Expensive Function Optimization with Stochastic Binary Outcomes

Matthew Tesch MTESCH@CS.CMU.EDU
Jeff Schneider JEFF.SCHNEIDER@CS.CMU.EDU
Howie Choset CHOSET@CS.CMU.EDU
Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213 USA

Abstract

Real world systems often have parameterized controllers which can be tuned to improve performance. Bayesian optimization methods provide for efficient optimization of these controllers, so as to reduce the number of required experiments on the expensive physical system. In this paper we address Bayesian optimization in the setting where performance is only observed through a stochastic binary outcome – success or failure of the experiment. Unlike bandit problems, the goal is to maximize the system performance after this offline training phase rather than minimize regret during training. In this work we define the stochastic binary optimization problem and propose an approach using an adaptation of Gaussian Processes for classification that presents a Bayesian optimization framework for this problem. We propose an experiment selection metric for this setting based on expected improvement. We demonstrate the algorithm's performance on synthetic problems and on a real snake robot learning to move over an obstacle.

1. Introduction

Many real-world optimization tasks take the form of an optimization problem where the number of objective function samples can be severely limited. This often occurs with physical systems which are expensive to test, such as choosing optimal parameters for a robot's control policy, or with design optimizations which take considerable effort to evaluate, such as using computational fluid dynamics simulations to test aircraft wing designs. In cases where the objective is a continuous real-valued function, the use of Bayesian sequential experiment selection metrics such as *expected improvement* has lead to efficient optimization of these objectives. A particular advantage of expected improvement is that it requires no tuning parameters.

We are interested in the problem setting where the objective is not a deterministic continuous-valued function, but a stochastic binary valued function. In the case of a robot, instead of choosing parameters which maximize locomotive speed, the task may be to choose the parameters of a policy which maximize the probability of successfully moving over an obstacle, where the success of this task is stochastic due to environmental factors or small uncontrollable variations in the commanded movements of the robot.

Inspired by the success of Bayesian optimization for continuous problems, we propose using a similar framework for the stochastic binary setting. This paper begins with a definition of the stochastic binary optimization problem and a brief overview of background material. We describe several existing algorithms which could be applied to this problem and propose a selection metric for stochastic binary functions based on expected improvement. Finally, we present a summary of results from a comparison of these methods on a set of synthetic test functions, and apply the proposed method to learn robust motions for a snake robot to overcome obstacles.

The primary contributions of this paper are the definition of the stochastic binary optimization problem, the application of Gaussian process classification (GPC) to adapt Bayesian optimization methods to this problem setting, and the definition of the expected improvement for stochastic binary outputs. Secondary contributions include the comparison of methods on synthetic test functions and the optimization of a new locomotive capability on a real snake robot.

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

2. Related Work

For optimization problems where each function evaluation is *expensive* (either requiring significant time or resources) the choice of which point to sample becomes more important than the speed at which a sample can be chosen. To this end, Bayesian optimization of such functions relies on a function regression method, such as Gaussian processes (GPs) (Rasmussen & Williams, 2006), to predict the entire unknown objective from limited sampled data. Given the information provided by this prediction of the true objective, the central challenge is the exploration/exploitation tradeoff – balancing the need to explore unknown areas of the space with the need to refine the knowledge in areas that are known to have high function values. Metrics such as the upper confidence bound (Auer et al., 2002), probability of improvement (Žilinskas, 1992), and expected improvement (Mockus et al., 1978) attempt to trade off these conflicting goals. A comprehensive survey on past work in this subject is given by Jones (2001). The existing literature primarily focuses on deterministic, continuous, real-valued functions, rather than stochastic ones or ones with binary outputs.

Active learning (c.f. the survey of Settles (2009)), however, is primarily focused on learning the binary class membership of a set of unlabeled data points. In general, this work focuses on accurately learning the class membership of all of the unlabeled points with high confidence, which is inefficient if the loss function is asymmetric, i.e. if it is more important to identify successes than failures. Of particular interest is the active binary-classification problem discussed in (Garnett et al., 2011); this problem focuses on finding a Bayesian optimal policy for identifying a particular class, but assumes deterministic class membership.

A particularly relevant set of subtopics in the bandit literature is continuous-armed bandits (Agrawal, 1995; Auer et al., 2007; Kleinberg & Upfal, 2008) or metric bandits (Bubeck et al., 2011b); these both have a similar problem structure to that described in our work. Metric-armed bandits embed the "arms" of the classic multi-arm bandit problem into a metric space, allowing a potentially uncountably infinite number of arms. These arms are often constrained to generate responses via an underlying (often Lipschitz continuous) probability function. The focus of bandit work is minimizing asymptotic bounds on the *cumulative regret* in an online setting, whereas we are not concerned with errors incurred during training, but rather the performance of the algorithm recommendation after an offline training phase. The recent work of (Bubeck et al., 2011a) begins to address the problem we describe here by investigating bounds on the *simple regret* (predictive quality of the model after training) as compared to bounds on cumulative regret, but the results in this paper aim to characterize the spaces in which cumulative regret can be minimized rather than the definition of practical algorithms for the simple regret case.

3. Problem Definition

The problem we attempt to solve is analagous to minimizing simple regret for a continuous-armed bandit that recieves a 1/0 binomial reward, with a budget of n function evaluations.

More formally, we state it as follows: given an input (parameter) space $X \subset \mathbb{R}$ and an unknown function $\pi\colon X \to [0,1]$ which represents the underlying binomial probability of success of an experiment, the learner sequentially chooses a series of points $\mathbf{x} = \{x_1, x_2 \dots x_n \,|\, x_i \in X\}$ at which to run these experiments. After choosing each x_i, the learner recieves feedback y_i where $y_i = 1$ with probability $\pi(x_i)$ and $y_i = 0$ with probability $1 - \pi(x_i)$. Note that the choice of x_i is made with knowledge of $\{y_1, y_2 \dots y_i - 1\}$. The goal of the learner is to recommend, after n experiments, a point x_r which minimizes the (typically unknown) error, or *simple regret*, $\max_{x \in X} \pi(x) - \pi(x_r)$; this is equivalent to maximizing $\pi(x_r)$.

4. Background

4.1. Bayesian Optimization

In Bayesian optimization of a continuous real-valued deterministic function, the goal is to find x_{best} which maximizes[1] the function $f\colon X \to \mathbb{R}$. The process relies on a probabilistic model $\hat{f}$ of the underlying function f which is generated from the data; often GPs are used for this model.

The optimization also relies on a selection metric (sometimes referred to as an infill criterion) which, at each iteration, selects the next point to sample. The algorithm is essentially an iterative process – at each step i, fit a model based on $\mathbf{x}$ and $\mathbf{y}$, select a next x_i, and evaluate x_i on the true function f to obtain y_i; see Alg. 1. The critical parameter then is the metric which is maximized to choose the next point.

The idea of expected improvement (Mockus et al., 1978) has been used as such a selection metric, and has been popularized by Jones in his Efficient Global

[1] When referencing other work in this field, note that often the goal is to minimize rather than maximize f.

Algorithm 1 Bayesian Optimization

```
x := space-filling design of k points
y := {}
for i = 1 to k do
  addToList(y,f(x{i}))
end for
for i = k + 1 to n do
  f̂ := conditionGP(x, y)
  x_i := argmax_X metric(f̂(x))
  addToList(x,x_i)
  addToList(y,f(x_i))
end for
```

Optimization algorithm (1998)[2]. Given a function estimate $\hat{f}$, improvement is defined as

$$I(\hat{f}(x)) = \max(\hat{f}(x) - y_{best}, 0), \tag{1}$$

where y_{best} was the maximizer of the previously sampled $\mathbf{y}$. Because the GP defines $\hat{f}$ as a posterior distribution over potential f, the expectation over these function estimates defines the expected improvement,

$$\begin{aligned} \mathrm{EI}(x) &= \mathbb{E}[I(\hat{f}(x))], \\ &= \int_{-\infty}^{\infty} p_f^x(y) \max(y - y_{best}, 0) dy, \\ &= (\hat{f}_\mu^x - y_{best}) \left(1 - \Phi((y_{best} - \hat{f}_\mu^x)/\hat{f}_\sigma^x)\right) \\ &\quad + \hat{f}_\sigma^x \phi((y_{best} - \hat{f}_\mu^x)/\hat{f}_\sigma^x), \end{aligned} \tag{2}$$

where p_f^x is the posterior probability density function at $\hat{f}(x)$, and $\hat{f}_\mu^x$ and $\hat{f}_\sigma^x$ are the mean and standard deviation of this pdf.

4.2. Gaussian Processes for Classification

One of the key ideas behind Bayesian optimization is the probabilistic modeling of the unknown function. Below we briefly describe the adaptation of GPs, which provide such a model in the continuous regression case, to a classification setting. This provides a similar probabilistic model for the underlying function in the stochastic binary case. More in-depth coverage of these ideas may be found in (Rasmussen & Williams, 2006).

Adapting GPs for a space of binary response variables uses concepts from linear binary classification. *Linear logistic regression* and *linear probit regression* use the logistic and the probit, respectively, as *response functions* σ to convert a linear model with a range of $(-\infty, \infty)$ to an output that lies within $[0, 1]$ (i.e., a valid probability). Therefore, given a linear regression model $y = w^T x$, the predicted class probability $\hat{\pi}(x)$ is $\sigma(wx)$. The choice of w for the latent linear regression model is typically accomplished via maximizing the likelihood of the data given the model.

Similarly, a GP can generate outputs in the range $(-\infty, \infty)$, and by using a response function σ can convert these outputs to values which can be interpreted as class probabilities. In particular, the latent GP $\hat{f}$ defines a Gaussian probability density function p_f^x for each $x \in X$ (as well as joint Gaussian pdfs for any set of points in X). We define the corresponding probability density over class probability functions as p_π^x.

Note that although the response function σ maps from the latent space F to the class probability space Π, $p_\pi^x(\bar{y}) \neq p_f^x(\sigma^{-1}(\bar{y}))$ (where $\bar{y}$ is a class probability in Π, not a 0/1 sample). Instead, due to the change of variables,

$$p_\pi^x(\bar{y}) = p_f^x(\sigma^{-1}(\bar{y})) \frac{\delta \sigma^{-1}}{\delta \bar{y}}(\bar{y}). \tag{3}$$

In this work, we will assume that σ is the standard normal cumulative density function; however, any monotonically increasing function mapping from $\mathbb{R}$ to the unit interval can be used.

Finally, because we do not observe values of f directly, the inference step for conditioning our GP posterior on the sampled observations $\mathbf{x} = \{x_i\}$ and $\mathbf{y} = \{y_i\}$ requires computing the following integral to determine the posterior $\hat{f}$ at x^*:

$$p(\hat{f}^*|\mathbf{x}, \mathbf{y}, x^*) = \int p(\hat{f}^*|\mathbf{x}, x^*, \mathbf{f}^*) p(\mathbf{f}^*|\mathbf{x}, \mathbf{y}) d\mathbf{f}^* \tag{4}$$

In this equation, $\mathbf{f}^*$ represents the GP prior on the latent function at x^*. Unfortunately, the second term in the integrand represents a non-Gaussian likelihood which makes this integral analytically intractable; approximate inference methods for GP classification rely on approximating this with a Gaussian. Advantages and disadvantages of different approximations are discussed in (Nickisch & Rasmussen, 2008); we use Minka's expectation propagation (EP) method (2001) due to its accuracy and reasonable speed.

[2] One contribution of this algorithm is initialization of $\mathbf{x}$ via an optimal space-filling experimental design.

4.2.1. Expectation of Posterior on Success Probability

As noted above, $p^x_\pi(\bar{y}) \neq p^x_f(\sigma^{-1}(\bar{y}))$ due to the nonlinearity of σ. Because of this, the expectation of the posterior over the success probability, $\mathbb{E}[p^x_\pi]$, is not generally equal to $\sigma(\mathbb{E}[p^x_f])$. To calculate the former, we use the definition of expectation along with a change-of-variables substitution ($\pi = \sigma(f)$ and $\bar{y} = \sigma(z)$) to take this integral in the latent space (where approximations for the standard normal CDF can be used):

$$\begin{aligned}
\mathbb{E}[p^x_\pi] &= \int_0^1 \bar{y} p^x_\pi(\bar{y}) d\bar{y} && (5)\\
&= \int_0^1 \bar{y} p^x_f(\sigma^{-1}(\bar{y})) \frac{\delta \sigma^{-1}}{\delta \bar{y}}(\bar{y}) d\bar{y} \\
&= \int_{\sigma^{-1}(0)}^{\sigma^{-1}(1)} \sigma(z) p^x_f(z) \frac{\delta \sigma^{-1}}{\delta \bar{y}}(\sigma(z)) \frac{\delta \sigma}{\delta z}(z) dz \\
&= \int_{-\infty}^{\infty} \sigma(z) p^x_f(z) dz && (6)
\end{aligned}$$

As noted in section 3.9 of (Rasmussen & Williams, 2006), if σ is the Gaussian cumulative density function, this can be rewritten as

$$\mathbb{E}[p^x_\pi] = \Phi\left(\frac{\mathbb{E}[p^x_f]}{\sqrt{1 + \mathbb{V}[p^x_f]}}\right). \quad (7)$$

For notational simplicity, we define $\bar{\pi}(x) = \mathbb{E}[p^x_\pi]$ for use later in the paper.

5. Baseline Algorithms

Using GPC to model this problem allows us to infer a posterior probability distribution $\hat{\pi}$ over the unknown true function π from observing several (x_i, y_i), and also to obtain a posterior over a latent function $\hat{f}$. Although this latent function could technically be used for experiment selection, it does not have a direct probabalistic interpretation except through the response function σ.

As baselines to compare against the binary expected improvement metric we propose in §6, we use a uniform random experiment selection method along with the following approaches.

First, as upper confidence bounds (UCB) methods are often used in bandit and expensive optimization problems (e.g., (Auer et al., 2002)), we compare against UCB on the latent function $\hat{f}$, with β a tuneable metric parameter:

$$\text{UCB}^\beta_f(x) = \hat{f}_\mu(x) + \beta \hat{f}_\sigma(x) \quad (8)$$

For comparison, we also test the standard expected improvement metric in the latent space, EI_f, and on a GP directly fit to the binary data. For the former, because we are not directly observing the sampled function value we must redefine the y_{best} term in the improvement quantity from Eqn. (1) as

$$y_{best} = \max_{\mathbf{x}} \{\sigma^{-1}(\bar{\pi}(x))\}, \quad (9)$$

where $\mathbf{x}$ is all sampled x_i. This represents the latent space projection of the maximizer of $\bar{\pi}(x)$ at the previously sampled points.

Finally, we compare against the continuous-armed bandit algorithm UCBC (Upper Confidence Bound for Continuous-armed bandits) proposed in (Auer et al., 2007). This algorithm divides X into a set of n equal-sized intervals, and runs the multi-arm bandit UCB algorithm to select the interval from which to sample. The point to sample is then chosen uniformly at random from this interval. Recommendations for how to choose the algorithm parameter n are given in the paper.

6. Expected Improvement for Binary Responses

In the case of stochastic binomial feedback, the notion of improvement that underlies the definition of expected improvement must change. Because the only potential values for y_i are 1 and 0, after the first 1 is sampled y_{best} would be set to 1. Because there is no possibility for a returned value higher than 1, the improvement (and therefore the expected improvement) would then be identically zero for each $x \in X$.

Instead, we query the GP posterior at each point in $\mathbf{x}$, and let

$$\hat{\pi}_{max} = \max_{\mathbf{x}} \bar{\pi}(x). \quad (10)$$

As the 0 and 1 responses are samples from a Bernoulli distribution with mean $\pi(x)$, we define the improvement as if we could truly sample the underlying mean. Choosing this rather than conditioning our improvement on 0/1 is consistent with the fact that our $\hat{\pi}_{max}$ represents a probability, not a single sample of 0/1. In this case,

$$I_\pi(\pi(x)) = \max(\pi(x) - \hat{\pi}_{max}, 0) \tag{11}$$

To calculate the expected improvement, we follow a similar procedure to that in §4.2.1 to calculate the expecation of $I_\pi(\pi(x))$:

$$\begin{aligned} \mathrm{EI}_\pi(\hat{\pi}(x)) &= \int_{\hat{\pi}_{max}}^{1} (\bar{y} - \hat{\pi}_{max}) p_\pi^x(\bar{y}) d\bar{y} \\ &= \int_{\sigma^{-1}(\hat{\pi}_{max})}^{\infty} (\sigma(z) - \hat{\pi}_{max}) p_f^x(z) dz \end{aligned} \tag{12}$$

Unfortunately, the marginalization trick that allowed us to evaluate this integral and obtain a solution only requiring the Gaussian CDF in the case of $\bar{\pi}$ (Eqn. (7)) does not work because these integrals are not from $-\infty$ to ∞; fortunately these are one dimensional integrals regardless of the dimension of X and are easy to numerically evaluate in practice.

7. Empirical Results on Synthetic Functions

7.1. Synthetic Test Functions

To validate the performance of our expected improvement metric for stochastic binary outputs, we created several synthetic test functions on which we could run a large number of optimizations. Shown in Fig. 1 are three of these functions, these exhibit properties such as multiple local optima and a narrow global optimum to challenge optimization algorithms; moreover, they are stochastic ($\pi(x) \notin \{0, 1\}$) over much of X.

7.2. Experimental Setup

To compare the various algorithms, we allowed each algorithm to sequentially choose a series of $\mathbf{x} = \{x_1, x_2 \dots x_{50}\}$, with feedback of y_i generated from a Bernoulli distribution with mean $\pi(x)$ (according to the test function) after each choice of x_i. This was completed 100 times for each test function.

For our random selection baseline, at each step i, a random point was chosen and evaluated. For the baseline EI_f and UCB_f metrics (which used the latent GP) as well as the proposed EI_π metric, the standard Bayesian optimization framework described in Alg. 1 was used with an initial Latin hypercube sampling of 5 points. The UCB baseline tests were run with various values of the β parameter from 0.5 to 10; 1 was found to work as well or better than other values and was used for the comparison here. The maximization of the metric was done by evaluating the metric on a dense grid over the space; in practice and in higher dimensions one would typically apply another global optimization method to obtain the maximizer.

Often in the Bayesian optimization framework, the covariance function and hyperparameters of the GP are chosen at each iteration through likelihood maximization; we chose to use a simple squared exponential covariance with fixed hyperparameters (length scale of $e^{0.75}$ and signal variance of e^5) to reduce the variance in algorithm performance due to optimization of this likelihood function. The GP inference step (including the expectation propagation step) was done using the Gaussian Processes for Machine Learning MATLAB software package (Rasmussen & Williams).

For comparison with the continuous-armed bandit literature, we implemented the UCBC algorithm described in (Auer et al., 2007); the algorithm parameter of n was chosen as recommended therein for unknown functions, $n = (T/ln(T))^{1/4} = 2$, assuming the number of samples $T = 50$. We also ran UCBC with $n = 10$, but did not get appreciably different performance.

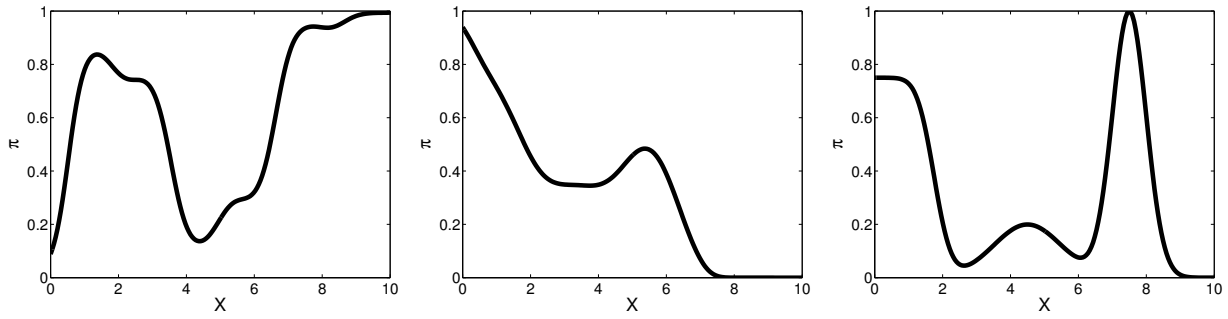
Our binary Bayesian optimization MATLAB code used to run these experiments, including implementations of the algorithms described herein, is available at `http://www.mtesch.net/ICML2013/`, along with more complete results with varied algorithm parameters on a wider variety of benchmark test functions.

7.3. Measuring Performance

To obtain a measure of the algorithm's performance at step i, we use the natural Bayesian recommendation strategy of choosing the point which has the highest expected probability of success $\mathbb{E}[p_\pi^x]$ given knowledge only of the sampled points $\{x_1, x_2 \dots x_i\}$ and $\{y_1, y_2 \dots y_i\}$. In practice, one may wish to optimize a utility function that also considers risk (e.g., the uncertainty in that probability).

After choosing $x_{best} = \mathrm{argmax}_X \, \mathbb{E}[p_\pi^x]$, this point is evaluated on the underlying true success probability function π, and the resulting value $\pi(x_{best})$ is given as the expected performance of the algorithm at step i. For the random selection and UCBC[3] algorithms which do not have a notion of $\hat{\pi}$, a GP was fit to the data collected by the algorithm to obtain this $\hat{\pi}$ using the same parameters as for the Bayesian optimization algorithms.

[3]UCBC does not define a recommendation strategy; the natural choice of a point uniformly at random from the interval with the highest mean performed very poorly and was therefore omitted from the results.



(a) Test Function 1 (b) Test Function 2 (c) Test Function 3

Figure 1. A number of synthetic test functions were created for algorithm comparison and validation. The equations for the three 1-D functions referenced in this paper are: **Test Function 1:** $\Phi\left(\sin(x) - \cos(3x)/4 + \frac{x^3-13x^2-29x-55}{50}\right)$ **Test Function 2:** $\sin(5x/4)/4 - \cos(3(x-1)/5)/20 - \frac{5x^3+54x^2-179x+159}{100}$ **Test Function 3:** $3\Phi((-40x+7)/16)/4 + \phi(x - 9/2)/2 + 5\phi(2x-15)/2$

7.4. Comparison of Results

In Fig. 2, we plot the average performance over 100 runs of the proposed stochastic binary expected improvement EI_π as well as the random baseline and the continuous-armed bandit UCBC algorithm. As expected, the knowledge of the underlying function grew slowly but steadily as random sampling characterized the entire function. The focus of EI_π on areas of the function with the highest expectation for improvement led to a more efficient strategy which still chose to explore, but focused experimental evaluations on more promising areas of the search space. Notably, EI_π matched or outperformed tuned versions of all other algorithms tested, *without* requiring a tuning parameter.

The UCBC algorithm worked well for simple cases (test function 1 had a significant region with high probability of success) but faltered as the functions became more difficult to optimize. One challenge with this algorithm is that there is no shared knowledge between nearby intervals – if a function is continuous, the performance at interval k is likely to be similar to that of $k-1$ for a large enough number of intervals. Another challenge is the dependence on a tuning parameter for the number of intervals. It is likely that different values for this parameter would significantly affect the results; we chose the parameter recommended by (Auer et al., 2007) ($n = 2$), but also varied this parameter (to $n = 10$) and obtained comparable performance on test functions 1 and 3 and slight improvements on test function 2. This reinforces the authors' observations that bandit algorithm parameters which produce the best theoretical bounds do not always translate to efficient algorithm performance.

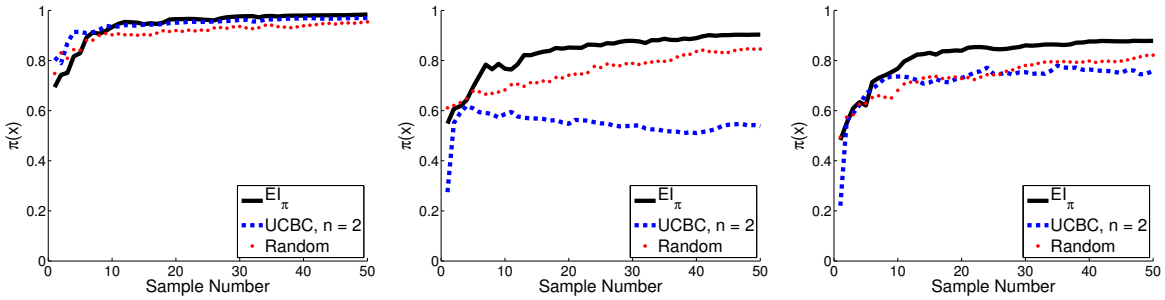
Another challenge is that UCBC is not defined for higher dimensions; the natural extension would be to use a grid of area elements instead of a set of intervals, but the choice of n for each dimension isn't clear; for this reason we limit the results herein to one-dimensional test functions.

We also note that EI_π outperforms the naïve use of Bayesian optimization techniques on the latent GP $\hat{f}$, as shown in Fig. 3. This is largely true because the interpretation of variances on the latent function when used in the classification framework are unintuitive – the variance $\hat{f}_\sigma$ is not based solely on the sampled points as in the regression case; instead larger values of $\hat{f}_\mu$ tend to have larger variances due to the nonlinear mapping into the space of probabilities $\hat{\pi}$.

This problem is especially apparent in test function 3, where the local maxima are given in fairly wide area likely to be sampled during the initial space-filling design, whereas the global maximum is narrower; because the variance of the latent function continues to be high at high values of the mean, and drop off very slowly, both EI_f and UCB_f tend to focus remaining evaluations in this localized area.

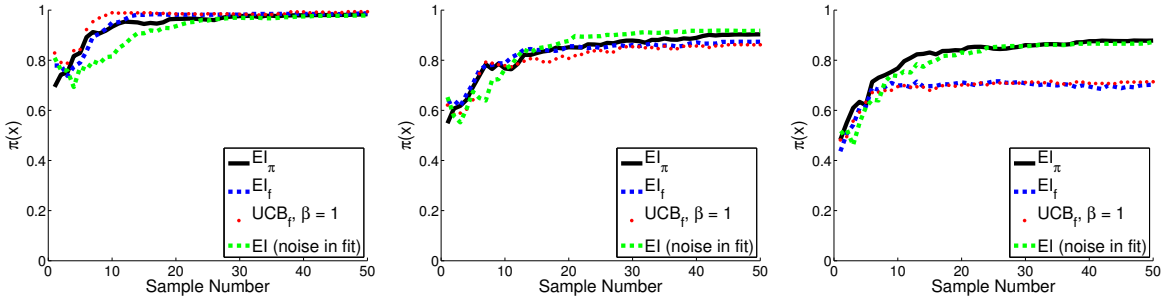
Because EI_π instead uses the posterior in the underlying success probability space, the variance decreases near the local maxima as expected, and the algorithm explores other areas of X with potential for improvement.

Finally, standard EI fit directly to the binary data performs remarkably well, although is slower to converge than EI_π. However, this method required additional careful model selection; we shown the best results after carefully fitting a noise term in the diagonal of the



(a) Test Function 1 (b) Test Function 2 (c) Test Function 3

Figure 2. After each sample, each algorithm was queried as to its recommendation for a point x that would have a maximum expectation of success $\pi(x)$; these results show the underlying probability value of that point averaged over 100 runs of each algorithm. Here we compare the stochastic binary expected improvement (EI_π) to the continuous-armed bandit algorithm UCBC suggested in (Auer et al., 2007) as well as uniform random selection. The cause of the unusual drop in performance of UCBC with more samples has not been determined.



(a) Test Function 1 (b) Test Function 2 (c) Test Function 3

Figure 3. As in Fig. 2, the average expected probability of success for each algorithm's recommendation at each step is shown above. These results, averaged over 100 runs of each algorithm, compare the proposed EI_π algorithm with use of expected improvement and upper confidence bounds on the latent function obtained while fitting a GP to binary data, and EI on a GP fit directly to the 0/1 data with a tuned noise parameter.

covariance; poor selection or omittance of this term resulted in performance far below any baseline shown.

8. Physical Robot Experiments

The goal of this work was to find task parameters that have a high expected success probability and to do so with a small number of experiments that give only binary (success/failure) feedback. The task that motivated this goal was improving the locomotion of a snake robot (Wright et al., 2012) so that it could reliably overcome obstacles encountered in the field, such as the dimensional lumber in these experiments.

Inspired by the approach taken in (Tesch et al., 2012), a master-slave system was set up to record an expert's input to move the robot over an obstacle. Using a sparse function approximation of the expert's input, we created a 7 parameter model that was able to overcome obstacles of various sizes, albeit unreliably – the same parameters would sometimes result in success and sometimes failure. We found that parameters of this model were difficult to optimize by hand to produce reliable results.

Using the EI_π metric in the Bayesian optimization framework described above, 2 and 3 dimensional subspaces of the model were searched to identify regions of the parameter space that resulted in a robust motion over the obstacle that was used to record the original unreliable motion. In each of these cases, running 40 experiments at 20 points[4] resulted in the recommendation of a parameter setting which produced robust, successful motions; the resulting motion is shown in Fig. 4.

[4]The model and test setup resulted in two experiments per selected parameter

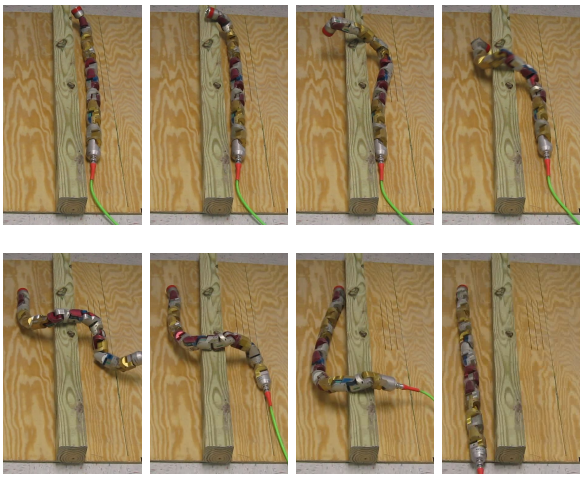
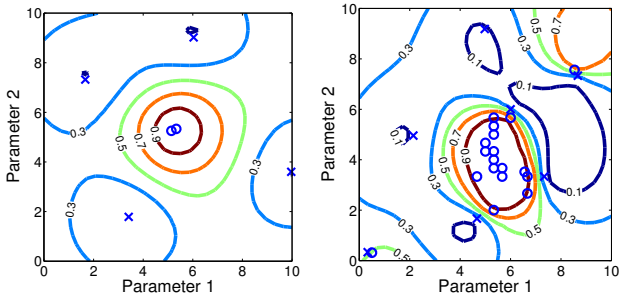


Figure 4. After completion of the optimization, the predicted best parameters result in a robust motion which successfully moves the snake robot over the obstacle.



(a) Experiment selection using EI_π

(b) Experiment selection using EI_π and ignoring areas of high certainty

Figure 5. Selected points and predicted success probability for optimization of robot motion over an obstacle using the EI_π experiment selection metric. The 20 parameters chosen for the 2D optimization are shown as an "O" if they resulted in a success, and an "X" if they resulted in a failure. In (a), the optimization only using the EI_π metric results in pure exploitation after confidently finding a good solution. In (b), avoiding selection of points with a high confidence generates more robust solutions.

After completing the first 2D optimization, it was noted that after the optimization found a successful solution it would not sample other areas of the space. This is because as $\hat{\pi}_\mu$ approaches 1, the maximum possible improvement approaches 0, as does EI_π; this discourages selection of points that are not near the current maximum. While this technically meets the objective of finding a robust solution, there is utility in the use of the remaining experiments to find other robust solutions in the parameter space. To accomplish this goal, we modified the selection metric to not select any point which had a high confidence in its estimate of the true probability. To measure this confidence in being within ϵ of the mean at a point x, we can take the integral

$$C(x) = \int_{max(\bar{\pi}-\epsilon,0)}^{min(\bar{\pi}+\epsilon,1)} p_\pi^x(\bar{y})d\bar{y}. \quad (13)$$

We reran the optimization over the same $2D$ parameter space, not considering points where, for $\epsilon = .1$, $C(x) >= 90\%$. As seen in Fig. 5, this generated a more diverse solution set which provided a more rich set of motions for the robot.

9. Conclusion and Future work

In this paper, we have defined the stochastic binary optimization problem for expensive functions, and presented a novel use of GPC to frame this problem as Bayesian optimization. We also presented a new optimization algorithm that computes expected improvement in the stochastic binary case, outperforming several baseline metrics as well as a leading continuous-armed bandit algorithm. Finally, we used our algorithm to learn a robust motion for moving a snake robot over an obstacle.

The problem we define is not limited to the demonstrated snake robot application, but applies to many expensive problems with parameterized policies and stochastic success/failure feedback. This includes variations of applications where continuous-armed bandits are currently used, such as auction mechanisms and oblivious routing (see references in (Kleinberg, 2004)), which could contain an offline training phase penalizing simple rather than continuous regret.

One promising topic that builds upon the current work is the application of these methods to transfer/multiple task learning. For the robot application, this could include using knowledge from optimization for a single obstacle to improve the learning rate for other similar obstacles, such as different heights and widths of the beam demonstrated here. A final future goal is the derivation of theoretic convergence guarantees for the binary stochastic expected improvement metric.

Acknowledgements

The authors thank Roman Garnett for his insights into the stochastic binary optimization problem. We also appreciate the reviewer comments, especially the suggestion of the direct GP fit to the 0/1 data baseline.

References

Agrawal, Rajeev. The Continuum-Armed Bandit Problem. *SIAM Journal on Control and Optimization*, 33(6):1926–1951, November 1995. ISSN 0363-0129. doi: 10.1137/S0363012992237273.

Auer, Peter, Cesa-Bianchi, N, and Fischer, P. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, pp. 235–256, 2002.

Auer, Peter, Ortner, Ronald, and Szepesvári, C. Improved rates for the stochastic continuum-armed bandit problem. *Learning Theory*, 2007.

Bubeck, Sébastien, Munos, Rémi, and Stoltz, Gilles. Pure exploration in finitely-armed and continuous-armed bandits. *Theoretical Computer Science*, 412 (19):1832–1852, April 2011a. ISSN 03043975. doi: 10.1016/j.tcs.2010.12.059.

Bubeck, Sébastien, Munos, Rémi, Stoltz, Gilles, and Szepesvári, Csaba. X -Armed Bandits. *Journal of Machine Learning Research*, 12:1655–1695, 2011b.

Garnett, Roman, Krishnamurthy, Yamuna, Wang, Donghan, Schneider, Jeff, and Mann, Richard. Bayesian optimal active search on graphs. In *Workshop on Mining and Learning with Graphs*, 2011. ISBN 9781450308342.

Jones, Donald R. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001.

Jones, Donald R., Schonlau, Matthias, and Welch, William J. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*, 13(4), 1998. ISSN 0925-5001.

Kleinberg, Robert. Nearly tight bounds for the continuum-armed bandit problem. In *Advances in Neural Information Processing Systems*, pp. 697–704, 2004.

Kleinberg, Robert and Upfal, Eli. Multi-Armed Bandits in Metric Spaces. In *STOC '08 Proceedings of the 40th annual ACM symposium on Theory of computing*, pp. 681–690, 2008. ISBN 9781605580470.

Minka, Thomas P. *A family of algorithms for approximate Bayesian inference*. Phd thesis, Massachusetts Institute of Technology, 2001.

Mockus, J, Tiesis, V, and Zilinskas, A. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2:117–129, 1978.

Nickisch, Hannes and Rasmussen, CE. Approximations for binary Gaussian process classification. *Journal of Machine Learning Research*, 9:2035–2078, 2008.

Rasmussen, Carl Edward and Williams, Christopher K. I. Gaussian Processes for Machine Learning. URL `http://www.gaussianprocess.org/gpml/code/gpml-matlab.tar.gz`.

Rasmussen, Carl Edward and Williams, Christopher K. I. *Gaussian Processes for Machine Learning*. The MIT Press, 2006. ISBN 026218253X.

Settles, Burr. Active Learning Literature Survey. Technical report, University of Wisconsin–Madison, 2009.

Tesch, Matthew, O'Neill, Alex, and Choset, Howie. Using Kinesthetic Input to Overcome Obstacles with Snake Robots. In *International Symposium on Safety, Security, and Rescue Robotics*, 2012.

Žilinskas, Antanas. A review of statistical models for global optimization. *Journal of Global Optimization*, 2(2):145–153, June 1992. ISSN 0925-5001. doi: 10.1007/BF00122051.

Wright, C., Buchan, A., Brown, B., Geist, J., Schwerin, M., Rollinson, D., Tesch, M., and Choset, H. Design and Architecture of the Unified Modular Snake Robot. In *2012 IEEE International Conference on Robotics and Automation*, St. Paul, MN, 2012.