Bayesian optimization with shape constraints

Michael Jauch
Department of Statistical Science
Duke University
michael.jauch@duke.edu

Víctor Peña
Department of Statistical Science
Duke University
vp58@stat.duke.edu

Abstract

In typical applications of Bayesian optimization, minimal assumptions are made about the objective function being optimized. This is true even when researchers have prior information about the shape of the function with respect to one or more argument. We make the case that shape constraints are often appropriate in at least two important application areas of Bayesian optimization: (1) hyperparameter tuning of machine learning algorithms and (2) decision analysis with utility functions. We describe a methodology for incorporating a variety of shape constraints within the usual Bayesian optimization framework and present positive results from simple applications which suggest that Bayesian optimization with shape constraints is a promising topic for further research.

1 Introduction

Prior knowledge about an unknown function often pertains to the shape of the function with respect to an argument. Consider growth curves in biology: excluding rare incidents, a child's height grows monotonically as a function of his or her age. Incorporating shape constraints such as monotonicity, convexity, or concavity is a well-established topic in function estimation. Such constraints, when warranted, can lead to more efficient and stable function estimates compatible with our prior understanding.

In this paper, we explore the possibility of introducing shape constraints on objective functions in Bayesian optimization. We argue that such constraints are often appropriate in two common application areas: (1) hyperparameter tuning of machine learning algorithms and (2) decision analysis with utility functions. We present an example wherein we optimize the hyperparameters of a support vector machine (SVM) [5] and a problem described in Müller and Parmigiani [11] in which we find the optimal sample size for a binomial experiment.

Since Gaussian processes (GPs) are standard in Bayesian optimization, it would behoove us to stay within that framework. Recent work has looked at ways in which GPs can accomodate shape constraints [13], [10], [18]. One approach is based on the convenient property that the partial derivative processes of a GP (with a sufficiently smooth covariance function) are jointly Gaussian with the original process [12]. Thus, one can obtain approximate shape constraints on the original process by imposing conditions on the partial derivatives at a sufficient number of points. We apply this work and expand upon it by introducing a quasiconvexity constraint which is particularly well-suited for applications involving tuning regularization hyperparameters.

The most common prior information used in Bayesian optimization relates to the smoothness of the objective function and is included via the covariance function. While prior knowledge of a function's smoothness allows one to guess what is happening in a neighborhood of an observation, a shape constraint allows one to extrapolate from even a small number of observations to draw conclusions about what is happening globally. Given that certain types of global shape information allow for the design of efficient global optimization algorithms [3], it is not hard to imagine that such information

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

can also improve the efficiency of Bayesian optimization procedures. Improvements may extend to cases when the objective function only approximately satisfies the shape constraint, or more generally to cases where a convexification of the problem is beneficial.

1.1 Hyperparameter tuning

Much of the recent work on Bayesian optimization has focused on the problem of tuning the hyperparameters of machine learning algorithms (see Snoek et al. [14] for an overview). Recent work by Klein et al. [8] and Swersky et al. [16] has taken advantage of the predictable behavior of error or runtime curves to extrapolate what will happen with a larger data set or more training, but there is shape information to be exploited even in more straightforward applications.

As our primary example, we consider hyperparameters controlling model complexity. Many authors have commented on the characteristic shape of the generalization error curve as a function of such a hyperparameter. See Figure 2.11 of Hastie et al. [7] or Figure 5.3 of Goodfellow et al. [6]. In the latter book, the authors refer to the curves as being "U-shaped." This is not meant to be precise but rather to convey the idea that extreme hyperparameter values, corresponding to very low or very high complexity models, have high generalization error, due to underfitting or overfitting, respectively; the best generalization performance is achieved by a hyperparameter lying somewhere between the extremes. In more mathematical language, we could say these curves are smooth and *quasiconvex* (see section 2.1 for a formal definition of quasiconvexity).

This is useful prior information. Because we can formulate the quasiconvexity constraint in terms of the partial derivatives of the generalization error surface, we can incorporate it into our GP prior using our previous observation about partial derivative processes. We apply this idea in section 3.2 to tune the hyperparameters of an SVM.

1.2 Decision analysis with utility functions

Another area of potential application is expected utility maximization. In particular, shape constraints are likely to prove useful in optimizing elicited multiattribute utility functions and in Bayesian experimental design.

Multiattribute utility functions In some decision problems, the shape of the utility function with respect to an attribute is determined if we accept certain assumptions. For example, utility as a function of monetary rewards is concave and monotonically increasing for a risk-averse agent. Incorporating shape constraints not only improves efficiency by reducing posterior variance, but can correct for systematic elicitation biases that manifest themselves in deviations from the shape constraints dictated by rational choice theory [17].

Bayesian experimental design In most Bayesian models, the posterior expected utility is a high-dimensional integral which is estimated through Markov Chain Monte Carlo methods. The intensive computation and simulation error involved imply that we can only obtain noisy expected utility estimates at a limited number of design points. It is often the case that the utility as a function of sample size satisfies a convexity or quasiconvexity constraint, reflecting the trade off between uncertainty reduction and sampling costs. In Section 3.1, we consider the binomial sample size problem from [11] with Bayesian optimization and a convexity constraint on the expected utility surface.

2 Method

The shape constraints we are concerned with are *componentwise* shape constraints. Each of the motivating examples involves prior knowledge of the shape of the objective function as a function of one argument, holding all other arguments fixed. We make this notion more precise. Let $f(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}$ be an objective function we wish to minimize over a convex set $\mathcal{A} \subset \mathbb{R}^d$. For the d-vector $\boldsymbol{x}$, let $\boldsymbol{x}_{-j}$ be the $(d-1)$-vector omitting the j^{th} element of $\boldsymbol{x}$. Then $f(\boldsymbol{x}) = f(x_j, \boldsymbol{x}_{-j})$ and the function $g_{j,\boldsymbol{a}}(x_j) = f(x_j, \boldsymbol{a})$ is the component function that arises from fixing the vector $\boldsymbol{x}_{-j} = \boldsymbol{a}$ and letting x_j vary. The next subsection describes how to implement a variety of shape constraints on component functions.

2.1 Shape constrained Gaussian processes

We build most directly on the approach of Wang and Berger [18]. The crucial fact used in that work is that, assuming a sufficiently smooth covariance function, a GP and its partial derivatives form a joint GP. Suppose $f(\boldsymbol{x})$ is a GP with covariance function $k(\boldsymbol{x}, \boldsymbol{x'})$ for $\boldsymbol{x}, \boldsymbol{x'} \in \mathcal{A}$. Then

$$\text{Cov}\left(\frac{\partial^o}{\partial x_j^o} f(\boldsymbol{x}), \frac{\partial^{o'}}{\partial x_k'^{o'}} f(\boldsymbol{x'})\right) = \frac{\partial^{o+o'}}{\partial x_j^o x_k'^{o'}} k(\boldsymbol{x}, \boldsymbol{x'}) \quad (1)$$

with $j, k \in \{1, ..., d\}$ and o, o' being the orders of the partial derivatives. We can impose componentwise shape constraints by restricting the partial derivative processes. For example, if we have prior information that $g_{j,\boldsymbol{a}}(x_j)$ is monotonically increasing (decreasing) for all $\boldsymbol{a}$, we can restrict its partial derivative with respect to x_j to be nonnegative (nonpositive). Similarly, if we want $g_{j,\boldsymbol{a}}(x_j)$ to be componentwise convex (concave), we can restrict second partial derivatives to be nonnegative (nonpositive).

An additional and novel constraint which we apply in our hyperparameter tuning application is quasiconvexity (also referred to as unimodality) [3]. We only consider continuous component functions of a single variable, for which it is easy to characterize quasiconvexity. The continuous function $h : \mathbb{R} \to \mathbb{R}$ is quasiconvex if and only if (1) h is monotone or (2) there exists some point c such that h is nonincreasing for $t < c$ and h is nondecreasing for $t \geq c$. Again, this is a condition which can be stated in terms of partial derivatives and is therefore amenable to a GP implementation.

As discussed in Wang and Berger [18], the partial derivative constraints are only enforced at a discrete set of points. Imposing requirements on the partial derivative at a set having a limit point would eliminate all sample paths of the GP. Realizations from a shape constrained GP only approximately follow the constraint; however, the discrepancy can be made practically irrelevant by enforcing the constraint at sufficiently many points.

The cases of monotonicity, convexity, or concavity constraints call for sampling high-dimensional truncated Gaussian random variables. We use the methodology introduced in Botev [1], which is available in the R package `TruncatedNormal` [2]. For the quasiconvexity case, rejection sampling was efficient enough for the applications presented.

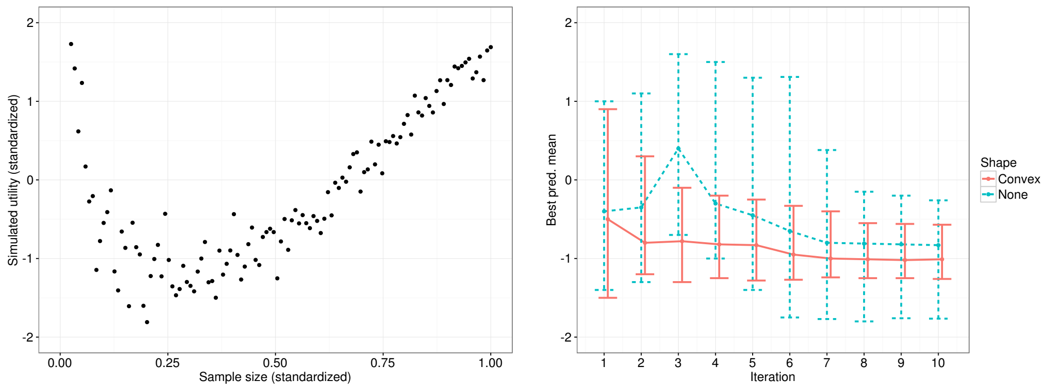
3 Experiments

In any Bayesian optimization application, one must make choices about the mean and covariance of the GP, the acquisition function, the priors, etc. Before getting to the experiments performed, we fix notation so we can give these details. Suppose the data are n observations $y_i = f(\boldsymbol{x}_i) + \epsilon_i$ with $i \in \{1, 2, ..., n\}$ and $\boldsymbol{x}_i \in \mathbb{R}^d$. We assume the errors ϵ_i are independent so that jointly $\boldsymbol{y} \sim N_n(\mathbf{1}_n\mu, \boldsymbol{K} + \sigma^2\boldsymbol{I}_n)$, where the (i, j)-th entry of $\boldsymbol{K}$ is $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. We use a squared-exponential kernel so that $k(\boldsymbol{x}, \boldsymbol{x}') = \tau^2 \exp\{-\sum_{i=1}^d \psi_d(x_i - x_i')^2\}$ and place independent Cauchy and half-Cauchy priors on the parameters: $\mu \sim C(0, 1)$, $\tau^2 \sim C_+(0, 0.25)$, $\sigma^2 \sim C_+(0, 0.25)$, and $\psi_d \sim C_+(0, 5)$. For ease of computation, we take a *partial likelihood* approach (which amounts to ignoring shape constraints when estimating the parameters, see [18]) and we don't perform fully Bayesian inference; instead, we plug-in the posterior median found using the `R` package `Rstan` [15]. Finally, the acquisition function we use is expected improvement. We acknowledge that we could likely achieve better results by using a Matérn kernel and performing fully Bayesian inference (as recommended in [14]), and by working with an acquisition function that uses the minimum expected first derivative. The merits of both will be explored elsewhere. In our applications, shape constraints are enforced at the observed values of $\boldsymbol{x}$ and 100 other locations that are selected by a maximin latin hypercube design using the R package `lhs` [4]. In all the cases considered here, the joint distribution of the observed data and derivatives is multivariate Gaussian with covariance matrices that can be constructed using Equation (1).

3.1 Binomial sample size

This experiment is Example 1 in Müller and Parmigiani [11]. The goal is selecting the sample size of a binomial experiment where the prior on the probability of success θ is an equal-weighted mixture of a Beta(3,1) and a Beta(3,3). The loss of an experiment with sample size n given data y and θ is

$L(n, y, \theta) = |\theta - m_y| + 0.0008n$ where m_y is the posterior median. For any given n, the loss is estimated as the Monte Carlo average of 100 simulations from the joint distribution of y and θ. The shape-constrained GP outperforms the unconstrained GP on average and appears to be more robust to the choice of starting points (see Figure 1). In contrast to many Bayesian optimization papers, we plot smallest expected loss rather smallest function evaluation because the target is noisy.

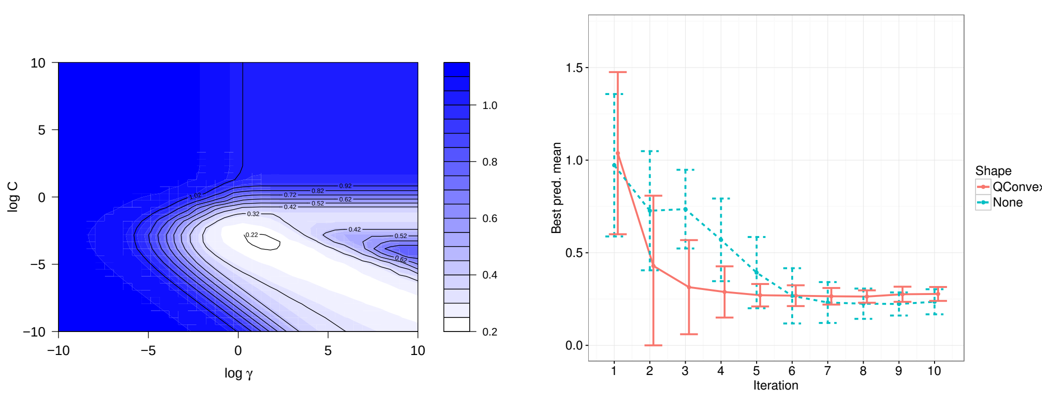


(a) (b)

Figure 1: (a) simulated losses for sample sizes ranging from 1 to 120 (axes are standardized). (b) minimum posterior expected loss among the observed data, 50 random starting values.

3.2 Support vector machine

The task is training an SVM with a squared-exponential kernel on the Ozone dataset (available in the R package `mlbench` [9]). We predict ozone readings using 12 other variables. We restrict $C, \gamma \in \{\exp(x) : x \in [-10, 10]\}$ and compare the performance of an unconstrained GP with that of a GP with quasiconvex constraints on both parameters. The performance metric is 10-fold cross-validated (CV) error. In this case, we only consider one starting point and compare the uncertainty in the minimum expected CV error in the observed samples. Figure 2 shows that the shape constrained GP stabilizes more quickly to a solution with lower uncertainty than the unconstrained version.



(a) (b)

Figure 2: (a) “true” error surface. (b) 95% CI posterior expected CV error.

References

[1] Z. I. Botev. The normal law under linear restrictions: simulation and estimation via minimax tilting. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2016.

[2] Zdravko I. Botev. *TruncatedNormal: Truncated Multivariate Normal*, 2015. URL `http://CRAN.R-project.org/package=TruncatedNormal`. R package version 1.0.

[3] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[4] Rob Carnell. *lhs: Latin Hypercube Samples*, 2016. URL `http://CRAN.R-project.org/package=lhs`. R package version 0.13.

[5] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, September 1995.

[6] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. Book in preparation for MIT Press, 2016. URL `http://www.deeplearningbook.org`.

[7] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning, Data Mining, Interference, and Prediction*. Springer, 2009.

[8] A. Klein, S. Bartels, S. Falkner, P. Hennig, and F. Hutter. Towards efficient bayesian optimization for big data. In *NIPS 2015 Bayesian Optimization Workshop*, December 2015.

[9] Friedrich Leisch and Evgenia Dimitriadou. *mlbench: Machine Learning Benchmark Problems*, 2010. R package version 2.1-1.

[10] Lizhen Lin and David B. Dunson. Bayesian monotone regression using gaussian process projection. *Biometrika*, 2014.

[11] Peter Müller and Giovanni Parmigiani. Optimal design via curve fitting of monte carlo experiments. *Journal of the American Statistical Association*, 90(432):1322–1330, 1995.

[12] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[13] Jaakko Riihimäki and Aki Vehtari. Gaussian processes with monotonicity information. *Journal of Machine Learning Research*, (AISTATS2010 special issue):645–652, 2010. VK: Workshop and Conference Proceedings 9.

[14] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2951–2959. Curran Associates, Inc., 2012.

[15] Stan Development Team. Stan: A c++ library for probability and sampling, version 2.10.0, 2015. URL `http://mc-stan.org/`.

[16] Kevin Swersky, Jasper Snoek, and Ryan Prescott Adams. Freeze-thaw bayesian optimization. *CoRR*, abs/1406.3896, 2014. URL `http://arxiv.org/abs/1406.3896`.

[17] Amos Tversky and Daniel Kahneman. Judgment under uncertainty: Heuristics and biases. In *Utility, probability, and human decision making*, pages 141–162. Springer, 1975.

[18] Xiaojing Wang and James O. Berger. Estimating shape constrained functions using gaussian processes. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):1–25, 2016.