identification [6, 7], outlier arm identification [10, 22], Top-K arm problems [14], and so on.

In this paper, we define and study a novel problem of multi-facet contextual bandits. In this problem, the users’ needs are characterized from multiple aspects, each associated with one bandit. Consider a task consisting of $K$ bandits, where each bandit presents a set of arms separately and the learner needs to choose and play one arm from each bandit. Therefore, a total of $K$ arms are played in one round. In accordance with the standard bandit problem, the learner can observe a reward after playing one arm from the bandit, which we call “sub-reward”, and thus $K$ sub-rewards are received in total. In addition, a reward that is a function with respect to these $K$ sub-rewards, called “final reward”, is observed to represent the overall feedback with respect to the $K$ selected arms. Note that the functions of final reward and $K$ sub-rewards are allowed to be either linear or non-linear. The goal of the learner in the multi-facet bandit problem is to maximize the final rewards of all the played rounds.

This problem finds many applications in real-world problems. For instance, in the recommender system, instead of the single item recommendation, an E-commerce company launches a promotion campaign, which sells collections of multiple types of products such as snacks, toiletries, and beverages. Each type of item can be formulated as a multi-armed bandit and the learner aims to select the best combination of snack, toiletry, and beverage. As a result, the final reward is the review of this combined recommendation, while the sub-reward is the review for a particular product. This problem also exists in healthcare. For a diabetes patient, the doctor usually
A major challenge of the proposed multi-facet bandit problem is the partial availability of sub-rewards, as not every sub-reward is easy to observe. For example, regarding the combined recommendation of E-commerce, the user may rate the combination but not individual items; regarding the comprehensive recommendation for a diabetes patient, some sub-rewards can be difficult to measure (e.g., the impact of low-calorie diets on the patient’s overall health conditions). Therefore, in our work, we allow only a subset of all sub-rewards to be observed in each round, which increases the flexibility of our proposed framework.

To address these challenges, we aim to learn the mappings from the selected $K$ arms (one from each bandit) to the final rewards, incorporating two crucial factors: (1) the collaborative relations exist among these facets as they formulate the aspects from one single user; (2) the facets contribute to the task with various weights because some aspects (facets) are decisive while some maybe not. Hence, we propose a novel algorithm, MuFasa, to learn $K$ facets jointly. It utilizes an assembled neural networks to learn the final reward function combined with $K$ facets. Although the neural networks have been adapted to the facet problem [34, 42, 45], they are designed for the single facet with one selected arm and one reward in each round. To balance the exploitation and exploration of arm sets, we provide a comprehensive upper confidence bound based on the assembled network linking the predicted reward with the expected reward. When the sub-rewards are partially available, we introduce a new approach to leverage them to train facets jointly. Furthermore, we carry out the theoretical analysis of MuFasa and prove a near-optimal regret bound under mild assumptions. Our major contributions can be summarized as follows:

1. **Problem.** We introduce the problem of multi-facet contextual bandits to characterize the users’ needs from multiple aspects, which can find immediate applications in E-commerce, healthcare, etc.

2. **Algorithm.** We propose a novel algorithm, MuFasa, which exploits the final reward and up to $K$ sub-rewards to train the assembled neural networks and explores potential arm sets with a UCB-based strategy.

3. **Theoretical analysis.** Under mild assumptions, we provide the upper confidence bounds for a neural network and the assembled neural networks. Then, we prove that MuFasa can achieve the $\mathcal{O}(\sqrt{KT})$ regret bound, which is near-optimal compared to a single contextual bandit.

4. **Empirical performance.** We conduct extensive experiments to show the effectiveness of MuFasa, which outperforms strong baselines on real-world data sets even with partial sub-rewards.

## 2 RELATED WORK

### Multi-armed bandit

The multi-armed bandit was first introduced by [38] and then further studied by many works that succeeded in both theory and practice such as $\epsilon$-greedy [26], Thompson sampling [2], and upper confidence bound [7]. In the contrast with traditional bandits [7, 10], the contextual bandit [1, 28, 40] has the better representation capacity where each arm is represented by a context vector instead of a scalar to infer the reward. Among them, the linear contextual bandits are extensively studied and many of them use the UCB strategy, achieving $\mathcal{O}(\sqrt{T})$ regret bound [1, 11, 23]. To further generalize the reward function, many works use a nonlinear regression model drawn from the reproducing kernel Hilbert space to learn the mapping from contexts to rewards such as the kernel-based methods [18, 39].

### Neural bandits

The authors of [4] use a neural work to model an arm and then applied $\epsilon$-greedy strategy to select an arm. In contrast, MuFasa utilizes a UCB-based strategy working on $K$ bandits instead of a single set of arms. In addition, the Thompson sampling has been combined with deep neural networks [9, 31, 34, 42]. For instance, [34, 42] regard the last layer of the neural network as the embeddings of contexts and then apply the Thompson sampling to play an arm in each round. NeuUCB [45] first uses the UCB-based approach constructed on a fully-connected neural network, while it only fits on the single bandit with one set of arms. On the contrary, MuFasa constructs an assembled neural networks to learn $K$ facets jointly. Deep neural network in multi-view learning has been well-studied [21, 25, 43, 44, 46], to extract useful information among multiple sources, which inspires one of the core ideas of MuFasa.

### Other variant bandit setting

In the non-contextual bandit, a number of works [16, 17, 32] study playing $K$ arms at the same time in a single bandit, while these approaches have limited representation power in the recommender system. The most similar setting is the contextual combinatorial MAB problem [30, 33], where the learner tends to choose the optimal subset of arms with certain constraints like the $K$-size. One key difference is that all the arms are from the same single bandit where only one reward function exists. On the contrary, in the multi-faceted bandits, the selected $K$ arms come from $K$ different bandits with $K$ different reward functions and the sub-rewards are allowed to be partially available. There is another line of works [11, 23, 29] for bandit clustering, where a bandit is constructed for each user. They try to leverage the dependency among users to improve the recommendation performance. However, in these works, they still play one arm in each round and the reward function is required to be linear.

## 3 PROBLEM DEFINITION

In this section, we formulate the problem of multi-facet bandits, with a total of $K$ bandits, where the learner aims to select the optimal set of $K$ arms in each round, in order to maximize the final accumulated rewards.

Suppose there are $T$ rounds altogether. In each round $t \in [T]$ (where $[T] = \{1, \ldots, T\}$), the learner is faced with $K$ bandits, and each bandit $k \in [K]$ has a set of arms $X_k^t = \{x_{t,i}^k, \ldots, x_{T,i}^k\}$, where $|X_k^t| = n_k$ is the number of arms in this bandit. In the bandit $k$, for each arm $x_{t,i}^k \in X_k^t$, it is represented by a $d_k$-dimensional feature vector and we assume $\|x_{t,i}^k\| \leq 1$. Subsequently, in each round $t$, the learner will observe $K$ arm sets $\{X_1^t, \ldots, X_K^t\}$ and thus a total of $\sum_{k=1}^{K} n_k$ arms. As only one arm can be played within each facet, the learner needs to select and play $K$ arms denoted as
where \( K \) is the logarithm. O

where \( K \) optimal presentation, given any set of selected applied to many real-world applications. For the convenience of presentation, given any set of selected \( K \) arms in each round \( t \), a final reward \( R_t \) will be received to represent the overall feedback of the group of \( K \) bandits. The final reward \( R_t \) is considered to be governed by an unknown reward function:

\[
R_t(r_t) = H\left(h_1(x^r_t), \ldots, h_K(x^K_t)\right) + \epsilon_t,
\]

where \( \epsilon_t \) is a noise drawn from a Gaussian distribution with zero mean. In our analysis, we make the following assumptions regarding \( h_k \) and \( H(\text{vec}(r_t)) \):

1. If \( x^k_t = 0 \), then \( h(x^k_t) = 0 \); If \( \text{vec}(r_t) = (0, \ldots, 0) \), then \( H(\text{vec}(r_t)) = 0 \).
2. \( \tilde{C} \)-Lipschitz continuity. \( H(\text{vec}(r_t)) \) is assumed to be \( \tilde{C} \). Lipschitz continuous with respect to \( r_t \). Formally, there exists a constant \( \tilde{C} > 0 \) such that

\[
|H(\text{vec}(r_t)) - H(\text{vec}(r'_t))| \leq \tilde{C} \sqrt{\sum_{k=1}^{K} (r^k_t - r'^k_t)^2}.
\]

Both assumptions are mild. For (1), if the input is zero, then the reward should also be zero. For (2), the Lipschitz continuity can be applied to many real-world applications. For the convenience of presentation, given any set of selected \( K \) arms \( X_t \), we denote the expectation of \( R_t \) by:

\[
\mathcal{H}(X_t) = E[R_t|X_t] = H\left(h_1(x^r_t), \ldots, h_K(x^K_t)\right).
\]

Recall that in multi-facet bandits, the learner aims to select the optimal \( K \) arms with the maximal final reward \( R^*_t \) in each round. First, we need to identify all possible combinations of \( K \) arms, denoted by

\[
S_t = \{(x^1_t, \ldots, x^K_t) | x^k_t \in X^k_t, k \in [K]\},
\]

where \( |S_t| = \prod_{k=1}^{K} n_k \) because bandit \( k \) has \( n_k \) arms for each \( k \in [K] \). Then, the regret of multi-facet bandit problem is defined as

\[
\text{Reg} = E\left[\sum_{t=1}^{T} (R^*_t - R_t)\right] = \sum_{t=1}^{T} (\mathcal{H}(X^*_t) - \mathcal{H}(X_t)),
\]

where \( X^*_t = \arg \max_{X_t} \mathcal{H}(X_t) \). Therefore, our goal is to design a bandit algorithm to select \( K \) arms in every round in order to minimize the regret. We use the standard \( O \) to hide constants and \( \tilde{O} \) to hide logarithm.

**Availability of sub-rewards.** In this framework, the final \( R_t \) is required to be known, while the sub-rewards \( r_t \) are allowed to be partially available. Because the feedback of some bandits cannot be directly measured or is simply not available in a real problem. This increases the flexibility of our proposed framework.

More specifically, in each round \( t \), ideally, the learner is able to receive \( K + 1 \) rewards including \( K \) sub-rewards \( \{r^1_t, \ldots, r^K_t\} \) and a final reward \( R_t \). As the final reward is the integral feedback of the entire group of bandits and reflects how the bandits affect each other, \( R_t \) is required to be known. However, the \( K \) sub-rewards are allowed to be partially available, because not every sub-reward is easy to obtain or can be measured accurately.

This is a new challenge in the multi-facet bandit problem. Thus, to learn \( \mathcal{H} \), the designed bandit algorithm is required to handle the partial availability of sub-rewards.

### 4 PROPOSED ALGORITHM

In this section, we introduce the proposed algorithm, MuFasa. The presentation of MuFasa is divided into three parts. First, we present the neural network model used in MuFasa; Second, we detail how to collect training samples to train the model in each round; In the end, we describe the UCB-based arm selection criterion and summarize the workflow of MuFasa.

#### 4.1 Neural network model

To learn the reward function \( \mathcal{H} \), we use \( K + 1 \) fully-connected neural networks to learn \( K \) bandits jointly, where a neural network \( f_k \) is built for each bandit \( k \in [K] \) to learn its reward function \( h_k \), and a shared neural network \( F \) is constructed to learn the mapping from the \( K \) neural networks \( \{f_1, \ldots, f_K\} \) to the final reward \( R_t \).

First, in round \( t \), for each bandit \( k \in [K] \), given any context vector \( x^k_t \in \mathbb{R}^{d_k} \), we use a \( L_1 \)-layer fully-connected network to learn \( h_k \), denoted by \( f_k \):

\[
f_k(x^k_t; \theta^k) = \sqrt{m_1}W_{1_k}\sigma(W_{1_k-1}\sigma(\ldots(\sigma(W_{1_k}x^k_t)))).
\]

where \( \sigma(x) \) is the rectified linear unit (ReLU) activation function.

Without loss of generality, we assume each layer has the same width \( m_1 \) for the sake of analysis. Therefore, \( \theta^k = (\text{vec}(W_{1_k})^T, \ldots, \text{vec}(W_{L_k})^T)^T \in \mathbb{R}^{L_k} \), where \( W_{1_k} \in \mathbb{R}^{m_1 \times d_k}, W_{i_k} \in \mathbb{R}^{m_{i_k} \times m_{i_k-1}}, \forall i \in [1 : L_k - 1] \), and \( W_{L_k} \in \mathbb{R}^{m_{L_k} \times m_1} \). Note that \( f_k(x^k_t; \theta^k) \in \mathbb{R}^{m_1} \), where \( m_1 \) is set as a tunable parameter to connect with the following network \( F \). Denote the gradient \( \nabla_{\theta^k} f_k(x^k_t; \theta^k) \) by \( g(x^k_t; \theta^k) \).

Next, to learn the final reward function \( H \), we use a \( L_2 \)-layer fully-connected network to combine the outputs of the above \( K \) neural networks, denoted by \( F \):

\[
F(f_1; \theta^1) = \sqrt{m_2}W_{L_2}\sigma(\ldots(\sigma(W_{1_k}f_k)))
\]

where \( f_t = (f_1(x^1_t; \theta^1), \ldots, f_k(x^k_t; \theta^k))^T \in \mathbb{R}^{m_{1+K}} \). Also, we assume that each layer has the same width \( m_2 \). Therefore, \( \theta^2 = (\text{vec}(W_{L_2})^T, \ldots, \text{vec}(W_{L_k})^T)^T \in \mathbb{R}^{L_k}, \) where \( W_{1_k} \in \mathbb{R}^{m_{1+K} \times m_2}, W_{i_k} \in \mathbb{R}^{m_{i_k+1} \times m_{i_k}}, \forall i \in [L_k - 1] \), and \( W_{L_k} \in \mathbb{R}^{L_{K+1} \times m_{L_k}} \). Denote the gradient \( \nabla_{\theta^2} F(f_1; \theta^2) \) by \( G(f_1; \theta^2) \).

Therefore, for the convenience of presentation, the whole assembled neural networks can be represented by \( F \) to learn \( \mathcal{H} \) (Eq.6),
where

\[ \text{Algorithm 1 MuFasa} \]

**Input:** \( \mathcal{F}, T, K, \delta, \eta, J \)
1. Initialize \( \theta_0 = (\theta_0^1, \cdots, \theta_0^K) \)
2. for each \( t \in [T] \) do
3. for each bandit \( k \in [K] \) do
4. Observe context vectors \( X_t^k = \{x_{t,n}^k \} \)
5. Collect \( S_t \) (Eq. (2))
6. Choose \( K \) arms, \( X_t \), by:
7. Play \( X_t \) and observe rewards \( R_t \) and \( r_t \).
8. if \( |r_t| = K \) then \# sub-rewards are all available.
9. \( \theta_t = \text{GradientDescentAll}(\mathcal{F}, \{X_t^i\}_{i=1}^T, \{R_t\}_{t=1}^T, \{r_t\}_{t=1}^T, \{\Omega_t\}_{t=1}^T, J, \eta) \)
10. else \# sub-rewards are partially available.
11. Collect \( \{\Omega_t\}_{t=1}^T \) (Eq. (4))
12. \( \theta_t = \text{GradientDescentPartial}(\mathcal{F}, \{\Omega_t\}_{t=1}^T, J, \eta) \)
13. Update UCB \( X_t^k \).

4.2 Training process

Only with the final reward \( R_t \) and \( K \) selected arms \( X_t \), the training of the neural network model \( \mathcal{F} \) is the following minimization problem:

\[ \min_{\theta} \mathcal{L}(\theta) = \sum_{t=1}^{T} (\mathcal{F}(X_t; \theta) - R_t)^2 / 2 + m_2 \lambda \| \theta_0 \|_2^2 / 2. \]  (3)

where \( \mathcal{L}(\theta) \) is essentially \( L_2 \)-regularized square loss function and \( \theta_0 \) is the randomly initialized network parameters. However, once the sub-rewards are available, we should use different methods to train \( \mathcal{F} \), in order to leverage more available information. Next, we will elaborate our training methods using the gradient descent.

Collection of training samples. Depending on the availability of sub-rewards, we apply different strategies to update \( \theta \) in each round. When the sub-rewards are all available, the learner receives one final reward and \( K \) sub-rewards. We apply the straightforward way to train each part of \( \mathcal{F} \) accordingly based on the corresponding input and the ground-truth rewards in each round, referring to the details in Algorithm 2, where \( \tilde{m} \in \mathcal{F} \) should be set as 1.

However, when the sub-rewards are partially available, the above method is not valid anymore because the bandits without available sub-rewards cannot be trained. Therefore, to learn the \( K \) bandits jointly, we propose the following training approach focusing on the empirical performance.

As the final reward is always available in each round, we collect the first training sample \( (X_t, R_t) \). Then, suppose there are \( K \) available sub-rewards \( r_t, K < K \). For each available sub-reward \( r_t^k \in r_t \) and the corresponding context vector \( x_t^k \), we construct the following pair:

\[ \tilde{X}_{t,k} = \{0, \ldots, x_t^k, \ldots, 0 \} \quad \text{and} \quad \tilde{r}_{t,k} = \{0, \ldots, r_t^k, \ldots, 0 \}. \]

We regard \( \tilde{X}_{t,k} \) as a new input of \( \mathcal{F} \). Now, we need to determine the ground-truth final reward \( H(\tilde{X}_{t,k}) = H(\text{vec}(\tilde{r}_{t,k})) \).

Unfortunately, \( H(\text{vec}(\tilde{r}_{t,k})) \) is unknown. Inspired by the UCB strategy, we determine \( H(\text{vec}(\tilde{r}_{t,k})) \) by its upper bound. Based on Lemma 4.1, we have \( H(\text{vec}(\tilde{r}_{t,k})) \leq \tilde{C}r_t^k \). Therefore, we set \( H(\text{vec}(\tilde{r}_{t,k})) \) as:

\[ H(\text{vec}(\tilde{r}_{t,k})) = \tilde{C}r_t^k \]

because it shows the maximal potential gain for the bandit \( k \). Then, in round \( t \), we can collect additional \( K \) sample pairs:

\[ \{ (\tilde{X}_{t,k}, \tilde{C}r_t^k) \}_{k \in [K]} \]

where \( [K] \) denotes the bandits with available sub-rewards.

Accordingly, in each round \( t \), we can collect up to \( K + 1 \) samples for training \( \mathcal{F} \), denoted by \( \Omega_t \):

\[ \Omega_t = \{ (\tilde{X}_{t,k}, \tilde{C}r_t^k) \} \cup \{ (X_t, R_t) \}. \]  (4)

---

**Algorithm 2 GradientDescentAll**

**Input:** \( \mathcal{F}, \{X_t^i\}_{i=1}^T, \{R_t^i\}_{i=1}^T, \{r_t^i\}_{i=1}^T, J, \eta \)

**Output:** \( \theta_t \)
1. for each \( k \in [K] \) do
2. Define \( \mathcal{L}(\theta_t^k) = \sum_{j=1}^{J} (f_k(x_t^j; \theta_t^k) - r_t^j)^2 / 2 + m_1 \lambda \| \theta_t^k - \theta_0^k \|_2^2 / 2 \)
3. for each \( j \in [J] \) do
4. \( \theta_t^k = \theta_t^{k-1} - \eta \nabla \mathcal{L}(\theta_t^{k-1}) \)
5. Define \( \mathcal{L}(\theta_t^2) = \sum_{j=1}^{J} (F(\text{vec}(r_t); \theta_t^2) - R_t)^2 / 2 + m_2 \lambda \| \theta_t^2 - \theta_0^2 \|_2^2 / 2 \)
6. for each \( j \in [J] \) do
7. \( \theta_t^2 = \theta_t^{2-1} - \eta \nabla \mathcal{L}(\theta_t^{2-1}) \)
8. return \( (\theta_t^k, \theta_t^1, \cdots, \theta_t^K) \)

**Algorithm 3 GradientDescentPartial**

**Input:** \( \mathcal{F}, \{\Omega_t^i\}_{i=1}^T, J, \eta \)

**Output:** \( \theta_t \)
1. Define \( \mathcal{L}(\theta_t) = \sum_{(X_t^i, R_t^i) \in \Omega_t} (\mathcal{F}(X_t^i; \theta) - R_t)^2 / 2 + m_2 \lambda \| \theta_t - \theta_0 \|_2^2 / 2 \)
2. for each \( j \in [J] \) do
3. \( \theta_t^j = \theta_t^{j-1} - \eta \nabla \mathcal{L}(\theta_t^{j-1}) \)
4. return \( \theta_t^j \)
Therefore, in each round, we train $F$ integrally, based on $\{\Omega_i\}_{i=1}^T$, as described in Algorithm 3.

**Lemma 4.1.** Let $0 = (0, \ldots, 0)$ and $|0| = K$. Given $X_{t,k}$ and $\bar{F}_{t,k}$, then we have $H(H(\vec{r}_{t,k})) \leq C r^k_t$.

**Proof 4.1.** As $H$ is $C$-Lipschitz continuous, we have

$$|H(H(\vec{r}_{t,k})) - H(0)| \leq C \sum_{r \in r_{t,k}} (r - 0)^2 = C r^k_t.$$

### 4.3 Upper confidence bound

In this subsection, we present the arm selection criterion based on the upper confidence bound provided in Section 5 and then summarize the high-level idea of MuFasa.

In each round $t$, given an arm combination $X_t$, the confidence bound of $F$ with respect to $\mathcal{H}$ is defined as:

$$\mathbb{P}(|F(X_t; \theta_t) - H(X_t)| > UCB(X_t)) \leq \delta,$$

where UCB($X_t$) is defined in Theorem 5.3 and $\delta$ usually is a small constant. Then, in each round, given all the possible arm combinations $S_t$, the selected $K$ arms $X_t$ are determined by:

$$X_t = \arg \max_{X_t' \in S_t} (F(X_t'; \theta_t) + UCB(X_t')).$$

(5)

With this selection criterion, the workflow of MuFasa is depicted in Algorithm 1.

### 5 REGRET ANALYSIS

In this section, we provide the upper confidence bound and regret analysis of MuFasa when the sub-rewards are all available.

Before presenting Theorem 5.3, let us first focus on an $L$-layer fully-connected neural network $f(x_t; \theta_t)$ to learn a ground-truth function $h(x_t)$, where $x \in \mathbb{R}^d$. The parameters of $f$ are set as $W_1 \in \mathbb{R}^{m \times d}$, $W_l \in \mathbb{R}^{m \times m}$, $\forall l \in [1 : L - 1]$, and $W_L \in \mathbb{R}^{1 \times m}$. Given the context vectors by $x_t \in T$ and corresponding rewards $\{r_t\}_{t=1}^T$, conduct the gradient descent with the loss $L$ to train $f$.

Built upon the Neural Tangent Kernel (NTK) [15, 24], $h(x_t)$ can be represented by a linear function with respect to the gradient $g(x_t; \theta_t)$ introduced in [45] as the following lemma.

**Lemma 5.1 (Lemma 5.1 in [45]).** There exist a positive constant $C$ such that with probability at least $1 - \delta$, if $m \geq C T^4 L^6 \log(T^2L/\delta)/\lambda^4$ for any $x_t \in \{x_t\}_{t=1}^T$, there exists a $\theta^*$ such that

$$h(x_t) = \langle g(x_t; \theta_0), \theta^* \rangle$$

Then, with the above linear representation of $h(x_t)$, we provide the following upper confidence bound with regard to $f$.

**Lemma 5.2.** Given a set of context vectors $\{x_t\}_{t=1}^T$ and the corresponding rewards $\{r_t\}_{t=1}^T$, $\mathbb{E}(r_t) = h(x_t)$ for any $x_t \in \{x_t\}_{t=1}^T$. Let $f(x_0; \theta_t)$ be the $L$-layers fully-connected neural network where the width is $m$, the learning rate is $\eta$, and $\theta \in \mathbb{R}^F$. Assuming $\|\theta^*\|_2 \leq S/\sqrt{m}$, then, there exist positive constants $C_1, C_2$ such that if

$$m \geq \max\{O(T^7 \lambda^{-7} L^2 (\log m)^3), O(\lambda^{-1/2} L^{-3/2} (\log(TL^2/\delta)^{3/2})\}$

$$\eta = O(T \lambda mL + m)^{-1}, J \geq O(T \lambda L),$$

then, with probability at least $1 - \delta$, for any $x_t \in \{x_t\}_{t=1}^T$, we have the following upper confidence bound:

$$|h(x_t) - f(x_t; \theta_t)| \leq y_1 \|g(x_t; \theta_t)\|_{\mathcal{A}^*} + y_2 \|g(x_t; \theta_0)\|_{\mathcal{A}_t^*} + \gamma_1 y_3 + y_4,$$

where

$$y_1(m, L) = (\lambda + tO(L)) - ((1 - \eta m \lambda)^{1/2} \sqrt{\eta/\lambda}) + 1$$

$$y_2(m, L, \delta) = \log \frac{\det(A_t^*)}{\det(\lambda)} - 2 \log \delta + \lambda^{1/2} S$$

$$y_3(m, L) = C_2 m^{-1/6} \sqrt{\log m L^{1/6}} L^{7/2}$$

$$y_4(m, L) = C_1 m^{-1/6} \sqrt{\log m L^{2/3}} L^{2/3}$$

$$A_t = \lambda I + \sum_{i=1}^t g(x_i; \theta_t) g(x_i; \theta_t)^T / m$$

$$A_t^* = \lambda I + \sum_{i=1}^t g(x_i; \theta_0) g(x_i; \theta_0)^T / m.$$

Now we are ready to provide an extended upper confidence bound for the proposed neural network model $F$.

**Theorem 5.3.** Given the selected contexts $\{X_t\}_{t=1}^T$, the final rewards $\{R_t\}_{t=1}^T$, and all sub-rewards $\{r_t\}_{t=1}^T$, let $F$ be the neural network model in MuFasa. In each round $t$, with the conditions in Lemma 5.2 and suppose $m = 1$, then, with probability at least $1 - \delta$, for any $t \in [T]$, we have the following upper confidence bound:

$$|F(X_t; \theta_t) - H(X_t)| \leq C \sum_{k=1}^{K} B^k + B^F = UCB(X_t),$$

where

$$B^k = y_1 \|g_k(x_k^t; \theta^*_t)\|_{\mathcal{A}_t^*} + y_2 \|g_k(x_k^t; \theta_0)\|_{\mathcal{A}_t^*} + y_1 y_3 + y_4$$

$$B^F = y_1 \|G(f; \theta^*_t)\|_{\mathcal{A}_t^*} + y_2 \|G(f; \theta_0)\|_{\mathcal{A}_t^*} + y_1 y_3 + y_4$$

$$A^*_t = \lambda I + \sum_{i=1}^t g_k(x_k^t; \theta^*_t) g_k(x_k^t; \theta^*_t)^T / m_1$$

$$A^*_t = \lambda I + \sum_{i=1}^t g_k(x_k^t; \theta^*_t) g_k(x_k^t; \theta^*_t)^T / m_2$$

$$A^*_t = \lambda I + \sum_{i=1}^t G(f; \theta^*_t) G(f; \theta^*_t)^T / m_2.$$

With the above UCB, we provide the following regret bound of MuFasa.

**Theorem 5.4.** Given the number of rounds $T$ and suppose that the final reward and all the sub-rewards are available, let $F$ be the neural network model of MuFasa, satisfying the conditions in Theorem 5.3. Then, assuming $C = 1$, $m_1 = m_2 = m$, $L_1 = L_2 = L$, and thus
$P_1 = P_2 = P$, with probability at least $1 - \delta$, the regret of MuFasa is upper bounded by:

$$\text{Reg} \leq (\tilde{C}K + 1) \sqrt{T} \sqrt{P} \log (1 + T/\lambda) + 1/\lambda + 1$$

$$\cdot \left(\sqrt{\left(\tilde{P} - 2\right) \log \left(\frac{(\lambda + T)(1 + K)}{\lambda \delta}\right)} + 1/\lambda + \lambda^{1/2} S + 2\right) + 2(\tilde{C}K + 1),$$

where $\tilde{P}$ is the effective dimension defined in Appendix (Definition 8.3).

**Prove 5.4.** First, the regret of one round $t$:

$$\text{Reg}_t = \mathcal{H}(X_t^*) - \mathcal{H}(X_t)$$

$$\leq |\mathcal{H}(X_t^*) - \mathcal{F}(X_t^*)| + |\mathcal{F}(X_t^*) - \mathcal{H}(X_t)|$$

$$\leq \text{UCB}(X_t^*) + |\mathcal{F}(X_t^*) - \mathcal{H}(X_t)|$$

$$\leq \text{UCB}(X_t) + |\mathcal{F}(X_t) - \mathcal{H}(X_t)| \leq 2\text{UCB}(X_t)$$

where the third inequality is due to the selection criterion of MuFasa, satisfying $\mathcal{F}(X_t^*) + \text{UCB}(X_t^*) \leq \mathcal{F}(X_t) + \text{UCB}(X_t)$. Thus, it has

$$\text{Reg} = \sum_{t=1}^{T} \text{Reg}_t \leq 2 \sum_{t=1}^{T} \text{UCB}(X_t) \leq 2 \sum_{t=1}^{T} \left(\tilde{C}K \cdot x^k + g^k\right)$$

First, for any $k \in [K]$, we bound

$$\sum_{t=1}^{T} x^k \leq y_1 \sum_{t=1}^{T} \|g(x_t; \theta_t)/\sqrt{m}\|_{A_t^{-1}}^{2} + y_2 \sum_{t=1}^{T} \|g(x_t; \theta_0)/\sqrt{m}\|_{A_t^{-1}}^{2}$$

$$+ T_{1,2} + T_{4}$$

Because the Lemma 11 in [1], we have

$$I_1 \leq y_1 \sqrt{T \left(\sum_{t=1}^{T} \|g(x_t; \theta_t)/\sqrt{m}\|_{A_t^{-1}}^{2}\right)} \leq y_1 \sqrt{T \left(\log (\det(A_t^T) / \det(\lambda I))\right)}$$

$$\leq y_1 \sqrt{T \left(\log (\det(A_t^T) / \det(\lambda I)) + \log (\det(A_t^T) / \det(\lambda I))\right)}$$

$$\leq y_1 \sqrt{T \left(\tilde{P} \log (1 + T/\lambda) + 1/\lambda + 1\right)}$$

where the last inequality is based on Lemma 8.4 and the choice of $m$. Then, applying Lemma 11 in [1] and Lemma 8.4 again, we have

$$I_2 \leq y_2 \sqrt{T \left(\log (\det(A_t^T) / \det(\lambda I))\right)}$$

$$\leq \sqrt{T \left(\tilde{P} - 2\right) \log \left(\frac{(\lambda + T)(1 + K)}{\lambda \delta}\right)} + 1/\lambda + \lambda^{1/2} S$$

As the choice of $J$, $y_1 \leq 2$. Then, as $m$ is sufficiently large, we have

$$T_{1,2} \leq 1, T_{4} \leq 1.$$ Then, because $m_1 = m_2$, $L_1 = L_2$ and $\tilde{C} = 1$, we have

$$\text{Reg} \leq 2(\tilde{C}K + 1) \sum_{t=1}^{T} g^k.$$ Putting everything together proves the claim.

$\tilde{P}$ is the effective dimension defined by the eigenvalues of the NTK (Definition 8.3 in Appendix). Effective dimension was first introduced by [39] to analyze the kernelized context bandit, and then was extended to analyze the kernel-based Q-learning [41] and the neural-network-based bandit [45]. $\tilde{P}$ can be much smaller than the real dimension $P$, which alleviates the predicament when $P$ is extremely large.

Theorem 5.4 provides the $\tilde{O}((K + 1) \sqrt{T})$ regret bound for MuFasa, achieving the near-optimal bound compared with a single bandit ($O(\sqrt{T})$) that is either linear [1] or non-linear [39, 45]. With different widths $m_1, m_2$ and the Lipschitz continuity $C$, the regret bound of MuFasa becomes $\tilde{O}((\tilde{C}K + 1) \sqrt{T})$.

### 6 EXPERIMENTS

To evaluate the empirical performance of MuFasa, in this section, we design two different multi-facet bandit problems on three real-world data sets. The experiments are divided into two parts to evaluate the effects of final rewards and availability of sub-rewards. The code has been released 1.

**Recommendation:Yelp**. Yelp is a data set released in the Yelp data set challenge, which consists of 4.7 million rating entries for 1.57 $\times$ 10^5 restaurants by 1.18 million users. In addition to the features of restaurants, this data set also provides the attributes of each user and the list of his/her friends. In this data set, we evaluate MuFasa on personalized recommendation, where the learner needs to simultaneously recommend a restaurant and a friend (user) to a served user. Naturally, this problem can be formulated into 2 bandits in which one set of arms $X_t^j$ represent the candidate restaurants and the other set of arms $X_t^j$ formulates the pool of friends for the recommendation. We apply LocallyLinearEmbedding [35] to train a 10-dimensional feature vector $x^j_t$ for each restaurant and a 6-dimensional feature vector $x^j_t$ for each user. Then, for the restaurant, we define the reward according to the rating star: The reward $r^j_t$ is 1 if the number of rating stars is more than 3 (5 in total); Otherwise, the reward $r^j_t$ is 0. For friends, the reward $r^j_t = 1$ if the recommended friend is included in the friend list of the served user in fact; Otherwise $r^j_t = 0$. To build the arm sets, we extract the rating entries and friends lists of top-10 users with the most ratings. In each round $t$, we build the arm set $X_t^j$ and $X_t^j$ by picking one restaurant/friend with 1 reward and then randomly picking the other 9 restaurants/friends with 0 rewards. Thus $|X_t^j| = |X_t^j| = 10$.

**Classification:MNIST** [27] + **NotMNIST**. These are two well-known 10-class classification data sets. The evaluation of contextual bandit has been adapted to the classification problem [18, 39, 45]. Therefore, we utilize these two similar classification data sets to construct 2 bandits, where the 10-class classification is converted into a 10-armed contextual bandit. Considering a sample figure $x \in \mathbb{R}^d$, we tend to classify it from 10 classes. Under the contextual bandit setting, $x$ is transformed into 10 arms: $x_1 = (0, 0, \ldots, 0); x_2 = (1, 0, \ldots, 0); \ldots; x_{10} = (0, 0, \ldots, 0) \in \mathbb{R}^{10d}$ matching the 10 classes. In consequence, the reward is 1 if the learner plays the arm that

1https://github.com/banyikun/KDD2021_MuFasa
2https://www.yelp.com/dataset
matches the real class of \(x\); Otherwise, the reward is 0. Using this way, we can construct two contextual bandits for these two data sets, denoted by \((X^1_t, r^1_t)\) and \((X^2_t, r^2_t)\). Then, in each round, the arm pools will be \(|X^1_t| = |X^2_t| = 10\).

To evaluate the effect of different final reward function, with the sub-rewards \(r_t = \{r^1_t, r^2_t\}\), we design the following final reward function:

\[
H_1(\text{vec}(r_t)) = r^1_t + r^2_t; \quad H_2(\text{vec}(r_t)) = 2r^1_t + r^2_t.
\]

(6)

For (1), it describes the task where each bandit contributes equally. Of (2), it represents some tasks where each bandit has different importance.

As the problem setting is new, there are no existing algorithms that can directly adapt to this problem. Therefore, we construct baselines by extending the bandit algorithms that work on a single bandit, as follows:

1. **(K-)LinUCB.** LinUCB [28] is a linear contextual bandit algorithm where the reward function is assumed as the dot product of the arm feature vector and an unknown user parameter. Then, apply the UCB strategy to select an arm in each round. To adapt to the multi-facet bandit problem, we duplicate LinUCB for \(K\) bandits. For example, in Yelp data set, we use two LinUCB to recommend restaurants and friends, respectively.

2. **(K-)KerUCB.** KerUCB [39] makes use of a predefined kernel matrix to learn the reward function and then build a UCB for exploration. We replicate \(K\) KerUCB to adapt to this problem.

3. **(K-)NeuUCB.** NeuUCB [45] uses a fully-connected neural network to learn one reward function with the UCB strategy. Similarly, we duplicate it to \(K\) bandits.

**Configurations.** For MuFasa, each sub-network \(f_k(x^k; \theta^k)\) is set as a two-layer network: \(f_k(x^k; \theta^k) = \sqrt{m_1}W_2\sigma(W_1x^k)\), where \(W_1 \in \mathbb{R}^{m_1 \times d_k}, W_2 \in \mathbb{R}^{m_2 \times 1}\) and \(m_1 = m = 100\) for \(H_1\), \(C\) is set as 1 and set as 2 for \(H_2\). To learn \(K\) bandits jointly, in the experiments, we evaluate the performance of Algorithm 1 + 3. For \(K\)-NeuUCB, for each NeuUCB, we set it as a 4-layer fully-connected network with the same width \(m = 100\) for the fair comparison. The learning rate \(\eta\) is set as 0.01 and the upper bound of ground-truth parameter \(S = 1\) for these two methods. To accelerate the training process, we update the parameters of the neural networks every 50 rounds. For the KerUCB, we use the radial basis function (RBF) kernel and stop adding contexts to KerUCB after 1000 rounds, following the same setting for Gaussian Process in [34, 45]. For all the methods, the confidence level \(\delta = 0.1\), the regularization parameter \(\lambda = 1\). All experiments are repeatedly run 5 times and report the average results.

### 6.1 Result 1: All sub-rewards with different \(H\)

With the final reward function \(H_i\) (Eq.(6)), Figure 1 and Figure 3 report the regret of all methods on Yelp and Mnist+NotMnist data sets, where the first two top-sub-figure shows the regret of each bandit and the bottom sub-figure shows the regret of the whole task (2 bandits). These figures show that MuFasa achieves the best performance (the smallest regret), because it utilizes a comprehensive upper confidence bound built on the assembled neural networks to select two arms jointly in each round. This indicates the good performance of MuFasa on personalized recommendation and classification. Among these baselines, NeuUCB achieves the best performance, which thanks to the representation power of neural networks. However, it chooses each arm separately, neglecting the collaborative relation of \(K\) bandits. For KerUCB, it shows the limitation of the simple kernels like the radial basis function compared to neural network. LinUCB fails to handle each task, as it assume a linear reward function and thus usually cannot to learn the complicated reward functions in practice.

With the final reward function \(H_2\) (Eq.(6)), Figure 2 and Figure 4 depict the regret comparison on Yelp and Mnist+NotMnist data sets. The final reward function \(H_2\) indicates that the bandit 1 weights more than bandit 2 in the task. Therefore, to minimize the regret, the algorithm should place the bandit 1 as the priority when making decisions. As the design of MuFasa, the neural network \(F\) can learn the relation among the bandits. For example, on the Mnist and NotMnist data sets, consider two optional select arm sets \(\{x^1_t, x^2_t\}\) and \(\{x^3_t, x^4_t\}\). The first selected arm set receives 1 reward on Mnist while 0 reward on NotMnist. In contrast, the second selected arm set receives 0 reward on Mnist while 1 reward on NotMnist. However, these two bandits have different weights.
Regret

Round

Regret

Round Round

Figure 3: Regret comparison on Mnist+NotMnist with $H_1$.

Figure 4: Regret comparison on Mnist+NotMnist with $H_2$.

and thus the two arm sets have different final rewards, i.e., $R^1_t = 2$ and $R^2_t = 1$, respectively. To maximize the final reward, the learner should select the first arm set instead of the second arm set. As MuFasa can learn the weights of bandits, it will give more weight to the first bandit and thus select the first arm set. On the contrary, all the baselines treat each bandit equally, and thus they will select these two arm sets randomly. Therefore, under the setting of $H_2$, with this advantage, MuFasa further decreases the regret on both Yelp and Mnist+NotMnist data sets. For instance, on Mnist+NotMnist data sets, MuFasa with $H_2$ decrease 20% regret over NeuUCB while MuFasa with $H_1$ decrease 17.8% regret over NeuUCB.

6.2 Result 2: Partial sub-rewards

As the sub-rewards are not always available in many cases, in this subsection, we evaluate MuFasa with partially available sub-rewards on Yelp and Mnist+NotMnist data sets. Therefore, we build another two variants of MuFasa: (1) MuFasa (One sub-reward) is provided with the final reward and one sub-reward of the first bandit; (2) MuFasa (No sub-reward) does not receive any sub-rewards except the final reward. Here, we use the $H_1$ as the final reward function.

Figure 5 and Figure 6 show the regret comparison with the two variants of MuFasa, where MuFasa exhibits the robustness with respect to the lack of sub-rewards. Indeed, the sub-reward can provide more information to learn, while MuFasa (One sub-reward) still outperforms all the baselines, because the final reward enables MuFasa to learn the all bandits jointly and the sub-reward strengthens the capacity of learning the exclusive features of each bandit.

In contrast, all the baselines treat each bandit separately. Without any-rewards, MuFasa still achieves the acceptable performance. On the Yelp data set, the regret of MuFasa (No sub-reward) is still lower than the best baseline NeuUCB while lacking considerable information. On the Mnist+NotMnist data set, although MuFasa (No sub-reward) does not outperform the baselines, its performance is still closed to NeuUCB. Therefore, as long as the final reward is provided, MuFasa can tackle the multi-facet problem effectively. Moreover, MuFasa can leverage available sub-rewards to improve the performance.

7 CONCLUSION

In this paper, we define and study the novel problem of the multi-facet contextual bandits, motivated by real applications such as comprehensive personalized recommendation and healthcare. We propose a new bandit algorithm, MuFasa. It utilizes the neural networks to learn the reward functions of multiple bandits jointly and explores new information by a comprehensive upper confidence bound. Moreover, we prove that MuFasa can achieve the $O((K + 1)\sqrt{T})$ regret bound under mild assumptions. Finally, we conduct extensive experiments to show the effectiveness of MuFasa on personalized recommendation and classification tasks, as well as the robustness of MuFasa in the lack of sub-rewards.
8 APPENDIX

Definition 8.1. Given the context vectors \( \{x_t\}_{t=1}^T \) and the rewards \( \{r_t\}_{t=1}^T \), then we define the estimation \( \theta' \) via ridge regression:

\[
A_t' = \lambda I + \sum_{i=1}^t g(x_t; \theta_0)g(x_t; \theta_0)\tau/m
\]

\[
b_t' = \sum_{i=1}^t r_t g(x_t; \theta_0)/\sqrt{m}
\]

\[
\theta' = A_t^{-1}b_t
\]

\[
A_t = \lambda I + \sum_{i=1}^t g(x_t; \theta_t)g(x_t; \theta_t)^\tau/m
\]

Definition 8.2 (NTK [15, 24]). Let \( \mathcal{N} \) denote the normal distribution. Define

\[
\mathcal{M}_{l,j}^0 = \Sigma_{l,j}^0 = (x_l, x_j),\quad \mathcal{N}_{l,j}^1 = \left( \Sigma_{l,i}^1 \Sigma_{i,j}^1 \right)
\]

\[
\Sigma_{l,j}^1 = 2\mathbb{E}_{a,b \sim \mathcal{N}(0, \Sigma_{l,j}^0)} [\sigma(a)\sigma(b)]
\]

\[
\mathcal{M}_{l,j}^1 = 2\mathcal{M}_{l,j}^0 \mathbb{E}_{a,b \sim \mathcal{N}(0, \Sigma_{l,j}^0)} [\sigma'(a)\sigma'(b)] + \Sigma_{l,j}^1.
\]

Then, given the contexts \( \{x_t\}_{t=1}^T \), the Neural Tangent Kernel is defined as \( \mathcal{M} = (\mathcal{M} + \Sigma^1)/2 \).

Definition 8.3 (Effective Dimension [45]). Given the contexts \( \{x_t\}_{t=1}^T \), the effective dimension \( \bar{p} \) is defined as

\[
\bar{p} = \frac{\log \det(1 + \mathcal{M}/\lambda)}{\log(1 + T/\lambda)}.
\]

Proof of Lemma 5.2. Given a set of context vectors \( \{x_t\}_{t=1}^T \) with the ground-truth function \( h \) and a fully-connected neural network \( f \), we have

\[
|h(x_t) - f(x_t; \theta_0)| \leq \left| h(x_t) - (g(x_t; \theta_0), \theta'/\sqrt{m}) \right| + \left| f(x_t; \theta_t) - (g(x_t; \theta_0), \theta'/\sqrt{m}) \right|
\]

where \( \theta' \) is the estimation of ridge regression from Definition 8.1. Then, based on the Lemma 5.1, there exists \( \theta' \in \mathbb{R}^p \) such that \( h(x_t) = (g(x_t; \theta_0), \theta') \). Thus, we have

\[
|h(x_t) - f(x_t; \theta_t)| \leq \left| (g(x_t; \theta_0)/\sqrt{m}, \theta'/\sqrt{m}) \right| - \left| (g(x_t; \theta_0)/\sqrt{m}, \theta') \right| \leq \sqrt{\frac{\log \det(A_t')}{\det(I)} - 2 \log 3 + \lambda^{1/2}S} \left| (g(x_t; \theta_0)/\sqrt{m}, \theta') \right|
\]

where the final inequality is based on the theorem 2 in [1], with probability at least \( 1 - \delta \), for any \( t \in [T] \).

Second we need to bound

\[
|f(x_t; \theta_t) - (g(x_t; \theta_0), \theta'/\sqrt{m})| \leq |f(x_t; \theta_t) - (g(x_t; \theta_0), \theta_t - \theta_0)| + \left| (g(x_t; \theta_t), \theta_t - \theta_0) - (g(x_t; \theta_0), \theta'/\sqrt{m}) \right|
\]

To bound the above inequality, we first bound

\[
|f(x_t; \theta_t) - (g(x_t; \theta_0), \theta_t - \theta_0)| \leq C2^{4/3}L^3\sqrt{m \log m} \leq C2^m \frac{1/2 - 1}{\sqrt{\log m} m^{2/3} \lambda^{-1/2} L^3}
\]

where \( f(x_t; \theta_0) = 0 \) due to the random initialization of \( \theta_0 \). The first inequality is derived by Lemma 8.5. According to the Lemma 8.7, it has \( \|\theta_t - \theta_0\|_2 \leq 2\sqrt{\frac{T}{4m}} \). Then, replacing \( r \) by \( 2\sqrt{\frac{T}{4m}} \), we obtain the second inequality.

Next, we need to bound

\[
| (g(x_t; \theta_0), \theta_t - \theta_0) - (g(x_t; \theta_0), \theta'/\sqrt{m}) | = |(g(x_t; \theta_0)/\sqrt{m}, \sqrt{m}(\theta_t - \theta_0) - \theta'/\sqrt{m}) | \leq \sqrt{|g(x_t; \theta_0)/\sqrt{m}|^2 + \sqrt{|\theta_t - \theta_0 - \theta'/\sqrt{m}|}|A_t|2} \cdot \|\theta_t - \theta_0 - \theta'/\sqrt{m}\|_2.
\]

Due to the Lemma 8.6 and Lemma 8.7, we have

\[
\sqrt{|\theta_t - \theta_0 - \theta'/\sqrt{m}| |A_t|2} \leq \sqrt{|\sqrt{m}\lambda + tO(L)} \cdot (1 - \eta m \lambda L/\sqrt{1 + T/\lambda}) \leq \lambda + tO(L) \cdot (1 - \eta m \lambda L/\sqrt{1 + T/\lambda}) + 1
\]

where the last inequality is because \( \lambda = m \) is sufficiently large. Therefore, we have

\[
\|h(x_t) - f(x_t; \theta_t)\|_2 \leq \left( \lambda + rO(L) \right) \cdot (1 - \eta m \lambda L/\sqrt{1 + T/\lambda}) + 1 \|g(x_t; \theta_0)/\sqrt{m}\|_{\mathcal{A}_t}^2 + 2C_2 m^{-1/6} \sqrt{\log m} L^{2/3} \lambda^{-1/2} L^3
\]

And we have

\[
\|g(x_t; \theta_0)/\sqrt{m}\|_{\mathcal{A}_t}^2 \leq \left| \left( (1 - \eta m \lambda L/\sqrt{1 + T/\lambda}) + 1 \right) \|g(x_t; \theta_0)/\sqrt{m}\|_{\mathcal{A}_t}^2 \right| \leq \left| \left( (1 - \eta m \lambda L/\sqrt{1 + T/\lambda}) + 1 \right) \|g(x_t; \theta_0)/\sqrt{m}\|_{\mathcal{A}_t}^2 \right|
\]

where the last inequality is because of Lemma 8.8 with Lemma 8.7 and \( \|\mathcal{A}_t\|_2 \geq (1 + T/\lambda). \)

Finally, putting everything together, we have

\[
|h(x_t) - f(x_t; \theta_t)| \leq \gamma_1 \|g(x_t; \theta_t)/\sqrt{m}\|_{\mathcal{A}_t}^2 + \gamma_2 \|g(x_t; \theta_0)/\sqrt{m}\|_{\mathcal{A}_t}^2 + \gamma_3 \gamma_4.
\]

Proof of Theorem 5.3. First, considering an individual bandit \( k \in [K] \) with the set of context vectors \( \{x_t^k\}_{t=1}^T \) and the set of sub-rewards \( \{r_t^k\}_{t=1}^T \), we can build upper confidence bound of \( f_k(x_t^k; \theta_t^k) \) with respect to \( h_k(x_t^k) \) based on the Lemma 5.2. Denote the UCB by \( \mathcal{B}(x_t^k, m, L_1, \delta') \), with probability at least \( 1 - \delta' \), for any \( t \in [T] \) we have

\[
\|f_k(x_t^k; \theta_t^k) - h_k(x_t^k)\| \leq \mathcal{B}(x_t^k, m, L_1, \delta', t) = \mathcal{B}_k^t
\]

Next, apply the union bound on the $K + 1$ networks, we have $\delta' = \delta/(K + 1)$ in each round $t$.

Next we need to bound

\[
\left| H(\text{vec}(r_t)) - H(f_t) \right| \\
\leq \tilde{C} \sum_{k=1}^{K} \parallel f_k(x_t^k; \theta_k^T) - h_k(x_t^k) \parallel^2 \leq \tilde{C} \sum_{k=1}^{K} \parallel b_k \parallel^2 \\
\leq \tilde{C} \sum_{k=1}^{K} \parallel b_k \parallel^2 \leq C \sum_{k=1}^{K} \parallel b_k \parallel^2
\]

where the first inequality is because $H$ is a $\tilde{C}$-lipschitz continuous function. Therefore, we have

\[
\left| \mathcal{F}(X_t) - \mathcal{H}(X_t) \right| = \left| \mathcal{F}(f_t; \theta^T) - H(\text{vec}(r_t)) \right| \\
\leq \left| \mathcal{F}(f_t; \theta^T) - H(f_t) \right| + \left| H(f_t) - H(\text{vec}(r_t)) \right| \leq \tilde{C} \sum_{k=1}^{K} \parallel b_k \parallel^2 + \parallel b \parallel^2.
\]

This completes the proof of the claim.

**Lemma 8.4.** With probability at least $1 - \delta'$, we have

\[
(1) \parallel A_1 \parallel_2, \parallel A_2' \parallel_2 \leq \lambda + tO(L),
\]

\[
(2) \log \frac{\det(A_2')}{\det(\lambda I)} \leq \tilde{P} \log(1 + T/\lambda) + 1/\lambda
\]

\[
(3) \log \frac{\det(A_1)}{\det(\lambda I)} = \frac{\det(A_1')}{\det(\lambda I)} \leq O(\sqrt{mL}).
\]

where (3) is referred from Lemma B.3 in [45].

**Proof of Lemma 8.4.** For (1), based on the Lemma 8.6, for any $x_t \in \{x_t\}_T^T$,

\[
\parallel g(x_t; \theta_0) \parallel_F \leq O(\sqrt{mL}).
\]

Then, for the first item:

\[
\parallel A_1 \parallel_2 = \parallel A + \sum_{t=1}^{T} g(x_t; \theta_t) g(x_t; \theta_t)^T / m \parallel_2 \\
\leq \parallel A \parallel_2 + \parallel \sum_{t=1}^{T} g(x_t; \theta_t) g(x_t; \theta_t)^T / m \parallel_2 \\
\leq \lambda + \sum_{t=1}^{T} \parallel g(x_t; \theta_t) \parallel_F^2 / m \leq \lambda + \sum_{t=1}^{T} \parallel g(x_t; \theta_t) \parallel_F^2 / m \\
\leq \lambda + tO(L).
\]

Same proof workflow for $\parallel A_2' \parallel_2$. For (2), we have

\[
\log \frac{\det(A_2')}{\det(\lambda I)} = \log \det(1 + \sum_{t=1}^{T} g(x_t; \theta_0) g(x_t; \theta_0)^T / (m \lambda)) \\
= \det(1 + GG^T / \lambda)
\]

where $G = (g(x_1; \theta_0), \ldots, g(x_T; \theta_0)) / \sqrt{m}$.

According to the Theorem 3.1 in [5], when $m = \Omega(\frac{L \log L / \delta}{\epsilon^2})$, with probability at least $1 - \delta$, for any $x_t, x_{t'} \in \{x_t\}_T^T$, it has

\[
\parallel g(x_t; \theta_0)^T g(x_{t'}; \theta_0) / m - M_{i,j} \parallel \leq \epsilon.
\]

Therefore, we have

\[
\parallel GG^T - M \parallel_F = \sqrt{\sum_{t=1}^{T} \sum_{j=1}^{J} g(x_t; \theta_0)^T g(x_{t'}, \theta_0) / m - M_{i,j}}^2 \\
\leq \epsilon.
\]

Then, we have

\[
\log \det(1 + GG^T / \lambda) = \log \det(1 + M \lambda + (GG^T - M) / \lambda) \\
\leq \log \det(1 + M \lambda) + \log(1 + M \lambda)^{-1} \parallel GG^T - M \parallel_F / \lambda \\
\leq \log \det(1 + M \lambda) + \lambda^{-1} \parallel GG^T - M \parallel_F / \lambda \\
\leq \log \det(1 + M \lambda) + \lambda^{-1} \parallel GG^T - M \parallel_F / \lambda
\]

The first inequality is because the concavity of $\log \det$ ; The third inequality is due to $\parallel (I + M \lambda)^{-1} / \lambda \parallel \leq \parallel I^{-1} \parallel / \lambda$; The last inequality is because of the choice of the $m$; The last equality is because of the Definition 8.3.

**Lemma 8.5 (Lemma 4.1 in [15]).** There exist constants $\{c_i\}_{i=1}^3 \geq 0$ such that for any $\delta \geq 0$, if $\tau$ satisfies that

\[
\tau \leq C_1 L^{-d} \log m^{-3/2},
\]

then with probability at least $1 - \delta$, for all $\theta^1, \theta^2$ satisfying $\parallel \theta^1 - \theta_0 \parallel \leq \tau$, $\parallel \theta^2 - \theta_0 \parallel \leq \tau$ and for any $x_t \in \{x_t\}_T^T$, we have

\[
\parallel f(x; \theta^1) - f(x; \theta^2) - \langle g(x; \theta^1), \theta^1 - \theta^2 \rangle \parallel \leq C_4 \tau^{3/2} \sqrt{m} \log m.
\]

**Lemma 8.6 (Lemma B.3 in [15]).** There exist constants $\{c_i\}_{i=1}^2$ such that for any $\delta \geq 0$, if $\tau$ satisfies that

\[
\tau \leq C_2 L^{-d} \log m^{-3/2},
\]

then, with probability at least $1 - \delta$, for any $\parallel \theta - \theta_0 \parallel \leq \tau$ and $x_t \in \{x_t\}_T^T$ we have

\[
\parallel g(x_t; \theta) \parallel_F \leq \sqrt{m}.
\]

**Lemma 8.7 (Lemma B.2 in [15]).** For the L-layer full-connected network $f$, there exist constants $\{c_i\}_{i=1}^5 \geq 0$ such that for $\delta > 0$, if for all $t \in [T]$, $\eta, m$ satisfy

\[
2 \sqrt{\tau/(m \lambda)} \geq C_1 m^{-3/2} L^{-3/2} \log(\tau L^2 / \delta)^{3/2},
\]

\[
2 \sqrt{\tau/(m \lambda)} \leq C_2 \min\{L^{-6} \log m^{-3/2}, (m \eta)^2 L^{-6} \log m^{-1} \}^{3/8},
\]

\[
\eta \leq C_3 (m \lambda + m \eta L)^{-1},
\]

\[
m^{1/6} \geq C_4 \sqrt{\tau/(m \lambda)} L^{7/6} \lambda^{-7/6} (1 + \sqrt{\tau / \lambda}),
\]

then, with probability at least $1 - \delta$, it has

\[
\parallel \theta - \theta_0 \parallel \leq 2 \sqrt{\tau/(m \lambda)} \\
\parallel \theta - \theta_0 - \theta' \parallel \leq (1 - \eta m L)^{1/2} \sqrt{\tau/(m \lambda)} + C_3 m^{-2/3} \sqrt{\log m L^{7/2} \lambda^{-5/3} (1 + \sqrt{\tau / \lambda})}.
\]

**Lemma 8.8 (Theorem 5 in [3]).** With probability at least $1 - \delta$, there exist constants $C_1, C_2$ such that if $\tau \leq C_1 L^{-d} \log m$, for any $\parallel \theta - \theta_0 \parallel \leq \tau$, we have

\[
\parallel g(x_t; \theta_1) - g(x_t; \theta_0) \parallel_2 \leq C_2 \sqrt{\tau^{1/2} L^2 \log m L^{7/2} \log m L^{7/2}} \parallel g(x_t; \theta_0) \parallel_2.
\]
8.1 Empirical Setting of UCB

UCB is the key component of MuFasa, determining the empirical performance. \( UCB(X_t) \) (Lemma 5.2) can be formulated as

\[
UCB(X_t) = (1 - \lambda) \|g(x; \theta_0)\|_{A^{-1}} + \lambda \|g(x; \theta_t)\|_{A^{-1}}
\]

where \( g(x; \theta_0) \) is the gradient at initialization, \( g(x; \theta_t) \) is the gradient after \( k \) iterations of gradient descent, and \( \lambda \) is a tunable parameter to trade off between them. Intuitively, \( g(x; \theta_0) \) has more bias as the weights of neural network function \( f \) are randomness initialized, which brings more exploration portion in decision making of each round. In contrast, \( g(x; \theta_t) \) has more variance as the weights should be nearer to the optimum after gradient descent.

In the setting where the set of arms are fixed, given an arm \( x_i \), let \( m_i \) be the number of rounds that \( x_i \) has been played before. When \( m_i \) is small, the learner should explore \( x_i \) more (\( \lambda \) is expected to be large). Instead, when \( m_i \) is large, the learner does not need more exploration on it. Therefore, \( \lambda \) can be defined as a decreasing function with respect to \( m_i \), such as \( \frac{1}{\sqrt{m_i + \epsilon}} \) and \( \frac{1}{\log(m_i + \epsilon)} \). In the setting without this condition, we can set \( \lambda \) as a decreasing function with respect to the number of rounds \( t \), such as \( \frac{1}{\sqrt{t + \epsilon}} \) and \( \frac{1}{\log(t + \epsilon)} \).

8.2 Future Work

This paper introduces a novel bandit problem, multi-facet contextual bandits. Based on empirical experiments and theoretical analysis, at least three factors play important roles in multi-facet bandits: (1) As all bandits are serving the same user, mutual influence does exist among bandits. Thus, how to extract and harness the mutual influence among bandits is one future-work direction; (2) The weights of bandits usually determine the final reward, since each bandit has a different impact on the serving user. For example, in the diabetes case, the bandit for selecting medications should have a higher weight than the bandit for recommending diabetes articles; (3) Sub-rewards contain crucial information and thus exploiting sub-rewards matters in this problem.

To solve multi-facet bandit problem, we propose the algorithm, MuFasa. However, there are three limitations implicitly shown in this paper: (1) For leveraging the sub-rewards, we assume the \( \tilde{C} \)-lipschitz continuity of final reward function, while \( \tilde{C} \) is usually unknown; (2) In the selection criteria of MuFasa, it needs to select one arm with the highest score from all possible combinations \( S_t \) (Eq. (2) and Eq. (5)). However, the size of \( S_t \) grows exponentially with respect to the number of bandits \( K \). (3) MuFasa only uses fully-connected work to exploit rewards, while many more advanced neural network models exist such as CNN, residual net, self-attention, etc.