EFFORT ALLOCATION AND STATISTICAL INFERENCE FOR MULTISTART STOCHASTIC GRADIENT DESCENT

Saul Toscano-Palmerin, Peter I. Frazier
st684@cornell.edu, pf98@cornell.edu
School of Operations Research & Information Engineering
Cornell University, Ithaca, NY 14853

Abstract

Multistart stochastic gradient descent methods are widely used for gradient-based stochastic global optimization. While these methods are effective relative to other approaches for these challenging problems, they seem to waste computational resources: when several starts are run to convergence at the same local optimum, all but one fail to produce useful information; when a start converges to a local optimum worse than an incumbent solution, it also fails to produce useful information. We propose a rule for allocating computational effort across starts, Most Likely to Succeed (MLS), which allocates more resources to the most promising starts. This allocation rule is based on a novel Gaussian-Process-based statistical model (SGD-GP) for a start's limiting objective value. Unlike previously proposed statistical models, ours agrees with known convergence rates for SGD. Numerical results show our approach outperforms equal and random allocation of effort across starts, and a machine learning method.

1 INTRODUCTION

Stochastic gradient descent (SGD) methods (Nemirovski and Yudin 1978, Kingma and Ba 2014) are popular algorithms for training machine learning algorithms (Krizhevsky et al. 2012, Murphy 2012) and optimization via simulation (Fu 2015). Unfortunately, when the objective function is non-convex, these methods may converge to a saddle point or local optimum, instead of a global optimum. Furthermore, even when the objective function is convex, SGD may converge slowly to a global optimum when started from a disadvantageous location.

A common solution to these challenges is multi-start SGD (Schoen 1991, Martí et al. 2016), in which several starting points are chosen uniformly at random, and an SGD algorithm is operated separately starting from each of these points. (We use the term "start" to refer both to an individual starting point, and to the path of SGD starting from an individual point.) If the number of starts is large enough relative to the number of local optima in the problem, and their starting positions are sufficiently diverse, multi-start SGD is likely to find a global optimum.

Multi-start SGD, however, often seems to waste computation. Several starts may converge to the same local optimum, providing essentially the same information about that local optimum's location and value as a single start among them, but consuming several-fold more computation. Also, starts that converge to sub-optimal local optima or stationary points contribute little to the final solution. While simply reducing the number of starts risks failing to find the global optimum, intuition suggests that substantial computational savings might be achieved by intelligently allocating computational effort across starts: if a start seems unlikely to provide a new globally optimal solution, then we can spend less effort computing additional iterations, and focus instead on allocating effort toward the most promising starts.

In this paper, we make two contributions. The first is a rule for sequentially allocating computational effort across starts, called "Most Likely to Succeed" (MLS). This allocation rule is built on a Bayesian statistical model for the limiting value of the objective function along SGD's path from a single start. It chooses to allocate effort to the start that is most likely (according to this statistical model) to provide an objective value at convergence that exceeds an incumbent best solution.

The second contribution is this statistical model, which we call SGD-GP. Using the iterates $x_1, \ldots, x_n$ from one start of SGD, this statistical model calculates a posterior probability distribution on the value of the objective at the limit of the iterate sequence, $f(x_\infty)$, and on the scaled distance between the current iterate and this limit, $\sqrt{n}(x_n - x_\infty)$. This statistical model is derived from asymptotic theory for stochastic gradient descent (Kushner and Yin 2003), which shows that $\sqrt{n}(x_n - x_\infty)$ behaves asymptotically like an Ornstein-Uhlenbeck process under some conditions. This use of the asymptotic theory stands in contrast to existing approaches for modeling paths from SGD, discussed in more detail below. In some cases these existing approaches assume exponential rates of convergence, and none to our knowledge show that their model is consistent with SGD's known asymptotic behavior.

We demonstrate that MLS together with SGD-GP substantially outperforms an Equal Allocation (EA) and Random Allocation rules for allocating effort across starts, and MLS together with Swerky's statistical model (Swersky et al. 2014). EA is a meaningful benchmark because this is typically how effort is allocated in multistart SGD today. Swersky's statistical model is important because it is a non-parametric statistical model used to model learning curves in the machine learning community. Our demonstrations use the 20-dimensional Rosenbrock function, and three one-dimensional objective functions: a concave function (we maximize rather than minimize), a multi-modal function, and an objective whose gradient is nearly zero over a substantial part of the interval. This last interval is intended to replicate problems with many stationary points, as is believed to arise when training the weights of neural networks Bengio et al. (1994). We also provide an implementation of our method on github Toscano-Palmerin, S. and P. I. Frazier (2018).

~~Our method has one primary limitation, which we plan to address in future work: that we assume f has a one-dimensional input. We believe that our approach can be extended to vector-valued input through the use of similar asymptotic theory.~~

Related Work: Previously Boender and Kan (1987) developed Bayesian procedures for choosing the number of starts, assuming equal allocation of effort across starts. This procedure is based on a posterior distribution on the number of local optima, and uses a stopping rule that trades the computational cost of adding more starts against the increased risk of failing to include a start that converges to the globally optimal local optimum. More recently, Frazier et al. (2009) developed a statistical model for the path generated by an SGD-based approximate dynamic programming algorithm, and Chen et al. (2016) presents a method to estimate confidence intervals for the limit of an SGD iterate sequence.

Related work also appears in work from the machine learning community on modeling learning curves (Swersky et al. 2014, Domhan et al. 2015, Klein et al. 2016, Kandasamy et al. 2017, Li et al. 2016). Some of this work studies how loss on a validation dataset changes with respect to the number of iterations in an SGD algorithm (Swersky et al. 2014, Domhan et al. 2015) and with respect to the amount of training data used (Klein et al. 2016). Some of this work (Domhan et al. 2015) focuses primarily on statistical modeling of the path of a single start from SGD without considering effort allocation. In other work (Swersky et al. 2014, Klein et al. 2016), a statistical model is used by a search procedure to find a set of hyperparameters in a low-dimensional vector space for which the test loss provided by SGD after some fixed finite number of iterations is minimal.

Of the work on statistical models for the paths of SGD, ours is unique in that it is derived from SGD's asymptotic convergence theory, and matches known convergence rates. For example, Swersky et al. (2014) assumes exponential convergence rates. In addition, Swersky et al. (2014), Domhan et al. (2015) both model $f(x_n)$ while our approach is built on modeling $\sqrt{n}(x_n - x_\infty)$. In addition, Domhan et al. (2015) proposes a parametric model, while our approach is non-parametric.

Existing work on search procedures from this line of literature differs from our own in that it focuses on choosing an optimal set of hyperpamaters, where the value of a set of hyperparameters is the value after some fixed amount T of effort used to find an approximate local optimum corresponding to a single start. In other words, it focuses on solving $\max_z g(x_T^1; z)$, where $z \in \mathbb{R}^d$ indexes sets of hyperparameter, and $g(x_T^1; z)$ is the *test* loss for a set of model parameters x_T^1 obtained using T iterations of SGD (or a training dataset of size T) to minimize the *training* loss $f(x; z)$ over x with z held fixed. In contrast, we focus on solving $\min_{i=1,\ldots,I} f(x_\infty^i)$ where x^∞ is the limiting point of a sequence of SGD iterates.

In addition, Domhan et al. (2015) differs in that the statistical model is used to terminate paths early, and once a path is terminated it cannot be restarted. Klein et al. (2016) differs in that the amount of training data (analogous to the number of iterations of SGD) is chosen up-front, and cannot be augmented once evaluation begins. Li et al. (2016) differs in that it is not based on a statistical model, but instead on a simple tournament framework. The most closely related of these search papers is Swersky et al. (2014), which uses the statistical model to "freeze" and "thaw" paths from SGD associated with different hyperparameters. This freeze-thaw framework bears similarity to our notion of effort allocation. Our allocation rule, however, is substantially different, as it is computed from the probability of outperforming some incumbent best solution, while Swersky et al. (2014) uses an entropy search criterion (Hennig and Schuler 2012). Our allocation rule is substantially simpler to compute than the entropy search criterion.

Our work is also related to the larger literature on Bayesian optimization (Jones et al. 1998, Forrester et al. 2008, Brochu et al. 2010, Frazier et al. 2009), where a Gaussian process model is used to select points to sample. Indeed, our MLS allocation rule is similar in spirit to the probability of improvement acquisition function (Brochu et al. 2010) that arises in that literature.

Our use of a Bayesian statistical model to allocate computational effort across starts resembles allocation rules in ranking and selection (Bechhofer et al. 1995, Kim and Nelson 2007, 2006), especially Bayesian ranking and selection (Frazier et al. 2009), which allocate effort across a finite set of alternatives to find the best one. These problems differ in that samples in ranking and selection are typically independent and identically distributed, while in our setting samples are correlated and not identically distributed.

The rest of this paper is organized as follows: §2 describes our SGD-GP model in one dimension. §3 describes our Most Likely to Succeed allocation rule in one dimension. §4 extends our SGD-GP model and Most Likely to Succeed allocation rule to high-dimensional spaces. §5 presents numerical experiments. §6 concludes.

2 THE 1-DIMENSIONAL SGD-GP STATISTICAL MODEL ON THE PATH OF SGD

Stochastic gradient descent (SGD) algorithms are used to minimize an objective function $f(x)$ using stochastic gradients, i.e., unbiased estimates of $\nabla f(x)$. SGD iterations are usually of the form

$$X_{n+1} = \Pi_A \left[X_n + \lambda_n Y_n \right], \tag{1}$$

where Π_A is the projection onto some feasible set A (for unconstrained optimization, A can be taken to be the domain of f, and the projection effectively dropped), λ_n is the learning rate at iteration n, and Y_n is a stochastic gradient of $-f$ at X_n. A common choice of the learning rate is given by $\lambda_n := \lambda_0/n$, where λ_0 is a chosen parameter, although other more sophisticated rules are also often considered Powell (2007). In this paper, we assume that $\lambda_n := 1/n$. (We believe our approach can be generalized to other stepsize sequences, and discuss one such extension below.) In addition, we use the term SGD to refer generically to the use of (1) to minimize or maximize a function f, rather than using the separate terminology stochastic gradient ascent (SGA) when we maximize (in which case Y_n is a stochastic gradient of f, rather than $-f$). Whether we are maximizing or minimizing will be clear from context.

In this section, we develop a Bayesian statistical model for the value of the objective function f at the limiting value of a sequence of SGD iterates, $f(X_\infty)$. Within our MLS allocation rule, we will apply this model separately to the sequence of iterates from each start.

We create our statistical model in two steps. In the first step (§2.1), we construct a Bayesian statistical model over the *distance* between the current iterate X_n and the limit point X_∞. This statistical model uses the asymptotic theory of SGD (see the appendix), which shows that $M(n) := \sqrt{n}(X_n - X_\infty)$ behaves like an Ornstein-Uhlenbeck process when n is large. By taking rescaled differences between iterates, and noting that an O-U process is a Gaussian process, we are able to construct a Gaussian process over these observed differences and $M(n)$. We then calculate the conditional distribution of $M(n)$ by conditioning on the observed differences.

In the second step (§2.3), we use the posterior on $M(n)$ to infer the posterior distribution on $f(X_\infty)$. To accomplish this, we use the mean value theorem, an estimate for the slope in this theorem, and the observed value of $f(X_n)$.

Before proceeding with this development, we briefly discuss generalization to other stepsize sequences. If the step size sequence λ_n is not $1/n$ but instead satisfies $o(\lambda_n) = (\lambda_n/\lambda_{n+1})^{1/2} - 1$, then our definition $M(n) := \sqrt{n}(X_n - X_\infty)$ can be changed to $M(n) := (X_n - X_\infty)/\sqrt{\lambda_n}$. By Theorem 2.1 of section 10.2.1 of Kushner and Yin (2003), this new $M(n)$ behaves like an an Ornstein-Uhlenbeck process when n is large under mild assumptions. Thus, similar arguments to the ones given below can be used to build a statistical model of $f(X_\infty)$. More generally, when the limiting behavior of some rescaled version of $X_n - X_\infty$ is understood, we can apply techniques similar to the ones below.

2.1 Inference Over $M(n)$ Given Hyperparameter θ

As discussed above, $M(n) = \sqrt{n}(X_n - X_\infty)$ behaves like an Ornstein-Uhlenbeck process when n is large under mild assumptions. We assume that these mild assumptions hold in the problem studied, and so

$$M(\cdot) \mid \theta \sim GP(0, \Sigma_0(\cdot,\cdot;\theta)),$$

where Σ_0 is the parametric positive kernel of the Ornstein-Uhlenbeck process, which is $\Sigma_0(n,n';\theta) := \sigma^2 e^{-\theta|n-m|}$ (the *kernel*). We additionally place a Bayesian prior distribution π on θ Neal (1997). If we lack a strong prior belief on θ, we may use a flat prior.

We now describe how to compute the posterior distribution on $M(n)$ given θ and the historical data $X_{1:n} = x_{1:n}$. By the conditioning formula for normal random vectors Glasserman (2013), this posterior distribution at time n is given by

$$M(n) \mid X_{1:n} = x_{1:n}, \theta \sim N(\mu_n(n;\theta), \Sigma_n(n,n;\theta))$$

$$\begin{aligned} \mu_n(n;\theta) &= \gamma_n A_n^{-1} c_n^\intercal \\ \Sigma_n(n,n;\theta) &= \Sigma_0(n,n;\theta) - \gamma_n A_n^{-1} \gamma_n^\intercal, \end{aligned}$$

where $\gamma_n := \left(\Sigma_0(n,1;\theta) - \sqrt{\frac{1}{2}}\Sigma_0(n,2;\theta), \ldots, \Sigma_0(n,n-1;\theta) - \sqrt{\frac{n-1}{n}}\Sigma_0(n,n;\theta)\right)$, the vector c_n is defined by $c_n := \left(x_1 - x_2, \ldots, \sqrt{n-1}(x_{n-1} - x_n)\right)$, and the matrix $A_n := (\Sigma_0(i,j;\theta))_{i,j=1}^n$.

2.2 Inference Over $M(n)$, Marginalizing over Hyperparameter θ

We now present the inference methodology given the historical data $x_{1:n}$, which uses the result in the previous section but marginalizes over θ.

We first show to compute $p(\theta \mid X_{1:n} = x_{1:n})$ where X_1 is the first point of the SGD algorithm, which allows us to sample θ from its posterior distribution given $X_{1:n} = x_{1:n}$ via slice sampling (Neal 2003). Assuming that $X_1 = x_1$ is given, the density of the posterior distribution of θ given $X_{1:n} = x_{1:n}$ is

$$\begin{aligned} p(\theta \mid X_{1:n} = x_{1:n}) &\propto P(X_{1:n} = x_{1:n} \mid \theta, X_1 = x_1)\, p(\theta) \\ &= P(X_1 = x_1, \ldots, X_n = x_n \mid \theta, X_1 = x_1)\, p(\theta) \\ &= P\Big(R(n-1) = \sqrt{n-1}\,(x_{n-1} - x_n), \ldots, R(1) = (x_1 - x_2) \mid \theta, X_1 = x_1\Big)\, p(\theta) \end{aligned}$$

under the assumption that the prior on θ is independent of X_1, X_m is the random point at time m of the SGD algorithm, and $R(n) := \sqrt{n}\,(X_n - X_{n+1})$. In order to compute the previous density, we only need to compute the distribution of the vector $(R(1), \ldots, R(n-1))$ given θ.

To compute the distribution of $(R(1), \ldots, R(n-1))$, we first observe that

$$R(n) = \sqrt{n}\,(X_n - X_{n+1}) = M(n) - \sqrt{\frac{n}{n+1}} M(n+1).$$

Since $M(n)$ follows a Gaussian process given θ, we then have that

$$R(\cdot) \mid \theta \sim GP(0, \Gamma_0(\cdot, \cdot; \theta))$$

where

$$\begin{aligned} \Gamma_0(n, m; \theta) &:= \Sigma_0(n, m; \theta) + \sqrt{\frac{n}{n+1}}\sqrt{\frac{m}{m+1}}\Sigma_0(n+1, m+1; \theta) \\ &\quad - \sqrt{\frac{m}{m+1}}\Sigma_0(n, m+1; \theta) - \sqrt{\frac{n}{n+1}}\Sigma_0(n+1, m; \theta). \end{aligned}$$

Since we can compute $p(\theta \mid X_{1:n} = x_{1:n})$, we can sample θ from its posterior distribution using slice sampling.

2.3 Inference Over $f(X_\infty)$

We now complete our description of the SGD-GP method for statistical inference by describing how it infers an approximate posterior distribution for $f(X_\infty)$ given $X_{1:n} = x_{1:n}$ and $f(X_n) = f(x_n)$. It uses the mean-value theorem together with the posterior on $M(n)$ described in the previous section. This model is generalized below in Section 4 where we allow noisy observations of f and extend our model to high-dimensional spaces.

The statistical model presented here requires access to to $f(X_n)$. However, $f(X_n)$ can only be observed with noise in many applications of SGD. When noise-free observations of $f(X_n)$ are not available, the value of $f(X_n)$ we use in SGD-GP may be replaced by an average of many noisy observations. Below in Section 3 we describe a batch allocation strategy that uses this approach. This batching strategy obtains multiple replications of $f(X_n)$ for only a small fraction of the overall iterates X_n, and thus represents a small increase in effort compared to applying SGD-GP and MLS where $f(X_n)$ could be observed without noise.

To infer $f(X_\infty)$, we first note that the mean-value theorem implies

$$f(X_\infty) = f(X_n) + L_n |X_\infty - X_n| \tag{2}$$

and

$$f(X_\infty) = f(X_{n-1}) + L_{n-1} |X_\infty - X_{n-1}|,$$

for non-negative numbers L_n, L_{n-1}. When n is large, L_n is close to L_{n-1} because X_n is close to X_{n+1}, and they are both close to X_∞. Thus we assume that $L_n \approx L_{n-1}$, and so

$$\begin{aligned} f(X_n) - f(X_{n-1}) &= L_n |X_\infty - X_{n-1}| - L_n |X_\infty - X_n| \\ &= L_n \left(\frac{|M(n-1)|}{\sqrt{n-1}} - \frac{|M(n)|}{\sqrt{n}} \right). \end{aligned}$$

To approximate L_n, we use plug-in estimators for $|M(n-1)|$ and $|M(n)|$ equal to their expectation under the posterior, leverage our assumption that $f(x_n)$ and $f(x_{n-1})$ are observed without noise, where $X_n = x_n$ and $X_{n-1} = x_{n-1}$, and then solve the previous equation for L_n to obtain the estimator given θ,

$$\hat{L}_n(\theta) = \frac{|f(x_n) - f(x_{n-1})|}{\left| E\left[\frac{|M(n-1)|}{\sqrt{n-1}} - \frac{|M(n)|}{\sqrt{n}} \mid X_{1:n} = x_{1:n}, \theta \right] \right|}.$$

We can then use this estimator $\hat{L}_n$ along with (3) and a noise-free observation of $f(x_n)$ to describe the posterior distribution of $f(x_\infty)$ as,

$$f(X_\infty) \mid x_{1:n} \sim f(x_n) + \frac{1}{\sqrt{n}} Z_n$$

where the density of Z_n is given by

$$g(z) = \int_\theta p(\theta \mid x_{1:n}) \, \gamma_\theta(z) \, d\theta,$$

where γ_θ is the density of the absolute value of the conditionally (given θ) normal random variable $\hat{L}_n(\theta) M(n)$. (Here, $\hat{L}_n(\theta)$ is treated as conditionally constant given θ, and $M(n)$ is random.)

3 THE MOST LIKELY TO SUCCEED ALLOCATION RULE

As discussed above, multi-start stochastic gradient ascent can waste computation: typically only the limiting value of the best start is used as the final solution, but a substantial amount of computation is spent computing iterates for other starts. In this section, we use the statistical model from the previous section for choosing how to allocate computational effort across starts. The goal is to allocate more effort to starts that are likely to produce good limiting values, so that a final solution of equal quality can be produced with less computational effort. We call this rule for allocating effort Most Likely to Succeed (MLS).

To support our definition, we first augment our existing notation, using X_n^i to indicate the n^{th} iterate of SGD using start i. We let I denote the number of starts. The starting values of X_1^i may be generated arbitrarily, but we recommend choosing them by sampling uniformly at random from the input space. Then, X_n^i is given recursively from X_{n-1}^i by (1).

We consider a situation in which starts are advanced one at a time, with an allocation rule deciding which start to advance to its next iteration in each timestep. We will let $t \in \mathbb{Z}_+$ count the number of iterations that have been performed across all of the starts, and we let $n(t,i)$ count the number of iterations performed for start i by time t. Thus, $X_{n(t,i)}^i$ is the value of start i at time t.

Formally, an allocation rule is a sequence of mappings, one for each t, from the observable state $H_t := (X_m^i : m \le n(t,i), i \in [I])$ to $[I] = \{1, \ldots, I\}$. Let π_t be the mapping for time t, so that $\pi_t(H_t)$ is the start that the allocation rule chooses to advance next at time t. As a result of operating a particular policy, the number of iterations assigned to each start over time, $(n(t,i) : t \ge 0, i \in [I])$ is defined by $n(t+1,i) = n(t,i) + 1\{\pi_t(H_t) = i\}$.

Our MLS allocation rule uses the previously described SGD-GP statistical model on the value of the limiting objective obtained from each start, $f(X_\infty^i)$. We first describe this allocation rule in the setting where $f(x)$ can be observed without noise, and only the gradient $\nabla f(x)$ is noisy. In this setting, the MLS allocation rule allocates

the next unit of computational effort to the start whose limiting objective value is most likely to be at least $\varepsilon > 0$ better than the best objective value seen so far. Formally, this rule is defined as,

$$\pi_t(H_t) := \text{argmax}_i P\left(f\left(X_\infty^i\right) > Y_t + \varepsilon \mid H_t\right),$$

where Y_t is the best value of the objective function f seen by time t, and ε is a positive number.

This allocation rule naturally balances exploration and exploitation. First, it favors allocating effort to those starts that are predicted to have a high value for $f(X_\infty^i)$, thus providing exploitation. Second, starts with few iterations will naturally have a great deal of uncertainty about its limiting value. If we imagine that the posterior distribution on $f(X_\infty^i)$ as approximately normally distributed, and we think of the uncertainty as quantified by the variance under this posterior, then $P\left(f\left(X_\infty^i\right) > Y_t + \varepsilon \mid H_t\right)$ will approach $1/2$ as this uncertainty grows large. Conversely, another start that has enough iterations to be close to convergence will have only a very small amount of uncertainty, and $P\left(f\left(X_\infty^i\right) > Y_t + \varepsilon \mid H_t\right)$ will approach 0 for any positive ε as this uncertainty shrinks to 0. This suggests that all starts, even though with a value for $f(X_\infty^i)$ predicted to be poor, do get advanced forward eventually, and that we explore as well as exploit.

Although we do not offer a proof here, we conjecture that these ideas can be used to show that all starts are advanced infinitely often with probability 1 under MLS, and that this in turn employs that MLS allocation rule provides an asymptotically consistent estimator of $\max_i f(X_\infty^i)$. We also conjecture that the parameter ε can be used to trade off exploration vs. exploitation, with larger values of ε leading to more effort allocated to starts with substantial uncertainty, and smaller values to more exploitation. However, in our numerical experiments we simply set $\varepsilon = 0.1$.

When observations of $f(x)$ are obscured by noise, we estimate it by repeated sampling. If we did this after each sample, then the savings generated using MLS would be overwhelmed by the extra effort required to estimate $f(x)$ after each sample. Thus, we do this within a batch framework where MLS is used to allocate a batch of B SGD iterations to a start, and then we perform repeated simulation on the start's iterate at the end of this batch. This cuts the number of replications used to estimate $f(x)$ by a factor of B. It does reduce the responsiveness of MLS: when it would be clear from, say, $B/2$ iterations that a start is performing poorly and effort should be allocated elsewhere, the batching strategy continues to allocate the rest of the batch to this start. One additional benefit of evaluating $f(X_{n(t,i)}^i)$ using repeated simulation after each batch is that progress can be tracked in a simple and easy-to-communicate way. Indeed, a similar approach is often used when training neural networks, where progress is evaluated after each epoch.

We describe this batching strategy in detail here:

1. Choose a batch size B and a number of replications L.
2. Run B iterations of SGD for each start $i \in I$.
3. For each $i \in I$, let $\hat{f}_t^i$ be an estimate of f at $X_{n(t,i)}^i$ obtained by averaging L independent replications.
4. While budget remains:
   (a) Choose a starting point i using the MLS allocation rule, where $f(X_t^i)$ is replaced by $\hat{f}_t^i$.
   (b) Run B iterations of SGD for start i.
   (c) Let $n(t+B,i) = n(t,i) + B$, $n(t+B,j) = n(t,j)$ for $j \neq i$.
   (d) Let $t = t + B$.
   (e) Let $\hat{f}_t^i$ be an estimate of f at $X_{n(t,i)}^i$ obtained by averaging L independent replications.

4 THE GENERAL MOST LIKELY TO SUCCEED ALLOCATION RULE

In this section, we extend our statistical model (GD-GP) to high-dimensional spaces, and we allow noisy observations of the objective function without having to use the batching strategy described above. This new statistical model can be used within our allocation rule Most Likely to Succeed (MLS) as described in the previous section.

As discussed previously, $M(n) = \sqrt{n}\,(X_n - X_\infty)$ behaves like an Ornstein-Uhlenbeck process, so we assume that

$$M_i(\cdot) \mid \theta_i \sim GP\left(0, \Sigma_0\left(\cdot,\cdot;\theta_i\right)\right),$$

where $M_i(\cdot)$ is the $i-th$ entry of $M(\cdot)$, $\Sigma_0(n,n';\theta_i) := \sigma_i^2 e^{-\theta_i|n-m|}$. We additionally place a Bayesian prior distribution π on $(\theta_1,\ldots,\theta_d)$ where d is the dimension of the domain of the problem.

Following the same argument given in §2, we can do inference over M_i marginalizing over hyperparameters θ_i, which allows us to do inference over M.

To infer $f(X_\infty)$, we first note that the Taylor theorem implies

$$f\left(X_\infty\right) \approx f\left(X_n\right) + \nabla f\left(X_n\right)\left(X_\infty - X_n\right). \tag{3}$$

Thus, we only need to estimate $f\left(X_n\right)$ and $\nabla f\left(X_n\right)$ since we already have a statistical model for $(X_\infty - X_n)$. In order to allow noisy observations, we assume that locally the objective function satisfies:

$$f\left(x\right) \approx \frac{1}{2}\sum_{i=1}^{d} a_i\left(x_i - b_i\right)^2 + c.$$

The previous parameters a_i, b_i and c are updated on the fly. In order to do that, we use the following linear relationship: $\nabla f\left(x\right)_i \approx a_i(x_i - b_i)$. We then estimate each a_i and b_i via linear regression on the fly: we maintain exponential moving averages for each dimension $\left\{\bar{h}_{k,i}, \bar{x}_{k,i}, \bar{x^2}_{k,i}, \bar{xh}_{k,i}\right\}$ where $h_{k,i} = \nabla f\left(x_k\right)_i$. Similarly, we maintain exponential moving averages of $\tilde{f}\left(x_k\right) - \frac{1}{2}\sum_{i=1}^{d} a_i\left(x_{k,i} - b_i\right)^2$ to estimate c where $\tilde{f}\left(x_k\right)$ is the noisy observation of $f(x_k)$.

Consequently, the previous arguments can be used to infer $f(X_\infty)$, and then use this statistical model within our allocation rule Most Likely to Succeed (MLS).

5 NUMERICAL EXPERIMENTS

In this section we present numerical experiments exploring the accuracy and coverage of our statistical approach, and the efficiency of our allocation rule in maximizing functions using multi-start stochastic gradient ascent.

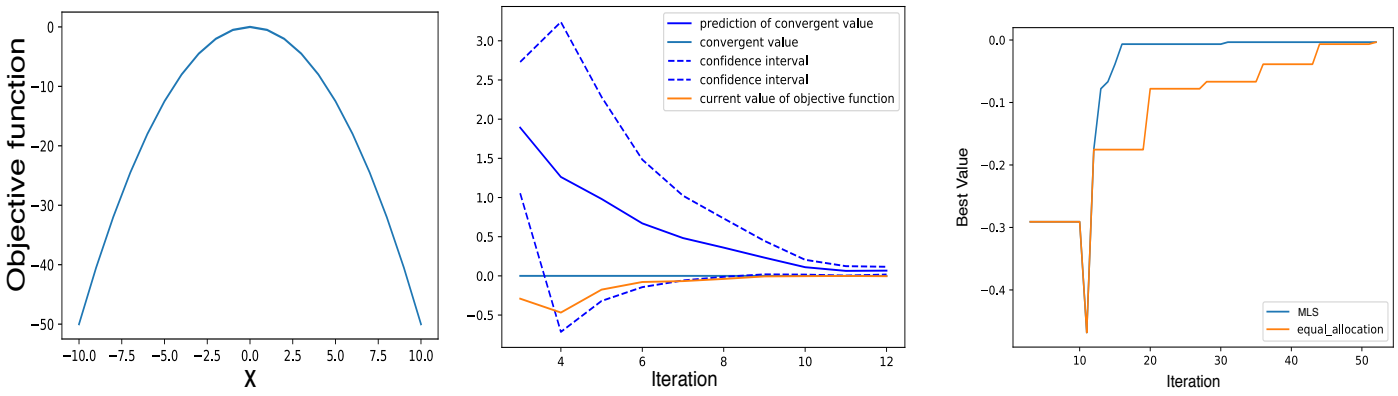
We compare our MLS allocation rule (§3) against two allocations rules. We consider an equal allocation (EA) rule in which each starting point is chosen the same number of times in round-robin fashion. Formally, this rule is $\pi_t(H_t) = n \bmod I$. In addition, we consider the statistical model defined in Swersky et al. (2014) used with the MLS allocation rule. This rule differs from MLS in its statistical model alone. Experiments demonstrate MLS significantly outperforms these benchmarks on one-dimensional problems in which we can observe function values (but not gradients) without noise , and on the 20-dimensional Rosenbrock function with noisy evaluations of the objective.

We compare on maximization problems: a concave maximization problem (§5.1), a maximization problem with many local maxima (§5.2), a maximization problem where the gradient is almost zero in some complete intervals of the domain (§5.3), and a maximization problem of the 20-dimensional Rosenbrock function (§5.4).

We assume that we observe the objective functions without noise, and that gradients are observable with independent normally distributed noise.

5.1 A Concave Objective Function

Here we compare MLS and EA on a concave maximization problem $\max_x f(x) := \max_x -0.5x^2$ where the stochastic gradient is equal to the true gradient plus standard normal noise. All starting points converge to the same point in this problem. Figure 1 pictures the results using 9 starting points. MLS identifies the optimal solution faster.



(a) Objective function $f(x) := -0.5x^2$

(b) SGD-GP's predictions and confidence intervals for the limiting value of SGD (0) from a single start.

(c) Performance comparison between the MLS and equal allocation rules using 9 starting points.

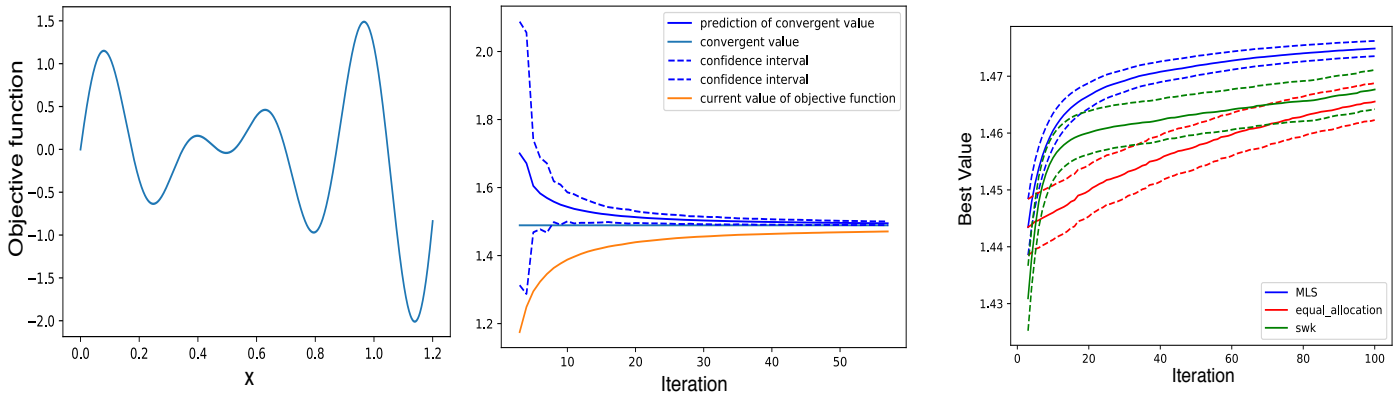
Figure 1: The SGD-GP statistical model (b) and MLS allocation rule (c) on the problem (a) from §5.1.

5.2 An Objective Function with Many Local Maxima

Here we consider an objective function with many local maxima, $f(x) := (1.4 - 3x)\sin 18x$, with a domain of $[0, 1.2]$. The stochastic gradient is equal to the true gradient plus standard normal noise. Figure 2 compares the performance of the MLS and EA with 20 starting points, plotting the number of iterations beyond the first stage on the x axis, and the average maximum solution. We average over 1200 independent runs of MLS equal allocation, and Swersky's statistical model with the MLS allocation rule. Our allocation rule identifies the optimal solution much faster than these other rules.

5.3 Objective Function with a Vanishing Gradient

Here we consider an objective function whose gradient is almost zero in a portion of the domain, $f(x) := (x + \sin x)e^{-x^2}$, with a domain of $[-10, 10]$. The stochastic gradient is equal to the true gradient plus normally distributed noise with mean 0 and variance 100. Figure 3 compares the performance of MLS and equal allocation with 20 starting points, plotting the number of iterations beyond the first stage on the x axis, and the average maximum solution. We average over 1100 independent runs of MLS , equal allocation, and Swersky's statistical model with the MLS allocation rule. Our allocation rule identifies the optimal solution much faster than these other rules.

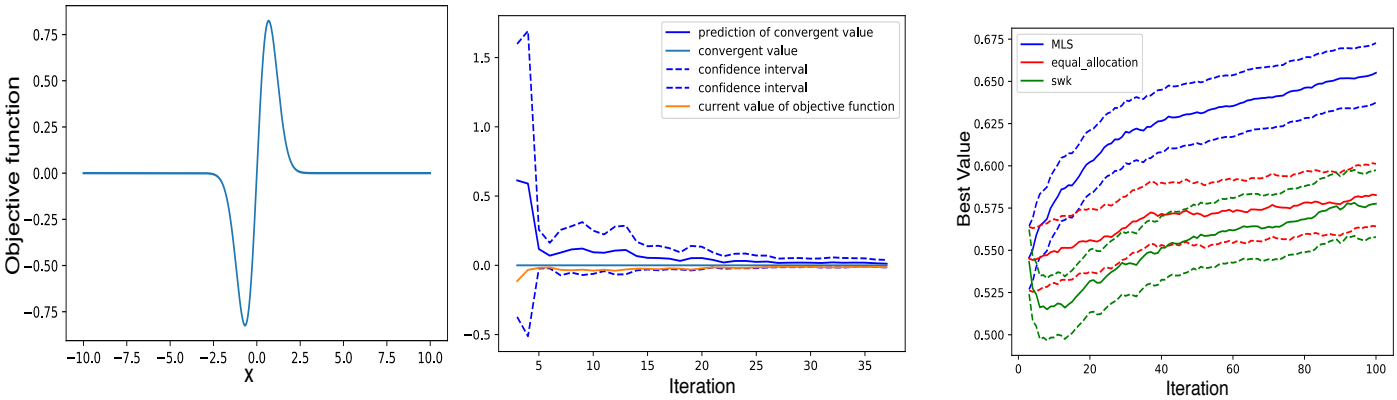


(a) Objective function $f(x) := (1.4 - 3x)\sin 18x$

(b) SGD-GP's predictions and confidence intervals for the limiting value of SGD (1.49) from a single start.

(c) Performance comparison between MLS equal allocation, and Swersky's statistical model with the MLS rule using 20 starting points.

Figure 2: The SGD-GP statistical model (b) and MLS allocation rule (c) on the problem (a) from §5.2.



(a) Objective function $f(x) := (x + \sin x)e^{-x^2}$

(b) SGD-GP's predictions and confidence intervals for the limiting value of SGD (0.0) from a single start.

(c) Performance comparison between MLS equal allocation, and Swersky's statistical model with the MLS rule using 20 starting points.

Figure 3: The SGD-GP statistical model (b) and MLS allocation rule (c) on the problem (a) from §5.3.

5.4 The 20-Dimensional Rosenbrock Function

Here we consider the 20-dimensional Rosenbrock function:

$$f(x) = \sum_{i=1}^{19} \left[100 \left(x_{i+1} - x_i^2 \right)^2 + (1 - x_i)^2 \right].$$

We assume that we can only observe noisy evaluations of f: $y(x) = f(x) + \varepsilon(x)$ where $\varepsilon(x) \sim N(0, 0.1)$. The stochastic gradient is equal to the true gradient plus normally distributed noise with mean 0 and variance 0.1. We average over 1000 independent runs of MLS equal allocation, and random allocation using 30 starting points. Figure 4 shows that MLS performs much better than the other policies considered.

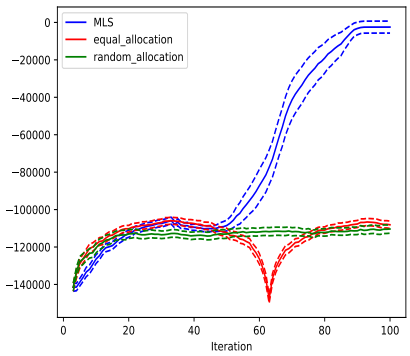


Figure 4: Performance comparison between MLS equal allocation, and random allocation using 30 starting points.

5.5 Comparison of SGD-GP to a Statistical Model with Exact Gradients

We also compare the accuracy of the approximations SGD-GP to an alternate statistical model that can be constructed using exact gradients, and that avoids the approximations SGD-GP makes in section 2.3.

Recall that Section 2.3 performed inference over $f(x_\infty)$ using the mean value theorem, the relationship $f(x_\infty) = f(x_n) + L_n |x_\infty - x_n|$, and an estimate of L_n based on approximations. Another method we considered was to use Taylor's theorem to write,

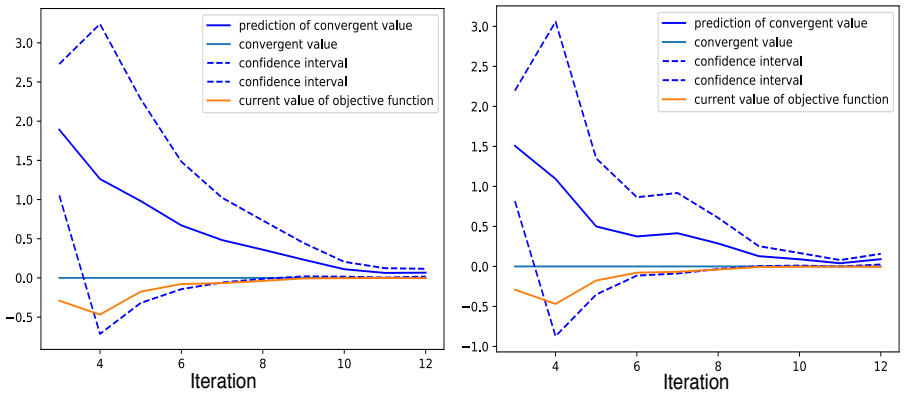
$$f(x_\infty) \approx f(x_n) - \nabla f(x_n) \frac{M_n}{\sqrt{n}}.$$

Then, if we had access to exact observations of $f(x_\infty)$ or could estimate it with high precision based on repeated simulation after each batch, then we could use this relationship together with our posterior on $M(n)$ to construct a posterior on $f(x_\infty)$.

Figure 5 compares the posterior distributions created by the two methods on the problem from Sections 5.1, where SGD-GP has access only to stochastic gradients, and this alternate method sees exact gradients. We see that the inference performed by the two methods is similar.

6 CONCLUSION

We presented Most Likely to Succeed (MLS), which is a new allocation rule across starts that decreases computational effort when using multi-start stochastic gradient ascent to globally optimize a function. MLS depends



(a) SGD-GP

(b) Alternative statistical method with exact gradients

Figure 5: Comparison between SGD-GP and an alternate statistical model that assumes exact gradient observations and avoids the approximations SGD-GP uses in Sections 2.3. Data is generated from the quadratic problem from §5.1. The predictions and confidence intervals from the two methods are similar.

on a new non-parametric statistical model, which is derived from the asymptotic theory of stochastic gradient ascent. It outperforms two benchmarks in numerical experiments.

ACKNOWLEDGMENTS

The authors were partially supported by NSF CAREER CMMI-1254298, NSF CMMI-1536895, and AFOSR FA9550-15-1-0038.

Appendix

A APPENDICES

In this section we state two theorems from Kushner and Yin (2003) about limit theory of SGD algorithms. The first theorem shows the existence of limit points of SGD algorithms. The second theorem shows that if convergence occurs, the normalized sequence of points of SGD $M_n := \sqrt{n}(x_n - x_\infty)$ converges weakly to a known stochastic process, which is a Ornstein-Uhlenbeck process under some assumptions.

Theorem 1. *(Theorem 2.1 of Section 5.2 of Kushner and Yin (2003)) Suppose that the learning rates $\{\lambda_n\}$ of the stochastic gradient descent satisfy that $\sum_{n=1}^{\infty} \lambda_n = \infty$, $\lambda_n \geq 0$, $\lambda_n \to 0$ and $\sum_{n=1}^{\infty} \lambda_n^2 < \infty$. In addition, suppose the following*

1. *$sup_n E\left[|Y_n|^2\right] < \infty$.*
2. *There is a measurable and continuous function g and random variables β_n such that*

$$E_n Y_n = E\left[Y_n \mid x_1, Y_i, i < n\right] = g(x_n) + \beta_n,$$

and $\sum_i \lambda_i |\beta_i| < \infty$ a.s.

Then $\{x_n\}$ converges to some limit set of the ODE $\dot{x} = g(x)$ in A (see section 5.2 of Kushner and Yin (2003)). If the objective function f is continuously differentiable, $g = -\nabla f$, and f is constant on each of the disjoint compact and connected subsets S_j of the stationary points, then x_n converges almost surely to a unique S_i.

Theorem 2. *(Theorem 2.1 of section 10.2.1 of Kushner and Yin (2003)) Let x_∞ be a limit point in the interior of A. Suppose that there is a measurable and continuous function g such that $E_n Y_n = E[Y_n \mid x_1, Y_i, i < n] = g(x_n)$. In addition, assume that*

1. $\lambda_n := 1/n$.
2. $\left\{Y_n 1_{\{|x_n - x_\infty|\} \leq \rho}\right\}$ *is uniformly integrable for small* ρ.
3. $M_n := \left(\frac{x_n - x_\infty}{\sqrt{\varepsilon_n}}\right)$ *is tight.*
4. $E_n Y_n = g_n(x_n)$, *where* g_n *is continuously differentiable for each n, and* $g_n(x) = g_n(x_\infty) + g'_{n,x}(x_\infty)(x - x_\infty) + o(|x - x_\infty|)$.
5. $\lim_{n,m} \frac{1}{\sqrt{m}} \sum_{i=n}^{n+mt-1} g_i(x_\infty) = 0$, *where the limit is uniform in some bounded t-interval.*
6. *There is a Hurwitz matrix Q (i.e., the real parts of the eigenvalues of Q are negative) such that*

$$\lim_{n,m} \frac{1}{m} \sum_{i=n}^{n+m-1} \left[g'_{i,x}(x_\infty) - Q\right] = 0,$$

   and $Q + I/2$ *is also a Hurwitz matrix.*

7. *For some* $p > 0$ *and small* $\rho > 0$, $\sup_n E|\delta R_n|^{2+p} 1_{\{|x_n - x_\infty| \leq \rho\}} < \infty$, *where* $\delta R_n = Y_n - E[Y_n \mid x_1, Y_i, i < n]$.

We then have that the process $M^n(t) := \frac{(x_{n+i} - x_\infty)}{\sqrt{\varepsilon_{n+i}}}$ *if* $t \in [i, i+1]$, *converges weakly to*

$$M(t) := \int_{-\infty}^{t} e^{(Q+I/2)(t-s)} dW(s)$$

where W is a Wiener process with some covariance matrix Σ_1.

In the previous theorem, the limit of the process $M^n(t)$ is defined by the SDE $dM(t) = (Q + I/2)M(t)dt + dW(t)$, and so M is an Ornstein-Uhlenbeck process when the domain is an interval in $\mathbb{R}$, because $Q + I/2$ is negative in this case.

References

Bechhofer, R. E., T. Santner, and D. M. Goldsman. 1995. *Design and Analysis of Experiments for Statistical Selection, Screening and Multiple Comparisons.* New York, NY: J.Wiley and Sons.

Bengio, Y., P. Simard, and P. Frasconi. 1994. "Learning Long-Term Dependencies with Gradient Descent Is Difficult.". *IEEE transactions on neural networks* 5(2):157–166.

Boender, C. G. E., and A. Kan. 1987. "BBayesian Stopping Rules for Multistart Global Optimization Methods.". *Mathematical Programming* 37(1):59–80.

Brochu, E., V. M. Cora, and N. De Freitas. 2010. "A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning". *arXiv preprint arXiv:1012.2599*.

Chen, X., J. Lee, X. Tong, and Y. Zhang. 2016. "Statistical Inference for Model Parameters in Stochastic Gradient Descent.". *arXiv preprint arXiv:1610.08637*.

Domhan, T., J. Springenberg, and F. Hutter. 2015. "Speeding Up Automatic Hyperparameter Optimization of Deep Neural Networks by Extrapolation of Learning Curves". In *IJCAI*, edited by Q. Yang and M. Wooldridge, Volume 15, 3460–3468. Palo Alto, California: AAAI Press.

Forrester, A., A. Sobester, and A. Keane. 2008. *Engineering Design via Surrogate Modelling: A Practical Guide*. Chichester, UK: John Wiley & Sons.

Frazier, P., W. Powell, and S. Dayanik. 2009. "The Knowledge-Gradient Policy for Correlated Normal Beliefs". *INFORMS Journal on Computing* 21(4):599–613.

Frazier, P., W. Powell, and H. SimÃ£o. 2009. "Simulation Model Calibration with Correlated Knowledge-Gradients.". In *Proceedings of the 2009 Winter Simulation Conference*, edited by M. D. Rossetti et al., 339–351. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Fu, M. C. 2015. "Stochastic Gradient Estimation". In *In Handbook of simulation optimization*, edited by M. C. Fu, 105–147. New York, NY: Springer.

Glasserman, P. 2013. *Monte Carlo Methods in Financial Engineering*. New York, NY: Springer Science & Business Media.

Hennig, P., and C. Schuler. 2012. "Entropy Search for Information-Efficient Global Optimization.". *Journal of Machine Learning Research* 13(6):1809–1837.

Jones, D. R., M. Schonlau, and W. J. Welch. 1998. "Efficient Global Optimization of Expensive Black-Box Functions". *Journal of Global Optimization* 13(4):455–492.

Kandasamy, K., G. Dasarathy, J. Schneider, and B. Poczos. 2017. "Multi-Fidelity Bayesian Optimisation with Continuous Approximations.". *arXiv preprint arXiv:1703.06240.*.

Kim, S., and B. Nelson. 2006. "Selecting the Best System". In *Handbook in Operations Research and Management Science: Simulation*, edited by S. Henderson and B. Nelson, 501–534. Elsevier, Amsterdam.

Kim, S., and B. Nelson. 2007. "Recent Advances in Ranking and Selection". In *Proceedings of the 2007 Winter Simulation Conference*, edited by S. Henderson et al., 162–172. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Kingma, D., and J. Ba. 2014. "Adam: A Method for Stochastic Optimization.". *arXiv 1412.6980*.

Klein, A., S. Falkner, S. Bartels, P. Hennig, and F. Hutter. 2016. "Fast Bayesian Optimization of Machine Learning Hyperparameters on Large Datasets". *arXiv preprint arXiv:1605.07079*.

Krizhevsky, A., I. Sutskever, and G. Hinton. 2012. "Imagenet classification with deep convolutional neural networks". In *Advances in Neural Information Processing Systems*, 1097–1105. Lake Tahoe, Nevada: Curran Associates, Inc.

Kushner, H., and G. Yin. 2003. *Stochastic Approximation and Recursive Algorithms and Applications (Vol. 35)*. New York, NY: Springer Science & Business Media.

Li, L., K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar. 2016. "Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization.". *arXiv preprint arXiv:1603.06560.*.

Martí, R., J. Lozano, A. Mendiburu, and L. Hernando. 2016. "Multi-Start Methods". In *Handbook of Heuristics*, 1–21. New York, NY: Springer International Publishing.

Murphy, K. P. 2012. *Machine Learning: A Probabilistic Perspective*. Cambridge, MA: MIT press.

Neal, R. 1997. "Monte Carlo Implementation of Gaussian Process Models for Bayesian Regression and Classification". *arXiv physics/9701026*.

Neal, R. M. 2003. "Slice Sampling". *Annals of Statistics* 31(3):705–741.

Nemirovski, A., and D. Yudin. 1978. "On Cezari's Convergence of the Steepest Descent Method for Approximating Saddle Point of Convex-Concave Functions.". *Soviet Math* 19.

Powell, W. B. 2007. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, Volume 703. Hoboken, NJ: John Wiley & Sons.

Schoen, F. 1991. "Stochastic Techniques for Global Optimization: A Survey of Recent Advances". *Journal of Global Optimization* 1(3):207–228.

Swersky, K., J. Snoek, and R. Adams. 2014. "Freeze-Thaw Bayesian Optimization.". *arXiv preprint arXiv:1406.3896*.

Toscano-Palmerin, S. and P. I. Frazier 2018. "Most Likely to Succeed Code". `https://github.com/toscanosaul/bayesian_quadrature_optimization/tree/master/multi_start`. Accessed May 10th, 2018.