Statistical Learning for Decision

Matthew Speers

January 20, 2022

Abstract

This report aims to motivate the use of Bayesian optimisation for finding global optima. An overview of the methodology is given, as well as a discussion of what different adaptions to this methodology can provide. Finally, areas of further development in the field are discussed.

1 Introduction

Many optimisation problems require the global maximisation (minimisation) of expensive to evaluate objective functions. Bayesian optimisation algorithms cope with this high computational cost by using knowledge of all past evaluations of the objective function to intelligently select the next evaluation point. This is done by constructing a model for the objective function dependent on this past data, usually in the form of a Gaussian process (discussed in Sections 2.2 and 2.3). Bayesian optimisation algorithms have been applied to a broad spectrum of industries, from pharmaceuticals (Sano et al., 2020) to aerospace (Lam et al., 2018).

2 Bayesian Optimisation

2.1 The Problem

Consider a real-valued objective function $f : \mathcal{X} \to \mathbb{R}$ for $\mathcal{X} \subset \mathbb{R}^D$. Often we wish to solve a problem of the form;

$$x^* = \underset{x \in \mathcal{X}}{\operatorname{argmin}} f(x), \tag{1}$$

i.e. we wish to find the global minimum of f. We assume f is a black-box function with no closed form and that we can query it any arbitrary point $x \in \mathcal{X}$. It is common to assume this query gives an exact evaluation of the objective function, however, this is not realistic in many contexts. For example, due to imprecise instruments or statistical approximation.

As such, we consider an evaluation y at a point $x \in \mathcal{X}$ to be a noisy observation of the true value of the objective function, with $\mathbb{E}(y) = f(x)$. Mathematically, we have;

$$y = f(x) + \epsilon, \tag{2}$$

where ϵ is some error term. There are many such models with the form of Equation 2, perhaps the most simple being described by Garnett (2022);

$$y|x, f(x), \sigma^2 \sim N(f(x), \sigma^2), \tag{3}$$

where σ^2 is referred to as the 'observation noise scale'. This is an important example of an observation model as its properties allow for nice Bayesian inference on the distribution of f, as is explained in 2.3. However, the choice of observation model does not impact the general Bayesian optimisation framework and so we assume a generic model unless stated otherwise.

The objective of Bayesian optimisation is to devise a scheme that selects a new query point x_{n+1} given n previous query-evaluation tuples $\mathcal{D}_n = \{(x_i, y_i)\}_{i=1}^n$. To do this, a model must be formed that reflects our uncertainty regarding the objective function f. This is discussed in 2.2. The choice of the next query point is then determined by maximising a function α_n, known as the acquisition function. This is discussed in 2.4. A summary of this process is given by Shahriari et al. (2015) in Algorithm 1.

Algorithm 1 Bayesian Optimisation

1: **for** $n = 1, 2, \ldots$ **do**
2: select new x_{n+1} by optimising acquisition function α

$$x_{n+1} = \underset{x \in \mathcal{X}}{\operatorname{argmax}}\, \alpha_n$$

3: query objective function to obtain y_{n+1}
4: augment data $\mathcal{D}_{n+1} = \{\mathcal{D}_n, (x_{n+1}, y_{n+1})\}$
5: update statistical model
6: **end for**

2.2 Gaussian Process Model

We need to select a model for our uncertainty regarding the objective function f. A parametric approach, whilst seemingly the most straightforward, is not suitable for many forms of objective function (Garnett, 2022). Instead, we take a nonparametric route and model f as a Gaussian process. We specify the process as;

$$f \sim \mathcal{GP}(\mu, K), \tag{4}$$

where $\mu : \mathcal{X} \to \mathbb{R}$ is the mean function and $K : \mathcal{X}^2 \to \mathbb{R}$ is the positive semi-definite covariance (kernel) function.

As an aside, it is worth noting that other models for f are available. For example, Shah et al. (2013) argue that a Student-t process is preferable as it offers more flexibility over the Gaussian option (e.g. the ability to learn heavy tailed function behaviour), whilst retaining many of the benefits (such as having a closed form for many acquisition functions - see Section 2.4).

The model described by (4) can be thought of loosely as an infinite-dimensional multivariate normal random variable in the sense that it exhibits much of the same behaviour. For example, for a finite subset $\mathbf{x} \in \mathcal{X}$, we can calculate a marginal distribution of $f(\mathbf{x})$ as;

$$f(\mathbf{x}) \sim MVN(\mu(\mathbf{x}), \mathbf{K}), \tag{5}$$

where;

$$\mathbf{K} = \begin{pmatrix} K(x_1, x_1) & K(x_1, x_2) & \cdots & K(x_1, x_n) \\ K(x_2, x_1) & K(x_2, x_2) & \cdots & K(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ K(x_n, x_1) & K(x_n, x_2) & \cdots & K(x_n, x_n) \end{pmatrix},$$

i.e. the matrix formed by evaluating K at each pair of n points in $\mathbf{x}$. For a single $x \in \mathcal{X}$, Equation 5 reduces to $f(x) \sim N(\mu(x), \sigma^2(x))$. The prior choices of μ and K are clearly important in reflecting our belief about f. The function μ serves as a location parameter and shows the central tendency of the function. The function K provides information as to how the function is structured; for example, a kernel which falls off faster will result in a less smooth function than a kernel that maintains higher correlation between x values further apart.

This ability of the kernel to capture different behaviours allows us to model many forms of objective function and thus is another reason the Gaussian process model is desirable. Two examples of kernel functions, the automatic relevance determination (ARD) squared exponential kernel and the ARD Matérn 5/2 kernel, are described by Snoek et al. (2012). Formulations for these are given in Equations 6 and 7 respectively;

$$K_{\mathrm{SE}}(x, x') = \theta_0 \exp\left(-\frac{1}{2} r^2(x, x')\right) \qquad r^2(x, x') = \sum_{d=1}^{D} (x_d - x'_d)^2 / \theta_d^2, \tag{6}$$

$$K_{\mathrm{M52}}(x, x') = \theta_0 \left(1 + \sqrt{5r^2(x, x')} + \frac{5}{3} r^2(x, x')\right) \exp\left(-\sqrt{5r^2(x, x')}\right). \tag{7}$$

The K_{SE} kernel is a common choice for this problem. It does, however, result in unrealistically smooth sample functions for many applications (Snoek et al., 2012). Therefore, K_{M52} is often chosen as a more realistic alternative. Of course, these are just two illustrative examples amongst an infinite number of kernel choices.

The selection of the kernel is a nuanced problem, largely dependent on the specific problem being solved. For example, BOCK (Bayesian Optimisation with Cylindrical Kernels) has been developed for when it is desirable to prevent the search spending too much time near the boundaries of its search space (Oh et al., 2018). The process of kernel selection has become somewhat mystified, and attempts have been made to undo this. A kernel 'grammar' and method of searching over an infinite kernel space has been developed by Duvenaud et al. (2013). This has been built upon by the likes of Malkomes et al. (2016) who uses Bayesian optimisation to provide a completely automated kernel selection process.

2.3 Bayesian Model Updates

Next we discuss how to form our model for $f|\mathcal{D}_n$ by updating our prior Gaussian process model for f given observations $\mathcal{D}_n = (\mathbf{x}, \mathbf{y})$. Assume our observations come from the distribution described by Equation 3, so $\mathbf{y} = f(\mathbf{x}) + \boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon} \sim MVN(\mathbf{0}, \sigma^2 I_n)$.

Crucially, our observations $\mathbf{y}$ share a joint distribution with f, meaning we can write;

$$f, \mathbf{y} \sim \mathcal{GP}\left(\begin{bmatrix} f \\ \mathbf{y} \end{bmatrix}; \begin{bmatrix} \mu \\ \mu(\mathbf{x}) \end{bmatrix}, \begin{bmatrix} K & \kappa^\top \\ \kappa & \mathbf{K} + \sigma^2 I_n \end{bmatrix}\right), \tag{8}$$

where $\kappa(x)$ is the cross-covariance function between f and $\mathbf{y}$ so that $\kappa^{(i)}(x) = \mathrm{cov}\left(f(x), y_i\right) = k(x, x_i)$. This assumption holds for any affine transformation of function values, as well as any limits of such quantities (Garnett, 2022). Conditioning on $\mathcal{D}_n$ gives the posterior μ_n and K_n as;

$$\mu_n(x) = \mu(x) + \kappa(x)^\top (\mathbf{K} + \sigma^2 I_n)^{-1} (\mathbf{y} - \mathbf{m}), \tag{9}$$

$$K_n(x, x') = K(x, x') - \kappa(x)^\top (\mathbf{K} + \sigma^2 I_n)^{-1} \kappa(x'), \tag{10}$$

via Bayes' rule. Our posterior model is thus $f|\mathcal{D}_n \sim \mathcal{GP}(\mu_n, K_n)$, or, for finite $\mathbf{x} \in \mathcal{X}$;

$$f(\mathbf{x})|\mathcal{D}_n \sim MVN(\mu_n(\mathbf{x}), \mathbf{K}_n), \tag{11}$$

where $\mathbf{K}_n$ is defined similarly to $\mathbf{K}$ in Equation 4. We refer to this conditional model throughout Section 2.4.

2.4 Acquisition Functions

As described in Section 2.1, we require an acquisition function α_n in order to select the next query point x_{n+1} at each stage of Algorithm 1. This function takes the form;

$$\alpha_n(x) = \mathbb{E}\left(u(x)|\mathcal{D}_n\right), \tag{12}$$

where $u(x)$ is the utility function, a measure of goodness for a point $x \in \mathcal{X}$. Different utility functions ask different things of the search process and so result in a different choice for x_{n+1}. Consequently, the selection of this function requires some consideration.

Perhaps the simplest choice for utility function is $u(x) = I_{f(x)<f^*}$, where $f^* = f(x^*)$ is the current optimal value. Here, Equation 12 becomes;

$$\alpha_n(x) = \mathbb{E}\left(I_{f(x)<f^*}|\mathcal{D}_n\right), \tag{13}$$

being equal to the probability that a query point $x \in \mathcal{X}$ will result in an evaluation smaller than the current global minimum. In other words, Equation 13 is equivalent to $\mathbb{P}\left(f(x) < f^*\right)$, i.e. the *probability of improvement*. Under the Gaussian process model described in (4), we can find a closed-form expression for $\alpha_n(x)$ to be;

$$\alpha_n(x) = \Phi\left(\frac{f^* - \mu_n(x)}{\sigma_n(x)}\right), \tag{14}$$

where Φ is the standard normal cdf. Here, the subscripts in the terms $\mu_n(x)$ and $\sigma_n(x)$ distinguish these terms as the posterior values discussed in Section 2.3, different from the $\mu(x)$ and $\sigma(x)$ assigned *a priori* as in Equation 4. Equation 14 can then be maximised as in step 2 of Algorithm 1 to locate the next query point x_{n+1}.

While this is a seemingly reasonable choice of utility function, the resulting acquisition function has its drawbacks. The function does not seek query points which promise a large potential improvement over the current optimal solution, so can waste time by selecting query points very near to previously queried points which will only give small improvements at best.

We can overcome this issue by instead using utility function $u(x) = \max(0, f^* - f(x))$. Equation 12 now becomes;

$$\alpha_n(x) = \mathbb{E}\left(\max(0, f^* - f(x))|\mathcal{D}_n\right), \tag{15}$$

i.e the expected size of improvement over the current optimum f^*. We thus call Equation 15 the *expected improvement* acquisition function. Again, under the model described by (4) we can find a closed-form expression for α_n. This has the form;

$$\alpha_n(x) = (f^* - \mu_n(x))\Phi\left(\frac{f^* - \mu_n(x)}{\sigma_n(x)}\right) + \sigma_n(x)\phi\left(\frac{f^* - \mu_n(x)}{\sigma_n(x)}\right), \tag{16}$$

where ϕ is the standard normal pdf. This has been shown to converge near-optimally (Bull, 2011), however, this convergence is only achieved with intelligent implementation. As stated by Diaconis

and Freedman (1986), whether or not Bayesian methods will find the correct solution is reliant on careful consideration of the problem. This acquisition function has the added benefit of innately balancing exploration and exploitation. The first term in Equation 16 increases for query points which provide a low mean, thus encouraging exploitation; the second term increases for points which provide a high variance, encouraging exploration.

The above method, however, does not allow for control over the balance of emphasis between exploration versus exploitation. An alternative acquisition function that does facilitate this is known as GP-UCB or UCB (Upper Confidence Bound), proposed by Srinivas et al. (2009). The acquisition function has form;

$$\alpha_n(x;\beta) = -\mu_n(x) + \beta\sigma_n(x), \tag{17}$$

where β is a trade-off parameter. An increase in β prioritises exploration, while a decrease favours exploitation. Notably, Equation 17 is not the expectation of a utility function as is in the case for other methods. The authors Srinivas et al. (2009) calculate regret bounds for this method and show it to perform similarly to EI.

Another method of searching, proposed by Hennig and Schuler (2012), is *entropy search*. Here, we consider the distribution on the optimal value x^* induced by our distribution for $f(x^*)$. The idea behind entropy search is to prioritise new query points that give a large reduction in entropy of this induced distribution. The utility function has the form;

$$u(x) = H(x^*|\mathcal{D}_n) - H(x^*|\mathcal{D}_n, x, f(x)), \tag{18}$$

i.e. the reduction in entropy of $x^*|\mathcal{D}_{n+1}$ over $x^*|\mathcal{D}_n$. Combining Equations 12 and 18, we obtain an acquisition function of;

$$\alpha_n(x) = H(x^*|\mathcal{D}_n) - \mathbb{E}\left[H(x^*|\mathcal{D}_n, x, f(x))\right], \tag{19}$$

where expectation is taken w.r.t. $f(x)|\mathcal{D}_n$. It is argued by Hennig and Schuler (2012) that this approach is more closely aligned with a practitioner's philosophy than alternative methods. They also demonstrate that, within a set number of iterations, it achieves the smallest distance between the best estimate and true global minimum of all the searches mentioned here. These benefits come with caveat that the approximations required to handle Equation 19 result in increased computational cost (a constant multiple of more traditional searches). It could therefore been argued that entropy search is only beneficial when it is particularly computationally expensive to evaluate f, in which case the added complexity is less relevant. As this is the case within many Bayesian optimisation settings, entropy search is often the most suitable method of those discussed here.

3 Conclusion

Bayesian optimisation offers a flexible framework for solving global optimisation problems and, with intelligent application, convergence to an optima is often provable. However, outstanding problems remain. The selection of hyperparameters for the Gaussian process model is an underdeveloped and often challenging endeavour, see Wang and de Freitas (2014). The scaling of Bayesian optimisation algorithms to higher dimensions has also proven difficult, although attempts have been made to expand high-dimensional methods to less restrictive settings (Kandasamy et al., 2015). Finally, the high computational cost of evaluating some objective functions is an issue, with efforts being made to reduce this including those suggested by Kandasamy et al. (2018).

References

Bull, A. D. (2011). Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12(10).

Diaconis, P. and Freedman, D. (1986). On the consistency of Bayes estimates. *The Annals of Statistics*, pages 1–26.

Duvenaud, D., Lloyd, J., Grosse, R., Tenenbaum, J., and Zoubin, G. (2013). Structure discovery in nonparametric regression through compositional kernel search. In Dasgupta, S. and McAllester, D., editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1166–1174, Atlanta, Georgia, USA. PMLR.

Garnett, R. (2022). *Bayesian Optimization*. Cambridge University Press. in preparation.

Hennig, P. and Schuler, C. J. (2012). Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(6).

Kandasamy, K., Krishnamurthy, A., Schneider, J., and Póczos, B. (2018). Parallelised bayesian optimisation via thompson sampling. In *International Conference on Artificial Intelligence and Statistics*, pages 133–142. PMLR.

Kandasamy, K., Schneider, J., and Póczos, B. (2015). High dimensional bayesian optimisation and bandits via additive models. In *International conference on machine learning*, pages 295–304. PMLR.

Lam, R., Poloczek, M., Frazier, P., and Willcox, K. E. (2018). Advances in bayesian optimization with applications in aerospace engineering. In *2018 AIAA Non-Deterministic Approaches Conference*, page 1656.

Malkomes, G., Schaff, C., and Garnett, R. (2016). Bayesian optimization for automated model selection. In Hutter, F., Kotthoff, L., and Vanschoren, J., editors, *Proceedings of the Workshop on Automatic Machine Learning*, volume 64 of *Proceedings of Machine Learning Research*, pages 41–47, New York, New York, USA. PMLR.

Oh, C., Gavves, E., and Welling, M. (2018). BOCK : Bayesian optimization with cylindrical kernels. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3868–3877. PMLR.

Sano, S., Kadowaki, T., Tsuda, K., and Kimura, S. (2020). Application of bayesian optimization for pharmaceutical product development. *Journal of Pharmaceutical Innovation*, 15(3):333–343.

Shah, A., Wilson, A. G., and Ghahramani, Z. (2013). Bayesian optimization using student-t processes. In *NIPS Workshop on Bayesian Optimization*.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and De Freitas, N. (2015). Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. (2009). Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*.

Wang, Z. and de Freitas, N. (2014). Theoretical analysis of bayesian optimisation with unknown gaussian process hyper-parameters. *arXiv preprint arXiv:1406.7758*.