Eberhard Karls University of Tübingen
Faculty of Mathematics and Natural Sciences
Wilhelm-Schickard-Institut for Computer Science

Bachelor's Thesis Computer Science

Empirical analysis of the re-weighting trick in Bayesian quadrature

Simon Hanrath

21.08.2022

Examiner

Prof. Dr. Philipp Hennig
Wilhelm-Schickard-Institut for Computer Science
University of Tübingen

Thesis Advisor

Dr. Maren Mahsereci
Wilhelm-Schickard-Institut for Computer Science
University of Tübingen

Hanrath, Simon:
Empirical analysis of the re-weighting trick in Bayesian quadrature
Bachelor's Thesis Computer Science
Eberhard Karls University of Tübingen

Abstract

A task that frequently occurs in machine learning is the computation of integrals. These integrals are often intractable, and we must resort to approximation methods. One of these approximation methods is Bayesian quadrature. It seeks to turn the problem of evaluating the integral into a Bayesian inference task. We start with a prior over the integrand and make inferences about it from a set of samples giving the posterior distribution over the integrand. A convenient way of putting priors over the integrand is through a Gaussian process. For some kernel embeddings, the integral over the posterior Gaussian process can be computed analytically. If we want to use Bayesian quadrature for other kernel embeddings, an importance re-weighting trick becomes necessary. Similar to importance sampling, we rewrite the integral by introducing a new probability density. However, the re-weighting trick has not been explored in-depth, and it is unclear if re-weighting affects the performance of Bayesian quadrature. In this thesis, we show that, depending on the new probability density, re-weighting might severely affect the accuracy of Bayesian quadrature. We propose ways of quantifying the expected performance drop and design algorithms to choose parameters for the new probability density in order to minimize the effect of re-weighting. Further, we conduct empirical experiments that suggest that the proposed methods help reduce the potential negative impact of re-weighting on Bayesian quadrature performance.

Acknowledgements

I would like to thank my supervisor Maren Mahsereci, who was always available and helpful whenever I had questions.

Contents

Chapter 1

Introduction

Fitting machine learning algorithms, doing probability queries, summarizing model performance, and training reinforcement learning agents all have in common that they require the calculation of expectations. Calculating expectations over continuous random variables amounts to determining the value of an integral. Integrals are ubiquitous in machine learning and certainly not limited to previously mentioned tasks. The integrals we encounter are often intractable, meaning we can not calculate them exactly and, therefore, must resort to approximation methods.

One method to approximate integrals is Bayesian quadrature. It seeks to turn the problem of calculating the integral into a Bayesian inference task. We start with a prior distribution over the integrand and make inferences about the integrand from a set of samples giving the posterior distribution over the integrand. A convenient way of putting a prior over the integrand is through a Gaussian process. It allows us to incorporate knowledge about properties of our integrand, such as smoothness or continuity. If we condition our Gaussian process prior on evaluations of the integrand, we get a posterior distribution over the integrand. We can then integrate over the Gaussian process posterior instead of the true integrand and get an estimate of the value of the actual integral. Since the Gaussian process gives us a distribution over the integrand, the integral of this model is also a distribution. Bayesian quadrature thus allows us to turn the intractable integration problem into a regression problem on the integrand and an integration problem on the regression model. Depending on the chosen covariance function of the Gaussian process and the integral at hand, the integral over the regression model might be analytical. If we want to use Bayesian quadrature in cases where we do not have an analytical solution, an importance re-weighting trick becomes necessary. Similar to importance sampling, we rewrite the integral by introducing a new probability density. However, the re-weighting trick has not been explored in-depth, and it is unclear if re-weighting affects the performance of Bayesian quadrature.

In this thesis, we will first provide some background on Bayesian quadrature and Gaussian processes. We then introduce the re-weighting trick, provide some context on its relation to importance sampling, and show how it might negatively affect the performance of Bayesian quadrature. We propose ways of quantifying the expected performance drop and design algorithms to choose parameters for the new probability density in order to minimize the potential unfavorable effects of re-weighting. We then evaluate the proposed methods on test integration problems to assess their viability and compare them. Lastly, we will give an outlook on ways to continue or improve this work.

Chapter 2

Background

This chapter is concerned with introducing the basic concepts used throughout this thesis. We will first introduce the concept of Gaussian Processes and then show how we can numerically approximate integrals with Bayesian quadrature. For the introduction to Gaussian processes, we rely on standard textbooks for this topic [12, 16]. For the section on Bayesian quadrature, we rely on work by O'Hagen [14], Rasmussen et al. [15] and the textbook by Hennig et al. [7].

2.1 Gaussian processes

2.1.1 Regression

Regression is a fundamental concept in the field of machine learning. It is a form of supervised learning wherein the algorithm is trained with input features and real-valued output labels. The data, consisting of the input features and output labels, is used to predict the unknown labels of new input features. This is done by establishing a relationship among the variables by estimating how one variable affects the other. So we assume that the data comes from an underlying function, and our goal is to estimate that underlying function by only considering some, possibly noisy, samples from it.

An example of a regression problem would be estimating the price of houses based on properties like the number of bathrooms and age of the house. A regression algorithm would use known examples of houses to infer a relationship between the number of bathrooms and the age of the house with its price, to then make predictions about the prices of other never before seen houses based on the number of bathrooms and their age.

2.1.2 Parametric approaches to regression

In a parametric approach to regression, we try to get as close as possible to the unknown function $f(\mathbf{x})$ underlying the data by choosing a nonlinear function $f(\mathbf{x};\mathbf{w})$, which is parameterized by parameters $\mathbf{w}$. We could for example define $f(\mathbf{x};\mathbf{w})$ by using a set of nonlinear basis functions $\{\phi_h(\mathbf{x})\}_{h=1}^{H}$,

$$f(\mathbf{x};\mathbf{w}) = \sum_{h=1}^{H} w_h \phi_h(\mathbf{x}). \tag{2.1}$$

While $f(\mathbf{x};\mathbf{w})$ is not a linear function of $\mathbf{x}$, it is linearly dependent on the parameters $\mathbf{w}$. We might therefore refer to this as a linear model. Using Bayes' theorem, we can infer the value of the weights $\mathbf{w}$. The posterior probability of the parameters $\mathbf{w}$ is given by

$$P(\mathbf{w}|\mathbf{f}_N, \mathbf{X}_N) = \frac{P(\mathbf{f}_N|\mathbf{w}, \mathbf{X}_N)P(\mathbf{w})}{P(\mathbf{f}_N|\mathbf{X}_N)}. \tag{2.2}$$

Here $\mathbf{X}_N = \{\mathbf{x}_n\}_{n=1}^{N}$ denotes the set of N input vectors and $\mathbf{f}_N = \{f_n\}_{n=1}^{N}$ the corresponding target values. The concrete inference can be realized in different ways. One of them is the Laplace method, where we minimize an objective function

$$M(\mathbf{w}) = -ln(P(\mathbf{f}_N|\mathbf{w}, \mathbf{X}_N)P(\mathbf{w})), \tag{2.3}$$

and thereby locate the locally most probable parameters. Another option we have, would be Markov chain Monte Carlo methods to create samples from the posterior $P(\mathbf{w}|\mathbf{f}_N, \mathbf{X}_N)$. Since we only want to introduce Gaussian Processes, we will not go into further detail about the approaches of parametric regression here.

2.1.3 Introduction to Gaussian process regression

Gaussian Process Regression (GPR) aims to model an unknown function $f(\mathbf{x})$ that underlies the observed data. The adaption of the model to the data corresponds to an inference of the function given the data. By inference, we mean that Bayes' theorem is used to update the probability for a hypothesis as more information becomes available. So if we have a set of input vectors $\mathbf{X}_N$ and a set of corresponding target values $\mathbf{f}_N$ the inference of f($\mathbf{x}$) can be described by

$$P(f(\mathbf{x})|\mathbf{f}_N, \mathbf{X}_N) = \frac{P(\mathbf{f}_N|f(\mathbf{x}), \mathbf{X}_N)P(f(\mathbf{x}))}{P(\mathbf{f}_N|\mathbf{X}_N)}. \tag{2.4}$$

In the case of GPR, the prior, $P(f(\mathbf{x}))$, is a Gaussian process (GP), which encodes some prior beliefs we have about the distribution over the function space. The GP framework allows us to place this prior on $f(\mathbf{x})$ without directly parameterizing it.

2.1.4 Definition of Gaussian processes

To introduce Gaussian Processes, we first look at a regression problem using H fixed basis functions $\{\phi_h(\mathbf{x})\}_{h=1}^{H}$, as we did in section 2.1.2. If we are given N input points $\{\mathbf{x}_n\}_{n=1}^{N}$, we can define the $N \times H$ matrix $\mathbf{R}$ as the matrix of basis function values at the given input points

$$R_{nh} := \phi_h(x_n). \tag{2.5}$$

We can then further define a vector $\mathbf{f}_N$ as the vector of values of $f(\mathbf{x})$ at the given N input points

$$f_n := \sum_{h=1}^{H} R_{nh} w_h. \tag{2.6}$$

If we assume the prior distribution of $\mathbf{w}$ is Gaussian with zero mean

$$P(\mathbf{w}) = \mathcal{N}(\mathbf{w}; \mathbf{0}, \sigma_w^2 \mathbf{I}), \tag{2.7}$$

then $\mathbf{f}$, as a linear function of $\mathbf{w}$, is also Gaussian distributed with zero mean and covariance matrix $\mathbf{Q}$,

$$P(\mathbf{f}) = \mathcal{N}(\mathbf{f}; \mathbf{0}, \mathbf{Q}). \tag{2.8}$$

The covariance matrix $\mathbf{Q}$ is given by

$$\mathbf{Q} = \mathbb{E}(\mathbf{f}\mathbf{f}^{\mathrm{T}}) = \mathbb{E}(\mathbf{R}\mathbf{w}\mathbf{w}^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}) = \mathbf{R}\mathbb{E}(\mathbf{w}\mathbf{w}^{\mathrm{T}})\mathbf{R}^{\mathrm{T}} = \mathbf{R}\sigma_w^2\mathbf{I}\mathbf{R}^{\mathrm{T}} = \sigma_w^2\mathbf{R}\mathbf{R}^{\mathrm{T}}. \tag{2.9}$$

Under the assumption that the prior distribution of $\mathbf{w}$ is Gaussian, no matter what points $\{\mathbf{x}_n\}_{n=1}^{N}$ we select the vector $\mathbf{f}$ always has a Gaussian distribution. This is also the characterizing property of a Gaussian process. The probability distribution over a function $f(\mathbf{x})$ is a Gaussian process, if for any $\{\mathbf{x}_n\}_{n=1}^{N}$ with finite N, the density $P(f(\mathbf{x}_1), f(\mathbf{x}_2), ..., f(\mathbf{x}_N))$ is a Gaussian.

If we now look at a single entry (n, n') of the covariance matrix

$$Q_{nn'} = (\sigma_w^2 \mathbf{R}\mathbf{R}^T)_{nn'} = \sigma_w^2 \sum_{h}^{H} \phi_h(\mathbf{x}_n)\phi_h(\mathbf{x}_{n'}), \tag{2.10}$$

we see that we have to compute a sum for each entry of the covariance matrix of the prior distribution of $\mathbf{f}$. In mathematics, certain sums remain tractable even if the number of entries of the sum goes to infinity. By computing the integral, we can use this circumstance to add infinitely many features for each entry of the covariance function. This, however, does not work for arbitrary basis functions. We have to specifically choose certain basis functions and place them in a regular fashion. To illustrate this, we have a look at an example using one-dimensional radial basis functions

$$\phi(x)_h = \exp\left(-\frac{(x-h)^2}{2r^2}\right). \tag{2.11}$$

We can now take the limit H → ∞ and thereby turn the sum over h into an Integral. We scale σ_w^2 as $\frac{S}{\Delta H}$ to prevent our covariance from diverging with H. Here S is a constant, and ΔH denotes the number of basis functions per unit length. A single entry of the covariance function can now be described as

$$\begin{aligned}
Q_{nn'} &= S\int_{h_{min}}^{h_{max}} \phi_h(x_n)\phi_h(x_{n'})dh \\
&= S\int_{h_{min}}^{h_{max}} \exp\left(-\frac{(x_n-h)^2}{2r^2}\right)\exp\left(-\frac{(x_{n'}-h)^2}{2r^2}\right)dh.
\end{aligned} \tag{2.12}$$

This integral is analytically solvable if we let $h_{max} \rightarrow \infty$ and $h_{min} \rightarrow -\infty$ and we get

$$Q_{nn'} = \sqrt{\pi r^2} S \exp\left(-\frac{(x_n - x_{n'})^2}{4r^2}\right). \tag{2.13}$$

This new form enables us to re-frame our view on the regression problem. Whereas before, we had to specify the prior distribution using finitely many basis functions and priors on parameters, we now can specify this distribution using the covariance function,

$$C(x_n, x_{n'}) = \theta \exp\left(-\frac{(x_n - x_{n'})^2}{4r^2}\right). \tag{2.14}$$

Many different covariance functions are possible. The one we have just constructed is called the squared exponential covariance function. The only constraint we have is that the chosen covariance function must generate a positive semidefinite covariance matrix for any set of input points $\{\mathbf{x}_n\}_{n=1}^N$. A matrix is positive semidefinite if all its eigenvalues are non-negative. By choosing different covariance functions and different hyperparameters, we specify prior beliefs about the distribution over the function space. In the case of the squared exponential covariance function, we encode the belief that the function should

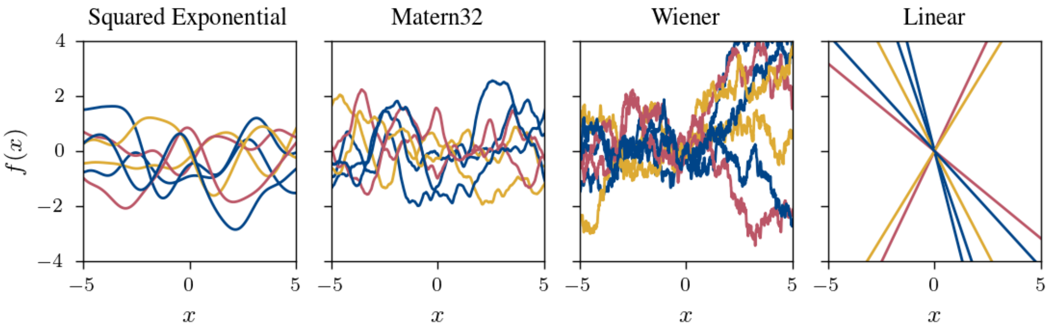

Figure 2.1: Samples of a Gaussian process with different covariance functions.

be infinitely differentiable. Figure 2.1 shows samples from Gaussian processes with different covariance functions. The formulas for the covariance functions of the GPs from which these samples where taken as well as more examples of covariance functions can be found in the well known Kernel Cookbook by David Duvenaud [5].

We have seen that a GP defines a prior over functions using a covariance function. Additionally a GP requires a mean function. However, the mean function is usually just assumed to be the zero function. This is because one can transform the original problem so that a zero mean function is appropriate. If we would choose a non-zero mean function $m(\mathbf{x})$, we can also model the function $h(\mathbf{x}) = f(\mathbf{x}) - m(\mathbf{x})$ instead of the function $f(\mathbf{x})$, in which case a zero mean function is suitable again. There are a few settings where $m(\mathbf{x}) = 0$ does not apply w.l.o.g., but these are not relevant for us for the time being. We can thus write,

$$f \sim \mathcal{GP}(\mathbf{0}, C(\mathbf{x}, \mathbf{x'})) \tag{2.15}$$

and thereby define the GP.

2.1.5 Gaussian Processes inference

We are now concerned with fitting the GP to data. This is done by incorporating the information the training data provides about the true function into our prior GP to form the posterior GP. In the case of noise-free observations, we know that the true function goes through the training data points and can therefore restrict our joint prior distribution to contain only functions which agree with the observed training data. An example of this can be seen in Figure 2.2. If we have observations $\{(\mathbf{x}_n, f_n)\}_{n=1}^{N}$, the joint distribution of the training outputs, $\mathbf{f}$, and the test outputs $\mathbf{f}_*$ according to our prior GP is

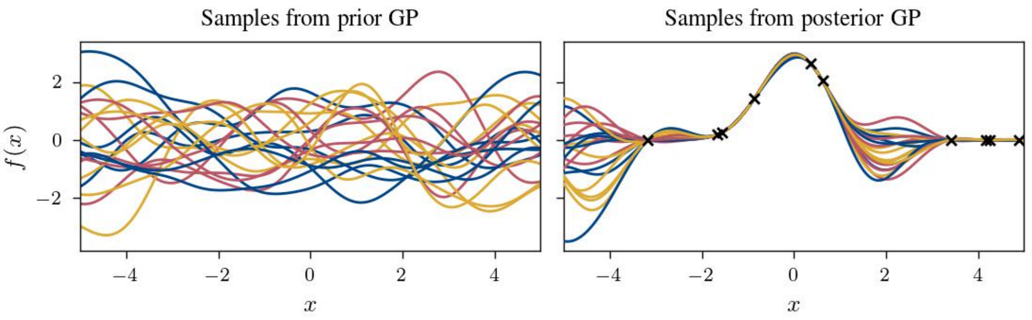

Figure 2.2: Samples of a Gaussian process with squared exponential covariance function before and after it has been fitted to the data.

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \begin{bmatrix} C(X_N, X) & C(X_N, X_{N*}) \\ C(X_{N*}, X_N) & C(X_{N*}, X_{N*}) \end{bmatrix}). \tag{2.16}$$

If there are N training points and N_* test points, then $C(X_N, X_{N*})$ denotes the $N \times N_*$ matrix of covariances for all pairs of test points and training points, the same applies analogously for for $C(X_N, X)$, $C(X_{N*}, X)$ and $C(X_{N*}, X_{N*})$. We could now generate functions from the prior and reject all that do not agree with the data points. Luckily we do not need to resent to such a computationally inefficient method. Using properties of the normal distribution, we can compute the posterior mean function and the posterior covariance function given by

$$\begin{aligned} \tilde{m}(x) &= C(x, X_N)C(X_N, X_N)^{-1}\mathbf{f} \\ \tilde{C}(x, x') &= C(x, x') - C(x, X_N)C(X_N, X_N)^{-1}C(X_N, x') \end{aligned} \tag{2.17}$$

and thus obtain the joint posterior distribution conditioned on the observations

$$\mathbf{f}_*|X, \mathbf{f} \sim \mathcal{N}(\tilde{m}(x), \tilde{C}(x, x')). \tag{2.18}$$

Depending on the context, we may only have access to noisy observations of the function values. If we consider Gaussian observations of the true function $f(\mathbf{x})$ of the form

$$y(\mathbf{x}) = f(\mathbf{x}) + \epsilon, \quad \epsilon = \mathcal{N}(0, \sigma_n^2). \tag{2.19}$$

we can rewrite the equations for the posterior covariance function and the posterior mean function to incorporate the noise:

$$\begin{aligned} \tilde{m}(x) &= C(x, X_N)\left(C(X_N, X_N) + \sigma_n^2\mathbf{I}\right)^{-1}\mathbf{y} \\ \tilde{C}(x, x') &= C(x, x') - C(x, X_N)\left(C(X_N, X_N) + \sigma_n^2\mathbf{I}\right)^{-1}C(X_N, x'). \end{aligned} \tag{2.20}$$

Hyperparameter optimization

In section 2.1.4 we briefly mentioned the hyperparameters that a covariance function might have. In our example of the squared exponential covariance function, the hyperparameters define the length scale and amplitude of the GP. These hyperparameters may significantly influence the GP's performance, and it is therefore desirable to choose them well. Various methods for selecting suitable hyperparameters such as maximum likelihood, maximum marginal likelihood, or maximum a posterior exist. For the experiments in this thesis, we will use the maximum likelihood to optimize the hyperparameters. The likelihood of a data set is the probability of obtaining that particular data set given the chosen model. This expression includes the unknown parameters. The values of the parameters that maximize the probability of the sample are called maximum likelihood estimates. In our case, we are interested in obtaining a maximum likelihood estimate for the hyperparameters $\boldsymbol{\theta}$ of the covariance function,

$$\arg\min_{\boldsymbol{\theta}} -\log(p(\mathbf{f}|X, \boldsymbol{\theta})). \tag{2.21}$$

The optimization of this function is done by the Broyden–Fletcher–Goldfarb–Shanno (BFGS) optimization algorithm [3]. The BFGS algorithm is an iterative quasi newton method for solving nonlinear optimization problems. It determines the descent direction by preconditioning the gradient with curvature information.

2.2 Bayesian quadrature

2.2.1 The problem of symbolic integration

Calculating integrals symbolically is not an easy task. Derivatives, on the other hand, are usually much easier to obtain. Nevertheless, integrals and derivatives are just opposites of each other. They are inverse operations. And they have the same rules. What about these rules makes this possible? Put simply, differentiation is a forward operation. We can apply the rules mechanically to get from our function to its derivative. This can be done by recursively applying the known rules to subterms of the function to get to the derivative of the whole function. Integration, on the other hand, is an inverse problem. Of course, there are analogues rules for integration, but here it is not immediately obvious when to apply them. For example, the well-known integration by parts formula,

$$\int f(x)g'(x)dx = f(x)g(x) - \int g(x)f'(x)dx, \tag{2.22}$$

is only useful if $f'(x)\int g(x)dx$ or $g'(x)\int f(x)dx$ is easier to integrate than $f(x)g(x)$. It is generally not obvious whether this is the case. During integration, one has to recognize patterns and even introduce substitutions to bring the expression into the desired form. This requires a lot of practice and intuition. Besides this difficulty, there is also the problem that we do not necessarily have access to a formal description of the function. In some cases, we can only take individual samples from this unknown black box function, sometimes with great computational effort for each sample. Luckily, in practice, integrals do not have to be computed symbolically. Many different numerical approximation methods exist.

2.2.2 Numerical quadrature

The term numerical quadrature represents many different algorithms that aim to approximate the numerical value of a definite integral,

$$\int_{\Omega} f(\mathbf{x})d\mathbf{x}. \tag{2.23}$$

Numerical quadrature methods usually evaluate the function we wish to integrate at specific points to infer the integral value. Since it can be expensive to evaluate the function, the goal is often to obtain as accurate an estimate as possible with as few samples as possible. For example, we could define an interpolating function that is easy to integrate and use it to estimate the integral between each two samples of our function. The simplest choice would be a constant function that passes through the point $(\frac{\mathbf{x}_i+\mathbf{x}_{i+1}}{2}, f(\frac{\mathbf{x}_i+\mathbf{x}_{i+1}}{2}))$, where $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ denote the x-coordinates of two neighbouring samples. So the approximation is calculated by dividing the region into rectangles that form a region similar to the measured region, then calculating the area for each of these rectangles, and finally adding up all these areas. This method of approximation is also known as the midpoint rule or the middle Riemann sum. Of course, many more interpolation functions are possible, or even completely different approaches that are not based on interpolation functions.

2.2.3 Monte Carlo estimators

To solve an intractable integral, practitioners often refer to Monte Carlo (MC) estimators for the integral value F. Such an estimator is denoted as $\hat{F}_{MC}$ and has the form of a weighted sum,

$$\hat{F}_{MC} = \sum_{i=1}^{T} w_i f(x_i) \approx \int_{\Omega} f(x)p(x)\mathrm{d}x = \mathbb{E}_p\left(f(x)\right) = F, \tag{2.24}$$

with equal weights $w_i = \frac{1}{N}$ and $x_i \sim p(x)$ i.i.d. from p in its standard form. It is to be noted that point selection (the retrieval of the x_i) is not arbitrary in Monte Carlo estimation and must follow the random scheme as described in order for the MC estimator to have favorable properties. We know that this approximation converges to the right answer as $T \to \infty$. Monte Carlo estimators are also unbiased and easily implemented. However, the variance of Monte Carlo estimators decreases linear with T and they therefore require many samples to provide an answer with satisfactory accuracy. Depending on the integrand, this might make a reasonable approximation expensive.

2.2.4 Bayesian quadrature

Unlike the Monte Carlo method, Bayesian Quadrature (BQ) correlates samples by using assumptions about our integrand, such as smoothness or continuity. It is a statistical approach to the numerical problem of computing integrals and a subarea of Probabilistic Numerics. BQ tries to determine the value of the integral F by a number of evaluations of the integrand f at the points $\{x_n\}_{n=1}^{N}$. These samples are used to emulate the integrand by a surrogate model, over which we then calculate the integral. Here it is important to note that the integral of the surrogate model should be easier to calculate than the true integral. A convenient choice of surrogate models are the earlier discussed Gaussian Processes. They allow us to directly encode our assumptions about the integrand by choosing different covariance functions for the GP. Depending on the chosen covariance function of the GP and the integral at hand, the integral over the regression model might be analytical. We can hence shift a possibly challenging integration problem to a regression problem on the integrand and an easier, often analytic integration problem of the regression model. Unlike the numerical quadrature methods presented earlier, BQ not only provides an approximation of the integral value. It also indicates how certain it is about this value by returning a distribution over the integral value. This is due to the fact that the surrogate model itself is a distribution. In the case of a GP as a surrogate model for our integrand, we get a Gaussian distribution over the integral value because f is Gaussian distributed, and the integral is a linear operation under which Gaussians are closed. So if we use a GP with mean function $m : \Omega \to \mathbb{R}, \mathbf{x} \mapsto m(\mathbf{x})$ and covariance function $C : \Omega \times \Omega \to \mathbb{R}, (\mathbf{x}, \mathbf{x}') \mapsto C(\mathbf{x}, \mathbf{x}')$, the distribution over the integral value of F is given by

$$F_{GP} \sim \mathcal{N}(\mu, \sigma). \tag{2.25}$$

Where the univariate mean μ is

$$\begin{aligned}\mu &:= \mathbb{E}_{\mathcal{GP}}(F)\\ &= \mathbb{E}_{\mathcal{GP}}\left(\int_\Omega f(\mathbf{x})p(\mathbf{x})d\mathbf{x}\right)\\ &= \int_\Omega \mathbb{E}_{\mathcal{GP}}\left(f(\mathbf{x})\right)p(\mathbf{x})\,dx\\ &= \int_\Omega m(\mathbf{x})p(\mathbf{x})d\mathbf{x}\end{aligned} \tag{2.26}$$

and the variance σ is

$$\begin{aligned}\sigma &:= \mathrm{var}(F)\\ &= \mathbb{E}_{\mathcal{GP}}\left(\int_\Omega f(\mathbf{x})p(\mathbf{x})d\mathbf{x}\int_\Omega f(\mathbf{x'})p(\mathbf{x'})d\mathbf{x'}\right) - \left(\mathbb{E}_{\mathcal{GP}}\left(\int_\Omega f(\mathbf{x})p(\mathbf{x})d\mathbf{x}\right)\right)^2\\ &= \int_\Omega\int_\Omega \mathbb{E}_{\mathcal{GP}}\left(f(\mathbf{x})f(\mathbf{x'})\right)p(\mathbf{x})\,p(\mathbf{x'})d\mathbf{x}d\mathbf{x'} - \mathtt{m}^2\\ &= \int_\Omega\int_\Omega C(\mathbf{x},\mathbf{x'})p(\mathbf{x})p(\mathbf{x'})d\mathbf{x}d\mathbf{x'} + \int_\Omega\int_\Omega m(\mathbf{x})m(\mathbf{x'})p(\mathbf{x})p(\mathbf{x'})d\mathbf{x}d\mathbf{x'} - \mathtt{m}^2\\ &= \int_\Omega\int_\Omega C(\mathbf{x},\mathbf{x'})p(\mathbf{x})p(\mathbf{x'})d\mathbf{x}d\mathbf{x'} + (\int_\Omega m(\mathbf{x})p(\mathbf{x})d\mathbf{x})^2 - \mathtt{m}^2\\ &= \int_\Omega\int_\Omega C(\mathbf{x},\mathbf{x'})p(\mathbf{x})p(\mathbf{x'})d\mathbf{x}d\mathbf{x'}.\end{aligned} \tag{2.27}$$

. If we now sample points $\mathcal{D} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}$ from f and update the mean function and covariance function of our GP as discussed in section 2.1.5, we get the posterior GP $f_{\mathcal{D}} \sim \mathcal{N}(\tilde{m}(\mathbf{x}), \tilde{C}(\mathbf{x}, \mathbf{x'}))$. When we apply formulas 2.26 and 2.27, the mean $\mu_{\mathcal{D}}$ of the posterior Gaussian distribution over the Integral is given by

$$\mu_{\mathcal{D}} = \mu + \left(\int_\Omega C(\mathbf{x}, X_N)p(\mathbf{x})d\mathbf{x}\right)\left(C(X_N, X_N) + \sigma_n^2\mathbf{I}\right)^{-1}\mathbf{f} \tag{2.28}$$

and the variance by

$$\sigma_{\mathcal{D}} = \sigma - \left(\int_\Omega C(\mathbf{x}, X_N)p(\mathbf{x})d\mathbf{x}\right)\left(C(X_N, X_N) + \sigma_n^2\mathbf{I}\right)^{-1}\left(\int_\Omega C(\mathbf{x'}, X_N)p(\mathbf{x'})d\mathbf{x'}\right)^T. \tag{2.29}$$

In general, these expressions may be difficult to evaluate, but several interesting special cases for which we can obtain analytical expressions exist.

Chapter 3

The re-weighting trick in Bayesian quadrature

As we have seen, BQ enables us to shift the problem of evaluating a difficult integral to a regression problem on the integrand and an often analytical integration problem on the regression model (usually a GP). The integration problem on the GP model is only analytical for some combinations of covariance functions and probability densities p. Most common covariance functions can be analytically integrated against the uniform density, which assigns equal probability to all values in its interval and the Gaussian density. If we wish to use BQ to integrate with respect to other probability densities, for which we do not have an analytical solution, we have to make use of the re-weighting trick,

$$\begin{aligned}\int_{\Omega} f(\mathbf{x})p(\mathbf{x})d\mathbf{x} &= \int_{\Omega} \frac{f(\mathbf{x})p(\mathbf{x})}{q(\mathbf{x})} q(\mathbf{x})d\mathbf{x} \\ &:= \int_{\Omega} g(\mathbf{x})q(\mathbf{x})d\mathbf{x},\end{aligned} \tag{3.1}$$

where we rewrite the original integration problem by introducing a new probability density q [15]. This new probability density q can be chosen arbitrarily as long as $f(x)p(x) = 0$ whenever $q(x) = 0$ holds true. Our Gaussian process now models the new integrand $g(\mathbf{x}) = f(\mathbf{x})p(\mathbf{x})/q(\mathbf{x})$ and we integrate against the probability density q. If we now choose q to be the Gaussian or uniform density, we can use BQ to integrate against any probability density p that can be evaluated.

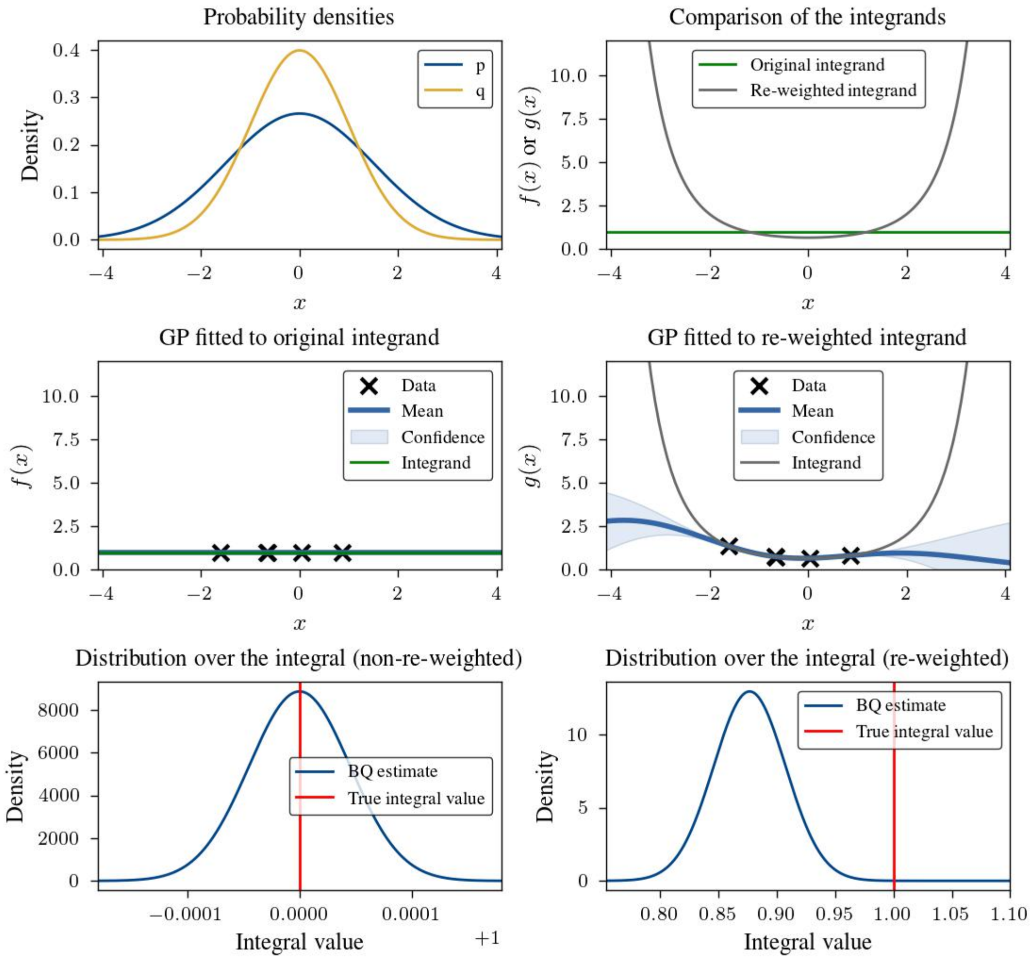

Figure 3.1: A simple example of the negative effects re-weighting might have on the performance of BQ. The integral $\int f(x)p(x)dx$ where $f(x) = 1$ and $p(x) = \mathcal{N}(x; 0, 1.5)$ has been re-weighted with $q(x) = \mathcal{N}(x; 0, 1)$. The first plot compares the probability densities p and q. The second plot shows the distortion of the original integrand caused by re-weighting it with q. The next two plots show the posterior GP for the original integrand and the re-weighted integrand. In the last two plots, we see that re-weighting can significantly decrease BQ performance even for these relatively similar probability densities.

3.1 Drawbacks of re-weighting

We do not change the value of the integral F when re-weighting. It, therefore, appears reasonable to use any probability density for q as long as we can integrate against it. However, while we do not change the value of our integral F when re-weighting, the new integrand g might be distorted negatively and therefore become significantly harder to capture for the given GP. This may lead to a less accurate estimation by the GP and, therefore, a less accurate estimation of the integral F. While both re-weighting with a Gaussian density and re-weighting with a uniform density might distort the original integrand, the distortion of re-weighting with a uniform density is relatively independent of the choice of parameters for the probability density q. The new integrand will always be $g(\mathbf{x}) = \frac{1}{k} f(\mathbf{x}) p(\mathbf{x})$, for any uniform density. The only thing we could influence by selecting different parameters for our uniform density would be the constant k, and scaling of the function we want to capture does not affect the performance of the GP. However, when re-weighting with a Gaussian density, the distortion of the new integrand g is very sensitive to the choice of parameters. An example can be seen in Figure 3.1. We see that the parameterization of q matters for the accuracy of the BQ estimate. This is often ignored in practice. Our goal is to find a metric that allows us to predict this performance drop to then, in a further step, choose a parameterization of q that minimizes the metric and thus the performance drop.

3.2 Relation to optimal importance sampling

Importance sampling has some structural similarities to the re-weighing trick in BQ in the sense that another distribution q is used instead of p [9]. To provide some context, we briefly introduce importance sampling here, its optimal distribution q^*_{IS}, and discuss relations to the re-weighing trick in BQ.

3.2.1 Integral estimation with importance sampling

Importance sampling (IS) is a specific form of the Monte Carlo estimation, described in section 2.2.3, that circumvents mainly either of the following problems: i) It is not possible to sample from $p(x)$ directly, or ii) The variance of $\hat{F}_{MC}$ is large or unbounded, and we would like to find an estimator whose variance is smaller or in fact bounded. An alternative MC estimator with an alternative sampling distribution may be used in these scenarios. This is the importance sampling estimator $\hat{F}_{IS}$.

The idea of importance sampling is similar in structure to the re-weighing trick in BQ. In Monte Carlo, $p(x)$ is called the "nominal" or "target" distribution, and another distribution with density $q(x)$ is introduced which is called the "importance" or "proposal" distribution. Analogously to the above, we can rewrite the original integration problem as

$$\begin{aligned} F = \mathbb{E}_p\left(f(x)\right) = \int_\Omega f(x)p(x)\mathrm{d}x &= \int_\Omega \frac{f(x)p(x)}{q(x)}q(x)\mathrm{d}x \\ &= \int_\Omega g(x)q(x)\mathrm{d}x \\ &= \mathbb{E}_q\left(g(x)\right), \end{aligned} \tag{3.2}$$

where we require that $f(x)p(x) = 0$ whenever $q(x) = 0$. Importance sampling now says that we can simply use the standard MC estimator on g if we sample the $x_i \sim q(x)$ from q instead of from p, that is

$$\hat{F}_{IS} = \sum_{i=1}^{N} w_i g(x_i), \quad \text{with} \quad w_i = \frac{1}{N}. \tag{3.3}$$

However, the importance sampling estimator $\hat{F}_{IS}$ may not have the same properties as the standard Monte Carlo estimator. For example, if the ratio $p(x)/q(x)$ is very large in some areas, the variance of $\hat{F}_{IS}$ may not be finite. This means that even for large N, $\hat{F}_{IS}$ may not be reliable. It is, therefore, essential to select q well.

3.2.2 Optimal proposal distribution for importance sampling

The importance sampling estimator $\hat{F}_{IS}$ is still unbiased,

$$\mathbb{E}_q\left(\hat{z}_{IS}\right) = \mathbb{E}_q\left(\frac{1}{N}\sum_{i=1}^{N} g^*(x_i)\right) = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_q\left(g^*(x_i)\right) = \frac{1}{N}\sum_{i=1}^{N} z^* = z^* \tag{3.4}$$

and its variance is given by,

$$\begin{aligned} \mathrm{Var}_q(\hat{z}_{IS}) &= \frac{1}{N}\mathrm{Var}_q(g^*(x)) \\ &= \frac{1}{N}\left(\mathbb{E}_q((g^*(x))^2) - (z^*)^2\right) \\ &= \frac{1}{N}\left(\int_\Omega \left(\frac{f^*(x)p(x)}{q(x)}\right)^2 q(x)\mathrm{d}x - z^*\int_\Omega f^*(x)p(x)\mathrm{d}x\right) \\ &= \frac{1}{N}\left(\int_\Omega (f^*(x)p(x) - z^*q(x))^2 q^{-1}(x)\mathrm{d}x\right). \end{aligned} \tag{3.5}$$

In practice, we usually only have access to one sample of the importance sampling estimator $\hat{F}_{IS}$. We are therefore interested in choosing our proposal distribution such that the variance of $\hat{F}_{IS}$ is minimized in order to increase our chances of getting an estimate that is close to the actual integral value.

From Eq. 3.5 we see that if $f^*(x)$ is positive $\mathrm{Var}_q(\hat{z}_{IS})$ is zero if $q^*_{IS}(x) = \frac{f^*(x)p(x)}{z^*}$. If f^* is not positive everywhere, it can be shown that the distribution $\bar{q}(x) = c^{-1}|f^*(x)|p(x)$ where c is the normalization constant of $\bar{q}$, minimizes the variance $\mathrm{Var}_q(\hat{z}_{IS})$, and is hence optimal.

Discussion of optimal importance sampling distribution

Unfortunately $q^*_{IS} = \bar{q}$ is not easily accessible since the normalization constant $c = \int_\Omega |f^*(x)|p(x)\mathrm{d}x$ is often as hard to compute as the original integration problem. Nevertheless, let us see what effect q^*_{IS} has on the integrand. In other words, let us look at the weighted integrand $g^*(x)$.

If $f^*(x)$ is positive everywhere, then $q^*_{IS}(x) = c^{-1}f^*(x)p(x)$ and $c = z^*$ and therefore

$$g^*(x) = \frac{f^*(x)p(x)}{q^*_{IS}} = z^*. \tag{3.6}$$

Hence, the re-weighted function $g^*(x)$ is constant at the precise value of the integral. The IS estimator would be done after one evaluation of $g^*(x)$; therefore, its variance is zero, as shown. If $f^*(x)$ is not necessarily positive everywhere, then $q^*_{IS}(x) = c^{-1}|f^*(x)|p(x)$ and $c = \int_\Omega |f^*(x)|p(x)\mathrm{d}x$ and therefore

$$g^*(x) = \frac{f^*(x)p(x)}{q^*_{IS}} = c\frac{f^*(x)p(x)}{|f^*(x)|p(x)} = c\,\mathrm{sign}(f(x)). \tag{3.7}$$

Hence, the re-weighted function $g^*(x)$ is piece-wise constant at the values $\pm c = \pm\int_\Omega |f^*(x)|p(x)\mathrm{d}x$. The IS estimation in this optimal case is therefore equivalent to estimating the probability $\tilde{p}$ to get "heads" when flipping a coin; this Bernoulli estimator has the well-known variance $\frac{1}{N}\tilde{p}(1-\tilde{p})$ after having thrown the coin N times. Here $\tilde{p}$ is the probability of $f(x_i)$ being positive when x_i are sampled from q^*_{IS}.

It should be noted that in practice, and especially in high dimensions, it is challenging to construct a q that would approximate $\bar{q}$ in a meaningful way and would improve the properties of the estimator $\hat{z}_{IS}$. It is equally possible that the properties of $\hat{z}_{IS}$ become worse and its variance unbounded when using a q other than the (intractable) $\bar{q}$.

3.2.3 Comparison of importance sampling to re-weighting trick in BQ

We see that importance sampling has some structural similarities to the re-weighing trick in BQ. However, the two do not necessarily have similar goals. In importance sampling, we aim to choose q such that the variance of our distribution $\hat{F}_{MC}$ is reduced. The re-weighting trick in Bayesian quadrature aims to replace the density p with a density q for which we can obtain analytical results for the integral over the surrogate model. We have seen that depending on the choice of q, the re-weighting trick might negatively affect the performance of Bayesian quadrature as it can distort the integrand in unfavorable ways. It is unclear whether the idea of q^*_{IS} can be of use for the re-weighting trick in BQ. However, it seems challenging for a GP to model piece-wise constant, non-continuous functions, hence g^* as a result of q^*_{IS} seems not well behaved for a GP, and q^*_{IS} rather seems tailored to and optimal for an MC estimator that has arguably the least hard to time estimate a constant function. We will, therefore, not further consider importance sampling in this thesis.

Chapter 4

Performance drop after re-weighting

As we have seen, depending on the new probability density q, there may be a more or less pronounced drop in the performance of our BQ algorithm. In this chapter, we will discuss different approaches to predict the performance drop and construct empirical scores that follow these approaches.

4.1 Scores for measuring performance drop after re-weighting

4.1.1 Similarity of outcome

A desirable metric to minimize the effect of the re-weighting on the estimation of our BQ algorithm would capture the difference between the Gaussian distribution over the integral before re-weighting and the Gaussian distribution over the integral after applying the re-weighting trick. Minimizing this metric could ensure that the re-weighting has as little effect as possible. We thus get:

$$\arg\max_{\theta} \text{similarity}(F_1, F_2), \tag{4.1}$$

where

$$\begin{aligned} F_1 &\sim \mathcal{N}\left(\int_\Omega m_1(\mathbf{x})p(\mathbf{x})d\mathbf{x}, \int_\Omega\int_\Omega C_1(\mathbf{x},\mathbf{x'})p(\mathbf{x})p(\mathbf{x'})d\mathbf{x}d\mathbf{x'}\right), \\ F_2 &\sim \mathcal{N}\left(\int_\Omega m_2(\mathbf{x})q(\mathbf{x};\theta)d\mathbf{x}, \int_\Omega\int_\Omega C_2(\mathbf{x},\mathbf{x'})q(\mathbf{x};\theta)q(\mathbf{x'};\theta)d\mathbf{x}d\mathbf{x'}\right) \end{aligned} \tag{4.2}$$

denote the distribution over the integral before and after re-weighting with the probability density q. Unfortunately, we usually cannot calculate the distribution F_1 because we typically use the re-weighting trick when we cannot use

BQ to integrate over the probability distribution p. So even though this score is useful, in practice, we can only use it in situations where re-weighting is not necessary.

4.1.2 Distortion of the integrand

As we have seen, the direct comparison between the Gaussian distributions over the integral values is impossible in most cases. However, if we assume that the reason for the performance drop is the distortion of the integrand f, we can use the similarity of the integrand f and the new integrand g as a proxy. The idea here is that due to the distortion, the properties of the integrand change, and thus it can no longer be guaranteed that the chosen Gaussian process can approximate it well. f might, for example, be stationary, but g does not have to be stationary as well. When our Gaussian process assumes stationarity, we have to expect a performance drop for the re-weighted estimation. Therefore, defining a score that indicates to what extent properties of f still apply g would be desirable. One approach is to measure the similarity of p and q to then choose parameters θ for q in order to maximize the similarity

$$\arg\max_{\theta} \text{similarity}(q(\mathbf{x}, \theta), p(\mathbf{x})). \tag{4.3}$$

However, this is not the only plausible option. While it guarantees us $f(\mathbf{x}) = g(\mathbf{x})$ and therefore no distortion, if it is possible to choose θ such that $p(\mathbf{x}) = q(\mathbf{x}|\theta)$, this is most often not possible. When we can not find a θ such that $p(\mathbf{x}) = q(\mathbf{x}|\theta)$, we have to define what "most similar" means exactly and have no guarantee that our definition of most similar ensures that the previous properties of f still apply to g.
Therefore, it might also be viable to think about the properties that our GP assumes in the integrand and then try to define a Score that measures to what extent these properties still hold after re-weighting.

4.1.3 Suitability of the integrand

In the above section, we argued for choosing a q such that the original integrand f and the new integrand g are as similar as possible. However, our goal is not necessarily to get the same results in the re-weighted case as in the non-re-weighted case but to get the best approximation of the integral F. Therefore, we might try to distort the original integrand f so that our GP easily captures the new integrand g. It is therefore desirable to define a score that measures how well our Gaussian process can capture our new integrand g. This could, for example, be achieved by measuring how probable it is to observe samples from g under a given Gaussian Process.

4.2 Empirical scores assessing the re-weighting trick

In this section, we will construct empirical scores based on the approaches mentioned above.

4.2.1 Similarity of p and q

As discussed in section 4.1.2, a possibly interesting metric would capture the similarity of the density p and the newly introduced density q. When the two are similar, we might expect the distortion of the new integrand to be relatively small and therefore expect that the GP still captures the integrand well. However, it is not obvious how exactly the similarity of the two densities should be measured in order to minimize the distortion of the original integrand. Some possible scores for determining the similarity of two probability densities are shown below.

Manhattan distance

One way of quantifying the similarity of two probability densities is to take the Manhattan distance between the two distributions. The Manhattan distance measures the area between the distributions and is defined as

$$\mathrm{Man}(p, q) := \frac{1}{2} \int |p(\mathbf{x}) - q(\mathbf{x})| d\mathbf{x}. \tag{4.4}$$

This score works for arbitrary probability densities and can easily be extended into higher dimensions.

Kolmogorov-Smirnov score

The Kolmogorov-Smirnov score can be used to quantify the equality of two probability distributions [17]. This is achieved by using the largest difference between the cumulative distribution functions of p and q as an indicator of the dissimilarity of the two samples. We thus get

$$\mathrm{KS}(p, q) := \max_{\mathbf{x}} |cdf_p(\mathbf{x}) - cdf_q(\mathbf{x})|. \tag{4.5}$$

The Kolmogorov-Smirnov score may be a valuable method for comparing two samples, as it is sensitive to differences in both location and shape of the cumulative distribution functions of the two samples. We do, however, have to compute the cumulative distribution functions in order to calculate this score.

Wasserstein distance

The Wasserstein distance is a distance function defined between probability distributions [8]. Intuitively, if each distribution is viewed as a unit amount of earth, the metric is the minimum ”cost” of turning one pile into the other, which is assumed to be the amount of earth that needs to be moved times the mean distance it has to be moved. The first Wasserstein distance between the distributions p and q is defined as

$$l_1(p,q) := \inf_{\pi \in \Gamma(p,q)} \int_{\mathbb{R}\times\mathbb{R}} |x-y| d\pi(x,y) \tag{4.6}$$

Here Γ(p,q) is the set of probability distributions on $\mathbb{R}\times\mathbb{R}$ whose marginals are p and q on the first and second factors, respectively. Or intuitively speaking: the set off all ways of moving the one pile of earth into the other. If cdf_p and cdf_q are the respective CDFs of the one dimensional functions p and q, this distance is also equal to:

$$l_1(p,q) = \int |cdf_p(x) - cdf_q(x)| dx \tag{4.7}$$

which is the previously mentioned Manhattan distance for the CDFs of p and q. This might, however, be useful since the Manhattan distance of the PDFs does not change much for two distributions with minimal overlap. The Wasserstein distance would clearly display the difference. For other cases, the Wasserstein distance cannot necessarily be computed as easily.

Kullback Leibler divergence

The Kullback Leibler (KL) divergence measures how one probability density differs from the reference probability density [10]. Here we measure the distance between p and q by looking at how likely the second distribution will be able to generate samples from the first distribution. By this definition, the KL divergence is not a distance metric as it is not symmetric. $KL(q||p)$ is not necessarily equivalent to $KL(p||q)$. It might, however, still be useful to capture the similarity of the two distributions. The KL divergence of p and q is given by

$$\begin{aligned} KL(p||q) &:= \int p(\mathbf{x}) log\left(\frac{1}{q(\mathbf{x})}\right) d\mathbf{x} - \int p(\mathbf{x}) \log\left(\frac{1}{p(\mathbf{x})}\right) d\mathbf{x} \\ &= \int p(\mathbf{x}) \left(\log\left(\frac{1}{q(\mathbf{x})}\right) d\mathbf{x} - \log\left(\frac{1}{p(\mathbf{x})}\right)\right) d\mathbf{x} \\ &= \int p(\mathbf{x}) \log\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right) d\mathbf{x}. \end{aligned} \tag{4.8}$$

We see that we have to calculate an integral. For some probability distributions, this integral can be calculated analytically. For example the KL divergence between a one dimensional Gaussian density $q(x|\mu_q, \sigma_q)$ and a one dimensional Gaussian mixture density $p(x|\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p, \mathbf{w})$is given by,

$$KL(p||q) = \sum_{i=1}^{I} \mathbf{w}_i \left(\frac{1}{2} log(\frac{\sigma_q}{\boldsymbol{\sigma}_{pi}}) + \frac{\boldsymbol{\sigma}_{pi} + (\boldsymbol{\mu}_{pi} - \mu_q)^2}{2\sigma_q^2} - \frac{1}{2} \right). \tag{4.9}$$

In cases where we cannot calculate the integral analytically, we could approximate the integral by Monte Carlo methods. This is still less expensive than MC on the original integral since evaluating the densities is usually cheap, while evaluations on the integrand are usually expensive.

Scoping experiments

It is not apparent which of the above scores is the most suitable to assess the performance drop of our BQ algorithm when re-weighting with a given q. For pre-selection, we make a quick empirical comparison between the different scores and the actual performance of BQ on the re-weighted integral. In figure 4.1, we see the values of the scores evaluated on different parameterizations of a Gaussian density q. Based on these empirical results and the fact that it is relatively easy to calculate, we decide to use the KL divergence as a measure of the similarity between p and q.

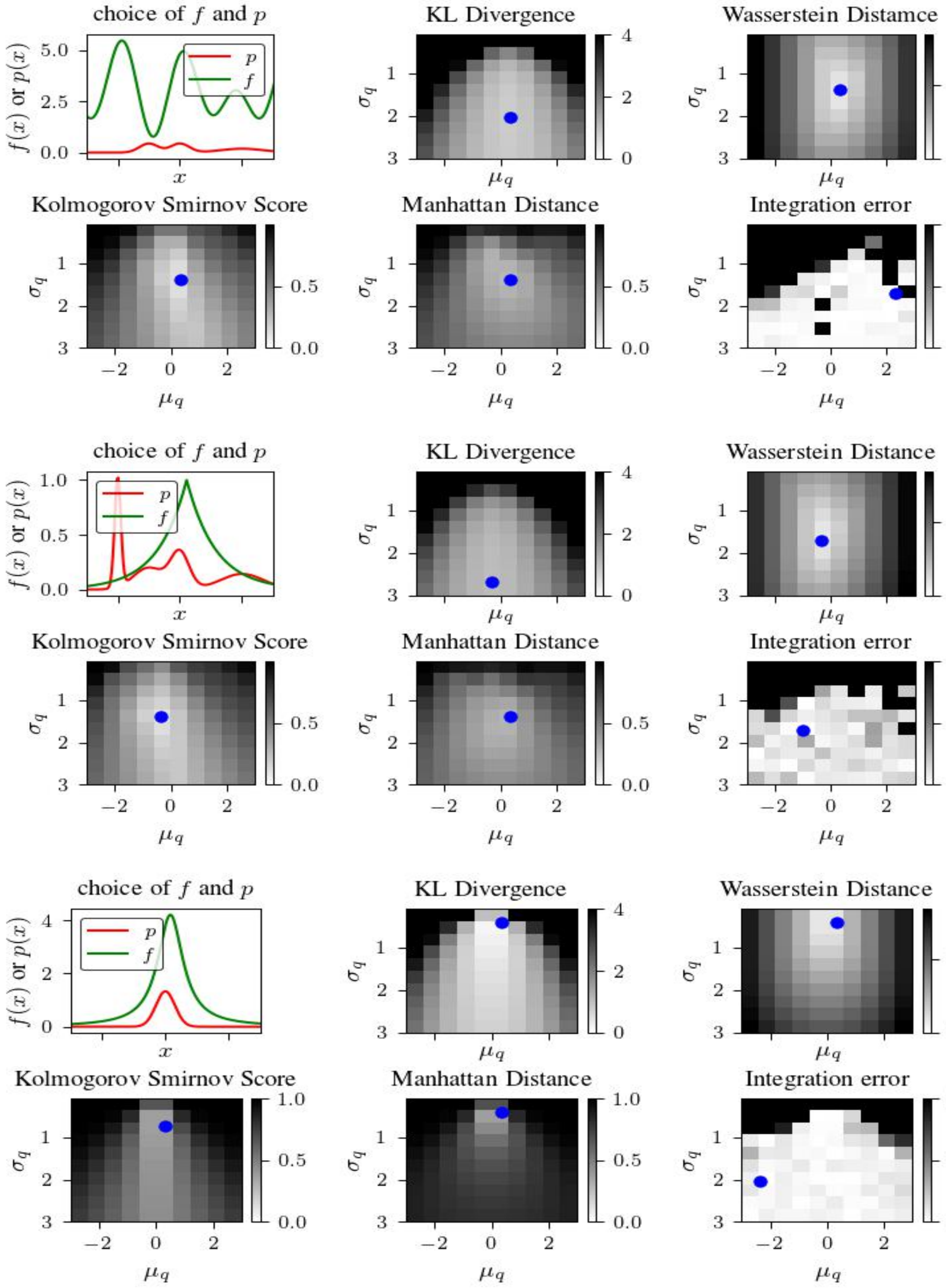

Figure 4.1: comparison of the four proposed similarity measures and the relative absolute error of the mean estimate of a BQ algorithm with a squared exponential covariance function for the GP for three different integrals $\int f(x)p(x)dx$. Shown are heatmaps for each score and the error metric for different parameters of the Gaussian density q. We see that the KL divergence score looks similar to the error metric.

4.2.2 “Non-stationarity” score for the re-weighting trick

As mentioned in section 3.2.2, maximizing the similarity between p and q is not the only possible approach to minimize the distortion of the integrand g. The score introduced in this section quantifies ”(non)-stationarity” which is a property of a random process rather than a metric, in this case of the process g implied by re-weighting f. We will assume f to be a draw of a Gaussian process with some unknown covariance function C. Note that this GP differs from the GP we may use to model f. We additionally assume that $f \sim \mathcal{GP}(m_f, C_f)$ is stationary. These are rather strong assumptions, and it needs to be seen empirically whether this score generalizes to non-stationary and deterministic integrands.

Stationarity and non-stationarity of a processes

A random process $\{\eta(x)\}_{x\in\Omega}$ is said to be stationary when its distribution is unchanged by an index shift, that is $\{\eta(x)\}_{x\in\Omega} = \{\eta(x+\delta)\}_{x+\delta\in\Omega}$ for arbitrary δ [11]. This implies that all its statistics, such as mean and covariance, obey this property as well. A GP is uniquely defined by its mean function $m(x) = \mathbb{E}_\eta(\eta(x))$ and its covariance function $C(x, x') = \operatorname{cov}(\eta(x), \eta(x'))$. Hence, a stationary GP must obey $m(x) = m(x+\delta)$ and $C(x, x') = C(x+\delta, x'+\delta)$ for arbitrary δ. Any covariance function that only depends on the distance $C(x, x') = C(r)$ with $r := \|x - x'\|$ is stationary by definition. For example the squared exponential covariance function is stationary with $C(r) = e^{-0.5r^2}$. The trivial mean function $m(x) = 0$ is also stationary.

Shift score

We will now construct a score in order to measure the change in stationarity of the original integrand f when re-weighting it with a probability density q. For a stationary process $f \sim \mathcal{GP}(m_f, C_f)$ we have $C_f(x, x') = C_f(x + \delta, x' + \delta)$. Assuming f, the re-weighting trick implies a non-stationary process $g \sim \mathcal{GP}(m_g, C_g)$. That is

$$g(x) := f(x)\frac{p(x)}{q(x)} \sim \mathcal{GP}(m_g, C_g) \tag{4.10}$$

where

$$\begin{aligned}
m_g(x) &= \mathbb{E}_f(g(x)) = \mathbb{E}_f\left(f(x)\frac{p(x)}{q(x)}\right) = m_f(x)\frac{p(x)}{q(x)} \\
C_g(x,x') &= \operatorname{cov}\left(g(x), g(x')\right) \\
&= \mathbb{E}_f\left(f(x)\frac{p(x)}{q(x)}f(x')\frac{p(x')}{q(x')}\right) - \mathbb{E}_f\left(f(x)\frac{p(x)}{q(x)}\right)\mathbb{E}_f\left(f(x')\frac{p(x')}{q(x')}\right) \\
&= \frac{p(x)p(x')}{q(x)q(x')}\left(\mathbb{E}_f(f(x)f(x')) - \mathbb{E}_f(f(x))\mathbb{E}_f(f(x'))\right) \\
&= \frac{p(x)p(x')}{q(x)q(x')}C_f(x,x').
\end{aligned} \tag{4.11}$$

Here $C_g(x,x') = C_g(x+\delta, x'+\delta)$ and $m_g(x) = m_g(x+\delta)$ must not necessarily hold true. For now we'll consider $m_f(x) = 0$ and hence $m_g(x) = 0$ which makes the object of interest the covariance function C_g. Recall that we do not have knowledge of the true process f and hence g. We construct a measure of non-stationarity as

$$\begin{aligned}
s(x,x'|\delta) &:= \left|\frac{C(x,x') - C(x+\delta,x'+\delta)}{C(x,x')}\right| \\
&= \left|1 - \frac{C(x+\delta,x'+\delta)}{C(x,x')}\right|.
\end{aligned} \tag{4.12}$$

We see that for a stationary covariance function $s(x,x'|\delta) = 0$ for all x, x' and all δ. When we now plug in C_g from equation 4.11,

$$\begin{aligned}
s_g(x,x'|\delta) &= \left|1 - \frac{C_g(x+\delta,x'+\delta)}{C_g(x,x')}\right| \\
&= \left|1 - \frac{p(x+\delta)p(x'+\delta)}{q(x+\delta)q(x'+\delta)}\frac{q(x)q(x')}{p(x)p(x')}\frac{C_f(x+\delta,x'+\delta)}{C_f(x,x')}\right| \\
&= \left|1 - \frac{p(x+\delta)p(x'+\delta)}{q(x+\delta)q(x'+\delta)}\frac{q(x)q(x')}{p(x)p(x')}\right|,
\end{aligned} \tag{4.13}$$

we get a way of quantifying the change in stationarity. We make the following observations:

- By definition $s_g(x,x'|\delta) \geq 0$.
- $s_g(x,x'|\delta)$ is independent of C_f and only depends on ratios of p and q.
- $s_g(x,x'|\delta) = 0$ for all x, x' and δ if both p and q are stationary that is if $p(x) = p(x+\delta)$ and $q(x) = q(x+\delta)$.

- $s_g(x,x'|\delta) = \left|1 - \frac{p(x+\delta)p(x'+\delta)}{p(x)p(x')}\right|$ if only q is stationary.
- $s_g(x,x'|\delta) = \left|1 - \frac{q(x)q(x')}{q(x+\delta)q(x'+\delta)}\right|$ if only p is stationary.

The score $s_g(x,x'|\delta)$ is defined for a given triplet (x,x',δ). Assuming we are interested in an expected score w.r.t p, we can define

$$\begin{aligned}
\bar{s}_g(\delta) &= \mathbb{E}_{p\times p}(s_g(x,x'|\delta)) \\
&:= \iint_{\Omega\times\Omega} s_g(x,x'|\delta)p(x)p(x')dxdx' \\
&= \iint_{\Omega\times\Omega} \left|1 - \frac{p(x+\delta)p(x'+\delta)q(x)q(x')}{q(x+\delta)q(x'+\delta)p(x)p(x')}\right| p(x)p(x')dxdx'.
\end{aligned} \tag{4.14}$$

We will from now on refer to this score as "shift score" trough out this thesis. It still depends on δ, and it is unclear which δs to choose or if the score can be defined independently of δ. Since we do not have an answer at the moment, for the experiments in chapter 5, we will evaluate the shift score for eight deltas evenly distributed between -4 and 4 and calculate their average to get an estimate for the change in stationarity. As already mentioned, the shift score is based on the assumption that the true integrand is a GP f with a stationary covariance function and mean function $m = 0$. This is not generally the case in practice. Hence some questions arise. Does the shift score correlate with performance under the given assumption? Does this also hold empirically for other integrands? In figure 4.2, we see the effect of re-weighting with different Gaussian densities q on integrands that are draws from a stationary Gaussian process, as well as the shift score for different δs. It seems that at least for these well-behaved integrands, the shift score is able to capture the change in stationarity. Whether the shift score can also be generalized to other integrands will be seen in the experiments in chapter 5.

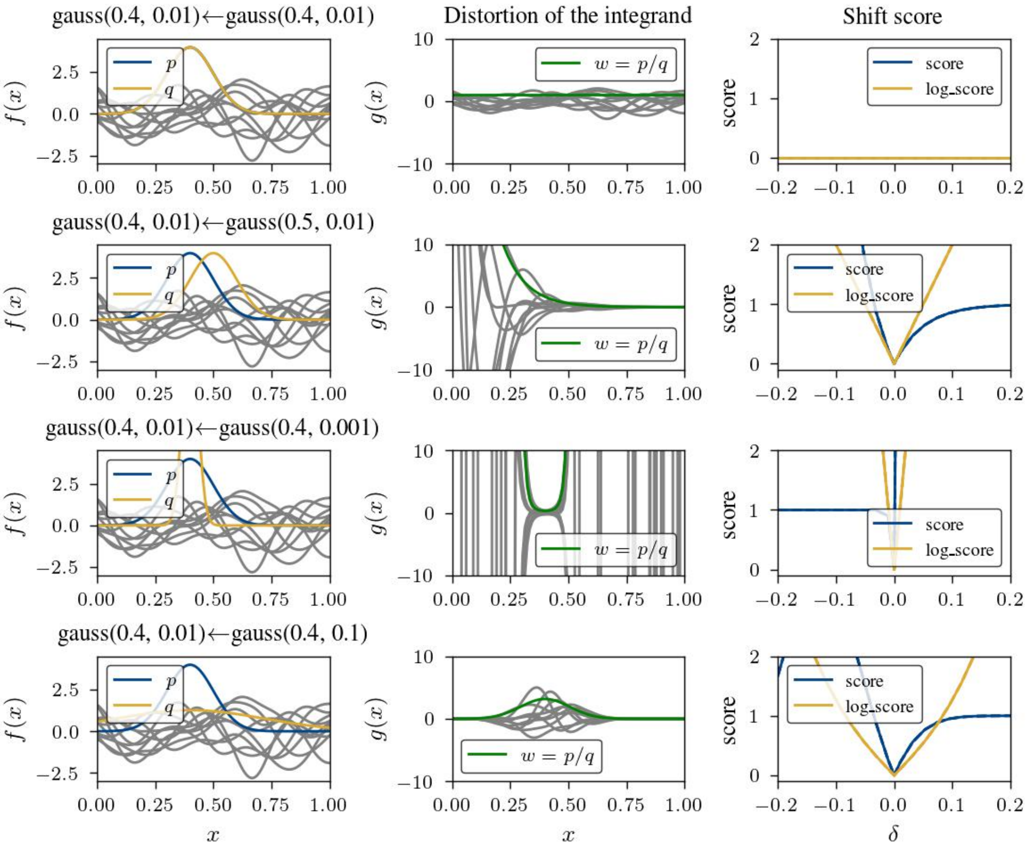

Figure 4.2: We see the distortion of samples from a stationary GP when re-weighting with different Gaussian densities q as well as the shift score for a range of δs. Each row shows the distortion of the integrands caused by re-weighting with a different q followed by the shift score values for different δs. We see that especially the logarithm of the shift score seems to yield higher values for the same choice of *delta* when the re-weighted integrand is highly non-stationary (row three) and is constantly zero for no increase in non-stationarity (row one).

4.2.3 Evidence of the model

In section 4.1.3, we have considered a score that not only aims not to distort the original integrand f but measures how suitable the re-weighted integrand g is for our GP model. The evidence enables us to do exactly that. Here we assess how probable it is to observe g under a given GP model with covariance function C. Calculating the evidence can be computationally expensive since it requires us to integrate over a possibly multidimensional space. But for Gaussian process models, integrals over the parameter space are analytically tractable [16]. When we consider n samples $(\mathbf{x}, y)$ from g the log evidence is given by

$$\log(p(\mathbf{y}|X)) = -\frac{1}{2}\mathbf{y}^T C(X,X)^{-1}\mathbf{y} - \frac{1}{2}\log(|C(X,X)|) - \frac{n}{2}\log(2\pi). \quad (4.15)$$

So far, all the metrics we used depended only on the probability density functions p and q. However, since we sample from g, the evidence also depends on the original integrand f. This is a possibly more accurate score since considering f allows us to make more precise claims about the properties of g than just relying on p and q. However, the cost of computing f forces us to use the same evaluations of f for the BQ algorithm and the evaluation of the density q.

4.2.4 Reflection on the scores

All the above scores aim to quantify how much performance drop we might expect in the BQ algorithm when re-weighting with a probability density q. Except for the evidence score, all mentioned scores only rely on the densities p and q and can therefore be computed independently of f. This is desirable as evaluating f is usually expensive. It also enables us to evaluate the suitability of q for a given p and reuse the results for any integrand f. However, relying on the p and q might not be desirable if f is not well-behaved. Hence, we may want to choose some q anyway that yields better performance. The only score that could do that is the evidence score, which measures how well the GP is suited for the function g. All the above constitutes a trade-off, and it must be shown whether the additional information contained in f is worth the additional computational cost. In the case of the shift score, it is unclear whether the strong assumptions we made on f (draw from a GP with stationary covariance function) can also be generalized to integrands that do not obey these restrictions.

Chapter 5

Empirical evaluation of the Scores

When re-weighting, we are forced to decide on a parameterization of the density q and are naturally interested in choosing the one that leads to the best performance of our BQ algorithm. In this chapter, we will discuss how to find the parameters for a Gaussian density q that are optimal according to each of the scores and further run empirical experiments to get a sense of the performance we can expect under these optimal qs.

5.1 Finding optimal parameters

We aim to find parameters μ_q and σ_q for the Gaussian density $q(x) = \mathcal{N}(x|\mu_q, \sigma_q)$ that minimize the given score. For this, we will use the L-BFGS-B algorithm, a version of the BFGS algorithm mentioned in section 2.1.5 [4]. As the L-BFGS-B algorithm is a gradient-based optimizer, we need to calculate the gradient w.r.t the parameters of q for each of the scores. The gradient of the KL divergence and the gradient of the shift score can be found in appendix A. The gradient of the evidence score is a little bit trickier since we do not only have to consider the gradient w.r.t μ_q and σ_q but also w.r.t. the hyperparameters of the covariance function. Alternatively, we could use an approach like coordinate descent [18]. Here we alternate between optimizing the kernel parameters and optimizing the parameters of q until all parameters are relatively stable. While both approaches are valid, for now, we will use a different optimizer that does not rely on gradient information for the optimization of the evidence score. The Nelder–Mead method is a commonly applied optimization method used to find the minimum of an objective function. It is a direct search method based on function comparison and is often applied to nonlinear optimization problems for which derivatives may not be known [13]. The Nelder-Mead method is both slower and less reliable than the L-BFGS-B algorithm, so for practical use of the evidence score, one of the earlier mentioned approaches would be more desirable. In the interest of time, we will, however, use the Nelder-Mead method for the optimization of the evidence score.

5.2 Experimental setup

In this section, we will describe the design choices made for the experiments.

5.2.1 BQ algorithm

For Gaussian process Regression on the integrand, we use the GPy Python library [2]. For the implementation of BQ, we use the EmuKit Python library, and its GPy wrappers [1]. Throughout all the experiments, we use the already mentioned exponential squared covariance function,

$$C(x_n, x_{n'}) = \theta \mathrm{exp}\left(-\frac{(x_n - x_{n'})^2}{4r^2}\right), \tag{5.1}$$

for the Gaussian process. The points we evaluate the integrand at are sampled from the probability density p.

5.2.2 Integrands

For the original integrand f, we will use functions of the Genz family [6], a set of parameterized function families that cover characteristics such as stationarity [1] or smoothness. Out of the available families, we use:

- Oscillatory Integrand Family: Smooth, stationary integrands.
- Gaussian peak Integrand Family: Smooth, non-stationary integrands.
- Continuous Integrand Family: Non-smooth, non-stationary integrands.

As a fourth function family, we will use random Fourier functions, which are also smooth and stationary, in order to have a second function family for which the assumptions of our GP are fulfilled. The ground truth integrals for functions of the Genz family are analytic w.r.t p, and the ground truth integrals for functions of the random Fourier family are calculated with SciPy's integrate.quad method.

[1] ”Stationarity” is usually used in the context random processes. We use the term ”stationarity” informally, meaning that a function looks similar at all x locations

5.2.3 Probability density p

For the probability density p, we could, of course, use any density we wish. All three scores work for arbitrary p that can be evaluated. In the interest of time, we will limit the experiments to Gaussian mixture densities,

$$p(x|\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{w}) = \mathcal{N}(x|\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{w}) = \sum_{i=1}^{I} \mathbf{w}_i \left(\frac{1}{\boldsymbol{\sigma}_i \sqrt{2\pi}} e^{-\frac{(x-\boldsymbol{\mu}_i)^2}{2\boldsymbol{\sigma}_i^2}} \right), \tag{5.2}$$

as they are relatively flexible and easily implemented.

5.2.4 Metrics to track performance

We want to asses how much impact the re-weighting with the optimized parameters of q has on the performance of the BQ algorithm. We, therefore, need a way to evaluate the performance of the BQ algorithm. The first evaluation method we will use compares the mean $\mu_{\mathcal{D}}$ of the BQ estimate with the true value of the integral F, which we will calculate with SciPy's integrate.quad method. We, therefore, look at the absolute value of the relative difference between the two,

$$s_{rmc}(F) = \left| \frac{F - \mu_{\mathcal{D}}}{F} \right|. \tag{5.3}$$

This evaluation method, however, does not take into consideration that BQ gives us a distribution over the integral value. If we have two BQ estimates that give an equally wrong mean estimate, we usually prefer the one with higher variance since this indicates its uncertainty. In order to assess the calibration of the distribution $F_{GP}|\mathcal{D}$, we use the expected logarithmic density ratio of $p(F_{GP}|\mathcal{D})$ and $p(F_{GP}|\mathcal{D} = F)$ evaluated on the ground truth F,

$$\begin{aligned} s_{UQ}(F) &= \mathbb{E}\left(\log \frac{p(F_{GP}|\mathcal{D})}{p(F_{GP}|\mathcal{D} = F)} \right) \\ &= \frac{1}{2} \left(\frac{(F - \mu_{\mathcal{D}})^2}{\sigma_{\mathcal{D}}} - 1 \right). \end{aligned} \tag{5.4}$$

A derivation of the second line in equation 5.4 can be found in appendix B. A value of zero here means that the BQ estimate is well calibrated. A value greater than zero indicates that the estimate is over-confident, and a value less than zero indicates that the estimate is under-confident. While overconfidence should be avoided if possible, it is usually less of a problem if the estimate is slightly under-confident.

5.2.5 Workflow

We begin by constructing 20 integrands f_i with random parameters from one of the four integrand families mentioned in section 5.2.2. We then choose three different Gaussian mixture densities for p. For each of the probability densities p, we will optimize the parameters of q according to the KL divergence score and according to the shift score. Recall that both these scores can be optimized independently of f. We also calculate the optimal parameters for q according to the evidence score for each combination of f and p. We then evaluate the performance of our BQ algorithm on the re-weighted integrals on a fixed budget of samples from f_i from n=3 to n=29 in increments of one. The performance will be evaluated with the two metrics introduced in section 5.2.4. For each budget, we will average the score of all 20 integrands and therefore get,

$$\frac{1}{20}\sum_{i=1}^{20} s_{rmc}(F_i) = \frac{1}{20}\sum_{i=1}^{20}\left|\frac{F_i - \mu_{\mathcal{D}i}}{F_i}\right|, \tag{5.5}$$

$$\frac{1}{20}\sum_{i=1}^{20} s_{UQ}(F_i) = \frac{1}{40}\sum_{t=1}^{20}\left(\frac{(F_i - \mu_{\mathcal{D}i})^2}{\Sigma_{\mathcal{D}i}} - 1\right) \tag{5.6}$$

as metrics. We then repeat the same procedure for the remaining four integrand families. In all plots, we also show the performance of the BQ algorithm if we re-weigh with $q(x) = \mathcal{N}(x; 0, 1)$. This is done to assess how the optimized qs perform with respect to the current standard practice for re-weighing with a Gaussian density.

5.3 Results

In the following, we will describe the experiments' results and briefly discuss possible explanations.

5.3.1 Random Fourier integrands

The random Fourier functions are smooth and stationary and are, therefore, relatively easy to capture for our GP. In figure 5.1, we see that in the case of a Gaussian density p, optimizing the KL divergence score and the shift score results in a q that is basically identical to p. We, therefore, see virtually no distortion of the integrand. This is to be expected since we know that both scores are minimized when p and q are identical. The evidence score also suggests a q similar to p. However, it is not identical and results in a slight distortion of the integrand, making it less stationary. When optimizing the evidence score, we only have access to a limited amount of samples from the original integrand f. Recall that, to remain somewhat efficient, we have to use the same samples from f to both calculate the optimal q and fit the GP to the integrand. This might make it hard for the evidence score to capture the actual structure of our integrand and, therefore, might lead to optimizations that make the new integrand look well-behaved for the points we sampled but not in general. This might explain the sub-optimal properties of the re-weighted integral for the evidence score. We also see that simply re-weighting with a standard Gaussian $q(x) = \mathcal{N}(x; 0, 1)$ leads to a highly distorted integrand, which was to be expected. We can see that all three optimized qs yield a similar relative absolute error of the mean estimate and that they all outperform the standard Gaussian significantly. The KL divergence score and the shift score seem well calibrated, while the evidence score appears slightly overconfident, which might result from insufficient samples, as mentioned above. The standard Gaussian, however, is highly overconfident. In figure 5.2 and 5.3 we can see the same setup but with different Gaussian mixture densities for p. They support the claims made about the experiment in figure 5.1. It is, however, interesting to note that in figure 5.2 re-weighting with the standard Gaussian q always results in a relative absolute error of the mean estimate of one. Here the GP always has a mean estimate of zero, and therefore our BQ mean estimate is also zero. When we recall the definition of the relative absolute error of the mean estimate, we see that if $\mu_{\mathcal{D}} = 0$, the score always equals one.

5.3.2 Oscillatory integrands

The oscillatory functions are also smooth and stationary and, therefore, relatively easy to capture for our GP. For the oscillatory integrands we see similar results as for the random Fourier integrands (figure 5.4, figure 5.5, figure 5.6). Again all three scores seem to perform relatively similarly, and with the exception of the experiment seen in figure 5.5, they significantly outperform the standard Gaussian q. The similar performance for the experiment in figure 5.5 is probably because the standard Gaussian just happened to be relatively similar to the probability density p. We can also see in figure 5.5 that even though p is Gaussian density, the optimal q proposed by the shift score does not resemble it exactly. It is unclear whether this is due to a failure of the optimization algorithm or the choice of δs on which we evaluate the score.

5.3.3 Gaussian peak integrands

The Gaussian peak functions are smooth, but they are highly non-stationary. They propose a challenge to our BQ algorithm as samples are likely to be taken at positions of the integrand that do not give valuable information about its structure. In figure 5.7, we see that for a Gaussian density p, the KL divergence score and the shift score perform similarly. They both recommend the same q and result in the same relative absolute error of the mean estimate and similarly well-calibrated BQ estimates. In figure 5.8 and figure 5.9 they do not yield the same optimal q. However, their recommendations are still relatively similar and again result in similarly minor distortions of the integrand and, therefore, also in similar relative absolute errors of the mean estimates and similarly well-calibrated BQ estimates. The standard Gaussian also seems to result in a minor distortion of the integrand and a relative absolute error of the mean estimate to the KL divergence score and the shift score. However, for the experiments in figure 5.8 and figure 5.9, the standard Gaussian q seems to result in an overconfident BQ estimate.

In figure 5.7, we see that the evidence score suggests an optimal q that is significantly thinner than the Gaussian density p. This results in a very flat integrand. This flattening of the Gaussian peak integrands results in a relative absolute error of the mean estimate close to zero and an uncertainty quantification score of -0.5, independent of the number of samples. In figure 5.8 and figure 5.9, we see similarly, though not as pronounced, effects. In both experiments, the q according to the evidence score flattens the integrand and outperforms the other scores for the relative absolute error of the mean estimate while remaining well calibrated. We see that for Gaussian peak integrands, the evidence score seems to be able to distort the integrand such that it can be better approximated by the GP.

5.3.4 Continuous integrands

The continuous functions are non-smooth and non-stationary and should pose a significant challenge to our GP. In figure 5.10, we again see that the KL divergence score and the shift score recommend an optimal q equal to Gaussian density p and therefore do not distort the integrand at all. The standard Gaussian q also results in a relatively minor distortion of the integrand. The optimal q according to the evidence score, however, causes a significant distortion of the integrand for $x > 3$ or $x < 3$. However, we also see that the relative absolute error of the mean estimate is similar for all qs, which is somewhat surprising. The distortion the optimal q according to the evidence causes in the integrand does not seem to affect the relative absolute error of the mean estimate. This might be because the distortion it causes is mainly affecting parts of the integrand where p has basically no mass and which are therefore not all too relevant for the estimation of the integrand. Except for the evidence score, which results in overconfident BQ estimates, all qs result in well-calibrated BQ estimates. In figure 5.11, all three optimized qs have a similar effect on the integrand, which gets distorted and resembles p. The standard Gaussian

q strongly distorts the original integrand. The relative absolute error of the mean estimate is similar for all three optimized qs and significantly better than for the standard Gaussian q. Except for the optimal q according to the KL divergence, all qs seem to result in an overconfident BQ estimation. In figure 5.12, all qs result in a relatively minor distortion of the integrand and similar performance for the relative absolute error of the mean estimate. However, while the optimal q according to the KL divergence score and the optimal q according to the shift score are well calibrated, the other two result in an overconfident BQ estimate. We see that the integrands shown in figure 5.10 and figure 5.12 do not get meaningfully distorted by any of our qs. The thin continuous integrands seem to be relatively robust against distortions from re-weighting with Gaussian densities. The stretched-out continuous integrand in 5.11, however, gets heavily distorted by the standard Gaussian q, and we see similar results as for the random Fourier integrands and the oscillatory integrand. Surprisingly, our BQ estimates seem to be relatively accurate for these integrands. The BQ algorithm seems to have similar performance for the continuous integrands as for the random Fourier integrands and the oscillatory integrands.

5.3.5 Reflection on the results

All three proposed scores seem to perform reasonably well for the test integrands in our experiments. Especially for the random Fourier integrands and the oscillatory integrands, all scores seem to outperform the standard method of just choosing a standard Gaussian significantly.

The shift score and the KL divergence score recommend qs that result in a relatively similar performance of the BQ algorithm. However, the KL divergence score tends to recommend wider Gaussian densities for q, which seems to lead to a better calibration of the BQ estimates. The KL divergence score is also easier to optimize and appears more robust, making it look like a favorable choice.

For the non-stationary Gaussian peak integrands, the evidence score enables us to distort the original integrands such that they can be better captured by our GP. We see that the evidence score can enable the BQ algorithm to make more accurate estimates of the integral value than in the non-re-weighted case we essentially see for the KL divergence score and the shift score in figure 5.7. The evidence score also outperforms the standard selection method for the random Fourier integrands and the oscillatory integrands. However, it does perform similarly to the other two scores and is therefore probably not worth the extra computation costs. Moreover, using the evidence score generally tends to result in slightly overconfident BQ estimates. The fact that we only have access to a limited amount of samples for the calculation of the evidence score might make it difficult to capture the underlying structure of the integrand. We can also only optimize the evidence for the sampled points. This might lead to an integrand that appears well-behaved for the sampled points but not necessarily for the other points, which might lead to overconfident BQ estimates.

The KL divergence score and the shift score are both less informed and easier to compute than the evidence score. While the evidence score seems to result in better recommendations for special cases, the other two scores seem to be more robust and often similar in performance, especially the KL divergence. For the time being, based on the empirical experiments we have done, we recommend the KL divergence score to select a suitable q for the re-weighting trick in Bayesian quadrature.

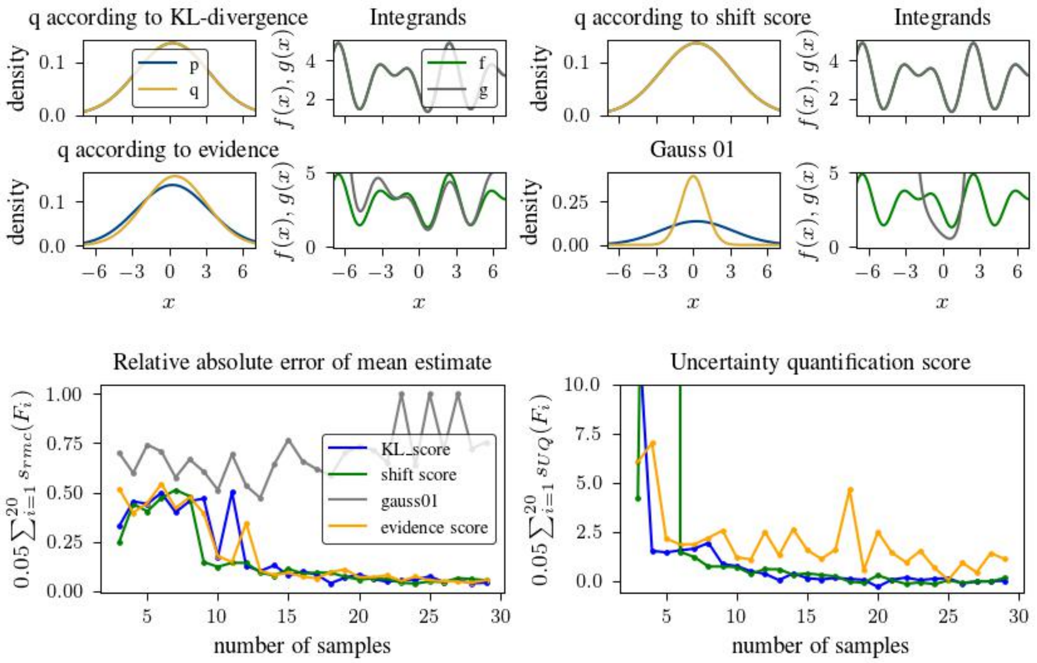

Figure 5.1: Effect of re-weighting for integrals with random Fourier integrands and probability density $p(x) = \mathcal{N}(x; 0.271, 2.908)$. The first six plots show the Gaussian probability densities q obtained by optimizing each of the scores and the effect the re-weighting has on one exemplary integrand of the integrand family, as well as the effect of re-weighting with a standard Gauss $q(x) = N(x; 0, 1)$. The optimized qs are: $q_{KL}(x) = N(x; 0.27, 2.91)$ according to the KL divergence score, $q_s(x) = N(x; 0.27, 2.91)$ according to the shift score, $q_e(x) = N(x; 0.48, 2.54)$ according to the evidence. Recall that the evidence score chooses a separate parameterization for each integrand; the shown optimized q is therefore specific to the integrand at hand. The bottom two plots show the performance of the BQ algorithm (described in 5.2.1) measured with the metrics described in 5.2.5 on the integrals re-weighted with the different qs. We see that q_{KL} and q_s are basically identical to p, and we, therefore, see virtually no distortion of the integrand. For this particular example, q_e is slightly thinner than p, and we see a slight distortion of the integrand. The standard Gaussian, however, causes a substantial distortion of the integrand. We can see that all three optimized qs yield a similar relative absolute error of the mean estimate and that they all outperform the standard Gaussian, which does not seem to improve with more samples. The KL divergence score and the shift score seem well calibrated, while the evidence score appears slightly overconfident. The standard Gauss can not be seen in the last plot since it is highly overconfident.

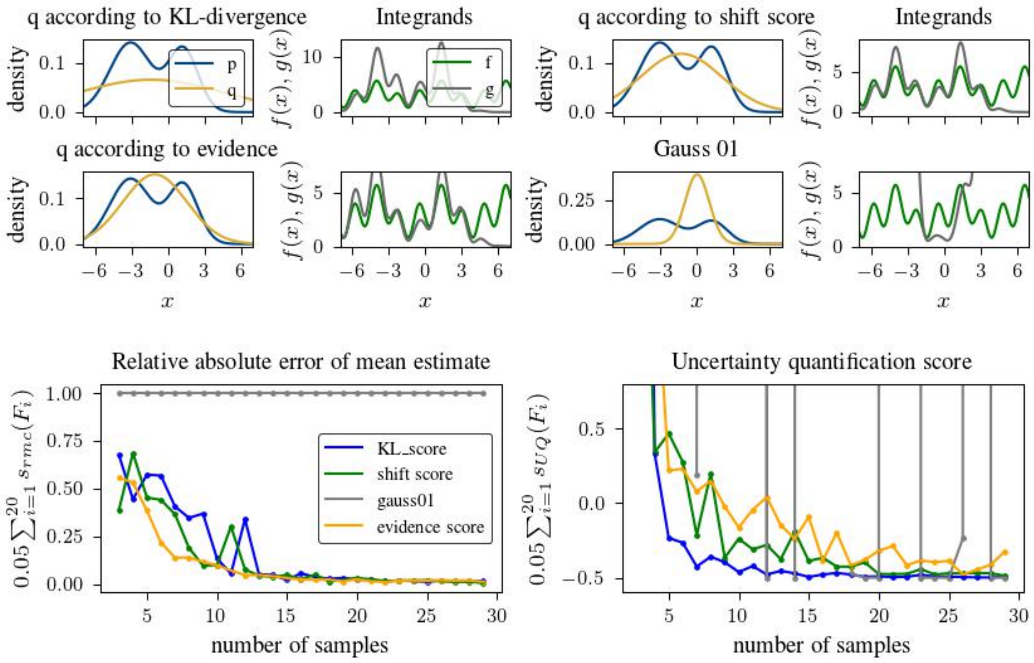

Figure 5.2: Effect of re-weighting for integrals with random fourier integrands and probability density $p(x) = \mathcal{N}(x; \boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{w})$, where $\boldsymbol{\mu} = [-3.08, 1.267]$, $\boldsymbol{\sigma} = [1.707, 1.216]$ and $\mathbf{w} = [0.602, 0.398]$. Setup as described in figure 5.1. The optimized qs are: $q_{KL}(x) = N(x; -1.35, 6.04)$ according to the KL divergence score, $q_s(x) = N(x; -1.26, 3.29)$ according to the shift score, $q_e(x) = N(x; -1.08, 2.63)$ according to the evidence. The optimization of all three scores yields a q that does not result in a strong distortion of the integrand. The standard Gaussian, however, causes a substantial distortion of the integrand. We can see that all three optimized qs yield a similar relative absolute error of the mean estimate and that they all outperform the standard Gaussian, which does not seem to improve with more samples. The BQ algorithm is also well calibrated for the three optimized qs. The standard Gauss seems to be highly overconfident in most cases.

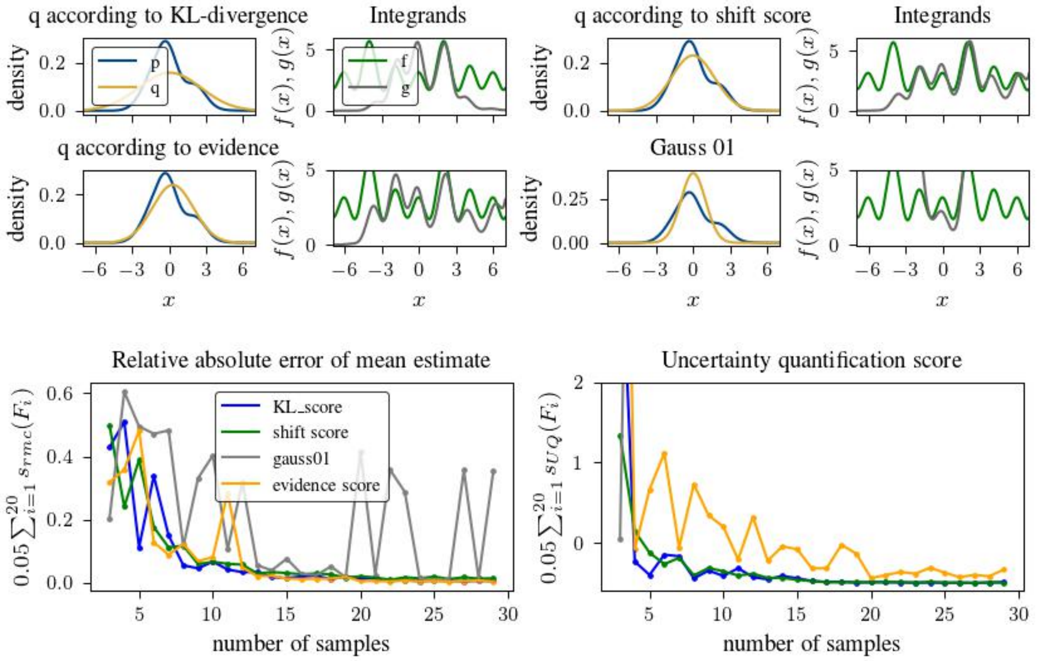

Figure 5.3: Effect of re-weighting for integrals with random fourier integrands and probability density $p(x) = \mathcal{N}(x; \boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{w})$, where $\boldsymbol{\mu} = [-1.244, -0.078, 2.07, 2.205]$, $\boldsymbol{\sigma} = [1.003, 0.795, 0.881, 1.681]$ and $\mathbf{w} = [0.35, 0.404, 0.218, 0.028]$. Setup as described in figure 5.1. The optimized qs are: $q_{KL}(x) = N(x; 0.05, 2.52)$ according to the KL-divergence score, $q_s(x) = N(x; -0.01, 2.47)$ according to the shift score, $q_e(x) = N(x; 0.0, 1.88)$ according to the evidence. The optimization of all three scores yields a q that increases the non-stationarity of the integrand but otherwise conserves the properties of the original integrand. The standard Gaussian, however, causes a substantial distortion of the integrand. We can see that all three optimized qs yield a similar relative absolute error of the mean estimate and that they all outperform the standard Gaussian, which does not seem converge to a reasonable approximation. The BQ algorithm is also well calibrated for the three optimized qs. However, for q_e we require more samples in order to be well calibrated. The standard Gauss can not be seen in the last plot since it is highly overconfident.

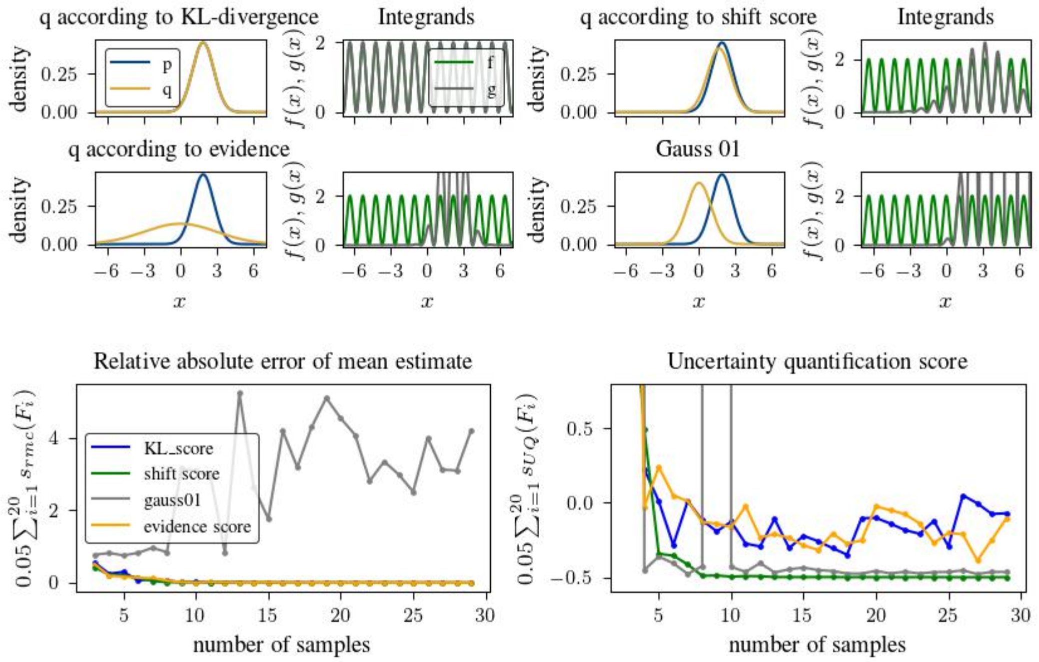

Figure 5.4: Effect of re-weighting for integrals with oscillatory integrands and probability density $p(x) = \mathcal{N}(x; 1.883, 0.879)$. Setup as described in figure 5.1. The optimized qs are: $q_{KL}(x) = N(x; 1.88, 0.88)$ according to the KL divergence, $q_s(x) = N(x; 2.01, 1.35)$ according to the shift score, $q_e(x) = N(x; 1.88, 0.86)$ according to the evidence. We see that the optimal q according to the KL divergence conserves the original integrand, the shift score yields a q that is slightly slightly different from p and reduces the stationarity of the integrand and the evidence score yields a q that is much wider than p and heavily reduces the stationarity of the integrand. The standard Gaussian, however, causes a substantial distortion of the integrand. We can see that all three optimized qs yield a similar relative absolute error of the mean estimate and that they all outperform the standard Gaussian, which does not seem converge to a reasonable approximation. The BQ algorithm is well calibrated for the three optimized qs as well as the standard Gaussian.

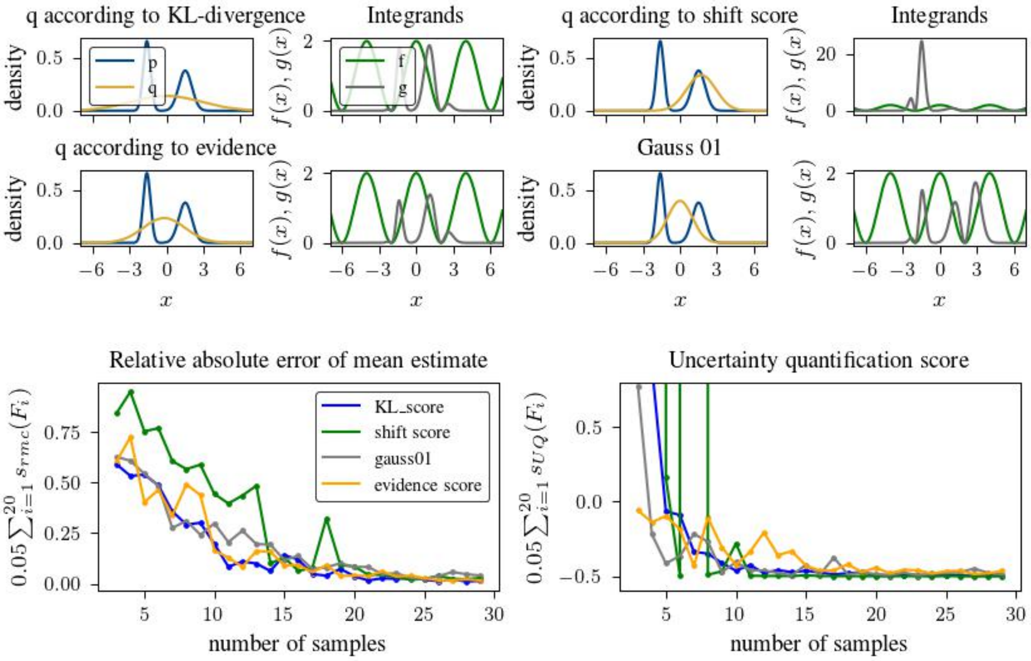

Figure 5.5: Effect of re-weighting for integrals with oscillatory integrands and probability density $p(x) = \mathcal{N}(x; \boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{w})$, where $\boldsymbol{\mu} = [-1.603, 1.510]$, $\boldsymbol{\sigma} = [0.295, 0.533]$ and $\mathbf{w} = [0.49, 0.51]$. Setup as described in figure 5.1. The optimized qs are: $q_{KL}(x) = N(x; 0.02, 2.84)$ according to the KL divergence, $q_s(x) = N(x; 1.72, 1.17)$ according to the shift score, $q_e(x) = N(x; -0.2, 1.69)$ according to the evidence. All three optimised qs as well as the standard Gaussian have a similar distortion effect on the integrand. The BQ algorithm has a similar relative absolute error of the mean estimate for all four qs and they appear to be similarly well calibrated.

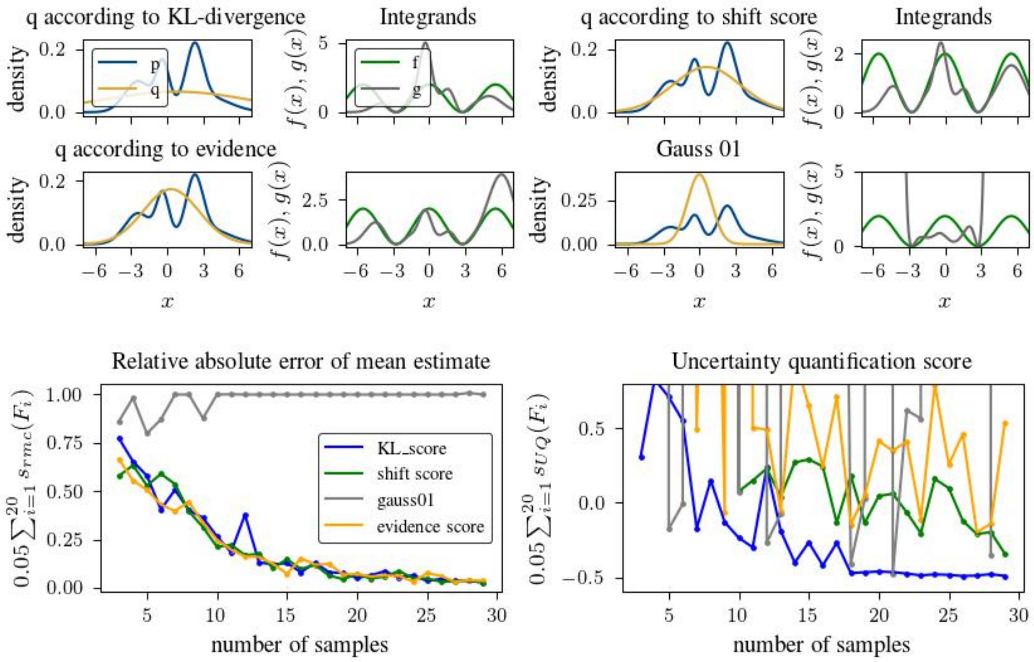

Figure 5.6: Effect of re-weighting for integrals with oscillatory integrands and probability density $p(x) = \mathcal{N}(x; \boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{w})$, where $\boldsymbol{\mu} = [-2.415, -0.357, 2.285, 3.117]$, $\boldsymbol{\sigma} = [1.112, 0.522, 0.68, 1.961]$ and $\mathbf{w} = [0.271, 0.182, 0.295, 0.252]$. Setup as described in figure 5.1. The optimized qs are: $q_{KL}(x) = N(x; 0.74, 6.14)$ according to the KL divergence, $q_s(x) = N(x; 0.6, 2.83)$ according to the shift score, $q_e(x) = N(x; 0.32, 2.3)$ according to the evidence. We see that all three optimised qs have a similarly minor distortion effect on the integrand. The standard Gaussian q, however, greatly distorts the integrand. The BQ algorithm has a similar relative absolute error of the mean estimate for all three optimised qs and they all perform significantly better than the standard Gaussian q, for which the error is almost always one. The BQ algorithm is well calibrated for the KL divergence score and the shift score, while it seems to be slightly overconfident for the evidence and highly overconfident for the standard Gaussian.

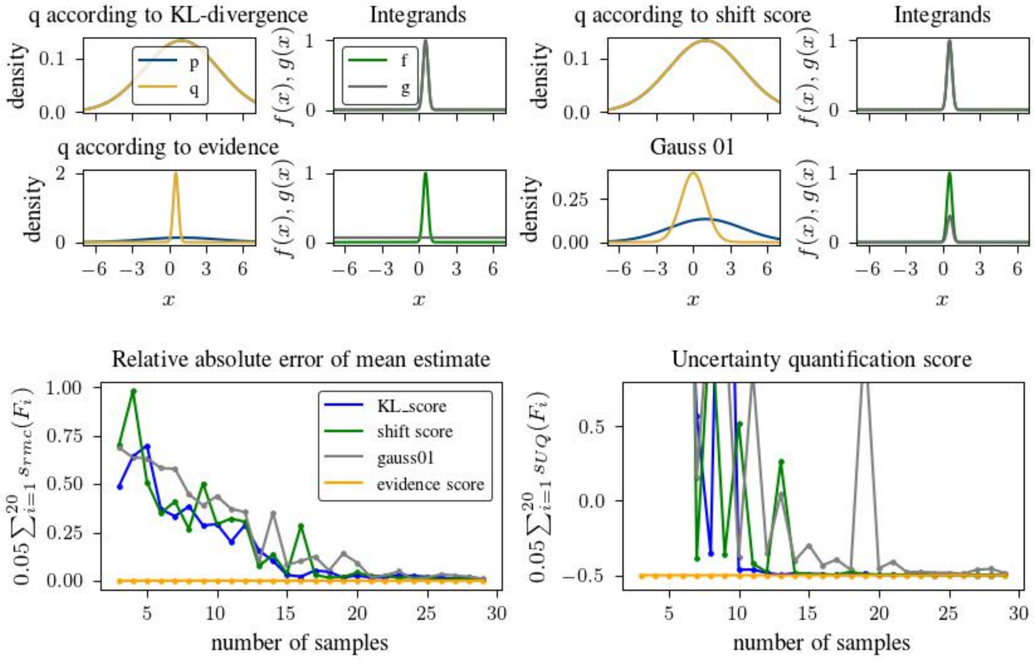

Figure 5.7: Effect of re-weighting for integrals with Gaussian peak integrands and probability density $p(x) = \mathcal{N}(x; 1, 3)$. Setup as described in figure 5.1. The optimized qs are: $q_{KL}(x) = N(x; 1, 3)$ according to the KL divergence, $q_s(x) = N(x; 1, 3)$ according to the shift score, $q_e(x) = N(x; 0.54, 0.2)$ according to the evidence. We see that both the q_{KL} and the q_s do not distort the integral at all. Optimizing the evidence score, for the shown example, results in a very thin q that heavily flattens the original integrand. The standard Gaussian q also flattens the original integrand but not nearly as much as the q_e. We further see that the KL divergence score, the shift score and the standard Gaussian all have similar results for the relative absolute error of the mean estimate and are similarly well calibrated. The q_e, however, results in a relative absolute error of the mean estimate of close to zero independent of the number of samples. The uncertainty quantification score is also always -0.5 independent of the number of samples.

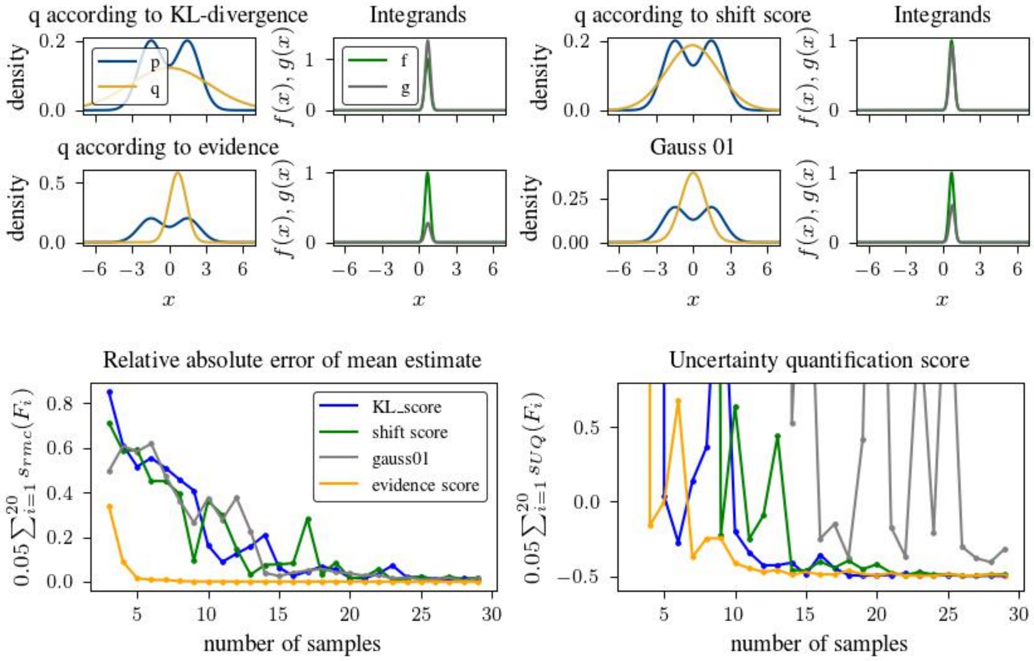

Figure 5.8: Effect of re-weighting for integrals with Gaussian peak integrands and probability density $p(x) = \mathcal{N}(x; \boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{w})$, where $\boldsymbol{\mu} = [-1.5, 1.5]$, $\boldsymbol{\sigma} = [1, 1]$ and $\mathbf{w} = [0.5, 0.5]$. Setup as described in figure 5.1. The optimized qs are: $q_{KL}(x) = N(x; 0, 3.25)$ according to the KL divergence, $q_s(x) = N(x; 0.07, 2.12)$ according to the shift score, $q_e(x) = N(x; 0.7, 0.68)$ according to the evidence. We see that both the q_{KL} and q_s cause only minor distortions of the integrand. Optimizing the evidence score, for the shown example, results in a thin q that flattens the original integrand. The standard Gaussian q also flattens the original integrand but not as much as the q_e. We further see that the KL divergence score, the shift score and the standard Gaussian all have similar results for the relative absolute error of the mean estimate. The q_e, however, has a relative absolute error of the mean estimate of close to zero for sample sizes greater than five. The three optimised qs seem to be similarly well calibrated. The standard Gaussian, however, is slightly overconfident most of the time.

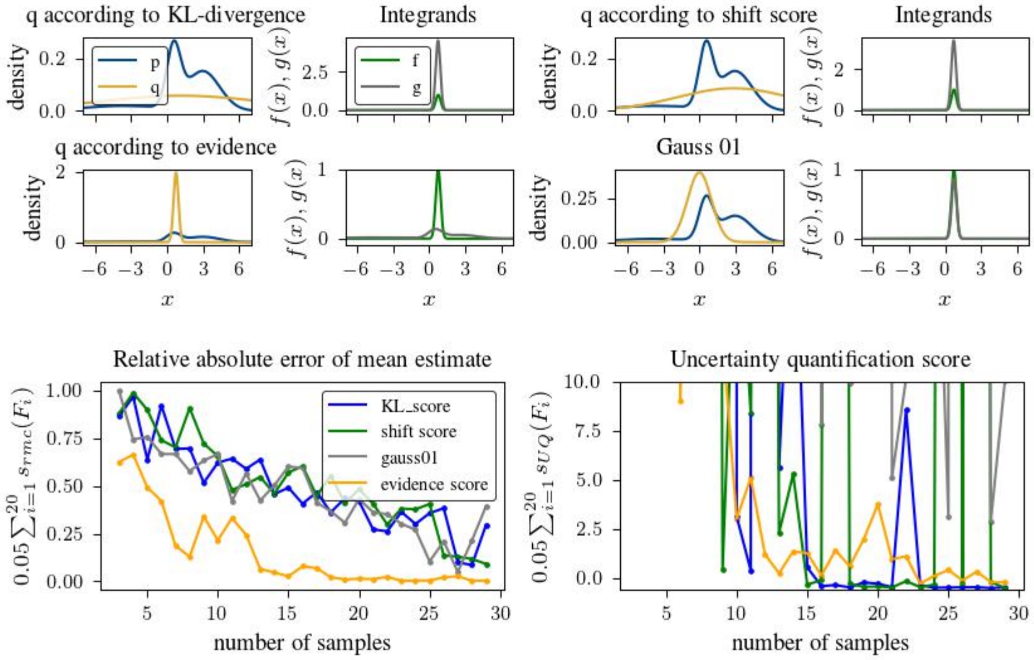

Figure 5.9: Effect of re-weighting for integrals with Gaussian peak integrands and probability density $p(x) = \mathcal{N}(x; \boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{w})$, where $\boldsymbol{\mu} = [-3.732, 0.537, 2.967]$, $\boldsymbol{\sigma} = [3.232, 0.573, 1.4]$ and $\mathbf{w} = [0.154, 0.323, 0.523]$. Setup as described in figure 5.1. The optimized qs are: $q_{KL}(x) = N(x; 0.8, 7.61)$ according to the KL divergence, $q_s(x) = N(x; 1.74, 4.21)$ according to the shift score, $q_e(x) = N(x; 0.68, 0.18)$ according to the evidence. We see that re-weighting with q_{KL}, q_s or the standard Gaussian seems to distort the integrand only slightly. They also perform similar for the relative absolute error of the mean estimate. While for q_{KL} and q_s the estimate seems to be somewhat well calibrated, for the standard Gaussian the estimate is generally overconfident and varies greatly. q_e is again very thin and results in a spreading of the integrand integrand. The evidence score has a significantly lower relative error of the mean estimate and is, while somewhat overconfident, reasonably well calibrated.

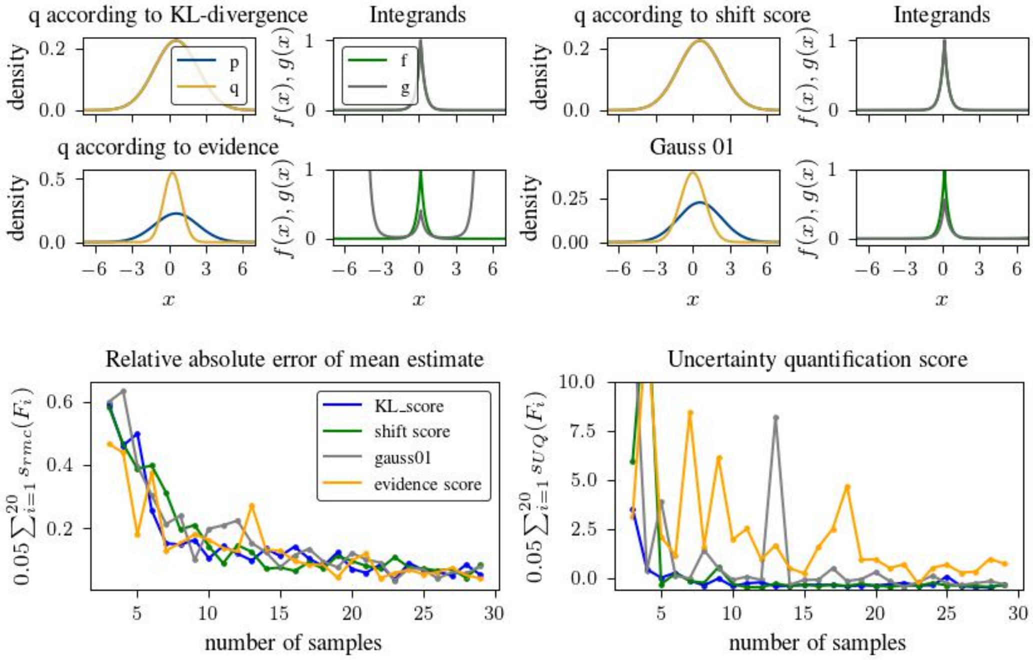

Figure 5.10: Effect of re-weighting for integrals with continuous integrands and probability density $p(x) = \mathcal{N}(x; 0.574, 1.756)$. Setup as described in figure 5.1. The optimized qs are: $q_{KL}(x) = N(x; 0.57, 1.76)$ according to the KL divergence, $q_s(x) = N(x; 0.57, 1.76)$ according to the shift score, $q_e(x) = N(x; 0.84, 0.8)$ according to the evidence. We see that the q_{KL}, q_S and the standard Gaussian do not have a strong distorting effect on the integrand. Optimizing the evidence score, for the shown example, results in a relatively thin q that heavily distorts the original integrand for values greater than three or below -3. We further see that all qs have similar results for the relative absolute error of the mean estimate. Except for q_e all qs result in a well calibrated estimate of the BQ algorithm. For q_e the BQ algorithm appears to be slightly overconfident.

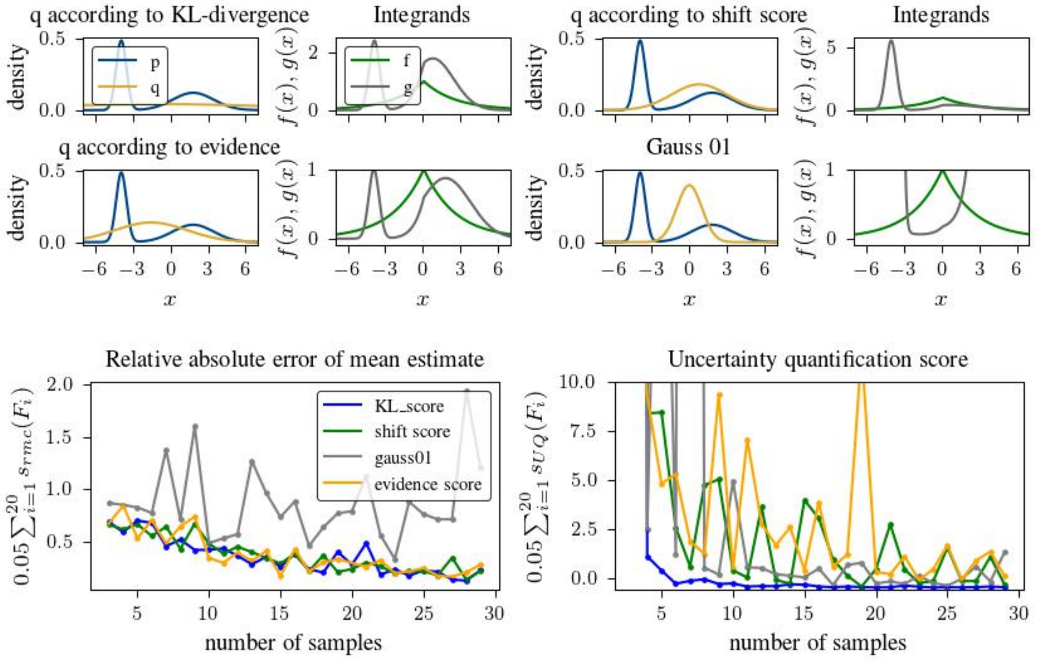

Figure 5.11: Effect of re-weighting for integrals with continuous integrands and probability density $p(x) = \mathcal{N}(x; \boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{w})$, where $\boldsymbol{\mu} = [-1.617, 3.181]$, $\boldsymbol{\sigma} = [1.578, 0.178]$ and $\mathbf{w} = [0.138, 0.862]$. Setup as described in figure 5.1. The optimized qs are: $q_{KL}(x) = N(x; -1.06, 9.26)$ according to the KL divergence, $q_s(x) = N(x; 0.79, 2.21)$ according to the shift score, $q_e(x) = N(x - 1.56, 2.89)$ according to the evidence. We see that all three optimized qs have a similar effect on the integrand; which gets distorted and resembles p. The standard Gaussian q strongly distorts the integrand. The BQ algorithm has a similar relative absolute error of the mean estimate for all three optimised qs and they all perform significantly better than the standard Gaussian q. Except for q_{KL} all qs result in a slightly overconfident estimate of the BQ algorithm. For q_{KL} the BQ algorithm appears to be well calibrated.

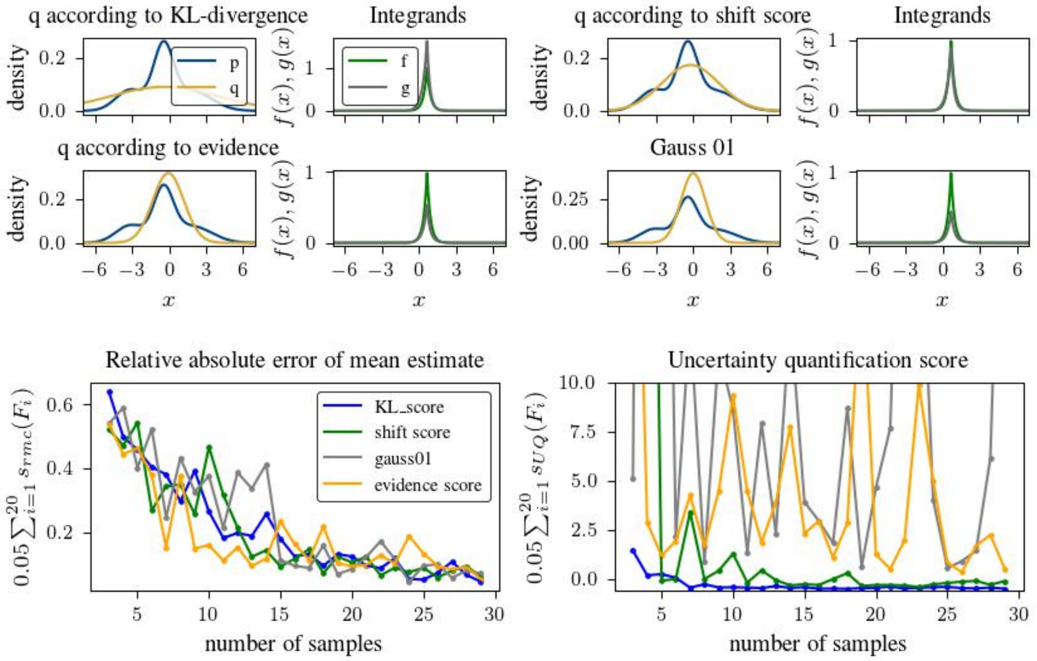

Figure 5.12: Effect of re-weighting for integrals with continuous integrands and probability density $p(x) = \mathcal{N}(x; \boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{w})$, where $\boldsymbol{\mu} = [-3.14, -0.47, 1.95, 1.764]$, $\boldsymbol{\sigma} = [1.177, 0.796, 1.7478, 1.6]$ and $\mathbf{w} = [0.235, 0.455, 0.54, 0.256]$. Setup as described in figure 5.1. The optimized qs are: $q_{KL}(x) = N(x; -0.39, 4.41)$ according to the KL divergence, $q_s(x) = N(x; -0.23, 2.31)$ according to the shift score, $q_e(x) = N(x; -0.08, 1.26)$ according to the evidence. We see that all four qs distort the integral only slightly. We also see that the BQ algorithm has a similar relative absolute error of the mean estimate for all qs. The BQ estimate seems to be well calibrated for q_s and q_{KL} and overconfident for the q_e and the standard Gaussian.

Chapter 6

Discussion and Outlook

We have shown that re-weighting can have a significant influence on the performance of Bayesian quadrature. The main contribution of this thesis are the proposed methods of choosing parameters for q in order to minimize the negative effects re-weighting might have on the performance of Bayesian quadrature. The empirical experiments in chapter 5 have shown that the proposed methods have the potential to improve the accuracy of the BQ estimation compared to the common practice of simply using a standard Gaussian q for the re-weighting.

In some cases, the evidence score seems to distort the integrand so that it can be more easily captured by the GP. However, for most of the tested integrand families, its performance was similar to the shift score and the KL-divergence score, which is underwhelming if we consider the difference in computational cost. It is unclear whether this indicates that the other two scores have already found an optimal q or if the evidence score cannot find a better one. For the experiments in this thesis, we relied on the Nelder–Mead method, which does not use gradient information, to find the optimal parameters for q according to the evidence score. It is unclear if this relatively uninformed optimization method might have negatively affected the recommendation of the evidence score. Another potential problem of the evidence score is that we have to rely on a limited amount of samples from the integrand to asses how suitable the new integrand is. For performance reasons, we have to recycle the samples we use for the optimization of the evidence score to fit the GP to the integrand. We decided to select the samples according to the density p, which might not necessarily be the best possible approach. The limited amount of samples also might make it hard for the evidence score to capture the structure of the integrand. When choosing an optimal q, the evidence score only considers its effect on the sampled points and ignores the global effect on the integrand. This might also lead to overconfident BQ estimates since we only optimize the samples we use for fitting the GP.

The shift and KL divergence scores had very similar performance for all test integrals. The shift score still depends on the choice of parameters δ, and it is unclear if it can be defined independently of δ. We decided to average over the shift score evaluated for a set of δs in order to avoid this problem. This might have negatively affected its performance. However, we can still conclude that the idea of minimizing the increase in non-stationarity seems to be a useful concept that generalizes to integrands that are not draws from a GP.

The experiments we conducted to assess our scores' usefulness were relatively limited. All test integrands used Gaussian mixture densities for p. While we know that all scores are compatible with arbitrary densities ps, we have only evaluated them for Gaussian mixture densities and can not make any definite statements about their usefulness for other densities. Especially the KL divergence score and the shift score, which only depend on q and p, should be tested on other densities p. The proposed scores should also generalize to multiple dimensional integrals. However, this has not been tested, and it is unclear whether or how much the recommendations lose their significance for multi-dimensional integrals. We also limited ourselves to GPs with squared exponential covariance function, which further weakens the expressiveness of our experiments. The same can be said about the choice of integrands. While we specifically chose integrands that cover a wide range of properties, it would certainly be advisable to expand the number of different integrand families before making a final recommendation about which scores are useful in which scenarios. While the metrics we used to evaluate the performance of the BQ algorithm for the different re-weighted integrals seem to be useful, the results we got from the uncertainty quantification score are to be taken with caution since uncertainty quantification in Probabilistic Numerics in itself is a complicated topic.

Appendix A

Gradients w.r.t parameters of q

A.1 KL divergence score

Since in the experiments all measures p are Gaussian mixture measures, we can limit ourselves to the derivation of equation 4.9 for the gradient of the KL divergence w.r.t to the parameters of q:

$$\begin{aligned}
\frac{\partial \mathrm{KL}(p,q)}{\mu_q} &= \frac{\partial}{\mu_q}\sum_{i=1}^{I}\mathbf{w}_i(\frac{1}{2}log(\frac{\sigma_q}{\boldsymbol{\sigma}_{pi}}) + \frac{\boldsymbol{\sigma}_{pi} + (\boldsymbol{\mu}_{pi} - \mu_q)^2}{2\sigma_q^2} - \frac{1}{2}) \\
&= \frac{\partial}{\mu_q}\sum_{i=1}^{I}\mathbf{w}_i(\frac{\boldsymbol{\sigma}_{pi} + (\boldsymbol{\mu}_{pi} - \mu_q)^2}{2\sigma_q^2}) \\
&= \frac{\partial}{\mu_q}\sum_{i=1}^{I}\mathbf{w}_i(\frac{\boldsymbol{\sigma}_{pi} + \boldsymbol{\mu}_{pi}^2 - 2\mu_q\boldsymbol{\mu}_{pi} + \mu_q^2}{2\sigma_q^2}) \\
&= \frac{\partial}{\mu_q}\sum_{i=1}^{I}\mathbf{w}_i(-\frac{2\boldsymbol{\mu}_{pi}\mu_q}{2\sigma_q^2} + \frac{\mu_q^2}{2\sigma_q^2}) \\
&= \sum_{i=1}^{I}\mathbf{w}_i(\frac{-\boldsymbol{\mu}_{pi} + \mu_q}{\sigma_q^2})
\end{aligned} \tag{A.1}$$

$$\begin{aligned}
\frac{\partial \mathrm{KL}(p,q)}{\sigma_q} &= \frac{\partial}{\sigma_q}\sum_{i=1}^{I}\mathbf{w}_i(\frac{1}{2}log(\frac{\sigma_q}{\boldsymbol{\sigma}_{pi}}) + \frac{\boldsymbol{\sigma}_{pi} + (\boldsymbol{\mu}_{pi} - \mu_q)^2}{2\sigma_q^2} - \frac{1}{2}) \\
&= \frac{\partial}{\sigma_q}\sum_{i=1}^{I}\mathbf{w}_i(\frac{1}{2}log(\sigma_q) - \frac{1}{2}log(\boldsymbol{\sigma}_{pi})) + \frac{\boldsymbol{\sigma}_{pi} + (\boldsymbol{\mu}_{pi} - \mu_q)^2}{2\sigma_q^2}) \\
&= \sum_{i=1}^{I}\mathbf{w}_i(\frac{1}{2\sigma_q} + 4\sigma_q\frac{\boldsymbol{\sigma}_{pi} + (\boldsymbol{\mu}_{pi} - \mu_q)^2}{4\sigma_q^4}) \\
&= \sum_{i=1}^{I}\mathbf{w}_i(\frac{1}{2\sigma_q} + \frac{\boldsymbol{\sigma}_{pi} + (\boldsymbol{\mu}_{pi} - \mu_q)^2}{\sigma_q^3})
\end{aligned} \tag{A.2}$$

A.2 Shift score

$$\begin{aligned}
s_{g^*}(x,x'|\delta) &= \left|1 - \frac{p(x+\delta)p(x'+\delta)}{q(x+\delta)q(x'+\delta)}\frac{q(x)q(x')}{p(x)p(x')}\right| \\
&= \left|1 - \frac{p(x+\delta)p(x'+\delta)}{p(x)p(x')}\frac{q(x)q(x')}{q(x+\delta)q(x'+\delta)}\right| \\
&= \left|1 - \eta(x,x',\delta)\frac{q(x)q(x')}{q(x+\delta)q(x'+\delta)}\right| \\
&= |\tilde{s}_{g^*}(x,x'|\delta)|
\end{aligned} \tag{A.3}$$

with $\eta(x,x',\delta) = \frac{p(x+\delta)p(x'+\delta)}{p(x)p(x')}$ independent of q and $\tilde{s}_{g^*}(x,x'|\delta)$ the signed score. Let θ be the parameters of q and θ_α be the αth element of θ. Then the αth element of the gradient of the score w.r.t. θ is

$$\begin{aligned}
\frac{\partial s_{g^*}(x,x'|\delta)}{\partial\theta_\alpha} &= \mathrm{sign}(\tilde{s}_{g^*}(x,x'|\delta))\frac{\partial}{\partial\theta_\alpha}\left(1 - \eta(x,x',\delta)\frac{q(x)q(x')}{q(x+\delta)q(x'+\delta)}\right) \\
&= -\,\mathrm{sign}(\tilde{s}_{g^*}(x,x'|\delta))\eta(x,x',\delta)\frac{\partial}{\partial\theta_\alpha}\left(\frac{q(x)q(x')}{q(x+\delta)q(x'+\delta)}\right).
\end{aligned} \tag{A.4}$$

The last term is

$$\begin{aligned}\frac{\partial}{\partial\theta_\alpha}\left(\frac{q(x)q(x')}{q(x+\delta)q(x'+\delta)}\right) &= \frac{\partial}{\partial q(x)}\left(\frac{q(x)q(x')}{q(x+\delta)q(x'+\delta)}\right)\frac{\partial q(x)}{\partial\theta_\alpha}\\ &+ \frac{\partial}{\partial q(x')}\left(\frac{q(x)q(x')}{q(x+\delta)q(x'+\delta)}\right)\frac{\partial q(x')}{\partial\theta_\alpha}\\ &+ \frac{\partial}{\partial q(x+\delta)}\left(\frac{q(x)q(x')}{q(x+\delta)q(x'+\delta)}\right)\frac{\partial q(x+\delta)}{\partial\theta_\alpha}\\ &+ \frac{\partial}{\partial q(x'+\delta)}\left(\frac{q(x)q(x')}{q(x+\delta)q(x'+\delta)}\right)\frac{\partial q(x'+\delta)}{\partial\theta_\alpha}\\ &= \left(\frac{q(x')}{q(x+\delta)q(x'+\delta)}\right)\frac{\partial q(x)}{\partial\theta_\alpha}\\ &+ \left(\frac{q(x)}{q(x+\delta)q(x'+\delta)}\right)\frac{\partial q(x')}{\partial\theta_\alpha}\\ &- \left(\frac{q(x)q(x')}{q^2(x+\delta)q(x'+\delta)}\right)\frac{\partial q(x+\delta)}{\partial\theta_\alpha}\\ &- \left(\frac{q(x)q(x')}{q(x+\delta)q^2(x'+\delta)}\right)\frac{\partial q(x'+\delta)}{\partial\theta_\alpha}.\end{aligned} \tag{A.5}$$

Combining Eq. A.4 with Eq. A.5 the gradient is (adding dependency on θ into notation at this point which is a bit sloppy and should have been done prior to deriving the formulas)

$$\begin{aligned}\nabla_\theta s_{g^*}(\theta;x,x',\delta) = -\operatorname{sign}(\tilde{s}_{g^*}(\theta;x,x',\delta))\eta(x,x',\delta)\Big(&\frac{q(x')}{q(x+\delta)q(x'+\delta)}\nabla_\theta q(\theta;z)|_{z=x}\\ &+ \frac{q(x)}{q(x+\delta)q(x'+\delta)}\nabla_\theta q(\theta;z)|_{z=x'}\\ &- \frac{q(x)q(x')}{q^2(x+\delta)q(x'+\delta)}\nabla_\theta q(\theta;z)|_{z=x+\delta}\\ &- \frac{q(x)q(x')}{q(x+\delta)q^2(x'+\delta)}\nabla_\theta q(\theta;z)|_{z=x'+\delta}\Big).\end{aligned} \tag{A.6}$$

Appendix B

Derivation of Uncertainty Calibration Score

$$\begin{aligned} s_{UQ}(F) &= \mathbb{E}\left(\log \frac{p(F_{GP}|\mathcal{D})}{p(F_{GP}|\mathcal{D}=F)}\right) \\ &= \mathbb{E}\left(\log p(F_{GP} \mid \mathcal{D}) - \log p(F_{GP} \mid \mathcal{D}=F)\right). \\ &= \mathbb{E}\left(\log p(F_{GP} \mid \mathcal{D})\right) - \log p(F_{GP} \mid \mathcal{D}=F) \end{aligned} \tag{B.1}$$

We will now have a look at the individual terms. Recall that

$$p(F_{GP} \mid \mathcal{D}) = \mathcal{N}(F_{GP}; \mu_{\mathcal{D}}, \sigma_{\mathcal{D}}) = \frac{1}{\sqrt{2\pi\sigma_{\mathcal{D}}}} e^{-\frac{1}{2}\frac{(F_{GP}-\mu_{\mathcal{D}})^2}{\sigma_{\mathcal{D}}}} \tag{B.2}$$

and hence its logarithm is

$$\log p(F_{GP} \mid \mathcal{D}) = -\frac{1}{2}\log 2\pi\sigma_{\mathcal{D}} - \frac{1}{2}\frac{(F_{GP}-\mu_{\mathcal{D}})^2}{\sigma_{\mathcal{D}}}. \tag{B.3}$$

Its expecation is

$$\begin{aligned} \mathbb{E}\left(\log p(F_{GP} \mid \mathcal{D})\right) &= \mathbb{E}\left(-\frac{1}{2}\log 2\pi\sigma_{\mathcal{D}} - \frac{1}{2}\frac{(F_{GP}-\mu_{\mathcal{D}})^2}{\sigma_{\mathcal{D}}}\right) \\ &= -\frac{1}{2}\log 2\pi\sigma_{\mathcal{D}} - \frac{1}{2}\frac{\mathbb{E}\left((F_{GP}-\mu_{\mathcal{D}})^2\right)}{\sigma_{\mathcal{D}}} \\ &= -\frac{1}{2}\log 2\pi\sigma_{\mathcal{D}} - \frac{1}{2}\frac{\sigma_{\mathcal{D}}}{\sigma_{\mathcal{D}}} \\ &= -\frac{1}{2}\log 2\pi\sigma_{\mathcal{D}} - \frac{1}{2} \end{aligned} \tag{B.4}$$

We now consider the second term of the last line if Eq.B.1 which is simply the density evaluate at the true value F, hence

$$\log p(F_{GP} \mid \mathcal{D}=F) = -\frac{1}{2}\log 2\pi\sigma_{\mathcal{D}} - \frac{1}{2}\frac{(F-\mu_{\mathcal{D}})^2}{\sigma_{\mathcal{D}}} \tag{B.5}$$

We now plug Eqs. B.4 and B.5 into Eq. B.1

$$\begin{aligned} s_{UQ}(F) &= -\frac{1}{2}\log 2\pi\sigma_{\mathcal{D}} - \frac{1}{2} - \left(-\frac{1}{2}\log 2\pi\sigma_{\mathcal{D}} - \frac{1}{2}\frac{(F-\mu_{\mathcal{D}})^2}{\sigma_{\mathcal{D}}}\right) \\ &= -\frac{1}{2} + \frac{1}{2}\frac{(F-\mu_{\mathcal{D}})^2}{\sigma_{\mathcal{D}}} \\ &= \frac{1}{2}\left(\frac{(F-\mu_{\mathcal{D}})^2}{\sigma_{\mathcal{D}}} - 1\right) \end{aligned} \tag{B.6}$$

Bibliography

[1] Paleyes A., Pullin M., Mahsereci M., Lawrence N., and Gonzalez J. Emulation of physical processes with Emukit. *Second Workshop on Machine Learning and the Physical Sciences, Neurips*, 2019.

[2] GPy authors. GPy: A Gaussian process framework in Python. `http://github.com/SheffieldML/GPy`, august 2022.

[3] C. G. Broyden. The convergence of a class of double rank minimization algorithms. *The new algorithm*, 2(6):222 – 231, 1970.

[4] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing*, 16(5):1190 – 1208, 1995.

[5] D. Duvenaud. The Kernel Cookbook: Advice on Covariance functions. `https://www.cs.toronto.edu/~duvenaud/cookbook/`, august 2022.

[6] A. Genz. Testing multidimensional integration routines. *In Proc. of international conference on Tools, methods and languages for scientific and engineering computation*, pages 81 – 94, 1984.

[7] P. Hennig, M. A. Osborne, and H. P. Kersting. *Probabilistic Numerics: Computation as Machine Learning.* Cambridge University Press, 2022.

[8] L. V. Kantorovich. Mathematical methods of organizing and planning production. *Management science*, 6(4):366 – 422, 1960.

[9] T. Kloek and H. K. Van Dijk. Bayesian estimates of equation system parameters: an application of integration by Monte Carlo. *Econometrica: Journal of the Econometric Society*, pages 1 – 19, 1978.

[10] S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86, 1951.

[11] D. Kwiatkowski, P. C. Phillips, P. Schmidt, and Y. Shin. *Testing the null hypothesis of stationarity against the alternative of a unit root: How sure*

are we that economic time series have a unit root? Journal of econometrics, 1992.

[12] D. J MacKay. *Information theory, inference and learning algorithms.* Cambridge university press, 2003.

[13] J. A. Nelder and R. Mead. A simplex method for function minimization. *The computer journal*, 7(4):308 – 313, 1965.

[14] A. O'Hagan. Bayes–hermite quadrature. *Journal of statistical planning and inference*, 29(3):245 – 260, 1991.

[15] C. E. Rasmussen and Z. Ghahramani. Bayesian monte carlo. *Advances in neural information processing systems*, pages 505 – 512, 2003.

[16] C. E. Rasmussen and CKI. Williams. Gaussian Processes for Machine Learning. *MIT Press*, 2(3):4, 2006.

[17] N. Smirnov. On the estimation of the discrepancy between empirical curves of distributions for two independent samples. *Bulletin Mathematique de l'Univesite de Moscou*, 2(2), 1939.

[18] S. J. Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3 – 34, 2015.

Selbständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbständig und nur mit den angegebenen Hilfsmitteln angefertigt habe und dass alle Stellen, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen sind, durch Angaben von Quellen als Entlehnung kenntlich gemacht worden sind. Diese Bachelorarbeit wurde in gleicher oder ähnlicher Form in keinem anderen Studiengang als Prüfungsleistung vorgelegt.

Ort, Datum

Unterschrift