UCBoost: A Boosting Approach to Tame Complexity and Optimality for Stochastic Bandits

Fang Liu[1], **Sinong Wang**[1], **Swapna Buccapatnam**[2], **Ness Shroff**[1]
[1] The Ohio State University
[2] AT&T Labs Research
liu.3977@osu.edu, wang.7691@osu.edu, sb646f@att.com, shroff.11@osu.edu

Abstract

In this work, we address the open problem of finding low-complexity near-optimal multi-armed bandit algorithms for sequential decision making problems. Existing bandit algorithms are either suboptimal and computationally simple (e.g., UCB1) or optimal and computationally complex (e.g., kl-UCB). We propose a boosting approach to Upper Confidence Bound based algorithms for stochastic bandits, that we call UCBoost. Specifically, we propose two types of UCBoost algorithms. We show that UCBoost(D) enjoys $O(1)$ complexity for each arm per round as well as regret guarantee that is $1/e$-close to that of the kl-UCB algorithm. We propose an approximation-based UCBoost algorithm, UCBoost(ϵ), that enjoys a regret guarantee ϵ-close to that of kl-UCB as well as $O(\log(1/\epsilon))$ complexity for each arm per round. Hence, our algorithms provide practitioners a practical way to trade optimality with computational complexity. Finally, we present numerical results which show that UCBoost(ϵ) can achieve the same regret performance as the standard kl-UCB while incurring only 1% of the computational cost of kl-UCB.

1 Introduction

Multi-armed bandits, introduced by Thompson [1933], have been used as quintessential models for sequential decision making. In the classical setting, at each time, a decision maker must choose an arm from a set of K arms with unknown probability distributions. Choosing an arm i at time t reveals a random reward $X_i(t)$ drawn from the probability distribution of arm i. The goal is to find policies that minimize the expected regret due to uncertainty about arms' distributions over a given time horizon T. Lai and Robbins [1985], followed by Burnetas and Katehakis [1996], have provided an asymptotically lower bound on the expected regret.

Upper confidence bounds (UCB) based algorithms are an important class of bandit algorithms. The most celebrated UCB-type algorithm is UCB1 proposed by Auer *et al.* [2002], which enjoys simple computations per round as well as $O(\log T)$ regret guarantee. Variants of UCB1, such as UCB-V proposed by Audibert *et al.* [2009] and MOSS proposed by Audibert and Bubeck [2010], have been studied and shown improvements on the regret guarantees. However, the regret guarantees of these algorithms have unbounded gaps to the lower bound. Recently, Maillard *et al.* [2011] and Garivier and Cappé [2011] have proposed a UCB algorithm based on the Kullback-Leibler divergence, kl-UCB, and proven it to be asymptotically optimal when all arms follow a Bernoulli distribution, i.e., they reach the lower bound by Lai and Robbins [1985]. They have generalized the algorithm to KL-UCB [Cappé *et al.*, 2013], which is asymptotically optimal under general distributions with bounded supports.

However, these UCB algorithms exhibit a complexity-optimality dilemma in the real world applications that are computationally sensitive. On one hand, the UCB1 algorithm enjoys closed-form updates per round while its regret gap to the lower bound can be unbounded. On the other hand, the kl-UCB algorithm is asymptotically optimal but it needs to solve a convex optimization problem for each arm at each round. Though there are many standard optimization tools to solve the convex optimization problem numerically, there is no regret guarantee for the implemented kl-UCB with arbitrary numerical accuracy. Practitioners usually set a sufficient accuracy (for example, 10^{-5}) so that the behaviors of the implemented kl-UCB converge to the theory. However, this means that the computational cost per round by kl-UCB can be out of budget for applications with a large number of arms. The complexity-optimality dilemma is because there is currently no available algorithm that can trade-off between complexity and optimality.

Such a dilemma occurs in a number of applications with a large K. For example, in an online recommendation system [Li *et al.*, 2010; Buccapatnam *et al.*, 2017], the algorithm needs to recommend an item from hundreds of thousands of items to a customer within a second. Another example is the use of bandit algorithms as a meta-algorithm for other machine learning problems, e.g., using bandits for classifier boosting [Busa-Fekete and Kégl, 2010]. The number of data points and features can be large.

Another scenario that the dilemma appears is in real-time applications such as robotic systems [Matikainen *et al.*, 2013], 2D planning [Laskey *et al.*, 2015] and portfolio optimization [Moeini *et al.*, 2016]. In these applications, a delayed decision may turn out to be catastrophic.

Cappé *et al.* [2013] proposed the open problem of finding a

	kl-UCB	UCBoost(ϵ)	UCBoost(D)	UCB1
Regret/$\log(T)$	$O\left(\sum_a \frac{\mu^*-\mu_a}{d_{kl}(\mu_a,\mu^*)}\right)$	$O\left(\sum_a \frac{\mu^*-\mu_a}{d_{kl}(\mu_a,\mu^*)-\epsilon}\right)$	$O\left(\sum_a \frac{\mu^*-\mu_a}{d_{kl}(\mu_a,\mu^*)-1/e}\right)$	$O\left(\sum_a \frac{\mu^*-\mu_a}{2(\mu^*-\mu_a)^2}\right)$
Complexity	unbounded	$O(\log(1/\epsilon))$	$O(1)$	O(1)

Table 1: Regret guarantee and computational complexity per arm per round of various algorithms

low-complexity optimal UCB algorithm, which has remained open till now. In this work, we make the following contributions to this open problem. (Table 1 summarizes the main results.)

- We propose a generic UCB algorithm. By plugging a semi-distance function, one can obtain a specific UCB algorithm with regret guarantee (Theorem 1). As a by-product, we propose two new UCB algorithms that are alternatives to UCB1 (Corollary 1 and 2).
- We propose a boosting algorithm, UCBoost, which can obtain a strong (i.e., with regret guarantee close to the lower bound) UCB algorithm from a set of weak (i.e., with regret guarantee far away from the lower bound) generic UCB algorithms (Theorem 2). By boosting a finite number of weak generic UCB algorithms, we find a UCBoost algorithm that enjoys the same complexity as UCB1 as well as a regret guarantee that is $1/e$-close to the kl-UCB algorithm (Corollary 3)[1]. That is to say, such a UCBoost algorithm is low-complexity and near-optimal under the Bernoulli case.
- We propose an approximation-based UCBoost algorithm, UCBoost(ϵ), that enjoys ϵ-optimal regret guarantee under the Bernoulli case and $O(\log(1/\epsilon))$ computational complexity for each arm at each round for any $\epsilon > 0$ (Theorem 3). This algorithm provides a non-trivial trade-off between complexity and optimality.

Related Work. There are other asymptotically optimal algorithms, such as Thompson Sampling [Agrawal and Goyal, 2012], Bayes-UCB [Kaufmann *et al.*, 2012] and DMED [Honda and Takemura, 2010]. However, the computations involved in these algorithms become non-trivial in non-Bernoulli cases. First, Bayesian methods, including Thompson Sampling, Information Directed Sampling [Russo and Van Roy, 2014; Liu *et al.*, 2017] and Bayes-UCB, require updating and sampling from the posterior distribution, which is computationally difficult for models other than exponential families [Korda *et al.*, 2013]. Second, the computational complexity of DMED policy is larger than UCB policies because the computation involved in DMED is formulated as a univariate convex optimization problem. In contrast, our algorithms are computationally efficient in general bounded support models and don't need the knowledge of prior information on the distributions of the arms.

Our work is also related to DMED-M proposed by Honda and Takemura [2012]. DMED-M uses the first d empirical moments to construct a lower bound of the objective function involved in DMED. As d goes to infinity, the lower bound converges to the objective function and DMED-M converges to DMED while the computational complexity increases. However, DMED-M has no explicit form when $d > 4$ and there is no guarantee on the regret gap to the optimality for any finite d. Unlike DEMD-M, our UCBoost algorithms can provide guarantees on the complexity and regret performance for arbitrary ϵ, which offers a controlled tradeoff between complexity and optimality.

Agarwal *et al.* [2017] proposed a boosting technique to obtain a strong bandit algorithm from the existing algorithms, that is adaptive to the environment. However, our boosting technique is specifically designed for stochastic setting and hence allows us to obtain near-optimal algorithms that have better regret gurantees than those obtained using the boosting technique by Agarwal *et al.* [2017].

2 Preliminaries

We consider a stochastic bandit problem with finitely many arms indexed by $a \in \mathcal{K} \triangleq \{1, \dots, K\}$, where K is a positive integer. Each arm a is associated with an unknown probability distribution v_a over the bounded support[2] $\Theta = [0, 1]$. At each time step $t = 1, 2, \dots$, the agent chooses an action A_t according to past observations (possibly using some independent randomization) and receives a reward $X_{A_t, N_{A_t}(t)}$ independently drawn from the distribution v_{A_t}, where $N_a(t) \triangleq \sum_{s=1}^{t} \mathbb{1}\{A_s = a\}$ denotes the number of times that arm a was chosen up to time t. Note that the agent can only observe the reward $X_{A_t, N_{A_t}(t)}$ at time t. Let $\bar{X}_a(t)$ be the empirical mean of arm a based on the observations up to time t.

For each arm a, we denote by μ_a the expectation of its associated probability distribution v_a. Let a^* be any optimal arm, that is $a^* \in \arg\max_{a \in \mathcal{K}} \mu_a$. We write μ^* as a shorthand notation for the largest expectation μ_{a^*} and denote the gap of the expected reward of arm a to μ^* as $\Delta_a = \mu^* - \mu_a$. The performance of a policy π is evaluated through the standard notion of expected regret, defined at time horizon T as

$$R^\pi(T) = \sum_{a \in \mathcal{K}} \Delta_a \mathbb{E}[N_a(T)]. \tag{1}$$

The goal of the agent is to minimize the expected regret.

Now, we introduce the concept of semi-distance functions, which measure the distance between two expectations of random variables over Θ, and show several related properties.

Definition 1. (Candidate semi-distance) *A function $d : \Theta \times \Theta \to \mathbb{R}$ is said to be a candidate semi-distance function if*

1. $d(p, p) \leq 0$, $\forall p \in \Theta$;
2. $d(p, q) \leq d(p, q')$, $\forall p \leq q \leq q' \in \Theta$;
3. $d(p, q) \geq d(p', q)$, $\forall p \leq p' \leq q \in \Theta$.

[1] Note that e is the natural number

[2] If the supports are bounded in another interval, rescale to [0,1].

Clearly, a candidate semi-distance function satisfies the monotone properties[3] of a distance function. However, it does not need to be non-negative and symmetric.

Definition 2. (Semi-distance) *A function $d : \Theta \times \Theta \to \mathbb{R}$ is said to be a semi-distance function if it is a candidate semi-distance function such that $d(p, q) \geq 0$, $\forall p, q \in \Theta$.*

A semi-distance function satisfies the non-negative condition, and is stronger than a candidate semi-distance function. The following lemma reveals a simple way to obtain a semi-distance function from a candidate semi-distance function.

Lemma 1. *If $d_1 : \Theta \times \Theta \to \mathbb{R}$ is a candidate semi-distance function and $d_2 : \Theta \times \Theta \to \mathbb{R}$ is a semi-distance function, then $\max(d_1, d_2)$ is a semi-distance function.*

Remark 1. *In particular, $d \equiv 0$ is a semi-distance function. So one can easily obtain a semi-distance function from a candidate semi-distance function.*

As discussed in Remark 1, a semi-distance function may not distinguish two different distributions. So we introduce the following strong notion of semi-distance functions.

Definition 3. (Strong semi-distance) *A function $d : \Theta \times \Theta \to \mathbb{R}$ is said to be a strong semi-distance function if it is a semi-distance function such that $d(p, q) = 0$ if and only if $p = q$.*

One can obtain a strong semi-distance function from a candidate semi-distance function as shown in Lemma 2.

Lemma 2. *If $d_1 : \Theta \times \Theta \to \mathbb{R}$ is a candidate semi-distance function and $d_2 : \Theta \times \Theta \to \mathbb{R}$ is a strong semi-distance function, then $\max(d_1, d_2)$ is a strong semi-distance function.*

A typical strong semi-distance function is the Kullback-Leibler divergence between two Bernoulli distributions,

$$d_{kl}(p, q) = p \log\left(\frac{p}{q}\right) + (1 - p) \log\left(\frac{1 - p}{1 - q}\right). \quad (2)$$

In this work, we are interested in semi-distance functions that are dominated by the KL divergence as mentioned above.

Definition 4. (kl-dominated) *A function $d : \Theta \times \Theta \to \mathbb{R}$ is said to be kl-dominated if $d(p, q) \leq d_{kl}(p, q)$, $\forall p, q \in \Theta$.*

Consider a set of candidate semi-distance functions. A formal definition of feasible set is presented in Definition 5.

Definition 5. (Feasible set) *A set D of functions from $\Theta \times \Theta$ to $\mathbb{R}$ is said to be feasible if $\max_{d \in D} d$ is a kl-dominated and strong semi-distance function.*

The following proposition shows a sufficient condition for a set to be feasible.

Proposition 1. *A set D of kl-dominated and candidate semi-distance functions from $\Theta \times \Theta$ to $\mathbb{R}$ is feasible if $\exists d \in D$ such that d is a strong semi-distance function.*

Note that we only need one of the functions to be a strong semi-distance function in order to have a feasible set. This allows us to consider some useful candidate semi-distance functions in our boosting approach.

[3]The monotone properties are equivalent to the triangle inequality in one-dimensional case.

Algorithm 1 The generic UCB algorithm

Require: semi-distance function d
  Initialization: t **from** 1 **to** K, play arm $A_t = t$.
  for t **from** $K + 1$ **to** T **do**
    Play arm $A_t = \arg\max_{a \in \mathcal{K}} \max\{q \in \Theta : N_a(t - 1) d(\bar{X}_a(t - 1), q) \leq \log(t) + c \log(\log(t))\}$
  end for

3 Boosting

We first present a generic form of UCB algorithm, which can generate a class of UCB algorithms. We then provide a boosting technique to obtain a good UCBoost algorithm based on these UCB algorithms.

3.1 The Generic UCB Algorithm

Algorithm 1 presents a generic form of UCB algorithm, which only uses the empirical means. The instantiation of the UCB algorithm requires a semi-distance function. Given a semi-distance function d, UCB(d) algorithm finds upper confidence bounds $\{u_a(t)\}_{a \in \mathcal{K}}$ such that the distance $d(\bar{X}_a(t - 1), u_a(t))$ is at most the exploration bonus $((\log(t) + c \log(\log(t)))/N_a(t - 1))$ for any arm a. Note that c is a constant to be determined. In other words, $u_a(t)$ is the solution of the following optimization problem $P_1(d)$,

$$P_1(d) : \max_{q \in \Theta} \; q \quad s.t. \; d(p, q) \leq \delta, \quad (3)$$

where $p \in \Theta$ is the empirical mean and $\delta > 0$ is the exploration bonus. The computational complexity of the UCB(d) algorithm depends on the complexity of solving the problem $P_1(d)$. The following result shows that the regret of the UCB(d) algorithm depends on the semi-distance function d.

Theorem 1. *If $d : \Theta \times \Theta \to \mathbb{R}$ is a strong semi-distance function and is also kl-dominated, then the regret of the UCB(d) algorithm when $c = 3$ satisfies:*

$$\limsup_{T \to \infty} \frac{\mathbb{E}[R^{\mathrm{UCB}(d)}(T)]}{\log T} \leq \sum_{a : \mu_a < \mu^*} \frac{\Delta_a}{d(\mu_a, \mu^*)}. \quad (4)$$

Theorem 1 is a generalization of the regret gurantee of kl-UCB proposed by Garivier and Cappé [2011], which is recovered by UCB(d_{kl}). Recall that d_{kl} is the KL divergence between two Bernoulli distributions. Note that Theorem 1 holds for general distributions over the support Θ. If the reward distributions are Bernoulli, the kl-UCB algorithm is asymptotically optimal in the sense that the regret of kl-UCB matches the lower bound provided by Lai and Robbins [1985]:

$$\liminf_{T \to \infty} \frac{\mathbb{E}[R^{\pi}(T)]}{\log T} \geq \sum_{a : \mu_a < \mu^*} \frac{\Delta_a}{d_{kl}(\mu_a, \mu^*)}. \quad (5)$$

However, there is no closed-form solution to the problem $P_1(d_{kl})$. Practical implementation of kl-UCB needs to solve $P_1(d_{kl})$ via numerical methods with high accuracy, which means that the computational complexity is non-trivial.

In addition to the KL divergence function d_{kl}, we can find other kl-dominated and strong semi-distance functions such that the complexity of solving $P_1(d)$ is $O(1)$. Then we can obtain some low-complexity UCB algorithms with possibly

Algorithm 2 UCBoost

Require: candidate semi-distance function set D
Initialization: t **from** 1 **to** K, play arm $A_t = t$.
for t **from** $K+1$ **to** T **do**
Play arm $A_t = \arg\max_{a\in\mathcal{K}} \min_{d\in D} \max\{q \in \Theta : N_a(t-1)d(\bar{X}_a(t-1), q) \leq \log(t) + c\log(\log(t))\}$
end for

weak regret performance. For example, consider the l_2 distance function, $d_{sq}(p,q) = 2(p-q)^2$. It is clear that d_{sq} is a kl-dominated and strong semi-distance function. Note that UCB(d_{sq}) recovers the traditional UCB1 [Auer *et al.*, 2002].

Now, we introduce two alternative functions to the function d_{sq}: biquadratic distance function and Hellinger distance function. The biquadratic distance function is $d_{bq}(p,q) = 2(p-q)^2 + \frac{4}{9}(p-q)^4$. The Hellinger distance function[4] is $d_h(p,q) = \left(\sqrt{p}-\sqrt{q}\right)^2 + \left(\sqrt{1-p}-\sqrt{1-q}\right)^2$. As shown in Lemma 3 and Lemma 4, they are kl-dominated and strong semi-distance functions and the solutions of the corresponding $P_1(d)$ have closed forms.

Lemma 3. *The biquadratic distance function d_{bq} is a kl-dominated and strong semi-distance function. The solution of $P_1(d_{bq})$ is* $q^* = \min\left\{1, p + \sqrt{-\frac{9}{4} + \sqrt{\frac{81}{16} + \frac{9}{4}\delta}}\right\}$.

Lemma 4. *The Hellinger distance function d_h is a kl-dominated and strong semi-distance function. The solution of $P_1(d_h)$ is* $q^* =$

$$\left(\left(1-\frac{\delta}{2}\right)\sqrt{p} + \sqrt{(1-p)\left(\delta - \frac{\delta^2}{4}\right)}\right)^{2\times\mathbb{1}\{\delta<2-2\sqrt{p}\}},$$

where $\mathbb{1}\{\cdot\}$ is the indicator function.

The following result follows from Theorem 1 and Lemma 3. Note that UCB(d_{bq}) enjoys the same complexity of UCB1 and better regret guarantee than UCB1.

Corollary 1. *If $c = 3$, then the regret of UCB(d_{bq}) satisfies*

$$\limsup_{T\to\infty} \frac{\mathbb{E}[R^{\text{UCB}(d_{bq})}(T)]}{\log T} \leq \sum_{a:\mu_a<\mu^*} \frac{\Delta_a}{d_{bq}(\mu_a,\mu^*)}. \quad (6)$$

The following result follows from Theorem 1 and Lemma 4. Note that UCB(d_h) enjoys the same complexity of UCB1. In terms of regret guarantees, no one dominates the other.

Corollary 2. *If $c = 3$, then the regret of UCB(d_h) satisfies*

$$\limsup_{T\to\infty} \frac{\mathbb{E}[R^{\text{UCB}(d_h)}(T)]}{\log T} \leq \sum_{a:\mu_a<\mu^*} \frac{\Delta_a}{d_h(\mu_a,\mu^*)}. \quad (7)$$

3.2 The UCBoost Algorithm

The generic UCB algorithm provides a way of generating UCB algorithms from semi-distance functions. Among the class of semi-distance functions, some have closed-form solutions of the corresponding problems $P_1(d)$. Thus, the corresponding algorithm UCB(d) enjoys $O(1)$ computational complexity for each arm in each round. However, these UCB(d) algorithms are weak in the sense that the regret guarantees of these UCB(d) algorithms are worse than that of kl-UCB. Moreover, the decision maker does not know which weak UCB(d) is better when the information $\{\mu_a\}_{a\in\mathcal{K}}$ is unknown. A natural question is: *is there a boosting technique that one can use to obtain a stronger UCB algorithm from these weak UCB algorithms?* The following regret result of Algorithm 2 offers a positive answer.

Theorem 2. *If D is a feasible set, then the regret of UCBoost(D) when $c = 3$ satisfies:*

$$\limsup_{T\to\infty} \frac{\mathbb{E}[R^{\text{UCBoost}(D)}(T)]}{\log T} \leq \sum_{a:\mu_a<\mu^*} \frac{\Delta_a}{\max_{d\in D} d(\mu_a,\mu^*)}.$$

The UCBoost algorithm works as the following. Given a feasible set D of candidate semi-distance functions, UCBoost(D) algorithm queries the upper confidence bound of each weak UCB(d) once and takes the minimum as the upper confidence bound. Suppose that for any $d \in D$, UCB(d) enjoys $O(1)$ computational complexity for each arm in each round. Then, UCBoost(D) enjoys $O(|D|)$ computational complexity for each arm in each round, where $|D|$ is the cardinality of set D. Theorem 2 shows that UCBoost(D) has a regret guarantee that is no worse than any UCB(d) such that $d \in D$. Hence, the UCBoost algorithm can obtain a stronger UCB algorithm from some weak UCB algorithms. Moreover, the following remark shows that the ensemble does not deteriorate the regret performance.

Remark 2. *If D_1 and D_2 are feasible sets, and $D_1 \subset D_2$, then the regret guarantee of UCBoost(D_2) is no worse than that of UCBoost(D_1).*

By Theorem 2, UCBoost($\{d_{bq}, d_h\}$) enjoys the same complexity as UCB1, UCB(d_{bq}) and UCB(d_h), and has a no worse regret guarantee. However, the gap between the regret guarantee of UCBoost($\{d_{bq}, d_h\}$) and that of kl-UCB may still be large since d_{bq} and d_h are bounded while d_{kl} is unbounded. To address this problem, we are ready to introduce a candidate semi-distance function that is kl-dominated and unbounded. The candidate semi-distance function is a lower bound of the KL divergence function d_{kl}, $d_{lb}(p,q) = p\log(p) + (1-p)\log\left(\frac{1-p}{1-q}\right)$.

Lemma 5. *The function d_{lb} is a kl-dominated and candidate semi-distance function. The solution of $P_1(d_{lb})$ is*

$$q^* = 1 - (1-p)\exp\left(\frac{p\log(p) - \delta}{1-p}\right). \quad (8)$$

By Lemma 3-5, Proposition 1 and Theorem 2, we have the following result.

Corollary 3. *If $D = \{d_{bq}, d_h, d_{lb}\}$, then the regret of UCBoost(D) when $c = 3$ satisfies:*

$$\limsup_{T\to\infty} \frac{\mathbb{E}[R^{\text{UCBoost}(D)}(T)]}{\log T} \leq \sum_{a:\mu_a<\mu^*} \frac{\Delta_a}{\max_{d\in D} d(\mu_a,\mu^*)}.$$

Note that $d_{kl}(\mu_a,\mu^*) - 1/e \leq \max_{d\in D} d(\mu_a,\mu^*) \leq d_{kl}(\mu_a,\mu^*)$ for any $a \in \mathcal{K}$ such that $\mu_a < \mu^*$. Thus, we have that

$$\limsup_{T\to\infty} \frac{\mathbb{E}[R^{\text{UCBoost}(D)}(T)]}{\log T} \leq \sum_{a:\mu_a<\mu^*} \frac{\Delta_a}{d_{kl}(\mu_a,\mu^*) - 1/e}.$$

[4] Actually, d_h is 2 times the square of the Hellinger distance.

Although d_{lb} is not a strong semi-distance function, the set $D = \{d_{bq}, d_h, d_{lb}\}$ is still feasible by Proposition 1. The advantage of introducing d_{lb} is that its tightness to d_{kl} improves the regret guarantee of the algorithm. To be specific, the gap between $d_{lb}(\mu_a, \mu^*)$ and $d_{kl}(\mu_a, \mu^*)$ is $\mu_a \log(1/\mu^*)$, which is uniformly bounded by $1/e$ since $\mu_a < \mu^*$. Note that e is the natural number. Hence, UCBoost($\{d_{bq}, d_h, d_{lb}\}$) achieves near-optimal regret performance with low complexity.

3.3 The UCBoost(ϵ) Algorithm

First, we show an approximation of the KL divergence function d_{kl}. Then we design a UCBoost algorithm based on the approximation, which enjoys low complexity and regret guarantee that is arbitrarily close to that of kl-UCB.

Recall that $p \in \Theta$ and $\delta > 0$ are the inputs of the problem $P_1(d_{kl})$. Given any approximation error $\epsilon > 0$, let $\eta = \frac{\epsilon}{1+\epsilon}$ and $q_k = 1 - (1-\eta)^k \in \Theta$ for any $k \geq 0$. Then there exist $\tau_1(p) = \left\lceil \frac{\log(1-p)}{\log(1-\eta)} \right\rceil$ and $\tau_2(p) = \left\lceil \frac{\log(1-\exp(-\epsilon/p))}{\log(1-\eta)} \right\rceil$ such that $p \leq q_k \leq \exp(-\epsilon/p)$ if $\tau_1(p) \leq k \leq \tau_2(p)$. For each $\tau_1(p) \leq k \leq \tau_2(p)$, we construct a step function, $d_s^k(p,q) = d_{kl}(p, q_k)\mathbb{1}\{q > q_k\}$. These step functions can approximate the function d_{kl} on the interval $[p, \exp(-\epsilon/p)]$. Then we use d_{lb} to approximate d_{kl} on the interval $[\exp(-\epsilon/p), 1]$. The following result shows that the step function $d_s^k(p,q)$ is a kl-dominated and semi-distance function.

Lemma 6. *For each $k \geq \tau_1(p)$, the step function $d_s^k(p,q)$ is a kl-dominated and semi-distance function. The solution of $P_1(d_s^k)$ is $q^* = q_k^{\mathbb{1}\{\delta < d_{kl}(p,q_k)\}}$.*

Let $D(p) = \{d_{sq}, d_{lb}, d_s^{\tau_1(p)}, d_s^{\tau_1(p)+1}, \ldots, d_s^{\tau_2(p)}\}$. Then the following result shows that the envelope $\max_{d \in D(p)} d$ is an ϵ-approximation of the function d_{kl} on the interval $[p, 1]$.

Proposition 2. *Given $p \in \Theta$ and $\epsilon > 0$. Let $D(p) = \{d_{sq}, d_{lb}, d_s^{\tau_1(p)}, d_s^{\tau_1(p)+1}, \ldots, d_s^{\tau_2(p)}\}$. For any $q \in [p, 1]$, we have that $0 \leq d_{kl}(p,q) - \max_{d \in D(p)} d(p,q) \leq \epsilon$.*

Lemma 6 and Proposition 2 allow us to bound the regret of the UCBoost algorithm based on the approximation, which is shown in the following result.

Theorem 3. *Given any $\epsilon > 0$, let $D = \{d_{sq}, d_{lb}\} \cup \{d_s^k : k \geq 0\}$. The regret of UCBoost(D) with $c = 3$ that restricts D to $D(p)$ for each arm with empirical mean p, satisfies*

$$\limsup_{T \to \infty} \frac{\mathbb{E}[R^{\text{UCBoost}(D)}(T)]}{\log T} \leq \sum_{a:\mu_a < \mu^*} \frac{\Delta_a}{d_{kl}(\mu_a, \mu^*) - \epsilon}. \tag{9}$$

The complexity for each arm per round is $O(\log(\frac{1}{\epsilon}))$.

We denote the algorithm described in Theorem 3 as UCBoost(ϵ) for shorthand. The UCBoost(ϵ) algorithm offers an efficient way to trade regret performance with complexity.

Remark 3. *The practical implementation of kl-UCB needs numerical methods for searching the q^* of $P_1(d_{kl})$ with some sufficiently small error ϵ. For example, the bisection search can find a solution q' such that $|q' - q^*| \leq \epsilon$ with $O(\log(\frac{1}{\epsilon}))$ iterations. However, there is no regret guarantee of the implemented kl-UCB when ϵ is arbitrary. Our UCBoost(ϵ) algorithm fills this gap and bridges computational complexity to regret performance. Moreover, the empirical performance of the implemented kl-UCB when ϵ is relatively large, becomes unreliable. This is because the gap $|d_{kl}(p,q^*) - d_{kl}(p,q')|$ is unbounded even though $|q' - q^*|$ is bounded. On the contrary, our approximation method guarantees bounded KL divergence gap, thus allowing reliable regret performance.*

4 Numerical Results

In this section, we support our results by numerical experiments that compare our algorithms with the baseline algorithms in three scenarios. All the algorithms are run exactly as described in the previous sections. For implementation of kl-UCB, we use the MATLAB code in py/maBandits package developed by Cappé *et al.* [2012]. All the other algorithms are also implemented using MATLAB for fairness. Note that we choose $c = 0$ in the experiments as suggested by Garivier and Cappé [2011]. All the results are obtained from $10,000$ independent runs of the algorithms.

Bernoulli Scenario 1. We first consider the basic scenario with Bernoulli rewards. There are $K = 9$ arms with expectations $\mu_i = i/10$ for each arm i. The average regret of various algorithms as a function of time is shown in Figure 1a.

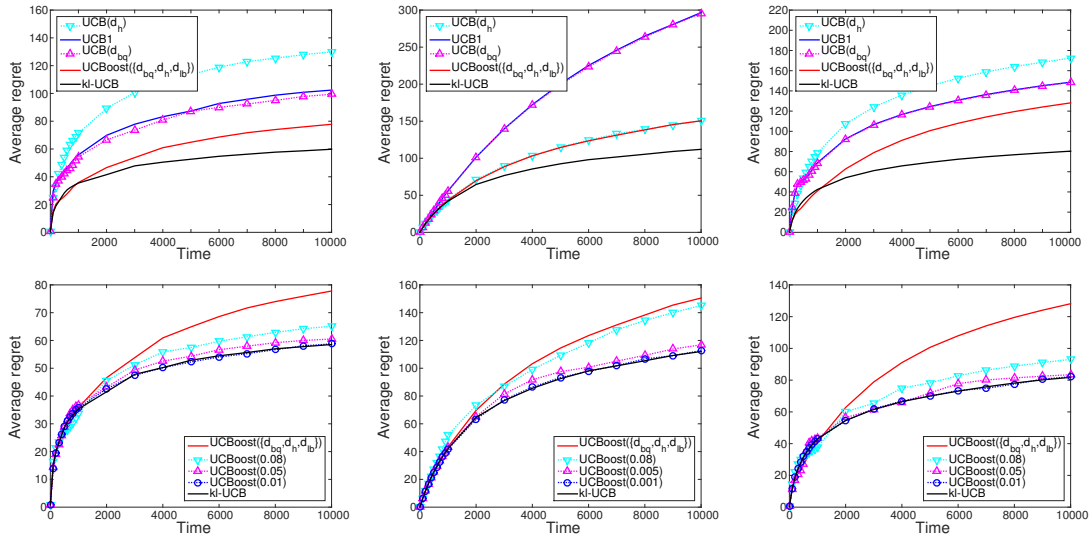
First, UCB(d_{bq}) performs as expected, though it is slightly better than UCB1. However, UCB(d_h) performs worse than UCB1 in this scenario. The reason is that the regret guarantee of UCB(d_h) under this scenario is worse than that of UCB1.

Second, the performance of UCBoost($\{d_{bq}, d_h, d_{lb}\}$) is between that of UCB1 and kl-UCB. UCBoost($\{d_{bq}, d_h, d_{lb}\}$) outperforms UCB(d_h) and UCB(d_{bq}) as expected, which demonstrates the power of boosting. The candidate semi-distance function d_{lb} plays an important role in improving the regret performance.

Third, UCBoost(ϵ) algorithm fills the gap between UCBoost($\{d_{bq}, d_h, d_{lb}\}$) and kl-UCB with moderate ϵ. As ϵ decreases, UCBoost(ϵ) approaches to kl-UCB, which verifies our result in Theorem 3. When $\epsilon = 0.01$, UCBoost(ϵ) matches the regret of kl-UCB. Note that the numerical method for kl-UCB, such as Newton method and bisection search, usually needs the accuracy to be at least 10^{-5}. Otherwise, the regret performance of kl-UCB becomes unreliable. Compared to kl-UCB, UCBoost(ϵ) can achieve the same regret performance with less complexity by efficiently bounding the KL divergence gap.

Bernoulli Scenario 2. We consider a more difficult scenario of Bernoulli rewards, where the expectations are very low. This scenario has been considered by Garivier and Cappé [2011] to model the situations like online recommendations and online advertising. For example, in Yahoo! Front Page Today experiments [Li *et al.*, 2010], the rewards are the click through rates of the news and articles. The rewards are binary and the average click through rates are very low. In this scenario, we consider ten arms, with $\mu_1 = \mu_2 = \mu_3 = 0.01$, $\mu_4 = \mu_5 = \mu_6 = 0.02$, $\mu_7 = \mu_8 = \mu_9 = 0.05$ and $\mu_{10} = 0.1$. Figure 1b shows the average regret of various algorithms as a function of time.

First, the performance of UCB(d_{bq}) is the same as UCB1. This is because the term Δ_a^4 vanishes for all suboptimal arms in this scenario. So the improvement of UCB(d_{bq}) over UCB1 vanishes as well. However, UCB(d_h) outperforms UCB1



(a) Bernoulli scenario 1 (b) Bernoulli scenario 2 (c) Beta scenario

Figure 1: Regret of the various algorithms as a function of time in three scenarios.

Scenario	kl-UCB	UCBoost(ϵ) $\epsilon = 0.01(0.001)$	UCBoost(ϵ) $\epsilon = 0.05(0.005)$	UCBoost(ϵ) $\epsilon = 0.08$	UCBoost($\{d_{bq}, d_h, d_{lb}\}$)	UCB1
Bernoulli 1	$933\mu s$	$7.67\mu s$	$6.67\mu s$	$5.78\mu s$	$1.67\mu s$	$0.31\mu s$
Bernoulli 2	$986\mu s$	$8.76\mu s$	$7.96\mu s$	$6.27\mu s$	$1.60\mu s$	$0.30\mu s$
Beta	$907\mu s$	$8.33\mu s$	$6.89\mu s$	$5.89\mu s$	$2.01\mu s$	$0.33\mu s$

Table 2: Average computational time for each arm per round of various algorithms.

because the Hellinger distance between μ_a and μ^* is much larger than the l_2 distance in this scenario. So UCB(d_h) enjoys better regret performance than UCB1 in this scenario.

Second, UCBoost($\{d_{bq}, d_h, d_{lb}\}$) performs as expected and is between UCB1 and kl-UCB. Although the gap between UCB1 and kl-UCB becomes larger when compared to Bernoulli scenario 1, the gap between UCBoost($\{d_{bq}, d_h, d_{lb}\}$) and kl-UCB remains. This verifies our result in Corollary 3 that the gap between the constants in the regret guarantees is bounded by $1/e$. This result also demonstrates the power of boosting in that UCBoost($\{d_{bq}, d_h, d_{lb}\}$) performs no worse than UCB(d_h) and UCB(d_{bq}) in all cases.

Third, UCBoost(ϵ) algorithm fills the gap between UCBoost($\{d_{bq}, d_h, d_{lb}\}$) and kl-UCB, which is consistent with the results in Bernoulli scenario 1. The regret of UCBoost(ϵ) matches with that of kl-UCB when $\epsilon = 0.001$. Compared to the results in Bernoulli scenario 1, we need more accurate approximation for UCBoost when the expectations are lower. However, this accuracy is moderate compared to the requirements in numerical methods for kl-UCB.

Beta Scenario. Our results in the previous sections hold for any distributions with bounded support. In this scenario, we consider $K = 9$ arms with Beta distributions. More precisely, each arm $1 \leq i \leq 9$ is associated with Beta(α_i,β_i) distribution such that $\alpha_i = i$ and $\beta_i = 2$. Note that the expectation of Beta(α_i,β_i) is $\alpha_i/(\alpha_i + \beta_i)$. The regret results shown in Figure 1c are consistent with that of Bernoulli scenario 1.

Computational time. We obtain the average running time for each arm per round by measuring the total computational time of $10,000$ independent runs of each algorithms in each scenario. Note that kl-UCB is implemented by the py/maBandits package developed by Cappé *et al.* [2012], which sets accuracy to 10^{-5} for the Newton method. The average computational time results are shown in Table 2. The average running time of UCBoost(ϵ) that matches the regret of kl-UCB is no more than 1% of the time of kl-UCB.

5 Conclusion

In this work, we introduce the generic UCB algorithm and provide the regret guarantee for any UCB algorithm generated by a kl-dominated strong semi-distance function. Then, we propose a boosting framework, UCBoost, to boost any set of generic UCB algorithms. We find a specific finite set D, such that UCBoost(D) enjoys $O(1)$ complexity for each arm per round as well as regret guarantee that is $1/e$-close to the kl-UCB algorithm. Finally, we propose an approximation-based UCBoost algorithm, UCBoost(ϵ), that enjoys regret guarantee ϵ-close to that of kl-UCB as well as $O(\log(1/\epsilon))$ complexity for each arm per round. This algorithm bridges the regret guarantee to the computational complexity, thus offering an efficient trade-off between regret performance and complexity for practitioners. By experiments, we show that UCBoost(ϵ) can achieve the same regret performance as standard kl-UCB with only 1% computational cost of kl-UCB.

Acknowledgments

This work has been supported in part by grants from the Army Research Office W911NF-14-1-0368 and MURI W911NF-12-1-0385, and grants from the Office of Naval Research N00014-17-1-2417 and N00014-15-1-2166.

References

[Agarwal *et al.*, 2017] Alekh Agarwal, Haipeng Luo, Behnam Neyshabur, and Robert E Schapire. Corralling a band of bandit algorithms. In *Conference on Learning Theory*, pages 12–38, 2017.

[Agrawal and Goyal, 2012] Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *COLT*, pages 39–1, 2012.

[Audibert and Bubeck, 2010] Jean-Yves Audibert and Sébastien Bubeck. Regret bounds and minimax policies under partial monitoring. *Journal of Machine Learning Research*, 11(Oct):2785–2836, 2010.

[Audibert *et al.*, 2009] Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902, 2009.

[Auer *et al.*, 2002] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[Buccapatnam *et al.*, 2017] Swapna Buccapatnam, Fang Liu, Atilla Eryilmaz, and Ness B Shroff. Reward maximization under uncertainty: Leveraging side-observations on networks. *arXiv preprint arXiv:1704.07943*, 2017.

[Burnetas and Katehakis, 1996] Apostolos N Burnetas and Michael N Katehakis. Optimal adaptive policies for sequential allocation problems. *Advances in Applied Mathematics*, 17(2):122–142, 1996.

[Busa-Fekete and Kégl, 2010] Róbert Busa-Fekete and Balázs Kégl. Fast boosting using adversarial bandits. In *27th International Conference on Machine Learning (ICML 2010)*, pages 143–150, 2010.

[Cappé *et al.*, 2012] Olivier Cappé, Aurélien Garivier, and Emilie Kaufmann. py/mabandits: Matlab and python packages for multi-armed bandits. http://mloss.org/software/view/415/, 2012.

[Cappé *et al.*, 2013] Olivier Cappé, Aurélien Garivier, Odalric-Ambrym Maillard, Rémi Munos, and Gilles Stoltz. Kullback–leibler upper confidence bounds for optimal sequential allocation. *The Annals of Statistics*, 41(3):1516–1541, 2013.

[Garivier and Cappé, 2011] Aurélien Garivier and Olivier Cappé. The kl-ucb algorithm for bounded stochastic bandits and beyond. In *Proceedings of the 24th annual Conference On Learning Theory*, pages 359–376, 2011.

[Honda and Takemura, 2010] Junya Honda and Akimichi Takemura. An asymptotically optimal bandit algorithm for bounded support models. In *COLT*, pages 67–79, 2010.

[Honda and Takemura, 2012] Junya Honda and Akimichi Takemura. Stochastic bandit based on empirical moments. In *Artificial Intelligence and Statistics*, pages 529–537, 2012.

[Kaufmann *et al.*, 2012] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On bayesian upper confidence bounds for bandit problems. In *Artificial Intelligence and Statistics*, pages 592–600, 2012.

[Korda *et al.*, 2013] Nathaniel Korda, Emilie Kaufmann, and Remi Munos. Thompson sampling for 1-dimensional exponential family bandits. In *Advances in Neural Information Processing Systems*, pages 1448–1456, 2013.

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

[Laskey *et al.*, 2015] Michael Laskey, Jeff Mahler, Zoe McCarthy, Florian T Pokorny, Sachin Patil, Jur Van Den Berg, Danica Kragic, Pieter Abbeel, and Ken Goldberg. Multi-armed bandit models for 2d grasp planning with uncertainty. In *Automation Science and Engineering (CASE), 2015 IEEE International Conference on*, pages 572–579. IEEE, 2015.

[Li *et al.*, 2010] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.

[Liu *et al.*, 2017] Fang Liu, Swapna Buccapatnam, and Ness Shroff. Information directed sampling for stochastic bandits with graph feedback. *arXiv preprint arXiv:1711.03198*, 2017.

[Maillard *et al.*, 2011] Odalric-Ambrym Maillard, Rémi Munos, and Gilles Stoltz. A finite-time analysis of multi-armed bandits problems with kullback-leibler divergences. In *Proceedings of the 24th annual Conference On Learning Theory*, pages 497–514, 2011.

[Matikainen *et al.*, 2013] Pyry Matikainen, P Michael Furlong, Rahul Sukthankar, and Martial Hebert. Multi-armed recommendation bandits for selecting state machine policies for robotic systems. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 4545–4551. IEEE, 2013.

[Moeini *et al.*, 2016] Mahdi Moeini, Oliver Wendt, and Linus Krumrey. Portfolio optimization by means of a χ-armed bandit algorithm. In *Asian Conference on Intelligent Information and Database Systems*, pages 620–629. Springer, 2016.

[Russo and Van Roy, 2014] Dan Russo and Benjamin Van Roy. Learning to optimize via information-directed sampling. In *Advances in Neural Information Processing Systems*, pages 1583–1591, 2014.

[Thompson, 1933] William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.