Input Warping for Bayesian Optimization of Non-stationary Functions

Jasper Snoek
Harvard University
jsnoek@seas.harvard.edu

Kevin Swersky
University of Toronto
kswersky@cs.toronto.edu

Richard S. Zemel
University of Toronto
zemel@cs.toronto.edu

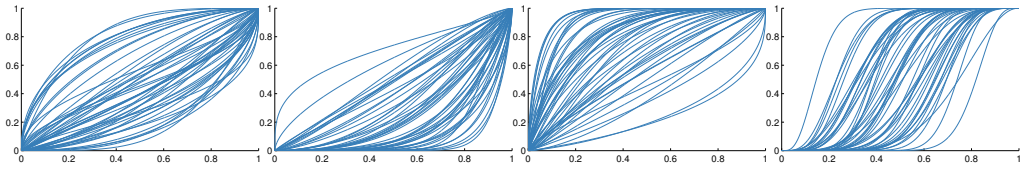
Ryan P. Adams
Harvard University
rpa@seas.harvard.edu

Abstract

Bayesian optimization has proven to be a highly effective methodology for the global optimization of unknown, expensive and multimodal functions. The ability to accurately model distributions over functions is critical to the effectiveness of Bayesian optimization. Although Gaussian processes provide a flexible prior over functions which can be queried efficiently, there are various classes of functions that remain difficult to model. One of the most frequently occurring of these is the class of non-stationary functions. The optimization of the hyperparameters of machine learning algorithms is a problem domain in which parameters are often manually transformed a-priori, for example by optimizing in "log-space", to mitigate the effects of extreme non-stationarity. We develop a methodology for automatically learning monotonic, bijective transformations or *warpings* of the input space using the beta cumulative distribution function. Marginalizing over the parameters of the of the beta distribution allows the Bayesian optimization to integrate over a wide class of flexible warping functions. We demonstrate on four challenging machine learning problems that the optimization converges to significantly better solutions much faster when this input warping is used.

1 Introduction

Bayesian optimization has proven to be a highly effective strategy for the global optimization of noisy, black-box functions. The methodology relies on fitting a relatively cheap surrogate function approximating an expensive function of interest, on which a proxy optimization is performed to select the next expensive evaluation. Naturally, the ability of the surrogate to accurately model the underlying function is crucial to the success of the optimization routine. Recent work in machine learning has revisited the idea of Bayesian optimization [Brochu et al., 2010, Srinivas et al., 2010, Hutter et al., 2011, Osborne et al., 2009, Bergstra et al., 2011, Bull, 2011, Snoek et al., 2012] in large part due to advances in the ability to efficiently and accurately model statistical distributions over large classes of real-world functions. Recent advances in Gaussian processes [Rasmussen and Williams, 2006] have provided a powerful framework to express flexible prior distributions over smooth functions yielding accurate estimates of the expected value of the function at any given input, but crucially also uncertainty estimates over that value. These are the two main components that enable the exploration and exploitation tradeoff which makes Bayesian optimization so powerful. A major limitation of the most commonly used form of Gaussian process regression is the assumption of stationarity — that the covariance between two outputs is invariant to translations in input space. This simplifies the regression task, but hurts the ability of the Gaussian process to model non-stationary functions. This presents a problem for Bayesian optimization, as many problems of interest are inherently non-stationary.



(a) Linear (b) Exponential (c) Logarithmic (d) Sigmoidal

Figure 1: Each figure shows 50 warping functions resulting from the beta CDF where the shape parameters α and β are sampled from a lognormal prior with a different mean and variance. The flexible beta CDF captures many desirable warping functions and adjusting the prior over input warpings allows one to easily encode prior beliefs over the form of non-stationarity.

We introduce a simple solution that allows Gaussian processes to model a large variety of non-stationary functions that is particularly well suited to Bayesian optimization. We automatically learn a monotonic, bijective *warping* of the inputs that removes major non-stationary effects. This is achieved by projecting each dimension of the input through the cumulative distribution function of the beta distribution, while marginalizing over the distribution's shape parameters. This approach has several advantages over existing approaches. It is computationally efficient, captures a variety of desirable transformations, such as logarithmic, exponential, sigmoidal, etc., and it is easily interpretable. In the context of Bayesian optimization, understanding the parameter space is often just as important as achieving the best possible result. Our approach lends itself to an easy analysis of the inherent non-stationarities in a given problem domain. In our empirical analysis, we observe that on four different challenging machine learning optimization tasks our method significantly outperforms that of Snoek et al. [2012], consistently converging to a better result in significantly fewer function evaluations.

2 Background

2.1 Gaussian Processes

The Gaussian process (GP) is a tractable and flexible prior distribution over functions $f : \mathcal{X} \to \mathbb{R}$ and is a widely used model for non-linear Bayesian regression. The properties of the Gaussian process are specified by a mean function $m : \mathcal{X} \to \mathbb{R}$ and a positive definite covariance, or kernel, function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Given a finite set of training points $\{\mathbf{x}_n, y_n\}_{n=1}^N$, where $\mathbf{x}_n \in \mathcal{X},\ y_n \in \mathbb{R}$, the predictive mean and covariance under a GP can be respectively expressed as:

$$\mu(\mathbf{x}\,;\,\{\mathbf{x}_n, y_n\}, \theta) = K(\mathbf{X}, \mathbf{x})^\top K(\mathbf{X}, \mathbf{X})^{-1}(\mathbf{y} - m(\mathbf{X})), \tag{1}$$

$$\Sigma(\mathbf{x}, \mathbf{x}'\,;\,\{\mathbf{x}_n, y_n\}, \theta) = K(\mathbf{x}, \mathbf{x}') - K(\mathbf{X}, \mathbf{x})^\top K(\mathbf{X}, \mathbf{X})^{-1} K(\mathbf{X}, \mathbf{x}'). \tag{2}$$

Here $K(\mathbf{X}, \mathbf{x})$ is the N-dimensional column vector of cross-covariances between $\mathbf{x}$ and the set $\mathbf{X}$. The $N \times N$ matrix $K(\mathbf{X}, \mathbf{X})$ is the Gram matrix for the set $\mathbf{X}$.

2.2 Non-stationary Gaussian Process Regression

Various related approaches have been proposed to extend GPs to model non-stationary functions. Gramacy [2005] proposed a Bayesian treed GP model which accommodates various complex non-stationarities through modeling the data using multiple GPs with different covariances. Various non-stationary covariance functions have been proposed [Higdon et al., 1998, Rasmussen and Williams, 2006]. Previously, Sampson and Guttorp [1992] proposed projecting the inputs into a stationary latent space using a combination of metric multidimensional scaling and thin plate splines. Schmidt and O'Hagan [2003] extended this warping approach for general GP regression problems using a flexible GP mapping. Bornn et al. [2012] project the inputs into a higher dimensional stationary latent representation. In a complementary approach, Snelson et al. [2003] apply a warping to the output space, $\mathbf{y}$.

Compared to these approaches, the advantages of the proposed approach are that it is conceptually simple, computationally efficient, captures a wide variety of non-stationarities, can be used within the context of many common covariance functions operating on bounded continuous sets without changing the analytic properties of the GP and easily lends itself to post-hoc analysis.

2.3 Bayesian Optimization

Bayesian optimization is a general framework for the global optimization of noisy, expensive, black-box functions [Mockus et al., 1978] (see Brochu et al. [2010] for an in-depth explanation). The strategy relies on the use of a relatively cheap probabilistic model which can be queried liberally as a surrogate in order to more efficiently evaluate an expensive function of interest. Bayes' rule is used to derive the posterior estimate of the true function given observations, and the surrogate is then used to determine, via a proxy optimization over an *acquisition function*, the next most promising point to query. We follow the common approach, which is to use a GP to define a distribution over objective functions from the input space to a loss that one wishes to minimize. Our approach is based on that of Snoek et al. [2012]. Specifically, we use a GP surrogate, and the expected improvement acquisition function [Mockus et al., 1978, Jones, 2001]. Note that our methods are independent of the acquisition function used and do not affect its analytic properties.

3 Input Warping

We assume that we have a positive definite covariance function $K(\mathbf{x}, \tilde{\mathbf{x}})$ where $\mathbf{x}, \tilde{\mathbf{x}} \in [0, 1]^D$ due to projecting the bounded input range to the unit hypercube. In practice, when tuning the hyper-parameters of an algorithm, e.g., the regularization parameter of a support vector machine using grid search, researchers often first transform the input space using a monotonic function such as the natural logarithm. Such an optimization in "log-space" takes advantage of a-priori knowledge of the non-stationarity that is inherent in the input space. Often however, the non-stationary properties of the input space are not known a-priori and such a transformation is generally a crude approximation to the ideal (unknown) transformation. Our approach is to instead consider a variety of bijective, monotonic warping functions that will be estimated, and from which functions such as the log transform can be specified as a prior. Specifically, we change the kernel function to be $k(w(\mathbf{x}), w(\tilde{\mathbf{x}}))$ where $w : \mathcal{X} \rightarrow [0, 1]^D$ is the cumulative distribution function of the beta distribution with distinct shape parameters $\alpha_d > 0$ and $\beta_d > 0$ for each dimension $d \in \{1, 2, ..., D\}$ of the input.

3.1 Integrating over warpings

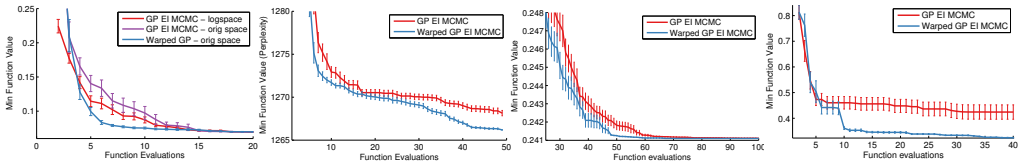
Rather than assume an explicit transformation function, we instead integrate over warpings by marginalizing over the shape parameters, α and β, of the beta distribution with an appropriate prior. We treat α and β as hyperparameters of the covariance function and integrate them out using Markov chain Monte Carlo via slice sampling following the treatment of covariance hyperparameters from Snoek et al. [2012]. We use a log-normal distribution, i.e. $\log(\alpha) \sim \mathcal{N}(\mu_\alpha, \sigma_\alpha)$ and $\log(\beta) \sim \mathcal{N}(\mu_\beta, \sigma_\beta)$, to express a prior for a wide family of desirable functions. Figure 1 demonstrates example warping functions arising from sampling beta parameters from various instantiations of the prior. Note that the geometric mean or median of the zero-mean log-normal distribution for α and β corresponds to the identity transform. With this prior the model assumes no transformation of the input space without evidence to the contrary. In the following empirical analysis we use this formulation with a variance of 0.75, assuming no prior knowledge of the form of the transformation for a particular problem. However, a nice property of this formulation is that a user can easily specify a prior for a specific form of warping, as we show in Figure 1.

4 Empirical Analyses

As an empirical analysis we evaluate the standard Gaussian process expected improvement algorithm (GP EI MCMC) as implemented by Snoek et al. [2012], with and without warping. As in Snoek et al. [2012] we use the Matérn $5/2$ kernel and we marginalize over kernel parameters θ using slice sampling [Murray and Adams, 2010]. We repeat three of the experiments[1] from Snoek et al. [2012], and an experiment involving the tuning of a deep convolutional neural network[2] on a subset of the popular CIFAR-10 data set [Krizhevsky, 2009]. The deep network consists of 3 convolutional layers and 2 fully connected layers and we optimize over two learning rates, one for each layer type, 6 dropout regularization rates, 6 weight norm constraints, the number of hidden units per layer, a convolutional kernel size and a pooling size for a total of 21 hyperparameters. On the logistic regression problem we also compare to warping the input space a-priori using the log-transform (optimizing in log-space). Figure 2 shows that in all cases, dealing with non-stationary effects via input

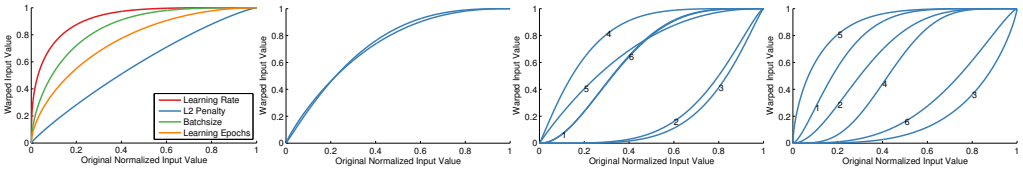
[1] See Snoek et al. [2012] for details of these experiments.

[2] We use the Deepnet package from `https://github.com/nitishsrivastava/deepnet`



(a) Logistic Regression (b) Online LDA (c) Structured SVM (d) Cifar 10 Subset

Figure 2: An empirical comparison of Bayesian optimization following the standard Gaussian process expected improvement algorithm (GP EI MCMC) and our strategy (Warped GP EI MCMC) for modeling input non-stationarity.



(a) Logistic Regression (b) DNN Learning Rates (c) DNN Weight Norms (d) DNN Dropout

Figure 3: Example input warpings learned for the logistic regression problem (Figure 3a) and the parameters of the deep convolutional neural network (Figures 3b, 3d, 3c). Each plot shows the mean warping, averaged over 100 samples, of each of the parameters. Each curve in Figures 3c and 3d is annotated with the depth of the layer that each parameter is applied to.

warpings significantly improves the convergence of the optimization. In particular, we notice on the higher-dimensional convolutional network problem a *profound* improvement (Figure 2d) when the non-stationarities are learned.

In Figure 3 we plot examples of some of the inferred warpings. For logistic regression, Figure 3a shows that model learns different logarithmic-like warpings for three dimensions and no warping for the fourth. Figure 3b shows that on both convolutional and dense layers, the intuition that one should log-transform the learning rates holds. For transformations on weight norm constraints, shown in Figure 3c, the weights connected to the inputs and outputs use a sigmoidal transformation, the convolutional-layer weights use an exponential transformation, and the dense-layer weights use a logarithmic transformation. Effectively, this means that the most variation in the error occurs in the medium, high and low scales respectively for these types of weights. Especially interesting are the wide variety of transformations that are learned for dropout on different layers, shown in Figure 3d. These show that different layers benefit from different dropout rates, which was also confirmed on test set error, and challenges the notion that they should just be set to 0.5 [Hinton et al., 2012].

It is clear that the learned warpings are non-trivial. In some cases, like with learning rates, they agree with intuition, while for others like dropout they yield surprising results. Given the number of hyperparameters and the variety of transformations, it is highly unlikely that even experts would be able to determine the whole set of appropriate warpings. This highlights the utility of learning them automatically.

5 Conclusion

In this paper we develop a novel formulation to elegantly model non-stationary functions using Gaussian processes that is especially well suited to Bayesian optimization. Our approach uses the cumulative distribution function of the beta distribution to warp the input space in order to remove non-stationary effects. This approach allows us to automatically infer a wide variety of warpings in a computationally efficient way. In our empirical analysis we see that an inability to model non-stationary functions is a major weakness in the GP Bayesian optimization framework. Our simple approach to learn the form of the non-stationarity, with no additional prior information, significantly outperforms the standard Bayesian optimization routine of Snoek et al. [2012] both in the number of evaluations it takes to converge and the value reached. A key advantage of our approach is that the learned transformations can be analyzed post-hoc, and our analysis of a convolutional neural network architecture leads to surprising insights that challenges established doctrine. An interesting follow-up would be to determine whether consistent patterns emerge across architectures, datasets and domains.

References

Eric Brochu, Tyson Brochu, and Nando de Freitas. A Bayesian interactive optimization approach to procedural animation design. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2010.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: no regret and experimental design. In *International Conference on Machine Learning*, 2010.

Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization 5*, 2011.

Michael A. Osborne, Roman Garnett, and Stephen J. Roberts. Gaussian Processes for Global Optimization. In *Learning and Intelligent Optimization*, 2009.

James S. Bergstra, Rémi Bardenet, Yoshua Bengio, and Bálázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*. 2011.

Adam D. Bull. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, (3-4):2879–2904, 2011.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, 2012.

Carl E. Rasmussen and Christopher Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Robert B. Gramacy. *Bayesian treed Gaussian process models*. PhD thesis, UC Santa Cruz, 2005.

D. Higdon, J. Swall, and J. Kern. Non-stationary spatial modeling. *Bayesian Statistics*, 6, 1998.

Paul D. Sampson and Peter Guttorp. Nonparametric estimation of nonstationary spatial covariance structure. *Journal of the American Statistical Association*, 87(417):108–119, 1992.

Alexandra M. Schmidt and Anthony O'Hagan. Bayesian inference for nonstationary spatial covariance structures via spatial deformations. *Journal of the Royal Statistical Society Series B*, 65:743–758, 2003.

Luke Bornn, Gavin Shaddick, and James V. Zidek. Modeling nonstationary processes through dimension expansion. *Journal of the American Statistical Society*, 107(497), 2012.

Edward Snelson, Carl Edward Rasmussen, and Zoubin Ghahramani. Warped Gaussian processes. In *Advances in Neural Information Processing Systems*, 2003.

Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2, 1978.

Donald R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21, 2001.

Iain Murray and Ryan P. Adams. Slice sampling covariance hyperparameters of latent Gaussian models. In *Advances in Neural Information Processing Systems*. 2010.

Alex Krizhevsky. Learning multiple layers of features from tiny images. *Technical report, Department of Computer Science, University of Toronto*, 2009.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.