Utility Functions

Drew Gjerstad

Contents

1 Introduction

In the Bayesian decision theory notes, one of the mechanisms crucial to making optimal decisions was the **utility function**. The utility function evaluates the quality of the dataset returned by the optimization process based on the preferences we express via the utility function itself. Given a model of the objective, the utility function allows us to derive the optimal policy very simply: maximize the expected utility of each observation and consequently maximize the expected utility of the returned dataset. Therefore, we can identify very clearly what is required of us: define a model that is consistent with our beliefs about the objective and define a utility function that expresses our preferences across outcomes.

However, while we can clearly identify *what* we need to do, the *how* we do it is not exactly trivial. For instance, as Garnett points out, our beliefs and preferences are innately internalized which makes it difficult to break them down into mathematical expressions. Luckily, there are avenues we can take to mathematically express our beliefs and preferences using surrogate models (i.e., Gaussian processes) and utility functions (i.e., expected improvement), respectively. In these notes, we focus on how to define utility functions that are consistent with our preferences, including a review of utility functions frequently used in Bayesian optimization. Many of these utility functions' underlying motivation often contribute to novel approaches. Finally, we follow Garnett's position when delving into utility functions: while Gaussian processes are common in Bayesian optimization, we will not assume that the surrogate model is a Gaussian process.

2 Expected Utility of the Final Recommendation

Simply put, the purpose of optimization is to search the space of candidates for the best candidate that we implicitly decide to use, often in another routine. Recall the applications provided in the *Introduction to Bayesian Optimization* notes: when applying (Bayesian) optimization for AutoML purposes, we are searching the space of candidate *hyperparameters* that will be used to train a neural network—in hopes that when the network is re-evaluated its performance (i.e., classification accuracy) improves. However, as we have seen, the best candidate is typically found closer to the termination of the optimization process meaning that for the most part, the dataset that we acquire during optimization is only used to guide us towards the best candidate.

Clearly, the selection of the “best candidate” can be regarded as another **decision**. Thus, if the goal of optimization is to guide us to the optimal final decision then the optimization policy should be designed to maximize the expected utility of the final decision. In this section, we will focus on (1) defining the “final recommendation” as a decision, (2) selecting an action space to move across, and (3) selecting a utility function to use when evaluating candidate observations.

2.1 Defining the Final Recommendation Decision

We stated previously that the best candidate is typically viewed as the “final” recommendation for another system and can thus be regarded as a decision. Here, we will define the final recommendation decision in a mathematical manner such that it becomes evident how the utility function aids this decision. Note that in some literature, the terms *final recommendation* and *terminal recommendation* are used interchangeably.

First, suppose that our optimization process returned an arbitrary dataset $\mathcal{D} = (\mathbf{x}, \mathbf{y})$. Then, suppose that we aim to use this returned dataset to recommend a candidate $x \in \mathcal{X}$ that will be used in another, external routine. In this external routine, the performance is quantified by the underlying objective function value denoted $\phi = f(x)$. Once again, this recommendation is regarded as a **decision under uncertainty** about the objective value that is informed by the posterior predictive distribution denoted by $p(\phi|x, \mathcal{D})$.

Recall that in order to completely define the decision problem, we must identify the action space $\mathcal{A} \subset \mathcal{X}$ for our recommendation and identify a utility function $v(\phi)$ to evaluate a recommendation post-hoc based on the objective value ϕ. Once these are defined, a **rational** recommendation should maximize the expected utility:

$$x \in \underset{x' \in \mathcal{A}}{\arg\max}\, \mathbb{E}\left[v(\phi')|x', \mathcal{D}\right]$$

Notice that the recommendation's expected utility above is only dependent on the returned dataset from the optimization process. This property brings about a natural utility that we can use in optimization. The **natural utility** function computes the expected quality of an optimal final recommendation given the returned dataset $\mathcal{D}$:

$$u(\mathcal{D}) = \max_{x' \in \mathcal{A}} \mathbb{E}\left[v(\phi')|x', \mathcal{D}\right]$$

Furthermore, the recommendation's expected utility will also not depend on the optimal recommendation itself (i.e, x') since we are selecting the maximal expected objective function value. Thus, since we are computing the expectation of the objective function value given the candidate and returned dataset, the expected utility will not depend on the optimal recommendation's objective value either.

Referencing the depiction of the sequential decision tree in Figure 5.4, Garnett notes that the utility function will effectively "collapse" the expected utility of a final decision into a utility of the returned dataset. This means that we are able to select the action space and utility function for the final recommendation based purely on the problem at hand. We will consider Garnett's advice for these selections next.

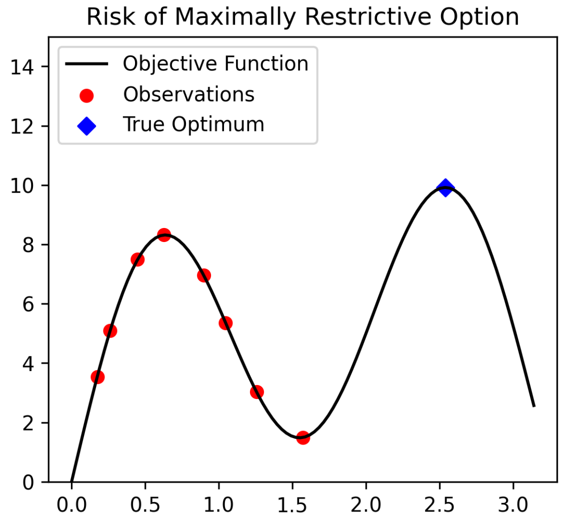
2.2 Selecting an Action Space

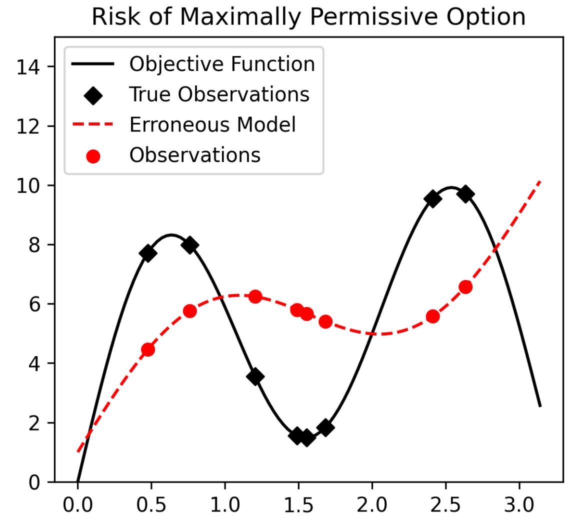
We need to select an action space for our recommendation, denoted by $\mathcal{A} \subset \mathcal{X}$. Let's take a look at two extreme options with one being maximally restrictive and the other being maximally permissive.

- The **maximally restrictive** option restricts our recommendation choices to only the points visited during optimization, $\mathbf{x}$. Although this option will ensure that we have at least some knowledge of the objective function at our recommended point, it does not allow for any exploration. In other words, we may not have visited the best point so it will not be contained in $\mathbf{x}$ and we are therefore unable to recommend it.
- The **maximally permissive** option defines the action space to be the entire domain $\mathcal{X}$. However, opting to have the entire domain be our action space will require us to have faith in the objective function model's belief, particularly when recommending an unvisited point. In other words, while it gives us more freedom to explore, it also means we have to be careful about where our model has higher uncertainty.

The plots on the following page provide examples of these two extreme options. On the left, the plot shows an example of the maximally restrictive option that only allows visited points to be recommended. Clearly, our observations denoted by the red markers do not include the true optimum denoted by the blue marker.

On the other hand, the plot to the right shows an example of the maximally permissive option that denotes the entire domain to be the action space. As shown by the dashed red line, our model is very erroneous and does not capture the underlying objective. In particular, we could end up recommending points that have the highest model value but not the highest objective model. While these examples themselves are extreme and a bit exaggerated, they represent the risks present if we use either extreme to select an action space.





While these are the extreme options that one could take to select an action space, there have been some more reasonable suggestions in the literature. One reasonable suggest is shown in Osborne et al. and is essentially a compromise between the two extremes. Specifically, the final recommendation choice is restricted to points where the objective value is known within some acceptable tolerance. Osborne et al. defined a parametric and data-dependent action space (with form given below) to achieve this compromise.

$$\mathcal{A}(\varepsilon; \mathcal{D}) = \{x | \text{std}\,[\phi | x, \mathcal{D}] \leq \varepsilon\}$$

In the expression above, ε denotes a threshold for the highest amount of acceptable uncertainty in the objective function value. This approach should, for the most part, avoid issues of recommending points at locations where the objective function is not sufficiently known while allowing for some exploration beyond the points visited during optimization.

2.3 Selecting a Utility Function

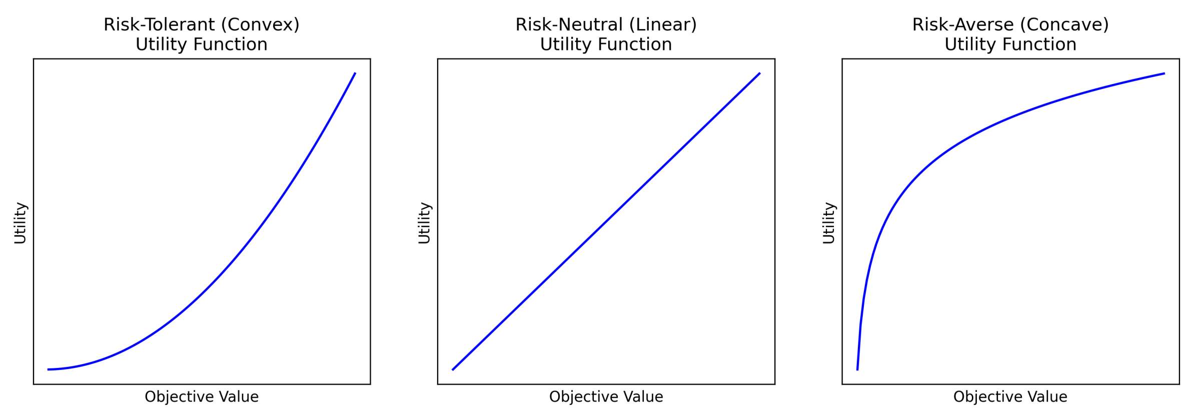
In addition to selecting an action space for our recommendation, we also need to select a utility function $v(\phi)$ that will evaluate a recommendation x after we observe its corresponding objective function value ϕ. For our purposes, we have been focusing on maximization so the utility function should be **monotonically increasing** in ϕ. This means that as the objective value ϕ increases, its utility $v(\phi)$ should also increase.

- Remember that if the decision problem calls for it, it is fairly trivial to change the focus and setup from maximization to minimization, and vice versa.

While we know that the utility function should always be increasing in ϕ, there is still the question of what shape the function ought to assume. The answer to this question will depend on our **risk tolerance**. Risk tolerance refers to the tradeoff between potentially obtaining a higher expected value but with greater uncertainty thereby imposing some risk in our recommendation. Alternatively, we could have a lower expected value with lower uncertainty making it a safer recommendation.

- A **risk-tolerant** (convex) utility function will allow for more risk to potentially attain greater reward (i.e., higher expected value but higher uncertainty).
- A **risk-neutral** (linear) utility function is indifferent between points with equal expected value without regard to their uncertainty.
- A **risk-averse** (concave) utility function will avoid risk and err towards lower risk even if it means a lower reward (i.e., lower uncertainty but lower expected value).

The plots below illustrate the shape the utility function would take based on the risk tolerance.



The most simple and commonly used utility function in Bayesian optimization is a risk-neutral, linear utility function:

$$v(\phi) = \phi$$

When the utility function is linear, the expected utility from recommending x is simply the posterior mean of ϕ:

$$\mathbb{E}\left[v(\phi)|x, \mathcal{D}\right] = \mu_{\mathcal{D}}(x)$$

Recall that the risk-neutral, linear utility function does not consider the uncertainty in the objective function when making a decision. While this type of utility is simple and therefore computationally convenient, it may not truly be consistent with our preferences.

In addition to expressing our risk preferences, we can reason about them using the **certainty equivalent**. The certainty equivalent is an objective value that corresponds to a (hypothetically) risk-free alternative recommendation to which our preferences would be indifferent. For instance, suppose that we have a risky potential recommendation x where we do not know its true corresponding value of the objective function. Then, the certainty equivalent for x is an objective function value ϕ' such that

$$v(\phi') = \mathbb{E}\left[v(\phi)|x, \mathcal{D}\right]$$

When using a risk-neutral utility, the certainty equivalent of some point x is its expected value: $\phi' = \mu_{\mathcal{D}}(x)$. Therefore, we would only abandon one recommendation in favor of another if the if the latter had a higher expected value without considering risk. Alternatively, we may want to express our risk-aware preferences using **nonlinear** utility functions.

3 A Note on Nonlinear Utility Functions

As mentioned in the previous section, we may wish (or even truly need) to express our risk-aware preferences using nonlinear utility functions. To illustrate this, let's consider a scenario where our preferences lean more towards risk aversion. With these preferences, we may accept a recommendation point with a lower expected value if it also results in less risk. From our previous discussion on the shape of utility functions based on risk tolerance, we should express these preferences using a concave utility function. Garnett uses Jensen's inequality as an example of a utility function expressing risk averse preferences:

$$v(\phi') = \mathbb{E}\left[v(\phi)|x, \mathcal{D}\right] \leq v\left(\mathbb{E}\left[\phi|x, \mathcal{D}\right]\right) = v(\mu_{\mathcal{D}}(x))$$

With this utility function, the certainty equivalent of a risky recommendation will be less than its expected value. On the other hand, we could express risk-seeking preferences using a convex utility function. In such a case, the certainty equivalent of a risky recommendation will be greater than its expected value. Then, our preferences will implicitly push toward gambling. For obvious reasons, literature on economic and decision theory have proposed several risk-averse utility functions although these (and risk-seeking utility functions) are not typically used in Bayesian optimization. Note that these types of utilities can be useful in certain settings, especially where risk neutrality is questionable.

Before continuing, let's review a natural approach to quantify the risk associated with a recommendation of an uncertain value ϕ. To do this, we simply use its standard deviation:

$$\sigma = \mathrm{std}\left[\phi|x, \mathcal{D}\right]$$

Then, we can establish our preferences for potential recommendations using a weighted combination of a point's expected reward and its risk:

$$\mu_{\mathcal{D}}(x) + \beta\sigma = \mu + \beta\sigma$$

where β represents a tunable risk-tolerance parameter with $\beta < 0$ penalizing risk (inducing risk-averse behavior), $\beta > 0$ rewarding risk (inducing risk-seeking behavior), and $\beta = 0$ inducing risk neutrality. This framework serves as the base for two common utility functions in Bayesian optimization: *simple reward* and *global reward*. We will discuss these next.

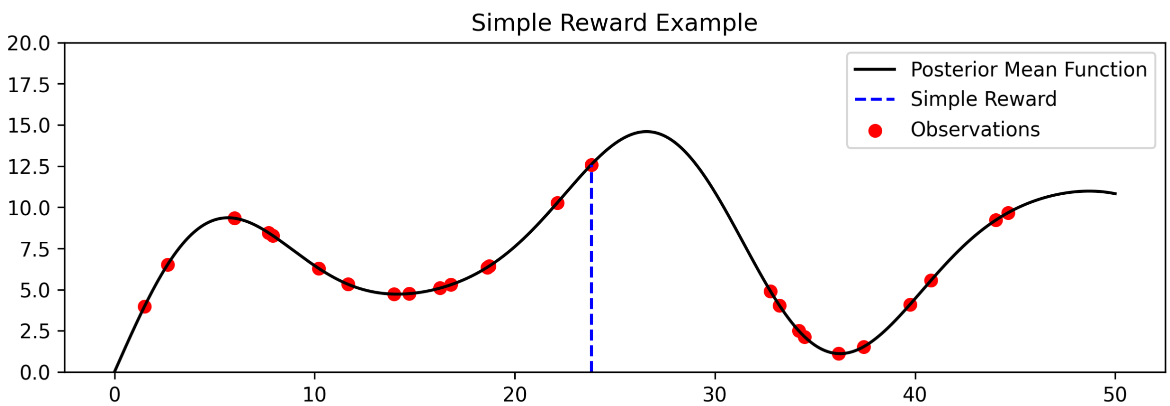
4 Common Utility Functions

In this section, we review utility functions common in Bayesian optimization literature and practice. This is by no means an exhaustive review of each and every utility function out there but rather an exposition of the utility functions frequently used in Bayesian optimization.

4.1 Simple Reward

The **simple reward** is named as such since it assumes that we are risk-neutral and that we limit the action space to only points visited during optimization. Suppose an optimization process returned dataset $\mathcal{D}(\mathbf{x}, \mathbf{y})$ to guide the final recommendation made using the risk-neutral utility function $v(\phi) = \phi$. Then, the expected utility of the optimal recommendation is:

$$u(\mathcal{D}) = \max \mu_{\mathcal{D}}(\mathbf{x})$$



Garnett points out one *technical caveat*: if the returned dataset is empty, then the maximum degenerates and will have that $u(\emptyset) = -\infty$. Additionally, in the special case where we have exact observations (i.e., $\mathbf{y} = f(\mathbf{x}) = \boldsymbol{\phi}$), the simple reward will reduce to the maximal objective value seen during the optimization routine:

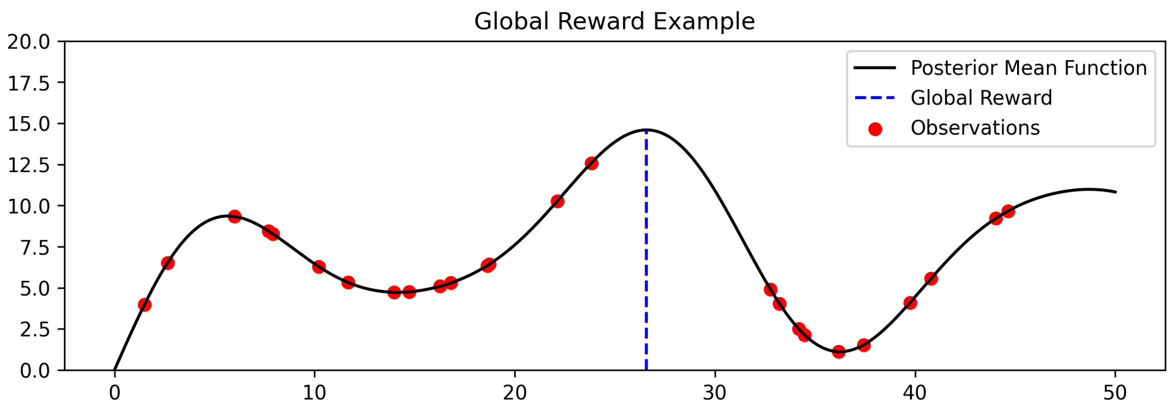
$$u(\mathcal{D}) = \max \boldsymbol{\phi}$$

4.2 Global Reward

The **global reward** is named as such since, similar to the simple reward, assumes that we are risk-neutral but now we allow the action space to be the entire domain $\mathcal{X}$. Using the same notation as before, the expected utility of the optimal recommendation is the global maximum of the posterior mean:

$$u(\mathcal{D}) = \max_{x \in \mathcal{X}} \mu_{\mathcal{D}}(x)$$

Consider that by expanding the action space to include the entire domain, which includes points where we are inherently uncertain about the objective value at, this often leads to a different and potentially more risky recommendation.



4.3 A Tempting, Nonsensical Alternative to Simple Reward

Here we discuss the tempting, but not entirely rational alternative to the simple reward brought up by Garnett. Specifically, we are referring to an alternative utility function that is deceptively similar to the simple reward: the maximum noisy observed value in the dataset, denoted

$$u(\mathcal{D}) = \max \mathbf{y}$$

As Garnett notes, if we are dealing with *exact* observations of the objective function then this utility reduces to the simple reward defined above. However, such reduction does not apply when dealing with inexact or noisy observations. In those cases, this utility is "rendered absurd". Furthermore, this absurdity is more prevalent in situations where an observed (noisy) maximum value reflects noise rather than actual optimization progress. Finally, we must add that if the signal-to-noise ratio is relatively high (i.e., the noise is not extreme) then this utility function can serve as an approximation to the simple reward. See Garnett Figure 6.2 for an extreme but helpfully illustrative example.

4.4 Cumulative Reward

The **cumulative reward** differs from the simple and global rewards in that it is based on the objective value of all points in the dataset. In particular, it rewards, and thus encourages, acquiring points with high *average* objective values. Compared to the simple and global rewards which focus on the idea that our goal is to find the best point from the search space, the cumulative reward can be useful when the objective values of each and every point is important (i.e., if our optimization routine is responsible for controlling a critical, external system). For a dataset $\mathcal{D} = (\mathbf{x}, \mathbf{y})$, the cumulative reward is given by the sum of observed objective values:

$$u(\mathcal{D}) = \sum_i y_i$$

Garnett mentions a notable use of cumulative reward: *active search*. **Active search** is a mathematical model of scientific discovery wherein we select points for evaluation in a successive manner, aiming to identify points in a rare, value class $\mathcal{V} \subset \mathcal{X}$. When we observe a point $x \in \mathcal{X}$, it will yield a binary observation denoting whether or not the point is in the class (i.e., $y = [x \in \mathcal{V}]$). The motivation behind using cumulative reward is that during the search, we hope to find as many items in the desired class as we can.

4.5 Information Gain

The idea of **information gain** is that a dataset will be evaluated based on the (quantitative) amount of information it provides about a random variable of interest, where we prefer datasets that contain more knowledge about the random variable of interest. This approach is referred to as an *information-theoretic* one and is derived from the domain of *information theory*. Finally, it is an alternative approach to the simple, global, and cumulative rewards that evaluate a dataset based on the objective values it contains.

Let ω be a random variable of interest we want to gain information about as we observe data. As Garnett notes, the choice of ω is open-ended and dependent on the application at hand. However, some natural choices include the location of global optimum, x^*, and the maximum objective value, f^*. Below, we follow Garnett's notation for quantifying the information contained by a dataset.

First, we quantify our initial uncertainty about ω using the differential *entropy* of its prior distribution, $p(\omega)$:

$$H[\omega] = -\int p(\omega) \log p(\omega) \, \mathrm{d}\omega$$

Then, the *information gain* provided by a dataset $\mathcal{D}$ is given by the difference in entropy between the prior and posterior distribution:

$$u(\mathcal{D}) = H\left[\omega\right] - H\left[\omega|\mathcal{D}\right]$$

where $H\left[\omega|\mathcal{D}\right]$ is the differential entropy of the posterior. Note that Garnett's notation is not standard and he notes one particular caveat with it. The notation for the *conditional entropy* of ω given $\mathcal{D}$ is the same as the notation used here for the differential entropy of the posterior, $H\left[\omega|\mathcal{D}\right]$. However, for our context, this is fine and as noted by Garnett, if necessary we will denote the conditional entropy with an explicit expectation: $\mathbb{E}\left[H\left[\omega|\mathcal{D}\right]|\mathbf{x}\right]$.

Additionally, Garnett also raises a potential point of confusion due to an alternative definition of information gain used in literature: the *Kullback-Leibler (KL) divergence* between the posterior and prior distributions.

$$u(\mathcal{D}) = D_{\mathrm{KL}}\left[p(\omega|\mathcal{D})||p(\omega)\right] = \int p(\omega|\mathcal{D}) \log \frac{p(\omega|\mathcal{D})}{p(\omega)}\, \mathrm{d}\omega$$

These expressions allow us to quantify the information a dataset contains based on how our prior belief in ω changes after it is collected. The Kullback-Leibler divergence definition for information gain has some added convenience compared to the definition of information gain that came before it. In particular, the KL divergence expression of information gain is *invariant* to reparameterization of ω and is always nonnegative. Such properties are useful if unexpected observations are collected since such observations could cause the previous definition to become negative.

Luckily for us, the connection between these two definitions for information gain are surprisingly strong, especially for sequential decision making. Specifically, the expectation with respect to observations are equal which means that when we maximize the expected utility using either definition, it will lead to the same decisions.

4.6 Comparison of Utility Functions

Now that we have reviewed multiple utility functions that (quantitatively) evaluate a dataset returned by an optimization process in some way, we can examine the subtle differences between their respective approaches. First, we will compare the simple reward to the other utility functions that we presented.

The simple reward is the most common utility function in Bayesian optimization due to its use in conjunction with the expected improvement (EI) acquisition function. A key distinction between this utility function and the others is that its approach uses only the *local* properties of the objective function posterior to evaluate the data. As such, this "locality property" is both rational from the conceptual perspective and convenient from the computational perspective.

Alternatively, other utility functions use the *global* properties of the objective function. For instance, recall that the global reward allows the action space to be the entire domain so that utility function will take the entire posterior mean function into account. Furthermore, it means that this utility function is able to recommend a point after termination, even if it was unobserved during optimization.

Moving away from the objective value-based approaches, information gain makes use of information theory and evaluates data based on the change in knowledge regarding the location or value of the optimum. Note that this approach will consider the posterior entropy of the location or value of the optimum so it still uses a global property.

Finally, Garnett notes that since these utility functions use either the local or global properties of the posterior, there can be disagreement between the simple reward and other utility functions. Garnett provides some examples about datasets that have a good global outcome but poor local outcome (and vice versa). It is recommended to page through the last few pages of Chapter 6 to read about these examples and connect the concepts to the visual representations of these examples.

5 Relationship Between Model of Objective Function and Utility Function

Having discussed several common utility functions in Bayesian optimization, one hopefully recognizes the relationship between the model of the objective function and utility function: most of the utility functions are dependent on the underlying model of the objective function.

For example, the first two utility functions (simple and global reward) are both defined based on the posterior mean function $\mu_{\mathcal{D}}$. Information gain relies on the posterior belief about location and value of the optimum $p(x^*, f^*|\mathcal{D})$. Both the posterior mean and the posterior beliefs are byproducts of the posterior distribution of the objective function.

However, as Garnett explains, there are approaches to mitigate the dependence of the utility function on the model of the objective function. One approach is a computational mitigation strategy using *model averaging*. Recall that model averaging is a process where we marginalize the model with respect to the model posterior. Another approach is to define model-agnostic utility functions that are based on the data alone (i.e., no relation to a model) but this is fairly limited under the assumption that the utility should be sensible.

- One example of the latter approach is the cumulative reward since it is only defined based on the observed values $\mathbf{y}$.
- Another example of the latter approach is the maximum function value utility but as noted previously, that utility's rationale diminishes if observation values are plagued by noise.

Alternatives defined in the same manner as the examples above often have similar difficulties: noise will bias many natural measures such as order statistics (i.e., minimum, maximum, etc.) of the observations. In particular, while for additive noise with zero mean, we would expect that the noise would not impact the cumulative reward at all but it will still affect the aforementioned order statistics.

6 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)