Practical Bayesian Optimization of Machine Learning Algorithms Supplementary Materials

Jasper Snoek
Department of Computer Science
University of Toronto
jasper@cs.toronto.edu

Hugo Larochelle
Department of Computer Science
University of Sherbrooke
hugo.larochelle@usherbrooke.edu

Ryan P. Adams
School of Engineering and Applied Sciences
Harvard University
rpa@seas.harvard.edu

1 Appendix

In this section we specify additional details of our Bayesian optimization algorithm which, for brevity, were omitted from the paper. For more detail, the code used in this work is made publicly available at `http://www.cs.toronto.edu/~jasper/software.html`. The code is designed to be flexible and accessible such that other researchers can use it to optimize the hyperparameters of any desired machine learning model.

1.1 Optimizing the Acquisition Function

We start by projecting the observations to the unit hypercube, as defined by bounds of the optimization. Gaussian process hyperparameters, θ, are sampled using the slice sampling algorithm of Murray and Adams [1]. In order to find the maximum of the multimodal acquisition function $a(\mathbf{x}\,;\,\{\mathbf{x}_n, y_n\}, \theta)$ in a continuous domain, first discrete candidate points are densely sampled in the unit hypercube using a low discrepancy Sobol sequence [2]. Each of these candidates is then subjected to a bounded optimization over the integrated acquisition function. Precisely, the minimum of the acquisition function, averaged over GP hyperparameter samples, is computed with the input initialized at each of the candidate points. As the acquisition functions in this work can all be expressed analytically in closed form, standard gradient descent techniques can be used. This yields a new set of candidate points, each of which is located at a local optimum of the integrated acquisition function. The next point to be evaluated in the Bayesian optimization procedure is then selected as the candidate point with the highest integrated acquisition function. Algorithm 1 outlines the procedure for selecting the next candidate point to evaluate while integrating over hyperparameter samples. In the event of pending experiments, $\{\mathbf{x}_j\}_{j=1}^{J}$, fantasized corresponding outcomes, $\{\mathbf{y}_j\}_{j=1}^{J}$, can be efficiently sampled from the Gaussian process posterior for each hyperparameter sample and added to the observation set, before computing the integrated acquisition function.

1.2 Hyperparameter Priors

After choosing the form of the Gaussian process covariance, we must also manage the hyperparameters that govern its behavior. In our empirical evaluation, unless otherwise specified, we have $D + 3$ Gaussian process hyperparameters: D length scales $\theta_{1:D}$, the covariance amplitude θ_0, the observation noise ν, and a constant mean m. For a fully-Bayesian treatment of hyperparameters, it is desirable to marginalize over hyperparameters and compute the *integrated acquisition function*. As stated in the paper, a Monte Carlo estimate of the integrated acquisition function is computed via slice sampling [1]. Appropriate priors for each of the hyperparameters are chosen for use within the context of the slice sampling algorithm, which we will clarify here. We specify a uniform prior for

the mean, m, and width 2 top-hat priors for each of the D length scale parameters. As we expect the observation noise generally to be close to or exactly zero, ν is given a *horseshoe* prior [3]. The covariance amplitude θ_0 is given a zero mean, unit variance lognormal prior, $\theta_0 \sim \ln \mathcal{N}(0, 1)$.

Algorithm 1 Selecting the next point to evaluate

Input: Observations $\{\mathbf{x}_n, y_n\}_{n=1}^N$
{Generate a set of M candidate points from the Sobol sequence [2]}
$\mathbf{X}_{\mathsf{cand}} = Sobol(M)$
{Generate H Gaussian process hyperparameter samples [1]}
for $h = 1$ **to** H **do**
  Sample θ_h {See Section 1.2}
end for
for $\mathbf{x}_m$ **in** $\mathbf{X}_{\mathsf{cand}}$ **do**
  $\mathbf{x}_m = \max_{\mathbf{x}} \sum_{h=1}^H a(\mathbf{x}_m\,;\, \{\mathbf{x}_n, y_n\}, \theta_h)$
end for
$\mathbf{x}_{\mathsf{next}} = \operatorname{argmax}_{\mathbf{x}} \sum_{h=1}^H a(\mathbf{X}_{\mathsf{cand}}\,;\, \{\mathbf{x}_n, y_n\}, \theta_h)$
return $\mathbf{x}_{\mathsf{next}}$

References

[1] Iain Murray and Ryan P. Adams. Slice sampling covariance hyperparameters of latent Gaussian models. In *Advances in Neural Information Processing Systems 24*, pages 1723–1731. 2010.

[2] Paul Bratley and Bennett Fox. Algorithm 659: Implementing sobol's quasirandom sequence generator. *ACM Transactions on Mathematical Software*, 14:88–100, 1988.

[3] Carlos M. Carvalho, Nicholas G. Polson, and James G. Scott. Handling sparsity via the horseshoe. In *AISTATS*, 2009.