Bayesian Decision Theory

Drew Gjerstad

Contents

1 Introduction

Decisions are at the heart of optimization and in its most basic form, the optimization process is just a *series of decisions*. Ideally, these decisions should be optimal and that will require us to make decisions via a principled approach rather than a random one. Thus, the notion of **decision theory** is a framework for making optimal decisions based on the data and our preferences. Specifically, it uses the available observations in the dataset and our preferences expressed in the *utility function* to make decisions about where each observation is made. Unfortunately, in many practical scenarios it is not entirely obvious how to make these decisions due to our typically incomplete and ever-changing knowledge about the underlying objective.

- In Bayesian optimization, one of the main difficulties when making decisions is the *uncertainty* inherent in the objective. Furthermore, our knowledge of the objective is only updated from the outcomes of our own decisions which means that making decisions optimally is even more important.
- Connecting decision theory to the optimization process is the *optimization policy* that determines where the next observation will be made, acquires the observation, and updates our knowledge of the objective. Therefore, we want to derive the **optimal policy** where optimal refers to *maximizing the expected utility* of the observed data.
- Keep in mind that the theoretical optimal policy is usually impossible to compute, and has little practical value. However, in these notes, the process of deriving this policy will allow us to see how we can define effective *approximations*.

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

2 Defining Optimization Policies

When defining an optimization policy, we use an intermediary referred to as the **acquisition function** that scores each candidate observation based on its utility to aiding the optimization process. Using the acquisition function, we can design the policy to observe the point which is deemed to be the most useful, or promising. This manner of decision making is used by nearly all Bayesian optimization policies. Bear in mind that, as noted by Garnett, some literature uses the term “acquisition function” interchangeably with “policy”.

Incorporating the Bayesian approach, the acquisition function is typically defined by computing the posterior belief of the objective given the data and then expressing our preferences for the next observations with respect to this belief. To mathematically outline these ideas, we will follow Garnett’s notation.

Let $\alpha(x; \mathcal{D})$ denote a general acquisition function where $\mathcal{D}$ is the data that serves as parameters to shape our preferences. The acquisition function α defines our preferences over candidate observations by inducing an order across the entire domain. Specifically, given data $\mathcal{D}$, observing candidate x is preferred over another candidate x' if

$$\alpha(x; \mathcal{D}) > \alpha(x'; \mathcal{D})$$

Simply put, we prefer to observe the point that maximizes the acquisition function:

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

The formulation above can be treated as a “sub-optimization problem”. Once it is solved, the acquisition function will use the observed data to identify a candidate $x \in \mathcal{X}$ to observe next. Thus, it fills the exact role of an optimization policy: to identify and acquire observations that ought to benefit the optimization process.

The idea of solving global optimization problems by repeatedly solving global optimization problems is unintuitive. However, Garnett points out that this paradox is resolved since many common acquisition functions have properties that make their optimization more tractable than the primary optimization problem we wish to solve. In particular, common acquisition functions are typically both inexpensive to evaluate and are analytically differentiable such that we can use pre-defined optimizers to compute the policy. From these ideas, we are able to reduce a difficult problem to several simpler problems, a practical first step!

3 Formalizing Bayesian Decision Theory

Bayesian decision theory is a framework that enables us to make decisions under uncertainty and is flexible enough that it can be applied to nearly any problem. In this section, we will formalize Bayesian decision theory with a focus on its ideas through the lense of optimization as opposed to discussing the entire theory in an abstract manner. For a more thorough and in-depth review, Garnett recommends the following supplementary texts:

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

This topic is crucial to understanding the main concepts in Bayesian optimization even though they are examined in the existing literature less thoroughly than they ought to be. For instance, Garnett refers to Bayesian decision theory as the "hidden origin" of several acquisition functions. In the next few sections we will follow Garnett's approach, starting with using the theory to make a single, isolated decision and then extending the reasoning to make a series of decisions.

3.1 Case 1: Single, Isolated Decisions

Before we jump into using decision theory to make decisions we need to review two defining characteristics of a decision problem uncertainty: the action space and the presence of uncertain elements in the environment.

- The **action space** denoted $\mathcal{A}$ is the set of all available decisions. Recall that the task at hand is to select an action from this space. In the context of sequential optimization, we select 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 directly influence the results of our actions, complicating our decision. Let ψ denote a random variable that encompasses any relevant uncertain elements. Even if we do not have all the information about uncertainty, we can use Bayesian inference to reason about ψ given $\mathcal{D}$ via the posterior distribution $p(\psi|\mathcal{D})$.

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

- A **utility function** denoted $u(a, \psi, \mathcal{D})$ evaluates the quality of choosing action a if the true state of the environment is ψ. It guides our decisions by preferring higher utilities since it indicates a likely, favorable outcome. Notice that the arguments passed to the utility function are all that is required to evaluate the quality of the decision.

However, since we have incomplete knowledge of ψ, we cannot compute the exact utility of any action. Using Bayesian inference, we can compute the *expected utility* of 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 induces an order across the entire action space by mapping each action to a real value representing its potential benefit to the optimization process. It provides a simple method to make a decision since all we need to do is select the action that maximizes the expected utility:

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

When using this approach, we consider the decision to be optimal as there are no other actions in the action space that would result in greater expected utility. This notion of optimally selecting actions under uncertainty forms the core of Bayesian decision making.

3.2 Case 2: Sequential Decisions

In the previous case, we examined how to use Bayesian decision theory as a framework to select a single, isolated optimal decision. The main premise was to use a utility function to evaluate a proposed action's benefit and select the action that maximize the expected utility. In this case, we will extend this reasoning to make optimal, *sequential* decisions leading to the construction of optimization policies. Recall that this is more complicated since any single decision directly impacts all future decisions we will make.

To construct an optimization routine, we need to define a policy that adaptively designs a sequence of actions which moves us closer to the optimum. Similar to the previous case, each decision 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.

Just as in the previous case, we will reason about the uncertainty in the objective using our probabilistic beliefs via the posterior process $p(f|\mathcal{D})$. In addition, we will also use a utility function that expresses our preferences for what data we *want* to acquire. The utility function will guide our policy by "choosing" observations expected to maximally improve the utility.

In another set of notes, we will define several utility functions in detail whereas here we will continue the development of Bayesian decision theory using an *arbitrary utility function.*

Facing Uncertainty During Optimization

Regardless of the problem, we will always face some sort of uncertainty during optimization. However, the impact that the uncertainty has is what distinguishes the isolated, single-decision case from the sequential decisions case. For instance, in the former case we simply select the observation x that maximizes the expected utility. For the latter case, however, we will successively select an observation x in a similar manner but now, each observation and its corresponding objective value will augment our dataset. This means that the observations we choose will impact the remainder of the optimization process and additional consideration will be required when making sequential decisions.

Intuitively, we can recognize that the closer decisions are to termination, the easier it is to make them optimally since there are less (if any) future decisions that will rely on their outcomes. Thus, we will use this to our advantage and design optimization policies in *reverse*. Specifically, we will initially reason about the last decision and move backwards from there, defining optimal behavior as we go.

Construction of Optimization Policies

During the construction of optimization policies, we assume that we are constrained by a pre-defined and fixed search budget representing the maximum number of observations we can make. This assumption is common in practice and is also convenient when analyzing policy design since we can disregard the question of termination. It will also imply that each observation has a constant acquisition cost that may not always be reasonable. We will consider acquisition costs and the question of termination later on.

Under the constraint of a fixed search budget, we can analyze policies using the number of future observations before termination. The problem can be defined as follows: given a set of data, how should we select our next evaluation point when exactly τ observations remain before we terminate?

- The **decision horizon** denoted by τ indicates the number of remaining observations before termination.

Before we analyze optimization policies, we will define some notation used by Garnett.

- 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.

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

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

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 $\{x_2, \ldots, x_\tau\}$ and corresponding values $\{y_2, \ldots, y_\tau\}$ of future observations.

Using the expected terminal utility to evaluate a candidate observation at location x, we can extend the treatment of isolated, single decisions:

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

Again, we define an optimization policy that maximizes the expected utility:

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

Although the ideas discussed above are theoretically simple, we must consider how we will actually compute the expected terminal utility across future observations. To illustrate this, Garnett provides the unwieldy expression for the expectation over future observations:

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

As one can see, explicitly computing this integral is cumbersome and we will instead prefer computing this expression under the assumption that *all future decisions are made optimally* (Bellman's Principle of Optimality). We will discuss how this type of analysis obtains the optimal optimization policy later on.

For now, we will use **backward induction** to determine the optimal behavior if there is only one observation remaining and then continue backwards inductively to consider increasingly long horizons. Below is the expression 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:

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

Notice that this is simply the difference between the expected terminal utility and the utility of the existing dataset. Additionally, this notation is similar to what is used for acquisition functions, enabling us to define the optimal optimization policy using acquisition functions.

Base Case: One Observation Remaining

To develop our reasoning using backward induction, we will start with the *base case* where there is only one observation remaining in the horizon; there are $\tau = 1$ steps left before termination. In this case, the terminal dataset $\mathcal{D}_\tau$ is the current dataset augmented with one additional observation. Since this is essentially the single-decision case, we can use the framework we developed previously.

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

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

Under the framework we developed previously, the optimal observation is the one that maximizes the expected marginal gain in utility:

$$x \in \arg\max_{x' \in \mathcal{X}} \alpha_1(x';\mathcal{D})$$

This results in the returned dataset having expected utility:

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

Thus, $\alpha_\tau^*(\mathcal{D})$ denotes the **value** of the dataset and represents the expected increase in the dataset's utility if we start with an arbitrary dataset $\mathcal{D}$ and continue optimally for τ more observations. This is key in future 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 deriving the inductive case, we will develop a special case where there are two observations remaining; there are $\tau = 2$ steps left before termination. This will help us when we develop the inductive case.

Once again, suppose we have an arbitrary dataset $\mathcal{D}$ but now we need to decide where the next to last observation x should be. First, we will consider the expected increase in terminal utility after two observations:

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

From the definition of expectation, the expectation above should require that we marginalize the observation y, the final observation x_2, and its corresponding value y_2. Fortunately, we can apply Bellman's Principle of Optimality to assume that all future behavior will be optimal. This assumption allows us to simplify how we approach the final decision x_2.

We start by redefining the two-step expected gain in utility α_2 in terms of the single-step case α_1, a function we understand much better. The *two-step difference* in utility can be expressed as a *telescoping sum*:

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

By redefining the two-step expected gain, we separate the expected increase in terminal utility after two observations into two terms: (1) the expected increase after the first observation, called the *expected immediate gain* and (2) the expected additional increase after the final observation, called the *expected future gain.* This is shown below.

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

Although it is not evident how to address the second term corresponding to the expected future gain, we can reason about this using our analysis of the base case. 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 $\alpha_1^*(\mathcal{D}_1)$. As this is a quantity we can compute, under the assumption of optimal future behavior, we can express the expectation with the current and known observation y:

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

Again, the optimal next-to-last observation will maximize the expected gain:

$$x \in \arg\max_{x' \in \mathcal{X}} \alpha_2(x'; \mathcal{D})$$

The optimal next-to-last observation will provide an expected terminal utility given by:

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

The analysis above shows that we are able to achieve optimal behavior and can 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: τ Observations Remaining

With the base and special cases under our belt, we can now analyze the general inductive case. Recall that the inductive case will be similar to the special case with $\tau = 2$ steps left before termination.

Let τ be an arbitrary decision horizon. Assume that we can compute the value of any dataset with a horizon of $\tau - 1$. Suppose that we have an arbitrary dataset $\mathcal{D}$ and must decide where the next observation should be made. Here, we will use backwards induction to do this optimally and then show how to compute the value of a returned dataset.

The τ-step expected utility gain from observing x is given by:

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

We want to maximize the expected utility gain above and we start by redefining it (similar to what we did in the special case) using *shorter-horizon quantities* via a telescoping sum:

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

As long as we know the corresponding value y and thus $\mathcal{D}_1$, we can use the assumption of future optimal behavior to attain an expected further utility gain of $\alpha^*_{\tau-1}(\mathcal{D}_1)$. We can compute this quantity via the inductive hypothesis. It can be expressed as follows:

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

Next, we maximize the expected utility gain at step τ to choose the optimal decision and compute the τ-step value of the dataset:

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

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

This result concludes our analysis and demonstrates we can behave optimally for a horizon of τ given an arbitrary dataset $\mathcal{D}$, and compute its value.

Bellman's Principle of Optimality and the Bellman Equation

The fundamental assumption of our analysis to make optimal, sequential decisions is the assumption that all future behavior will be optimal. This assumption is known as **Bellman's Principle of Optimality**. Specific to optimization, the principle states that we will always act optimally to maximize the expected terminal utility given the available data.

Mathematically, the **Bellman equation** forms the recursive definition of the value in terms of the value of future data. To obtain this equation, we substitute the expected utility gain expressed using shorter-horizon quantities into the expression maximizing the expected utility at step τ:

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

This equation is a key result in the theory of optimal sequential decision-making and 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.

Thus, to create a sequence of optimal decisions, we select 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 illustrate said barrier, recall the expression for the expected utility gain when $\tau = 2$ and under the assumption of optimal behavior:

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

As noted by Garnett, the second term appears to be a standard expectation over the random variable y but actually evaluating this requires us to solve a *non-trivial* global optimization problem outlined below:

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

Even more problematic, even if we were only considering a horizon with two decisions remaining, we would need to solve a *dually nested* global optimization problem (no simple feat!). Correspondingly, if the horizon is τ then from the recursively defined optimal policy, we would need to solve τ-nested optimization problems to attain the optimal decision. Garnett illustrates this issue using the (temporarily) compact notation below:

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

The notation above demonstrates that the design of any optimal decision requires repeated maximization over the domain plus expectation over unknown observations until we reach the horizon. Figure 5.4 in Garnett's text provides a visual of this problem as a decision tree where each unknown quantity will contribute a significant branching factor. Additionally, it also shows that computing the expected utility at x requires a traversal of the entire tree.

We will use this structure to compute the cost of the optimal policy, specifically the running time for a naïve implementation via exhaustive traversal of the decision tree. Suppose that for each *maximization* we use an off-the-shelf optimization routine and for each *expectation* we use a numerical quadrature routine.

If we allow n evaluations of the objective per call to the optimizer 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. Thus, the 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 will discuss how to avoid such costs by *approximating* the optimal policy.

5 Approximation of the Optimal Policy

In the previous section, we determined that the computational effort required to obtain the optimal policy becomes intractable due to the exponential growth of the running time with respect to the horizon. However, as mentioned, we can use approximation schemes to compute the optimal policies. These schemes are extensively studied in the field of **approximate dynamic programming**.

Recall the *intractable* optimal expected marginal gain:

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

The most difficult part of the expression above is the recursively defined future value α^*. To bypass this, we can substitute in a tractable approximation. Although this will induce a suboptimal, approximate policy it is still rationally guided. Specifically, we will review two approximation schemes common in Bayesian optimization: *limited lookahead* and *rollout*.

5.1 Limited Lookahead

The **limited lookahead** approximation method, as the name suggests, limits how far into the future we look when making decisions. Specifically, we restrict how many future observations will be considered in each decision. This approach is practical since we move the decisions closer to "termination" making them easier to compute. Under this premise, we will develop a family of approximations to the optimal policy by "artificially" limiting the horizon considered during optimization to a feasible maximum $\ell < \tau$. Thus, the approximation is expressed as:

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

When maximizing, we will act optimally under the *incorrect but convenient assumption* that only ℓ observations remain. Effectively, we assume $u(\mathcal{D}_\tau) \approx u(\mathcal{D}_\ell)$. This approximation method is considered to be *myopic* since we limit ourselves to only considering the next few observations on the horizon instead of considering 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. A special case is the **one-step lookahead** that aims to successively maximize the expected marginal gain from one more observation (α_1) making it the most efficient lookahead approximation.

Considering computational complexity, we are able to bound the effort required when we limit the horizon. The effort is bounded to be 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.

5.2 Rollout

Recall that in our theoretical analysis, the optimal policy evaluates a candidate observation by simulating the rest of the optimization. While this is rational, it's also intractable. The **rollout** approximation approach emulates the structure of the optimal policy but instead uses a tractable *suboptimal* policy to simulate future decisions.

Here is an outline of the rollout approximation routine:

- Given another observation (x, y), we will use an inexpensive *base* or *heuristic* policy to simulate a reasonable but likely suboptimal realization of the next decision x_2.
- Then, we take the 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, take the expectation with respect to the unknown value at x_3, and so forth until we reach the decision horizon.

Since this approach does not lead to branching in the decision tree, we avoid the expensive subtree that would typically be required by the optimal policy. Instead, we use the terminal utilities from the *pruned tree* to estimate the expected marginal gain α_τ that we maximize as a function x. Note that while there are no restrictions on the design of the base policy, since we are trying to improve efficiency, it should be fairly efficient.

- A common and effective choice for the base policy, provided by Garnett, is to simulate the future decisions using the *one-step lookahead* approach. Then, if we use off-the-shelf optimizers and quadrature routines to traverse the resulting, rollout decision tree, the computational complexity is $\mathcal{O}(n^2 q^\tau)$.

This approach is considerably faster than the exact 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 this approximation approach itself extremely flexible.

For example, Garnett proposed that we could combine the rollout and limited lookahead schemes to define approximate policies with *tunable running time.* Additionally, we could use a base policy that designs all of the remaining observations *simultaneously.* By ignoring the dependence between these decisions, we can achieve a computational advantage while still being aware of the horizon. These so-called **batch rollout** schemes have been known to work well in Bayesian optimization. Consider the fact that 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 operated under the assumption that our optimization policies are constrained by a pre-defined and constant search budget. Although this is a common setup, it is not universal and in some scenarios we might want to leverage our changing beliefs about the objective to help us *dynamically* decide when termination is the optimal decision.

The notion of dynamic termination is valuable when we want to account for the cost of acquiring data during optimization. For instance, if the cost of gaining more observations varies across the search space then it does not make sense to define our budget based on the number of evaluations. Rather, we ought to account for the acquisition costs in the utility function to explicitly reason about the *cost-benefit tradeoff* for each observation. If the cost of acquiring an additional point outweighs its expected benefit, we can seek to terminate the optimization process.

6.1 Modeling Termination as a Decision

To model dynamic termination, we will modify the case for sequential decisions that assumed a pre-defined and constant search budget. Now, we will allow ourselves to terminate the optimization process at any point, as we see fit. Suppose we are performing optimization and have already acquired a dataset $\mathcal{D}$. However, unlike the known-budget case, in addition to deciding where to sample next we also 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).

While we can derive the optimal optimization policy for dynamic termination using induction, we must address the issue of the base case which represents the "final" decision. The problem is that the case breaks down now that we allow for a non-terminating sequence of decisions. To address this, we will assume that there is a fixed, known-upper bound $\tau_{\max}$ on the total number of observations we can make. Once we reach this bound, the optimization process will terminate regardless of our progress.

Under the assumption of an upper-bound on the decision horizon, the inductive argument we derived for the known-budget case applies although we need to re-define how to compute the value of the termination action. Luckily, this is clear: since termination (1) does not augment our dataset and (2) means no further actions can be taken, the expected marginal gain from termination will always be zero:

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

Thus, apart from substituting $\mathcal{A}$ for $\mathcal{X}$ in the previous set of derived expressions, we have obtained the optimal policy for the dynamic termination case. See Figure 5.8 in Garnett's text for an illustration of one-step lookahead with the option to terminate. In that example, the 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 and could be unknown or stochastic but we will examine this in other sets of notes and in Garnett's Chapter 11.

To reason about observation costs, location-dependent or otherwise, we must change our approach to the utility function. One natural approach would be to select a utility function that exclusive measures the returned dataset's quality (ignoring the costs incurred to acquire the dataset). 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.

Below, we adjust the data utility to account for the acquisition cost of observations. Typically, the acquisition costs are additive so the cost of acquiring an arbitrary dataset $\mathcal{D}$ is:

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

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

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

In real-world applications, such as the ones shown in the *Introduction to Bayesian Optimization* notes, the cost of acquiring observations is usually known. For example, when applying Bayesian optimization to molecule or protein discovery, the cost of acquisition would likely include the cost of performing experiments to evaluate the molecule or protein.

7 References

[1] Roman Garnett. 2023. *Bayesian optimization.* Cambridge University Press, Cambridge, United Kingdom ;

[2] Peng Liu. 2023. *Bayesian optimization : Theory and practice using python.* Apress, New York, NY.