Gaussian Process Bandits with Aggregated Feedback

Mengyan Zhang[1,2], **Russell Tsuchida**[2], **Cheng Soon Ong**[1,2]

[1] The Australian National University [2] Data61, CSIRO
mengyan.zhang@anu.edu.au, russell.tsuchida@data61.csiro.au, chengsoon.ong@anu.edu.au

Abstract

We consider the continuum-armed bandits problem, under a novel setting of recommending the best arms within a fixed budget under aggregated feedback. This is motivated by applications where the precise rewards are impossible or expensive to obtain, while an aggregated reward or feedback, such as the average over a subset, is available. We constrain the set of reward functions by assuming that they are from a Gaussian Process and propose the Gaussian Process Optimistic Optimisation (GPOO) algorithm. We adaptively construct a tree with nodes as subsets of the arm space, where the feedback is the aggregated reward of representatives of a node. We propose a new simple regret notion with respect to aggregated feedback on the recommended arms. We provide theoretical analysis for the proposed algorithm, and recover single point feedback as a special case. We illustrate GPOO and compare it with related algorithms on simulated data.

1 Introduction

In the continuum-armed bandit problem with a fixed budget, an agent adaptively chooses a sequence of N options from a continuous set (*arm space*) in order to minimise some objective given an oracle that provides noisy observations of the objective evaluated at the options (Agrawal 1995; Bubeck, Munos, and Stoltz 2011). The objective may measure the total cost, for example the *cumulative regret*, or may give an indication of the quality of the final choice, for example the *simple regret*. The simple regret setting may be viewed as black-box, zeroth order optimisation of the objective under noisy observations. In practical settings, it is possible that one cannot observe the objective directly. This motivates a more flexible notion of an oracle. In this work we consider an oracle that provides noisy average evaluations of the objective over some grid (defined in a precise sense in (1)).

For the problem of black-box optimisation of a function f under single point stochastic feedback, Munos (2014) proposed a continuum-armed bandit algorithm called Stochastic Optimistic Optimisation (StoOO) with adaptive hierarchical partitioning of arm space, under the *optimism in the face of uncertainty principle*. For bandits with aggregated feedback, Rejwan and Mansour (2020) studied finite-armed case for the combinatorial bandits under full-bandit feedback. Other related settings are discussed in § 6. We consider one important gap in the literature, best arm(s) identification for continuum-armed bandits with average rewards under a fixed budget.

Our goal is to recommend a local area with best average reward feedback. We propose *aggregated regret* (Definition 2) to reflect this objective, devise an algorithm in § 3, and show upper bound of the aggregated regret under our algorithm in § 4. In § 5, we compare our algorithm with related algorithms in a simulated environment [1]. Our algorithm shows the best empirical performance in terms of aggregated regret.

Our **contributions** are (i) a new continuum-armed bandits setting under the aggregated feedback and corresponding new simple regret notion, (ii) the first fixed budget best arms identification algorithm (GPOO) for continuum-armed bandit with noisy average feedback, (iii) theoretical analysis for the proposed algorithm, and (iv) empirical illustrations of the proposed algorithm.

2 Formulation and Preliminaries

Motivation

Two unique properties of the setting we consider are (i) the reward signal is aggregated, and (ii) the aggregation occurs on hierarchically partitioned continuous space. Gaussian Process Optimistic Optimisation (GPOO) makes use of both of these properties, as illustrated in Figure 1 and described in Algorithm 1.

Aggregated Feedback. Quantitative observations of the real-world are often made through smooth rather than instantaneous measurements. Average observations may arise from physical, hardware, privacy constraints. We provide three potential applications to motivate our setting: 1) *Radio telescope.* Arm cells are the spatial-frequency (orientation and angular resolution) coordinates of objects in the sky. The average radio wave energy (reward) can be inferred from the radio telescope for the queried area. Only the aggregated reward is observable due to frequency binning in hardware and spatial averaging. The goal is to design a policy so that one

[1] Source code and Appendix are available at https://github.com/Mengyanz/GPOO

can identify the region with the average highest radio energy with a fixed amount of querying. The first radio telescope that was used to detect extra-terrestrial radio sources (Jansky 1933) and was able to determine that the source of the radiation was from the centre of the Milky Way. 2) *Census querying*. Take the age of respondents as an example. Arms are each respondent. The oracle will return the average age (reward) of respondents inside each queried area and each query cost is the same no matter what the query is. Only the aggregated reward is allowed due to privacy concerns. The goal is to design a policy so that one can identify the region with the average highest age with a fixed amount of querying. 3) *DNA design*. In synthetic biology, one can modify nucleotides to control protein expression level (reward) (Zhang and Ong 2021a). The arms are all possible DNA sequences. The goal is to find DNA sequences with highest possible protein expression level within a given budget. The experiment is expensive and the search space is too large to enumerate. We can make a mixed culture with similar DNAs in a queried feature space and measure their aggregated reward only.

These smoothing operations present in sensor hardware designs, survey sampling methodologies and privacy-preserving data sharing motivate data analysis techniques that account for smooth or average rather than point or instantaneous measurements (Zhang et al. 2020).

Tree Structures for Continuous Spaces. Function optimisation using bandits may be achieved by simultaneously estimating and maximising some estimated statistic of a black-box objective f. This usually involves an iterative algorithm, whereby at each step a point (or points) is (are) sampled and then the estimated statistic is updated and then maximised. The estimated statistic is called an *acquisition function*, and in continuous spaces, can be computationally expensive to optimise. For example, in GP-UCB, the acquisition function is the upper confidence bound, leading to an overall computational complexity of $\mathcal{O}(N^{2d+3})$ for running the algorithm.

Hierarchically forming arms allows adaptive discretisation over the arm space, which provides a computationally efficient approach for exploring the continuous arm space. Assuming smoothness of the unknown reward function and given a budget, Munos (2014) proposed a Stochastic Optimistic Optimisation (StoOO) algorithm. They adaptively construct a tree which partitions the design space. Each leaf node in the tree represents a subset of the design space and is a candidate to be expanded. The expanded leaf node is chosen based on the *optimisation under uncertainty criteria*. Here the notion of uncertainty captures both stochastic uncertainty due to reward sampling and the inherent function variation. The reward of the leaf node is summarised by the reward obtained by evaluating the noisy objective at a some point in the subset (Munos (2014) calls these *centres*, we call them *representative points*).

Our Proposed Algorithm, Gaussian Process Optimistic Optimisation (GPOO, Algorithm 1), extends the StoOO algorithm to the case where the f is sampled from an unknown Gaussian Process (GP) and the reward feedback is an average over representatives in a subset. Using a GP allows us to encode smoothness assumptions on the function f through a choice of kernel (see Assumptions 1, 2). It also allows us to exploit the closure of Gaussian vectors under linear maps to update our belief of f under aggregated feedback in a Bayesian framework. In order to build a well-behaved tree-structure, we have to assume a certain regularity of the tree with respect to the function f (see Assumptions 3, 4), mirroring those in the StoOO algorithm. Assumption 3 ensures that as the depth of the tree grows, the allowable function variation around any node decreases. Assumption 4 rules out nodes that represent pathologically shaped subsets of the design space, such as those consisting of sets with measure zero like single points or curves.

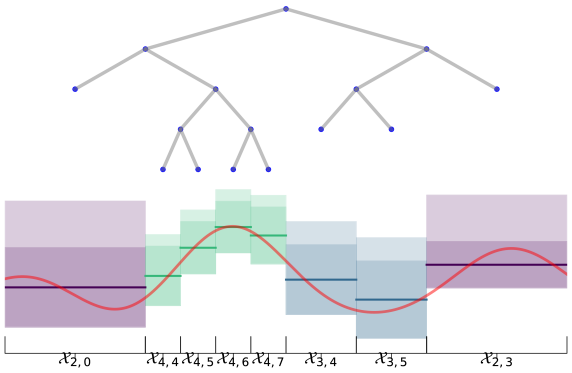

Figure 1: GPOO adaptively constructs a tree where the value associated with each node is an estimate of the aggregated reward over a cell. Red shows the reward function to be optimised. Solid horizontal lines show estimated mean aggregated reward. Dark shaded regions shows probable objective function ranges based on Bayesian uncertainty. Light shaded regions additionally account for potential function variation due to smoothness assumptions.

Our problem setting recovers the single state reward feedback as a special case. To the best of our knowledge, we are the first work address the continumm-armed bandits function optimisation problem under aggregated feedback.

Problem Setting

Let the decision space and the function to be optimised be $\mathcal{X} \subset [0,1]^d$ and $f : \mathcal{X} \to \mathbb{R}$ respectively. We consider a hierarchical partitioning of the space $\mathcal{X}$ through an adaptively-built K-ary tree. Each node (h,i) in the tree is placed at a depth h and an index i. In order to partition the space, each node (h,i) is associated with an attribute $\mathcal{X}_{h,i}$ called a *cell*. At depth h, there are K^h cells. That is, $0 \leq i \leq K^h - 1$. For any fixed h, the cells form a partition of $\mathcal{X}$. Here partition is meant in the formal sense, that is, a partition of $\mathcal{X}$ is a collection of non-empty subsets of $\mathcal{X}$ such that every $\boldsymbol{x} \in \mathcal{X}$ is in exactly one of these subsets.

We may obtain a reward from a given node (h,i) through some abstract reward signal $\mathcal{R}\big((h,i)\big)$, where the mapping $\mathcal{R}$ takes as input the attributes of the node (h,i). These attributes include the cell described above, and may also

include other attributes like the representative points, described below. Here we will focus only on a special case of $\mathcal{R}$ and leave other choices of $\mathcal{R}$ for future work.

Each node is associated with S points $\boldsymbol{x}_{h,i^s}$, $1 \leq s \leq S$, where $\boldsymbol{x}_{h,i^s} \in \mathcal{X}_{h,i}$. We stress that S is a quantity associated with the problem and may *not* be controlled by the agent. We call the collection $\mathcal{C}_{h,i} = \{\boldsymbol{x}_{h,i^s}\}_{1 \leq s \leq S}$ the *representative points* of $\mathcal{X}_{h,i}$ or (h,i). The reward of each node (h,i) is summarised by the average reward evaluated over the representative points of the cell. More precisely, we denote by $X_{h,i} \in \mathbb{R}^{S \times d}$ the *feature matrix* of cell $\mathcal{X}_{h,i}$ with each row being exactly one element of the representative points $\mathcal{C}_{h,i}$ (where the order of the rows is not important). In the round t, we select a cell $\mathcal{X}_{h_t,i_t}$ to obtain the reward r_t of cell $\mathcal{X}_{h,i}$,

$$r_t = \bar{F}(X_{h_t,i_t}) + \epsilon_t, \quad \bar{F}(X_{h_t,i_t}) := \frac{\sum_{\boldsymbol{x} \in \mathcal{C}_{h_t,i_t}} f(\boldsymbol{x})}{|\mathcal{C}_{h_t,i_t}|}, \tag{1}$$

where $\epsilon_t \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$. $\mathcal{X}_N, X_N$ and $\boldsymbol{x}_N$ will respectively denote the recommended cell, feature matrix and point (if an algorithm returns these objects). A typical goal of best arm identification with fixed budget is to minimise the simple regret (Audibert and Bubeck 2010).

Definition 1 (Simple regret)**.** *We denote an optimal arm* $\boldsymbol{x}^* \in \operatorname{argmax}_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x})$. *Let* $\boldsymbol{x}_N$ *be our recommended point after* N *rounds. The* simple regret *is defined as*

$$\hat{R}(\boldsymbol{x}_N) = f(\boldsymbol{x}^*) - f(\boldsymbol{x}_N). \tag{2}$$

In our setting, a slightly different surrogate notion of regret will be easier to analyse. Correspondingly, we introduce the *aggregated regret* under our setting in Definition 2. The goal is to minimise the aggregated regret with a fixed budget of N reward evaluations. That is, we aim to recommend a local area with highest possible aggregated reward. This is highly motivated by the applications where the measurement is over a local area instead of precise point querying, for example, the sensor hardware designs and radio telescope we mentioned in the introduction. When $S = 1$, the aggregated regret in Definition 2 is the same as *simple regret*.

Definition 2 (Aggregated regret)**.** *Let* $\mathcal{X}_N$ *be our recommended cell after* N *rounds and* X_N *the corresponding feature matrix. The* aggregated regret *is defined as*

$$R_N = f(\boldsymbol{x}^*) - \bar{F}(X_N). \tag{3}$$

This surrogate may be used to upper bound the simple regret. Note that $\min_{\boldsymbol{x} \in \mathcal{C}_N} \hat{R}_N(\boldsymbol{x}) \leq R_N$, where $\mathcal{C}_N$ is the set of representative points in $\mathcal{X}_N$. Given that cell $\mathcal{X}_N$ minimises the aggregated regret, one may apply a (finite) multi-armed bandit algorithm over the set $\mathcal{C}_N$ to solve $\min_{\boldsymbol{x} \in \mathcal{C}_N} \hat{R}_N(\boldsymbol{x})$.

We note that other choices of the abstract reward $\mathcal{R}$ are also natural. For example, a continuous analogue of (1) replaces the discrete sum over $\mathcal{C}_{h_t,i_t}$ with a continuous integral over $\mathcal{X}_{h_t,i_t}$. We sketch in Appendix D how our results might extend to this setting, but leave the details for future work.

Gaussian Processes

In order to develop our algorithm and perform our analysis, we will require f to possess a degree of smoothness. We will also need to simultaneously estimate and maximise f.

Assumption 1. *The black-box function* f *is a sample from zero-mean GP with known covariance function* k*.*

A GP $\{f(\boldsymbol{x})\}_{\boldsymbol{x} \in \mathcal{X}}$ is a collection of random variables indexed by $\boldsymbol{x} \in \mathcal{X}$ such that every finite subset $\{f(\boldsymbol{x}_i)\}_{i=1,\ldots m}$ follows a multivariate Gaussian distribution (Rasmussen and Williams 2006). A GP is characterised by its mean and covariance functions, respectively

$$\mu(\boldsymbol{x}) = \mathbb{E}[f(\boldsymbol{x})] \quad \text{and}$$
$$k(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}[(f(\boldsymbol{x}) - \mu(\boldsymbol{x}))(f(\boldsymbol{x}') - \mu(\boldsymbol{x}'))].$$

GPs find widespread use in machine learning as Bayesian functional priors. Some function of interest f is *a priori* believed to be drawn from a Gaussian process with some covariance function k and some mean function μ, where in a typical setting $\mu \equiv 0$. After observing some data, the conditional distribution of f given the data, that is, the posterior of f, is obtained. If the likelihood is Gaussian, the prior is conjugate with the likelihood and the posterior update may be performed in closed form. We describe a precise instantiation of this update tailored to our setting in § 3.

Modelling f as a GP allows us to encode smoothness properties through an appropriate choice of covariance function and also to estimate f in a Bayesian framework. This choice additionally allows us to take advantage of the closure of Gaussian distributions under linear transformations, providing us with a tool to analyse aggregated feedback.

Assumption 2. *The kernel* k *is such that for all* $j = 1, \ldots, d$*, some* $a, b > 0$ *and any* $L > 0$*,*

$$\mathbb{P}\left(\sup_{\boldsymbol{x} \in D} |\partial f / \partial x_j| \geq L\right) \leq a \exp\left(-\frac{L^2 b}{2}\right).$$

Assumption 2 implies a tail bound on $|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)|$, and may be shown to hold for a wide class of covariance functions including the squared exponential and Matérn class with smoothness $\nu > 2$. Let ℓ denote the the L_1 distance. By directly applying Theorem 5 of Ghosal and Roy (2006), we show the following in Appendix A.

Proposition 1. *Assumption 2 implies that for some constants* $a, b > 0$ *and any* $L > 0$*,* $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{X}$*,*

$$\mathbb{P}\big(|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)| \geq L\ell(\boldsymbol{x}_1, \boldsymbol{x}_2)\big) \leq a e^{-L^2 b/2}.$$

If $\sup_{x \in \mathcal{X}} \frac{\partial^2}{\partial x_j \partial x_j'} k(x, x')\big|_{x=x'} < \infty$ *and* k *has mixed derivatives of order at least* 4*, then* k *satisfies Assumption 2.*

Assumptions 1 and 2 were also made by (Srinivas et al. 2009), but to the best of our knowledge we are the first to exploit closure of GPs under linear maps in the setting of best arm identification under fixed budget.

3 Algorithm: GPOO

Inspired by StoOO (Munos 2014), we propose Gaussian Process Optimistic Optimisation (GPOO), under the formulations introduced in § 2. In order to describe our algorithm, we first describe how to compute the posterior predictive GP.

Gaussian Process Posterior Update

In round $t \in 1, \ldots, N$, we represent our prior belief over f using a GP. Our prior in round t is the posterior after so-far observed noisy observations of groups up to round t. The posterior update is a minor adjustment to the typical posterior inference step that would be employed if we were to consider only single point (not aggregated) reward feedback, exploiting the fact that multivariate Gaussian vectors are closed under linear transformations.

Define $X_{1:t} \in \mathbb{R}^{tS \times d}$ to be the vertical concatenation of X_{h_j,i_j} for $j = 1, \ldots, t$. Similarly define $Y_{1:t} \in \mathbb{R}^t$ to be a vector with jth row equal to r_j, where $j = 1, \ldots, t$. In round t we observe the feature-reward tuple $\left(X_{h_t,i_t}, r_t\right)$. We may write $r_t = \boldsymbol{a}^\top f(X_{h_t,i_t}) + \epsilon_t$ for corresponding $\boldsymbol{a} \in \mathbb{R}^{S \times 1}$ having all entries equal to $1/S$.

More compactly, all rounds $t \in \{1, \ldots, N\}$ may be represented through a vector equation. Let $A_{1:t} \in \mathbb{R}^{t \times tS}$ with pqth entry equal to $1/K$ if the qth element of $X_{1:t}$ is sampled at round p and zero otherwise. In what follows, we define

$$Z_{1:t} := (X_{1:t}, Y_{1:t}, A_{1:t}),$$

which completely characterises the history of observations up to round t. We may write $Y_{1:t} = A_{1:t} f(X_{1:t}) + \epsilon_{1:t}$, where $\epsilon_{1:t} \in \mathbb{R}^t$ denotes a vector with ith entry ϵ_i.

Let $X_* \in \mathbb{R}^{n_* \times d}$ denote a matrix of test indices, and let $\boldsymbol{a}_* \in \mathbb{R}^{n_* \times 1}$ denote some corresponding weights. The posterior predictive distribution of $\boldsymbol{a}_*^\top f(X_*)$ given all the history $Z_{1:t}$ up to round t is also a Gaussian and satisfies

$$\begin{aligned} \boldsymbol{a}_*^\top f(X_*) \mid Z_{1:t} &\sim \mathcal{N}\left(\boldsymbol{a}_*^\top \mu(X_* \mid Z_{1:t}), \boldsymbol{a}_*^\top \Sigma(X_* \mid Z_{1:t}) \boldsymbol{a}_*\right), \\ \mu(X_* \mid Z_{1:t}) &= M Y_{1:t}, \\ \Sigma(X_* \mid Z_{1:t}) &= k\left(X_*, X_*\right) - M A_{1:t} k(X_t, X_*), \end{aligned} \tag{4}$$

where

$$M = k(X_t, X_*)^T {A_{1:t}}^\top \left(A_{1:t} k(X_t, X_t) {A_{1:t}}^\top + \sigma^2 \boldsymbol{I}\right)^{-1}.$$

With an iterative Cholesky update we may perform all N inference steps in one $\mathcal{O}(N^3)$ sweep. See Appendix B.

Notions of Uncertainty and Function Variation

We introduce the key concept for selecting which node to sample, the *b-value* $b\big(X_* \mid \beta, Z, \boldsymbol{a}_*\big)$, which is the sum of three terms: the posterior mean of the corresponding feature matrix X_*, the confidence interval and function variation,

$$\begin{aligned} &b\big(X_* \mid \beta, Z, \boldsymbol{a}_*\big) := \\ &\quad \boldsymbol{a}_*^\top \mu(X_* \mid Z) + CI(X_* \mid \beta, Z, \boldsymbol{a}_*) + \delta(h). \end{aligned}$$

The confidence interval is defined in terms of the posterior variance and an exploitation/exploration parameter β,

$$CI\big(X_* \mid \beta, Z, \boldsymbol{a}_*\big) := \beta^{1/2} \sqrt{\boldsymbol{a}_*^\top \Sigma(X \mid Z) \boldsymbol{a}_*}.$$

The last term $\delta(h)$ is the smoothness function depending on the node depth h that satisfies Assumption 3. $\delta(h)$ gives an upper bound of the function deviation of a cell in a given depth. Figure 2 illustrates the above quantities. We further define time-dependent b-value and confidence interval

$$b_{h,i}(t) := b\big(X_{h_t,i_t} \mid \beta_t, Z_{1:t-1}, \boldsymbol{a}\big) \quad \text{and} \tag{5}$$

$$CI_t(X_*) := CI\big(X_* \mid \beta_t, Z_{1:t-1}, \boldsymbol{a}\big), \tag{6}$$

where $\beta_t \sim \mathcal{O}(\log t)$ and will be specified in Lemma 1.

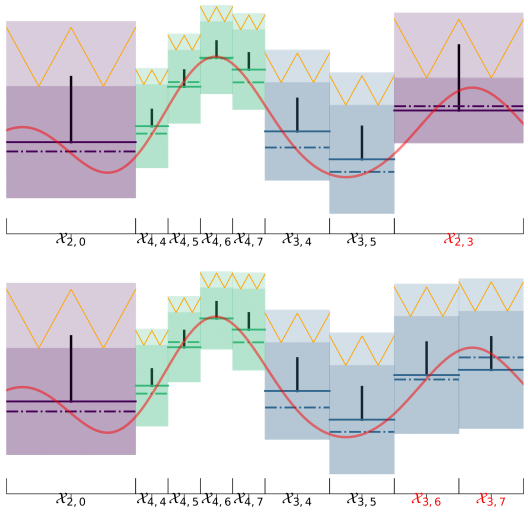

Figure 2: Quantities taken just before line 4 of Algorithm 1 for $t = 9$ (top) and $t = 10$ (bottom). For each cell $\mathcal{X}_{h,i}$, solid and dashed horizontal lines show $\boldsymbol{a}^\top \mu(X_{h,i} \mid Z_t)$ and $\overline{F}(X_{h,i})$ respectively. Dark shaded regions show $CI_t(X_{h,i})$ with colours indicating depth of cell. One-sided b-values are also shown by light shaded regions. Vertical black bars indicate $\delta(h)$. Shown in orange is a probable Lipschitz bound on the function value, due to event ξ and Proposition 1. Here $S = 2$ and the representative points form a uniform grid, so that the Lipschitz bounds form a W shape. This bound is in turn bounded by the b-values. When $t = 9$, $\mathcal{X}_{2,3}$ is expanded since it has the highest b-value. When $t = 10$, no cell will be expanded since $\delta(4) < CI_{10}(X_{4,6})$; instead the estimate of the function will be refined by sampling the reward of $\mathcal{X}_{4,6}$.

Assumption 3 (Decreasing diameter)**.** *There exists a decreasing sequence $\delta(h) > 0$ such that for some $L > 0$ (in Assumption 2), any depth h and any cell $\mathcal{X}_{h,i}$, $\sup_{\boldsymbol{x} \in \mathcal{X}_{h,i}} L\ell\left(\boldsymbol{x}_{h,i}, \boldsymbol{x}\right) \leq \delta(h)$ for any representative point $\boldsymbol{x}_{h,i}$.*

Assumption 3 means that at any depth h, the largest possible distance $\sup_{\boldsymbol{x} \in \mathcal{X}_{h,i}} L\ell\left(\boldsymbol{x}_{h,i}, \boldsymbol{x}\right)$ between any point to any representative point within cell $\mathcal{X}_{h,i}$ is decreasing with respect to h. This is intuitive since we have assumed smoothness (Assumption 2) and constructed cells hierarchically. This assumption links the distance in reward space to a $\delta(h)$, which is the core concept in theoretical analysis (see Figure 3). Compared with Munos (2014), we introduce a smoothness parameter L to suit our analysis with GPs.

We present the GPOO in Algorithm 1. The key idea is that we construct a tree by adaptively discretising over the arm space. The algorithm includes two main parts:

Algorithm 1: GPOO

Input: natural number K (-ary tree), function f to be optimised, smoothness function δ, budget N, the maximum depth nodes can be expanded $h_{\max}$.

Init: tree $\mathcal{T}_0 = \{(0,0)\}$ (corresponds to $\mathcal{X}$), leaves $\mathcal{L}_0 = \mathcal{T}_0$.

1: **for** $t = 1$ to N **do**
2: Select any $(h_t, i_t) \in \operatorname{argmax}_{(h,i)\in\mathcal{L}_{t-1}} b_{h,i}(t)$.
3: Observe reward $r_t = \overline{F}(X_{h_t,i_t}) + \epsilon_t$.
4: Update posteriors $\boldsymbol{a}^\top \mu(\cdot|Z_{1:t}), \boldsymbol{a}^\top \Sigma(\cdot|Z_{1:t})\boldsymbol{a}$
5: Update confidence intervals and b-values for all nodes $(h,i) \in \mathcal{L}_{t-1}$ according to E.q. (6)(5).
6: **if** $\delta(h_t) \geq CI_t(X_{h_t,i_t})$ and $h_t \leq h_{\max}$ **then**
7: Expand node (h_t, i_t) (partition $\mathcal{X}_{h_t,i_t}$ into K subsets) into children nodes $\mathcal{C}_t = \{(h_t+1, i_1), \ldots, (h_t+1, i_K)\}$. $\mathcal{T}_t = \mathcal{T}_{t-1} \cup \mathcal{C}_t$.
8: $\mathcal{L}_{t-1} = \mathcal{L}_{t-1} \backslash \{(h_t, i_t)\}; \mathcal{L}_t = \mathcal{L}_{t-1} \cup \mathcal{C}_t$.
9: **end if**
10: **end for**
11: **Return** The node with index (h', i') such that $h' = \operatorname{argmax}_{h|(h,i)\in\mathcal{T}_N\backslash\mathcal{L}_N} h$. and $i' = \operatorname{argmax}_{i|(h',i)\in\mathcal{T}_N} \mu(X_{h',i}|Z_{1:t})$

Select Node and Update: In each round $t \in [1, N]$, a node (h_t, i_t) is selected from leaves with the highest b-value (5). The reward is sampled as the average of the function value over S representative points plus Gaussian noise. We then update the posteriors over cells of leaf nodes, allowing the confidence interval and b-values to also be updated.

Expand Node: As we increase the number of samples for a given node, the confidence interval of that node continues to decrease. When the confidence interval is smaller than or equal to the function variation, our function estimate is more precise than the current cell range. That is, when the condition on line 6 satisfied, the node can be expanded into K children by partitioning cells into K subsets. When the budget is reached, we return the node with the highest posterior mean prediction among non-leaf nodes. We select recommendations from non-leaf nodes because predictions of non-leaf nodes (satisfying event ξ_2 in Section 4) are more precise than leaves. Figure 2 shows various quantities in the algorithm over two iterations.

Remark 1 (Comparison to StoOO)**.** *(i) StoOO assumes known smoothness of the function but does not utilise correlations between arms and recommend based on predictions. GPOO recommends nodes with the GP predictions. (ii) GPOO can address aggregated feedback, while StoOO can only deal with single state feedback. We also extend StoOO to address aggregated feedback in Appendix C.*

Remark 2 (Computational Complexity)**.** *The cost of performing global maximisation of an acquisition function can be exponential in the design space dimension d. The acquisition function for GP-UCB is the UCB at each point $\boldsymbol{x} \in \mathcal{X}$, leading to a total computational cost of $\mathcal{O}(N^{2d+3})$. The problem of selecting an optimal arm from a finite subset of arms is a case of combinatorial optimisation, and the computational cost can be exponential in the number of arms. For GPOO, for every round t, we consume $\mathcal{O}(t^2)$ for the GP inference procedure. Every time a leaf node is expanded, $K-1$ new leaf nodes are added to the tree, so that less than $(K-1)t$ b-values and confidence intervals must be computed. This leads to a total cost of $\mathcal{O}(N^4(K-1))$.*

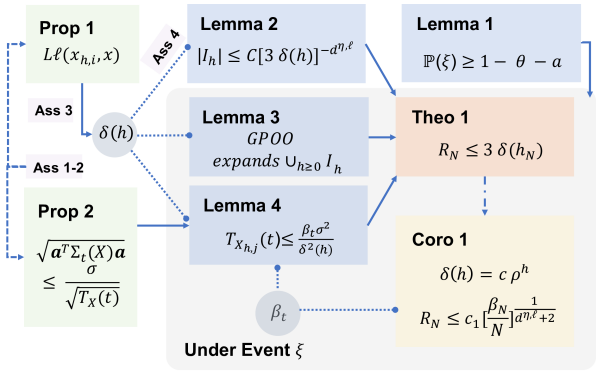

Figure 3: Proof Roadmap

4 Theoretical Analysis for Aggregated Regret

In this section, we provide the theoretical analysis of GPOO. We show our proof roadmap in Figure 3. The full proofs can be found in Appendix A. Our technical contribution is adapting the analysis of the hierarchical partition idea (Munos 2014) to the case reward is sampled from a GP (Srinivas et al. 2009) and is aggregated.

Recall N is the budget, $h_{\max}$ is a parameter representing the deepest allowable depth of tree, $\boldsymbol{x}^* \in \operatorname{argmax}_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x})$ is an optimal point in input space and $f^* = f(\boldsymbol{x}^*)$. Let $\boldsymbol{x}_{h_t^*, j_t^*}$ be a representative point inside the cell $\mathcal{X}_{h_t^*, j_t^*}$ containing $\boldsymbol{x}^*$ in round t. We first define event ξ, under which we present our aggregated regret upper bound in Theorem 1. Define event ξ as $\xi = \xi_1 \cap \xi_2$, where

$$\xi_1 := \left\{\forall 1 \leq t \leq N \quad f^* - f\left(\boldsymbol{x}_{h_t^*, j_t^*}\right) \leq L\ell(\boldsymbol{x}_{h_t^*, j_t^*}, \boldsymbol{x}^*)\right\},$$

$$\xi_2 := \{\forall 0 \leq h \leq h_{\max}, 0 \leq i < K^h, 1 \leq t \leq N \quad \left|\boldsymbol{a}^\top \mu(X_{h,i} \mid Z_{t-1}) - \overline{F}(X_{h,i})\right| \leq CI_t(X_{h,i})\}.$$

The event ξ provides the probabilities environment for our theoretical analysis, which is the union of two events: event ξ_1 describes the function f local smoothness around its maximum; event ξ_2 captures a concentration property from the estimation to representative summary statistics of reward, which has been shown as an critical part for the regret analysis (Lattimore and Szepesvári 2020; Zhang and Ong 2021b). In the following lemma, b is the constant in Proposition 1. Since θ is positive and less than $1-a$, $\mathbb{P}(\xi) \geqslant 0$.

Lemma 1. *Define $a = h_{\max}\exp(-\frac{L^2 b}{2})$ with L as a constant specified in Assumption 3. Let $\beta_t = 2\log(M\pi_t/\theta)$, where $M = \sum_{h=0}^{h_{\max}} K^h$, $\pi_t = \pi^2 t^2/6$. For K-ary tree and all $\theta \in (0, 1-a)$, we have $\mathbb{P}(\xi) \geq 1 - a - \theta$.*

To obtain the regret bound for our algorithm, we need to upper bound two pieces: the number of expanded nodes in

each depth of the tree (Lemma 2), which depends on the concept of near-optimality dimension in Definition 4 proposed in Munos (2014); and the number of times a node can be sampled before expansion (Lemma 4), which can be inferred from the GP posterior variance upper bound (Proposition 2). We define the set of expanded nodes at depth h as I_h

$$I_h := \{(h,i) | \bar{F}(X_{h_t,j_t}) + 3\delta(h) \geq f^*\}. \tag{7}$$

To upper bound the number of nodes in I_h, we introduce the concept of near-optimality dimension in Definition 4, which relates to function f, ℓ and depends on the constant η.

Definition 3 ($\epsilon-$optimal state)**.** *For any $\epsilon > 0$, define the $\epsilon-$optimal states as $\mathcal{X}_\epsilon := \{\boldsymbol{x} \in \mathcal{X} | f(\boldsymbol{x}) \geq f^* - \epsilon\}$.*

Definition 4 ((η,ℓ)-near-optimality dimension $d^{\eta,\ell}$, (Munos 2014))**.** *For any $\epsilon > 0$, with $\epsilon-$optimal states $\mathcal{X}_\epsilon$ defined in Defition 3, $d^{\eta,\ell}$ is the smallest $d \geq 0$ such that there exists $C > 0$ such that the maximal number of disjoint $\ell-$balls of radius $\eta\epsilon$ with centre in $\mathcal{X}_\epsilon$ is less than $C\epsilon^{-d}$.*

We further introduce the well-shaped cells assumption, which implies that the cells partitioned by our algorithm should contain each representatives points in a $\ell-$ball, e.g. the representatives points should not be on the boundary. This helps us upper bound the number of expanded nodes (Lemma 2). Unlike Munos (2014), we require it to hold for any representative point to suit our aggregated feedback setting. We then show that only the set of nodes in I_h are expanded with GPOO in Lemma 3.

Assumption 4 (Well-shaped cells)**.** *There exists $\nu > 0$ s.t. for any depth $h \geq 0$, any cell $\mathcal{X}_{h,i}$ contains an ℓ-ball of radius $\nu\delta(h)$ centred in each point in the representative set $\boldsymbol{x} \in \mathcal{C}_{h,i}$.*

Lemma 2. *Under Assumption 4, $|I_h| \leq C[3\delta(h)]^{-d^{\eta,\ell}}$.*

Lemma 3. *Under event ξ and Assumption 3 , GPOO only expands nodes in the set $I : \cup_{h\geq 0} I_h$.*

We now move to derive an upper bound of the number of draws of expanded nodes. Following Proposition 3 of Shekhar, Javidi et al. (2018), using mutual information, we upper bound the GP posterior variance in Proposition 2.

Proposition 2. *Let f be a sample from a GP with zero mean and covariance function k. Let $X \in \mathbb{R}^{S\times d}$ and let $\Sigma_t(X)$ denote the posterior predictive GP covariance $\Sigma(X \mid Z_{1:t})$ according to* (4)*. If the history $Z_{1:t}$ contains at least $T_X(t)$ observations following the model $Y = \boldsymbol{a}^\top f(X) + \epsilon$ where $\epsilon \sim \mathcal{N}(0,\sigma^2)$, then we have $\sqrt{\boldsymbol{a}^\top \Sigma_t(X)\boldsymbol{a}} \leq \frac{\sigma}{\sqrt{T_X(t)}}$, where $\boldsymbol{a} \in \mathbb{R}^{S\times 1}$ has all entries equal to $1/S$.*

With Proposition 2, we upper bound the number of draws for any nodes under event ξ in the following lemma.

Lemma 4. *Under event ξ, suppose a node (h,j) (with corresponding feature matrix $X_{h,j}$ as defined in § 2) is sampled at least $T_{X_{h,j}}(t)$ times up to round t. Then we have $T_{X_{h,j}}(t) \leq \frac{\beta_t\sigma^2}{\delta^2(h)}$, where σ^2 is the noise variance, β_t is defined in Lemma 1.*

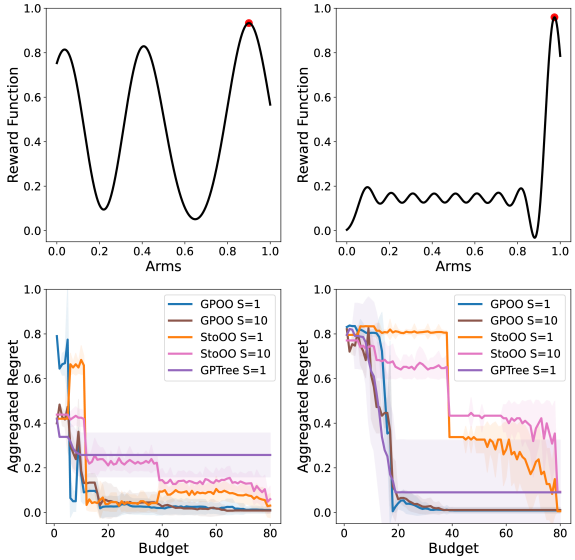

Figure 4: Reward functions f (row 1) and aggregated regrets (row 2), with shaded regions indicate one standard deviation. We perform 30 independent runs with a budget N up to 80.

We show the aggregated regret bound for any δ under our assumptions in Theorem 1, and show a special case of exponential diameter in Corollary 1.

Theorem 1. *Define h_N as the smallest integer h' up to round N such that*

$$\frac{N}{\beta_N} \leq K \sum_{h=0}^{h'} C[3\delta(h)]^{-d^{\eta,\ell}} \frac{\sigma^2}{\delta^2(h)}. \tag{8}$$

For constant a specified in Lemma 1, and $\theta \in (0, 1-a)$, with probability $1-\theta-a$, the simple regret of GPOO satisfies

$$R_N \leq 3\delta(h_N).$$

Corollary 1 (Regret bound for exponential diameters)**.** *Assume $\delta(h) = c\rho^h$ for some constants $c > 0$, $\rho < 1$. For constant a specified in Lemma 1, and $\theta \in (0, 1-a)$, with probability $1-\theta-a$, the simple regret of GPOO satisfies*

$$R_N \leq c_1 \left[\frac{\beta_N}{N}\right]^{\frac{1}{d^{\eta,\ell}+2}},$$

where $c_1 = \frac{3^{-d^{\eta,\ell}} KC\sigma^2}{\rho^{-(d^{\eta,\ell}+2)}-1}$. Recall β_N is in rate $\mathcal{O}(\log N)$.

5 Experiments

We investigate the empirical performance of GPOO on simulated data. We compare the aggregated regret obtained by GPOO with related algorithms, and illustrate how different parameters influence performance.

We show regret curves of different algorithms for two simulated reward functions in Figure 4. Our decision space is chosen to be $\mathcal{X} = [0,1]$. The reward functions are sampled from a GP posterior conditioned on hand-designed samples (listed in Appendix E), with radial basis function (RBF) kernel having lengthscale 0.05 and variance 0.1. The GP noise

standard variation was set to 0.005. The reward noise is i.i.d. sampled zero-mean Gaussian distribution with standard deviation 0.1.

For our experiment, we consider two cases of feedback: *single point feedback* (S = 1), where the reward is sampled from the centre (representative point) of the selected cell and *average feedback* (S = 10), where the reward is the average of samples from the centre of each sub-cell, which are obtained by splitting the cell into intervals of equal size. Following Corollary 1, we choose $\delta(h) = c2^{-h}$, where c is chosen via cross-validation. The algorithms are evaluated by the aggregated regret in Definition 2 ($S = 1$ for single point feedback, $S = 10$ for average feedback).

The related algorithms we compare with include: (i) StoOO (Munos 2014): the error probability needed for StoOO is chosen to be 0.1. (ii) AVE-StoOO: We extend the StoOO algorithm to the case where the rewards are aggregated feedback in Appendix C. (iii) GPTree (Shekhar, Javidi et al. 2018). We discuss the related algorithms in § 6.

The first reward function (first row) is designed to show the performance of algorithms when there are several relatively similar local maximums. To find the global optimal region, the algorithm needs to predict the function values of high-value cells in high precision quickly. The second reward function is designed to show the case where the reward function is periodic-like. To achieve low regret, the algorithm needs to avoid disruptions by the patterns hidden in rewards, e.g. avoid consistently sampling points with similar function values. GPOO (for both single and average feedback cases) has the best performance for the two reward functions. The aggregated regret convergences quickly and remains stable. GPTree tends to stuck in local maximum (in a finite budget) since their parameter β_N, balancing the posterior mean and standard deviation, only depends on the budget N and does not increase over different rounds, while in our algorithm β_t increases like $\mathcal{O}(\log t)$. As expected, StoOO-based algorithms converge more slowly, since they only use empirical means instead of GP predictions and they are designed with a more broad family of reward functions in mind. We also show an example where aggregated feedback can be beneficial for recommendations in Figure 5, which is deferred to Appendix E due to the page limit.

6 Related Work

We mainly review the related work in two aspects: GP-elated bandits and bandits work considering the aggregated feedback. Refer to Lattimore and Szepesvári (2020) for a comprehensive review of bandits literature.

Gaussian Process Bandits By assuming the unknown reward is sampled from a GP, bandit algorithms can be applied to black-box function optimisation. Srinivas et al. (2009) studied *regret minimisation* under single arm feedback, where arms are recommended sequentially under a Upper Confidence Bound (UCB) policy. They covered both the finite arm space and continuous arm space. GP bandit is also studied and applied in Bayesian optimisation in a simple regret setting (Shahriari et al. 2016).

As far as we know, no current GP bandit works address aggregated feedback. Accabi et al. (2018) studied the *semi-bandit* setting where both the individual labels of arms and the aggregated feedback in the selected subset are observed in each round, while we consider a harder setting where only the aggregated feedback is observed. For non-aggregated feedback, the most related work to ours is Shekhar, Javidi et al. (2018), which extends the StoOO algorithm to the case where the f is sampled from unknown GP with theoretical analysis. They did not provide empirical evaluations on the proposed algorithm and their algorithm highly depends on large amount of theoretical-based parameters. We compared with their algorithm in Section 5.

Aggregated Feedback Related Settings The "aggregated feedback" is first used in Pike-Burke et al. (2018), in which they studied bandits with *Delayed, Aggregated Anonymous Feedback*. At each round, the agent observes a delayed, sum of previously generated rewards. Different from our setting, they considered finite, independent arms setting and cumulative (pseudo-)regret minimisation problem.

There are two types of bandits problems that are related to the aggregated feedback setting. One related setting is *full-Bandits* (Rejwan and Mansour 2020; Du, Kuroki, and Chen 2021) which is studied under *combinatorial bandits* (Chen, Wang, and Yuan 2013) with finite arm space, where only the aggregated feedback over a combinatorial set is observed. No work has addresses GP bandits with full-bandit feedback. Our setting is different from full-bandits since we consider a continuous arm space and the objective is to identify local areas for function optimisation. Another related setting is *slate bandits* (Dimakopoulou, Vlassis, and Jebara 2019), where a slate has fixed number of positions named *slots*. The slate-level reward can be an aggregation over slot-level rewards. However, the slate bandits setting assumes slate-slot two levels rewards and each slate has fixed choices of slots, which is significantly different from our setting.

7 Conclusion and Future Work

We introduced a novel setting for continuum-armed bandits where only the aggregated feedback can be observed. This is motivated by applications where aggregated reward is the only feedback or precise reward is expensive to access. We proposed Gaussian Process Optimistic Optimisation (GPOO) in Algorithm 1 which adaptively searches a hierarchical partitioning of the space. We provided an upper bound on the aggregated regret in Theorem 1 and empirically evaluated our algorithm on simulated data in § 5.

It may be possible to extend our framework to some other interesting settings. For example, instead of only observing the aggregated function value corrupted by Gaussian noise, one may consider the case where the gradient is additionally observed. This is recently studied in Shekhar and Javidi (2021) on GP bandits with cumulative regret bound. It would be interesting to study the simple regret (pure exploration), under the aggregated feedback setting. Since GPs are closed under linear operators (including integrals, derivatives and Fourier transforms), one can potentially handle the case where some combination of these is observed.

References

Accabi, G. M.; Trovo, F.; Nuara, A.; Gatti, N.; and Restelli, M. 2018. When Gaussian Processes Meet Combinatorial Bandits: GCB. In *14th European Workshop on Reinforcement Learning*, 1–11.

Agrawal, R. 1995. The continuum-armed bandit problem. *SIAM journal on control and optimization*, 33(6): 1926–1951.

Audibert, J.-Y.; and Bubeck, S. 2010. Best Arm Identification in Multi-Armed Bandits. In *Proceedings of the 23rd Annual Conference on Learning Theory (COLT)*.

Bubeck, S.; Munos, R.; and Stoltz, G. 2011. Pure exploration in finitely-armed and continuous-armed bandits. *Theoretical Computer Science*, 412(19): 1832–1852.

Chen, W.; Wang, Y.; and Yuan, Y. 2013. Combinatorial multi-armed bandit: General framework, results and applications. *30th International Conference on Machine Learning, ICML 2013*, 151–159.

Dimakopoulou, M.; Vlassis, N.; and Jebara, T. 2019. Marginal Posterior Sampling for Slate Bandits. In *IJCAI*, 2223–2229.

Du, Y.; Kuroki, Y.; and Chen, W. 2021. Combinatorial pure exploration with full-bandit or partial linear feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 7262–7270.

Ghosal, S.; and Roy, A. 2006. Posterior consistency of Gaussian process prior for nonparametric binary regression. *The Annals of Statistics*, 34(5): 2413–2429.

Jansky, K. G. 1933. Electrical disturbances apparently of extraterrestrial origin. *Proceedings of the Institute of Radio Engineers*, 21(10): 1387–1398.

Lattimore, T.; and Szepesvári, C. 2020. *Bandit Algorithms*. Cambridge University Press.

Munos, R. 2014. From bandits to Monte-Carlo Tree Search: The optimistic principle applied to optimization and planning.

Pike-Burke, C.; Agrawal, S.; Szepesvari, C.; and Grunewalder, S. 2018. Bandits with delayed, aggregated anonymous feedback. In *International Conference on Machine Learning*, 4105–4113. PMLR.

Rasmussen, C.; and Williams, C. 2006. *Gaussian Process for Machine Learning*. MIT Press.

Rejwan, I.; and Mansour, Y. 2020. Top-k Combinatorial Bandits with Full-Bandit Feedback. In *Algorithmic Learning Theory*, 752–776. PMLR.

Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R. P.; and de Freitas, N. 2016. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, 104(1): 148–175. Conference Name: Proceedings of the IEEE.

Shekhar, S.; and Javidi, T. 2021. Significance of Gradient Information in Bayesian Optimization. In *International Conference on Artificial Intelligence and Statistics*, 2836–2844. PMLR.

Shekhar, S.; Javidi, T.; et al. 2018. Gaussian process bandits with adaptive discretization. *Electronic Journal of Statistics*, 12(2): 3829–3874.

Srinivas, N.; Krause, A.; Kakade, S. M.; and Seeger, M. 2009. Gaussian process optimization in the bandit setting: No regret and experimental design. *International Conference on Machine Learning*.

Zhang, M.; and Ong, C. S. 2021a. Opportunities and Challenges in Designing Genomic Sequences. *ICML Workshop on Computational Biology*.

Zhang, M.; and Ong, C. S. 2021b. Quantile Bandits for Best Arms Identification. In *International Conference on Machine Learning*, 12513–12523. PMLR.

Zhang, Y.; Charoenphakdee, N.; Wu, Z.; and Sugiyama, M. 2020. Learning from Aggregate Observations. *Advances in Neural Information Processing Systems*, 33.