Active Search

Active search is a variant of Bayesian optimization that is particularly relevant in many scientific discovery problems. Specifically, active search assumes a *discrete* domain, $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$, within which there is hidden some rare, valuable subset of points:

$$\mathcal{V} \subset \mathcal{X} \text{ and typically (but not necessarily) } |\mathcal{V}| \ll |\mathcal{X}|.$$

Each point in the domain is either valuable or not, which can be conceptualized as a binary label, $y_i = \mathbb{1}(\mathbf{x}_i \in \mathcal{V})$. The goal of active search is to find as many valuable points as possible using some pre-specified budget of observations.

Here's a somewhat colorful analogy: suppose the design space is a vast ocean with some scattered islands throughout. Bayesian optimization is trying to find the highest peak among the islands while active search tries to map out the location of all the islands. A common motivating application for active search is drug or material discovery: suppose you have a list of candidate substances or chemicals and a desired property you wish to satisfy (e.g., the substance binds with a specified biological target or exhibits a certain rigidity while being able to float in water). Active search provides an algorithmic, principled mechanism for quickly and efficiently identifying candidates that satisfy the specified property.

A natural and computationally convenient dataset utility for this setting is the simple cumulative reward:

$$u'(\mathcal{D}) = \sum_{i=1}^{n} y_i \text{ where } \mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n.$$

This gives rise to the following, intuitive acquisition function:

$$\begin{aligned} a_{\text{AS}}(\mathbf{x}) = \mathbb{E}\big[u(\mathbf{x}) \mid \mathbf{x}, \mathcal{D}\big] &= \mathbb{E}\left[u'\Big(\mathcal{D} \cup \big(\mathbf{x}, y)\big)\Big) - u'(\mathcal{D})\right] \\ &= \mathbb{E}\big[\mathbb{1}(\mathbf{x} \in \mathcal{V}) \mid \mathbf{x}, \mathcal{D}\big] = \Pr\left(\mathbf{x} \in \mathcal{V} \mid \mathbf{x}, \mathcal{D}\right). \end{aligned}$$

In words, given a probabilistic model over the domain (e.g., a Gaussian process classifier), this acquisition function simply chooses to observe the location with the highest posterior probability of being valuable.

One of the key differences between active search and Bayesian optimization is the relative success of the "greedy" or "myopic" policies: in Bayesian optimization, many of the commonly-used acquisition functions correspond to one-step optimal decisions or decisions that maximize the expected marginal utility gain; despite ignoring the remaining budget and only considering the impact of the next observation, these policies have enjoyed great empirical success in Bayesian optimization and remain popular because of their low computational costs. Conversely, the myopic policy for active search does quite poorly, both in theory and in practice. As a simple example, consider the setting depicted in Figure 1.

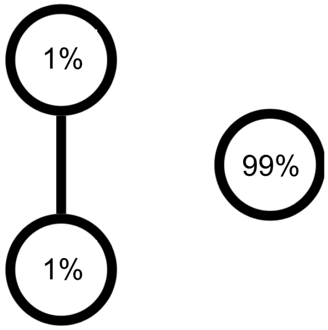

Figure 1: A simple active search problem consisting of three points. The two points on the left have a low posterior probability of being valuable (as indicated by the value inside the circles) but are perfectly correlated (as indicated by the solid edge connecting them): if one is valuable, then the other is also guaranteed to be valuable and vice versa. The point on the right is independent of the other two and has a high posterior probability of being valuable.

Given a budget of two observations, the myopic one-step optimal policy would first choose to observe the point on the right and then arbitrarily pick one of the points on the left to observe; the expected number of valuable points discovered by this policy is just $0.99 + 0.01 = 1$. However, this is suboptimal! The optimal policy, *taking into account that we have a budget of two*, is to fist observe one of the two correlated points on the left: if you discover that that point is valuable, you know for certain the other point on the left is also valuable. If not, no worries, you can still use your final observation to observe the point on the right anyway. The expected number of valuable points discovered by the optimal policy is $2(0.01) + 0.99(0.99) = 1.0001 > 1$.

Even though this is a seemingly small improvement, Garnett et al. (2012) used this kind of adversarial construction to prove that there exists an active search setting in which the optimal n-step look-ahead policy underperforms the optimal $n + 1$-step look-ahead policy by any arbitrary amount in expectation i.e., the one-step optimal policy can be made arbitrarily worse than the two-step optimal policy, which in turn can be made arbitrarily worse than the three-step optimal policy and so on. Thus, considering anything less than the full budget can, in certain circumstances, perform unboundedly poorly relative to the optimal, full look-ahead policy. These theoretical findings have been verified in a variety of empirical settings.

It's interesting to pause and consider why this discrepancy between the optimization and search problems exists: in Bayesian optimization (under certain assumptions), myopic policies are guaranteed to asymptotically approach the optimal policy whereas in active search, no myopic policy can achieve any guarantees relative to the optimal non-myopic policy. Intuitively, this is because of the differences in the shapes of the utility curves over time. Figure 2 shows idealized representations of the utilities for Bayesian optimization and active search as a function of the number of observations.

Most utilities in Bayesian optimization naturally experience diminishing marginal returns: the closer we get to discovering the global optimum, the smaller any possible improvements become. In the extreme, if our next observation exactly discovers the global optimum, then we don't need to care about any future observations and looking ahead carries no benefit. On the other hand, the utility in active search grows linearly: every additional valuable point we discover contributes equally to the total utility. As such, because every observation all the way up to the last one has the potential to contribute "significantly" (i.e., without diminishing relative to previous observations), careful planning of the entire budget becomes imperative.

So if the non-myopic, full look-ahead policy is the only policy that can be guaranteed to be optimal, why not just use that? Unfortunately, this policy is typically intractable to compute: given a domain of N points and a budget of B observations, looking ahead all the way to the end of the budget requires $O(N^B)$ computation, an utterly unmanageable runtime for most interesting active search

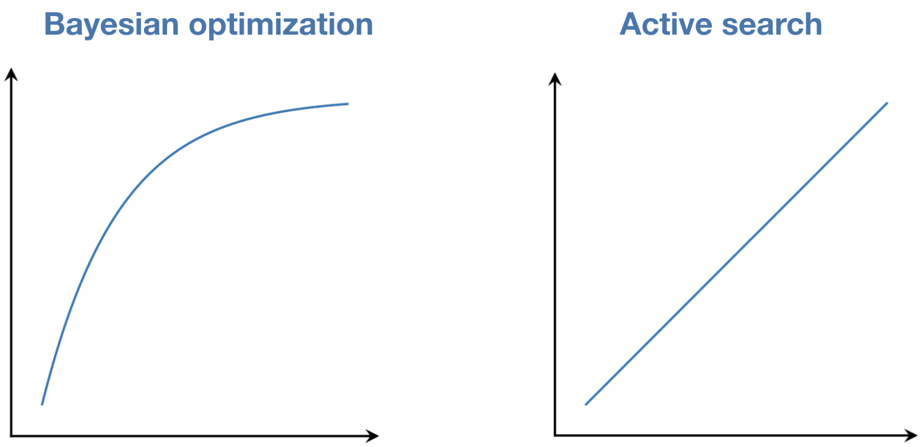

Figure 2: Prototypical curves of utility as a function of number of observations made for Bayesian optimization (left) and active search (right).

problems.

As such, we need to develop approximations to this optimal but intractable, non-myopic policy. Jiang et al. (2017) proposed one such approximation that is motivated by the idea of "batch rollout", a technique previously employed in Bayesian optimization that enjoyed some success (e.g., González et al. (2015)). Formally, given a budget of B remaining observations, they proposed the following acquisition function for active search:

$$a_{\text{AS}}(\mathbf{x}) = \Pr\left(\mathbf{x} \in \mathcal{V} \mid \mathbf{x}, \mathcal{D}\right) + \mathbb{E}_y\left[\sum_{\text{top } B-1} \Pr\left(\mathbf{x}' \in \mathcal{V} \mid \mathbf{x}', \mathcal{D} \cup (\mathbf{x}, y)\right) \mid \mathbf{x}, y, \mathcal{D}\right]$$

where the top $B-1$ subscript on the sum indicates that this is a sum over the $B-1$ unobserved locations with the highest posterior probabilities of being valuable. In effect, this acquisition function imagines that after the current observation, we need to spend the remaining $B-1$ observations all at once (in a batch); if that were the case, then by the linearity of expectations, the optimal batch of $B-1$ locations to observe is simply the $B-1$ locations with the highest posterior probability of being valuable, conditioned on $\mathcal{D}$ and the new observation.

Inspecting this acquisition function further, we can see that it structurally trades off between exploration (the second term) and exploitation (the first term), as many of our Bayesian optimization acquisition functions did. Indeed, this behavior is also observed empirically when using this acquisition function. Figure 3 shows the typical performance of this approximate, non-myopic acquisition function relative to a fixed-horizon look-ahead policy as a function of the number of observations made: it initially falls behind the greedier alternative as it explores the space but quickly catches back up as it transitions from exploring to exploiting its knowledge before ultimately surpassing the myopic look-ahead policy.

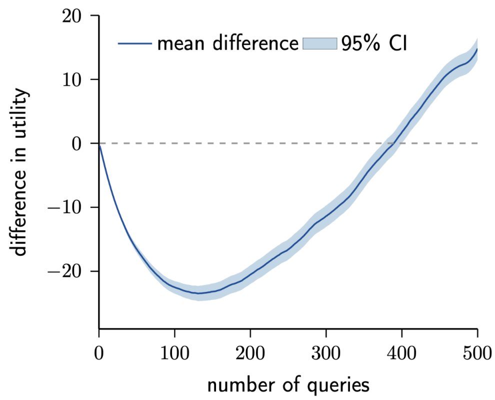

Figure 3: A common empirical finding when comparing the non-myopic acquisition function described in the text against a finite-horizon look-ahead policy (e.g., two- or three-step look-ahead). The credible interval shown in the figure is over multiple random initial datasets used to seed each policy. The dashed line shows the baseline performance of the myopic look-ahead policy at each iteration and the y-axis is the relative performance of the non-myopic acquisition function.

Finally, Figure 4 shows an example of this acquisition function being executed on a toy two-dimensional active search task. The distribution of observations over the domain clearly indicates that this approximate non-myopic acquisition function thoroughly explored the space and was able to leverage its exploration to find large clusters of valuable points.

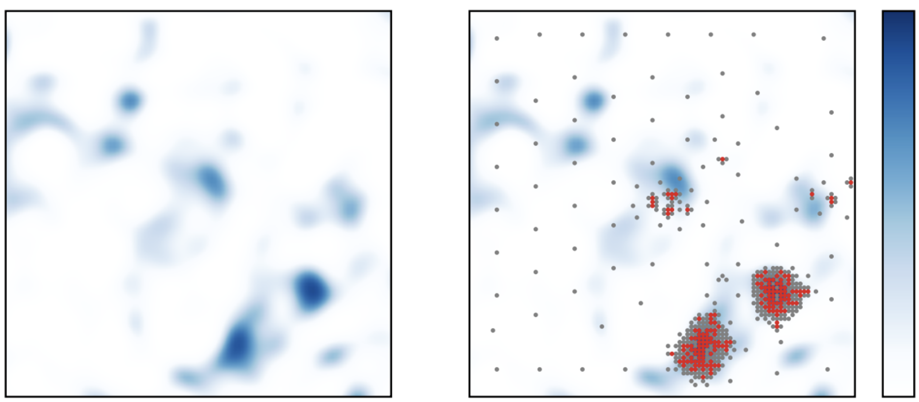

Figure 4: Active search using the non-myopic acquisition function described in the text. The true distribution of valuable points in this two-dimensional domain is shown on the left, with darker blues corresponding to a higher probability of being valuable. 500 observations are shown on the right: red points indicate valuable points discovered by the algorithm.