Appendix of Fast Information-theoretic Bayesian Optimisation

Binxin Ru [1] **Mark McLeod** [1] **Diego Granziol** [1] **Michael A. Osborne** [1 2]

1. Compare Methods for Approximating a Gaussian Mixture Entropy

1.1. Method 1: Taylor Expansion

Huber et al. (2008) 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 $q(y) = \sum_j^M w_j \mathcal{N}(y; m_j, \sigma_j^2)$. 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 mixture is:

$$H\left[q(y)\right] = -\int q(y) \log h(y) \mathrm{d}y \tag{1}$$

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_j in $h(y)$, the resultant R-th order Taylor series is

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

We then substitute equation 2 into equation 1 and obtain

$$\begin{aligned} H\left[q(y)\right] &= -\int q(y) \log h(y) dy \\ &= -\int \sum_j^M w_j \mathcal{N}(y; m_j, \sigma_j^2) \log h(y) \mathrm{d}y \\ &= -\sum_j^M w_j \sum_{k=0}^{R} \frac{1}{k!} \frac{\mathrm{d}^k\big(\log h(y)\big)}{\mathrm{d}y^k}\bigg|_{y=m_j} \\ &\quad \int \mathcal{N}(y; m_j, \sigma_j^2)(y-m_j)^k \mathrm{d}y \end{aligned}$$

where $\int \mathcal{N}(y; m_j, \sigma_j^2)(y-m_j)^k \mathrm{d}y$ is the k-th central moment of a Gaussian distribution and thus has a closed form (Requeima, 2016). 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 (Huber et al., 2008).

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 (Huber et al., 2008). Experiments with this approximation approach are carried out with a second-order Taylor-series expansion whose explicit form is provided by the Appendix in (Huber et al., 2008):

$$\begin{aligned} & H\left[\frac{1}{N}\sum_{j=1}^{M} p(y|D_n, \mathbf{x}, \eta^{(j)})\right] \\ &\approx H_0[y] + H_2[y] \\ &= -\sum_{j=1}^{M} w_j \log h(m_j) - \sum_{j=1}^{M} \frac{w_j \sigma_j^2}{2} F(m_j) \end{aligned}$$

where

$$\begin{aligned} F(y) =& h(y)^{-1} \sum_{i=1}^{N} w_i \sigma_i^{-2} [h(y)^{-1}(y-\mu_i)h'(y) \\ &+ \sigma_i^{-2}(y-\mu_i)^2 - 1] \mathcal{N}(y; \mu_i, \sigma_i^2). \end{aligned}$$

[1]Department of Engineering Science, University of Oxford, Oxford, UK [2]Mind Foundry Ltd., Oxford. Correspondence to: Binxin Ru <robin@robots.ox.ac.uk>, Mark McLeod <mark.mcleod@magd.ox.ac.uk>, Diego Granziol <diego@robots.ox.ac.uk>, Michael A. Osborne <mosb@robots.ox.ac.uk>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

1.2. 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.

1.3. Method 3: Simple Monte Carlo

Let $p(y|I^{(j)})$ denotes $p(y|D_n, \mathbf{x}, \eta^{(j)})$. The first term in our FITBO acquisition function can be reformulated in the following way:

$$
\begin{aligned}
& H\left[\sum_j^M w_j p(y|I^{(j)})\right] \\
&= -\int \left(\sum_j^M w_j p(y|I^{(j)})\right) \log\left(\sum_j^M w_j p(y|I^{(j)})\right) \mathrm{d}y \\
&= -\sum_j^M w_j \int p(y|I^{(j)}) \log\left(\sum_j^M w_j p(y|I^{(j)})\right) \mathrm{d}y
\end{aligned}
$$

where $w_j = \frac{1}{N}$ in our case. By drawing N samples of y from $p(y|I^{(j)})$ and using Monte Carlo integration, the entropy of a Gaussian mixture can be approximated as

$$
\begin{aligned}
& H\left[\sum_j^M w_j p(y|I^{(j)})\right] \\
&\approx -\sum_j^M w_j \left[\frac{1}{N}\sum_i^N \log\left(\sum_j^M w_j p(y^{(i)}|I^{(j)})\right)\right] \qquad (3)
\end{aligned}
$$

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.

1.4. 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 K different gaussian mixtures and compute the median running time and the median of the approximation errors.

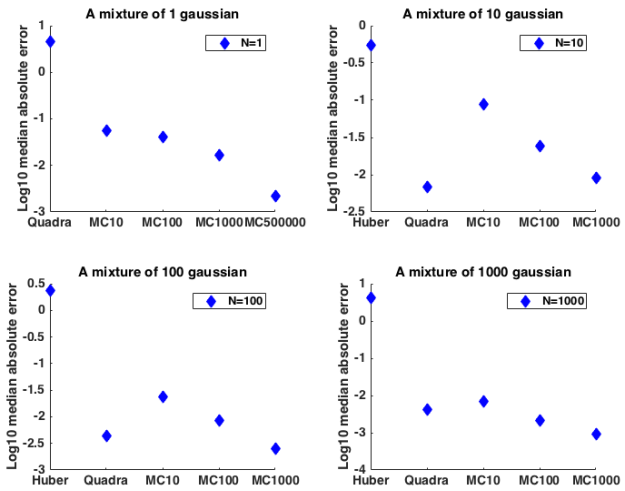

Figure 1. Log median absolute error in approximating the entropy of a Gaussian mixture.

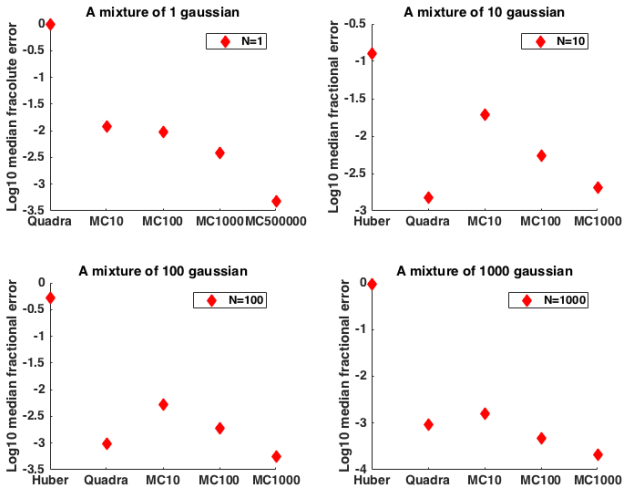

Figure 2. Log median fractional error in approximating the entropy of a gaussian mixture

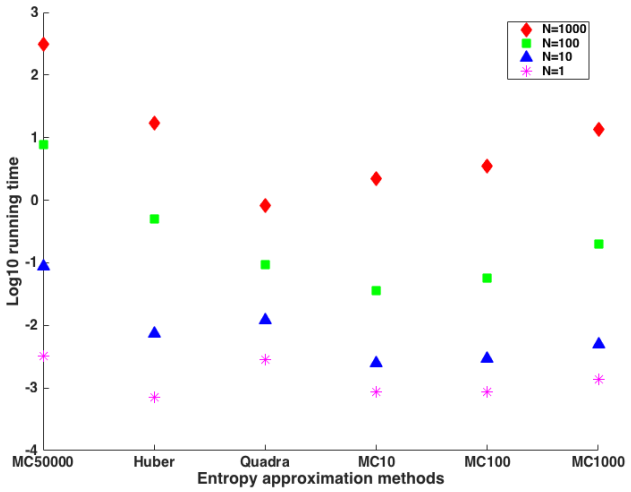

Figure 3. Compare the running times of the approximation methods.

With reference to Figure 1 and 2, 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.

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 1 and 2 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 3, 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.

2. Compute the Derivative of the Acquisition Function

The acquisition function of our FITBO approach has the following form:

$$\alpha(\mathbf{x}|D_n) = H\Big[\frac{1}{M}\sum_j^M p(y|D_n, \mathbf{x}, \boldsymbol{\theta}^{(j)}, \eta^{(j)})\Big] - \frac{1}{2M}\sum_j^M \log\left[2\pi e\big(v_f(\mathbf{x}|D, \boldsymbol{\theta}^{(j)}, \eta^{(j)}) + \sigma_n^2\big)\right] \tag{4}$$

where $v_f(\mathbf{x}|D, \boldsymbol{\theta}^{(j)}, \eta^{(j)}) = K_f^{(j)}(\mathbf{x}, \mathbf{x}') = m_g^{(j)}(\mathbf{x}) K_g^{(j)}(\mathbf{x}, \mathbf{x}') m_g^{(j)}(\mathbf{x}')$.

If the kernel function adopted is differentiable, we can compute the derivative of $\alpha(\mathbf{x}|D_n)$ with respect to x_d, the d^{th} dimension of $\mathbf{x}$, to facilitate the optimisation of the acquisition function.

To simplify our notation, let $G(y|\mathbf{x}) = \frac{1}{M}\sum_j^M p(y|\mathbf{x}, \boldsymbol{\psi}^{(j)}) = \frac{1}{M}\sum_j^M p(y|D_n, \mathbf{x}, \boldsymbol{\theta}^{(j)}, \eta^{(j)})$

and $v_j(\mathbf{x}) = v_f(\mathbf{x}|D, \boldsymbol{\theta}^{(j)}, \eta^{(j)}) + \sigma_n^2$. The acquisition function of FITBO then becomes:

$$\alpha(\mathbf{x}|D_n) = H\Big[G(y|\mathbf{x})\Big] - \frac{1}{2M}\sum_j^M \log\big[2\pi e\big(v_j(\mathbf{x})\big)\big].$$

2.1. Derivative of FITBO acquisition function

If we approximate the entropy of the Gaussian mixture using numerical method, the derivative of $\alpha(\mathbf{x}|D_n)$ can be computed as follows:

$$\begin{aligned}\frac{\partial\alpha(\mathbf{x}|D_n)}{\partial x_d} &= \frac{\partial\Big[-\int G(y|\mathbf{x})\log G(y|\mathbf{x})\mathrm{d}y\Big]}{\partial x_d} \\ &\quad - \frac{1}{2M}\sum_j^M \frac{\partial\log\big[2\pi e\big(v_j(\mathbf{x})\big)\big]}{\partial x_d} \\ &= \Big[-\int \frac{\partial G(y|\mathbf{x})}{\partial x_d} + \log G(y|\mathbf{x})\frac{\partial G(y|\mathbf{x})}{\partial x_d}\mathrm{d}y\Big] \\ &\quad - \frac{1}{2M}\sum_j^M \frac{1}{v_j(\mathbf{x})}\frac{\partial v_j(\mathbf{x})}{\partial x_d} \end{aligned} \tag{5}$$

The key partial derivative needs to solve is $\frac{\partial G(y|\mathbf{x})}{\partial x_d}$ and for a 1D Gaussian mixture:

$$\begin{aligned}\frac{\partial G(y|\mathbf{x})}{\partial x_d} &= \frac{1}{M}\sum_j^M \frac{\partial p(y|\mathbf{x}, \boldsymbol{\psi}^{(j)})}{\partial x_d} \\ &= \frac{1}{M}\sum_j^M \frac{\partial\big(\frac{1}{\sqrt{2\pi e v_j(\mathbf{x})}}\exp(-\frac{(y-m_j(\mathbf{x}))^2}{2v_j(\mathbf{x})})\big)}{\partial x_d} \\ &= \frac{1}{M}\sum_j^M \Big[-p(y|\mathbf{x}, \boldsymbol{\psi}^{(j)})\frac{1}{2v_j(\mathbf{x})}\frac{\partial v_j(\mathbf{x})}{\partial x_d} \\ &\quad + p(y|\mathbf{x}, \boldsymbol{\psi}^{(j)})\frac{(y-m_j(\mathbf{x}))^2}{2(v_j(\mathbf{x}))^2}\frac{\partial v_j(\mathbf{x})}{\partial x_d} \\ &\quad + p(y|\mathbf{x}, \boldsymbol{\psi}^{(j)})\frac{(y-m_j(\mathbf{x}))}{v_j(\mathbf{x})}\frac{\partial m_j(\mathbf{x})}{\partial x_d}\Big] \\ &= \frac{1}{M}\sum_j^M p(y|\mathbf{x}, \boldsymbol{\psi}^{(j)})\Bigg[\frac{(y-m_j(\mathbf{x}))}{v_j(\mathbf{x})}\frac{\partial m_j(\mathbf{x})}{\partial x_d} \\ &\quad + \Big(\frac{(y-m_j(\mathbf{x}))^2 - v_j(\mathbf{x})}{2(v_j(\mathbf{x}))^2}\Big)\frac{\partial v_j(\mathbf{x})}{\partial x_d}\Bigg] \end{aligned} \tag{6}$$

Thus, $\int \frac{\partial G(y|\mathbf{x})}{\partial x_d}\mathrm{d}y = 0$ and the first derivative term in function 5 can be expanded in the following way:

$$\begin{aligned}&-\int \frac{\partial G(y|\mathbf{x})}{\partial x_d} + \log G(y|\mathbf{x})\frac{\partial G(y|\mathbf{x})}{\partial x_d}\mathrm{d}y \\ &= -\int \log G(y|\mathbf{x})\frac{\partial G(y|\mathbf{x})}{\partial x_d}\mathrm{d}y \\ &= -\int \log G(y|\mathbf{x})\frac{1}{M}\sum_j^M \Bigg\{ \\ &\quad p(y|\mathbf{x}, \boldsymbol{\psi}^{(j)})\Bigg[\frac{(y-m_j(\mathbf{x}))}{v_j(\mathbf{x})}\frac{\partial m_j(\mathbf{x})}{\partial x_d} \\ &\quad + \Big(\frac{(y-m_j(\mathbf{x}))^2 - v_j(\mathbf{x})}{2(v_j(\mathbf{x}))^2}\Big)\frac{\partial v_j(\mathbf{x})}{\partial x_d}\Bigg]\Bigg\}\mathrm{d}y \end{aligned} \tag{7}$$

The derivative of the acquisition function then has the form:

$$\begin{aligned}&\frac{\partial\alpha(\mathbf{x}|D_n)}{\partial x_d} \\ &= -\int \log G(y|\mathbf{x})\frac{1}{M}\sum_j^M \Bigg\{p(y|\mathbf{x}, \boldsymbol{\psi}^{(j)})\Bigg[\\ &\quad \Big(\frac{(y-m_j(\mathbf{x}))^2 - v_j(\mathbf{x})}{2(v_j(\mathbf{x}))^2}\Big)\frac{\partial v_j(\mathbf{x})}{\partial x_d} \\ &\quad + \frac{(y-m_j(\mathbf{x}))}{v_j(\mathbf{x})}\frac{\partial m_j(\mathbf{x})}{\partial x_d}\Bigg]\Bigg\}\mathrm{d}y \\ &\quad - \frac{1}{2M}\sum_j^M \frac{1}{\big(v_j(\mathbf{x}) + \sigma_n^2\big)}\frac{\partial v_j(\mathbf{x})}{\partial x_d} \end{aligned} \tag{8}$$

where

$$\frac{\partial m_j(\mathbf{x})}{\partial x_d} = m_g^{(j)}(\mathbf{x})\frac{\partial m_g^{(j)}(\mathbf{x})}{\partial x_d}, \tag{9}$$

$$\begin{aligned}\frac{\partial v_j(\mathbf{x})}{\partial x_d} &= \frac{\partial K_g^{(j)}(\mathbf{x}, \mathbf{x}')}{\partial x_d}\big(m_g^{(j)}(\mathbf{x})\big)^2 \\ &\quad + 2K_g^{(j)}(\mathbf{x}, \mathbf{x}')m_g^{(j)}(\mathbf{x})\frac{\partial m_g^{(j)}(\mathbf{x})}{\partial x_d}\end{aligned} \tag{10}$$

and $\frac{\partial m_g(\mathbf{x})}{\partial x_d}$ and $\frac{\partial K_g(\mathbf{x}, \mathbf{x}')}{\partial x_d}$ can be computed from the definition of the chosen kernel function.

2.2. Derivative of FITBO-MM acquisition function

A faster alternative to approximate the entropy of the Gaussian mixture is to use simple moment-matching:

$$G(y|\mathbf{x}) = \frac{1}{M}\sum_j^M p(y|\mathbf{x}, \boldsymbol{\psi}^{(j)}) \approx \mathcal{N}(y|m_G(\mathbf{x}), v_G(\mathbf{x}))$$

where

$$m_G(\mathbf{x}) = \sum_j^M \frac{1}{M}m_j(\mathbf{x})$$

$$v_G(\mathbf{x}) = \sum_j^M \frac{1}{M}\big(v_j(\mathbf{x}) + (m_j(\mathbf{x}))^2\big) - \big(m_G(\mathbf{x})\big)^2.$$

This leads to an analytical form of the acquisition function:

$$\alpha(\mathbf{x}|D_n) \approx \frac{1}{2}\log\big[2\pi e\big(v_G(\mathbf{x})\big)\big] - \frac{1}{2M}\sum_j^M \log\big[2\pi e\big(v_j(\mathbf{x})\big)\big].$$

The derivative of the acquisition function then has a neat analytical form:

$$\frac{\partial\alpha(\mathbf{x}|D_n)}{\partial x_d} = \frac{1}{2v_G(\mathbf{x})}\frac{\partial v_G(\mathbf{x})}{\partial x_d} - \frac{1}{2M}\sum_j^M \frac{1}{v_j(\mathbf{x})}\frac{\partial v_j(\mathbf{x})}{\partial x_d}$$

where

$$\frac{\partial v_G(\mathbf{x})}{\partial x_d} = \sum_j^M \frac{1}{M}\Big(\frac{\partial v_j(\mathbf{x})}{\partial x_d} + 2m_j(\mathbf{x})\frac{\partial m_j(\mathbf{x})}{\partial x_d}\Big) - 2m_G(\mathbf{x})\Big(\sum_j^M \frac{1}{M}\frac{\partial m_j(\mathbf{x})}{\partial x_d}\Big)$$

with $\frac{\partial m_j(\mathbf{x})}{\partial x_d}$ and $\frac{\partial v_j(\mathbf{x})}{\partial x_d}$ have the same expressions as equations 9 and 10.

3. The Acquisition Functions Obtained by Numerical Integration and Moment-matching

In this section, we compare the resultant acquisition functions obtained by using (1) numerical integration (FITBO) and (2) moment-matching (FITBO-MM) to approximate Gaussian mixture entropy. Figure 4 shows the acquisition function of FITBO-MM in comparison with those of FITBO which use different tolerance levels (1e-3, 1e-4, 1e-6) for numerical integration. We assume the acquisition function obtained using numerical integration with a tolerance level of 1e-6 (pink line in Figure 4) to be a fair representation of the true value. It is evident that the moment-matching method leads to a looser upper bound than numerical integration but the resultant acquisition function manages to capture the true function shape quite well and recommend a query point that is very close to the best numerical approximation.

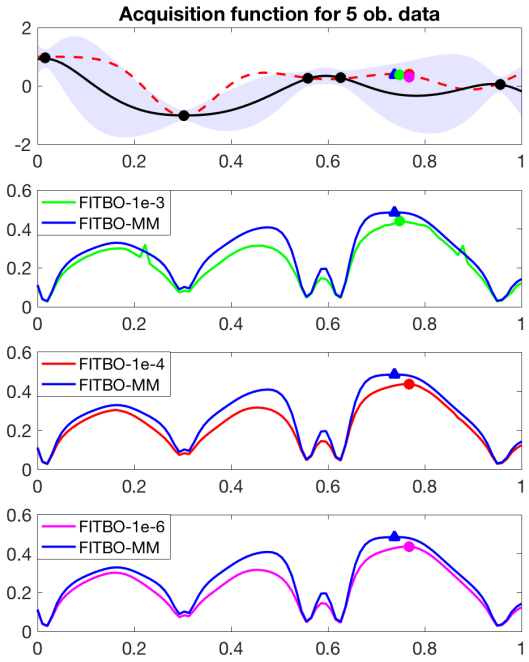

Figure 4. Acquisition functions obtained using numerical integration and moment-matching approximation methods. The top plot shows the objective function (red dotted line), the posterior mean (black solid line) and the 95% confidence interval (blue shaded area) estimated by the Gaussian process model as well as the observation points (black dot). The following three plots show the acquisition function value of FITBO-MM and those of FITBO with different tolerance level (1e-3, 1e-4, 1e-6) used for numerical integration.The next query points are recommended by maximising respective acquisition functions: FITBO-1e-3 (green dot), FITBO-1e-4 (red dot), FITBO-1e-6 (pink dot) and FITBO-MM (blue triangle). The acquisition functions are computed using 600 hyperparameter samples.

References

M. F. Huber, T. Bailey, H. Durrant-Whyte, and U. 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.

J. R. Requeima. Integrated predictive entropy search for Bayesian optimization. 2016.