Utility Functions

Drew Gjerstad

Contents

1 Introduction **2**

2 Expected Utility of the Final Recommendation **2**
2.1 Defining the Final Recommendation Decision 3
2.2 Selecting an Action Space 3
2.3 Selecting a Utility Function 4

3 A Note on Nonlinear Utility Functions **5**

4 Common Utility Functions **6**
4.1 Simple Reward 7
4.2 Global Reward 7
4.3 A Tempting, Nonsensical Alternative to Simple Reward 7
4.4 Cumulative Reward 8
4.5 Information Gain 8
4.6 Comparison of Utility Functions 10

5 Relationship Between Model of Objective Function and Utility Function **10**

6 References **12**

1 Introduction

In the notes discussing **Bayesian decision theory**, one of the key components used to make optimal decisions was the *utility function*. Recall that the utility function evaluates the quality of the dataset returned by the optimization process, based on our preferences across outcomes expressed in the utility function. Provided we have a model of the objective function, the utility function enables us to determine the optimal policy by simply maximizing the expected the utility of each observation thereby maximizing the expected utility of the returned dataset. From this viewpoint, we only need to define an model that is consistent with our beliefs regarding the objective and a utility function that expresses our preferences.

Take into account that neither defining a model for the objective function nor defining a utility function is a trivial task. As Garnett notes, breaking down our beliefs and preferences (concepts that are rather internalized in humans) to mathematical expressions is difficult. However, we do have avenues that we can take to mathematically express our beliefs and preferences such as surrogate models (i.e., Gaussian processes [GPs]) and utility functions (i.e., Expected Improvement [EI]). In these notes, we will focus on defining utility functions that are consistent with our preferences over outcomes. This will include examining several commonly used utility functions whose underlying motivation often contributes to novel approaches. Please note that while we often use Gaussian processes regularly in other notes, we will follow Garnett's notation which does not assume the surrogate model is a Gaussian process.

2 Expected Utility of the Final Recommendation

The point of optimization is to search the space of candidates in order to find the best candidate that we implicitly decide to use, likely in another procedure. For instance, as shown in the applications of Bayesian optimization section of the "Introduction to Bayesian Optimization" notes, our goal may be to find the next molecule or protein candidate that satisfies certain properties. However, as we have seen with optimization, the best candidate is often found later on in the optimization process which means that for the most part, the data (chosen candidate observations) acquired during the routine is only used to help guide us towards the best candidate.

From Garnett, it is obvious that selecting this "best candidate" for use in another procedure can be considered a *decision*. Therefore, if the goal of performing optimization across a search space is to guide the final decision then the optimization policy should maximize the expected utility of the final decision.

2.1 Defining the Final Recommendation Decision

Suppose that our (pre-run) optimization process returned an arbitrary dataset $\mathcal{D} = (\mathbf{x}, \mathbf{y})$. Then, suppose that our goal is to use the returned dataset to recommend a candidate $x \in \mathcal{X}$ for use in another procedure where the performance is determined by the underlying objective function value denoted $\phi = f(x)$. Aforementioned, this recommendation is considered to be a *decision under uncertainty* about the objective function value that is informed by the predictive distribution denoted $p(\phi|x, \mathcal{D})$.

Similarly to previous notes, fully defining the decision problem requires us to identify the action space, $\mathcal{A} \subset \mathcal{X}$, for our recommendation and a utility function $v(\phi)$ that will evaluate a recommendation post-hoc based on the objective function value ϕ. Having defined these, a recommendation considered to be *rational* should maximize the expected utility:

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

Since the expected utility of the recommendation is only dependent on the dataset returned by the optimization process, it leads to a natural utility that we can use in optimization. This *natural* utility function computes the expected quality of an optimal final (terminal) recommendation given the returned dataset:

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

Note that the expected utility of the recommendation will not depend on the optimal recommendation since we are maximizing the objective function value. Furthermore, the expected utility of the recommendation will not depend on the optimal recommendation's objective function value either since we are computing the expectation of the objective function value given the candidate and returned dataset.

Garnett explains, while referring back to the visualization of the sequential decision tree in Figure 5.4, that the utility function will in effect, "collapse" the expected utility of a final decision into a utility of the returned dataset. Thus, we can 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 below.

2.2 Selecting an Action Space

First, we need to select an action space, $\mathcal{A} \subset \mathcal{X}$, for our recommendation. There are two rather extreme choices, one being maximally restrictive and the other maximally permissive.

- At one extreme, the **maximally restrictive** option will restrict our recommendation choices to only visited points $\mathbf{x}$. While this option ensures we have at least some knowledge of the objective function at our recommended point, it does not allow for any exploration: we may not have visited the best point and so the best point may not be contained in $\mathbf{x}$.
- At the other extreme, we could be **maximally permissive** and select the action space to be the entire domain $\mathcal{X}$. However, choosing the entire domain to be our action space will require us to have faith in the objective function model's beliefs, particularly when recommending an unvisited point.

In literature, there have been some suggestions of compromise between the two extremes with Garnett citing an example from Osborne et al. wherein the final recommendation choice is restricted to points where the objective value is known within some acceptable tolerance. Osborne et al. accomplished this compromise by defining a parametric and data-dependent action space with form

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

where ε denotes a threshold representing the highest amount of acceptable uncertainty in the objective function value. Since this approach should, for the most part, avoid issues of recommending points at locations where the objective function is not sufficiently known, we will pause our discussion of selecting an action space and move onto our discussion of selecting a utility function.

2.3 Selecting a Utility Function

We need to select a utility function $v(\phi)$ that will evaluate a recommendation x after we have observed its corresponding objective function value ϕ. Since our focus has been on maximization, the utility should be *monotonically* increasing in ϕ.

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

Although we know that the utility should always be increasing in ϕ, there is still the question of what shape the function ought to assume. In truth, the shape that the function will take will inherently 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. The alternative could have a lower expected value with lower uncertainty, making it a safer albeit lower recommendation.

- A **risk-tolerant** utility function will allow for more risk to potentially attain greater reward (i.e., higher expected value but higher uncertainty).

- A **risk-averse** utility function will avoid imposing risk and will err towards lower risk even if it means a lower reward (i.e., lower uncertainty but lower expected value).

The most simple and common utility in Bayesian optimization is a linear utility:

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

If we are using a linear utility, then the expected utility from recommending x will be the posterior mean of ϕ:

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

Notice that when using a linear utility, the objective function's uncertainty is not even considered when making a decision. In fact, we are considered to be *risk neutral* since we don't differentiate between points with equal expected values based on their respective uncertainty. Garnett adds that while being risk neutral is computationally convenient due to its simplicity, such a utility may not be consistent with our true preferences.

In order to reason about risk preferences, we can consider the **certainty equivalent**. The certainty equivalent is an objective value corresponding to a (hypothetical) *risk-free* alternative recommendation to which our preferences would be indifferent. For example, suppose we have a risky potential recommendation x where we do not know the value of the objective function exactly at x. The certainty equivalent for x is an objective function value ϕ' such that

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

If we are using a risk-neutral utility then the certainty equivalent of some point x is its expected value: $\phi' = \mu_{\mathcal{D}}(x)$. This means that we would only abandon one recommendation in favor of another if the latter had a higher expected value (regardless of risk). Alternatively, we may want to express our risk-aware preferences using *nonlinear* utility functions.

3 A Note on Nonlinear Utility Functions

Aforementioned, we may wish (or truly need) to express our risk-aware preferences using *nonlinear* utility functions. Specifically, consider a scenario where our preferences lean more towards *risk aversion*. In such a scenario, we may accept a recommendation point with a lower expected value if it also results in less risk. To express these preferences in a utility function, we can use a concave function of the objective value. Garnett uses Jensen's inequality as an example of a utility function expressing risk averse preferences:

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

Under this utility function, the certainty equivalent of some risky recommendation will be *less* than its expected value. In a similar manner, we may wish to express *risk-seeking* preferences using a **convex** utility function. In such a case, the certainty equivalent of some risky recommendation will be *greater* than its expected value. Thus, our preferences will implicitly push toward gambling.

Various risk-averse utility functions have been proposed in literature pertaining to economics and decision theory, for obvious reasons. However, since risk-averse and risk-seeking utilities are not used much in Bayesian optimization, we follow Garnett in not including a full discussion as it would be out-of-scope. Note that these types of utilities can be useful in certain settings, especially where risk neutrality is questionable.

Before we move on, 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:

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

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 from Bayesian optimization literature and practice. This section is by no means exhaustive with respect to including each and every utility function out there but rather an exposition of the commonly used utility functions 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:

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

where $H[\omega|\mathcal{D}]$ 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, $H[\omega|\mathcal{D}]$. However, for our context, this is fine and as noted by Garnett, if necessary we will denote the conditional entropy with an explicit expectation: $\mathbb{E}[H[\omega|\mathcal{D}]|\mathbf{x}]$.

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.

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

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)