Expected Improvement for Contextual Bandits

Anonymous Author(s)
Affiliation
Address
email

Abstract

The expected improvement (EI) is a popular technique to handle the tradeoff between exploration and exploitation under uncertainty. However, compared to other techniques as Upper Confidence Bound (UCB) and Thompson Sampling (TS), the theoretical properties of EI have not been well studied even for non-contextual settings such as standard bandit and Bayesian optimization. In this paper, we introduce and study the EI technique as a new tool for the contextual bandit problem which is a generalization of the standard bandit. We propose two novel EI based algorithms for this problem, one when the reward function is assumed to be linear and the other for more general reward functions. With linear reward function, we demonstrate that our algorithm achieves a near-optimal regret. For more general reward functions, we use deep neural networks to model the reward function, and prove that our algorithm achieves a $\tilde{\mathcal{O}}(\tilde{d}\sqrt{T})$ regret, where $\tilde{d}$ is the "effective" dimension of a neural tangent kernel matrix and T is the number of iterations. Finally, we provide an empirical evaluation of the algorithms on both synthetic functions and various benchmark datasets. Our experiments show that our algorithms work well and consistently outperform existing approaches.

1 Introduction

The stochastic contextual bandit problem is an important problem in machine learning ([7, 21]) to optimize the trade-off between exploration and exploitation in sequential decision making, and has been extensively studied in real-world applications such as personalized recommendation [22], advertising [16], robotic control [23], and healthcare [17]. At each round, the agent observes a feature vector (the "context") for each of the K arms, pulls one of them, and in return receives a scalar reward. The goal is to maximize the cumulative reward, or minimize regret (see our definition in Section 2), in a total of T rounds. To do so, the agent must find a trade-off between exploration and exploitation.

There are two standard techniques to solve the trade-off in contextual bandits. The first technique uses the optimism in face of uncertainty which chooses promising actions by maximizing upper-confidence bounds (UCB), and the second one using Thompson Sampling (TS) whose basic idea is to estimate a posterior distribution on the reward, and sample an arm that maximises a random reward drawn from this distribution. A series of work has applied both UCB and TS techniques or their variants to explore in contextual bandits with many forms of reward functions - from linear to nonlinear. In the line of UCB, there are works of [22, 11, 1] for the linear bandits, works of [15, 32] for nonlinear contextual bandits, and very recently [36], which uses neural networks to learn the reward function. In the line of TS, [3, 29] are for linear bandits, [29, 20] for generalized linear functions, and [28, 35] for nonlinear bandits using deep neural networks.

Besides UCB and TS, the expected improvement (EI) [24] is also one of the oldest and popular techniques to handle the tradeoff between exploration and exploitation under uncertainty. Different from UCB and TS, the EI is a greedy improvement-based heuristic that samples an action offering the greatest expected improvement over an incumbent. Due to its simplicity and ability to handle

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

uncertainty, EI has been widely used in Bayesian optimization [26, 34] - a problem that is closely related to infinite-arm multi-arm bandits. However, *it is yet to be seen whether EI can handle well the trade-off between exploration and exploitation in a contextual bandit problem and further in reinforcement learning which is a generalization of contextual bandits?*

We note that despite its wide use, the theoretical aspects of this technique have not been well studied compared to those of UCB and TS. A notable exception is the work in [9], who provides a convergence rate of EI for BO in the noise-free setting. [33] and [25] consider the regret analysis of EI for BO in the noisy setting. There are also several papers [31, 27] using EI to study the best-arm identification problem (also known as "pure exploration") which is a finite variant of Bayesian optimization. A key challenge of analyzing EI algorithms theoretically comes from its improvement function involving nonlinear, nonconvex term. In contextual bandit problem which is harder than the standard bandit, EI strategy has not been used, let alone an EI algorithm with optimal rate of convergence. Therefore, the design of a provable EI algorithm in contextual bandits is an open problem.

In this work, we study the EI technique for contextual bandits from both theoretical and practical perspectives. Our main contributions in this paper are:

- We introduce and formalize Expected Improvement as a new strategy for contextual bandits creating a parallel to UCB and TS.
- We propose two EI-based algorithms. The first algorithm (LinEI) is designed for the linear bandits whilst the second algorithm (NeuralEI) is designed for more general reward function and we model it by a deep neural network.
- For the linear reward function, our LinEI algorithm is able to achieve $\mathcal{O}(d\sqrt{T\ln^2(T)\ln\frac{T}{\delta}})$ regret t with probability $1-\delta$, which matches the information theoretic lower bound $\Omega(d\sqrt{T})$ for this problem (up to $\ln(T)$). For the general reward function, we prove that, under standard assumptions (see section 4.1), our NeuralEI algorithm is able to achieve $\tilde{\mathcal{O}}(\tilde{d}\sqrt{T})$ regret, where $\tilde{d}$ is the "effective" dimension of a neural tangent kernel matrix and T is the number of rounds.
- Finally, we demonstrate the performance of our proposed EI-based algorithms against UCB and TS based approaches and other baselines on both synthetic functions and various benchmark datasets. Our experiments show that LinEI outperforms other baselines in the linear setting, and when the reward function is non-linear, NeuralEI outperforms all baselines.

2 Problem Setting

We consider the problem of K-arm contextual bandits. At time $t = 1, 2, ...$, the agent observes K contextual vectors $x_{i,t} \in \mathbb{R}^d$, then selects an arm $a(t)$ and receives a reward $r_{a(t),t}$ which has a general form as follows:

$$r_{a(t),t} = h(x_{a(t),t}) + \xi_{a(t),t},$$

where h is an unknown reward function satisfying $0 \leq h(x) \leq 1$ for any $x \in \mathbb{R}^d$, and $\xi_{a(t),t}$ is conditionally R-subGaussian for a constant $R \geq 0$, i.e., $\forall\lambda \in \mathbb{R}, \mathbb{E}[e^{\lambda\xi_{a(t),t}}|\{x_{i,t}\}_{i=1}^K] \leq \exp(\frac{\lambda^2R^2}{2})$. In our setting, we assume that these context vectors may be chosen by an adversary in an adaptive manner after observing the arms played and their rewards up to time $t-1$. For the unknown function h, we consider two cases as follows:

- The reward function h is linear, i.e., $h(x_{t,i}) = x_{t,i}^T\theta^*$, where $\theta^* \in \mathbb{R}^d$ are fixed but unknown parameters. Without loss of generality, we here assume that $||x_{i,t}|| \leq 1, ||\theta^*|| \leq 1$.
- The reward function h is modelled by a fully connected neural network with depth $L \geq 2$ defined recursively by

$$f(x;\theta) = \sqrt{m}W_L\sigma(W_{L-1}\sigma(...\sigma(W_1x))),$$

  where $\sigma(x) := \max\{x, 0\}$ is the ReLU activation, $\theta = (\text{vec}(W_1); ...; \text{vec}(W_L)) \in \mathbb{R}^p$ is the collection of parameters of the neural network with $p = dm + m^2(L-2) + m$ which is the number of parameters of the network. Without loss of generality, we assume that the width of each hidden layer is the same (i.e., m) for convenience in analysis. We denote the gradient of the neural network function by $g(x;\theta) = \bigtriangledown_\theta f(x;\theta) \in \mathbb{R}^p$.

Algorithm 1 The Linear Expected Improvement Algorithm (LinEI)

Input: Parameters C_0, β

for $t = 1$ to T **do**
  Observe contexts $\{x_{i,t}\}_{i=1}^K$
  Set $\overline{a}(t) := \text{argmax}_{i\in\mathcal{K}} \alpha_{i,t}^{EI}$, $\tilde{a}(t) = \text{argmax}_{i\in\mathcal{K}}\{x_{i,t}^\top \hat{\theta}_t\}$
  if $\alpha_{\overline{a}(t),t}^{EI} \geq \frac{C_0}{t^\beta}$ **then**
    $a(t) = \overline{a}(t)$
  else
    $a(t) = \tilde{a}(t)$
  end if
  Play arm $a(t)$, and observe reward $r_{a(t),t}$
  Update $X(t+1) = \lambda I_d + \sum_{j=1}^t x_{a(j),j} x_{a(j),j}^\top$, $\hat{\theta}_{t+1} = X(t+1)^{-1}(\sum_{j=1}^t x_{a(j),j} r_{a(j),j})$
end for

Performance Measure. Let $a^*(t)$ denote the optimal arm at time t. The objective is to minimize the cumulative regret $R(T) = \sum_{t=1}^T (x_{a^*(t),t}^T \theta^* - x_{a(t),t}^T \theta^*)$.

3 The LinEI algorithm for linear bandits

In this section, we design a provable version of EI algorithm for linear bandits. For theoretical guarantees, we make some assumptions on the prior distribution of the reward function, and a form of the expected improvement in context of linear bandits.

Prior and Posterior Distributions. We follow the design for priors of the reward function like TS algorithm [3]. We assume that the reward $r_{i,t}$ of each arm i follows a Gaussian distribution $\mathcal{N}(x_{i,t}^\top \theta^*, v^2)$, where the variance v^2 is a free parameter (possibly time-dependent) that can be set specific to an algorithm. Let $X(t) = \lambda I + \sum_{j=1}^{t-1} x_{a(j),j} x_{a(j),j}^\top$, $\hat{\theta}_t = X(t)^{-1}(\sum_{j=1}^{t-1} x_{a(j),j} r_{a(j),j})$. Then if we assume that the prior for θ^* at time t is given by $\mathcal{N}(\hat{\theta}_t, v^2 X^{-1}(t))$, then the posterior distribution of θ^* at time $t+1$ is $\mathcal{N}(\hat{\theta}_{t+1}, v^2 X^{-1}(t+1))$ (see the proof in Appendix A.1 in [3]).

Expected Improvement. We now use this posterior distribution update to define the form of the expected improvement of each arm in contextual bandits. We denote $r_t^+ = \max_{i\in\mathcal{K}}\{x_{i,t}^T \hat{\theta}_t\}$ which is the largest mean estimate of reward among all arms at time t. We define the expected improvement of an arm i at time t as

$$\alpha_{i,t}^{EI} = \mathbb{E}_{\mu\sim\mathcal{N}(\hat{\theta}_t, v_t^2 X(t)^{-1})}[\max\{0, x_{i,t}^\top \mu - r_t^+\}], \tag{1}$$

The $\alpha_{i,t}^{EI}$ measures the potential of arm i to improve upon the largest posterior reward mean at time t. We note that in Eq(1), r_t^+ plays a role as an incumbent. In Bayesian optimization, the incumbent is usually selected as the best reward value so far, or the largest reward mean so far. In our setting with contextual bandits, we choose the latter for convenient in analysis.

Further, we define $s_{i,t} = \sqrt{x_{i,t}^T X(t)^{-1} x_{i,t}}$. We see that if $\mu \sim \mathcal{N}(\hat{\theta}_t, v^2 X(t)^{-1})$ then the marginal distribution of $x_{i,t}^T \mu$ is Gaussian with mean $x_{i,t}^T \hat{\theta}_t$ and standard deviation $v s_{i,t}$. Setting $z_{i,t} = \frac{x_{i,t}^\top \hat{\theta}_t - r_t^+}{v s_{i,t}}$, we can express the expected improvement in closed form as follows

$$\alpha_{i,t}^{EI} = v s_{i,t}[z_{i,t}\Phi(z_{i,t}) + \phi(z_{i,t})], \tag{2}$$

where $\Phi(.)$ and $\phi(.)$ are the standard normal cdf and pdf function of the normal distribution respectively.

Proposed LinEI Algorithm. An improvement-based strategy like EI seems suitable for BAI problem and BO problem whose goal is to seek the best arm with the highest reward mean. However, in contextual bandits as in our setting, the mean reward of arm i is additionally associated with a

context $x_{i,t}$ which may be controlled by an adversary. It follows that the mean reward for each arm i can be different in each iteration depending on contexts, thus the notion of "best arm" does not exist. Due to this, the framework of EI as in BAI or BO problem does not naturally extend to contextual bandits. To make use of EI technique in contextual bandits with convergence guarantees, we here propose a new variant of EI through a simple modification. We select the arm suggested by the EI strategy only if the expected improvement (measured by $\alpha_{i,t}^{EI}$ of all arms) is higher than a threshold. Otherwise we simply select the arm with the largest mean estimate of reward (computed as $\text{argmax}_{i\in[K]}\{x_{i,t}^\top \hat{\theta}_t)$ for the pure exploitation. A challenge here is how to determine the threshold . We propose to use a decreasing function of t in the form of $g(t) = \frac{C_0}{t^\beta}$ which is controlled by two parameters C_0, β. Our algorithm is summarized in Algorithm 1. The threshold function $g(t)$ is a crucial factor in achieving optimal convergence rate for our algorithm. A relevant choice of parameters of the threshold function is discussed in the next section.

3.1 Theoretical Analysis

In this section, we provide the regret bound for the proposed LinEI algorithm.

The EI technique is used so far in BAI and BO problems to seek the best arm with the highest reward mean. In such problems, the reward mean of each arm is fixed in advance. To leverage this property, [27] uses the *allocation matching* technique to allocate enough number of samples to suboptimal arms so that we can eliminate these arm with a high confidence. In BO, [9] uses the monotonicity property of incumbent to derive directly an upper bound for the simple regret. However, in contextual bandits where the best arm does not exist, and the mean reward of any arm can be different in different iterations, the monotonicity property of incumbent disappears, and the allocation technique is also no longer applicable. The proposed LinEI addresses these challenges. Moreover, it selects an arm based on not only EI strategy but also a greedy strategy depending on a threshold function. This makes our regret analysis different from the standard EI technique.

In our proof, we first extend several results in [9] in Bayesian optimization in a non-contextual setting to our setting with contexts and noises. For instance, the following lemma is used to upper bound and lower bound the expected improvement for each arm i.

Lemma 3.1. *Pick $0 < \delta < 1$. Set $I_{i,t} = max\{0, x_{i,t}^T\theta^* - r_t^+\}$. Then with probability $1 - \frac{\delta}{t^2}$ we have*

$$I_{i,t} - \beta_t s_{i,t} \leq \alpha_{i,t}^{EI} \leq I_{i,t} + (\beta_t + v_t)s_{i,t}.$$

Here, $\beta_t = R\sqrt{d\ln(\frac{t^3}{\delta})} + 1$, $s_{i,t} = \sqrt{x_{i,t}^\top X(t)^{-1}x_{i,t}}$ and v_t is used instead of v. Then, depending on the arm selection by our LinEI algorithm, we can upper bound the instantaneous regret $r_t = x_{a^*(t),t}^\top\theta^* - x_{a(t),t}^\top\theta^*$ by two different ways. Combining them, we achieve an upper bound for the instantaneous regret $r_t = x_{a^*(t),t}^\top\theta^* - x_{a(t),t}^\top\theta^*$ as in Lemma 9 in our Supplementary Material:

$$r_t \leq \frac{\tau(\frac{\beta_t}{v_t})}{\tau(-\frac{\beta_t}{v_t})}(2\beta_t + v_t)s_{a(t),t} + \beta_t s_{a(t),t} + \frac{\tau(\frac{\beta_t}{v_t})}{\tau(-\frac{\beta_t}{v_t})}\frac{C_0}{t^\beta} + \sqrt{2\ln(Rt^\beta) + 2\sqrt{3C_0^{-2}d\ln(\frac{t}{\delta})}}v_t s_{a(t),t},$$

with probability at least $1 - \delta$. The function τ is defined as $\tau(z) = z\Phi(z) + \phi(z)$. Here, we note that parameter v_t plays the role of parameter v at time t we discussed above. In our analysis, v_t is used to eliminate the influence of β_t so that $\frac{\beta_t}{v_t}$ is bounded as t grows.

Finally, by a relevant choice of parameters C_0 and β, we achieve the regret bound for our proposed algorithm as follows:

Theorem 3.2. *Given any $\delta \in (0, 1)$. If $v_t = R\sqrt{3dln\frac{t}{\delta}} + 1$, $\sqrt{d} \leq C_0 \leq d$ and $0.5 \leq \beta \leq 3$ then with probability $1 - \delta$, the cumulative regret of the LinEI algorithm is bounded as*

$$R(T) = \mathcal{O}(d\sqrt{Tln^2(T)ln\frac{T}{\delta}}).$$

Algorithm 2 Neural Expected Improvement Algorithm (NeuralEI)

Input: Number of rounds T, exploration variance ν, network width m, regularization parameter λ and parameters C_0, β

Set $U_0 = \lambda I$
for $t = 1$ to T **do**
  Set $\overline{a}(t) := \text{argmax}_{i\in[K]} \alpha_{i,t}^{EI}$, $\tilde{a}(t) = \text{argmax}_{i\in[K]}\{f(x_{i,t};\theta_{t-1})\}$
  if $\alpha_{\overline{a}(t),t}^{EI} \geq \frac{C_0}{t^\beta}$ **then**
    $a(t) = \overline{a}(t)$
  else
    $a(t) = \tilde{a}(t)$
  end if
  Play arm $a(t)$, and observe reward $r_{a(t),t}$
  Set θ_t to be the output of gradient descent for solving Eq(3)
  $U_t = U_{t-1} + g(x_{a(t),t};\theta_t)g^\top(x_{a(t),t};\theta_t)/m$
end for

4 The NeuralEI algorithm for neural contextual bandits

In this section, we extend our Algorithm 1 to the more general setting where the reward function is modelled by a fully connected neural network. Similar to the Neural Thompson Sampling approach [35], our algorithm maintains a Gaussian distribution for each arm's reward. At time t, the posterior distribution of the reward of arm i is updated as follows. The mean is set to the output of the neural network, denoted by $f(x_{i,t};\theta_{t-1})$, and the variance is defined as $\sigma_{i,t}^2 = \lambda g^\top(x_{i,t};\theta_{t-1})U_{t-1}^{-1}g(x_{i,t};\theta_{t-1})/m$, where the matrix U_t^{-1} is updated as $U_t = U_{t-1} + g(x_{a(t),t};\theta_t)g^\top(x_{a(t),t};\theta_t)/m$ and parameter θ_t is the solution to the following minimization problem:

$$\min_\theta \sum_{i=1}^{t}[f(x_{a(i),i;\theta}) - r_{a(i),i}]^2/2 + m\lambda||\theta - \theta_0||_2^2/2, \tag{3}$$

where θ_0 is randomly initialized network parameter. We can adapt gradient descent algorithms to solve this problem with step size η and total number of iterations J like the gradient descent algorithm of [36].

Expected Improvement for Neural Contextual Bandits. We now define the form of the expected improvement in this setting. At each time step t, we denote $f_t^+ = \max_{i\in[K]}\{f(x_{i,t};\theta_{t-1})\}$ which is the highest mean estimate of $f(x,\theta_{t-1})$ among all arms at time t. We define the expected improvement value of an arm i at time t as

$$\alpha_{i,t}^{EI} = \mathbb{E}_{\widetilde{f}_{i,k}\sim\mathcal{N}(f(x_{i,t};\theta_{t-1}),\nu^2\sigma_{i,t}^2)}[\max\{0, \widetilde{f}_{i,k} - f_t^+\}].$$

Further, by setting $z_{i,t} = \frac{f(x_{i,t};\theta_{t-1}) - f_t^+}{\nu\sigma_{i,t}}$, the above expectation can be computed analytically as follows

$$\alpha_{i,t}^{EI} \quad = \quad \nu\sigma_{i,t}[z_{i,t}\Phi(z_{i,t}) + \phi(z_{i,t})] \tag{4}$$

Our NeuralEI algorithm is given in Algorithm 2. It starts by initializing $\theta_0 = (\text{vec}(W_1); ...; \text{vec}(W_L))$, where for each $1 \leq l \leq L-1, W_l = (W, 0; 0, W)$, each entry of W is generated independently from $N(0, 4/m)$; $W_L = (w^\top, -w^\top)$, each entry of w is generated independently from $N(0, 2/m)$. NeuralEI extends our LinEI algorithm to the setting where the reward function h is modelled by a fully connected neural network.

4.1 Regret Analysis

In this section, we provide a regret analysis of the NeuralEI algorithm. We first provide necessary background on the neural tangent kernel (NTK) theory, which plays an important role in our analysis. Following a recent line of research [36, 35], we define the covariance between two data point $x, y \in \mathbb{R}^d$

as follows: $\tilde{H}^{(1)}(x,y) = \sigma^{(1)}(x,y) = x^\top y$, $A^{(l)}(x,y) = \begin{pmatrix} \sigma^{(l)}(x,x) & \sigma^{(l)}(x,y) \\ \sigma^{(l)}(x,y) & \sigma^{(l)}(y,y) \end{pmatrix}$, $\sigma^{l+1}(x,y) = 2\mathbb{E}_{(u,v)\sim N(0,A^{(l)}(x,y))}[\sigma(u)\sigma(v)]$, $\tilde{H}^{(l+1)}(x,y) = 2\tilde{H}^{(l)}(x,y)\mathbb{E}_{(u,v)\sim N(0,A^{(l)}(x,y))}[\sigma'(u)\sigma'(v)] + \sigma^{(l+1)}(x,y)$. Similar to [36, 35], we assume that the number of rounds T is *known* and denote the neural tangent kernel (NTK) matrix $H \in \mathbb{R}^{TK\times TK}$ based on all contextual vectors $\{x_{t,k}\}_{t\in[T],k\in[K]}$. Renumbering $\{x_{t,k}\}_{t\in[T],k\in[K]}$ as $\{x_i\}_{i=1,\ldots,TK}$, then each entry H_{ij} is defined as

$$H_{ij} = (\tilde{H}^{(L)}(x_i,x_j) + \sigma^{(L)}(x_i,x_j))/2, \tag{5}$$

for all $i,j \in [TK]$. Based on the above definition, we impose the following assumption on the contexts generated by the adversary and the corresponding NTK matrix H.

Assumption 4.1. Let H be defined in Eq(5). There exists $\lambda_0 > 0$ such that $H \geq \lambda_0 I$. In addition, for any $t \in [T], k \in [K]$, $||x_{t,k}||_2 = 1$ and $[x_{t,k}]_j = [x_{t,k}]_{j+d/2}$.

Remark 1. Compared to Algorithm 1 for linear bandits, our Algorithm 2 needs an additional Assumption 1 to guarantee the convergence. The assumption that the NTK matrix is positive definite has been considered in prior work on NTK which is a mild condition and also imposed in other related works [4, 13, 36, 35]. The assumption on contexts ensures that $f(x_{i,t};\theta_0) = 0$ for any $i \in [K], t \in [T]$.

The NTK technique builds a connection between deep neural networks and kernel methods. It enables us to adapt some complexity measures for kernel methods to describe the complexity of the neural network through the notation of the effective dimensions as defined in [36, 35]. The effective dimension $\tilde{d}$ of matrix H with regularization parameter λ is defined as $\tilde{d} = \frac{\log\det(I+H/\lambda)}{\log(1+TK/\lambda)}$.

Using these notations, we are now ready to present the second main result of the paper. Let $a^*(t) = \text{argmax}_{i\in[K]}\mathbb{E}[r_{i,t}]$ be the optimal action at round t that maximizes the expected reward, we define the *expected cumulative regret* after T iterations as $\overline{R}(T) = \mathbb{E}[\sum_{t=1}^{T}(r_{a^*(t),t} - r_{a(t),t})]$. Then, we achieve the following upper regret bound for our Algorithm 2 by combining our EI techniques for LinEI with NTK techniques. A completed proof is provided in Supplementary Material.

Theorem 4.2. *Under Assumption 1, set the parameters in Algorithm 2 as* $\lambda = 1 + 1/T$, $\nu = B + R\sqrt{\tilde{d}log(1+TK/\lambda) + 2 + 2log(1/\delta)}$, *where* $B = max\{1, \sqrt{2h^\top H^{-1}h}\}$ *with* $h = (h(x_1),\ldots,h(x_{TK}))^\top$. *If* $\sqrt{\tilde{d}} \leq C_0 \leq \tilde{d}$, $\beta \geq 2$, *and the network width* m *satisfies* $m \geq poly(\gamma, T, K, L, log(1/\delta))$, *then with probability at least* $1-\delta$, *the regret of Algorithm 2 is bounded as*

$$\overline{R}(T) \leq \mathcal{O}(\tilde{d}\sqrt{\beta log(1+TK)log(T)T}).$$

Remark 2. The regret bound depends on the parameter β. The best choice is $\beta = 2$ that tightens the regret. Theorem 4.2 implies the regret of NeuralEI is on the order of $\tilde{\mathcal{O}}(\tilde{d}\sqrt{T})$. This result matches the regret bound of NeuralUCB ([36]), NeuralTS ([35]), as well as of [10]. The effective dimension $\tilde{d}$ measures how quickly the eigenvalues of H diminish, and only depends on T logarithmically in several special cases ([32]), according to [36]. Furthermore, [35] shows that $\tilde{d}$ can be upper bounded if all contexts are nearly on some low-dimensional subspace of the RKHS space spanned by NTK. Currently, [18] gives an explicit sublinear regret bound for a neural network based UCB algorithm. However, their solution requires that contexts lie on on the d-dimensional hyper-sphere. Compared to these works, our EI based results may be of independent interest.

Remark 3. Similar to most of existing results in neural bandits, our results require a large value of m. This is rooted in the current deep learning theory based on the neural tangent kernel.

4.2 Technical Challenges

While the proof of NeuralEI follows the same lines as that of LinEI, we emphasize several key differences. First, we define filtration $\mathcal{F}_{t-1}$ as the union of history until time $t-1$, and the contexts at time t. Given a time t, we define an event $\mathcal{E}_t^\sigma$ as follows:

$$\mathcal{E}_t^\sigma = \{\omega \in \mathcal{F}_t : \forall i \in [K],\ |\tilde{f}_{i,t} - f(x_{i,t};\theta_{t-1})| \leq c_t\nu\sigma_{i,t}\},$$

where $c_t = \sqrt{4\log t + 2\log K}$.

Similarly, we define an event $\mathcal{E}_t^\mu$ as follows:

$$\mathcal{E}_t^\mu = \{\omega \in \mathcal{F}_t : \forall i \in [K],\ |f(x_{i,t};\theta_{t-1}) - h(x_{t,k})| \leq \nu\sigma_{i,t} + \epsilon(m)\},$$

where $\epsilon(m)$ is defined as in [35].

Based on the definitions of event $\mathcal{E}_t^\sigma$ and $\mathcal{E}_t^\mu$, we achieve two different ways to lower bound and upper bound the expected improvement as follows:

Lemma 4.3. *Assume that the event $\mathcal{E}_t^\sigma$ holds. Set $I_{i,t} = max\{0, \tilde{f}_{i,t} - f_t^+\}$. Then for every $i \in [K]$, we have*

$$I_{i,t} - c_t\nu\sigma_{i,t} \leq \alpha_{i,t}^{EI} \leq I_{i,t} + \nu(c_t + 1)\sigma_{i,t}.$$

Lemma 4.4. *Assume that the event $\mathcal{E}_t^\mu$ holds. Set $J_{i,t} = max\{0, h(x_{i,t}) - f_t^+\}$. Then for every $i \in [K]$, we have*

$$J_{i,t} - \nu\sigma_{i,t} - \epsilon(m) \leq \alpha_{i,t}^{EI} \leq J_{i,t} + \nu\sigma_{i,t} + \epsilon(m).$$

While Lemma 4.3 is an adaptation of Lemma 3.1 to neural contextual bandits, Lemma 4.4 is a novel result compared to the regret analysis in our Section 3.1 for linear bandits. Another key difference from the regret analysis for linear bandits is that we upper bound the instantaneous regret $h(x_{a^*(t),t}) - h(x_{a(t),t})$ in a different way, not only depending on the arm selection but also the condition $\sigma_{a^*(t),t} > \epsilon(m)$. The detailed (step-by-step) proof can be found in the Supplementary Material.

5 Related Works and Discussion

Given the vast literature on bandit algorithms, we restrict our review to linear bandits and neural contextual bandits.

Linear Contextual Bandit. A lower bound of $\Omega(d\sqrt{T})$ for linear bandits was given in [12], when the number of arms is allowed to be infinite. [1] analyze a UCB-style algorithm and provide a regret upper bound $\mathcal{O}(d\log(T)\sqrt{T} + \sqrt{dT\log(T/\delta)})$. Compared to this work, the regret of our proposed LinEI has an additional $\sqrt{\ln T}$, however, it still matches the information theoretic lower bound in this setting. When the number of arms K is finite, [11] achieve a regret bound of $\mathcal{O}(\sqrt{Td\ln^3(KT\ln T)/\delta})$ with probability at least $1-\delta$. [8] provides an algorithm based on exponential weights, with regret of order $\mathcal{O}(d\sqrt{T\log K})$. These algorithms may not be effective when the number of arms K is large. For example, when K is exponential in d, the regret bound of [11] would become $\tilde{\mathcal{O}}(d^2\sqrt{T})$ showing a quadratic growth in d. We note that although our algorithm is a Bayesian approach, our regret bounds will hold irrespective of whether or not the actual reward distribution matches the Gaussian likelihood function. Thus, our bounds for EI algorithm are directly comparable to the UCB family of algorithms. This is also mentioned in [3].

The Thompson Sampling algorithm [3] and an alternative given in [2] often bear an additional $\sqrt{d}$ in the regret bound compared to UCB based bounds. Very recently, [19] improved this regret bound of TS by integrating a doubly robust estimator with TS. However, this work requires additional significant computations and their setting is restrictive in the sense that contexts need to be independent. Our EI-based algorithm achieves the same regret bound order as in [19], however, it caters to a more general setting where the contexts may be controlled by an adaptive adversary and therefore may not be independent.

Another approach for linear bandits is the Information Directed Sampling (IDS) which was introduced in [30]. It provides an action-selection mechanism by minimizing the information ratio between the squared expected regret and the mutual information between optimal action and the next observation over all action sampling distributions. IDS obtains a performance improvement over TS and UCB algorithms in some cases, but has heavy sampling requirements. It has been shown in their experiments that IDS requires significantly more compute time than Thompson sampling and UCB algorithms.

Table 1: Characteristics of benchmark datasets used in Section 5.2.

Dataset	letter	pendigits	covertype	avila	magic	dry bean	statlog
Classes (K)	26	10	7	12	2	7	7
Feature Dimension	17	16	54	10	10	16	8
Dataset size	20000	10992	581012	20867	19020	13611	58000

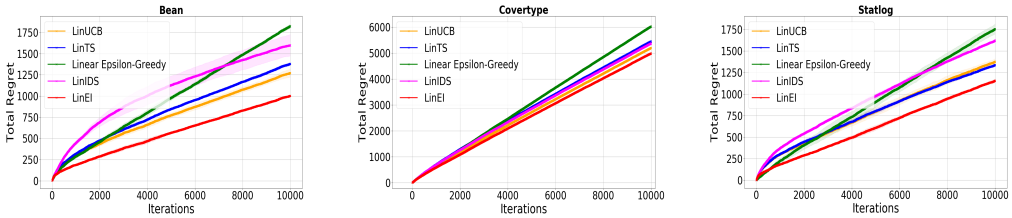

Figure 1: Comparison of our proposed LinEI and baseline algorithms in linear bandits.

Recently, [5] provided a modification of the arm scoring rule of IDS to reduce computations. However, both [30] and [5] only provide the bounds on *expected regret*. In contrast, our work provides regret bounds in terms of cumulative regret which is tighter than expected regret.

Neural Contextual Bandit. Neural contextual bandits are becoming attractive due to the current advancement in optimization and generalization of deep neural networks [4, 13]. Neural contextual bandits have been considered in both popular techniques UCB [36] and Thompson Sampling [35]. Currently, [6] proposes a novel neural exploration strategy and their solution achieves a sublinear regret on T. Compared to all existing works in this sub-field, our EI based algorithm is new, and may be of independent interest.

Due to space limit, we will add additional related works in Section B of Supplementary Material.

6 Experiments

6.1 Linear Bandits

In this subsection, we assess the performance of our LinEI algorithm on several benchmark dataset including `covertype`, `magic`, `avila`, `dry bean`, `statlog`, `letter`, `pendigits`, all from UCI [14]. We compare the LinEI with methods designed for linear bandits including: LinTS [3], LinUCB [1], Linear Epsilon Greedy for the linear reward, LinIDS [30] for linear bandits. To transform these classification problems into multi-armed bandits, we adapt the disjoint models to build a context feature vector for each arm: given an input feature $x \in \mathbb{R}^d$ of a k-class classification problem, we build the context feature vector with dimension kd as $x_1 = (x; 0; ...; 0), x_2 = (0; x; ...; 0), ..., x_k = (0; 0...; x)$. The algorithm generates a set of predicted reward and pulls the greedy arm. For these classification problems, if the algorithm selects a correct class by pulling the corresponding arm, it will receive a reward as 1, otherwise 0. The cumulative regret over time horizon T is measured by the total mistakes made by the algorithm.

We set the time horizon of our algorithm to 10000 for all data sets. In the experiments, we shuffle all datasets randomly. For (λ, ν) used in LinUCB and LinTS and our algorithm, we set $\lambda = 1$ following previous works and do a grid search of $\nu \in \{1, 0.1, 0.01\}$ to select the parameter with the best performance. All experiments are repeated 10 times, and the average with standard error are reported. For LinIDS, we use the number of samples $M = 100$. For Linear Epsilon Greedy, we use $\epsilon = 0.1$. For our LinEI algorithm, we can choose any value $C_0 \in [\sqrt{d}, d]$ and $\beta \in [0.5, 3]$. For LinEI, we set $C_0 = \sqrt{d}$ and $\beta = 2$.

Figure 1 shows the total regret of all algorithms for datasets `bean`, `covertype` and `statlog`. The Linear Epsilon Greedy performs the worst. This implies that the random exploration is not as effective as other methods. While LinUCB, LinTS and LinIDS are competitive, all these methods

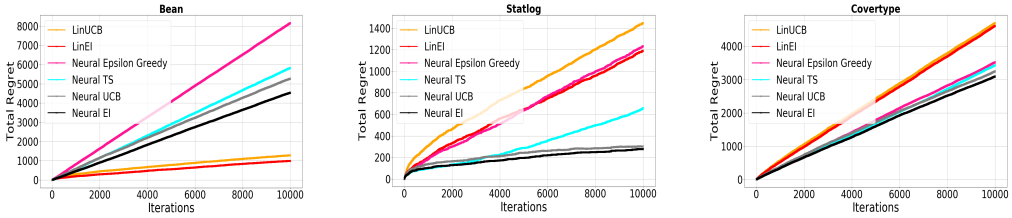

Figure 2: Comparison of NeuralEI and baseline algorithms on real-world datasets.

are significantly outperformed by the proposed algorithm. The arm-selection of our LinEI bases on two strategies: expected improvement and greedy strategy. Compared to Linear Epsilon Greedy, our greedy strategy is similar. It confirms that the exploration of the expected improvement is effective. This suggests that using the expected improvement strategy is efficient in linear bandits. Due to space limit, the additional results on `magic`, `pendigits`, `letter`, and `avila` are shown in Section A of **Supplementary Material**.

6.2 Neural Bandits

We compare the proposed NeuralEI with baselines including: LinUCB [1], our LinEI for linear bandits problem, Neural Epsilon Greedy, NeuralUCB [36], NeuralTS [35]. We do the same classification problems as the experiments in subsection Linear Bandits. For methods using the neural network, we use one-hidden layer neural networks with 100 neurons to model the reward function. During posterior updating, gradient descent is run for 100 iterations with learning rate 0.001. For Neural UCB/Thompson Sampling and Neural EI, we use a grid search on $\lambda \in \{1, 10^1, 10^{-2}, 10^{-3}\}$ and $\nu \in \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$. We consider our algorithm on both synthetic datasets and real-world datasets.

Real-world Datasets. Similar to the subsection Linear Bandit, we build the context feature vector with dimension kd as $x_1 = (x; 0; ...; 0), x_2 = (0; x; ...; 0), ..., x_k = (0; 0...; x)$. We also estimate our algorithm on datasets `bean`, `covertype` and `statlog`. Figure 2 show our results in the case of the neural bandits problem. Neural-based methods perform better because they can capture the nonlinearity of the underlying reward function. In real-datasets, while neural-based methods outperform LinEI and LinUCB for datasets `covertype` and `statlog`, these methods are not sample-efficient for learning the reward function of dataset `bean`. Perhaps, the reward function for dataset `bean` is linear. However, in all cases, our NeuralEI algorithm performs better than other neural-based methods. This suggests that using the expected improvement strategy is effective in both linear bandits and neural contextual bandits.

7 Conclusion

We introduced and formalized Expected Improvement as a new strategy for contextual bandits. We proposed two EI-based algorithms and analyzed them theoretically. The first algorithm assumes the reward function to be linear whilst the second algorithm is designed for the case when the reward function is general and can be modelled by a deep neural network. Our promising empirical results on both synthetic and real-world datasets suggested that our algorithms work well in practice compared to other approaches. We believe our work would be useful for further improvements and extensions. There are several interesting open questions. For example, we can extend our work to non-Gaussian distributions to model the unknown reward model parameters and derive regret bounds as long as concentration inequalities can be established. In another direction, we have used NTK in our work, but the NTK theory assumes overparameterised networks and an extension to narrow networks while maintaining the generalization of the reward function is an interesting open problem.

References

[1] Yasin Abbasi-yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24, pages 2312–2320. Curran Associates, Inc., 2011.

[2] Marc Abeille and Alessandro Lazaric. Linear Thompson Sampling Revisited. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 176–184. PMLR, 20–22 Apr 2017.

[3] Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 127–135, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[4] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[5] Jackie Baek and Vivek F. Farias. Ts-ucb: Improving on thompson sampling with little to no additional computation, 2021.

[6] Yikun Ban, Yuchen Yan, Arindam Banerjee, and Jingrui He. EE-net: Exploitation-exploration neural networks in contextual bandits. In *International Conference on Learning Representations*, 2022.

[7] Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems, 2012.

[8] Sébastien Bubeck, Nicoló Cesa-Bianchi, and Sham M. Kakade. Towards minimax policies for online linear optimization with bandit feedback. In Shie Mannor, Nathan Srebro, and Robert C. Williamson, editors, *Proceedings of the 25th Annual Conference on Learning Theory*, volume 23 of *Proceedings of Machine Learning Research*, pages 41.1–41.14, Edinburgh, Scotland, 25–27 Jun 2012. PMLR.

[9] Adam D. Bull. Convergence rates of efficient global optimization algorithms. *J. Mach. Learn. Res.*, 12:2879–2904, November 2011.

[10] Sayak Ray Chowdhury, Aditya Gopalan, and abc. On kernelized multi-armed bandits. ICML'17, page 844–853. JMLR.org, 2017.

[11] Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 208–214, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.

[12] Varsha Dani, Thomas P. Hayes, and Sham M. Kakade. Stochastic linear optimization under bandit feedback. In *21st Annual Conference on Learning Theory - COLT 2008, Helsinki, Finland, July 9-12, 2008*, pages 355–366, 2008.

[13] Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1675–1685. PMLR, 09–15 Jun 2019.

[14] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[15] Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

[16] Thore Graepel, Joaquin Quiñonero Candela, Thomas Borchert, and Ralf Herbrich. Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. ICML'10, page 13–20, Madison, WI, USA, 2010. Omnipress.

[17] Kristjan Greenewald, Ambuj Tewari, Predrag Klasnja, and Susan Murphy. Action centered contextual bandits, 2017.

[18] Parnian Kassraie and Andreas Krause. Neural contextual bandits without regret, 2021.

[19] Wonyoung Kim, Gi soo Kim, and Myunghee Cho Paik. Doubly robust thompson sampling for linear payoffs, 2021.

[20] Branislav Kveton, Manzil Zaheer, Csaba Szepesvari, Lihong Li, Mohammad Ghavamzadeh, and Craig Boutilier. Randomized exploration in generalized linear bandits. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2066–2076. PMLR, 26–28 Aug 2020.

[21] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

[22] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In Michael Rappa, Paul Jones, Juliana Freire, and Soumen Chakrabarti, editors, *WWW*, pages 661–670. ACM, 2010.

[23] Jeffrey Mahler, Florian T. Pokorny, Brian Hou, Melrose Roderick, Michael Laskey, Mathieu Aubry, Kai Kohlhoff, Torsten Kröger, James Kuffner, and Ken Goldberg. Dex-net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, page 1957–1964. IEEE Press, 2016.

[24] J. Močkus. On bayesian methods for seeking the extremum. In G. I. Marchuk, editor, *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974*, pages 400–404, Berlin, Heidelberg, 1975. Springer Berlin Heidelberg.

[25] Vu Nguyen, Sunil Gupta, Santu Rana, Cheng Li, and Svetha Venkatesh. Regret for expected improvement over the best-observed value and stopping condition. In Min-Ling Zhang and Yung-Kyun Noh, editors, *Proceedings of the Ninth Asian Conference on Machine Learning*, volume 77 of *Proceedings of Machine Learning Research*, pages 279–294. PMLR, 15–17 Nov 2017.

[26] Michael A. Osborne. Bayesian gaussian processes for sequential prediction, optimisation and quadrature. 2010.

[27] Chao Qin, Diego Klabjan, and Daniel Russo. Improving the expected improvement algorithm. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 5387–5397, Red Hook, NY, USA, 2017. Curran Associates Inc.

[28] Carlos Riquelme, George Tucker, and Jasper Snoek. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. In *International Conference on Learning Representations*, 2018.

[29] Daniel Russo and Benjamin Van Roy. Learning to optimize via posterior sampling, 2014.

[30] Daniel Russo and Benjamin Van Roy. Learning to optimize via information-directed sampling. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[31] Ilya O. Ryzhov. On the convergence rates of expected improvement methods. *Oper. Res.*, 64(6):1515–1528, 2016.

[32] Michal Valko, Nathan Korda, Rémi Munos, Ilias Flaounas, and Nello Cristianini. Finite-time analysis of kernelised contextual bandits. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, UAI'13, page 654–663, Arlington, Virginia, USA, 2013. AUAI Press.

[33] Ziyu Wang and Nando de Freitas. Theoretical analysis of bayesian optimisation with unknown gaussian process hyper-parameters, 2014.

[34] Dawei Zhan and Huanlai Xing. Expected improvement for expensive optimization: a review. *J. Glob. Optim.*, 78(3):507–544, 2020.

[35] Weitong Zhang, Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural thompson sampling. In *International Conference on Learning Representations*, 2021.

[36] Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural contextual bandits with UCB-based exploration. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11492–11502. PMLR, 13–18 Jul 2020.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]