Supplementary Material for: Fast Information-theoretic Bayesian Optimisation

Binxin Ru
University of Oxford
Oxford OX1 3PJ
robin@robots.ox.ac.uk

Mark Mcleod
University of Oxford
Oxford OX1 3PJ
mark.mcleod@magd.ox.ac.uk

Michael Osborne
University of Oxford
Oxford OX1 3PJ
mosb@robots.ox.ac.uk

1 FITBO Algorithm

The procedures of FITBO approach can be summarised by Algorithm 1 and Figure 1 illustrates the sampling behaviour of FITBO method for a simple 1-D Bayesian optimisation problem. The optimisation process is started with 3 initial observation data. As more samples are taken, the mean of the posterior distribution for the objective function gradually resembles the objective function and the distribution of η converges to the global minimum.

Algorithm 1 FITBO

```
Input: a test input $\mathbf{x}$; noisy observation data $D_n = \{(\mathbf{x}_i, y_i) | i = 1, \dots, n\}$

sample hyperparameters and $\eta$ from $p(\boldsymbol{\psi}|D_n)$: $\boldsymbol{\Psi} = \{\boldsymbol{\theta}^{(j)}, \eta^{(j)} | j = 1, \dots, M\}$
for j=1,..., $M$ do
    use $f(\mathbf{x}) = \eta + 1/2 g(\mathbf{x})^2$ to approximate $p(f|D_n, \mathbf{x}, \boldsymbol{\theta}^{(j)}, \eta^{(j)}) = \mathcal{GP}\big(m_f(\cdot), K_f(\cdot, \cdot)\big)$
        3.1) compute $D_g = \{(\mathbf{x}_i, g_i) | i = 1, \dots, n\}$ where $g_i = \sqrt{2(y_i - \eta^{(j)})}$
        3.2) compute posterior distribution $p(g|D_g, \mathbf{x}, \boldsymbol{\theta}^{(j)}, \eta^{(j)})$
        3.3) approximate $p(f|D_n, \mathbf{x}, \boldsymbol{\theta}^{(j)}, \eta^{(j)})$ using the linearisation technique
    compute $p(y|D_n, \mathbf{x}, \boldsymbol{\theta}^{(j)}, \eta^{(j)})$
    compute $H[p(y|D_n, \mathbf{x}, \boldsymbol{\theta}^{(j)}, \eta^{(j)})]$
end for
estimate entropy of the Gaussian mixture :
        $E1(\mathbf{x}|D_n) = H\Big[\frac{1}{M}\sum_j^M p(y|D_n, \mathbf{x}, \boldsymbol{\theta}^{(j)}, \eta^{(j)})\Big]$
compute the entropy expectation: $E2(\mathbf{x}|D_n) = \frac{1}{2M}\sum_j^M \log\big[2\pi e\big(v_f(\mathbf{x}|D_n, \boldsymbol{\theta}^{(j)}, \eta^{(j)}) + \sigma_n^2\big)\big]$
return $\alpha_n(\mathbf{x}|D_n) = E1(\mathbf{x}|D_n) - E2(\mathbf{x}|D_n)$
```

2 Compare Methods for Approximating a Gaussian Mixture Entropy

2.1 Method 1: Taylor Expansion

[1] propose a novel method for approximating the entropy of a Gaussian mixture model by using a Taylor-series expansion of the logarithm of the Gaussian mixture.

Let $q(y) = \sum_i^N w_i p(y|D_n, \mathbf{x}, \eta^{(i)}) = \sum_i^N w_i \mathcal{N}(y; m_i, \sigma_i^2)$. The Gaussians in the mixture are univariate in our case because the function value at a test location $\mathbf{x}$ is 1-D. The entropy of this

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

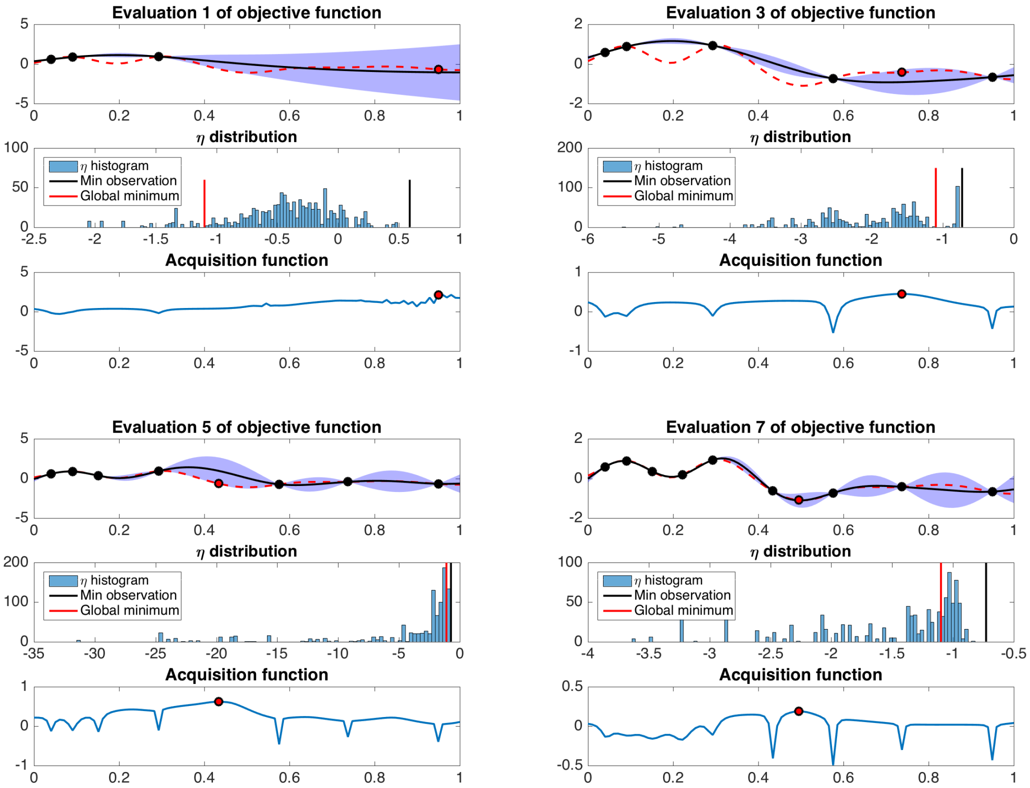

Figure 1: Bayesian optimisation for a 1-D objective function using FITBO method at the 1st, 3rd, 5th, 7th evaluations. In each subfigure, the top plot shows the objective function (red dotted line), the posterior mean (black solid line) and the 95% confidence interval (blue shaded area) estimated by the Gaussian process model as well as the observation points (black dot) and the next query point (red dot). The middle plot is the histogram of η samples and its relation to the minimum observation (black vertical line) and the true global minimum (red vertical line).

mixture is:

$$H\left[\frac{1}{N}\sum_{i}^{N} p(y|D,\mathbf{x},\eta^{(i)})\right] = H\left[q(y)\right] = -\int q(y)\log h(y)\mathrm{d}y \tag{1}$$

where $h(y) = q(y)$ but we uses different notations to differentiate the Gaussian mixture that's argument of the logarithm from that in front of the logarithm.

By expanding the logarithm term around the mean of each Gaussian term m_i in $h(y)$, the resultant R-th order Taylor series is

$$\log h(y) = \sum_{k=0}^{R} \frac{(y-m_i)^k}{k!} \frac{\mathrm{d}^k\big(\log h(y)\big)}{\mathrm{d}y^k}\bigg|_{y=m_i}. \tag{2}$$

We then substitute equation 2 into equation 1 and obtain

$$\begin{aligned}
H\left[q(y)\right] &= -\int q(y)\log h(y)dy \\
&= -\int \sum_{i}^{N} w_i \mathcal{N}(y; m_i, \sigma_i^2)\log h(y)\mathrm{d}y \\
&= -\sum_{i}^{N} w_i \int \mathcal{N}(y; m_i, \sigma_i^2) \sum_{k=0}^{R} \frac{(y-m_i)^k}{k!} \frac{\mathrm{d}^k\big(\log h(y)\big)}{\mathrm{d}y^k}\Big|_{y=m_i} \mathrm{d}y \\
&= -\sum_{i}^{N} w_i \sum_{k=0}^{R} \frac{1}{k!} \frac{\mathrm{d}^k\big(\log h(y)\big)}{\mathrm{d}y^k}\Big|_{y=m_i} \int \mathcal{N}(y; m_i, \sigma_i^2)(y-m_i)^k \mathrm{d}y
\end{aligned}$$

where $\int \mathcal{N}(y; m_i, \sigma_i^2)(y-m_i)^k \mathrm{d}y$ is the k-th central moment of a Gaussian distribution and thus has a closed form. The k-th derivative of the logarithm of Gaussian mixture $h(y)$ can also be computed analytically because the derivatives of a Gaussian distribution always exist and Kronecker algebra can be used to achieve a compact representation [1].

The entropy approximation by Taylor expansion faces the trade-off between the accuracy and computational burden as we can obtain more accurate approximations by including higher order Taylor-series terms at the expense of computational speed [1]. Experiments with this approximation approach are carried out with a second-order Taylor-series expansion whose explicit form is provided by the Appendix in [1]:

$$H\left[\frac{1}{N}\sum_{i=1}^{N} p(y|D_n, \mathbf{x}, \eta^{(i)})\right] \approx H_0[y] + H_2[y] = -\sum_{i=1}^{N} w_i \log h(m_i) - \sum_{i=1}^{N} \frac{w_i\sigma_i^2}{2} F(m_i) \qquad (3)$$

where $F(y) = h(y)^{-1}\sum_{j=1}^{N} w_j\sigma_j^{-2}\left[h(y)^{-1}(y-\mu_j)h'(y) + \sigma_j^{-2}(y-\mu_j)^2 - 1\right]\mathcal{N}(y; \mu_j, \sigma_j^2)$.

2.2 Method 2: Numerical Integration

As mentioned before, one advantage of FITBO method is that it allows us to transform the entropy calculation from the multi-dimensional input space to the one-dimensional output space. This, thus, permits the use of numerical integration techniques to effectively compute the entropy of a Gaussian mixture. Experiments with numerical integration are performed with the *quad* function in MATLAB which utilises the adaptive Simpson quadrature.

2.3 Method 3: Simple Monte Carlo

The first term in our FITBO acquisition function can be reformulated in the following way:

$$\begin{aligned}
& H\left[\frac{1}{N}\sum_{i}^{N} p(y|D_n, \mathbf{x}, \eta^{(i)})\right] \\
=\; & H\left[\sum_{i}^{N} w_i p(y|D_n, \mathbf{x}, \eta^{(i)})\right] \qquad \text{where} \qquad w_i = \frac{1}{N} \\
=\; & -\int \left(\sum_{i}^{N} w_i p(y|D_n, \mathbf{x}, \eta^{(i)})\right) \log\left(\sum_{i}^{N} w_i p(y|D_n, \mathbf{x}, \eta^{(i)})\right) \mathrm{d}y \\
=\; & -\sum_{i}^{N} w_i \int p(y|D_n, \mathbf{x}, \eta^{(i)}) \log\left(\sum_{i}^{N} w_i p(y|D_n, \mathbf{x}, \eta^{(i)})\right) \mathrm{d}y
\end{aligned}$$

By drawing M samples of y from $p(y|D_n, \mathbf{x}, \eta^{(i)})$ and using Monte Carlo integration, the entropy of a Gaussian mixture can be approximated as

$$H\left[\frac{1}{N}\sum_{i}^{N} p(y|D_n, \mathbf{x}, \eta^{(i)})\right] \approx -\sum_{i}^{N} w_i \left[\frac{1}{M}\sum_{j}^{M} \log\left(\sum_{i}^{N} w_i p(y^{(j)}|D_n, \mathbf{x}, \eta^{(i)})\right)\right] \qquad (4)$$

The accuracy of the simple Monte Carlo approximation can be enhanced by increasing the sample size M. But larger number of samples will increase the computational burden. Thus, we also face a trade-off between the approximation precision and computational speed.

2.4 Experiments for Comparing Approximation Methods

The following experiments are conducted to validate as well as compare the three entropy approximation methods: 1) Huber's method that uses Taylor series expansion (Huber), 2) numerical integration that uses adaptive Simpson quadrature (Quadra) and 3) the simple Monte Carlo integration (MC). The approximation performance is assessed in terms of accuracy and computational speed. The optimal approximation method is then chosen based on the trade-off between the accuracy and computational demand.

The methodology of the tests can be summarised as follows:

[1] Generate a Gaussian mixture as a weighted sum of N 1-D random Gaussian distributions

[2] For the Gaussian mixture, estimate its true entropy by using simple Monte Carlo method with large sample size (e.g. MC50000)

[3] Use the 3 approximation methods to approximate the entropy of the Gaussian mixture. For the MC method, try it with different sample sizes (e.g. MC10,MC100,MC1000)

[4] Compute and record the running time as well as absolute and fractional approximation errors for each method.

[5] Repeat the above processes for M different gaussian mixtures and compute the median running time and the median of the approximation errors.

With reference to Figure 2 and 3, in the case of a single Gaussian($N = 1$) distribution, there is a closed-form expression for its entropy. The Huber's method gives the exact true entropy solution, thus having 0 approximation error. The other 2 approximation methods (Quadra and MC) are compared against the true entropy value. It is evident that the approximation by Monte Carlo with 50000 samples (MC50000) is very close to the true value, which justifies our use of the approximation results of MC50000 as our yardstick for the cases of more than one Gaussians in the mixture.

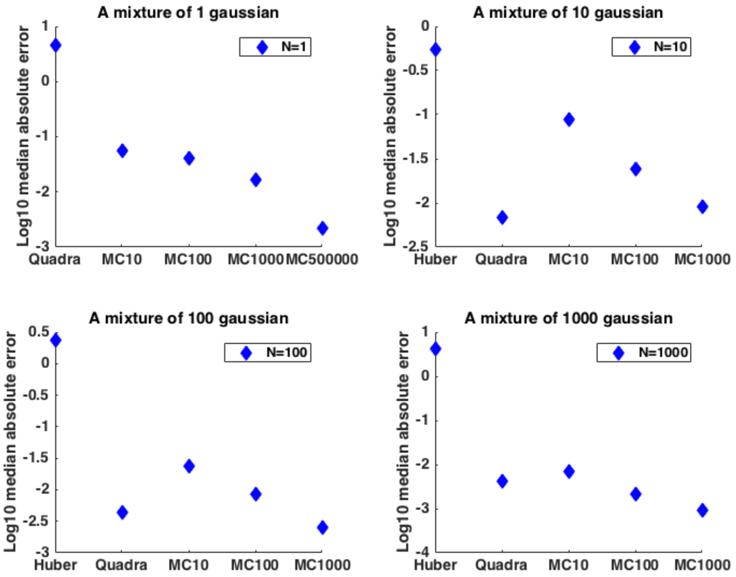

Figure 2: Log median absolute error in approximating the entropy of a Gaussian mixture.

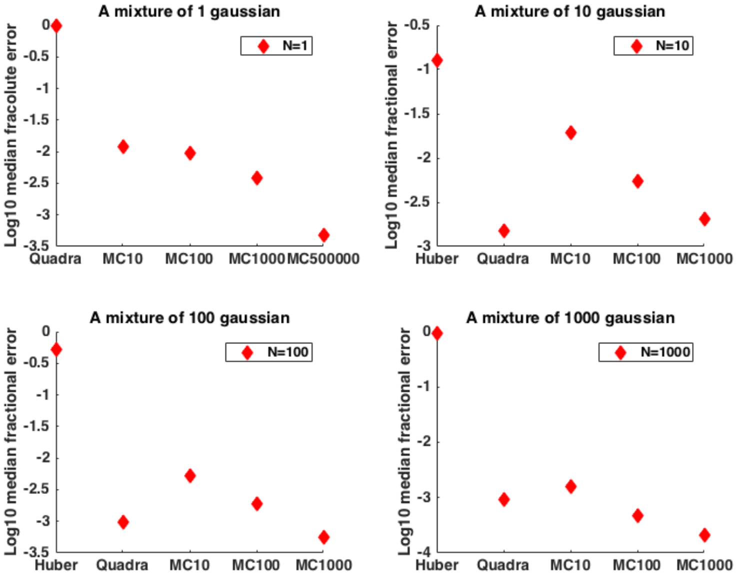

Figure 3: Log median fractional error in approximating the entropy of a gaussian mixture

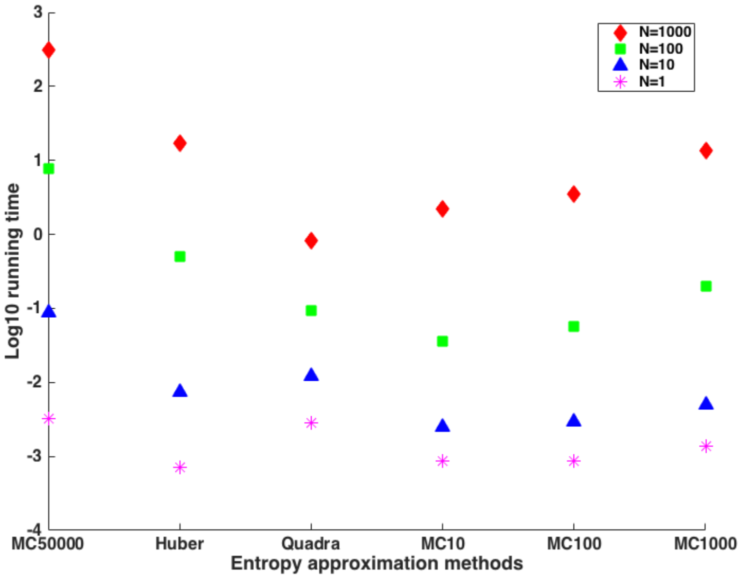

Figure 4: Compare the running times of the approximation methods.

For a mixture of more than one Gaussian distribution ($N > 1$), the performances of all 3 approximation methods (Huber, Quadra, MC) are compared against the entropy value estimated by MC50000. The results in Figure 2 and 3 show that Monte Carlo with a sample size of 1000 (MC1000) produces the most accurate approximation in terms of absolute and fractional approximation errors. MC100 and quadrature (Quadra) also have relatively accurate approximation with low absolute and fractional error. The Huber method leads to the highest approximation errors. This may be due to the low order (order of 2) of Taylor-series expansion used in our experiments.

In Figure 4, the running times of all 3 approximation methods are compared. As expected, the results show that the computation time increases as the number of Gaussians in the mixture rises. This is mainly due to the computation burden associated with the construction of the Gaussian mixture. More importantly, the quadrature method (Quadra) gains speed advantage as the number of gaussians in the mixture increases because the computational cost of approximation using quadrature does not increase with the number of Gaussian components in the mixture. The speed advantage of the quadrature method becomes more salient as we have more Gaussians in the mixture which is reflected in the growing difference among the running times of these methods. Since the number of gaussians in the mixture (N) is determined by the number of hyperparameter samples we use for marginalisation in our algorithm, if we want to use a larger number of hyperparameter sets, we should adopt the quadrature method for fast approximation of the Gaussian mixture entropy at decent accuracy.

References

[1] Marco F Huber, Tim Bailey, Hugh Durrant-Whyte, and Uwe D Hanebeck. On entropy approximation for gaussian mixture random vectors. In *Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE International Conference on*, pages 181–188. IEEE, 2008.