Multi-Step Bayesian Optimization for One-Dimensional Feasibility Determination

J. Massey Cashore Lemuel Kumarga Peter I. Frazier

Abstract

Bayesian optimization methods allocate limited sampling budgets to maximize expensive-to-evaluate functions. One-step-lookahead policies are often used, but computing optimal multi-step-lookahead policies remains a challenge. We consider a specialized Bayesian optimization problem: finding the superlevel set of an expensive one-dimensional function, with a Markov process prior. We compute the Bayes-optimal sampling policy efficiently, and characterize the suboptimality of one-step lookahead.

1 Introduction

We consider the problem of adaptively allocating sampling effort to efficiently estimate sub- and super-level sets of a one-dimensional Markov process, or more general additive functionals of this process. We use a decomposition property to show how the optimal procedure may be computed efficiently, circumventing the curse of dimensionality. We then use our ability to compute the optimal policy to study the suboptimality gap of commonly used one-step lookahead procedures in this problem.

This problem arises in the large and rapidly growing literature on Bayesian optimization [19, 22, 18, 11, 30], which seeks to develop adaptive sampling algorithms that estimate functionals, especially the location of a global maximum, of some underlying and unknown function in a query efficient way. Such problems arise when optimizing an objective that is computed via a long-running computer code [11, 30] or some other expensive process [4, 14] that severely limits the number of times it may be sampled. This literature places a Bayesian prior distribution on the underlying function, and views it is a stochastic process, most frequently as a Gaussian process.

In such problems, a Bayes-optimal algorithm is one that minimizes the expected loss under the prior suffered from mis-estimation of the underlying functional of interest, where the cost of sampling is either factored directly into the objective (as considered by [10, 3]), or a sampling budget is enforced as a constraint (as considered by [16, 12]). When optimization is goal, this loss function is the opportunity cost — the

difference in value between the point that is believed to be the best, and the value of the true global optimum — but when other functionals are of interest another loss function may be appropriate.

In principle, a Bayes-optimal algorithm may be computed using stochastic dynamic programming by understanding that this problem is a partially observable Markov decision process (POMDP) [16]. However, the curse of dimensionality [26] prevents actually computing the solution through brute-force approaches.

Thus, almost all of the literature has focused on approximate schemes, which in many cases are inspired by this view of the problem as a partially observable Markov decision process, but that do not actually solve the POMDP. The most common of these are the expected improvement method [22, 18] and the knowledge-gradient method [13, 29], which use one-step lookahead approaches, based on different assumptions about what points are eligible for selection once sampling stops [14]. Two-step lookahead approaches have also been implemented computationally in [2, 16].

In contrast, we focus on calculating the Bayes-optimal algorithm, and show that it can be computed efficiently in Bayesian optimization problems that satisfy three assumptions:

- the underlying function should be one-dimensional, as considered by [19, 6, 7, 9, 8, 24, 20].
- the Bayesian prior on this function should have the Markov property (e.g., a Wiener process prior, as used by [19, 24, 1, 35, 27, 21, 20, 9], or an Ornstein-Uhlenbeck prior [23, 25];
- the loss function is additive across location, as arises when the goal is to determine feasibility of points, as in [15], or to determine the set of points that are better than some known standard, as in [34].
- the limit on sampling is imposed as an additive cost in the objective, as in [10, 3], or as a constraint on the expected number of samples taken, as in [16, 12].

Additionally, we note that our ability to compute the optimal policy under the setting described above also provides an upper bound on the value of the Bayes optimal policy when limit on sampling is imposed as an almost sure constraint on the number of samples taken.

While one dimensional feasibility determination problems do arise in practice, and we expect that the optimal policy can provide a great deal of value in those settings, a large fraction of practical Bayesian optimization problems violate one or more of the assumptions above, because many problems are in more than one dimension, and because non-Markov Gaussian processes are often used as priors [28, 4]. Optimization is also a more common goal than super-level set determination, but we feel that in practice, it is often just as useful to provide a set of points that perform well, i.e., that reside in some super-level set, from which a final decision can be selected based on other criteria.

Thus, we view our primary contribution as providing a specialized but nevertheless rich class of Bayesian optimization problems on which the performance of widely applicable heuristic procedures, such as the one-step lookahead procedures described above, may be studied relative to Bayes-optimal procedures. This guide algorithm development — if a heuristic procedure performs close to optimal on a set of problems, then this is suggestive that further improvement is not necessary even for other similar problems for which the optimality gap cannot be evaluated. In contrast, if all known heuristic procedures perform substantially worse than optimal on a set of problems, then this suggests that further algorithm development is worthwhile for these and similar problems.

There is some complementary theoretical analysis of Bayes-optimal procedures for this and related problems in the literature. Much of it focuses on asymptotic analyses, and includes proofs of consistency for the Efficient Global Optimization (EGO) [32] and P algorithms [6], as well as convergence rates for these and closely related algorithms [7, 5, 9]. In terms of finite-time analyses, [31, 17] provide regret bounds for the closely related problem of Bayesian optimization in the bandit setting, but while these bounds characterize performance, slack in the bounds' constants creates a potentially large multiplicative gap in which performance may lie. In the problems of multiple comparisons with a known standard and stochastic root-finding, procedures for computing explicit Bayes optimal procedures have been developed [34, 33], but these problems are only distantly related to Bayesian optimization. Thus, exact performance of optimal finite-time procedures has remained unknown in Bayesian optimization.

Below, in Section 2, we provide a formal description of the problem. Our main results are in Section 3, where we significantly reduce the state-space for a dynamic program giving rise to a Bayes-optimal policy. In Section 4 we consider the relationship between the cost-per-sample setting and the constrained-budget setting, showing how an optimal policy for the former can be used to compute the optimal value of the latter. In Section 5 we present numerical results, illustrating the behavior of the optimal policy and using it to analyze the optimality gap for a one-step lookahead procedure. Finally, in Section 6, we conclude.

2 Problem Description

Let $Y = (Y(x) : x \geq 0)$ be a Markov process over the positive real line, and let $[a, b]$ be a given interval, $0 < a < b < \infty$. We consider adaptive sampling policies that characterize Y over $[a, b]$.

We will consider histories of the form $\{(x_t, Y(x_t)) : t = 1, \ldots, T\}$ for some sequence of adaptively chosen points $(x_t : t = 1, \ldots, T)$ at which measurements occur.

Let $\mathcal{H} = \cup_{T=0}^{\infty}(\mathbb{R}_+ \times \mathbb{R})^T$ be the space of all possible histories. A policy $\pi : \mathcal{H} \mapsto \mathbb{R}_+ \cup \{\Delta\}$ is a measurable function that maps the current history to either a point to be sampled next, or to the symbol

Δ, which indicates the decision to stop. We let Π indicate the space of all such policies.

We begin with an initial history $H_0 \in \mathcal{H}$. For simplicity of analysis we assume that H_0 contains endpoint observations, that is $(a, Y(a)), (b, Y(b)) \in H_0$, but our results can be extended to the case where it does not. We define histories H_t and decisions x_t recursively, letting $x_{t+1} = \pi(H_t)$ and letting

$$H_{t+1} \coloneqq \begin{cases} H_t \cup \{(x_{t+1}, Y(x_{t+1})\}, & \text{if } x_{t+1} \neq \Delta, \\ H_t, & \text{if } x_{t+1} = \Delta, \end{cases} \tag{2.1}$$

so that the point sampled and the resulting observation of Y is added to the history if the policy chooses to sample, and the history remains unchanged once the policy chooses to stop sampling.

We define $\tau = \inf\{t \geq 0 : \pi(H_t) = \triangle\}$ to be the total number of samples taken by a policy. When necessary, we will write τ^π to emphasize the policy on which τ depends.

We also define $\mathbb{P}^\pi$ to be the distribution over histories with respect to the randomness in Y and the decisions made by π, for any $\pi \in \Pi$. We let $\mathbb{E}^\pi$ denote the expectation with respect to this distribution.

We seek to characterize Y by assigning each point $x \in [a, b]$ a label, or class, based on our knowledge of $Y(x)$. Suppose there are $n < \infty$ classes to which a point might belong and let I be an index set such that each element corresponds to one class. At time τ, we will use the information collected, encoded in H_τ, to classify each point in $[a, b]$. Based on H_τ we construct a partition $\{B_i : i \in I\}$ of $[a, b]$, such that each B_i is a measurable subset of $[a, b]$. If $x \in B_i$, we say that x belongs to the ith class.

We also choose measurable functions $f_i : \mathbb{R} \to \mathbb{R}$ for each $i \in I$. The function f_i is meant to reward or penalize the classification $x \in B_i$ given the true value $Y(x)$. In addition to requiring the f_i be measurable, we also require that for each $i \in I$, at least one of the following properties hold:

1. f_i is bounded below.
2. The following inequality holds: $\int_{[a,b]} \mathbb{E}\left[|f_i \circ Y(x)|\right] dx < \infty$.

These properties will allow us below to interchange the integral over the domain of Y and the expectation over the randomness in Y, by Tonelli's theorem if property 1 is satisfied or Fubini's theorem if property 2 is satisfied.

Now, fix any partition $\mathbb{B} = \{B_i : i \in I\}$ and $H \in \mathcal{H}$. We define the expected reward of $\mathbb{B}$ given $H_\tau = H$ over $[a, b]$ to be

$$R_{[a,b]}(H, \mathbb{B}) = \mathbb{E}\left[\sum_{i \in I} \int_{B_i} f_i \circ Y(x) dx \mid H\right] = \sum_{i \in I} \int_{B_i} \mathbb{E}\left[f_i \circ Y(x) \mid H\right] dx, \tag{2.2}$$

where the last equality holds due to property 1 and Tonelli's theorem or property 2 and Fubini's theorem. Observing (2.2), a partition maximizing $R_{[a,b]}(H, \mathbb{B})$ is any $\mathbb{B}^* = \{B_i^* : i \in I\}$ such that for all $j \in I$, if $x \in B_j$, then

$$j \in \operatorname*{argmax}_i \mathbb{E}\left[f_i \circ Y(x) \mid H\right].$$

That is, x belongs to any class j maximizing $\mathbb{E}\left[f_j \circ Y(x) \mid H\right]$.

We define the expected reward $R_{[a,b]}(H)$ for any $H \in \mathcal{H}$ to be the expected reward of any optimal partition given H. That is,

$$R_{[a,b]}(H) = R_{[a,b]}(H, \mathbb{B}^*) = \int_{[a,b]} \max_i \mathbb{E}\left[f_i \circ W(x) \mid H\right] dx. \tag{2.3}$$

Recall the superlevel set of a function $g : [a, b] \to \mathbb{R}$ with respect to the threshold k is the set $\{x \in [a, b] : g(x) \geq k\}$. In this context we use the index set $I = \{+, -\}$ corresponding to the classification of a point as above or below the threshold. We give two reasonable choices for the functions f_+ and f_-:

1. $f_+(y) = \mathbb{1}\{y \geq k\}$ and $f_-(y) = \mathbb{1}\{x \leq k\}$. These functions clearly satisfy property 1.
2. $f_+(y) = y - k$ and $f_-(y) = k - y$. We only consider these reward functions for Markov processes Y such that property 2 is satisfied.

Although we focus on superlevel set detection, this framework can be used to classify points based on other properties of $Y(x)$. For example, we could consider two thresholds for the range of $Y(x)$, and classify each point as being below both, above both, or in between them.

Finally, the performance of a policy π at state H over $[a, b]$ is the expected value of the final reward less the cost associated with the expected number of samples starting from an initial history H. We define the value of a state to be the supremum of the performance over all policies. That is,

$$V_{[a,b]}(H) = \sup_{\pi \in \Pi} \mathbb{E}^\pi \left[R_{[a,b]}(H_\tau) - c\tau \mid H\right]. \tag{2.4}$$

We call (2.4) the cost-per-sample setting. Below, in section 4, we consider two other related settings: constrained budget and expected constrained budget.

3 Main Results for Cost-Per-Sample Case

In order tocompute a Bayes-optimal policy, we focus on efficiently computing the value function defined in (2.4). Naive dynamic programming can be used, but the state space grows exponentially in the number of

observations. Our main result decomposes the value function, showing that it is completely determined by its value on some 4-dimensional set, leading to its computation as a tractable dynamic program. In particular, we prove the following:

Theorem 1. *Fix interval $[a,b]$ and let $H \in \mathcal{H}$ be such that observations of a and b are included. Let $x_1, \ldots, x_{t+2}$ be the observations in H contained within $[a,b]$. Suppose they are ordered such that $x_i < x_{i+1}$ for all $1 \leq i < t+2$, so $x_1 = a$ and $x_{t+2} = b$. Define $H_i = \{(x_i, y_i), (x_{i+1}, y_{i+1})\}$ for each $1 \leq i < t+2$. Then*

$$V_{[a,b]}(H) = \sum_{i=1}^{t+1} V_{[x_i, x_{i+1}]}(H_i). \tag{3.1}$$

The importance of this theorem is that $V_{[a,b]}(H)$ is completely determined by its values on $\{H \in \mathcal{H} : |H| = 2\}$, greatly reducing the relevant dynamic program's state space.

We can further reduce the state space when Y satisfies additional structure. For any $\ell \in \mathbb{R}$, define the shift operator $T_\ell : \mathbb{R} \to \mathbb{R}$ by $T_\ell(x) = x + \ell$. We will apply T_ℓ to elements of $\mathcal{H}$, and adopt the convention that $T_\ell(H) = \{(x + \ell, y) : (x, y) \in H\}$, i.e. T_ℓ only translates the location of the observations in H, and not their values. We say the Markov process Y is *translation invariant* if, for any $H \in \mathcal{H}$ and $y \in \mathbb{R}$,

$$\mathbb{P}(Y(x) \in dy \mid H) = \mathbb{P}(Y(x + \ell) \in dy \mid T_\ell(H)). \tag{3.2}$$

The following theorem establishes that if the Markov process is translation invariant, so is the value function.

Theorem 2. *Suppose Y is translation invariant. Fix interval $[a,b]$ and pick any $\ell \in \mathbb{R}$ such that $a + \ell \geq 0$. Pick any history $H \in \mathcal{H}$ and let $H' = T_\ell(H)$. Then*

$$V_{[a,b]}(H) = V_{[a',b']}(H') \tag{3.3}$$

where $a' = a + \ell$ and $b' = b + \ell$.

Thus when Y satisfies translation invariance, the value function is completely determined by its values on $\{H \in \mathcal{H} : |H| = 2, (0, y_0) \in H\}$. (The $(0, y_0) \in H$ condition is arbitrary, any constant in the domain of Y works). In this case, $V_{[a,b]}$ can be computed as the result of a 3-dimensional dynamic program.

To establish these theorems, we begin with two lemmas. The first says that if $H \in \mathcal{H}$ contains endpoint observations then the only points in H that affect $V_{[a,b]}(H)$ are those within $[a,b]$.

Lemma 3. *Let $H \in \mathcal{H}$ contain endpoing observations. Let $H^I = \{(x,y) \in H : x \in [a,b]\}$ denote the set of initial observations inside $[a,b]$. Then*

$$V_{[a,b]}(H) = V_{[a,b]}(H^I). \tag{3.4}$$

Proof. We first show $V_{[a,b]}(H) \geq V_{[a,b]}(H^I)$. Let $\sigma \in \Pi$. Let $H^O = H \setminus H^I$ denote the set of initial observations outside $[a,b]$. Define $\pi \in \Pi$ by $\pi(K) \coloneqq \sigma(K \setminus H^O)$ for all $K \in \mathcal{H}$.

Consider the following Markov processes:

1. $(H_t^\pi)_{t\geq 0}$ with initial state $H_0 = H$.
2. $(H_t^{I,\sigma})_{t\geq 0}$ with initial state $H_0^I = H^I$.

Now, note that

$$R_{[a,b]}(H_\tau^\pi) = R_{[a,b]}(H_\tau^\pi \setminus H^O) \approx R_{[a,b]}(H_\tau^{I,\sigma}) \tag{3.5}$$

where the first equality holds because Y is a Markov process and H contains endoint observations, and the second equality (in distribution) holds because $H_\tau^\pi \setminus H^O \approx H_\tau^{I,\sigma}$. Thus $\mathbb{E}^\pi \left[R_{[a,b]}(H_\tau^\pi) - c\tau \mid H\right] = \mathbb{E}^\sigma \left[R_{[a,b]}(H_\tau^{I,\sigma}) - c\tau \mid H^I\right]$, and the inequality $V_{[a,b]}(H) \geq V_{[a,b]}(H^I)$ is established.

To establish the reverse inequality, we apply the same logic as above. Let $\pi \in \Pi$ and define $\sigma \in \Pi$ by $\sigma(K) = \pi(K \cup H^O)$. Then the Markov processes $(H_t^\pi)_{t\geq 0}$ and $(H_t^{I,\sigma})_{t\geq 0}$ with initial states H and H^I, respectively, share the same properties as before, but since σ is now constructed based on π we conclude $V_{[a,b]}(H^I) \geq V_{[a,b]}(H)$.

□

For the rest of this section we adopt the notation that, for $H \in \mathcal{H}$ and $A \subseteq \mathbb{R}$, $H \cap A = \{(x,y) \in H : x \in A\}$. We will also write $x \in H$ to mean there exists $y \in \mathbb{R}$ such that $(x,y) \in H$. The above lemma tells us that, if the range of π is contained in $[a,b]$, then $\pi(H) = \pi(H \cap [a,b])$ for all $H \in \mathcal{H}$.

Our second lemma constructs a subset of Π containing an optimal policy.

Lemma 4. *Fix interval $[a,b]$. Let*

- $\Pi^1 = \{\pi \in \Pi : \mathbb{P}^\pi(\tau < \infty \mid H) = 1 \ \forall H \in \mathcal{H}\}$ *be the set of policies that almost surely take finitely many samples.*
- $\Pi^2_{[a,b]} = \{\pi \in \Pi : \pi(H) \in [a,b] \ \forall H \in \mathcal{H}\}$ *be the set of policies that only take samples in $[a,b]$.*
- $\Pi^3 = \{\pi \in \Pi : \pi(H) \neq x \text{ if } x \in H \ \forall H \in \mathcal{H}\}$ *be the set of policies that do not sample the same point twice.*

Let $\bar{\Pi}_{[a,b]} = \Pi^1 \cap \Pi^2_{[a,b]} \cap \Pi^3$. For all $H \in \mathcal{H}$, define

$$\bar{V}_{[a,b]}(H) = \sup_{\pi \in \bar{\Pi}_{[a,b]}} \mathbb{E}^\pi \left[R_{[a,b]}(H_\tau) - c\tau \mid H\right]. \tag{3.6}$$

Then $V_{[a,b]}(H) = \bar{V}_{[a,b]}(H)$ for all $H \in \mathcal{H}$.

Proof. Fix $H_0 \in \mathcal{H}$. Since $\bar{\Pi}_{[a,b]} \subseteq \Pi$ it follows that $\bar{V}_{[a,b]}(H_0) \leq V_{[a,b]}(H_0)$. Let $\pi \in \Pi$. Define $\mathcal{I} = \{H \in \mathcal{H} : \mathbb{P}^\pi(\tau < \infty \mid H) \neq 1\}$. Suppose $\mathcal{I} \neq \emptyset$ (equivalently, $\pi \notin \Pi^1$). We consider two cases:

Case 1 Suppose, for some $t > 0$, $\mathbb{P}^\pi(H_t \in \mathcal{I} \mid H_0) \neq 0$. Then

$$\begin{aligned}\mathbb{E}^\pi\left[R_{[a,b]}(H_\tau) - c\tau \mid H_0\right] &= \mathbb{E}^\pi\left[R_{[a,b]}(H_\tau) - c\tau \mid H_\tau \in \mathcal{I}\right]\mathbb{P}^\pi(H_t \in \mathcal{I} \mid H_0) + C \\ &= \left(\mathbb{E}^\pi[R_{[a,b]}(H_\tau) \mid H_t \in \mathcal{I}] - c\mathbb{E}^\pi[\tau \mid H_t \in \mathcal{I}]\right)\mathbb{P}^\pi(H_t \in \mathcal{I} \mid H_0) + C\end{aligned}$$

where C is a finite constant. Note $\mathbb{E}^\pi[R_{[a,b]}(H_\tau) \mid H_t \in \mathcal{I}]$ is bounded[1] because even with perfect information, we only earn a finite reward as can be seen in equation (2.2). However,

$$\mathbb{E}^\pi[\tau | H_t \in \mathcal{I}] = (-\infty)\mathbb{P}^\pi(\tau = \infty | H_t \in \mathcal{I}) + D = -\infty$$

where D is some finite constant. Hence $\mathbb{E}^\pi\left[R_{[a,b]}(H_\tau) - c\tau \mid H_0\right] = -\infty$. Thus, since every policy in Π^1 has finite performance, every policy in Π^1 has greater performance than π.

Case 2 Suppose $\mathbb{P}^\pi(H_t \in \mathcal{I} \mid H_0) = 0$ Define $\bar{\pi} \in \Pi^1$ by $\bar{\pi}(H) = \pi(H)$ if $H \notin \mathcal{I}$ and $\bar{\pi}(H) = \Delta$ otherwise. Then π and $\bar{\pi}$ have the same performance under initial history H_0, and $\bar{\pi} \in \Pi^1$.

The above cases show that, for every $\pi \in \Pi$, there is some $\sigma \in \Pi^1$ of greater or equal performance. Thus Π^1 contains an optimal policy.

Note that $\Pi^1 \subseteq \Pi^3$. Indeed, suppose $\pi \in \Pi$ such that there exists some $H_t \in \mathcal{H}$ such that $\pi(H_t) \in H_t$. Write $x = \pi(H_t)$. Then $H_{t+1} = H_t \cup \{(x, Y(x))\} = H_t$. It follows that $H_{t+k} = H_t$ for all $k \in \mathbb{N}$, and π never chooses to stop sampling at x. Thus $\pi \notin \Pi^3 \implies \pi \notin \Pi^1$, and $\Pi^1 \subseteq \Pi^3$ is established. It follows that $\Pi^1 \cap \Pi^3$ contains an optimal policy.

Now, let $\pi \in \Pi^1 \cap \Pi^3$ and let $\mathcal{J} = \{H \in \mathcal{H} : \pi(H) \notin [a,b]\}$. Suppose $\mathcal{J} \neq \emptyset$ (equivalently, $\pi \notin \Pi^2_{[a,b]}$). For each $H \in \mathcal{H}$, let $x_1, \ldots, x_{n_H}$ denote the sequence of points π samples until it chooses to stop sampling, or it samples inside $[a,b]$. That is, if $\pi(H) \in [a,b]$ then n_H=1. If π never samples inside $[a,b]$ after H, then $x_{n_H} = \Delta$. Define $\sigma(H) = x_{n_H}$. It follows that H^σ_τ with initial state H_0 is equal in distribution to $H^\pi_\tau \cap [a,b]$ also with initial state H_0. Thus $\mathbb{E}^\sigma[R_{[a,b]}(H_\tau) \mid H_0] = \mathbb{E}^\pi[R_{[a,b]}(H_\tau) \mid H_0]$. However, $\mathbb{E}^\sigma[\tau \mid H_0] \leq \mathbb{E}^\pi[\tau \mid H_0]$. Thus, the performance of σ is equal or greater to the performance of π. Hence there is an optimal policy contained in $\Pi^1 \cap \Pi^2_{[a,b]} \cap \Pi^3$ and the result is established.

□

We are now in a position to prove the main decomposition theorem, stated earlier:

[1]Note that we gloss over the precise definition of $R_{[a,b]}(H_\tau)$ when $\tau = \infty$

Theorem 1. *Fix interval $[a,b]$ and let $H \in \mathcal{H}$ be such that observations of a and b are included. Let $x_1, \ldots, x_{t+2}$ be the observations in H contained within $[a,b]$. Suppose they are ordered such that $x_i < x_{i+1}$ for all $1 \leq i < t+2$, so $x_1 = a$ and $x_{t+2} = b$. Define $H_i = \{(x_i, y_i), (x_{i+1}, y_{i+1})\}$ for each $1 \leq i < t+2$. Then*

$$V_{[a,b]}(H) = \sum_{i=1}^{t+1} V_{[x_i, x_{i+1}]}(H_i). \tag{3.1}$$

Proof. We proceed by induction on t. When $t = 0$ the summation contains only one term and the result is established. Fix $t > 0$ and suppose the decomposition (3.1) holds for any $|H| < t+2$.

Define τ_A with respect to any policy π to be the number of points π chooses to sample inside the set A, for some $A \subseteq [a,b]$. Thus if $\{x_i : 1 \leq i \leq \tau\}$ is the set of points sampled by π, $\tau_A = \sum_{i=1}^{\tau} \mathbb{1}\{x_i \in A\}$. By Lemma 4 we restrict our attention to $\pi \in \bar{\Pi}_{[a,b]}$. In particular if $(x, Y(x)) \in K$, π will not choose to sample at x again given initial state K. Thus $\tau_{[a,b]} = \tau_{[a,x]} + \tau_{[x,b]}$ conditioned on any initial history containing $(x, Y(x))$. This is becaues the only point in $[a,x] \cap [x,b]$ is $\{x\}$, and the lone sample of $\{x\}$ is in the initial history, it is not counted in $\tau_{[a,x]}$ or $\tau_{[x,b]}$. From the Markov property, it is also clear that $R_{[a,b]}(H_\tau) = R_{[a,x]}(H_\tau) + R_{[x,b]}(H_\tau)$ conditioned on any initial history containing x.

Now, fix some $1 < i < t+2$, so that x_i is not a or b. Note that

$$\begin{aligned}
V_{[a,b]}(H) &= \sup_{\pi \in \bar{\Pi}_{[a,b]}} \mathbb{E}^\pi \left[R_{[a,b]}(H_\tau) - c\tau_{[a,b]} \mid H \right] && (3.7)\\
&= \sup_{\pi \in \bar{\Pi}_{[a,b]}} \left(\mathbb{E}^\pi \left[R_{[a,x_i]}(H_\tau) - c\tau_{[a,x_i]} \mid H \right] + \mathbb{E}^\pi \left[R_{[x_i,b]}(H_\tau) - c\tau_{[x_i,b]} \mid H \right] \right) && (3.8)\\
&\leq \sup_{\pi \in \bar{\Pi}_{[a,b]}} \mathbb{E}^\pi \left[R_{[a,x_i]}(H_\tau) - c\tau_{[a,x_i]} \mid H \right] + \sup_{\sigma \in \bar{\Pi}_{[a,b]}} \mathbb{E}^\sigma \left[R_{[x_i,b]}(H_\tau) - c\tau_{[x_i,b]} \mid H \right] && (3.9)\\
&= \sup_{\pi \in \bar{\Pi}_{[a,x_i]}} \mathbb{E}^\pi \left[R_{[a,x_i]}(H_\tau) - c\tau_{[a,x_i]} \mid H \right] + \sup_{\sigma \in \bar{\Pi}_{[x_i,b]}} \mathbb{E}^\sigma \left[R_{[x_i,b]}(H_\tau) - c\tau_{[x_i,b]} \mid H \right] && (3.10)\\
&= V_{[a,x_i]}(H) + V_{[x_i,b]}(H). && (3.11)
\end{aligned}$$

The equality between (3.7) and (3.8) holds because $x_i \in H$. The equality between (3.9) and (3.10) holds because $\bar{\Pi}_{[a,x_i]} \subseteq \bar{\Pi}_{[a,b]}$ and Lemma 4 shows the supremum is achieved in $\bar{\Pi}_{[a,x_i]}$ and similarly for $\bar{\Pi}_{[x_i,b]}$. The equality between (3.10) and (3.11) holds because $\tau^\pi_{[a,x_i]} = \tau^\pi$ for any $\pi \in \bar{\Pi}_{[a,x_i]}$.

We now show that $V_{[a,b]}(H) \geq V_{[a,x_i]}(H) + V_{[x_i,b]}(H)$. Let $\pi \in \bar{\Pi}_{[a,x_i]}$ and $\sigma \in \bar{\Pi}_{[x_i,b]}$. Define the policy γ by

$$\gamma(H) = \begin{cases} \pi(H \cap [a,x_i]), & \text{if } \pi(H \cap [a,x_i]) \neq \triangle, \\ \sigma(H \cap [x_i,b]), & \text{otherwise.} \end{cases} \tag{3.12}$$

That is, γ is the policy that executes π with input from $[a,x_i]$ until π chooses to stop sampling, and then executes σ with input from $[x_i,b]$ until σ chooses to stop sampling. Since $\pi \in \bar{\Pi}_{[a,x_i]}$ and $\sigma \in \bar{\Pi}_{[x_i,b]}$ we

know τ^π and τ^σ are almost surely finite, so γ will fully execute both π and σ. Observe the expectated of the performance under γ is

$$\begin{aligned}\mathbb{E}^\gamma\left[R_{[a,b]}(H_\tau^\gamma) - c\tau_{[a,b]} \mid H\right] &= \mathbb{E}^\gamma\left[R_{[a,x_i]}(H_\tau^\gamma) - c\tau_{[a,x_i]} \mid H\right] + \mathbb{E}^\gamma\left[R_{[x_i,b]}(H_\tau^\gamma) - c\tau_{[x_i,b]} \mid H\right] \quad (3.13)\\ &= \mathbb{E}^\pi\left[R_{[a,x_i]}(H_\tau^\pi) - c\tau_{[a,x_i]} \mid H\right] + \mathbb{E}^\sigma\left[R_{[x_i,b]}(H_\tau^\sigma) - c\tau_{[x_i,b]} \mid H\right]. \quad (3.14)\end{aligned}$$

where the decomposition $\tau_{[a,b]} = \tau_{[a,x_i]} + \tau_{[x_i,b]}$ holds under γ because $(x_i, Y(x_i)) \in H$ and so γ never samples x_i. Thus $V_{[a,b]}(H) \geq V_{[a,x_i]}(H) + V_{[x_i,b]}(H)$. As we have already established the reverse inequality, it follows that $V_{[a,b]}(H) = V_{[a,x_i]}(H) + V_{[x_i,b]}(H)$.

Now, we partition H about x_i: Let $H_{\leq i} = \{(x,w) \in H : x \leq x_i\}$ and $H_{\geq i} = \{(x,w) \in H : x \geq x_i\}$. By Lemma 3, $V_{[a,x_i]}(H) = V_{[a,x_i]}(H_{\leq i})$ and $V_{[x_i,b]}(H) = V_{[x_i,b]}(H_{\geq i})$. Hence $V_{[a,b]}(H) = V_{[a,x_i]}(H_{\leq i}) + V_{[x_i,b]}(H_{\geq i})$. However, since x_i was chosen to not be an endpoint, $|H_{\leq i}| < t+2$ and $|H_{\geq i}| < t+2$. Thus by the induction hypothesis,

$$\begin{aligned}V_{[a,b]}(H) &= \sum_{j=1}^{i-1} V_{[x_j,x_{j+1}]}(H_j) + \sum_{j=i}^{t} V_{[x_j,x_{j+1}]}(H_j) \quad (3.15)\\ &= \sum_{j=1}^{t} V_{[x_j,x_{j+1}]}(H_j), \quad (3.16)\end{aligned}$$

and the induction holds. □

As mentioned above, the importance of the above theorem is that the value function is completely determined by its values on $\{H \in \mathcal{H} : |H| = 2\}$. The value function over this set can be computed in a computationally tractable way using a four-dimensional dynamic program. This allows computing an optimal policy in a computationally efficient way.

We now prove the theorem further reducing the state space in the presence of translation invariance. Recall Theorem 2:

Theorem 2. *Suppose Y is translation invariant. Fix interval $[a,b]$ and pick any $\ell \in \mathbb{R}$ such that $a + \ell \geq 0$. Pick any history $H \in \mathcal{H}$ and let $H' = T_\ell(H)$. Then*

$$V_{[a,b]}(H) = V_{[a',b']}(H') \quad (3.3)$$

where $a' = a + \ell$ and $b' = b + \ell$.

Proof. By Lemma 3 we assume all observations in H are contained in $[a,b]$, i.e. $x \in [a,b]$ for all $(x,y) \in H$.

We show that for any policy $\pi \in \bar{\Pi}_{[a,b]}$ there exists a policy $\sigma \in \bar{\Pi}_{[a,b]}$ on $[a', b']$ such that $\mathbb{E}^{\pi}\left[R_{[a,b]} - c\tau | H\right] = \mathbb{E}^{\sigma}\left[R_{[a',b']} - c\tau | H'\right]$.

Fix any policy $\pi \in \bar{\Pi}_{[a,b]}$. Define $\sigma \in \bar{\Pi}_{[a',b']}$ by $\sigma(K) \coloneqq T_{\ell} \circ \pi \circ T_{-\ell}(K \cap [a', b'])$ for every $K \in \mathcal{H}$. We use the intersection $K \cap [a', b']$ so that all observations live at or above 0, i.e. $T_{-\ell}(K \cap [a', b']) \in \mathcal{H}$.

Now, consider the two Markov processes:

1. $(H_t)_{t \geq 0}$ under π with initial state $H_0 = H$.
2. $\left(T_{-\ell}\left(H'_t\right)\right)_{t \geq 0}$ under σ with initial state $T_{-\ell}(H'_0) = T_{-\ell}(H')$.

In (2), we apply the shift operator $T_{-\ell}$ to H'_t so that the two Markov Processes have the same initial state.

We now show the two Markov processes have the same transition kernel. Suppose $T_{-\ell}(H'_t) = H_t$ for some $t \geq 0$, that is, both Markov Processes are in the same state at time t. Note that π and $T_{-\ell} \circ \sigma$ choose to sample the same point:

$$T_{-\ell} \circ \sigma(H'_t) = T_{-\ell} \circ T_{\ell} \circ \pi \circ T_{-\ell}(T_{\ell}(H_t) \cap [a', b']) = \pi(H_t \cap [a, b]) = \pi(H_t)$$

where the final equality holds because H has all observations contained in $[a, b]$ and $\pi \in \bar{\Pi}_{[a,b]}$, so H_t must be contained in $[a, b]$ for all t. Call this point x_{t+1}. It follows that every state K with nonzero probability for the $t+1$th time of both Markov Processes is of the form $K = H_t \cup \{(x_{t+1}, y)\}$ for some $y \in \mathbb{R}$. Then,

$$\begin{aligned}
\mathbb{P}^{\pi}\left(H_{t+1} = K | H_t\right) &= \mathbb{P}\left(Y(x_{t+1}) \in dy | H_t\right) && (3.17)\\
&= \mathbb{P}\left(Y(x_{t+1}) \in dy | T_{-\ell}(H'_t)\right) && (3.18)\\
&= \mathbb{P}^{\sigma}\left(T_{-\ell}(H'_t) = K | T_{-\ell}(H'_t)\right). && (3.19)
\end{aligned}$$

Hence the two Markov Processes have the same transition kernel, and since they have the same initial state, it follows they have the same distribution.

A simple consequence of this is that τ under π and τ under σ are identically distributed. Indeed, $\tau_{\pi} \sim |H_{\tau}| - |H_0| \sim |H'_{\tau}| - |H'_0| \sim \tau_{\sigma}$.

Finally, observe that the reward is translation invariant: $R_{[a,b]}(K) = R_{[a',b']}(T_{\ell}(K))$ for any $K \in \mathcal{H}$. Thus,

$$\begin{aligned}
\mathbb{E}^{\pi}\left[R_{[a,b]}(H_{\tau}) \mid H_0\right] &= \mathbb{E}^{\pi}\left[R_{[a',b']}(T_{\ell}(H_{\tau}) \mid H_0\right] && (3.20)\\
&= \mathbb{E}^{\pi}\left[R_{[a',b']}(T_{\ell}(H_{\tau})) \mid T_{\ell}(H_0)\right] && (3.21)\\
&= \mathbb{E}^{\sigma}\left[R_{[a',b']}(H'_{\tau}) \mid H'_0\right] && (3.22)
\end{aligned}$$

where the equality between (3.20) and (3.21) holds because T_ℓ is a bijection, and equality between (3.21) and (3.22) holds because $T_\ell(H_\tau) \mid T_\ell(H_0)$ under π is equal in distribution to $H'_\tau \mid H'_0$ under σ.

Thus $\mathbb{E}^\pi\left[R_{[a,b]} - c\tau | H\right] = \mathbb{E}^\sigma\left[R_{[a',b']} - c\tau | H'\right]$, as we set out to show. It follows that $V_{[a,b]}(H) \leq V_{[a',b']}(H')$. Setting $a \coloneqq a + \ell$, $b \coloneqq b + \ell$ and $\ell \coloneqq -\ell$ establishes the reverse inequality, and equality follows. □

We conclude this section by showing how the value function can be used to construct an optimal policy. Fix $H \in \mathcal{H}$. For any $x \in [a, b]$, define the expected value of the value function $\mathrm{EV}(x, H)$ by:

$$\mathrm{EV}(x, H) \coloneqq \mathbb{E}\left[V_{[a,b]}(H \cup \{(x, Y(x))\}) \mid H\right]. \tag{3.23}$$

Let $x^* \in \mathrm{argmax}_{x\in[a,b]}\mathrm{EV}(x, H)$ be any point maximizing $\mathrm{EV}(x^*, H)$. Define $\pi \in \Pi$ by

$$\pi(H) = \begin{cases} x^*, & \text{if } \mathrm{EV}(x^*, H) - V_{[a,b]}(H) > c, \\ \Delta, & \text{otherwise.} \end{cases} \tag{3.24}$$

That is, π chooses to sample at x^* if the expected gain in the value is larger than the cost of sampling, otherwise π chooses to stop sampling.

We claim that π is an optimal policy. To see this, note that the Bellman equation takes the following form for our problem:

$$V_{[a,b]}(H) = \sup_{x\in[a,b]\cup\{\Delta\}} \mathbb{E}\left[V_{[a,b]}(H \cup \{(x, Y(x))\}) - cI_{\{x=\Delta\}} \mid H\right] \tag{3.25}$$

where we adopt the convention that $\{(x, Y(x))\} = \emptyset$ when $x = \Delta$. Since the supremum in the above is always achieved by the point $\pi(H)$, it follows π is optimal. Thus, efficient computation of the value function will lead to a computationally tractable optimal policy.

4 Upper Bound on Budget-Constrained Optimization

So far in this paper we have considered the cost-per-sample scenario, where the policy may choose how many samples to make without any additional constraints. In this section, we show how the cost-per-sample problem relates to optimizing over budget constraints, that is, when there is some constraint on how many samples the policy can take in total.

We first introduce the notion of a randomized policy, which will be useful in our subsequent analysis. Let $\Pi_R = \{\pi : [0, 1] \times \mathcal{H} \to \mathbb{R}_+ \cup \{\Delta\}\}$, that is, the set of policies which take an additional argument inside

$[0,1]$. For such policies, we adopt the convention that histories are still updated according to (2.1), with the slight modification that $x_{t+1} = \pi(U, H_t)$ where $U \sim \text{Uniform}([0,1])$. We call these randomized policies because they may take different actions depending on the random variable U. We will often write $\pi(H)$ instead of $\pi(U, H)$ when it is clear that π is randomized. Note that taking the supremum in equation (2.4) over Π_R instead of Π does not affect the optimal value, because the deterministic policy we construct based on Theorem 1 remains optimal.

Fix some constant $T > 0$ and interval $[a, b]$. We define the following sets of constrained policies:

$$\Pi_1 = \{\pi \in \Pi_R : \mathbb{E}^\pi[\tau] = T\} \qquad \text{and} \qquad \Pi_2 = \{\pi \in \Pi_R : \mathbb{P}^\pi(\tau = T) = 1\}. \tag{4.1}$$

We refer to Π_1 as the expected budget constrained policies and Π_2 as the budget constrained policies. We define the corresponding value functions as

$$V_1(H) = \sup_{\pi \in \Pi_1} \mathbb{E}^\pi\left[R_{[a,b]}(H_\tau) \mid H\right] \qquad \text{and} \qquad V_2(H) = \sup_{\pi \in \Pi_2} \mathbb{E}^\pi\left[R_{[a,b]}(H_\tau) \mid H\right]. \tag{4.2}$$

Such policy types are common in practice - it is easier to allocate a predetermined number of samples than to come up with a suitable cost, as the cost-per-sample case requires. Indeed, note the above are defined without a cost. This is because $\mathbb{E}^\pi[\tau] = T$ for any $\pi \in \Pi_1 \cup \Pi_2$, so any cost term would be constant and not affect the optimal solution.

For the rest of this section we will write the cost-per-sample value function, as defined in equation (2.4), as a function of both the state and the cost. That is, let $V(H, \lambda)$ mean $V_{[a,b]}(H)$ with a cost of λ. Now observe that for any λ,

$$\begin{aligned} V_1(H) &= \sup_{\pi \in \Pi_1} \mathbb{E}^\pi\left[R_{[a,b]}(H_\tau) - \lambda(\tau - T) \mid H\right] && (4.3)\\ &\leq \sup_{\pi \in \Pi} \mathbb{E}^\pi\left[R_{[a,b]}(H_\tau) - \lambda\tau \mid H\right] + \lambda T && (4.4)\\ &= V(H, \lambda) + \lambda T. && (4.5) \end{aligned}$$

Thus it follows that for any $H \in \mathcal{H}$,

$$V_2(H) \leq V_1(H) \leq \inf_\lambda V(H, \lambda) + \lambda T \tag{4.6}$$

where the first inequality holds because $\Pi_2 \subseteq \Pi_1$. The following theorem shows the second inequality is tight.

Theorem 5. *The second inequality in equation (4.6) is tight. That is,*

$$V_1(H) = \inf_{\lambda} V(H, \lambda) + \lambda T,$$

for all $H \in \mathcal{H}$.

Proof. We first instroduce some notation. Fix $H \in \mathcal{H}$. For any $\pi \in \Pi_R$ let $r(\pi) = \mathbb{E}^{\pi}[R_{[a,b]}(H_\tau) \mid H]$ and $t(\pi) = \mathbb{E}^{\pi}[\tau \mid H]$. For any $\lambda > 0$, let

$$\Pi^*(\lambda) = \left\{\pi \in \Pi_R : V_{[a,b]}(H, \lambda) = \mathbb{E}^{\pi}\left[R_{[a,b]}(H_\tau) - \lambda\tau \mid H\right] \ \forall H \in \mathcal{H}\right\}.$$

That is, $\Pi^*(\lambda)$ contains the set of all optimal randomized policies in the cost-per-sample setting with cost λ. Finally, let $s(\lambda) = \sup\{t(\pi) : \pi \in \Pi^*(\lambda)\}$ denote the maximum expected number of samples an optimal policy makes given cost λ. It is easy to note that $\lim_{\lambda\to\infty} s(\lambda) = 0$ and $\lim_{\lambda\to 0^+} s(\lambda) = \infty$.

We claim that $s(\lambda)$ is monotonically decreasing. Let $\lambda < \lambda'$ and pick any $\pi \in \Pi^*(\lambda)$ and $\pi' \in \Pi^*(\lambda')$. By the optimality of π with respect to λ and π' with respect to λ', the following inequalities hold:

$$r(\pi) - \lambda t(\pi) \geq r(\pi') - \lambda t(\pi') \tag{4.7}$$

$$r(\pi') - \lambda' t(\pi') \geq r(\pi) - \lambda' t(\pi). \tag{4.8}$$

Subtracting (4.8) from (4.7) it it follows that $(\lambda' - \lambda)t(\pi) \geq (\lambda' - \lambda)t(\pi')$, and since $\lambda' > \lambda$, $t(\pi) \geq t(\pi')$. Thus $s(\lambda)$ is monotonically decreasing.

Since $s(\lambda)$ is monotonically decreasing, approaches ∞ as $\lambda \to 0^+$, approaches 0 as $\lambda \to \infty$, and $T > 0$, it follows there is some $\lambda^* > 0$ such that, for $\lambda < \lambda^*$, $t(\pi) \geq T$ for all $\pi \in \Pi^*(\lambda)$, and for $\lambda > \lambda^*$, $t(\pi) \leq T$ for all $\pi \in \Pi^*(\lambda)$. Pick any sequence $(\overline{\lambda}_n, \underline{\lambda}_n)$ such that $(\overline{\lambda}_n)$ is decreasing in n, $(\underline{\lambda}_n)$ is increasing in n, and $\lim_{n\to\infty} \overline{\lambda}_n = \lambda^* = \lim_{n\to\infty} \underline{\lambda}_n$.

For each n, pick any $\overline{\pi}_n \in \Pi^*(\overline{\lambda}_n)$ and $\underline{\pi}_n \in \Pi^*(\underline{\lambda}_n)$. By the previous remarks, we know $t(\underline{\pi}_n) \geq T \geq t(\overline{\pi}_n)$ for all n. For each n, define the probability p_n by

$$p_n \coloneqq \begin{cases} \frac{T - t(\overline{\pi}_n)}{t(\underline{\pi}) - t(\overline{\pi})}, & \text{if } t(\underline{\pi}) \neq t(\overline{\pi}), \\ \frac{1}{2}, & \text{otherwise.} \end{cases} \tag{4.9}$$

Define the randomized policy $\pi'_n \in \Pi_R$ by

$$\pi'_n(H) \coloneqq \begin{cases} \overline{\pi}_n(H), & \text{with probability } 1-p_n \\ \underline{\pi}_n(H), & \text{with probability } p_n. \end{cases} \tag{4.10}$$

Note that $t(\pi'_n) = T$ for all n.

For technical purposes, we claim the following equality holds:

$$\liminf_n p_n \underline{\lambda}_n[t(\underline{\pi}_n) - T] + (1-p_n)\overline{\lambda}_n[t(\overline{\pi}_n) - T] = 0. \tag{4.11}$$

Let $L_n = p_n \underline{\lambda}_n[t(\underline{\pi}_n) - T] + (1-p_n)\overline{\lambda}_n[t(\overline{\pi}_n) - T]$. Then we can write

$$\begin{aligned} L_n &= p_n \overline{\lambda}_n[t(\underline{\pi}_n) - T] + (1-p_n)\overline{\lambda}_n[t(\overline{\pi}_n) - T] - p_n(\overline{\lambda}_n - \underline{\lambda}_n)[t(\underline{\pi}_n) - T] && (4.12) \\ &= \overline{\lambda}_n[t(\pi'_n) - T] - p_n(\overline{\lambda}_n - \underline{\lambda}_n)[t(\underline{\pi}_n) - T]. && (4.13) \end{aligned}$$

We know that $t(\pi'_n) = T$ for all n, so the first term above is precisely equal to 0. By choice of $(\overline{\lambda}_n)$ and $(\underline{\lambda}_n)$ we know $\liminf_n(\overline{\lambda}_n - \underline{\lambda}_n) = 0$. Since (p_n) and $(t(\underline{\pi}_n) - T)$ are bounded sequences in n, it follows that $\liminf_n p_n(\overline{\lambda}_n - \underline{\lambda}_n)[t(\underline{\pi}_n) - T] = 0$, and the claimed equality is established.

Now let $U = \inf_\lambda V(H,\lambda) + \lambda T$. For any n, we have

$$\begin{aligned} U &\leq p_n\left[V(H,\underline{\lambda}_n) + \underline{\lambda}_n T\right] + (1-p_n)\left[V(H,\overline{\lambda}_n) + \overline{\lambda}_n T\right] && (4.14) \\ &= p_n\left[r(\underline{\pi}_n) - \underline{\lambda}_n(t(\underline{\pi}_n) - T)\right] + (1-p_n)\left[r(\overline{\pi}_n) - \overline{\lambda}_n(t(\overline{\pi}_n) - T)\right] && (4.15) \\ &= r(\pi'_n) - p_n\underline{\lambda}_n[t(\underline{\pi}_n) - T] - (1-p_n)\overline{\lambda}_n[t(\overline{\pi}_n) - T] && (4.16) \\ &\leq \liminf_n r(\pi'_n) - p_n\underline{\lambda}_n[t(\underline{\pi}_n) - T] - (1-p_n)\overline{\lambda}_n[t(\overline{\pi}_n) - T] && (4.17) \\ &= \liminf_n r(\pi'_n). && (4.18) \end{aligned}$$

Additionally, since $t(\pi'_n) = T$ for every n, we know $V_1(H) \geq r(\pi'_n)$. In particular, $V_1(H) \geq \limsup_n r(\pi'_n)$. Combining this with (4.6) and (4.18), it follows

$$\limsup_n r(\pi'_n) \leq V_1(H) \leq U \leq \liminf_n r(\pi'_n).$$

Thus $V_1(H) = \lim_{n\to\infty} r(\pi'_n) = U$.

□

Finally, note that

$$V(H,\lambda) + \lambda T = \sup_{\pi \in \Pi} \mathbb{E}^{\pi} \left[R_{[a,b]}(H_{\tau} \mid H \right] - \lambda \mathbb{E}^{\pi} \left[\tau \mid H \right]. \tag{4.19}$$

Thus $V(H,\lambda)$ is a supremum over linear functions of λ, implying that it is a convex function of λ. Thus we can find its minimum, and hence compute $V_1(H)$, as the solution to a convex program.

5 Experimental Analysis

In this section we run simulations to better understand the behaviour of the optimal policy. We focus on superlevel set detection, and consider two choices for the Markov process Y. The first is a standard Brownian motion. The second is a compound Poisson process, that is

$$Y(t) = \sum_{i=0}^{N(t)} D_i$$

where $N(t)$ is a Poisson process with parameter μ and D_i are independent standard normal variables. We consider the interval $[a,b] = [0,1]$, the threshold $k = 0$ and assume we have observed endpoint observations $Y(0) = Y(1) = 0$. For the compound Poisson process we use a parameter of $\mu = 20$. The algorithm we use to solve the dynamic program is given in Algorithm 1. Both the standard Brownian motion and compound Poisson process satisfy translation invariance (as defined in equation (3.2)), so we are only concerned with the 3-dimensional dynamic program. In order to compute the optimal policy we are required to discretize the domain and range of Y, but the effect this has on the overall policy is small. For all of our experiments we use the indicator reward functions: $f_+(y) = \mathbb{1}\{y \geq k\}$ and $f_-(y) = \mathbb{1}\{x \leq k\}$.

Figure 5.1 depicts the behavior of the optimal policy defined in equation 3.24 for a Brownian motion over the interval $[0,1]$. The optimal policy exhibits several intuitive properties. For example, note that the difference between the expected value of sampling and the expected reward of not sampling is larger for the intervals where the endpoints are further away from the threshold.

One benefit of being able to compute an optimal policy is being able to characterize suboptimality of the common one-step lookahead heuristic. In this problem setup, the one-step lookahead heuristic policy samples the point maximizing the expected immediate reward, or chooses to stop sampling when the gain in reward is lower than the cost. We ran simulations for both this policy and the optimal policy, varying the cost c. We used both the compound Poisson process and Brownian motion prior on Y, again over $[0,1]$. To increase the variability in the problem, we assume the initial observations $Y(0) = Y(1) = 0$, and use a threshold of $k = 0$. The value of a policy is estimated running the policy under the above conditions

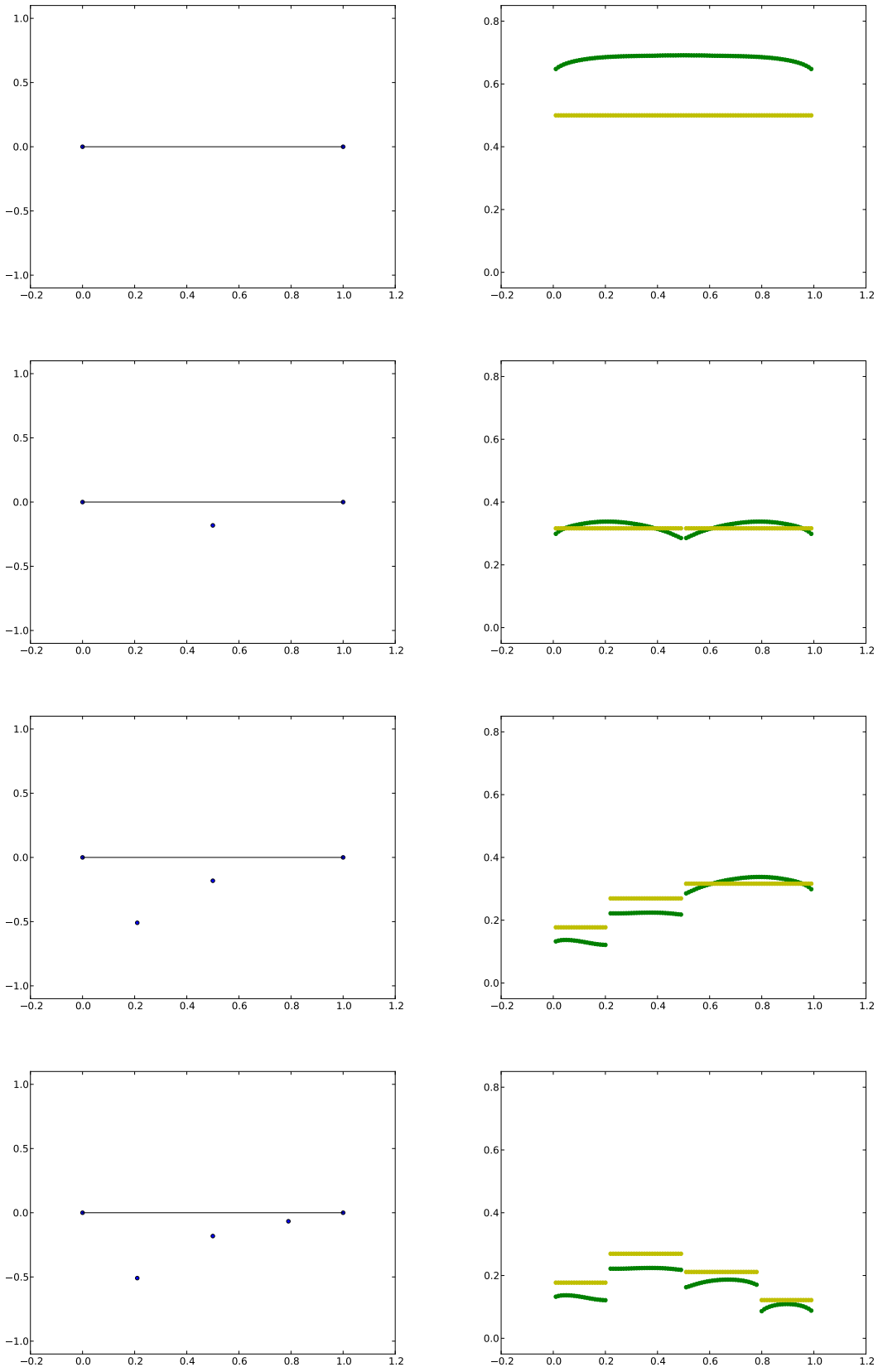


Figure 5.1: Depiction of an optimal policy. Here Y is a standard Brownian motion, set the threshold $k = 0$, x-axis discretization of 100, cost $c = 0.05$. **Left:** Sampled points at each iteration. **Right:** Expected value function (in green) plotted against expected reward (in yellow). The policy samples the point maximizing the difference between these quantities (assuming the difference is positive).

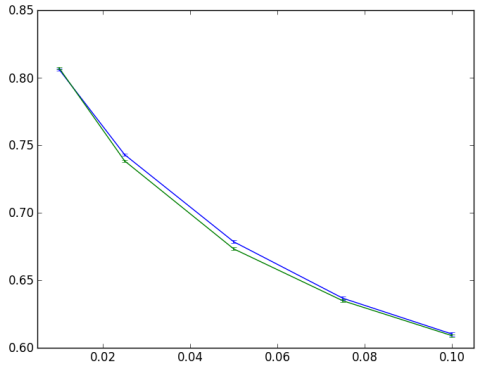
Algorithm 1 Algorithm for computing the value function. Note the computation on line 10 is possible, since $V[w_1, w_2, x'']$ will already be stored for all $w_1, w_2 \in \{y_1, \ldots, y_n\}$ and $x'' \in \{x_1, \ldots, x'\}$.

Require: Interval length ℓ, Y-range discretization $y_1, \ldots, y_n$, $[0, \ell]$-domain discretization $x_1, \ldots, x_m$.
Ensure: $x_1 = 0$ and $x_m = \ell$.

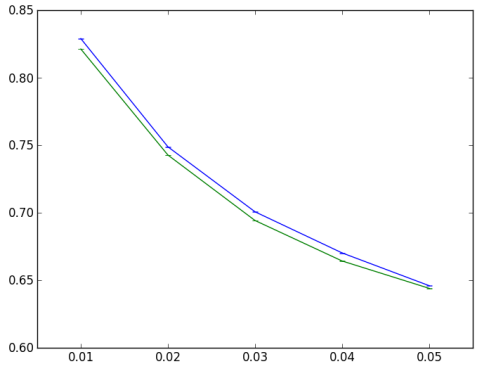
```
1: for y_L = y_1, ..., y_n do
2:   for y_R = y_1, ..., y_n do
3:     for x = x_1, ..., x_m do
4:       if x = 0 then
5:         V[y_L, y_R, x] ← 0
6:       else
7:         Let H = {(0, y_L), (ℓ, y_R)}
8:         T ← 𝔼[R_[0,ℓ](H) | H]
9:         for x' = x_1, ..., x do
10:          T' ← 𝔼[R_[0,ℓ](H ∪ {(x', Y(x'))}) | H] − c
11:          if T' > T then
12:            T ← T'
13:          end if
14:        end for
15:        V[y_L, y_R, x] ← T
16:      end if
17:    end for
18:  end for
19: end for
20: return V
```

(where we sample the observations of Y from the corresponding conditional distribution), and looking at the expected reward once the policy chooses to stop sampling. We obtain accurate estimates by running each policy a large number of times. The results are shown in Figure 5.2. While the one-step lookahead policy is clearly suboptimal, for each value of c we tried, the value of the one-step lookahead policy was within 98% of the optimal. This bodes well for the performance of more realistic one-step lookahead algorithms, which are common in practice.

Recall in Section 4 we defined $V_1(H)$ as the optimal value of a policy with expected budget constraints, and $V_2(H)$ as the optimal value of a policy with almost sure budget constraints. Also recall we proved $V_1(H)$ is equal to the solution of the convex optimization program (4.6). In Figure 5.3a we plot $V_1(H)$ as a function of the expected number of samples T, when Y is a Brownian motion, with the same parameters as above. Clearly $V_1(H) \geq V_2(H)$. A simple lower bound for $V_2(H)$ is the one-step lookahead policy that takes exactly T samples. For each $T \in \{1, \ldots, 10\}$ we estimated this lower bound by simulating the one-step lookahead policy 50000 times. In Figure 5.3b we plot a region containing $V_2(H)$: the lower bound is provided by the one-step lookahead policy, and the upper bound is provided by $V_1(H)$. The fact that shaded region in Figure 5.3b is small means we have characterized $V_2(H)$ to high accuracy.

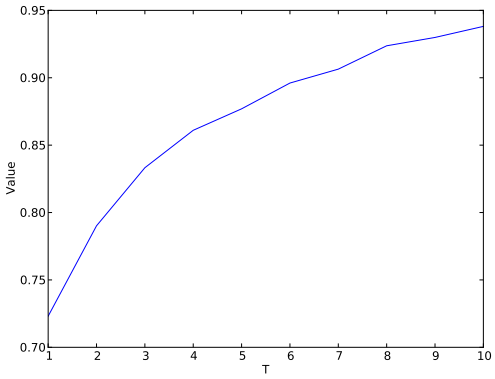


(a) Y is a **Brownian motion**.

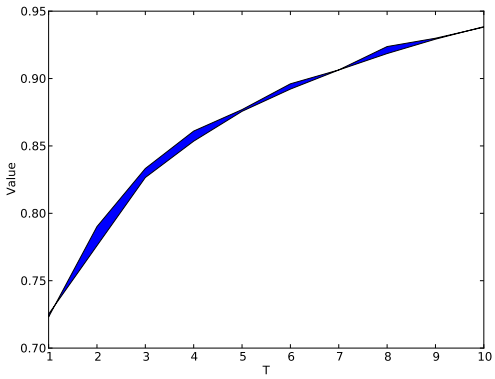


(b) Y is a **compound Poisson process**.

Figure 5.2: Value of optimal policy (in blue) and value of one-step lookahead policy (in green) vs. the cost of sampling.



(a) $V_1(H)$ vs. expected number of samples T



(b) Region containing $V_2(H)$ vs. number of samples T

Figure 5.3: Constrained-budget value plots (expected and almost sure contraints) when Y is a Brownian motion

6 Conclusion

In this paper, we consider a class of Bayesian optimization problems where the underlying prior is a Markov process and we pay a cost for each sample. We show that the Bayes-optimal policy is computationally tractable, by way of showing that the value function is completely determined by its values on a 3- or 4-dimensional set. We use this optimal cost-per-sample policy to compute the optimal value when there is no cost to sample, but there is a constraint on the expected number of values taken, as the result of a simple convex optimization problem. Computational experiments show that the optimal policy outperforms the commonly used one-step lookahead policy, but also that the optimality gap between one-step lookahead and the optimal policy is small, justifying the use of one-step lookahead in practice.

References

[1] F Archetti and B Betro. A probabilistic algorithm for global optimization. *Calcolo*, 16(3):335–343, 1979.

[2] B. Betrò and F. Schoen. A stochastic technique for global optimization. *Computers and Mathematics with Applications*, 21(6–7):127–133, 1991.

[3] Bruno Betrò. Bayesian methods in global optimization. *Journal of Global Optimization*, 1(1):1–14, 1991.

[4] E Brochu, M Cora, and N de Freitas. A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning. Technical Report TR-2009-023, Department of Computer Science, University of British Columbia, November 2009.

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

[6] J Calvin and A Žilinskas. On the convergence of the P-algorithm for one-dimensional global optimization of smooth functions. *Journal of Optimization Theory and Applications*, 102(3):479–495, 1999.

[7] J Calvin and A Žilinskas. One-Dimensional P-Algorithm with Convergence Rate O (n- 3+ δ) for Smooth Functions. *Journal of Optimization Theory and Applications*, 106(2):297–307, 2000.

[8] J M Calvin and A Zilinskas. One-dimensional Global Optimization Based on Statistical Models. *Nonconvex Optimization and its Applications*, 59:49–64, 2002.

[9] James M. Calvin. A One-Dimensional Optimization Algorithm and Its Convergence Rate under the Wiener Measure. *Journal of Complexity*, 17(2):306–344, June 2001.

[10] S. E. Chick and P. I. Frazier. Sequential sampling for selection with economics of selection procedures. *Management Science*, 58(3):550–569, 2012.

[11] A Forrester, A Sobester, and A Keane. *Engineering design via surrogate modelling: a practical guide*. Wiley, West Sussex, UK, 2008.

[12] P. I. Frazier. Tutorial: Optimization via simulation with bayesian statistics and dynamic programming. In C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, editors, *Proceedings of the 2012 Winter Simulation Conference Proceedings*, pages 79–94, Piscataway, New Jersey, 2012. Institute of Electrical and Electronics Engineers, Inc.

[13] P. I. Frazier, W. B. Powell, and S. Dayanik. The knowledge gradient policy for correlated normal beliefs. *INFORMS Journal on Computing*, 21(4):599–613, 2009.

[14] P.I. Frazier and J. Wang. Bayesian optimization for materials design. arXiv preprint, http://arxiv.org/pdf/1506.01349.pdf, 2015.

[15] Jacob Gardner, Matt Kusner, Zhixiang Xu, Kilian Weinberger, and John Cunningham. Bayesian optimization with inequality constraints. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 937–945, 2014.

[16] David Ginsbourger and Rodolphe Le Riche. Towards gaussian process-based optimization with finite time horizon. In *mODa 9–Advances in Model-Oriented Design and Analysis*, pages 89–96. Springer, 2010.

[17] Steffen Grünewälder, Jean-Yves Audibert, Manfred Opper, and John Shawe-Taylor. Regret bounds for gaussian process bandit problems. In *International Conference on Artificial Intelligence and Statistics*, pages 273–280, 2010.

[18] D.R. Jones, M. Schonlau, and W.J. Welch. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

[19] H. J. Kushner. A new method of locating the maximum of an arbitrary multi-peak curve in the presence of noise. *Journal of Basic Engineering*, 86:97–106, 1964.

[20] Marco Locatelli. Bayesian algorithms for one-dimensional global optimization. *Journal of Global Optimization*, 10(1):57–76, 1997.

[21] Marco Locatelli and Fabio Schoen. An adaptive stochastic global optimization algorithm for one-dimensional functions. *Annals of Operations research*, 58(4):261–278, 1995.

[22] J. Mockus. *Bayesian approach to global optimization: theory and applications*. Kluwer Academic, Dordrecht, 1989.

[23] C Perttunen and B.E. Stuckman. The rank transformation applied to a multi-univariate method of global optimization. In *Proceedings of the IEEE International Conference on Systems Engineering*, pages 217–220, 1989.

[24] Cary D Perttunen. A study of alternate stochastic models in kushner-based global optimization methods. In *Systems, Man, and Cybernetics, 1991.'Decision Aiding for Complex Systems, Conference Proceedings., 1991 IEEE International Conference on*, pages 597–601. IEEE, 1991.

[25] Cary D Perttunen and Bruce E Stuckman. The rank transformation applied to a multivariate method of global optimization. *IEEE Transactions on Systems, Man and Cybernetics*, 20(5):1216–1220, 1990.

[26] W. B. Powell. *Approximate Dynamic Programming: Solving the curses of dimensionality*. John Wiley and Sons, New York, 2007.

[27] Klaus Ritter. Approximation and optimization on the wiener space. *Journal of Complexity*, 6(4):337–364, 1990.

[28] M.J. Sasena. *Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations*. PhD thesis, University of Michigan, 2002.

[29] Warren Scott, Peter I. Frazier, and Warren B. Powell. The correlated knowledge gradient for simulation optimization of continuous parameters using gaussian process regression. *SIAM Journal on Optimization*, 21(3):996–1026, 2011.

[30] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, pages 2951–2959, 2012.

[31] Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, 2010.

[32] Emmanuel Vazquez and Julien Bect. Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. *Journal of Statistical Planning and inference*, 140(11):3088–3095, 2010.

[33] R. Waeber, P. I. Frazier, and S. G. Henderson. Bisection search with noisy responses. *SIAM Journal on Control and Optimization*, 51(3):2261–2279, 2013.

[34] J. Xie and P. I. Frazier. Sequential bayes-optimal policies for multiple comparisons with a known standard. *Operations Research*, 61(5):1174–1189, 2013.

[35] Antanas Zilinskas. Axiomatic characterization of a global optimization algorithm and investigation of its search strategy. *Operations Research Letters*, 4(1):35–39, 1985.