Supplementary Information

Closed-loop automatic gradient design for liquid chromatography using Bayesian optimization

Jim Boelrijk[a,b,*], Bernd Ensing[a,c], Patrick Forré[a,b], Bob W.J. Pirok[a,d,**]

[a] *AI4Science Lab, Informatics Institute, University of Amsterdam, Amsterdam, Science Park 904, 1098 XH, The Netherlands*

[b] *AMLab, Informatics Institute, University of Amsterdam, Amsterdam, Science Park 904, 1098 XH, The Netherlands*

[c] *Computational Chemistry Group, Van 't Hoff Institute for Molecular Sciences, University of Amsterdam, Amsterdam, Science Park 904, 1098 XH, The Netherlands*

[d] *Analytical Chemistry Group, Van 't Hoff Institute for Molecular Sciences, University of Amsterdam, Amsterdam, Science Park 904, 1098 XH, The Netherlands*

[*] Corresponding author.

[**] Corresponding author.

Email addresses: j.h.m.boelrijk@uva.nl (Jim Boelrijk), bob.pirok@uva.nl (Bob W.J. Pirok)

Preprint submitted to Analytica Chimica Acta *December 12, 2022*

Contents

S-1 Additional Theory on Bayesian Optimization

In this section, we give additional theory on Bayesian Optimization. If these sections are heavy to read, we refer to the excellent books on Gaussian Procceses by Rasmussen [1] and on Bayesian Optimization by Garnett [2]. Parts of the explanation shown here are adapted from earlier work [3].

S-1.1 Gaussian Process

A Gaussian process (GP) is a probabilistic regression model, which given observations of an objective function f at inputs $\mathbf{x}$ can make predictions at unobserved inputs and quantify the uncertainty around them. A GP can be viewed as a multivariate normal distribution which is specified by a mean function $\mu(\mathbf{x})$ and a covariance function. The covariance function is typically defined by a kernel function $k(\mathbf{x}, \mathbf{x}')$, to which we return later.

Now consider a regression problem with N pairs of potentially noisy observations $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, so that we have $\mathbf{y} = \mathbf{f} + \boldsymbol{\epsilon}$, where $\boldsymbol{y} = [y(\mathbf{x}_1), y(\mathbf{x}_2), \ldots, y(\mathbf{x}_n)]^{\mathrm{T}}$ are the outputs, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^{\mathrm{T}}$ are the inputs, and $\boldsymbol{\varepsilon} = [\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n]^{\mathrm{T}}$ are independent identically distributed Gaussian noise with mean 0 and variance σ^2. Then the Gaussian process for $\mathbf{f}$ can be described as:

$$\mathbf{f} = \begin{bmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_N) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(\mathbf{x}_1) \\ \vdots \\ \mu(\mathbf{x}_N) \end{bmatrix}, \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \ldots & k(\mathbf{x}_1, \mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_N, \mathbf{x}_1) & \ldots & k(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix} \right) \tag{1}$$

Then $\mathbf{y}$ is also a Gaussian process, since the sum of two independent random variables is also Gaussian distribution, so that:

$$\mathbf{y} \sim \mathcal{N}\left(\boldsymbol{\mu}(\mathbf{X}), K(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}\right) \tag{2}$$

Here $\mathcal{N}$ is the normal distribution, $\mathbf{I}$ is the identity matrix and $K(\mathbf{X}, \mathbf{X})$ is the Gram matrix (i.e. the right handside of the normal distribution in Eq. 1). By standardizing the observations $\mathbf{y}$ so that it has unit variance and zero mean, the mean function can be set to $\boldsymbol{\mu}(\mathbf{X}) = \mathbf{0}$, which simplifies things and is considered common practice. Then the GP is fully described by the kernel function $k(\mathbf{x}, \mathbf{x}')$, which we will further discuss in Section S-1.2.

First we turn to the task of making predictions using our Gaussian process model given the observed experiments and our kernel, where given some new test inputs $\mathbf{X}^*$, we want to predict the noiseless function outputs $\mathbf{f}^*$. We can do this by defining a joint distribution of both the previous observations and the test inputs so that:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}(\mathbf{X}) \\ \boldsymbol{\mu}(\mathbf{X}^*) \end{bmatrix}, \begin{bmatrix} \mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I} & \mathbf{K}(\mathbf{X}, \mathbf{X}^*) \\ \mathbf{K}(\mathbf{X}^*, \mathbf{X}) & \mathbf{K}(\mathbf{X}^*, \mathbf{X}^*) \end{bmatrix} \right) \tag{3}$$

Then the elegant conditioning properties of Gaussian distributions allow for the computation of the posterior predictive distribution in closed form:

$$p(\mathbf{f}^* \mid \mathbf{X}^*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(\mathbf{y}^* \mid \boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*) \tag{4}$$

with

$$\boldsymbol{\mu}^* = \boldsymbol{\mu}\left(\mathbf{X}^*\right) + \mathbf{K}(\mathbf{X}^*, \mathbf{X})^T \left[\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}\right]^{-1} \left(\mathbf{y} - \boldsymbol{\mu}\left(\mathbf{X}\right)\right) \tag{5}$$

and

$$\boldsymbol{\Sigma}^* = \mathbf{K}(\mathbf{X}^*, \mathbf{X}^*) - \mathbf{K}(\mathbf{X}^*, \mathbf{X}) \left[\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}\right]^{-1} \mathbf{K}(\mathbf{X}, \mathbf{X}^*) \tag{6}$$

So at new test inputs $\mathbf{X}^*$ we can obtain mean predictions $\boldsymbol{\mu}^*$ with uncertainty $\boldsymbol{\Sigma}^*$. Note that when dealing with multiple objectives, we fit independent GPs (as described above) to each objective.

S-1.2 Sorting Kernel

S-1.2.1 Covariance Kernel

As discussed in the previous section, the Gaussian process receives most of its modeling flexibility from the covariance kernel. A popular choice of covariance kernel is the automatic relevance determination (ARD) Matérn 5/2 kernel, which is defined as follows:

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

with:

$$r^2\left(\mathbf{x}, \mathbf{x}'\right) = \sum_{d=1}^{D} \left(x_d - x'_d\right)^2 / \theta_d^2 \tag{8}$$

where $\mathbf{x}$ and $\mathbf{x}'$ could correspond to two different gradient programs, e.g. $\mathbf{x} = [\varphi_1, ..., \varphi_D, t_1, ..., t_D]$ and $\mathbf{x}' = [\varphi'_1, ..., \varphi'_D, t'_1, ..., t'_D]$. Here parameter θ_0 dictates the covariance amplitude and $\theta_{1:D}$ are length scale parameters that determine the smoothness of your model (see Reference [1] for more information regarding intuition behind these parameters or view interactive examples here [1]). The values of these kernel parameters and the measurement noise σ (introduced in the previous section) can be estimated from the previously observed experiments by maximizing the marginal likelihood of the model, which is defined as follows:

$$\ln p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta}, \sigma) = -\frac{1}{2} \mathbf{y}^T \left[\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}\right]^{-1} \mathbf{y} - \frac{1}{2} \ln \left|\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}\right| - \frac{C}{2} \ln 2\pi \tag{9}$$

The three terms have interpretable roles. The first term is a data-fit term, while the second term is a complexity penalty, which favors longer length scales over shorter ones (smooth over oscillating) and hence takes into account overfitting. Lastly, the third parameter is just a constant, originating from the normalizing constant of the normal distribution.

[1] https://distill.pub/2019/visual-exploration-gaussian-processes/

S-1.2.2 Sorting Kernel

It is sensible to have a gradient program in which the mobile phase concentration is monotonically increasing, i.e., $\varphi_i \leq \varphi_{i+1}$. Likewise time points should be ordered, so that $t_i \leq t_{i+1}$. In addition, values of φ_i should be bounded between a volume fraction of 0 and 1, and values of t_i should be between 0 and a user-defined maximum measurement time. While the latter constraints are easily set by box constraints (lower and upper bounds on the input parameters). Constraining the acquisition function so that $\varphi_i \leq \varphi_{i+1}$ and $t_i \leq t_{i+1}$, is less straightforward. Therefore in this work, we use an augmented version of the ARD Matérn 5/2 kernel to incorporate these constraints and is defined as follows:

$$k\left(\mathbf{x}, \mathbf{x}'\right) := k_{\mathrm{M52}}\left([\mathbf{x}], [\mathbf{x}']\right) \tag{10}$$

where $[\mathbf{x}]$ and $[\mathbf{x}']$ are sorted versions of $\mathbf{x}$ and $\mathbf{x}'$ so that all elements $\varphi_{1:D}$ and elements $t_{1:D}$ are ordered from low to high. This introduces a permutation invariance to the kernel and hence the model. This is illustrated in the toy model example in Fig. 3 of the main text, where the model is symmetric with respect to the bottom-left to top-right diagonal, despite that all measured data points are positioned in the upper triangle of the input space. This sorting operation reduces the input parameter space significantly.

S-1.3 Acquisition Functions

We repeat that in single-objective Bayesian Optimization we are considering the problem of finding the maximum of an unknown objective function $f(\mathbf{x})$:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) \tag{11}$$

Where $\mathcal{X}$ denotes the input space, which in our setting are the gradient method parameters to be optimized together with their lower and upper bounds.

The role of the acquisition (denoted as α) is to query the Gaussian process and to propose parameters $\mathbf{x}$ so that we maximize $f(\mathbf{x})$. As the Gaussian process model is probabilistic, it provides us with values of the objective function $\mu(\mathbf{x})$ which are normally distributed with a standard deviation $\sigma(\mathbf{x})$. Here $\mu(\mathbf{x})$ is computed using equation 5 and $\sigma(\mathbf{x})$ using equation 6. As explained in the main text, acquisition functions exploit these quantities to make a trade-off between exploration and exploitation.

S-1.3.1 Expected Improvement

The Expected Improvement (EI) acquisition function is an improvement based policy that favors points that are likely to improve on the previous best experiment f' and has proven convergence rates [4, 5]. It defines the following improvement function:

$$I(\mathbf{x}) := (f(\mathbf{x}) - f')\mathbb{I}(f(\mathbf{x}) > f') \tag{12}$$

Where $\mathbb{I}$ is defined as the indicator function, which is 1 if and only if $f(\mathbf{x}) > f'$ and 0 otherwise. Therefore $I(\mathbf{x}) > 0$ if and only if there is an improvement of $f(\mathbf{x})$ over f'. As $f(\mathbf{x})$

is described by a Gaussian process, it is a Gaussian random variable, and the expectation of this function can be computed analytically as follows:

$$\alpha_{\mathrm{EI}}(\mathbf{x}) := \mathbb{E}[I(\mathbf{x})] = (\mu(\mathbf{x}) - f')\,\Phi\left(\frac{\mu(\mathbf{x}) - f'}{\sigma(\mathbf{x})}\right) + \sigma(\mathbf{x})\phi\left(\frac{\mu(\mathbf{x}) - f'}{\sigma(\mathbf{x})}\right) \tag{13}$$

Here Φ is the standard normal cumulative distribution function, and ϕ is the standard normal probability density function. By maximizing $\alpha_{\mathrm{EI}}(\mathbf{x})$, the amount of improvement is taken into account, and naturally balances between exploration and exploitation.

S-1.3.2 Expected Hypervolume Improvement

In the setting of multi-objective optimization, where we have $M \geq 2$ objective functions, $f_1, ..., f_M$, it is desirable, but usually not possible to find a single point $\mathbf{x}^*$ that maximizes all objectives at the same time. For instance, a specific value of $\mathbf{x}$ might maximize f_1, but might lead to a non-optimal value of f_2, etc. So therefore, the best we can do in this setting is to explore (and find) the Pareto front (See Figure 7 of the main text and accompanying explanation.). The hypervolume (i.e. area in 2D, volume in 3D) of the points on the Pareto front indicates the quality of the Pareto front we have found thus far. Therefore finding the point $\mathbf{x}$ which yields values $f_1, ..., f_M$ that maximizes the hypervolume leads to a better estimation of the Pareto front. This is exactly what the Expected Hypervolume Improvement acquisition function [6] aims to do. It essentially is the natural extension of the Expected Improvement acquisition function to multiple objectives. We first define the Hypervolume Improvement (HVI) as follows:

$$\mathrm{HVI}(\mathbf{f}) := \mathrm{HV}_{\mathbf{r}}^{M}(\mathbf{f}, \mathbf{f}_1, \ldots, \mathbf{f}_N) - \mathrm{HV}_{\mathbf{r}}^{M}(\mathbf{f}_1, \ldots, \mathbf{f}_N). \tag{14}$$

Where $\mathrm{HV}_{\mathbf{r}}^{M}$ is the M-dimensional hypervolume, w.r.t a reference point $\mathbf{r} \in \mathbb{R}^M$, and $\mathbf{f} = f_1, ..., f_M$. The hypervolume improvement for an example point is shown in green in Figure 7 of the main text. As $\mathbf{f}(\mathbf{x})$ is coming from the Gaussian process it is normally distributed and we can take the expectation of the HVI (to balance exploration and exploitation) leading to the Expected Hypervolume Improvement (EHVI) acquistion function:

$$\alpha_{\mathrm{EHVI}}(\mathbf{x}) = \mathbb{E}\left[\mathrm{HVI}\left(\boldsymbol{f}\left(\mathbf{x}\right)\right)\right] \tag{15}$$

This expectation can in some cases be described analytically [6], but can also be approximated using Monte Carlo methods. In this work we use qEHVI, which is a Monte Carlo approximation of the α_{EHVI} described above, and uses auto-differentation for efficient optimization [7].

S-2 Variations of reference point

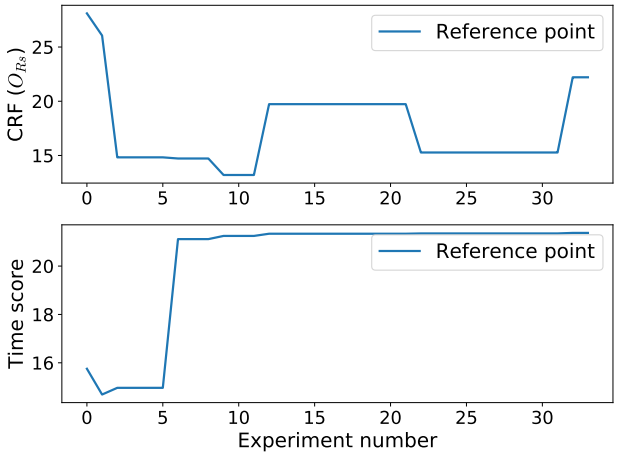

Figure 1: Variations of reference point values as a function of experiment number for sample A (see Section 4.4 of the main text).

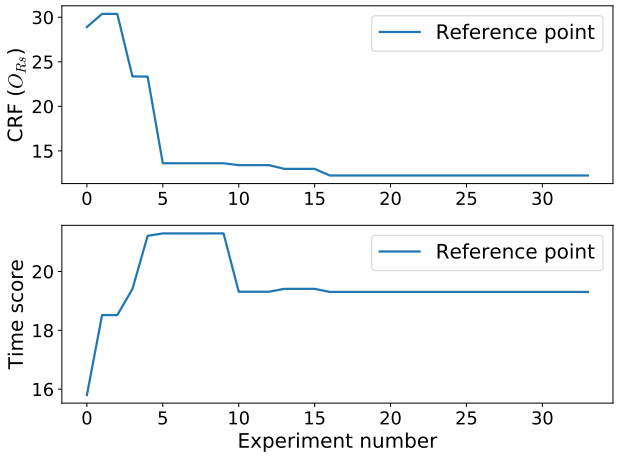

Figure 2: Variations of reference point values as a function of experiment number for sample B (see Section 4.4 of the main text).

References

[1] C. E. Rasmussen, Gaussian Processes in Machine Learning, volume 3176, Springer Verlag, 2004.

[2] R. Garnett, Bayesian Optimization, Cambridge University Press, 2022.

[3] J. Boelrijk, B. Pirok, B. Ensing, P. Forré, Bayesian optimization of comprehensive two-dimensional liquid chromatography separations, Journal of Chromatography A 1659 (2021) 462628.

[4] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, N. De Freitas, Taking the human out of the loop: A review of Bayesian optimization, Proceedings of the IEEE 104 (2016) 148–175.

[5] A. D. Bull, Convergence rates of efficient global optimization algorithms, Journal of Machine Learning Research 12 (2011) 2879–2904.

[6] K. Yang, M. Emmerich, A. Deutz, T. Bäck, Multi-Objective Bayesian Global Optimization using expected hypervolume improvement gradient, Swarm and Evolutionary Computation 44 (2019) 945–956.
[7] S. Daulton, M. Balandat, E. Bakshy, Differentiable Expected Hypervolume Improvement for Parallel Multi-Objective Bayesian Optimization, arXiv (2020).