## Bayesian Experimental Design

Many problems that arise in scientific discovery and design share the following properties:

- the *design space* or the space of viable candidates is massive,
- the percentage of relevant or desirable observations is low, and
- the act of discovery or exploring the space is inherently slow.

As a motivating example, consider the problem of crop breeding: agricultural scientists are interested in predicting (and often, maximizing) the yield of some crop but there are infinitely many possible genetic varieties or strains, most are not viable or will not outperform current commercial strains and crucially, gathering the relevant yield data requires a full growth cycle i.e., it takes months or even years before it can be determined whether or not some strain was a success. Similar problems arise in drug and material discovery, robotics and reinforcement learning settings, and many engineering disciplines.

The Bayesian approach to solving such **experimental design** problems has been found to be very powerful in terms of accelerating and improving the discovery process. One particularly notable success story is "AutoML" or automated hyperparameter optimization, which also fits neatly into this paradigm: the search space of possible hyperparameter settings is massive, especially as the number of hyperparameters in the models grows. Yet most hyperparameter settings perform quite poorly and the only way to evaluate some setting of the hyperparameters requires training (or partially training) the model, which could take days or weeks on expensive, specialized hardware. Famously, the hyperparameters of AlphaGo, the Go playing agent that beat top human experts, were tuned in part using a technique known as Bayesian optimization.[1]

At a high-level, the Bayesian approach to experimental design combines a variety of tools we have already developed in this course:

- We assume the existence of some unknown function that maps elements of the design space to some performance metric or property of interest. This function is typically, but not necessarily, modeled with a Gaussian process.
- We query some oracle (e.g., by running an experiment or gathering data) to refine/improve our model. The location of this query is determined by specifying an objective (or equivalently, a loss) and applying Bayesian decision theory.
- We update our model using Bayesian inference and repeat until some notion of convergence has been achieved or an allocated budget of queries has been expended.

Figure 1 depicts an outline of an experimental design pipeline.

The exact nature of the loss function used to define the query selection will depend on our inference goal. We will explore a variety of inference tasks in the following weeks that broadly fall under the umbrella of **probabilistic numerics**. Probabilistic numerics treats intractable or unknowable quantities as random variables to be estimated, commonly using Bayesian inference techniques. The next few lectures will focus on one such numerical analysis task: optimization.

[1] https://arxiv.org/pdf/1812.06855

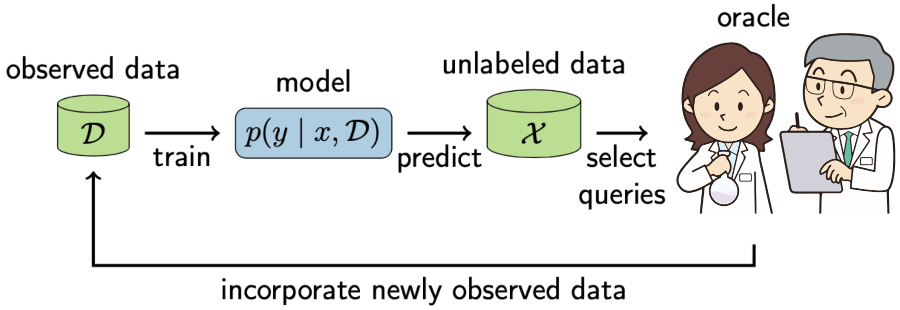

Figure 1: A high-level overview of the Bayesian approach to experimental design. Note the iterative, sequential nature of the problem: new observations are fed back into the model, which is subsequently used to gather more observations.

## Bayesian Optimization

Suppose we have a function $f\colon \mathcal{X} \to \mathbb{R}$ that we wish to maximize on some domain $X \subseteq \mathcal{X}$. That is, we wish to find

$$x^* = \operatorname*{arg\,max}_{x \in X} f(x).$$

In numerical analysis, this problem is typically called (global) **optimization** and has been the subject of decades of study. We draw a distinction between global optimization, where we seek the absolute optimum in $X$, and local optimization, where we seek to find a local optimum in the neighborhood of a given initial point $x_0$.

If an exact functional form for $f$ is not available (that is, $f$ behaves as a "black box"), what can we do? **Bayesian optimization** proceeds by maintaining a probabilistic belief about $f$ and designing a so-called **acquisition function** to determine where to evaluate the function next. Bayesian optimization is particularly well-suited to global optimization problems where $f$ is an *expensive* black-box function; for example, evaluating $f$ might require running an expensive simulation or training a large, machine learning model.

Although not strictly required, Bayesian optimization almost always reasons about $f$ by choosing an appropriate Gaussian process prior:

$$p(f) = \mathcal{GP}(f; \mu, K).$$

Given observations $\mathcal{D} = (\mathbf{X}, \mathbf{f})$,[2] we can condition our distribution on $\mathcal{D}$ as usual:

$$p(f \mid \mathcal{D}) = \mathcal{GP}(f; \mu_{f|\mathcal{D}}, K_{f|\mathcal{D}}).$$

We have already developed the machinery to perform this modeling and updating. The key question becomes, given this set of observations, how do we select where to observe the function next?

The meta-approach in Bayesian optimization is to design an **acquisition function**, $a(x)$. The acquisition function is typically an inexpensive function that can be evaluated at a given point that is commensurate with how desirable evaluating $f$ at $x$ is expected to be for the maximization

[2] We will assume these observations to be noiseless here, but we could extend the methods here to the noisy case without difficulty.

problem. We then optimize the acquisition function to select the location of the next observation. Of course, we have merely replaced our original optimization problem with another optimization problem, but on a much-cheaper, easier to optimize function $a(x)$.

### Exploration vs. Exploitation

A key consideration in designing an acquisition function for Bayesian optimization is navigating the tradeoff between exploration and exploitation. Exploration is when we use our queries to learn more about the function at places where we are uncertain about its behavior in order to set up potential future gains. Exploitation is when we use our queries to for short-term gain by improving our belief around locations we currently believe to be high yield. This tradeoff is illustrated in Figure 2 for the Bayesian optimization context, where "high yield" is defined in terms having a higher target function value.

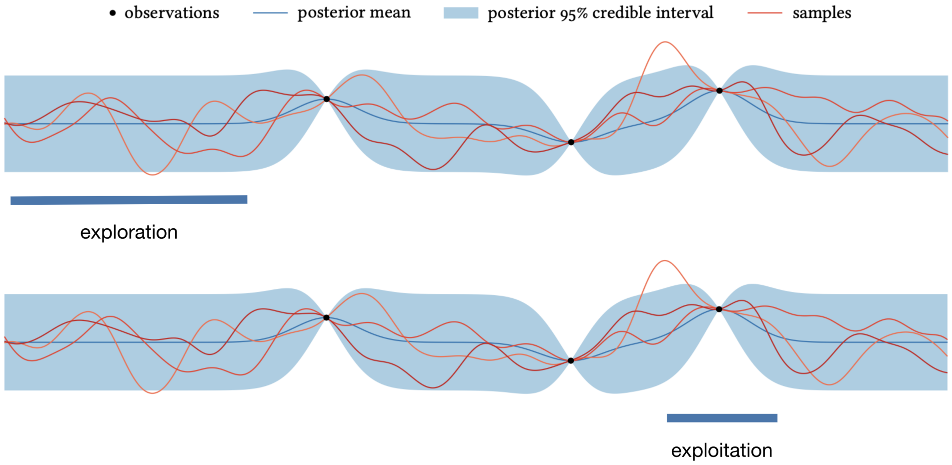

Figure 2: A depiction of the exploration-exploitation tradeoff in the context of Bayesian optimization: exploration (top) prioritizes exploring regions far from the observations, where the behavior of the function is most unknown, while exploitation (bottom) would query locations near where the model *currently* believes the highest function value to be.

We will see that many acquisition functions for Bayesian optimization naturally and elegantly navigate this trade-off, seamlessly and automatically transitioning from an exploration phase to an exploitation phase and back as needed.

## Acquisition Functions

Many acquisition functions can be interpreted in the framework of Bayesian decision theory as corresponding to an expected loss associated with evaluating $f$ at a point $x$. We then select the point with the lowest expected loss, as usual.

In the following sections, we will drop the $f \mid \mathcal{D}$ subscripts on the mean $\mu$ and covariance $K$ functions for $f$; assume everything is based on a posterior distribution when data is available.

### Probability of improvement

Perhaps the first acquisition function designed for Bayesian optimization was **probability of improvement.** Suppose

$$f' = \max \mathbf{f}$$

is the maximal value of $f$ observed so far. Probability of improvement evaluates $f$ at the point most likely to improve upon this value. This corresponds to the following utility function[3] associated with evaluating $f$ at a given point $x$:

$$u(x) = \begin{cases} 0 & f(x) < f' + \varepsilon \\ 1 & f(x) \geq f' + \varepsilon. \end{cases}$$

That is, we receive a unit reward if $f(x)$ turns out to be "better than" (i.e., greater than) $f'$ by at least $\varepsilon$, and no reward otherwise; here, $\varepsilon$ is a tunable parameter of this acquisition function. The probability of improvement acquisition function is then the expected utility as a function of $x$:

$$\begin{aligned} a_{\text{PI}}(x) = \mathbb{E}\big[u(x) \mid x, \mathcal{D}\big] &= \int_{f'+\varepsilon}^{\infty} \mathcal{N}\big(f; \mu(x), K(x,x)\big)\,\mathrm{d}f \\ &= 1 - \Phi\big(f' + \varepsilon; \mu(x), K(x,x)\big). \end{aligned}$$

The point with the highest probability of improvement (the maximal expected utility) is selected. This is the Bayes optimal action under this loss. Figure 3 shows the effect of the parameter $\varepsilon$ on this acquisition function: observe that as $\varepsilon$ grows, the acquisition policy becomes more exploratory.

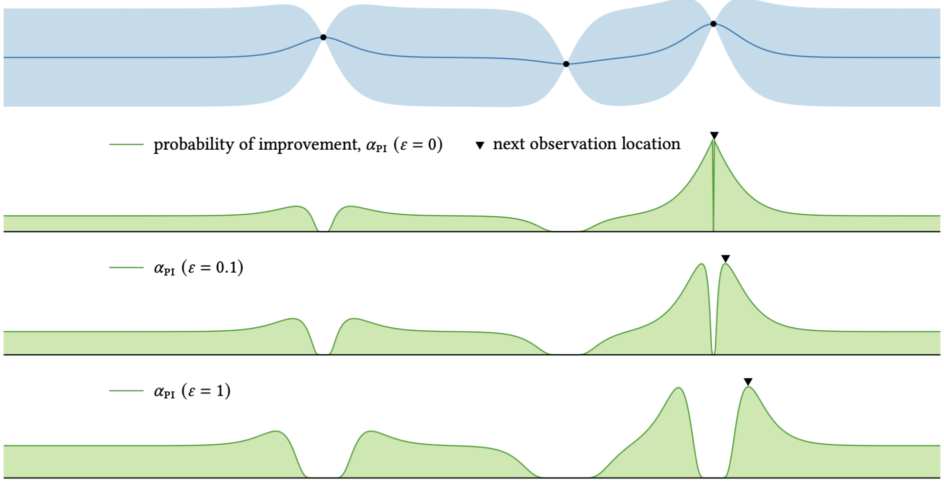

Figure 3: The probability of improvement acquisition function on our running example. Note that the acquisition function is symmetric about the observations and ties in the acquisition function are broken arbitrarily.

[3] Recall a utility function is simply a negative loss function.

### Expected improvement

The loss function associated with probability of improvement is somewhat odd: we get a reward for improving by at least a certain amount upon the current maximum, but that reward is independent of the size of the improvement! This acquisition function can sometimes lead to odd behavior, and in practice can get stuck in local optima and under-explore globally.

An alternative acquisition function that does account for the size of the improvement is **expected improvement.** Again suppose that $f'$ is the maximal value of $f$ observed so far. Expected improvement evaluates $f$ at the point that, in expectation, improves upon $f'$ the most. This corresponds to the following utility function:

$$u(x) = \max\bigl(0, f(x) - f'\bigr).$$

That is, we receive a reward equal to the "improvement" of the observation over our current best observation, $f(x) - f'$ The expected improvement acquisition function is then the expected utility as a function of $x$:

$$\begin{aligned} a_{\text{EI}}(x) = \mathbb{E}\bigl[u(x) \mid x, \mathcal{D}\bigr] &= \int_{f'}^{\infty} (f - f')\,\mathcal{N}\bigl(f; \mu(x), K(x,x)\bigr)\,\mathrm{d}f \\ &= \bigl(\mu(x) - f'\bigr)\Phi\left(\frac{\mu(x) - f'}{\sqrt{K(x,x)}}\right) + \sqrt{K(x,x)}\,\mathcal{N}\left(\frac{\mu(x) - f'}{\sqrt{K(x,x)}}\right). \end{aligned}$$

The point with the highest expected improvement (the maximal expected utility) is selected.

The expected improvement has two components. The first can be increased by increasing the mean function $\mu(x)$. The second can be increased by increasing the variance $K(x,x)$. These two terms can be interpreted as explicitly encoding the tradeoff between *exploitation* (evaluating at points with high mean) and *exploration* (evaluating at points with high uncertainty). The expected improvement acquisition function *automatically* captures both as a result of the Bayesian decision theoretic treatment.

Figure 4 depicts the expected improvement acquisition function on our running example as well as the observations made by the algorithm over 20 iterations; the optimal function value was located after 19 queries.

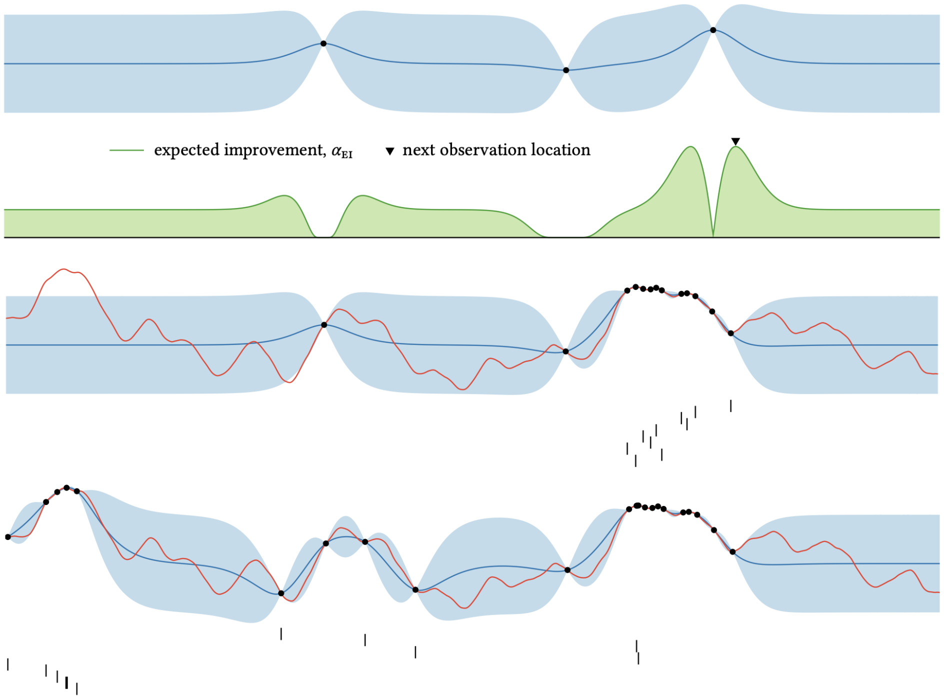

Figure 4: The expected improvement acquisition function evaluated on our running example (top) as well as the first ten observation locations selected by this policy (middle) and the next ten observation locations (bottom). The height of the tick marks indicates the relative order that points are selected to be observed in, with higher tick marks being selected earlier; the thicker tick marks indicate observations with 0.2 units of the true optimum.