Empirical Bayes Regret Minimization

Chih-Wei Hsu
Google Research

Branislav Kveton
Google Research

Ofer Meshi
Google Research

Martin Mladenov
Google Research

Csaba Szepesvári
DeepMind

Abstract

The prevalent approach to bandit algorithm design is to have a low-regret algorithm by design. While celebrated, this approach is often conservative because it ignores many intricate properties of actual problem instances. In this work, we pioneer the idea of minimizing an empirical approximation to the Bayes regret, the expected regret with respect to a distribution over problems. This approach can be viewed as an instance of learning-to-learn, it is conceptually straightforward, and easy to implement. We conduct a comprehensive empirical study of empirical Bayes regret minimization in a wide range of bandit problems, from Bernoulli bandits to structured problems, such as generalized linear and Gaussian process bandits. We report significant improvements over state-of-the-art bandit algorithms, often by an order of magnitude, by simply optimizing over a sample from the distribution.

1 Introduction

A stochastic bandit [29, 5, 30] is an online learning problem where the learning agent acts by pulling arms, each associated with a noisy reward. The goal of the agent is to maximize its expected cumulative reward. Since the agent does not know the expected rewards of the arms in advance, it must learn them by pulling the arms. This results in the well-known exploration-exploitation trade-off: explore, and learn more about an arm; or exploit, and pull the arm with the highest estimated reward thus far. In practice, the arm may be a treatment in a clinical trial and its reward is the outcome of that treatment on some patient population.

The most celebrated and studied exploration designs in stochastic multi-armed bandits are optimism in the face of uncertainty [5, 16, 1] and Thompson sampling (TS) [46, 3]. In many classes of problems, these strategies have a matching lower bound, which may indicate that there is no more work left to be done. We argue that this is not necessarily the case because of a mismatch of goals. In particular, in this work we are interested in the Bayes regret, which models the situation when one needs an algorithm that performs well on an unknown bandit instance chosen at random from some fixed distribution, while the said optimality results reflect either an asymptotic or a worst-case viewpoint. While these provide strong guarantees, they are not necessarily the best choice for the setting we consider.

In this work, assuming access to a sequence of bandit instances sampled independently from a common underlying distribution, which may be unknown, we propose to use these bandit instances to evaluate bandit algorithms based on data generated from the instances, and to eventually select the empirically best algorithm. Our approach can be seen as an instance of meta-learning, or learning-to-learn [47, 48, 9, 10]. It can also be seen as a straightforward application of empirical risk minimization to learning what bandit algorithm to use and hence we call this approach empirical Bayes regret minimization.

We make the following contributions in this paper. First, we review current algorithm designs in stochastic bandits and observe that although they are theoretically motivated and sometimes have matching lower bounds, they may generally be too conservative in their exploration strategy. Second, we propose a new approach for using problem instances sampled from a distribution over problems. The key idea is to choose an algorithm design that minimizes the average regret over the sampled problems. This can be viewed as minimizing the empirical approximation to the Bayes regret, the average regret over problems that are drawn i.i.d. from the distribution. Finally, we conduct an extensive empirical evaluation of
our approach in multi-armed, linear, generalized linear, and Gaussian process bandits. In most problems, our approach leads to significant improvements, which justifies more empirical designs, as we argue for in this paper.

2 Notation and Setting

We begin by describing our notation and providing background to our work. Throughout the paper we will use \( \mathbb{E} [\cdot] \) to denote the expectation operator, while we use \( \mathbb{P} \) to denote the corresponding probability measure. We let \( M_1 (X) \) to be the set of probability measures supported on set \( X \). We also define \([n] = \{1, \ldots, n\}\). The \( i \)-th component of vector \( x \) is denoted by \( x_i \).

We study the problem of cumulative regret minimization in stochastic multi-armed bandits. Formally, a stochastic multi-armed bandit [29, 5, 30] is an online learning problem where the learning agent interacts with an environment repeatedly by pulling one of \( K \) arms. Let \( (Y_t)_{t=1}^\infty \) be a sequence of \( K \)-tuples of arm rewards drawn i.i.d. from some distribution \( P \in \mathcal{S} \), where \( \mathcal{S} \subset M_1 ([0, 1]^K) \) is a subset of distributions on the \( K \)-dimensional unit cube \([0, 1]^K\). We denote the mean of \( P \) by \( \mu \in [0, 1]^K \).

Thus, \( \mu_i \) is the expected value of \( Y_{t,i} \), \( \mathbb{E} [Y_{t,i}] \). In round \( t \in [n] \), a learning agent pulls arm \( I_t \in [K] \) and observes its reward \( Y_{t,I_t} \). The agent does not know the distribution \( P \), or the expected rewards of the arms \( \mu_i \), in advance but can learn about these by pulling the arms.

The goal of the agent is to maximize its cumulative reward, or, equivalently, to minimize the \( n \)-round regret,

\[
R_n = \sum_{t=1}^n Y_{t,i^*(P)} - \sum_{t=1}^n Y_{t,I_t},
\]

where \( i^*(P) = \arg \max_{i \in [K]} \mu_i \) is the index of the optimal arm. In the setting we consider, \( P \) is also randomly chosen from a distribution over distributions, which we denote by \( \mathcal{P} \) and we define the goal as minimizing the expected regret, \( \mathbb{E} [R_n] \). Note that there are three sources of randomness in \( R_n \): \( P \) is random, the data \( Y_1, \ldots, Y_n \) are random for a fixed \( P \), and \( A \) can also randomize. The expectation “integrates out” all three sources of randomness. With this, this problem becomes an instance of the so-called Bayesian bandit setting [11] and to distinguish \( \mathbb{E} [R_n | P] \) from \( \mathbb{E} [R_n] \), we call \( \mathbb{E} [R_n | P] \) the expected regret on instance \( P \), while we call \( \mathbb{E} [R_n] \) the Bayes regret.

For future reference, to emphasize the dependence of the regret on the algorithm \( A \), we use \( R_n (A, P) \) to denote the random cumulative regret of \( A \) on \( P \), defined in (1).

We say that algorithm \( A \) has low \( n \)-round Bayes regret [42] if

\[
R_n (A) := \mathbb{E} [R_n (A, P)] \leq \varepsilon
\]

for some small \( \varepsilon > 0 \). Here, by the tower rule of expectations, \( \mathbb{E} [R_n (A, P)] = \mathbb{E} [\mathbb{E} [R_n (A, P) | P]] \), where in the last expression, the outer expectation is over \( P \sim \mathcal{P} \).

3 Empirical Bayes Regret Minimization

The key property of the Bayes regret, which we leverage in this work, is that it contains additional averaging over problems \( P \sim \mathcal{P} \). Suppose that the learning algorithm has access to an i.i.d. sequence \( P_1, \ldots, P_m \) of \( m \) bandit instances sampled from \( \mathcal{P} \). Let \( R_n (A, P_j) \) denote the random regret of \( A \) on instance \( P_j \), as defined in (1). Let

\[
\hat{R}_{n,m} (A) = \frac{1}{m} \sum_{j=1}^m R_n (A, P_j)
\]

be the corresponding empirical Bayes regret. The idea is to minimize (3) instead of \( \mathbb{E} [R_n (A, P_j)] \), where \( P_j \sim \mathcal{P} \). Note that for any fixed algorithm \( A \), \( \mathbb{E} [\hat{R}_{n,m} (A)] = \mathbb{E} [R_n (A, P_j)] \). Thus, minimizing \( \hat{R}_{n,m} (A) \) over \( A \) is akin to standard empirical risk minimization.

Note that \( \hat{R}_{n,m} (A) \) can be minimized without having access to the distributions \( P_j \). It is sufficient to have access to arm rewards, \( Y_{1,1}^{(j)}, \ldots, Y_{n,1}^{(j)} \sim P_j \). To see this, note that \( \sum_{t=1}^n Y_{t,I_t} \) is independent of the choice of the algorithm. Hence, minimization of the Bayes regret is equivalent to maximizing the total expected reward. Similarly, the minimizers of \( \hat{R}_{n,m} (A) \) are the same as the maximizers of the total reward across all problems, \( \sum_{j=1}^m \sum_{t=1}^n Y_{t,I_t} \), which can be optimized without having access to \( i^*(P_j) \), for \( j \in [m] \).

3.1 Generalization

For simplicity, assume that \( A \) is chosen from a finite set \( \mathcal{A} \) of algorithms. Let \( A_{n,m} = \arg \min_{A \in \mathcal{A}} \hat{R}_{n,m} (A) \) be the best empirical choice and \( A^* = \arg \min_{A \in \mathcal{A}} R_n (A) \) be the minimizer of the Bayes regret over \( \mathcal{A} \). Using standard reasoning, we get the following result.

Theorem 1. For any \( \delta \in (0, 1) \) with probability \( 1 - \delta \),

\[
R_n (A_{n,m}) \leq R_n (A^*) + \frac{2n^2 \log (|\mathcal{A}| / \delta)}{m}.
\]
Proof. The proof is standard and is included for completeness. First, recall that by Hoeffding’s inequality, with probability \(1 - \delta\), for any fixed algorithm \(A\),

\[
R_n(A) - \hat{R}_{n,m}(A) \leq C_{n,m}(\delta),
\]

where \(C_{n,m}(\delta) = 2\sqrt{n^2 \log(1/\delta)} / 2m\) because the random regret is in \([-n, n]\). Further, the same inequality holds if we change the left-hand side to \(\hat{R}_{n,m}(A) - R_n(A)\). By a union bound, with probability at least \(1 - \delta\),

\[
R_n(A) - \hat{R}_{n,m}(A) \leq C_{n,m}(\delta/|A|)
\]

for all algorithms \(A\). Hence,

\[
R_n(A_{n,m}) \leq \hat{R}_{n,m}(A_{n,m}) + C_{n,m}(\delta/|A|)
\]

\[
\leq \hat{R}_{n,m}(A_n) + C_{n,m}(\delta/|A|)
\]

\[
\leq R_n(A_n) + 2C_{n,m}(\delta/|A|),
\]

where the second inequality is due to the choice of \(A_{n,m}\).

The result shows that to expect reasonable guarantees one should have \(m \gg n^2 \log(|A|)\). The result can be greatly improved when the variance of the random regret of some fixed algorithm is small compared to \(2n\), the length of the interval that it belongs to by using Bernstein’s inequality (Theorem 3 of [36]) in place of Hoeffding’s.

**Theorem 2.** For any \(\delta \in (0, 1)\) with probability \(1 - \delta\),

\[
R_n(A_{n,m}) \leq R_n(A_n) + C_{n,m}(A_{n,m}, \delta/|A|) + C_{n,m}(A_n, \delta),
\]

where

\[
C_{n,m}(A, \delta) = \sqrt{2\text{Var}(R_n(A)) \log \left( \frac{2}{\delta} \right)} + \frac{2n}{3m} \log \left( \frac{1}{\delta} \right). \tag{4}
\]

The problem with Theorem 2 is that the right-hand side of the bound depends on the variance of the random regret of \(A_{n,m}\), which can be as large as \(n^2\) in some non-favorable cases. To remove this dependence, one may modify the algorithm to minimize the upper bound \(\hat{R}_{n,m}(A) + C_{n,m}(A, \delta/|A|)\). Although we do not experiment with such algorithms in this paper, we discuss their properties and potential advantages below. In particular, it is easy to derive the following result.

**Theorem 3.** Let

\[
A_{n,m} = \arg \min_{A \in A} \hat{R}_{n,m}(A) + C_{n,m}(A, \delta/|A|).
\]

Then for any \(\delta \in (0, 1)\) with probability \(1 - \delta\),

\[
R_n(A_{n,m}) \leq \hat{R}_{n,m}(A_{n,m}) + C_{n,m}(A_{n,m}, \delta/|A|) + C_{n,m}(A_n, \delta/|A|)
\]

\[
\leq R_n(A_n) + 2C_{n,m}(A_n, \delta/|A|).
\]

The same reasoning also gives the stronger inequality

\[
R_n(A_{n,m}) \leq \min_{A \in A} R_n(A) + 2C_{n,m}(A, \delta/|A|). \tag{5}
\]

The last issue is that to compute \(A_{n,m}\), we need to know the variance of the regret for some fixed algorithm \(A\). This can be overcome by replacing Bernstein’s inequality with the so-called empirical Bernstein inequality [38, 36]. The confidence width that comes from this inequality takes the same form as the confidence width that comes from Bernstein’s inequality, except that the constant of the \(\log(1/\delta)\) term in Equation (4) is slightly increased (by a fixed constant factor), while in this width the variance is replaced by its empirical estimate. This leads to the following result.

**Theorem 4.** Let

\[
A_{n,m} = \arg \min_{A \in A} \hat{R}_{n,m}(A) + \hat{C}_{n,m}(A, \delta/|A|)
\]

Then for any \(\delta \in (0, 1)\) with probability \(1 - \delta\),

\[
R_n(A_{n,m}) \leq R_n(A_n) + 2\hat{C}_{n,m}(A_n, \delta/|A|),
\]

where

\[
\hat{C}_{n,m}(A, \delta) = \sqrt{2V_{n,m}(A) \log \left( \frac{2}{\delta} \right)} + \frac{14n}{3m} \log \left( \frac{2}{\delta} \right),
\]

is chosen by Theorem 4 of Maurer and Pontil [36], and

\[
V_{n,m}(A) = \frac{1}{m(m-1)} \sum_{i,j} (R_n(A, P_i) - R_n(A, P_j))^2
\]

is the sample variance.

Finally, with some further calculation (in particular, using Theorem 10 of [36]), one can show that the random confidence width in Theorem 4 can be bounded from above by a constant multiple of \(C_{n,m}(A, \delta)\), which leads to a similar bound as in Theorem 3 with the constant multiplying the confidence width slightly increased.

As usual, the result can also be extended to infinite \(A\), e.g., by using covering numbers, stability, and other tools [35, 6, 7]. We leave the investigation of this for future work. We also note in passing that the idea of optimizing risk/loss upper bounds appears in multiple previous works in various contexts, such as model selection [8], to guide supervised learning [36], or to learn from logged bandit feedback [45]. Of these works, [8] is the only one that provides a clean justification of this; their result most closely resembles (5).

### 3.2 Empirical Tuning Approach

We next present our approach for optimizing the algorithm design in stochastic bandit problems. Specifically,
in light of the results in Section 3.1, we are interested in the optimization problem
\[
\min_{A \in \mathcal{A}} \hat{R}_{n,m}(A). \tag{6}
\]
An important question is what should be the set \( \mathcal{A} \) of algorithms to optimize over. To address this question, we first observe that many existing bandit algorithms are based on upper confidence bounds (UCBs). These terms control the trade-off between exploration and exploitation and are justified theoretically. For example, in UCB1, the term has the form of \( U_t(i) = \hat{\mu}_{t-1,i} + c\sqrt{\log t/T_{t-1,i}} \), where \( \hat{\mu}_{t,i} \) is the average reward of arm \( i \) in the first \( t \) rounds, \( T_{t,i} \) is the number of times that arm \( i \) is pulled in the first \( t \) rounds, \( \sqrt{\log t}/T_{t-1,i} \) represents a confidence interval, and \( c \) is some constant derived from the theoretical analysis. Generally speaking, higher values of \( U_t(i) \) mean that arm \( i \) is more likely to be pulled at round \( t \).

The confidence interval term provides a natural trade-off. In particular, it guarantees that any arm will not be ignored for too long, due to the \( \log t \) term; but also will not be overexploited, due to the \( 1/T_{t-1,i} \) term. We argue that the constant \( c \), which determines the width of the confidence interval, is determined by worst-case analysis, which often ignores the structure in the underlying problem. This conservative choice of \( c \) typically results in using wider confidence intervals, and hence more exploration, than is necessary.

Therefore, we propose to define the family of algorithms \( \mathcal{A} \) for a given bandit algorithm by allowing the width of the confidence interval to change. In particular, each choice of \( c \) defines another algorithm in \( \mathcal{A} \). In our experiments, we focus mostly on tuning UCB-like algorithms in order to streamline the presentation, but our results apply more generally. We provide more details on other algorithms in Section 4.1.4.

4 Empirical Evaluation

Given the wide applicability of UCB-type exploration strategies and their conceptual similarity, it is reasonable to expect that confidence tuning will have a broad impact over a variety of structured and unstructured domains. To demonstrate this, we conduct a comprehensive study across a sequence of problems with increasingly sophisticated structure.

In Section 4.1, we study multi-armed bandit problems. In Section 4.2, we study linear and generalized linear bandit problems. Finally, in Section 4.3, we study Gaussian process bandits. We demonstrate the benefits of empirical optimization in all of those settings.

4.1 Multi-Armed Bandit Problems

We first study our approach on three multi-armed bandit problems: a Bernoulli bandit (Section 4.1.1), a beta bandit (Section 4.1.2), and a Bernoulli bandit with a complex prior distribution (Section 4.1.3). In addition, in Section 4.1.4, we study the potential of similar empirical designs for Thompson sampling and KL-UCB.

4.1.1 Bernoulli Bandit

Our first experiment is conducted on a Bernoulli bandit, where arm \( i \) has reward 1 with probability \( \mu_i \) and reward 0 otherwise. The distribution of problem instances \( \mathcal{P} \) is defined as follows. Each problem is a Bernoulli bandit with \( K = 10 \) arms, where each \( \mu_i \) is drawn uniformly at random from Beta\((1, 1)\). The horizon is \( n = 5000 \) rounds and the performance of all compared algorithms is measured by their Bayes \( n \)-round regret, which is approximated by the average \( n \)-round regret on 1000 i.i.d. problems from \( \mathcal{P} \).

We experiment with tuned UCB1 whose UCB is
\[
U_t(i) = \hat{\mu}_{t-1,i} + c\sqrt{2\log t/T_{t-1,i}},
\]
where \( \hat{\mu}_{t,i} \) is the average reward of arm \( i \) in the first \( t \) rounds, \( T_{t,i} \) is the number of times that arm \( i \) is pulled in the first \( t \) rounds, and \( c \) is tuned. The parameter \( c \) is chosen from \( \{2^0, 2^{-1/3}, 2^{-2/3}, \ldots, 2^{-15/3}\} \) and the algorithm is tuned as follows. We draw \( m = 300 \) i.i.d. problems from \( \mathcal{P} \) and then choose \( c \) with the lowest average regret on these problems. In our plots, we show this regret as a function of \( c \) and refer to it as training.

Tuned UCB1 is compared to three baselines: UCB1 [5], KL-UCB [22], and Bernoulli TS [3]. The prior in TS is Beta\((1, 1)\), which corresponds to knowing \( \mathcal{P} \). Our results are reported in Figure 1a. We observe that the regret of tuned UCB1 is 10 times lower than that of UCB1. Perhaps surprisingly, tuned UCB1 is competitive with TS, which is known to be near-optimal in this setting. One reason is that we consider a weaker notion of regret, the Bayes regret, in our experiments. Nevertheless, these results show the potential of empirical tuning.

4.1.2 Beta Bandit

This experiment is conducted on a multi-armed bandit with \([0, 1]\) rewards that are sampled from a beta distribution. More precisely, the distribution of arm \( i \) is Beta\( (v\mu_i, v(1-\mu_i)) \) where \( v = 4 \). The parameter \( v \) controls the maximum variance of rewards. The higher the value of \( v \), the lower the variance of rewards. The distribution of problem instances \( \mathcal{P} \) is similar to Section 4.1.1,
TS can solve these problems but it is not statistically optimal. In particular, the reward distribution of any arm is a simple conjugate prior to represent the posterior efficiency, we assume that arm 1 is optimal. The distribution P can be sampled from as follows. First, draw uniformly at random \( \mu_1 \in [\Delta, 1] \). Then, for each remaining arm \( i > 1 \), draw uniformly at random \( \mu_i \in [0, \mu_1 - \Delta] \). The evaluation metric and horizon \( n \) are the same as in Section 4.1.1.

Our results are reported in Figure 1b. We observe that the regret of tuned UCB1 is more than 16 times lower than that of UCB1 and 4 times lower than that of TS. This improvement is much higher than in Section 4.1.1 and is due to lower variances of reward distributions in this setting. In particular, the reward distribution of any arm \( i \) in this experiment has a lower variance than that in Section 4.1.1. Since TS is no longer near-optimal, tuned UCB1 can outperform it. Again, note that we consider a weaker notion of regret, the Bayes regret, in our experiments.

Now we examine how tuning at a fixed horizon generalizes beyond it. We choose the best value of \( c \) from the last experiment, which was obtained by tuning at \( n = 5000 \) rounds, and plot the regret of UCB1, KL-UCB, TS, and tuned UCB1 up to \( n = 20000 \) rounds (Figure 2). We observe that the regret of tuned UCB1 is consistently the lowest over the whole horizon. This indicates that tuning at a fixed horizon can generalize beyond it.

### 4.1.3 Complex Prior

In all experiments so far, we only considered a simple prior distribution \( P \). In this experiment, we consider a prior distribution does not have a simple conjugate form. In this case, Bernoulli TS with a beta prior is no longer statistically optimal.

The distribution of problem instances \( P \) is a distribution over Bernoulli bandits where the expected reward of the optimal arm is by at least \( \Delta = 0.1 \) higher than that of the best suboptimal arm. Without loss of generality, we assume that arm 1 is optimal. The distribution \( P \) can be sampled from as follows. First, draw uniformly at random \( \mu_1 \in [\Delta, 1] \). Then, for each remaining arm \( i > 1 \), draw uniformly at random \( \mu_i \in [0, \mu_1 - \Delta] \). The evaluation metric and horizon \( n \) are the same as in Section 4.1.1.

Tuned UCB1 is implemented as in Section 4.1.1 and we compare it to the same baselines as there. The prior in TS is Beta(1, 1), which corresponds to knowing \( P \). TS is implemented with [0, 1] rewards as suggested by Agrawal and Goyal [3]. For any reward \( Y_{t,i} \in [0, 1] \), we draw pseudo-reward \( \hat{Y}_{t,i} \sim Ber(Y_{t,i}) \) and then use it in TS instead of \( Y_{t,i} \).

Our results are reported in Figure 1c. We observe that the regret of tuned UCB1 is 10 times lower than that of UCB1. Since the prior in TS is not correctly specified, tuned UCB1 can outperform it. The regret of tuned UCB1
is about 2 times lower than that of TS.

4.1.4 Tuning Other Algorithms

As mentioned in Section 3.2, it is possible to consider a similar tuning approach for other algorithms than UCB1. In this section, we study analogous formulations for KL-UCB and TS.

In Bernoulli KL-UCB [22], the UCB of arm $i$ in round $t$ is computed as

$$\max \left\{ q : d \left( \frac{S_{t-1,i}}{T_{t-1,i}}, q \right) \leq c \frac{\log t + 3 \log \log t}{T_{t-1,i}} \right\},$$

where $d(\cdot, \cdot)$ is the Kullback-Leibler divergence, $S_{t,i}$ is the cumulative reward of arm $i$ in the first $t$ rounds, and $c = 1$. As in UCB1, we define the set of algorithms $\mathcal{A}$ by varying the scaling factor $c$.

In Bernoulli TS [3], the posterior distribution is a function of pseudo-counts $(\alpha_{t,i}, \beta_{t,i})$ for each arm $i$, where $\alpha_{t,i} = S_{t,i} + 1$ and $\beta_{t,i} = T_{t,i} - S_{t,i} + 1$. In round $t$, the expected reward of arm $i$ is sampled from $\text{Beta}(\alpha_{t-1,i}, \beta_{t-1,i})$, which is then used to choose an arm to pull. To control the posterior, we propose sampling from $\text{Beta}(\alpha_{t-1,i}/c, \beta_{t-1,i}/c)$ instead, which makes the posterior wider for $c > 1$ and more concentrated for $c < 1$.

Figure 3 is similar to Figure 1, but additionally shows the results of tuning KL-UCB and TS. We observe that tuning can significantly improve performance of all algorithms.

4.2 Structured Problems

Existing algorithm designs and their analyses become increasingly looser as the structure of the domain becomes more sophisticated. We show this in two classes of structured problems: linear bands (Section 4.2.1) and generalized linear bands (Section 4.2.2).

4.2.1 Linear Bandit

In linear bandits, the rewards of arms have an additional structure. In particular, the reward of arm $i$ in round $t$ is $Y_{t,i} = x_i^T \theta^* + \epsilon_{t,i}$, where $x_i \in \mathbb{R}^d$ is a known feature vector of arm $i$, $\theta^* \in \mathbb{R}^d$ is an unknown parameter vector that is shared by the arms, and $\epsilon_{t,i}$ is i.i.d. $\sigma$-sub-Gaussian noise. We set $\lambda = 0.5$.

The distribution of problem instances $P$ is defined as follows. Each problem is a linear bandit with $K = 100$ arms and $d = 5$. Both $\theta^*$ and $x_i$ are drawn uniformly at random from $[-1, 1]^d$. The horizon is $n = 10000$ rounds and the performance of all compared algorithms is measured by their Bayes $n$-round regret, which is approximated by the average $n$-round regret on 500 i.i.d. problems from $P$.

We experiment with tuning LinUCB [1]. LinUCB estimates $\theta^*$ by ridge regression with regularization parameter $\lambda$. We set $\lambda = 1$. Let $\hat{\theta}_t$ be the solution to the ridge regression problem in round $t$, on data in the first $t − 1$ rounds, and $G_t$ be the corresponding sample covariance matrix. For $\delta > 0$, let

$$g(t) = \sigma \sqrt{d \log \left( \frac{1 + tL^2/\lambda}{\delta} \right) + \lambda^2 S}$$

be a slowly growing function of $t$, where $\|\theta^*\|_2 \leq S$ and $\|x_i\|_2 \leq L$. Since $\theta^* \in [-1, 1]^d$, we set $S = \sqrt{d}$. For these settings, the UCB of arm $i$ in round $t$ is

$$U(t)_i = x_i^T \hat{\theta}_t + cg(t) \sqrt{x_i^T G_t^{-1} x_i},$$

where $c$ is a tunable parameter for confidence width. We choose it from $\{2^{1/2}, 2^{3/2}, \ldots, 2^{-5/2}, 2^{-6/2}\}$ and the
algorithm is tuned as follows. We draw \( m = 200 \) i.i.d. problems from \( \mathcal{P} \) and then choose \( c \) with the lowest average regret on these problems. In our plots, we show this regret as a function of \( c \) and refer to it as training.

We compare tuned LinUCB to untuned LinUCB (\( c = 1 \)) and LinTS [4]. Both algorithms perform well in practice, but are known to be statistically suboptimal. The reason is that their gap-dependent and gap-free bounds contain extra factors of \( d \) and \( \sqrt{d} \), respectively. We compare to the best possible implementations of these algorithms. Specifically, in LinTS, the posterior distribution is \( N(\hat{\theta}_t, G_t^{-1}) \), where \( G_t^{-1} \) is smaller than suggested by the theory and leads to better performance in practice.

Our results are reported in Figure 4a. We observe that the regret of tuned LinUCB is almost 4 times lower than that of LinUCB. Tuned LinUCB also outperforms LinTS.

Now we examine how tuning at a fixed horizon generalizes beyond it. We choose the best value of \( c \) from the last experiment, which was obtained by tuning at \( n = 10000 \) rounds, and plot the regret of tuned LinUCB, untuned LinUCB (\( c = 1 \)), and LinTS up to \( n = 40000 \) rounds (Figure 4b). We observe that the regret of tuned LinUCB is consistently the lowest, which indicates that tuning at a fixed horizon can generalize beyond it.

### 4.2.2 Generalized Linear Bandit

Generalized linear bandits are another class of problems with structured rewards. We focus on logistic bandits where the rewards are binary. In particular, the reward of arm \( i \) in round \( t \) is \( Y_{t,i} = \mu(x_i^T \theta^*) + \epsilon_{t,i} \), where \( x_i \in \mathbb{R}^d \) is a known feature vector of arm \( i \), \( \theta^* \in \mathbb{R}^d \) is an unknown parameter vector that is shared by the arms, \( \mu(v) = 1/(1 + \exp(-v)) \) is a sigmoid function, and \( \epsilon_{t,i} \) is i.i.d. \( \sigma \)-sub-Gaussian noise.

The distribution of problem instances \( \mathcal{P} \) is defined as follows. Each problem is a logistic bandit with \( K = 200 \) arms and \( d = 5 \). Both \( \theta^* \) and \( x_i \) are drawn uniformly at random from \([-1, 1]^d\). The evaluation metric and horizon \( n \) are the same as in Section 4.2.1.

We experiment with tuning UCB-GLM [31]. Let \( \hat{\theta}_t \) be the maximum-likelihood estimate of \( \theta^* \) in round \( t \), on data in the first \( t - 1 \) rounds, in UCB-GLM and \( G_t \) be the corresponding sample covariance matrix, as in Section 4.2.1. For \( \delta > 0 \), let

\[
g(t) = \frac{\sigma}{\kappa} \sqrt{\frac{d}{2} \log \left( 1 + \frac{2t}{d} \right) + \log(1 + \delta)}
\]

be a slowly growing function of \( t \), where \( \kappa \) is the minimum derivative of the mean function \( \mu \), as defined in [31]. We set \( \kappa = 0.25 \), which is the maximum derivative of \( \mu \), and thus the most optimistic setting. Since \( Y_{t,i} \) is binary, we set \( \sigma = 0.5 \). For these settings, the UCB of arm \( i \) in round \( t \) is

\[
U_{t,i} = x_i^T \hat{\theta}_t + cg(t) \sqrt{x_i^T G_t^{-1} x_i},
\]

where \( c \) is a tunable parameter for confidence width. We tune \( c \) exactly as in Section 4.2.1. We find \( \hat{\theta}_t \) using iteratively reweighted least squares for Bayesian logistic regression [12]. We do not explicitly implement the initialization stage of UCB-GLM. Instead, we ensure that \( G_t \) is invertible by regularization.

We compare tuned UCB-GLM to untuned UCB-GLM (\( c = 1 \)) and GLM-TS [2]. Again, we compare to the best possible implementation of these algorithms. The most natural way of implementing GLM-TS is using the Laplace approximation, where the posterior distribution is approximated by a normal distribution with mean \( \hat{\theta}_t \). This approximation works well in practice though it does not come with regret guarantees.

Our results are reported in Figure 4c. We observe that UCB-GLM can be tuned to outperform both UCB-GLM and GLM-TS. Specifically, the regret of tuned UCB-GLM is
Figure 5: Evaluation of tuned GP-UCB in Section 4.3. The empirical Bayes regret over both tuned parameters in the synthetic problem is shown in plot a. In plot b, we show it at the best kernel width as a function of \( c \), together with the Bayes regret of tuned and untuned GP-UCB. Plots c and d are analogous plots to a and b on the real-world problem.

We run \( \text{GP-UCB} \) 0 width tune both its confidence interval scaling \( t \) rescaled such that its minimum and maximum are drawn is the scalability of Gaussian process regression. We conduct two experiments. The first experiment is\( \text{GP} \)-dimensional feature of arm \( i \). Then

\[
\mu_i \propto \sum_{k=1}^{10} w_k \exp \left[ -\left( x_i - y_k \right)^2 / (2\sigma_0^2) \right],
\]

where \( (w_k)_{k=1}^{10} \) and \( (y_k)_{k=1}^{10} \) are chosen uniformly at random from \([-1, 1]\), and \( \sigma_0 = 0.25 \). The mean function is rescaled such that its minimum and maximum are 0.1 and 0.9, respectively. Finally, the reward of arm \( i \) in round \( t \) is drawn i.i.d. from \( \text{Ber} (\mu_i) \).

We run GP-UCB with an RBF kernel of width \( \sigma \), and tune both its confidence interval scaling \( c \) and kernel width \( \sigma \). Both parameters are selected from a range of \( \{ 2^0, 2^{-1/3}, \ldots, 2^{-15/3} \} \). GP-UCB is run for \( n = 200 \) rounds. The main limiting factor in the number of rounds is the scalability of Gaussian process regression. We draw \( m = 200 \) i.i.d. problems from \( \mathcal{P} \), and then choose \( c \) and \( \sigma \) with the lowest average regret on these problems. We report this regret in Figure 5a. The Bayes regret is approximated by the average \( n \)-round regret on another 200 i.i.d. problems from \( \mathcal{P} \). We compare the Bayes regret of GP-UCB to that of untuned GP-UCB \( (c = 1, \sigma = 0.25) \) in Figure 5b. We observe an order of magnitude improvement.

The second experiment is conducted on data from a commercial recommender system. The data are collected from 100 days of A/B tests, each of which is associated with an 11-dimensional vector of policy parameters. On each day, new users are exposed to the policy and an aggregate user engagement metric is taken as the reward. In total, 148 policies are tested, for 100 days each. We draw problems from the distribution of problem instances \( \mathcal{P} \) as follows. For each policy, we randomly choose 15 days between days 1 and 70. The arm is a policy. When the arm is pulled, it randomly chooses one of the previously chosen 15 days and returns its reward.

We tune the parameters of GP-UCB as in the synthetic experiment, with the difference that the kernel width varies in a wider range of \( \{ 2^{-2}, \ldots, 2^3 \} \). The Bayes regret of the best performing parameter pair is measured on environments that are created from the remaining 30 days, which are not used for tuning. This would be a realistic scenario in practice. Similarly to the synthetic experiment, Figures 5c and 5d show significant gains from tuning GP-UCB.

5 Related work

Empirical studies of bandit algorithms can be seen as a precursor to our work. In such studies, the authors typically compare multiple bandit algorithms, which may or may not have “solid theoretical foundations”. Often, the comparisons are done on ad-hoc bandit instances (often, randomly generated), and conclusions are drawn based on the “overall picture” across the instances cho-
sen [50, 49, 28, 41, 18]. While informative, these studies appear not to test specific design objectives, such as having an algorithm that is best in the worst-case, in the limit, or on a given horizon, or on an instance-per-instance bandit, or on average. In contrast, we turn things around by being clear about the objective (finding an algorithm that is good on average across randomly chosen instances), and also by proposing to study the learning of such a good bandit algorithm based on testing algorithms on instances drawn from a fixed, possibly unknown distribution. Of course, this connects our work both to meta-learning—or learning-to-learn [47, 48, 43], and also to the Bayesian bandit setting.

From a meta-learning perspective, our setting is closest to that of Baxter [9], who formulates the learning to learn problem by assuming that data from multiple tasks are available to evaluate learners, where the tasks are sampled, independently of each other from a common unknown distribution, and the goal is to find a learner with a good performance on future tasks, sampled from the same common unknown distribution [9, 10].

Recent years have seen a surge of interest in meta-learning in the reinforcement learning context (e.g., Finn et al. [19, 20], Mishra et al. [37]), where the emphasis is on improving the data efficiency of deep learning combined with reinforcement learning. In comparison, the experimental part of our work is concerned with parameter tuning of well-established bandit algorithms, which is arguably simpler to implement, we provide (a simple) theoretical analysis, while the results are easier to interpret by design. At the same time, our approach is clearly more limited, which may limit the performance that can be achieved, though since we start with theoretically motivated algorithms and our theory predicts that our method will not do worse as these, we also inherit the guarantees available for the theoretical choice of the tuning parameter. The closest to our work is that of Maes et al. [34] who consider the same problem as us with the same motivation. They provide no theoretical analysis and their empirical evaluation is limited to 2-armed Bernoulli bandits, which leads to quite small gains compared to the baselines, while our experiments illustrate better the significance gains that one can expect from the meta-learning approach.

The Bayesian bandit literature is enormous [11, 39, 25]. Computing the Bayes optimal policy is computationally challenging. The Gittins index policy [24], which optimizes the infinite horizon discounted total expected reward, is considered as a good approximation to the undiscounted problem for large discount factors and horizons, and there is a large literature on efficient computation of the Gittins index, surveyed by Chakravorty and Mahajan [15]. However, these methods are limited to the \( k \)-armed bandit case (ruling out, e.g., linear bandits), they are nontrivial to implement and they often make specific assumptions about the bandit model. To be applicable to our setting, one would need a result for the case when the prior is a uniform mixture of a finite number of bandit models and we do not know of such results. Our work is also related to “empirical Bayes” procedures, which estimate the prior based on data [14] because we also take data in place of a prior. However, as opposed to empirical Bayes procedures, we aim at directly finding a good learning algorithm.

A sequential multitask setting [13] has been considered by Gheshlaghi Azar et al. [23]. In their problem, tasks, sampled independently of each other from an unknown finitely-supported probability distribution, arrive sequentially. The learner interacts with each task that arrives for \( n \) rounds, with the goal to maximize the total regret. As opposed to this problem, in our case the goal is to generalize to a new task, whereas the main challenge in the multitask setting is for the learner to learn the characteristics of each of the finitely many tasks that it sees and then when facing with a task, quickly identifying the task. Deshmukh et al. [17] considers multi-task learning in the contextual setting.

Finally, our problem is related to both hyperparameter optimization [21] and algorithm auto-configuration [7, 6].

6 Conclusions

In this work, we propose an approach called empirical Bayes regret minimization, which utilizes sampled problem instances in order to minimize the Bayes regret. Our approach is related to similar ideas from meta-learning, or learning-to-learn. We justify our methodology theoretically and evaluate it empirically on various bandit problems. In particular, we show how we can tune bandit algorithms using empirical Bayes regret minimization in a variety of settings and achieve much better performance than the vanilla untuned algorithms.

This work opens multiple avenues for future research. First, the suggested algorithm in Theorem 4 can be used to optimize an upper bound of the Bayes regret and obtain tighter bounds. Second, we used a simple enumeration strategy over a finite set of algorithms to optimize. It is interesting to explore more advanced optimization techniques. These techniques may leverage, for instance, the smoothness of regret with respect to tunable parameters, as suggested by our empirical results. Finally, we also plan to apply our proposed framework to real-world contextual bandit problems.
References

[1] Yasin Abbasi-Yadkori, David Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In Advances in Neural Information Processing Systems 24, pages 2312–2320, 2011.

[2] Marc Abeille and Alessandro Lazaric. Linear Thompson sampling revisited. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, pages 176–184, 2017.

[3] Shipra Agrawal and Navin Goyal. Further optimal regret bounds for Thompson sampling. In Proceedings of the 16th International Conference on Artificial Intelligence and Statistics, pages 99–107, 2013.

[4] Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In Proceedings of the 30th International Conference on Machine Learning, pages 127–135, 2013.

[5] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. Machine Learning, 47:235–256, 2002.

[6] Maria-Florina Balcan, Travis Dick, and Tuomas Sandholm. Learning to branch. In International Conference on Machine Learning, pages 353–362, 2018.

[7] Maria-Florina Balcan, Travis Dick, and Ellen Vitercik. Dispersion for data-driven algorithm design, online learning, and private optimization. In FOCS, pages 603–614, 2018.

[8] Peter L. Bartlett, Stephane Boucheron, and Gabor Lugosi. Model selection and error estimation. Machine Learning, 48(1):85–113, Jul 2002.

[9] Jonathan Baxter. Theoretical models of learning to learn. In Learning to Learn. Springer, 1998.

[10] Jonathan Baxter. A model of inductive bias learning. Journal of artificial intelligence research, 12: 149–198, 2000.

[11] D. Berry and B. Fristedt. Bandit problems: sequential allocation of experiments. Chapman and Hall, London ; New York ;, 1985.

[12] Christopher M. Bishop. Pattern recognition and machine learning, 5th Edition. Springer, 2007.

[13] Rich Caruana. Multitask learning. Machine learning, 28(1):41–75, 1997.

[14] George Casella. An introduction to empirical Bayes data analysis. American Statistician, 39:83–=87, 05 1985.

[15] J. Chakravorty and A. Mahajan. Multi-armed bandits, Gittins index, and its calculation. Methods and Applications of Statistics in Clinical Trials: Planning, Analysis, and Inferential Methods, Volume 2, pages 416–435, 2014.

[16] Varsha Dani, Thomas Hayes, and Sham Kakade. Stochastic linear optimization under bandit feedback. In Proceedings of the 21st Annual Conference on Learning Theory, pages 355–366, 2008.

[17] Aniket Anand Deshmukh, Urn Dogan, and Clay Scott. Multi-task learning for contextual bandits. In NeurIPS, pages 4848–4856, 2017.

[18] R. Devanand and P. Kumar. Empirical study of Thompson sampling: Tuning the posterior parameters. AIP Conference Proceedings, 1853(1): 050001, 2017.

[19] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the 34th International Conference on Machine Learning, pages 1126–1135, 2017.

[20] Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. In NeurIPS, pages 9537–9548, 2018.

[21] Luca Franceschi, Paolo Frasconi, Michele Donini, and Massimiliano Pontil. A bridge between hyperparameter optimization and learning-to-learn. In Workshop on Meta-Learning, 2018.

[22] Aurelien Garivier and Olivier Cappe. The KL-UCB algorithm for bounded stochastic bandits and beyond. In Proceeding of the 24th Annual Conference on Learning Theory, pages 359–376, 2011.

[23] Mohammad Gheorghe Azar, Alessandro Lazaric, and Emma Brunskill. Sequential transfer in multi-armed bandit with finite set of models. In Advances in Neural Information Processing Systems 26, pages 2220–2228, 2013.

[24] J. Gittins. Bandit processes and dynamic allocation indices. Journal of the Royal Statistical Society. Series B (Methodological), 41(2):148–177, 1979.

[25] J. Gittins, K. Glazebrook, and R. Weber. Multi-armed bandit allocation indices. John Wiley & Sons, 2011.

[26] Aditya Gopalan, Shie Mannor, and Yishay Mansour. Thompson sampling for complex online problems. In Proceedings of the 31st International Conference on Machine Learning, pages 100–108, 2014.

[27] Jaya Kawale, Hung Bui, Branislav Kveton, Long Tran-Thanl, and Sanjay Chawla. Efficient Thomp-
son sampling for online matrix-factorization recommendation. In Advances in Neural Information Processing Systems 28, pages 1297–1305, 2015.

[28] Volodymyr Kuleshov and Doina Precup. Algorithms for multi-armed bandit problems. arXiv preprint arXiv:1402.6028, 2014.

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

[30] Tor Lattimore and Csaba Szepesvári. Bandit Algorithms. Cambridge University Press (preprint), 2019.

[31] Lihong Li, Yu Lu, and Dengyong Zhou. Provably optimal algorithms for generalized linear contextual bandits. In Proceedings of the 34th International Conference on Machine Learning, pages 2071–2080, 2017.

[32] Zachary Lipton, Xiujun Li, Jianfeng Gao, Lihong Li, Faisal Ahmed, and Li Deng. Bbq-networks: Efficient exploration in deep reinforcement learning for task-oriented dialogue systems. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, pages 5237–5244, 2018.

[33] Xiuyuan Lu and Benjamin Van Roy. Ensemble sampling. In Advances in Neural Information Processing Systems 30, pages 3258–3266, 2017.

[34] Francis Maes, Louis Wehenkel, and Damien Ernst. Meta-learning of exploration/exploitation strategies: The multi-armed bandit case. In International Conference on Agents and Artificial Intelligence, pages 100–115. Springer, 2012.

[35] Andreas Maurer. Algorithmic stability and meta-learning. Journal of Machine Learning Research, 6:967–994, 2005.

[36] Andreas Maurer and Massimiliano Pontil. Empirical Bernstein bounds and sample variance penalization. In COLT, 2009.

[37] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In ICLR, 2018. URL https://openreview.net/forum?id=B1DmuUzWAW.

[38] V. Mnih, Cs. Szepesvári, and J.-Y. Audibert. Empirical Bernstein stopping. In Proceedings of the 25th International Conference on Machine Learning, ICML, pages 672–679, New York, NY, USA, 2008. ACM.

[39] E. L. Presman and I. N. Sonin. Sequential control with incomplete information. The Bayesian approach to multi-armed bandit problems. Academic Press, 1990.

[40] Carlos Riquelme, George Tucker, and Jasper Snoek. Deep Bayesian bandits showdown: An empirical comparison of Bayesian deep networks for Thompson sampling. In Proceedings of the 6th International Conference on Learning Representations, 2018.

[41] Nixon K. Ronoh, Reuben Odoyo, Edna Milgo, Madalina M. Drugan, and Bernard Mandereck. Bernoulli bandits: an empirical comparison. In ESANN, 2015.

[42] Daniel Russo and Benjamin Van Roy. Learning to optimize via posterior sampling. Mathematics of Operations Research, 39(4):1221–1243, 2014.

[43] Daniel L. Silver, Qiang Yang, and Lianghao Li. Lifelong machine learning systems: Beyond learning algorithms. In 2013 AAAI spring symposium series, 2013.

[44] Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting and experimental design. In Proceedings of the 27th International Conference on International Conference on Machine Learning, pages 1015–1022, 2010.

[45] Adith Swaminathan and Thorsten Joachims. Counterfactual risk minimization: Learning from logged bandit feedback. In ICML, pages 814–823, 2015.

[46] 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.

[47] Sebastian Thrun. Explanation-based neural network learning: A lifelong learning approach. Kluwer, 1996.

[48] Sebastian Thrun. Lifelong learning algorithms. In Learning to learn, pages 181–209. Springer, 1998.

[49] Michel Tokic and Günther Palm. Value-difference based exploration: Adaptive control between epsilon-greedy and softmax. In Ki 2011: Advances in Artificial Intelligence, 34th Annual German Conference on AI, pages 335–346, 2011.

[50] Joannes Vermorel and Mehryar Mohri. Multi-armed bandit algorithms and empirical evaluation. In ECML, pages 437–448. Springer, 2005.