Bayesian Decision Theory

Drew Gjerstad

Contents

1 Introduction

The optimization process in its most basic form is a series of decisions. Ideally, these decisions are made via a principled approach (i.e., strategically) which is where **decision theory** comes into play. Specifically, the approach used to make such decisions should take into account any available data when deciding where each observation is made. Unfortunately, it is not clear how to make these decisions primarily due to the likely incomplete and ever-changing information about the objective function.

Previously, we discussed how to use Bayesian inference as a framework that systematically and quantitatively reasons about the uncertainty in the objective function. This is one of the main difficulties when making decisions during optimization since our knowledge of the objective function is only updated from the outcomes of our own decisions.

In this notebook, we focus on Bayesian decision theory which in effect, “bridges the gap” between Bayesian inference and decision making in optimization. This principled approach allows us to make decisions using a probabilistic belief about the objective function to guide optimization policies under uncertainty.

At the heart of the optimization process is the *optimization policy* which determines where we will make an observation next (for the time being, ignoring the question of termination), acquires the next observation, and updates our knowledge of the objective. Therefore, we aim to obtain the **optimal policy** with optimal referring to maximizing the expected utility and quality of the observed data.

While the idea of deriving this optimal policy may seem simple, especially when deriving it in a theoretical manner, the theoretically optimal policy is often impossible to compute and has little practical value. Regardless, the process of deriving this policy will enable us to see how we can obtain effective *approximations*.

Coming back to the question of termination, this question itself represents a decision that is crucial in several applications. A **stopping rule** is a procedure that decides whether to terminate or continue optimization based on the observed data. In many cases, this rule is *deterministic* meaning it is fixed and known before we begin optimizing. One example is a preallocated search budget defined by the maximum number of allowed observations. This type of stopping rule will terminate the optimization once we obtain the maximum number of allowed observations, regardless of our progress.

Alternatively, we may want to consider the optimization progress (i.e., our understanding of the objective function and the expected cost of continuing) when deciding whether to terminate or not. This is a more *dynamic* stopping rule that will require more subtle, adaptive stopping rules. We will discuss this more later and how its formulation inspires better approximations.

2 Defining Optimization Policies

To define an optimization policy, we typically use an intermediate function called the **acquisition function** that will score each observation candidate based on its utility to aiding the optimization process. Then, the policy can be defined to observe the point deemed to be most useful (or most "promising") by the acquisition function. Such a definition is used by nearly all Bayesian optimization policies, with some literature (as noted by Garnett) using the term "acquisition function" interchangeably with "policy".

When using the Bayesian approach, the acquisition function is almost always defined by obtaining the posterior belief (distribution) of the objective given the data and then defining our preferences for the next observation with respect to this belief. Using the notation from Garnett's book, we will denote $\alpha(x; \mathcal{D})$ for a general acquisition function with the data, $\mathcal{D}$, serving as parameters that shape our preferences.

In more mathematical terms, an acquisition function α defines preferences over candidate observations by "inducing a total order over the domain". This means that given existing data $\mathcal{D}$, observing candidate x is preferred over another candidate x' if $\alpha(x; \mathcal{D}) > \alpha(x'; \mathcal{D})$. Rationally, the action we will prefer is one that maximizes the acquisition function:

$$x \in \arg\max_{x' \in \mathcal{X}} \alpha(x'; \mathcal{D})$$

We can use the formulation above as a kind of "sub-optimization problem". Once it is solved, the acquisition function will map a set of observed data to a candidate $x \in \mathcal{X}$ to observe next, filling the exact role of an optimization policy.

If you are thinking that the idea of solving global optimization problems by repeatedly solving global optimization problems is unintuitive, don't worry! In many cases, this paradox is resolved by the fact that common acquisition functions have properties making their optimization much more tractable than the primary optimization problem we are aiming to solve.

Commonly used acquisition functions are both inexpensive to evaluate and are analytically differentiable which means we can use pre-defined optimizers while computing the policy formulated above. However, recall that our objective function is assumed to be rather expensive to evaluate and lacks efficient (if any at all) gradients. Using the ideas outlined here, we are able to moderate a difficult problem to several simpler problems, a reasonable first step!

3 Formalizing Bayesian Decision Theory

Bayesian decision theory is a framework that we can use to make decisions under uncertainty while still being flexible enough that it can be applied to nearly any problem. Here, we introduce Bayesian decision theory in the same manner as Garnett: focusing on the key concepts through the lense of optimization rather than unloading the entire theory abstractly. Garnett recommends the following supplementary texts for a more thorough and in-depth review of the theory:

- *Optimal Statistical Decisions* by M. H. DeGroot
- *Statistical Decision Theory and Bayesian Analysis* by J. O. Berger

Being sufficiently familiar with this topic can help you understand key concepts in Bayesian optimization that are examined in the literature less thoroughly than they perhaps should be. In particular, this topic, as Garnett puts it, serves as the "hidden origin" of several typical acquisition functions.

Following from Garnett's text, we start with using the Bayesian decision theory approach for decision making and examine the case of making a single, isolated decision to see how the framework is used to make optimal decisions. Then, we will extend this reasoning to make several, or a sequence of, decisions.

3.1 Case 1: Single, Isolated Decisions

There are two defining characteristics of a decision problem under uncertainty: the action space and the presence of uncertain elements in the environment. We will review these characteristics first.

The **action space** $\mathcal{A}$ is the set of all available decisions. Keep in mind that the task at hand is to select an action from this space. In the context of sequential optimization, we are selecting a point in the domain $\mathcal{X}$ to observe so we have that $\mathcal{A} = \mathcal{X}$.

The **presence of uncertain elements** in the environment will inherently influence the results of our actions which complicates our decision. Using Garnett's notation, let ψ denote a random variable that encompasses any relevant uncertain elements when making and evaluating a decision. While we may not have all the information about the uncertainty, we can use Bayesian inference to reason about ψ given the observed data using the posterior distribution $p(\psi|\mathcal{D})$. We can use this belief to aid our decision.

Suppose that now we need to make a decision (selected from the action space, $\mathcal{A}$) under the uncertainty in ψ, and informed by observed data $\mathcal{D}$. We need some way to guide our decision selection process: a *utility function.*

A real-valued **utility-function** $u(a, \psi, \mathcal{D})$ is used to guide our choice by measuring the quality of choosing action a if the true state of the environment is ψ, with higher utilities being preferred since the higher the utility score, the more favorable the outcome. Notice that the arguments provided to the utility function are all that is required to judge the quality of a decision:

- the proposed action a

- observed data informing our current knowledge $\mathcal{D}$
- uncertain elements missing from our knowledge ψ

Since we have incomplete information about ψ, we are unable to know the exact utility of selecting any given action. However, we can compute the *expected* utility of selecting an action a based on our posterior belief:

$$\mathbb{E}\left[u(a, \psi, \mathcal{D})|a, \mathcal{D}\right] = \int u(a, \psi, \mathcal{D})p(\psi|\mathcal{D})d\psi$$

The expected utility above maps each action to a real value that induces a total order and provides a simple method to make our decision. We then select an action that maximizes the expected utility:

$$a \in \underset{a' \in \mathcal{A}}{\arg\max}\, \mathbb{E}\left[u(a', \psi, \mathcal{D})|a', \mathcal{D}\right]$$

By using this approach, the decision is considered to be optimal as there are no other actions that would result in greater expected utility. Furthermore, this method of selecting actions optimally under uncertainty is the central concept of Bayesian decision making.

3.2 Case 2: Sequential Decisions (with fixed budget)

Previously, we examined the single-decision case where we used Bayesian decision theory as a framework to select optimal decisions based on the data. The main idea was to evaluate a decision's quality after it occurs and then select actions that maximize the expected utility. Now, we will extend this reasoning to sequential decisions and specifically, the construction of optimization policies. This is a bit more complicated, however, as any single decision will impact all of the future decisions we will make.

To construct an optimization routine, we will need to define a policy that adaptively designs a sequence of observations (actions) that move us closer to the optimum. Using the concepts outlined previously, each choice can be modeled as a decision problem under uncertainty.

- Let $\mathcal{X}$ be the domain and action space of each decision.
- Let f be the objective function.

We will utilize the idea of probabilistic beliefs during optimization so that we can reason about the uncertainty in the objective. Recall that this is called the posterior predictive distribution or posterior process, $p(f|\mathcal{D})$. At this time, we do not need to make any assumptions about this distribution nor do we need to think of it as a Gaussian process. However, we can use this distribution to reason about the result of an observation at location x: $p(y|x, \mathcal{D})$.

Recall that the main goal of optimization is to collect and return a dataset $\mathcal{D}$. This means that we need to determine what data we *want* to acquire and this is accomplished by defining a utility function that evaluates the quality of data obtained by the optimizer. More specifically, the utility function establishes our preferences with the common preference being to obtain a dataset of higher utility than any dataset of lower utility. The utility will guide our policy design by "choosing" observations that we expect to improve the utility the most.

In the next notebook, we will define several utility functions in detail whereas here we will continue developing Bayesian decision theory using an *arbitrary utility function*.

Facing Uncertainty During Optimization

During optimization, we are always facing some sort of uncertainty. However, the kind of impact this uncertainty has is what distinguishes the isolated, single-decision case from the sequential decisions case. In the single-decision case, we simply select the next observation x that maximizes the expected utility. On the other hand, in the sequential decisions case, we repeatedly select the next observation x in a similar manner (i.e., maximizing the expected utility) but in this case, after each observation x and corresponding value y are obtained, they are added to our dataset. This means that the observations in the dataset, including the observations we chose, will be used to make future decisions. More generally, the observations we select will impact the "entire remainder of optimization" and extra consideration compared to the isolated, single-decision case.

From this realization, we can intuitively think that making decisions closer to termination are easier since there are less (if any) future decisions that will rely on their outcomes. Using this to our advantage, we will design optimization policies *in reverse* where we will initially reason about the last decision. The last decision is made using approach of the isolated, single-decision case since we do not have to consider any future decisions. Then, we will continue reasoning backwards through decisions until the first decision, defining optimal behavior as we go.

Construction of Optimization Policies

When considering the construction of optimization policies, we will assume that we are constrained by a pre-defined and fixed search budget representing the maximum number of observations we can make. In addition to being common in practice, this assumption also makes the analysis of policy design more convenient (particularly because we can ignore the question of termination).

As noted by Garnett, this assumption will also imply that every observation has a constant acquisition cost that may not always be reasonable. Considerations of acquisition costs and the question of termination will be examined later on.

Under this assumption of fixed budget, we can analyze policies using the number of future observations before termination–which is known. Then, using the setup from Garnett, the problem becomes this: given a set of data, how should we select our next evaluation point when exactly τ observations remain before we terminate? Note that τ denotes the **decision horizon** and indicates the number of remaining observations.

We will define notation used by Garnett to analyze optimization policies below.

- Let x denote the location of an observation.
- Let y denote the corresponding value of an observation at location x.
- Let $\mathcal{D}_i = \mathcal{D} \cup \{(x_i, y_i)\}, i \in \{1, \dots, \tau\}$ be the dataset that is available at the next stage of optimization where the subscript i indicates the number of future observations incorporated with the current data.
  - The dataset returned by our optimization procedure will be $\mathcal{D}_\tau$ with utility $u(\mathcal{D}_\tau)$.

To measure the utility of the data, we will use the same format as before:

$$u(\mathcal{D}_\tau) = u(D, x, y, x_2, y_2, \dots, x_\tau, y_\tau)$$

This format expresses the **terminal utility** in terms of the proposed current action x, observed data $\mathcal{D}$, and unknown future data that will be obtained: not-yet observed value y, locations $\{x_2, \dots, x_\tau\}$ and corresponding values $\{y_2, \dots, y_\tau\}$ of future observations.

Extending the treatment of isolated, single decisions, we can evaluate an candidate observation at point x using the expected *terminal* utility if we observed that point next:

$$\mathbb{E}\left[u(\mathcal{D}_\tau) | x, \mathcal{D}\right]$$

Just as before, we can now define an optimization policy that maximizes this utility:

$$x \in \underset{x' \in \mathcal{X}}{\arg\max}\, \mathbb{E}\left[u(\mathcal{D}_\tau) | x', \mathcal{D}\right]$$

Conceptually, these ideas are rather simple from a theoretical perspective but we must consider how we will actually compute the expected terminal utility. The explicit form from Garnett is the expectation over future observations:

$$\int \dots \int u(\mathcal{D}_\tau) p(y | x, \mathcal{D}) \prod_{i=2}^{\tau} p(x_i, y_i | \mathcal{D}_{i-1}) \,\mathrm{d}y \,\mathrm{d}\{(x_i, y_i)\}$$

Actually computing this integral is rather unwieldy and so we will instead compute this expression under the assumption that *all future decisions are made optimally* (Bellman's Principle of Optimality). Such analysis will obtain the optimal optimization policy, and will be covered later on.

For now, we will use **backward induction** to determine the optimal behavior if only one observation remains and continue backwards inductively to consider increasingly long horizons. We will use Garnett's notation for the expected *increase* in utility beginning from an arbitrary dataset $\mathcal{D}$, making an observation at x, and behaving optimally until we reach termination τ steps in the future. This is given by:

$$\alpha_\tau(x; \mathcal{D}) = \mathbb{E}\left[u(\mathcal{D}_\tau)|x, \mathcal{D}\right] - u(\mathcal{D})$$

Notice that this is merely the difference between the expected terminal utility and the utility of the existing dataset. Furthermore, such notation is similar to the notation used for acquisition functions and this enables us to define the optimal optimization policy using acquisition functions defined in this way.

Base Case: One Observation Remaining

Using the method of backward induction, we start with the case where there is only one observation remaining in the horizon; there are $\tau = 1$ steps left before termination. For this case, the terminal dataset $\mathcal{D}_\tau$ is the current dataset augmented with one additional observation. Recall that this case is essentially the isolated, single decision case and so we can use the framework developed for that case.

First, we need to compute the marginal gain in utility from a final evaluation at x. This is an expectation over the corresponding value y with respect to the posterior predictive distribution:

$$\alpha_1(x; \mathcal{D}) = \int u(\mathcal{D}_1) p(y|x, \mathcal{D})\, \mathrm{d}y - u(\mathcal{D})$$

Using the framework developed for the single decision case, the optimal observation is the observation that maximizes the expected marginal gain:

$$x \in \operatorname*{arg\,max}_{x' \in \mathcal{X}} \alpha_1(x'; \mathcal{D})$$

This leads to the dataset returned by the optimizer having expected utility:

$$u(\mathcal{D}) - \alpha_1^*(\mathcal{D}); \qquad \alpha_1^*(x'; \mathcal{D}) = \max_{x' \in \mathcal{X}} \alpha_1(x'; \mathcal{D})$$

From this, we denote the **value** of the dataset by $\alpha^*_\tau(\mathcal{D})$ and represents the expected increase in the dataset's utility if we start with arbitrary dataset $\mathcal{D}$ and continue in an optimal manner for τ more observations. This will be key in further analysis but for now, this concludes the *base case.*

See Figure 5.1 in Garnett's text for an illustration of the optimal optimization policy when the decision horizon is $\tau = 1$.

Special Case: Two Observations Remaining

Before we begin examining the inductive case, we will review a special case where there are two observations remaining (i.e., the decision horizon is $\tau = 2$). Just as in the base case, suppose we have an arbitrary dataset $\mathcal{D}$ but now we need to decide where the next to last observation x should be. The reasoning developed for this special case will help to highlight the inductive approach.

In the same manner as we did in the base case with $\tau = 1$, we consider the expected increase in terminal utility after two observations:

$$\alpha_2(x; \mathcal{D}) = \mathbb{E}\left[u(\mathcal{D}_2) | x, \mathcal{D}\right] - u(\mathcal{D})$$

Based on the definition of expectations, the expectation above should require that we marginalize the observation y, the final observation x_2, and its corresponding value y_2. Luckily, we can use Bellman's Principle of Optimality to assume that future behavior is optimal, allowing us to simplify how we approach the final decision x_2.

First, we will redefine the two-step expected gain in utility α_2 in terms of the single-step case α_1, a function that we have a much more established understanding of. From Garnett, this "two-step difference" in utility can be expressed as a *telescoping sum*:

$$u(\mathcal{D}_2) - u(\mathcal{D}) = [u(\mathcal{D}_1) - u(\mathcal{D})] + [u(\mathcal{D}_2) - u(\mathcal{D}_1)]$$

This allows us to separate the expected increase in terminal utility after two observations into two terms: the expected increase after the first observation (the **expected immediate gain**) and the expected additional increase after the final observation (the **expected future gain**), shown below.

$$\alpha_2(x; \mathcal{D}) = \alpha_1(x; \mathcal{D}) + \mathbb{E}\left[\alpha_1(x_2; \mathcal{D}_1) | x, \mathcal{D}\right]$$

While it is not fully clear how we should address the second term (the expected future gain), we can use our analysis of the base case to help us reason. Given the observation x's value y and knowledge of $\mathcal{D}_1$, the *optimal* final observation x_2 will result in an expected marginal gain of $\alpha_1^*(\mathcal{D}_1)$ (a quantity that we can compute). Thus, under the assumption of optimal future behavior, we can express the expectation with the current observation y only:

$$\alpha_2(x;\mathcal{D}) = \alpha_1(x;\mathcal{D}) + \mathbb{E}\left[\alpha_1^*(\mathcal{D}_1)|x,\mathcal{D}\right]$$

Once again, the optimal next-to-last observation maximizes the expected gain:

$$x \in \operatorname*{arg\,max}_{x' \in \mathcal{X}} \alpha_2(x';\mathcal{D})$$

Furthermore, it will provide an expected terminal utility of:

$$(\mathcal{D}) + \alpha_2^*(\mathcal{D}) \qquad \alpha_2^*(\mathcal{D}) = \max_{x' \in \mathcal{X}} \alpha_2(x';\mathcal{D})$$

This analysis shows that we are able to achieve optimal behavior and compute the value of any dataset with a horizon of $\tau = 2$.

See Figures 5.2 and 5.3 in Garnett's text for an illustration of an optimal two-step optimization policy.

Inductive Case

Now that we have examined and analyzed the base and special cases with horizons of $\tau = 1$ and $\tau = 2$, respectively, we can examine the general inductive case. As mentioned above, the inductive case will be rather similar to the special case with $\tau = 2$.

Let τ be an arbitrary decision horizon. Assume that we are able to compute the value of any dataset with a horizon of $\tau - 1$. Suppose we have an arbitrary dataset $\mathcal{D}$ and we need to decide where the next observation should be made. In this section, we will review how to do this in an optimal manner and how to compute its value.

From Garnett, the τ-step expected utility gain from observing x is:

$$\alpha_\tau(x;\mathcal{D}) = \mathbb{E}\left[u(\mathcal{D}_\tau)|x,\mathcal{D}\right] - u(\mathcal{D})$$

We wish to maximize the expected utility gain given above. Just as in the special case, we can express this using *shorter-horizon quantities* using a telescoping sum:

$$\alpha_\tau(x;\mathcal{D}) = \alpha_1(x;\mathcal{D}) + \mathbb{E}\left[\alpha_{\tau-1}(x_2;\mathcal{D}_1)|x,\mathcal{D}\right]$$

Then, if we knew the corresponding value y (and therefore $\mathcal{D}_1$), the assumption of optimal behavior provides an expected further gain of $\alpha_{\tau-1}^*(\mathcal{D}_1)$ which is a quantity we are able to compute via the inductive hypothesis. Therefore, if we assume optimal behavior for all future decisions, we can express the expected utility gain:

$$\alpha_\tau(x; \mathcal{D}) = \alpha_1(x; \mathcal{D}) + \mathbb{E}\left[\alpha^*_{\tau-1}(\mathcal{D}_1) | x, \mathcal{D}\right]$$

To determine the optimal decision and the τ-step value of the dataset, we maximize the expected utility gain at step τ:

$$x \in \arg\max_{x' \in \mathcal{X}} \alpha_\tau(x'; \mathcal{D})$$

$$\alpha^*_\tau(\mathcal{D}) = \max_{x' \in \mathcal{X}} \alpha_\tau(x'; \mathcal{D})$$

This concludes our analysis and shows that we can attain optimal behavior for a horizon of τ given a dataset $\mathcal{D}$ and compute its value.

Bellman's Principle of Optimality and the Bellman Equation

If we substitute the expected utility gain (expressed using shorter-horizon quantities) into the the final expression above (maximizing the expected utility gain at step τ), we obtain the **Bellman equation**:

$$\alpha^*_\tau(\mathcal{D}) = \max_{x' \in \mathcal{X}} \{\alpha_1(x'; \mathcal{D}) + \mathbb{E}\left[\alpha^*_{\tau-1}(\mathcal{D}_1) | x', \mathcal{D}\right]\}$$

The Bellman equation forms the recursive definition of the value in terms of the value of future data. It is a key result in the theory of optimal sequential decisions. In particular, it reflects **Bellman's Principle of Optimality** which states that we always act optimally to maximize the expected terminal utility given the available data. More generally, it characterizes optimal policies in terms of the optimality of sub-policies.

Here is a quote provided by Garnett from Bellman's *Dynamic Programming* book:

An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

Therefore, in order to create a sequence of optimal decisions, we choose the first decision optimally and then choose all future decisions optimally given the outcome.

4 Cost of the Optimal Policy

While the framework introduced in the previous sections is theoretically simple, actually computing the optimal policy is prohibitive. The only exception is if we are computing the policy for very short decision horizons.

To show this barrier, recall the expression for the expected utility gain where we assume optimal behavior (with $\tau = 2$):

$$\alpha_2(x; \mathcal{D}) = \alpha_1(x; \mathcal{D}) + \mathbb{E}\left[\alpha_1^*(\mathcal{D}_1) | x, \mathcal{D}\right]$$

As Garnett notes, even though the second term appears to be a rather standard expectation over the random variable y, evaluating this requires solving a *non-trivial* global optimization problem:

$$\alpha_1^*(\mathcal{D}) = \max_{x' \in \mathcal{X}} \alpha_1(x'; \mathcal{D})$$

Furthermore, even if we are only considering a horizon with two decisions left, we would need to solve a dually nested global optimization problem (no simple feat!). Similarly, from the recursively defined optimal policy, we can see that if the horizon is τ then we need to solve τ nested optimization problems to attain the optimal decision. Here we show this, following the temporary compact notation used by Garnett:

$$x \in \arg\max \alpha_\tau$$

$$\begin{aligned}
\alpha_\tau &= \alpha_1 + \mathbb{E}\left[\alpha_{\tau-1}^*\right] \\
&= \alpha_1 + \mathbb{E}\left[\max \alpha_{\tau-1}\right] \\
&= \alpha_1 + \mathbb{E}\left[\max\{\alpha_1 + \mathbb{E}\left[\alpha_{\tau-2}^*\right]\}\right] \\
&= \alpha_1 + \mathbb{E}\left[\max\{\alpha_1 + \mathbb{E}\left[\max\{\alpha_1 \mathbb{E}\left[\max\{\alpha_1 + \dots\right]\}\right]\}\right]
\end{aligned}$$

This means that the design of any optimal decision would require repeated maximization over the domain plus expectation over unknown observations until we reach the horizon. See Figure 5.4 in Garnett's text for a visualization of this problem as a decision tree. There we can clearly see that each unknown quantity will contribute a significant branching factor and that computing the expected utility at x will require a traversal of the entire tree.

We can use this structure to determine the cost of computing the optimal policy, which obviously grows with the horizon. Here, we outline the running time analysis for a naïve implementation via exhaustive traversal of the decision tree provided by Garnett.

Suppose that for each maximization we use an optimization routine and for each expectation we use a numerical quadrature routine.

Allowing n evaluations of the objective each time the optimizer is called and q observations of the integrand per call to the quadrature routine, each decision along the horizon will contribute a multiplicative factor of $\mathcal{O}(nq)$ to the total running time. Therefore, the amount of work required to compute the optimal decision with a horizon of τ is $\mathcal{O}(n^\tau q^\tau)$. Clearly, the running time will grow exponentially with respect to the horizon.

In the next section, we discuss how to avoid this prohibitive barrier by approximating the optimal policy.

5 Approximation of the Optimal Policy

Due to the exponential growth of the running time with respect to the horizon, the computational work required to obtain the optimal policy becomes intractable. However, we can utilize general approximation schemes to compute the optimal policy. These schemes are methods deeply studied in **approximate dynamic programming**.

Recall the *intractable* optimal expected marginal gain which is given by

$$\alpha_\tau(x;\mathcal{D}) = \alpha_1(x;\mathcal{D}) + \mathbb{E}\left[\alpha^*_{\tau-1}(\mathcal{D}_1)|x,\mathcal{D}\right].$$

To avoid the difficult part of the expression, the recursively defined future value α^*, we can substitute in a tractable approximation. While this will induce a suboptimal and approximate policy, it is still rationally guided. There are two specific approximation schemes that are commonly used in Bayesian optimization: *limited lookahead* and *rollout*.

5.1 Limited Lookahead

The idea behind the **limited lookahead** approximation scheme is to, as the name suggests, limit how far into the future we look when making decisions. Specifically, we restrict how many future observations we will consider in each decision. Such an approach is very practical since the closer decisions are to termination, the (significantly) less computation required compared to earlier decisions.

With this reasoning, we will develop a family of approximations to the optimal policy defined by "artificially" limiting the horizon used during optimization to a feasible maximum ℓ. Thus, if we face an infeasible decision horizon τ, then we use the approximation

$$\alpha_\tau(x;\mathcal{D}) \approx \alpha_\ell(x;\mathcal{D}).$$

When we maximize this score, we will act optimally under the *incorrect but convenient assumption* that there are only ℓ observations left. Effectively, this assumes $u(\mathcal{D}_\tau) \approx u(\mathcal{D}_\ell)$. This scheme is often regarded (sometimes in a disparaging manner) as *myopic* due to the fact that we limit ourselves to only considering the next few observations on the horizon instead of viewing the entire horizon.

An **ℓ-step lookahead policy** is a policy that selects each observation to maximize the limited-horizon acquisition function, denoted $\alpha_{\min\{\ell,\tau\}}$. It is also considered a *rolling horizon strategy* since the truncated horizon "rolls along" with us as we continue.

Considering computational complexity, we are able to bound the effort required when we limit the horizon. The effort is bounded to at most $\mathcal{O}(n^\ell q^\ell)$ for each decision. This can be a major speedup, particularly when the observation (search) budget is significantly larger than the selected maximum lookahead, ℓ.

One-Step Lookahead

A special case of the limited lookahead approach is the **one-step lookahead** method which is very important in Bayesian optimization. Since it aims to successively maximize the expected marginal gain from acquiring one more observation (α_1), it is often possible to derive closed-form, analytically differentiable expressions for α_1. This makes it the most efficient lookahead approximation.

5.2 Rollout

In our theoretical exploration, the optimal policy will evaluate a candidate observation point by simulating the rest of the optimization after that decision, under the recursive assumption that we decide optimally for every future decision. While rational, it's intractable. The **rollout** approach emulates the structure of the optimal policy but instead uses a tractable *suboptimal* policy to simulate future decisions.

Specifically, given another observation (x, y), we use an inexpensive *base* or *heuristic* policy to simulate a reasonable but potentially suboptimal realization of the next decision x_2. Then, we take an expectation with respect to the unknown value at x_2, y_2. We continue forward, using the base policy to select another point, x_3, and so forth until we reach the decision horizon. Since this approach does not lead to branching in the tree for this decision, we avoid the expensive subtree that would be required by the optimal policy. Instead, we use the terminal utilities from the resulting pruned tree to estimate the expected marginal gain α_τ that we maximize as a function of x.

While there are not any restrictions on how we design the base policy, since the point of this approximation is to improve efficiency, the base policy design should be something fairly efficient. One common and typically effective choice pointed out by Garnett is to simulate the future decisions using the one-step lookahead approach. That way, if we use off-the-shelf optimizers and quadrature routines to traverse the resulting, rollout decision tree (using one-step lookahead as the base policy), the computational complexity of the policy with decision horizon τ is $\mathcal{O}(n^2 q^\tau)$. As Garnett points out, this is considerably faster than the optimal policy and while we still have exponential growth with respect to the number of observations (q), in most cases $q << n$.

Furthermore, the flexibility in the design of the base policy makes the rollout approach an extremely flexible policy approximation scheme. One example provided by Garnett is if we were to combine rollout with limited lookahead to attain approximate policies with *tunable running time*. In particular, we can view ℓ-step lookahead as a special case of rollout where the base policy designs the $\ell - 1$ future decisions optimally assuming a myopic horizon and then *terminates early*, ignoring any observations left in the budget.

Additionally, we could use a base policy that designs all of the remaining observations in the budget *simultaneously*. By ignoring the dependence between these decisions, we can achieve a computational advantage while still being aware of the horizon. These **batch rollout** schemes have been shown to work well in Bayesian optimization and while we account for the entire horizon, the resulting tree depth is still much less compared to the optimal policy tree.

6 Cost-Aware Optimization and Treating Termination as a Decision

Up to this point, we have only considered optimization policies that are under a pre-defined and constant budget for the maximum number of observations. While this setup is common, it is not universal and in some cases we may want to leverage our changing beliefs about the objective function to help us decide *dynamically* when termination is the best decision.

The idea of dynamic termination is advantageous when we want to account for the cost of acquiring data during optimization. For example, if the cost of gaining more data varies across the search space then it makes no sense to define our budget based on the number of evaluations. Instead, we can account for the acquisition costs in the utility function to explicitly reason about the cost-benefit tradeoff for each additional observation. If the cost of acquiring an additional point outweighs its expected benefit it may provide, we can seek to terminate the optimization process.

6.1 Modeling Termination as a Decision

To model dynamic termination, we modify the previously defined sequential decisions case with a pre-defined and constant budget. Now, we will allow ourselves to terminate optimization at any point, as we see fit.

Suppose we are performing optimization and have already acquired data $\mathcal{D}$. Unlike the known-budget case where we would need to decide where to sample next, we face a new decision: is it best to terminate optimization and return dataset $\mathcal{D}$? If not, where should we sample next?

We can model this as a decision problem under uncertainty with the action space being the domain $\mathcal{X}$ but now we will augment the action space with a special additional action $\emptyset$ representing termination:

$$\mathcal{A} = \mathcal{X} \cup \{\emptyset\}$$

Note that we will follow Garnett's recommendation to model the decision process as not actually terminating after the termination action is selected but rather continuing with a collapsed action space: $\mathcal{A} = \{\emptyset\}$ (once we terminate, we cannot go back).

We could derive the optimal optimization policy for dynamic termination using induction but we need to address the issue of the base case (representing the "final" decision) breaking down once we allow the potential for a non-terminating sequence of decisions. To address this, we will operate under the assumption that there is a fixed, known upper-bound τ_{max} on the total number of observations we can make. Upon reaching this bound, the optimization will terminate no matter what.

Now that we assume the decision process is bounded, the inductive argument derived in the known-budget applies although we need to re-define how we will compute the value of the termination action. Luckily, this is fairly obvious: since termination does not augment our data and no actions can be taken after we terminate, the expected marginal gain from termination will always be zero:

$$\alpha_\tau(\emptyset; \mathcal{D}) = 0$$

Now, apart from substituting $\mathcal{A} for \mathcal{X}$ in the previous set of derived expressions, we obtain the optimal policy for the case of dynamic termination.

See Figure 5.8 in Garnett's text for an illustration of one-step lookahead with the option to terminate. In this example, the optimization policy accounts for the cost of observations across the domain.

6.2 Considering Location-Dependent Observation Costs

Suppose that the cost of acquiring an observation depends on the location and is defined by a known cost function $c(x)$. In practice, the observation cost function could be unknown or stochastic (this will be examined in other notebooks and in Garnett's Chapter 11).

To explicitly reason about observation costs, location-dependent or otherwise, we can change our approach to the utility function. One rather natural approach would be to select a utility function that exclusively measures the returned dataset's quality (i.e., ignores any costs incurred during acquisition). This is referred to as the **data utility**, denoted by $u'(\mathcal{D})$, and is parallel to the cost-agnostic utility from the known-budget case.

Next, we adjust the data utility to consider the acquisition cost of observations. In several applications, the acquisitions costs are additive meaning the cost of acquiring an arbitrary dataset $\mathcal{D}$ is:

$$c(\mathcal{D}) = \sum_{x \in \mathcal{D}} c(x)$$

If we express the cost of acquisition and data utility in the same units (i.e., monetary units such as dollars), then we could evaluate a dataset based on its utility and cost via the **cost-adjusted utility**:

$$u(\mathcal{D}) = u'(\mathcal{D}) - c(\mathcal{D})$$

7 References

For full-reference details, the BibTeX entries can be found in the `bibliography.bib` file.

- *Bayesian Optimization* by Roman Garnett (2023)
- *Bayesian Optimization: Theory and Practice Using Python* by Peng Liu (2023)