Driving Exploration by Maximum Distribution in Gaussian Process Bandits

Alessandro Nuara
Politecnico di Milano
Milan, Italy
alessandro.nuara@polimi.it

Francesco Trovò
Politecnico di Milano
Milan, Italy
francesco1.trovo@polimi.it

Dominic Crippa
Politecnico di Milano
Milan, Italy
dominic.crippa@mail.polimi.it

Nicola Gatti
Politecnico di Milano
Milan, Italy
nicola.gatti@polimi.it

Marcello Restelli
Politecnico di Milano
Milan, Italy
marcello.restelli@polimi.it

ABSTRACT

The problem of finding optimal solutions of stochastic functions over continuous domains is common in several real-world applications, such as, *e.g.*, advertisement allocation, dynamic pricing, and power control in wireless networks. The optimization process is customarily performed by selecting input points sequentially and receiving a noisy observation from the function. In this paper, we resort to the Multi-Armed Bandit approach, aiming at optimizing stochastic functions when keeping at a pace the *regret* (*i.e.*, the loss incurred during the learning process) during the learning process. In particular, we focus on *smooth* stochastic functions, as it is known that any algorithm suffers from a constant per-round regret when the domain is continuous, and the function does not satisfy any kind of regularity. Our main original contribution is the provision of a general family of algorithms, which, under the mild assumption that stochastic functions are a realization of a Gaussian Process, provides a regret of the order of $O(\sqrt{\gamma_T T})$, being γ_T the maximum information gain and T the time horizon used for the learning process. Furthermore, we design a specific algorithm of our family, called DAGP-UCB, which exploits the structure of GPs to select the next arm to pull more effectively than the previous algorithms available in the state of the art, thus speeding up the learning process. In particular, we show the superior performance of DAGP-UCB in both synthetic and applicative settings, comparing it with the state-of-the-art algorithms.

KEYWORDS

Gaussian Process; Multi-armed Bandit; Online Learning

ACM Reference Format:
Alessandro Nuara, Francesco Trovò, Dominic Crippa, Nicola Gatti, and Marcello Restelli. 2020. Driving Exploration by Maximum Distribution in Gaussian Process Bandits. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), Auckland, New Zealand, May 2020,* IFAAMAS, 9 pages.

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), , May 2020, Auckland, New Zealand. © 2020 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

1 INTRODUCTION

The problem of finding optimal solutions of stochastic functions over continuous domains is common in several real-world applications, such as, *e.g.*, advertisement allocation [21], dynamic pricing [26], power control in wireless communications [8] and learning optimal control strategies in RL settings [18]. The optimization process is customarily performed by selecting input points sequentially and receiving a noisy observation from the function. Two different approaches are customarily used to determine the sequence of the points to test: Bayesian optimization [19], aiming at optimizing a given function globally using the smallest possible number of samples, and Multi-Armed Bandit (MAB) approaches [2], aiming at learning the optimal solution while minimizing the loss incurred during the sampling process. In particular, Bayesian optimization is useful when the learner incurs in a loss proportional to the number of samples drawn in the learning process. For instance, this is the case when one aims at optimizing the hyperparameters of complex models and algorithms in machine learning, robotics, and computer vision [6, 16, 23, 27], where the main issue is the computational time required to sample the input data points. Conversely, in many real-world scenarios, selecting a suboptimal option during the optimization process causes the learner to lose value. For instance, in the ad allocation problem, in which the learner has to partition a given daily budget for advertising over a set of ads to maximize the number of clicks received by users, whenever the learner selects a suboptimal budget allocation, she causes a potential loss of profit for current day the company that advertises the product. Indeed, differently from the past settings, the loss incurred at each round is proportional to the difference between the maximum number of clicks we can obtain and the number of clicks provided by the allocation chosen for the round. In our work, we propose a novel algorithm for stochastic optimization following the MAB approach, which exploits the structure of models based on Gaussian Processes (GPs) [22].

The effectiveness of the MAB algorithms is usually measured by comparing their theoretical guarantees on their *cumulative regret* (*i.e.*, the expected cumulative loss incurred during the execution of the algorithm) and convergence to the optimal solution. This is done by balancing *exploration*, *i.e.*, the process of gathering information on potentially unexplored options, and *exploitation*, *i.e.*, the use of available information to choose the best option. The asymptotic convergence to the optimal solution is guaranteed when the algorithm provides a regret that is sublinear w.r.t. the *a priori* chosen number of rounds T. When the domain is continuous, and the stochastic function does not satisfy any kind of regularity, every algorithm suffers from a constant per-round regret [24]. For this reason, it is common to design MAB algorithms working on the assumption that some kind of regularity on either the sampled observations or, more often, the function itself is satisfied. For instance, if the function is assumed to be Lipschitz continuous, we can use Continuous-Armed Bandit (CAB) techniques: Kleinberg et al. ([2008]) propose techniques suffering from a regret of order $O(T^{\frac{d+1}{d+2}})$, where d is the so-called Zooming dimension, *i.e.*, a parameter that identifies the complexity of the specific setting. Nonetheless, the application of the CAB algorithms requires the knowledge of the Lipschitz constant of the function, a value that is often difficult to obtain in real-life problems.

In recent years, a milder assumption on the function has been used to design algorithms in this field: the function should be a realization of a GP, a model that is the generalization to infinite dimensions of the multivariate Gaussian distribution. Several algorithms have been specifically crafted to minimize the cumulative regret in the so-called GPMAB setting [9, 24]. Most of them, *e.g.*, GPUCB and IGPUCB, given the observations, build high-probability Upper Confidence Bounds (UCBs) over the function values and use them to drive the learning process. Another approach adopted in this setting, namely GP-TS [9], relies on sampled values from the estimated distribution to drive the optimization process. Although these algorithms have been shown to have good theoretical properties, *i.e.*, a cumulative regret of order $O(\sqrt{T\gamma_T})$, being γ_T the maximum information gain of T samples drawn from the original function, and empirical performance, they do not fully exploit the potential offered by the structure of the GPs. More specifically, the procedure used to select the next arm to play does not explicitly use the fact that the choice of an input point (a.k.a. arm in the MAB field) provides information over the entire function. In principle, one could select an arm not only because it is likely to be optimal, but also to obtain information on those regions that are most likely to be optimal. This kind of exploration strategy may be of paramount importance in the early stage of the learning process and can be addressed by explicitly incorporating the uncertainty reduction that one would have as a result of pulling an arm. The novel contributions of this paper are the following:

- the definition of a family of UCB-like algorithms, which also includes some of the currently available techniques in the literature, and a regret bound for this family of $O(\sqrt{T\gamma_T})$ in the GP setting, extending and corroborating theoretical results already present in the literature;
- the design of the Distribution-Aware GP-UCB (DAGP-UCB) algorithm, capable of leading the exploration using the uncertainty reduction provided to the most promising regions (having a large probability of being optimal) of the function;
- a wide experimental campaign on synthetically generated data and an application to advertising allocation optimization comparing the empirical performance of the proposed algorithm with the state-of-the-art ones.

2 RELATED WORKS

A first line of research related to our work is represented by algorithms specifically created to solve the GPMAB problem. Srinivas et al. [24] establish, for the first time in the literature, a new connection between experimental design and GP optimization. They introduce the GP-UCB algorithm, which uses statistical upper confidence bounds to drive the sequential selection of points, whose cumulative regret bound is expressed in terms of the maximum information gain γ_T. More precisely, the authors provide high confidence bounds of the order $O(\sqrt{\gamma_T T})$, meaning that, for most common kernel classes, the algorithm provides sublinear regret guarantees. Chowdhury and Gopalan [9] propose two algorithms: IGP-UCB and GP-TS. The former one is upper-confidence bound based and can be seen as a variant of GP-UCB using tighter confidence intervals. This algorithm achieves a high confidence bound of $O\left(B\sqrt{\gamma_T T} + \sqrt{\gamma_T^2 T}\right)$ on the cumulative regret, but it requires the *a priori* knowledge of an upper bound on the maximum information gain at round t and an upper bound of the norm of the target function B. A completely different approach is provided by GP-TS, which is an algorithm inspired by Thompson Sampling [14] modified to work in the GPMAB setting. At each round, the GP-TS algorithm samples a new GP from a specific probability distribution and plays the arm that is optimal for that sample. It is known that this algorithm suffers from a high confidence cumulative regret of $O\left(\sqrt{\gamma_T d \log(BdT)}\left[\sqrt{\gamma_T T} + B\sqrt{T}\right]\right)$, where d is the dimension of the input space. The EST [28] and MES [27] algorithms are based on the entropy search approach and provide an upper bound to the regret of $O(\sqrt{\gamma_T T})$. Both algorithms require the knowledge of the value of the optimal solution (or an upper bound) to run the algorithm, which, usually, is a strong assumption for real-world scenarios. Other heuristic approaches to balance exploration and exploitation in optimization have also been applied to the GP setting, for instance, EI [19] and PI [17]. They are based on the idea to choose at every round the arm that maximizes the expected improvements and the probability of improvement, respectively, w.r.t. a given threshold. To the best of our knowledge, no upper bound on the cumulative regret for these algorithms is known.

Another line of research focuses on the analysis of the algorithm in terms of *simple regret* [5, 13, 27], defined as the minimum distance between the value of the function of the points selected so far and the optimal solution of the function to be optimized. This concept of regret is weaker than the one used in the MAB setting. Indeed, we will show that, in our setting, an algorithm with sublinear cumulative regret, also has sublinear simple regret, but not *vice versa*.[1] Therefore, the theoretical results provided in these works cannot be compared with ours.

We also mention the works dealing with MAB settings with a continuous number of arms [1, 4, 15], in which the authors require some degree of continuity, *e.g.*, Lipschitz, of the reward function. The smoothness property of functions drawn from a GPs is different from the one required in these works (see [24] for details). Moreover, it has been shown that empirically these techniques perform poorly in terms of average cumulative regret [25].

[1] The relationship between the two in a more general setting having rewards with finite support is studied in [3].

Finally, one of the most promising applications of the GPMAB techniques in real-world setting is the optimization of advertising campaigns [20, 21]. Indeed, they have been used to approximate the functions providing the number of impressions or clicks given the advertiser's bid and the daily budget allocated to the specific add. GPs have been both used to optimize the bid/budget allocation over a set of ads in an online fashion [21], as well as to optimize the ads campaign offline [20]. In the aforementioned works, the framework used to model the problem is a generalization of the MAB one, namely the Combinatorial MAB [7]. Nonetheless, the improvement of algorithms to explore efficiently the space of the possible input, *i.e.*, minimizing the cumulative regret, is of paramount importance in this applicative setting.

3 PROBLEM STATEMENT

We focus on the problem of deciding sequentially which input $\mathbf{x}$, in a domain $\mathcal{D} \in \mathbb{R}^d$, to sample from an unknown function $f : \mathcal{D} \to \mathbb{R}$ to find the solution that maximizes the function while minimizing the loss incurred in the learning process. More precisely, at each round t over a finite time horizon $T \in \mathbb{N}$, we choose a point $\mathbf{x}_t \in \mathcal{D}$, a.k.a. arm, and observe a perturbed sample from that function $y_t = f(\mathbf{x}_t) + \epsilon_t$, where $\epsilon_t \sim \mathcal{N}(0, \lambda)$ is a zero-mean Gaussian noise with variance λ.

In this scenario, an algorithm or policy $\mathfrak{U}$ prescribes the arm $\mathbf{x}_t$ to be selected in a specific round t. The performance of a policy $\mathfrak{U}$ is evaluated in terms of the cumulative expected reward that it is capable of gaining over the finite time horizon T or, equivalently, in terms of its *cumulative expected pseudo-regret*, defined as follows:

$$R_T(\mathfrak{U}) = \sum_{t=1}^{T} \mathbb{E}\left[f(\mathbf{x}^*) - f(\mathbf{x}_t)\right], \tag{1}$$

where $\mathbf{x}^* := \arg\max_{\mathbf{x}\in\mathcal{D}} f(\mathbf{x})$ is the choice providing the largest value of the function f, and the expectation in the formula is w.r.t. the possible stochasticity of the policy.

Gaussian Process. As aforementioned, if function f does not show any kind of regularity, the problem of minimizing the regret does not admit any algorithm with sublinear cumulative regret. In our work, we require that the function has some smoothness properties assuming that is a sample from a GP [22], which is a set of random variables, one for each $\mathbf{x} = (x_1, \dots, x_d)$, $\mathbf{x} \in \mathcal{D}$, every finite subset of which is a multivariate Gaussian distributed. More specifically, a Gaussian Process $GP(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x})')$ is completely defined by its mean function $\mu(\mathbf{x}) = \mathbf{E}[f(\mathbf{x})]$ and covariance (or kernel) function $k(\mathbf{x}, \mathbf{x}') = \mathbf{E}[(f(\mathbf{x}) - \mu(x))(f(\mathbf{x}') - \mu(\mathbf{x}'))]$ $(\mathbf{x}, \mathbf{x}' \in \mathcal{D})$.[2] Under the GP assumption, and given a set of noisy observations $\{(\mathbf{x}_h, y_h)\}_{h=1}^t$ from f, we have a closed form formula to compute the posterior mean and variance of each input point $\mathbf{x} \in \mathcal{D}$ as follows:

$$\mu_t(\mathbf{x}) := \mathbf{k}_t(\mathbf{x})^\top (K_t + \lambda I)^{-1} \mathbf{y}_t, \tag{2}$$

$$\sigma_t^2(\mathbf{x}) := k(\mathbf{x}, \mathbf{x}) - \mathbf{k}_t(\mathbf{x})^\top (K_t + \lambda I)^{-1} \mathbf{k}_t(\mathbf{x}), \tag{3}$$

where $\mathbf{k}_t(x) := (k(\mathbf{x}_1, \mathbf{x}), \dots, k(\mathbf{x}_t, \mathbf{x}))^\top$, $(K_t)_{ij} := k(\mathbf{x}_i, \mathbf{x}_j)$, I is the identity matrix of order t, $\mathbf{y}_t := (y_1, \dots, y_t)^\top$.

Algorithm 1 UCB-like Algorithm for GPMAB

```
Input: input space $\mathcal{D}$, GP prior $GP(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$, confidence level $\delta$, time horizon $T$, exploration term $\beta(t, \delta)$, weight coefficients $w_t(\mathbf{x}, \mathbf{x}')$, uncertainty term $S_t(\mathbf{x}, \mathbf{x}')$
for $t \in \{1, \dots, T\}$ do
    Pull arm:
    $\mathbf{x}_t = \arg\max_{\mathbf{x}\in\mathcal{D}} \mu_{t-1}(\mathbf{x}) + \sqrt{\beta(t,\delta)} \sum_{\mathbf{x}'\in\mathcal{D}} w_t(\mathbf{x}, \mathbf{x}') S_t(\mathbf{x}, \mathbf{x}')$
    Get reward $y_t = f_t(\mathbf{x}_t) + \epsilon_t$
    Compute $\mu_t(\mathbf{x})$, $\sigma_t(\mathbf{x})$ $\forall \mathbf{x} \in \mathcal{D}$ with Eq. (2)-(3)
```

Maximum Information Gain. In what follows, we define the *Maximum Information Gain*, whose value depends on the kernel and the dimension of the input space and on which the regret strictly depends. Formally:

Definition 1 (Information Gain). *Given a set of realization $Z_t := \{(\mathbf{x}_h, y_h)\}_{h=1}^t$ from the function f, sampled from a GP, the Information Gain is defined as:*

$$IG(Z_t \mid f) := \frac{1}{2} \log \left| I + \frac{K_t}{\lambda} \right|.$$

For a GP, the information gain has an expression depending on the input points we selected in Z_t [24]:

$$IG(Z_t \mid f) := \frac{1}{2} \sum_{h=1}^{t} \log \left(1 + \frac{\sigma_{t-1}^2(\mathbf{x}_h)}{\lambda} \right). \tag{4}$$

We can now define the maximum information gain as:

Definition 2 (Maximum Information Gain). *Given a realization of a GP f, the Maximum Information Gain of a generic set of t samples $Z_t := \{(\mathbf{x}_h, y_h)\}_{h=1}^t$ is defined as:*

$$\gamma_t := \max_{\mathbf{x}_1, \dots, \mathbf{x}_t} IG(Z_t \mid f),$$

where the maximum is over the possible choice of the set of input points $\{\mathbf{x}_1, \dots, \mathbf{x}_t\}$ in Z_t.

The upper bound on the maximum information gain depends on to specific adopted kernel, the dimension of the input space d and the cardinality t of the set Z_t, *i.e.*, the number of samples. For instance, we have:

- Linear Kernel $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{x}'$, with information gain upper bounded of $\gamma_t = O(d \log t)$;
- Squared Exponential Kernel $k(\mathbf{x}, \mathbf{x}') = \exp\left\{-\frac{||\mathbf{x}-\mathbf{x}'||^2}{2l^2}\right\}$, with information gain upper bounded of $\gamma_t = O((\log t)^{d+1})$;
- Matern Kernel $k(\mathbf{x}, \mathbf{x}') = \frac{2^{1-\nu}}{\Gamma(\nu)} r^\nu B_\nu(r)$, with $r = \frac{\sqrt{2\nu}}{l}||\mathbf{x} - \mathbf{x}'||$, with information gain upper bounded of $\gamma_t = O\left(t^{\frac{d(d+1)}{2\nu+d(d+1)}} \log t\right)$;

where $|| \cdot ||$ denotes the vector norm, l is a lengthscale parameter, $\nu > 0$ is a smoothness term, and B_ν is the modified Bessel function of the second type. These expressions can be used to fully specify the regret of a method on a given class of GPs.

[2] In what follows, without loss of generality, we assume that GPs not conditioned on data have $\mu(\mathbf{x}) \equiv 0$, *i.e.*, we have uniform null prior for the mean, and we restrict $k(\mathbf{x}, \mathbf{x}) \leq 1$ for each $\mathbf{x} \in \mathcal{D}$, *i.e.*, we assume bounded variance.

4 UCB-LIKE ALGORITHM FAMILY FOR THE GPMAB SETTING

At first, we define a general family of algorithms, which includes most of the relevant works from the state of the art, as well as the DAGP-UCB algorithm proposed here. The pseudocode of DAGP-UCB is presented in Algorithm 1. A UCB-like algorithm for the GPMAB setting is a policy selecting the next point to chose as follows:

$$\mathbf{x}_t = \arg\max_{\mathbf{x}\in\mathcal{D}} \mu_{t-1}(\mathbf{x}) + \sqrt{\beta(t,\delta)} \sum_{\mathbf{x}'\in\mathcal{D}} w_t(\mathbf{x},\mathbf{x}')S_t(\mathbf{x},\mathbf{x}'), \tag{5}$$

where $\beta(t,\delta) > 0$ is an *exploration term*, nondecreasing in t, and depending on the confidence level δ, $w_t(\mathbf{x},\mathbf{x}')$ are *weight coefficients* s.t. $0 \leq w_t(\mathbf{x},\mathbf{x}') \leq 1$ for each $\mathbf{x},\mathbf{x}' \in \mathcal{D}$, and $\sum_{\mathbf{x}'\in\mathcal{D}} w(\mathbf{x},\mathbf{x}') \leq 1$, and $S_t(\mathbf{x},\mathbf{x}') > 0$ is an *uncertainty term*. The UCB-like algorithms available in the literature, *i.e.*, GP-UCB and IGP-UCB, use the following definitions for the coefficients and the uncertainty term:

$$w_t(\mathbf{x},\mathbf{x}') := \delta_{\mathbf{x},\mathbf{x}'},$$
$$S_t(\mathbf{x},\mathbf{x}') := \sigma_{t-1}(\mathbf{x}),$$

respectively, where $\delta_{\mathbf{x},\mathbf{x}'}$ denotes the Kronecker delta, *i.e.*, a function that is one for $\mathbf{x} = \mathbf{x}'$ and zero elsewhere. These two methods differ from each other in the choice of $\beta(t,\delta)$, which is, $\beta(t,\delta) = 2\log\left(\frac{t^2\pi^2|\mathcal{D}|}{6\delta}\right)$ for GP-UCB, and $\beta(t,\delta) = \left[B + \sqrt{2(\gamma_t + 1 + \log(1/\delta))}\right]^2$ for IGP-UCB. Notably, both the algorithms select the next arm to play mainly using the information on the performance of a single arm, while, for instance, they are not explicitly taking into account the reduction of uncertainty we have on all the other arms provided by choosing $\mathbf{x}_t$. We will see that exploiting this piece of information might improve the performance of algorithms designed for GPMAB settings.

4.1 Finite Domain

It is possible to show the following result, which upper bounds the regret of the UCB-like algorithms if the input set $\mathcal{D}$ is finite:

Theorem 1. *Assume to use an UCB-like algorithm $\bar{\mathfrak{U}}$ s.t. $0 \leq S_t(\mathbf{x},\mathbf{x}') \leq \max_{\mathbf{x}\in\mathcal{D}} \sigma_{t-1}(\mathbf{x})$ for each $t \in \{1,\dots,T\}$ to solve a GP-MAB problem over a finite domain $\mathcal{D}$. For each probability $\delta \in (0,1)$, the regret $R_T(\bar{\mathfrak{U}})$ is bounded with probability at least $1-\delta$ as follows:*

$$R_T(\bar{\mathfrak{U}}) \leq \sqrt{\frac{4\left[\beta(T,\delta) + 8\log\left(\frac{T^2\pi^2|\mathcal{D}|}{6\delta}\right)\right]}{\log(1+1/\lambda)}\gamma_T T}. \tag{6}$$

Proof. It has been shown in Lemma 5.1 in [24] that with probability at least $1-\delta$, choosing $b(t,\delta) = 2\log\left(\frac{t^2\pi^2|\mathcal{D}|}{6\delta}\right)$ the following bounds holds:

$$|f(\mathbf{x}) - \mu_{t-1}(\mathbf{x})| \leq \sqrt{b(t,\delta)}\sigma_{t-1}(\mathbf{x}), \tag{7}$$

at the same time for each $\mathbf{x} \in \mathcal{D}$ and for each $t \in \{1,\dots,T\}$. Under the assumption that the previous bounds hold, the instantaneous regret is:

$$\begin{aligned}
reg_t = f(\mathbf{x}^*) - f(\mathbf{x}_t) &\leq \mu_{t-1}(\mathbf{x}^*) + \sqrt{b(t,\delta)}\sigma_{t-1}(\mathbf{x}^*) \\
&- \mu_{t-1}(\mathbf{x}_t) + \sqrt{b(t,\delta)}\sigma_{t-1}(\mathbf{x}_t) && (8)\\
&= \mu_{t-1}(\mathbf{x}^*) + \underbrace{\sqrt{\beta(t,\delta)} \sum_{\mathbf{x}'\in\mathcal{D}} w(\mathbf{x}^*,\mathbf{x}')S_t(\mathbf{x}^*,\mathbf{x}') - \sqrt{\beta(t,\delta)} \sum_{\mathbf{x}'\in\mathcal{D}} w(\mathbf{x}^*,\mathbf{x}')S_t(\mathbf{x}^*,\mathbf{x}')}_{\leq 0} && (9)\\
&+ \sqrt{b(t,\delta)}\sigma_{t-1}(\mathbf{x}^*) - \mu_{t-1}(\mathbf{x}_t) + \sqrt{b(t,\delta)}\sigma_{t-1}(\mathbf{x}_t) && (10)\\
&\leq \mu_{t-1}(\mathbf{x}^*) + \sqrt{\beta(t,\delta)} \sum_{\mathbf{x}'\in\mathcal{D}} w(\mathbf{x}^*,\mathbf{x}')S_t(\mathbf{x}^*,\mathbf{x}') \\
&- \mu_{t-1}(\mathbf{x}_t) + 2\sqrt{b(t,\delta)} \max_{\mathbf{x}\in\mathcal{D}} \sigma_{t-1}(\mathbf{x}) && (11)\\
&\leq \mu_{t-1}(\mathbf{x}_t) + \sqrt{\beta(t,\delta)} \sum_{\mathbf{x}'\in\mathcal{D}} w(\mathbf{x}^*,\mathbf{x}')S_t(\mathbf{x}_t,\mathbf{x}') \\
&- \mu_{t-1}(\mathbf{x}_t) + 2\sqrt{b(t,\delta)} \max_{\mathbf{x}\in\mathcal{D}} \sigma_{t-1}(\mathbf{x}) && (12)\\
&\leq \sqrt{\beta(t,\delta)} \sum_{\mathbf{x}'\in\mathcal{D}} w(\mathbf{x}^*,\mathbf{x}') \max_{\mathbf{x}\in\mathcal{D}} \sigma_{t-1}(\mathbf{x}) + 2\sqrt{b(t,\delta)} \max_{\mathbf{x}\in\mathcal{D}} \sigma_{t-1}(\mathbf{x}) && (13)\\
&\leq (\sqrt{\beta(t,\delta)} + 2\sqrt{b(t,\delta)}) \max_{\mathbf{x}\in\mathcal{D}} \sigma_{t-1}(\mathbf{x}), && (14)
\end{aligned}$$

where Equation (12) follows from the definition of the UCB-like algorithm family $\bar{\mathfrak{U}}$, since $\mathbf{x}_t$ is the arm with the largest bound at round t, and Equation (13) from the assumption that $S_t(\mathbf{x},\mathbf{x}') \leq \max_{\mathbf{x}\in\mathcal{D}} \sigma_{t-1}(\mathbf{x})$.

Following the proof of Theorem 1 by Srinivas et al. [24] we have that:

$$\begin{aligned}
reg_t^2 &\leq \left(\sqrt{\beta(t,\delta)} + 2\sqrt{b(t,\delta)}\right)^2 (\max_{\mathbf{x}\in\mathcal{D}} \sigma_{t-1}(\mathbf{x}))^2 && (15)\\
&\leq (2\beta(t,\delta) + 8b(t,\delta)) (\max_{\mathbf{x}\in\mathcal{D}} \sigma_{t-1}(\mathbf{x}))^2 && (16)\\
&\leq 2\,(\beta(T,\delta) + 4b(T,\delta)) \max_{\mathbf{x}\in\mathcal{D}} \sigma_{t-1}^2(\mathbf{x}) && (17)\\
&= 2\lambda\,(\beta(T,\delta) + 4b(T,\delta)) \max_{\mathbf{x}\in\mathcal{D}} \frac{\sigma_{t-1}^2(\mathbf{x})}{\lambda} && (18)\\
&\leq \frac{4\,(\beta(T,\delta) + 4b(T,\delta))}{\log(1+1/\lambda)} \frac{1}{2} \max_{\mathbf{x}\in\mathcal{D}} \log\left(1 + \frac{\sigma_{t-1}^2(\mathbf{x})}{\lambda}\right), && (19)
\end{aligned}$$

where Equation (17) follows from the fact that $\beta(t,\delta)$ and $b(t,\delta)$ are nondecreasing in t, and Equation (19) follows from the fact that for $s \in (0,\lambda^{-1})$ we have $s^2 \leq \frac{\lambda^{-1}}{\log(1+1/\lambda)} \log(1+s^2)$ for $\sigma_{t-1}(\mathbf{x}) \leq k(\mathbf{x},\mathbf{x}) \leq 1$ (see [24] for details).

Finally, using the Cauchy-Schwarz inequality we have:

$$\begin{aligned}
R_T(\bar{\mathfrak{U}}) &\leq \sqrt{T \sum_{t=1}^{T} reg_t^2} \\
&= \sqrt{T \sum_{t=1}^{T} \frac{4\,(\beta(T,\delta) + 4b(T,\delta))}{\log(1+1/\lambda)} \frac{1}{2} \max_{\mathbf{x}\in\mathcal{D}} \log\left(1 + \frac{\sigma_{t-1}^2(\mathbf{x})}{\lambda}\right)} \\
&= \sqrt{T \frac{4\,(\beta(T,\delta) + 4b(T,\delta))}{\log(1+1/\lambda)} \underbrace{\sum_{t=1}^{T} \frac{1}{2} \max_{\mathbf{x}\in\mathcal{D}} \log\left(1 + \frac{\sigma_{t-1}^2(\mathbf{x})}{\lambda}\right)}_{\leq \gamma_T}}
\end{aligned}$$

$$\leq \sqrt{\frac{4\left[\beta(T,\delta) + 8\log\left(\frac{T^2\pi^2|\mathcal{D}|}{6\delta}\right)\right]}{\log(1+1/\lambda)}\gamma_T T},$$

where the last equation follows from the result on the information gain of a GP provided in Lemma 5.3 by Srinivas et al. [24]. □

Note that, restricted to the case f is sampled from a GP, the upper bound in Theorem 1 has the same order as that one presented by Srinivas et al. [24] for GP-UCB, and by Chowdhury and Gopalan [9] for IGP-UCB.

The results provided in Theorem 1 also hold for $w(\mathbf{x}, \mathbf{x}') = 0$ or $S_t(\mathbf{x}, \mathbf{x}') = 0$, meaning that in principle one can use the posterior mean of an arm $\mu_{t-1}(\mathbf{x})$ as a criterion to choose the next arm to select. Nonetheless, it has been shown by Srinivas et al. [24] that, in practice, this choice provides a poor average empirical performance. In Section 5 we propose and motivate a weight and exploration scheme for which Theorem 1 holds and is effective in practice.

4.2 Compact Domains

If we are dealing with compact and convex domains $\mathcal{D}$ which are not finite, we need to apply a specific time dependent discretization $\mathcal{D}_t$ of the original domain, i.e., $\mathcal{D}_T \subset \mathcal{D}$, to still keep at a pace the regret of a generic UCB-like algorithm. Moreover, we require further assumptions about the smoothness of kernel generating the function to avoid dealing with functions which are too much erratic. More specifically, similarly to what has been done in Srinivas et al. [24], we require that:

Assumption 1 (Kernel Smoothness). *A kernel $k(\mathbf{x}, \mathbf{x}')$ is said to be smooth on $\mathcal{D}$ if, for each $L > 0$ and for some constants $a, b > 0$, the functions f drawn from $GP(0, k(\mathbf{x}, \mathbf{x}'))$ satisfy:*

$$\mathbb{P}\left(sup_{\mathbf{x}\in\mathcal{D}}\left|\frac{\partial f}{\partial x_j}\right| \geq L\right) \leq ae^{-\left(\frac{L}{b}\right)^2} \qquad \forall j \in \{1, \dots, d\}. \tag{20}$$

Note that most of the kernels mentioned in Section 3 satisfy Assumption 1 [12], *e.g.*, the Gaussian kernel and the Matérn one with $\nu > \frac{1}{2}$. Thanks to this assumption and using a round dependent discretization we have:

Theorem 2. *Assume to use an UCB-like algorithm $\bar{\mathfrak{U}}$ s.t. $0 \leq S_t(\mathbf{x}, \mathbf{x}') \leq \max_{\mathbf{x}\in\mathcal{D}} \sigma_{t-1}(\mathbf{x})$ runs over a discretized space $\mathcal{D}_t$ of $\tau_t = dt^2 b\sqrt{\log(da/\delta)}$ evenly spaced points in each dimension of $\mathcal{D}$ for each $t \in \{1, \dots, T\}$ to solve a GP-MAB problem in which the kernel satisfies Assumption 1. For each probability $\delta \in (0, 1)$, the regret $R_T(\bar{\mathfrak{U}})$ is bounded with probability at least $1 - \delta$ as follows:*

$$R_T(\bar{\mathfrak{U}}) \leq \frac{\pi^2}{6} + \sqrt{\frac{4\left[\beta(T,\delta) + 8\log\left(\frac{t^2\pi^2}{3\delta}\right) + 8d\log\left(dt^2 b\sqrt{\frac{2da}{\delta}}\right)\right]}{\log(1+1/\lambda)}\gamma_T T}. \tag{21}$$

Proof. The proof is adapted from Theorem 2 in [24] and extends what has been done in Theorem 1. As shown in Theorem 1 with probability at least $1 - \frac{\delta}{2}$, choosing:

$$b(t,\delta) = 2\log\left(\frac{t^2\pi^2|\mathcal{D}_t|}{3\delta}\right) == 2d\log\left(\frac{t^2\pi^2 dt^2 b\sqrt{\ln(da/\delta)}}{3\delta}\right) \tag{22}$$

the following bounds holds:

$$|f(\mathbf{x}) - \mu_{t-1}(\mathbf{x})| \leq \sqrt{b(t,\delta)}\sigma_{t-1}(\mathbf{x}), \tag{23}$$

at the same time for each $\mathbf{x} \in \mathcal{D}_t$ and for each $t \in \{1, \dots, T\}$. From now on, let us assume that the previous bounds hold. Using a union bound w.r.t. the d dimensions of the input space $\mathcal{D}$ in Assumption 1 we have:

$$\mathbb{P}\left(\forall j, sup_{\mathbf{x}\in\mathcal{D}}\left|\frac{\partial f}{\partial x_j}\right| \leq L\right) \geq dae^{-\left(\frac{L}{b}\right)^2} =: \frac{\delta}{2}, \tag{24}$$

therefore, with probability at least $1 - \frac{\delta}{2}$ we have that for any $\mathbf{x} \in \mathcal{D}$:

$$|f(\mathbf{x} - f(\mathbf{x}')| \leq L||\mathbf{x} - \mathbf{x}'||_1. \tag{25}$$

Since we choose a discretization with step $\tau_t := dt^2 b\sqrt{\log(da/\delta)}$, for any $\mathbf{x} \in \mathcal{D}$ exists $[\mathbf{x}] \in \mathcal{D}_t$ s.t.:

$$||\mathbf{x} - [\mathbf{x}]||_1 \leq \frac{d}{\tau_t} = \frac{1}{t^2 L}, \tag{26}$$

where we used the definition of δ and τ_t.

Let us denote with $\mathbf{x}_t^* \in \mathcal{D}_t$ the nearest point in the discretized input space to the global optimum $\mathbf{x}^*$. The instantaneous regret of the algorithm is:

$$reg_t = f(\mathbf{x}^*) - f(\mathbf{x}_t) = f(\mathbf{x}^*) - f(\mathbf{x}_t^*) + f(\mathbf{x}_t^*) - f(\mathbf{x}_t) \tag{27}$$

$$\leq L||\mathbf{x}^* - \mathbf{x}_t^*||_1 + f(\mathbf{x}_t^*) - f(\mathbf{x}_t) \tag{28}$$

$$\leq \frac{1}{t^2} + \mu_{t-1}(\mathbf{x}_t^*) + \sqrt{b(t,\delta)}\sigma_{t-1}(\mathbf{x}^*) - \mu_{t-1}(\mathbf{x}_t) + \sqrt{b(t,\delta)}\sigma_{t-1}(\mathbf{x}_t) \tag{29}$$

$$\leq \frac{1}{t^2} + (\sqrt{\beta(t,\delta)} + 2\sqrt{b(t,\delta)}) \max_{\mathbf{x}\in\mathcal{D}} \sigma_{t-1}(\mathbf{x}), \tag{30}$$

where we used the same proof techniques used in Equations (8)-(14) since we are evaluating the regret over a finite set of arms, as in Theorem 1.

Overall, we have:

$$R_T(\mathfrak{U}) \leq \sum_{t=1}^{T} \frac{1}{t^2} + \underbrace{\sum_{t=1}^{T}(\sqrt{\beta(t,\delta)} + 2\sqrt{b(t,\delta)}) \max_{\mathbf{x}\in\mathcal{D}} \sigma_{t-1}(\mathbf{x})}_{R_A} \tag{31}$$

$$\leq \sum_{t=1}^{\infty} \frac{1}{t^2} + \sqrt{\frac{4\left[\beta(T,\delta) + 8\log\left(\frac{t^2\pi^2|\mathcal{D}_T|}{3\delta}\right)\right]}{\log(1+1/\lambda)}\gamma_T T} \tag{32}$$

$$= \frac{\pi^2}{6} + \sqrt{\frac{4\left[\beta(T,\delta) + 8\log\left(\frac{t^2\pi^2}{3\delta}\right) + 8d\log\left(dt^2 b\sqrt{\frac{2da}{\delta}}\right)\right]}{\log(1+1/\lambda)}\gamma_T T}, \tag{33}$$

where the term R_A is bounded as done in Theorem 1. Using the union bound over the two probabilities that these events occur (Assumption 1 holds and the bounds in Equation (23) are not violated), we have a bound over the cumulative regret which holds with probability at least $1 - \delta$, which concludes the proof. □

Thanks to this result, we are also able to bound the *simple regret*, formally defined as follows:

$$SR_T(\mathfrak{U}) = \max_{\mathbf{x}\in\mathcal{D}} f(\mathbf{x}) - \max_{\mathbf{x}\in\{\mathbf{x}_1,\dots,\mathbf{x}_T\}} f(\mathbf{x}). \tag{34}$$

It is possible to show the following:

Theorem 3. *If an algorithm $\mathbb{U}$ has guarantee on the cumulative regret for a GP-MAB setting s.t. $R_T(\mathfrak{U}) \leq C(T)$ over a time horizon of T, then it also guarantees that the simple regret is s.t. $SR_T(\mathfrak{U}) \leq \frac{C(T)}{T}$ on the same time horizon.*

Proof. Fix the time horizon T. The case in which the algorithm $\mathfrak{U}$ suffers the largest simple regret and $R_T(\mathfrak{U}) \leq C(T)$ is the one in which it selects over the time horizon T always arms providing an instantaneous regret of $reg_t = \frac{C(T)}{T}$, so that it has simple regret bounded by $\frac{C(T)}{T}$ and cumulative regret bounded by $C(T)$. By contradiction, assume there exist a time instant t' s.t. the instantaneous expected regret $reg'_t > \frac{C(T)}{T}$. This implies that there exists also $t'' \neq t'$ s.t. $reg''_t < \frac{C(T)}{T}$, which violates the initial assumption that the simple regret is $\frac{C(T)}{T}$. □

Notably, if one has some guarantees on simple regret, they cannot be used to bound the cumulative regret, since they are only providing a condition on the best point we selected over the time horizon T, while the cumulative regret also bounds the regret during the whole learning process. Concluding, Theorem 3 applied to UCB-like algorithms and provides a simple regret that is $O\left(\sqrt{\frac{\gamma_T}{T}}\right)$, which is of the same order of the one corresponding to algorithms from the optimization literature, such as the one in [27].

5 THE DAGP-UCB ALGORITHM

We propose a specific instance of the UCB-like algorithms, namely the DAGP-UCB algorithm, in which the weights $w(\mathbf{x}, \mathbf{x}')$ give more importance to the region of the domain $\mathcal{D}$ which has the largest probability of being optimal, and the uncertainty term $S_t(\mathbf{x}, \mathbf{x}')$ encourages the exploration of those arms that contribute most to the reduction of the standard deviation of the GP over each point in the input space $\mathcal{D}$. Regarding the exploration term, DAGP-UCB use the same as GP-UCB $\beta(t, \delta) := 2 \log\left(\frac{t^2 \pi^2 |\mathcal{D}|}{6\delta}\right)$. Notice that the combined use of $w(\mathbf{x}, \mathbf{x}')$ with $S_t(\mathbf{x}, \mathbf{x}')$ is crucial to provide improved empirical performance (more details will be shown below together with the experimental evaluation of the algorithm).

5.1 Weight Design: Maximum Distribution

As a weight, we use the posterior probability of an arm to be maximal. We refer to the formal definition of the weight computation provided in [11] for finite domains and in [10] for compact domains. Here, we focus on the finite domain formulation:

$$w_t(\mathbf{x}, \mathbf{x}') = w(\mathbf{x}') = \mathbb{P}(f(\mathbf{x}') \geq \max_{\mathbf{x} \in \mathcal{D}} f(\mathbf{x})) \tag{35}$$

$$= \int_{-\infty}^{+\infty} \left[pdf_{t-1}(s, \mathbf{x}') \prod_{\mathbf{x}' \neq \mathbf{x}} CDF_{t-1}(s, \mathbf{x}') \right] ds, \tag{36}$$

where $pdf_{t-1}(\cdot, \mathbf{x})$ and $CDF_{t-1}(\cdot, \mathbf{x})$ are the probability density function and the cumulative distribution function, respectively, of a Gaussian distribution with mean $\mu_{t-1}(\mathbf{x})$ and variance $\sigma^2_{t-1}(\mathbf{x})$. Even if this choice seems natural if one wants to gather more information on the region of the space where the optimum is most likely to be, the weights $w_t(\mathbf{x}, \mathbf{x}')$ cannot be computed in closed form. However, if the number of arms is finite ($|\mathcal{D}| \leq +\infty$), $w_t(\mathbf{x}, \mathbf{x}')$ can be computed by numerical integration (*e.g.*, trapezoidal rule or Monte Carlo methods). In Algorithm 2, we provide a high-level description of a Monte-Carlo-like approximation method to compute the integral in Equation (35). Let N be the number of iterations set, for each $n \in \{0, \ldots, N\}$, for each arm $\mathbf{x}_m \in \mathcal{D}$, we draw a sample s_m from the corresponding probability density function $pdf(\mathbf{x}_m) = \mathcal{N}(\mu(\mathbf{x}_m), \sigma^2(\mathbf{x}_m))$, and we store the index of the arm that generated the maximum sample. Each element $w(\mathbf{x}_m)$ of the output vector is the ratio $\frac{c_m}{N}$, where c_m is the number of times s.t. $\mathbf{x}_m$ generated the sample with the largest value.

Algorithm 2 Weights Computation for Finite Domains

```
Inputs: number of arms $|\mathcal{D}|$, number of iterations $N$
Output: computed weights $w(\mathbf{x}_m)$
$c_i \leftarrow 0, \quad \forall i \in \{1, \ldots, |\mathcal{D}|\}$
for $n \in \{1, \ldots, N\}$ do
    for $m \in \{1, \ldots, |\mathcal{D}|\}$ do
        sample $s_m$ from $\mathcal{N}(\mu(\mathbf{x}_m), \sigma^2(\mathbf{x}_m))$
    $i \leftarrow argmax_{m \in \{1, \ldots, |\mathcal{D}|\}} s_m$
    $c_i \leftarrow c_i + 1$
$w(\mathbf{x}_m) \leftarrow \frac{c_m}{N}, \quad \forall m \in \{1, \ldots, |\mathcal{D}|\}$
return $w$
```

5.2 Uncertainty Terms Design: Standard Deviation

The uncertainty terms we propose use the reduction of the standard deviation given by pulling an arm to drive the selection process, formally we have:

$$S_t(\mathbf{x}, \mathbf{x}') := \sigma_{t-1}(\mathbf{x}') - \sigma_{t, \mathbf{x}_t = \mathbf{x}}(\mathbf{x}'), \tag{37}$$

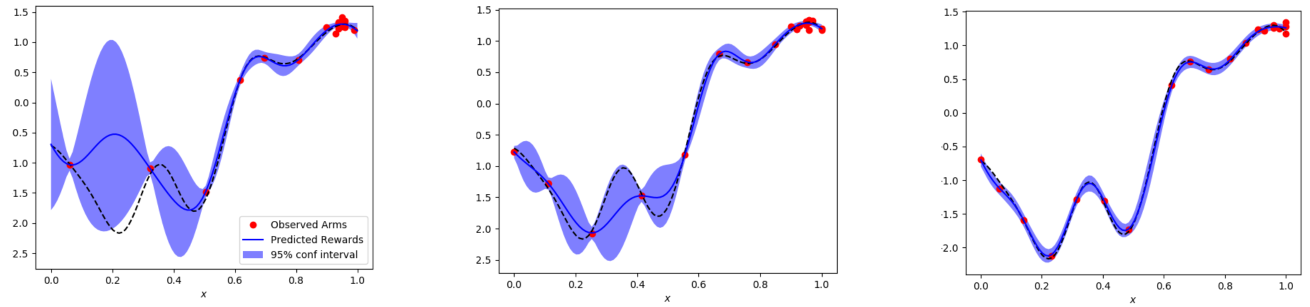
where $\sigma_{t, \mathbf{x}_t = \mathbf{x}}(\mathbf{x}')$ is the standard deviation we would have in $\mathbf{x}'$ by pulling arm $\mathbf{x}$ at round t. With the proposed definition of $S_t(\mathbf{x}, \mathbf{x}')$, the uncertainty term is bounded by $\max_{\mathbf{x} \in \mathcal{D}} \sigma_{t-1}(\mathbf{x})$ (due to the fact that $\sigma_{t-1}(\mathbf{x}') - \sigma_{t, \mathbf{x}_t = \mathbf{x}}(\mathbf{x}') \leq \sigma_{t-1}(\mathbf{x}') \; \forall \mathbf{x} \in \mathcal{D}, \; \forall \; t \geq 1$) and, therefore, Theorem 1 holds.

Note that this method requires to compute the GP conditional mean $\mu_{t-1}(\mathbf{x})$ and conditional standard deviation $\sigma_{t, \mathbf{x}_t = \mathbf{x}}(\mathbf{x}')$ for all the possible $|\mathcal{D}|^2$ pairs of arms, which, in principle, would require a complexity of $O(|\mathcal{D}| t^3)$, due to the inversion of the Gram matrix K_t built in the sampled points. Nonetheless, relying on the recursive formulas as done by Chowdhury and Gopalan [9] one could reduce the computational burden of each step to $O(|\mathcal{D}|^2)$. Finally, we can state the following result, whose proof is a simple variation of the proof of Theorem 1:

Theorem 4. *Assume to use the DAGP-UCB algorithm to solve a GP-MAB problem. For each probability $\delta \in (0, 1)$, the regret is bounded with probability at least $1 - \delta$ as follows:*

$$R_T(\bar{\mathfrak{U}}) \leq \sqrt{\frac{36 \log\left(\frac{T^2 \pi^2 |\mathcal{D}|}{6\delta}\right)}{\log(1 + 1/\lambda)} \gamma_T T}.$$

Notice that the upper bound we would obtain by applying Theorem 1 to the case stated in Theorem 4 would be the same upper bound reported in Theorem 4 except for a larger constant. More precisely, the ratio of the two upper bound is $\sqrt{2}$.



(a) DAGP-UCB

(b) GP-UCB

(c) URGP-UCB

Figure 1: Examples of optimization results provided by DAGP-UCB, GP-UCB, and URGP-UCB. The red dots are the samples selected by the algorithms, the dashed line is the real function f, the blue line represents the posterior of the expected value $\mu_{t-1}(\cdot)$, and the blue area represents the 95% confidence interval for the expected value of the function.

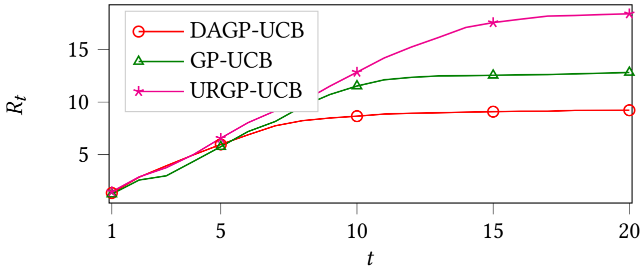


Figure 2: Comparison of the regret of DAGP-UCB with GP-UCB, and URGP-UCB.

6 EXPERIMENTS

In this section, we perform a wide range of experiments to test the DAGP-UCB algorithm on both synthetically-generated and applicative GP-MAB settings. We evaluate the performances of DAGP-UCB in terms of cumulative regret $R_t(\mathfrak{U})$, and compare them with the ones of GP-UCB [24], IGP-UCB [9], and GP-TS [9].

6.1 DAGP-UCB Rationale Evaluation

At first, we aim at showing that the specific combination of the weight coefficient scheme and uncertainty term we chose in the DAGP-UCB algorithm is crucial to provide an effective algorithm in practice. Over a setting with $|\mathcal{D}| = 100$ arms, we compare the proposed algorithm DAGP-UCB with GP-UCB, as a baseline, and a specifically crafted method, namely the Uncertainty Reduction GP-UCB (URGP-UCB), an UCB-like algorithm which uses the same uncertainty term as DAGP-UCB. Formally, URGP-UCB uses:

$$\beta(\delta, t) = 2\log\left(\frac{t^2\pi^2|\mathcal{D}|}{6\delta}\right),$$
$$w_t(\mathbf{x}, \mathbf{x}') = \delta_{\mathbf{x},\mathbf{x}'},$$
$$S_t(\mathbf{x}, \mathbf{x}') := \sigma_{t-1}(\mathbf{x}') - \sigma_{t,\mathbf{x}_t=\mathbf{x}}(\mathbf{x}').$$

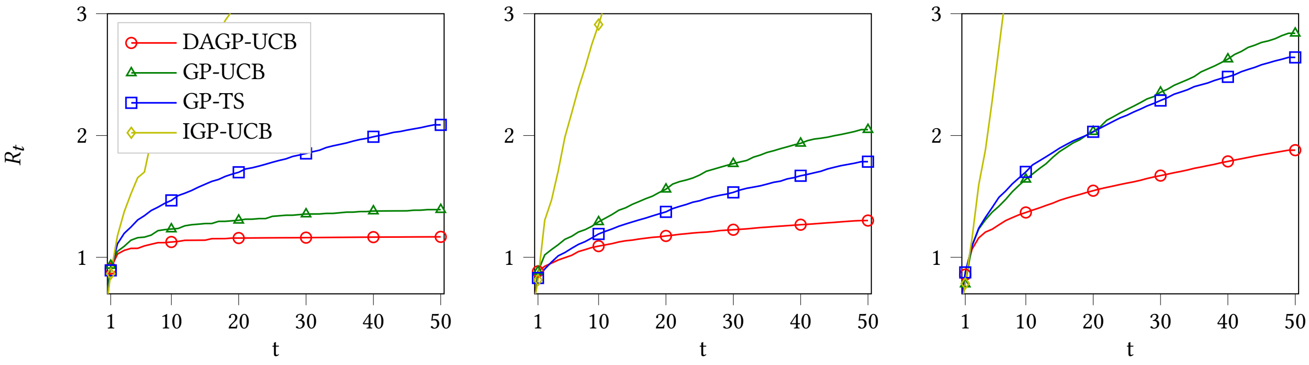
Note that also for the UR-GPUCB algorithm Theorem 1 holds, so it has the same theoretical properties of DAGP-UCB and GP-UCB. The functions f optimized in this setting are generated from a GP with zero mean, squared exponential kernel with lengthscale $l = 1$ and noise variance of $\lambda = 0.1$. The parameters of the algorithms are chosen s.t. they satisfy the assumptions provided by the regret bounds, and we set $\delta = 0.1$, and a matching kernel has been used for the posterior estimation. The presented results have been averaged over 100 independent runs of the analyzed algorithms.

Results. The results are presented in Figure 2. The algorithm that provides the best performance is DAGP-UCB, which focuses on the optimal arm after a few samples (≈ 7), after which the cumulative regret remains almost constant. While GP-UCB takes a few more samples to converge (≈ 10), the sole use of the uncertainty reduction to compute the exploration term in URGP-UCB does not allow it to match the GP-UCB performance. The reason for this behavior is evident by inspecting Figure 1, in which the URGP-UCB decreases the uncertainty of the estimates of function f over the entire input space $\mathcal{D}$ (Figure 1c). Conversely, the addition of the weighting scheme as in DAGP-UCB allows concentrating the efforts only in those areas of the domain that are closest to the optimal value $f(x^*)$ (Figure 1a). The concentration on such arms is even more evident than the one provided by GP-UCB in Figure 1b, which does not explicitly weigh them when selecting the next arm to be played.

6.2 Synthetic Settings

In the synthetic experiments, we sample the function f from a linear, a squared exponential (with lengthscale $l = 1$), and a Matérn (with parameters $\nu = 1.5$, $l = 0.2$) kernels, all with noise variance $\lambda = 0.1$. We used matching kernels as prior for the GP and the posterior computation. We analyzed a setting with $|\mathcal{D}| = 100$ arms evenly spaced over $[0, 1]$, and we set, as customary in the GP-MAB literature, the parameter $\delta = 0.1$ for all the analyzed methods. The parameters have been set as prescribed by theoretical regret upper bound results, *i.e.*, the parameter B of the IGP-UCB algorithm as the upper bound of the function norm $||f||_k$, and the parameter γ_t according to the theoretical upper bounds for the information gain provided above. We repeated the experiments for 100 runs over $T = 50$ rounds and generating 10 independent functions f.

Results. In Figure 3a-3c, we report $R_t(\mathfrak{U})$ at each round t of the algorithms averaged over the runs corresponding to the three different kernels. We observe that the DAGP-UCB algorithm has a



(a) (b) (c)

Figure 3: Regret over functions sampled from linear (a), squared exponential (b), and Matérn (c) kernel.

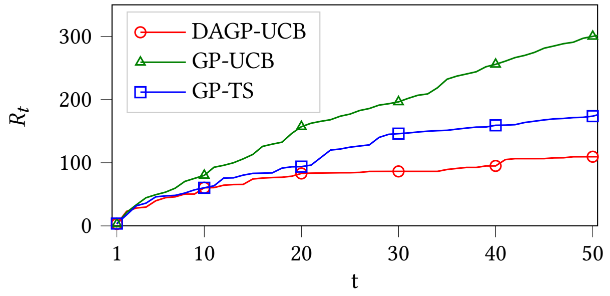


Figure 4: Regret on the advertising application.

cumulative regret significantly lower than all the other methods in all the settings. [3] Note that IGP-UCB does not present good performance, *i.e.*, its regret seems to grow linearly over time in two of the three experiments. This is due to the fact that the methods is designed for a more general case, thus, in this setting, it may perform poorly, if compared with algorithms suited for this case.

6.3 Advertising Settings

We evaluate the performance of the analyzed algorithms on the advertising budget optimization problem, defined as follows. An advertiser has to maximize the number of clicks over a set of $C = 3$ sub-campaigns, each of which is corresponding to a specific ad. At each day t, she chooses a partition $S_t = (x_t^{(1)}, x_t^{(2)}, x_t^{(3)})$ of a fixed budget $\bar{B} = 20$ to allocate on each sub-campaign i, *i.e.*, she has to pull three arms per turn under a overall budget constraint $\sum_{i=1}^{3} x_t^{(i)} \leq \bar{B}$. The number of clicks received (considered here as reward) is determined by a noisy function $cl_i(x_t^{(i)}) = cl_{\max}\left(1 - e^{-\eta_i(x_t^{(i)} - x_i)}\right) + \varepsilon$, where ε is a Gaussian, zero-mean noise with variance 0.1, and we set $cl_{\max} = 100$, $x_1 = 5$, $x_2 = 2$, $x_3 = 1$, $\eta_1 = 0.5$, $\eta_2 = 0.4$, and $\eta_3 = 0.1$. We run the analysed algorithms using $|\mathcal{D}| = 21$ evenly spaced arms in $[0, 20]$, and a Matérn kernel with $\nu = 1.5$ as correlation structure for the estimating GP and noise variance of $\lambda = 0.1$.

In this specific case, we are not necessarily interested in the estimation of the maximum value in each sub-campaign, but in those values of the budget $x_t^{(i)}$ which are most likely to be in the final budget allocation. Therefore, we define the weights values as $w_i(x_t^{(i)}) = \mathbb{P}(x_t^{(i)} = x^{*(i)})$, where $S^* = (x^{*(1)}, x^{*(2)}, x^{*(3)})$ is the optimal solution, which in our specific case is $S^* = (9, 6, 5)$. The estimation of $w_i(x_t^{(i)})$ is carried out by resorting to a repeated sampling/optimization procedure, in which the function $f(x_t^{(i)})$ is repeatedly sampled over the available budget values, and the solution is computed over these values.[4] We repeated 30 independent run of the algorithms over a time horizon of $T = 50$ days.

Results. From Figure 4, we can see that the proposed method outperforms the state-of-the-art ones starting from day $t \approx 25$. This suggests that the use of a more flexible criterion to compute the weights is able to provide a significant improvement to the choice of the budget over time. Notably, in this setting, GP-TS is able to provide significantly better performance than GP-UCB, probably due to the complexity of the environment.

7 CONCLUSIONS AND FUTURE WORKS

In this paper, we propose the DAGP-UCB algorithm, a novel MAB algorithm capable of optimizing a continuous stochastic function over a finite dataset $\mathcal{D}$ under the assumption the function is a sample from a GP. This algorithm is based on the UCB approach and exploits the regularity of the GP structure to minimize the amount of loss incurred by the algorithm during the learning process. On the one hand, we provide a theoretical result stating that the proposed algorithm meets the regret order of the algorithms constituting the state of the art for the GPMAB setting. On the other hand, we show that the proposed algorithm outperforms the state-of-the-art ones in the synthetic and in an advertising problems.

Possible lines of research are to study of tighter theoretical results on specific classes of problems for DAGP-UCB, and the design of *UCB*-like algorithms able to adapt to settings they are applied on.

[3] Confidence intervals have been omitted for visualization reasons: the 95% confidence intervals are non-overlapping for DAGP-UCB in all the experiments for $t \in [20, 50]$.

[4] The solution to such a problem can be computed by resorting to a modification of the dynamic programming procedure used to solve the knapsack problem. See [21].

REFERENCES

[1] Peter Auer, Ronald Ortner, and Csaba Szepesvári. 2007. Improved rates for the stochastic continuum-armed bandit problem. In *Proceedings of the International Conference on Machine Learning (ICML)*. Springer, 454–468.

[2] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning* 5, 1 (2012), 1–122.

[3] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. 2009. Pure exploration in multi-armed bandits problems. In *Proceedings of the International conference on Algorithmic learning theory (ALT)*. 23–37.

[4] Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári. 2011. X-armed bandits. *Journal of Machine Learning Research* 12, May (2011), 1655–1695.

[5] Adam D Bull. 2011. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research* 12, Oct (2011), 2879–2904.

[6] Roberto Calandra, André Seyfarth, Jan Peters, and Marc Peter Deisenroth. 2014. An experimental comparison of Bayesian optimization for bipedal locomotion. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 1951–1958.

[7] Wei Chen, Yajun Wang, and Yang Yuan. 2013. Combinatorial multi-armed bandit: General framework and applications. In *Proceedings of the International Conference on Machine Learning (ICML)*. 151–159.

[8] Mung Chiang, Prashanth Hande, Tian Lan, Chee Wei Tan, et al. 2008. Power control in wireless cellular networks. *Foundations and Trends® in Networking* 2, 4 (2008), 381–533.

[9] Sayak Ray Chowdhury and Aditya Gopalan. 2017. On kernelized multi-armed bandits. In *Proceedings of the International Conference on Machine Learning (ICML)*. 844–853.

[10] Carlo D'Eramo, Alessandro Nuara, Matteo Pirotta, and Marcello Restelli. 2017. Estimating the maximum expected value in continuous reinforcement learning problems. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*. 1840–1846.

[11] Carlo D'Eramo, Marcello Restelli, and Alessandro Nuara. 2016. Estimating maximum expected value through gaussian approximation. In *Proceedings of the International Conference on Machine Learning (ICML)*. 1032–1040.

[12] Subhashis Ghosal and Anindya Roy. 2006. Posterior consistency of Gaussian process prior for nonparametric binary regression. *The Annals of Statistics* 34, 5 (2006), 2413–2429.

[13] Steffen Grünewälder, Jean-Yves Audibert, Manfred Opper, and John Shawe-Taylor. 2010. Regret bounds for Gaussian process bandit problems. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*. 273–280.

[14] Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. 2012. Thompson sampling: An asymptotically optimal finite-time analysis. In *Proceedings of the International conference on Algorithmic learning theory (ALT)*. 199–213.

[15] Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. 2008. Multi-armed bandits in metric spaces. In *Proceedings of the annual ACM symposium on Theory of computing (STOC)*. ACM, 681–690.

[16] Andreas Krause and Cheng S Ong. 2011. Contextual gaussian process bandit optimization. In *Procedings of the neural information processing systems conference (NIPS)*. 2447–2455.

[17] Harold J Kushner. 1964. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering* 86, 1 (1964), 97–106.

[18] Daniel J Lizotte, Tao Wang, Michael H Bowling, and Dale Schuurmans. 2007. Automatic Gait Optimization with Gaussian Process Regression.. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)*, Vol. 7. 944–949.

[19] Jonas Močkus. 1975. On Bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference*. Springer, 400–404.

[20] Alessandro Nuara, Nicola Sosio, Francesco Trovò, Maria Chiara Zaccardi, Nicola Gatti, and Marcello Restelli. 2019. Dealing with Interdependencies and Uncertainty in Multi-Channel Advertising Campaigns Optimization. In *Proceedings of the ACM World Wide Web Conference (WWW)*. 1376–1386.

[21] Alessandro Nuara, Francesco Trovò, Nicola Gatti, and Marcello Restelli. 2018. A Combinatorial-Bandit Algorithm for the Online Joint Bid/Budget Optimization of Pay-per-Click Advertising Campaigns. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*. 2379–2386.

[22] C. E. Rasmussen and C. K. Williams. 2006. *Gaussian Processes for Machine Learning*. Vol. 1. MIT press Cambridge.

[23] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. 2012. Practical bayesian optimization of machine learning algorithms. In *Procedings of the neural information processing systems conference (NIPS)*. 2951–2959.

[24] N. Srinivas, A. Krause, S.M. Kakade, and M. Seeger. 2010. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the International Conference on Machine Learning (ICML)*. 1015–1022.

[25] Francesco Trovò, Stefano Paladino, Marcello Restelli, and Nicola Gatti. 2016. Budgeted multi-armed bandit in continuous action space. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*. IOS Press, 560–568.

[26] Francesco Trovò, Stefano Paladino, Marcello Restelli, and Nicola Gatti. 2018. Improving multi-armed bandit algorithms in online pricing settings. *International Journal of Approximate Reasoning* 98 (2018), 196–235.

[27] Zi Wang and Stefanie Jegelka. 2017. Max-value entropy search for efficient Bayesian optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*. JMLR. org, 3627–3635.

[28] Zi Wang, Bolei Zhou, and Stefanie Jegelka. 2016. Optimization as estimation with Gaussian processes in bandit settings. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*. 1022–1031.