Black-box density function estimation using recursive partitioning

Erik Bodin [1] **Zhenwen Dai** [2] **Neill D. F. Campbell** [3] **Carl Henrik Ek** [4]

Abstract

We present a novel approach to Bayesian inference and general Bayesian computation that is defined through a sequential decision loop. Our method defines a recursive partitioning of the sample space. It neither relies on gradients nor requires any problem-specific tuning, and is asymptotically exact for any density function with a bounded domain. The output is an approximation to the whole density function including the normalisation constant, via partitions organised in efficient data structures. Such approximations may be used for evidence estimation or fast posterior sampling, but also as building blocks to treat a larger class of estimation problems. The algorithm shows competitive performance to recent state-of-the-art methods on synthetic and real-world problems including parameter inference for gravitational-wave physics.

1. Introduction

Bayesian methods require the computation of posterior densities, expectations, and model evidence. In all but the simplest conjugate models, these calculations are intractable and require numerical approximations to be made. In many application areas, multiple computational tasks are carried out on the same distribution or model of interest. Using traditional methods, this typically involves specialised algorithms and expertise for different tasks, reducing ease-of-use of the overall methodology, and there is often a need for many expensive computations. The individual computations can have low utilisation of past results, increasing the total computational burden and slowing down experimentation.

The shared object on which the methods and computations operate is a density function, typically being an unnormalised joint distribution of parameters and fixed data. What is preventing tractable computations is that the density function does not have an amenable functional form. An ideal functional form would allow us properties that are typically only associated with specific families of parametric distributions, such as fast proportional sampling, tractable expectations of functions, deriving conditional and marginal densities, and compute quantities like divergences. There are families of methods dedicated to approximating the density function with a function of such form. However, these methods either assume specific original functional forms, require known gradients, or can be prohibitively expensive. This is today a challenge often even for problems which are moderate in the number of random variables [1], as many applications areas entail joint distributions with difficult properties. Such areas include the physical sciences, hyperparameter inference, and active learning, such as approximating distributions of inputs to functions.

In this work, we construct approximations which are amenable for tractable computation. The approximation is a tree structure which can be computed ahead of time, and from which we can produce piecewise constant approximations amenable for respective task. We present an efficient algorithm to produce such trees, driving a recursive partitioning of the domain. The algorithm is designed for black-box settings, does not rely on gradients nor a known form, and addresses challenges such as multi-modality, discontinuous structures, and zero density regions. Our approach is asymptotically exact and has no sensitive free parameters; this leads to an algorithm that accommodates general density functions of moderate dimension, and that is easy to use. The method defines a sequential decision loop sampling the integrand, prioritising regions of high probability mass within a recursive refinement of the approximation. The decision loop has similar algorithmic efficiency as MCMC and nested sampling, with a time complexity allowing for sub-millisecond decision times and scalability to a large number of observations to obtain sufficient precision. It is competitive to recent state-of-the-art implementations of these methods for their individually supported tasks, whilst in contrast to these methods produce a *density function approximation* defined over the whole domain, useful for many down-stream tasks beyond efficient sampling and evidence estimation. This algorithm we refer to as DEnsity Function

[1]Alan Turing Institute, United Kingdom [2]Spotify, United Kingdom [3]University of Bath, United Kingdom [4]University of Cambridge, United Kingdom. Correspondence to: Erik Bodin <mail@erikbodin.com>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

[1]Such as constituting up to around ten random variables.

Estimation using Recursive partitioning or DEFER.

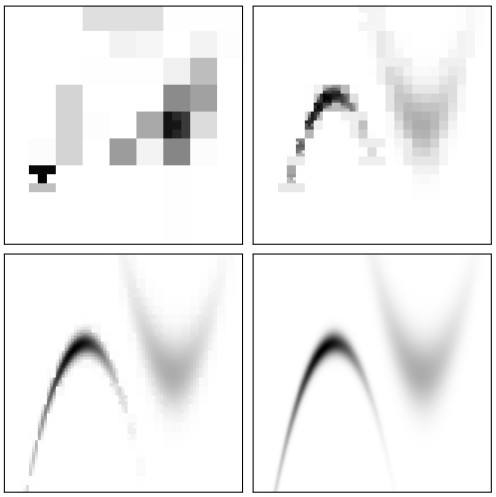


Figure 1. Partitions produced using DEFER, after 53, 303, 2 101, and 100 003 density function evaluations, respectively.

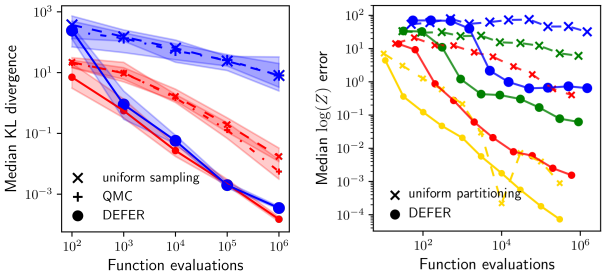


Figure 2. Shown on the left is an illustrative comparison to using standard rejection sampling for parameter inference on a Gaussian distribution in 5D, with a true scale parameter of $1/20$ and $1/100$ of the domain sides, corresponding to small and large marker sizes, respectively. The rejection methods use proposals drawn uniformly or in a Sobol sequence (Quasi-Monte Carlo), respectively. Shown on the right is a comparison to using a uniform grid for evidence estimation on the Student's t-distribution in 2D, 4D, 6D, and 10D corresponding to the markers of increasing size. The true scale parameter corresponds to $1/100$ of the domain sides. Both experiments were run 20 times with uniformly sampled true means, and the Student's t-distribution has $2.5 + (D/2)$ degrees of freedom.

2. Background

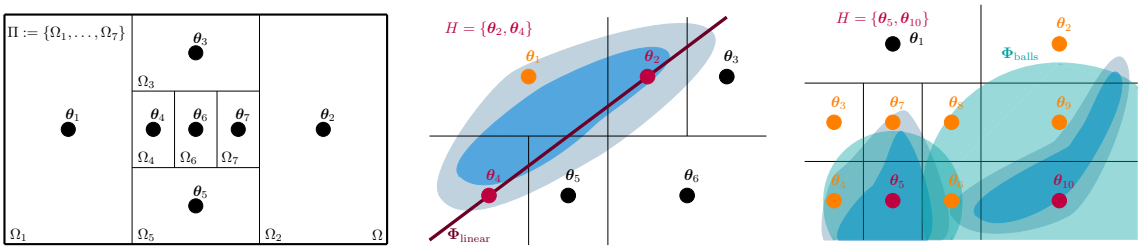
The need for substituting continuous distributions or processes with approximations exists in various forms in literature. For optimisation problems in economics (Tanaka & Toda, 2013; 2015; Farmer & Toda, 2017), and engineering (Nguyen-Hong et al., 2018; Ai et al., 2013) discrete approximations are often used, but these methods generally rely of knowing the parametric form (Miller III & Rice, 1983; DeVuyst & Preckel, 2007). Variational Inference (VI) (Blei et al., 2016; Hoffman et al., 2013) seeks a distribution approximation from a choice of variational family. However, VI typically requires differentiability and known gradients, which is not met in our setting. Acerbi (2018) combined VI with Bayesian Quadrature (O'Hagan, 1991) and a Gaussian Process (GP) surrogate to fit the variational distribution of an explicitly unknown distribution, and Jarvenpaa et al. (2020) uses a GP surrogate for Approximate Bayesian Computation (ABC). These methods are useful for scenarios where the unknown density function is significantly more expensive to evaluate than the predictive distribution of the GP surrogate, but this is typically not the case in the settings we address. To put these costs in perspective, evaluating the density function in the typical inference scenarios we address comes at a cost in the range of *milliseconds*, whilst the method in (Acerbi, 2018) takes around a *second* to decide where to evaluate already after a few evaluations. Furthermore, the time complexity of that method is $O(N^4)$, whilst we need a near-linear algorithm scaling to many thousands or millions of evaluations. Our method is, e.g., suited for the inner-loop of such methods, such as inference of the hyperparameters of the Gaussian Process model these methods use.

Monte Carlo (MC) methods approximate expectations of functions using a finite collection of samples drawn in proportion to a distribution. But the samples collected, for example via Markov Chain Monte Carlo (Geyer, 1992), only constitute a density function approximation [2] *at the collected samples*, and thus neither provide (probability) mass integral estimates, nor allow for density queries over the whole sample space Ω as we require.

One of our method's main requirements is flexibility and the ability to handle unknown distributions with little to no specification. This requirement we can fulfil through having the property of asymptotic exactness. In other words, that the approximation tends towards the true distribution and that given enough computational budget, sufficient precision may be achieved. This is to circumvent the need to specify distribution-specific details upon the usage of the method; which would be difficult, as the (true) distribution is assumed explicitly unknown and complicated, in that it may have strong correlations, be discontinuous, multi-modal, or have zero density regions of complex shapes.

To achieve asymptotic exactness, we take inspiration from quadrature, which is a classic approach to estimating an integral numerically over a bounded domain. These methods partition the domain and convert the integral into a weighted summation of integrands at each partition. By creating a partitioning of the full domain, quadrature methods provide asymptotic guarantees by design. Furthermore, quadrature methods can be robust to the characteristics of the function

[2] If the associated density values are saved.



(a) Partitioning definitions (b) Locally linear structure (c) Local neighbourhoods

Figure 3. Illustration of partition division. (a) The domain Ω is divided according to a *partitioning* Π composed of partitions $\{\Omega_i\}$ with corresponding centroids $\{\boldsymbol{\theta}_i\}$. (b) Locally linear structure: High mass partitions $H = \{\boldsymbol{\theta}_2, \boldsymbol{\theta}_4\}$ define a linear subspace $\boldsymbol{\Phi}_{\text{linear}}$ used to find candidates for division (e.g. $\boldsymbol{\theta}_1$). (c) Local neighbourhoods: High mass partitions $H = \{\boldsymbol{\theta}_5, \boldsymbol{\theta}_{10}\}$ define neighbourhood D-balls $\boldsymbol{\Phi}_{\text{balls}}$ used to find candidates for division (e.g. $\boldsymbol{\theta}_6, \boldsymbol{\theta}_9, \ldots$). The true density is shown in blue.

surface, such as discontinuities and zero density regions, as it is the domain partitioning that drives the asymptotic behaviour. Just as quadrature is asymptotically exact in the limit of decreasing size of the largest partition, we can similarly achieve this guarantee by ensuring the iterative partitioning algorithm will eventually divide all current partitions. The downside of the characteristic that provides the asymptotic guarantee is the curse of dimensionality (Gander & Gautschi, 2000; Smolyak, 1963; Gerstner & Griebel, 1998; Hewitt & Hoeting, 2019), where the number of partitions in a grid of fixed resolution grows exponentially with the number of dimensions. We will design the algorithm with this in mind, prioritising regions to refine into finer partitions iteratively. For a visual depiction of a partitioning, see Figure 1. This is especially important as the distribution's typical set may be concentrated to tiny regions, which would make an equidistant grid computationally infeasible. We illustrate this in Figure 2. In the figure, we also demonstrate the common lack of efficiency of using standard rejection sampling for posterior sampling, a method that otherwise provides much flexibility. The efficiency problem arises in typical realistic scenarios, where the likelihood function causes the typical set of the posterior to be concentrated to small regions. As illustrated, uninformed proposal distributions become infeasibly inefficient already in low dimension (5D in the example) when the posterior's typical set is even just moderately small. This is simply a consequence of the vast number of proposals needed on average per proposal landing in the typical set ($\approx 10^7$ in the example).

We are not the first to recognise that a tree-based partitioning of the whole domain has several desired properties. (McCool & Harwood, 1997) proposed using kd-trees to represent distributions, and (Lu et al., 2013) proposed using partitionings in Bayesian modelling and developed a prior distribution over partitions and a corresponding inference method. (Li et al., 2016) developed an algorithm to construct piecewise constant density function approximations from iid samples based on discrepancy. All these methods assume that we have samples drawn from the true distribution and wish to reconstruct the distribution. As such they address a different problem domain, one where samples are known a-priori. We do not have access to samples drawn from the distribution, but are instead provided the unnormalised density function. In our approach, we evaluate the density function as a part of an active sampling loop, where at each iteration we pick locations in the sample space for which to evaluate the density function. We will now proceed with our methodology.

3. Methodology

Let $f : \Omega \to \mathbb{R}_+$ be an explicitly unknown, unnormalised density function defined on a hyperrectangular sample space $\Omega \subset \mathbb{R}^D$, which can be evaluated for any $\boldsymbol{\theta} \in \Omega$ [3] . Our goal is to construct a representation of the function that enables the tractability of downstream tasks, such as fast, constant-time proportional sampling, evidence estimation, tractable expectations of functions, deriving conditional and marginal densities, and compute quantities like divergences. A density function implies a distribution

$$\mathcal{P}_f(\boldsymbol{\theta}) = \frac{f(\boldsymbol{\theta})}{\int_{\boldsymbol{\theta}\in\Omega} f(\boldsymbol{\theta}) d\boldsymbol{\theta}} = \frac{f(\boldsymbol{\theta})}{Z}, \tag{1}$$

where Z constitutes the unknown normalising constant. We will approximate both the function f and the normalising constant Z.

Typically f is formed via a joint distribution of data $\mathcal{D}$ and parameters $\boldsymbol{\theta}$, where we refer to Z as the *evidence* and the distribution as the *posterior*. In other words,

$$f(\boldsymbol{\theta}) := p(\mathcal{D}|\boldsymbol{\theta})\mathcal{P}(\boldsymbol{\theta}) = Z\mathcal{P}(\boldsymbol{\theta}|\mathcal{D}), \tag{2}$$

where $\mathcal{P}(\boldsymbol{\theta})$ and $p(\mathcal{D}|\boldsymbol{\theta})$ denotes the prior distribution and

[3] In practice, a function with a complicated non-convex domain may be treated by placing it within a hyperrectangle and assume that f evaluates to 0 everywhere it is not defined. Distributions with infinite support may be approximately treated if it is possible to enclose the typical set within the bounds of the hyperrectangle.

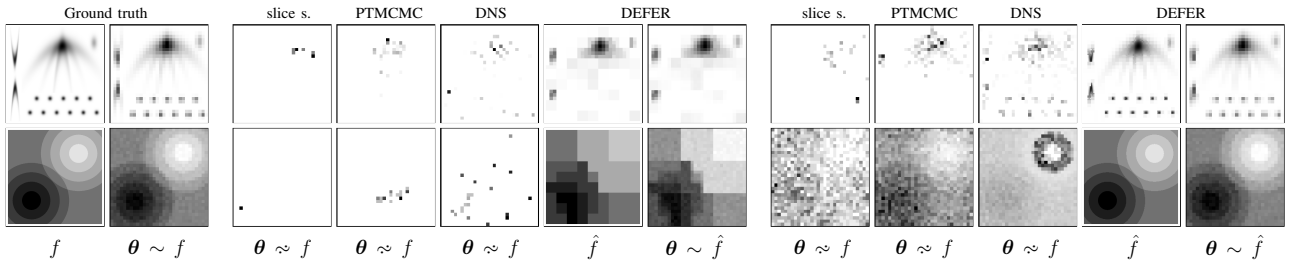


Figure 4. Robustness to characteristics in the density surface. In the upper row, a surface with heavy correlations and multi-modality is shown, and in the lower row a surface with discontinuities and a valley. The output of the methods is shown after approx. 150 (left) and 1k (right) density evaluations for the first function, to illustrate the relative inefficiency of MCMC in capturing modes. The same is shown after approx. 30 and 100k evaluations for the second function, to illustrate the early coarse approximation by DEFER as well as its asymptotic exactness in contrast to DNS. As different to the other methods, DEFER outputs a function $\hat{f}$.

the likelihood function, respectively.

3.1. Representation of $\hat{f}$

The combination of two data structures will represent our approximation $\hat{f}$. Firstly, a non-overlapping partitioning $\Pi := \{\Omega_i\}$ of the domain Ω in an array, with an observed value of f at each respective centroid $\boldsymbol{\theta}_i$ of the corresponding partition Ω_i, that represents a Riemann sum over the domain with hyper-rectangular partitions of non-constant side lengths as illustrated in Figure 1 and Figure 3. An estimate of the normalisation constant over the domain or a subdomain is obtained by summation of the masses $\{V_i f(\boldsymbol{\theta}_i)\}$ of the partitions within, where V_i is the volume of a partition with index i. Secondly, a tree-structure in which these partitions are organised, forming a search tree over volumetric objects. Together, these data structures permit integrals to be approximated by summation as in a quadrature rule, density queries of $\boldsymbol{\theta}$ or partitions by tree search, and constant-time sampling. Constant-time sampling will be achieved by the following. First, sample the index of a partition in constant-time using the alias method for categorical distributions (Kronmal & Peterson Jr, 1979), and then sample uniformly within the partition [4] . We will later demonstrate that, given such a representation, many quantities and constructs can be estimated within the same framework, including conditional and marginal distributions.

3.2. Iterative construction requirements of $\hat{f}$

We now consider how to obtain a sufficiently close approximation to f in an efficient manner. We address this in three ways. Firstly, we prioritise where in the domain Ω the approximation should be refined. Secondly, we allow for a large number of partitions. Lastly, we provide guarantees that the algorithm asymptotically approaches f. In a non-overlapping space partitioning, the first requirement translates into a decision-problem over which partitions to divide at a given step in sequence. The second requirement translates into making the decisions in an efficient manner, with an algorithmic complexity that allows for fast decisions that remain fast also for a large number of partitions. And lastly, for the asymptotic behaviour, guaranteeing that all partitions will eventually be divided.

[4]Note that constant time sampling is only available after the construction of the approximation, as the alias method requires linear time pre-processing. Without such pre-processing, logarithmic time sampling is available by tree-search.

4. Algorithm

Algorithm 1 DEFER

Input: General density function f defined over Ω with unknown normalisation constant Z.
Output: Approximation $\hat{f}$, $\hat{Z}$, as specified by the produced partitioning Π_T.
Initialize t = 1 and initial partitioning $\Pi_1 = \{\Omega\}$
repeat *# makes density acquisitions at each iteration*
  $\{\Omega_i\} =$ to_divide $[\,\Pi_t]$
  divide each partition Ω_i, each one resulting in $\{\Omega_j\}$ new partitions
  add all sets of $\{\Omega_j\}$ into Π_t
  remove the divided partitions $\{\Omega_i\}$ from Π_t
  set $t \rightarrow t + 1$ and update data structures
until $N_t \geq N_{\text{max}}$

Given the representation and requirements, we construct the approximation through an iterative refinement procedure of the current tree-structured domain partitioning. The algorithm starts from the base case of the whole sample space being one partition, with its density evaluated at the centre. A subset of the existing partitions will be selected at each iteration to be divided further, as specified by a few criteria, which we will address later. Note that the base case will always be divided as *at least* one partition will always be divided at each iteration. When a partition is divided, it will

result in new partitions according to a division procedure, where each new partition will receive an associated density by evaluating f at the partition centre, except for the centre partition which will inherit it from the divided (parent) partition. The new partitions will subsequently be incorporated into a few data structures (and the divided partition will be excluded), including the tree-structured partitioning, as well as into some data structures to enable the division criteria to be efficiently checked for all partitions in subsequent iterations. The outline of the algorithm is shown in Algorithm 1. The set of current partitions at a given iteration t in the sequence of decisions is denoted by Π_t. We define $N_t := |\Pi_t|$ to be the number of partitions at iteration t. At each iteration, we divide *all* partitions in Π_t that meet *any* of three criteria, to produce an updated set of partitions Π_{t+1} (referred to as `to_divide` in Algorithm 1). Note that only one criterion needs to be met for division of a given partition, and multiple partitions may be divided at each iteration.

Partition division routine We will now describe the division routine, which is carried out on a given partition after it has been decided to be divided. A partition division entails dividing the partition into three sub-partitions with equal side-lengths for each divided dimension. The set of dimensions to divide is the set of dimensions of maximum length for the partition, following normalisation of the domain Ω to a unit hyper-cube to avoid favouring dimensions spanning a wider value range. An illustration is shown in Figure 3 (a), where the initial partition (the whole domain Ω) was in the first iteration divided horizontally followed by vertically, and in the second iteration the centre partition was selected to be divided further, and divided horizontally. The divide order of the dimensions is the same as the descending order of the highest observed function value, in common with (Jones et al., 1993). In other words, after determining all the child partitions centroids and evaluating the density at those locations, associate each dimension with the highest density value observed along that dimension and rank the dimensions accordingly. Note that the centroid position do not depend on the dimension divide order even though the partition bounds do, allowing the function to be evaluated before the forming of the partitions. The forming of each new partition, with centroid $\boldsymbol{\theta}_j$, entails evaluating the true density function and storing the density value $f(\boldsymbol{\theta}_j)$ together with the partition boundaries. Note that each partition division results in a variable number of new partitions, depending on the number of dimensions being divided.

Search-tree To form the search-tree described in Section 3, we store each partition Ω_i together with its associated density function observation $f(\boldsymbol{\theta}_i)$ in a node. When a partition is divided (see Algorithm 1), a set of child nodes is created and stored in an array within the parent node. Each partition (Ω_i) is represented using two arrays; with the lower and upper bounds for each dimension, respectively. A query for the unique *leaf* node (and partition) that is associated with a given $\boldsymbol{\theta}$ can be performed by traversing the tree from the root. This is a consequence of the partitions associated with child nodes of each non-leaf node being non-overlapping, and their union being equal to the partition of their parent.

4.1. Partition division decision criteria

We now detail the three individually sufficient criteria for division, that we denote CR1 to CR3. For the first criterion, with the aim of robustness to degeneracies in f and to maintain an informed and efficient exploration, we will re-interpret (Jones et al., 1993) which proposed an approach to avoiding explicit assumptions of the Lipschitz constant in the context of global optimisation. In (Jones et al., 1993) the domain of the function to be optimised was iteratively split into finer partitions, where partitions to be divided had the maximum upper bound function value under *any* possible maximum rate of change (Lipschitz constant) between zero and infinity. We will translate and adapt this idea to our context of mass-based prioritisation over the whole function. To ensure the guarantee of the approximation's asymptotic exactness, at least one criterion will need to be eventually fulfilled for *all* partitions, see Section 2. This first sufficient criterion (CR1) will fulfil this. We will also derive two complementary criteria that exploit typical density functions structures, motivated for increased sample-efficiency in handling strong correlations.

4.1.1. Sufficient partition division criterion 1

One could imagine simply dividing the partition which currently has the largest estimated mass $V_k f(\boldsymbol{\theta}_k)$, where V_k is the volume of the partition Ω_k, and $f(\boldsymbol{\theta}_k)$ is the associated density value evaluated at its centroid $\boldsymbol{\theta}_k$. However, this would not consider that the density at the centroid $\boldsymbol{\theta}_k$ typically is less representative for the partition the larger the partition is. The purpose of this criterion will be to prioritise the division of partitions by their *potential for* (true) mass. An approach that may easily come to mind is to fit a function model to the density values and partition centroids to assess how the density function may behave outside the evaluated centroids. However, fitting a function model, as well as taking the function model into account, comes at a computational cost. Even if the function model fitting would come at constant cost per added partition with respect to the number of partitions, which it typically does not, considering all partitions under the model at a given iteration would at best come at a linear cost. A linear cost per decision would result in a quadratic time algorithm that will not scale well to millions of partitions. Moreover, an interpolating function model implies a function with no discontinuities, which we cannot assume in our setting. Instead,

we will derive a rule that allows us to consider all partitions and density evaluations simultaneously while carrying out decisions in *logarithmic* time with respect to the number of partitions so far. Furthermore, the rule will handle discontinuities and rapidly changing surfaces and does not require an assumption or prior of the Lipschitz constant.

Criterion The criterion (CR1) is the following. A partition Ω_k will be divided if there exists *any* rate-of-change constant [5] $\bar{K} > 0$ such that

$$V_k \cdot \left(f(\boldsymbol{\theta}_k) + \bar{K}\frac{d_k}{2} \right) \geq V_i \cdot \left(f(\boldsymbol{\theta}_i) + \bar{K}\frac{d_i}{2} \right), \quad (3)$$

$$\text{and} \quad V_k \cdot \left(f(\boldsymbol{\theta}_k) + \bar{K}\frac{d_k}{2} \right) \geq \beta \frac{\hat{Z}}{N_t + 1}, \quad (4)$$

for all $\forall\, i \in [1, N_t]$, where $\hat{Z} = \sum_{k=1}^{N_t} \hat{Z}_k = \sum_{k=1}^{N_t} V_k \cdot f(\boldsymbol{\theta}_k)$ is the normalising constant of the approximation, V_k is the volume of the partition Ω_k, $\boldsymbol{\theta}_k$ is the centroid of the partition, and d_k is the diameter of the partition $d_k = \sup_{\boldsymbol{\theta}_i, \boldsymbol{\theta}_j \in \Omega_k} ||\boldsymbol{\theta}_i - \boldsymbol{\theta}_j||$. Here, β is a positive parameter specifying what constitutes a non-trivial amount of *upper bound mass* of a partition in relation to the current estimate of average mass; we fix $\beta = 1$ for all the experiments in this work. Note that β only controls precision relative to the average partition mass contribution so far, and can in general be left at default.

The first statement (Equation 3) is true if the partition of index k has an upper bound on its true mass greater or equal to the upper bound of all the partitions under *any* choices of $\bar{K} > 0$. The second statement (Equation 4) is true if the partition, under the corresponding choice of $\bar{K}$, has an upper bound on the true mass that constitutes a non-trivial amount of mass.

Implementation We check the first statement (Equation 3) using the following. After construction of a partition Ω_j, with parameters $(f(\boldsymbol{\theta}_j), \boldsymbol{\theta}_j \in \Omega_i)$, we map it to ordinate $V_j \cdot f(\boldsymbol{\theta}_j)$ and abscissa $V_j \cdot {}^{d_j}/_2$ in a 2D space, see Figure 5. The criterion will be fulfilled for a partition Ω_k *if and only if* its corresponding coordinate is a member of the upper-right quadrant of the convex hull (URQH) in this 2D space.

The ordinate and abscissa of each partition is stored in a hash map of heaps, i.e. where each entry of a hash table maps to a heap (Williams, 1964) as its associated value. This data structure map is used to efficiently provide access to the *maximum* ordinate partition per unique abscissa at every iteration t, which are kept sorted using the individual heaps, allowing fast updates. These partitions are the only ones that have corresponding coordinates that *may* be part of the URQH described, and the only ones that need to be a part of a convex hull calculation. The number of unique abscissas we will demonstrate is small and near constant with respect to the number of partitions.

[5] Note that this is not an (explicit) Lipchitz constant assumption, as we simultaneously consider all possible $\bar{K} > 0$.

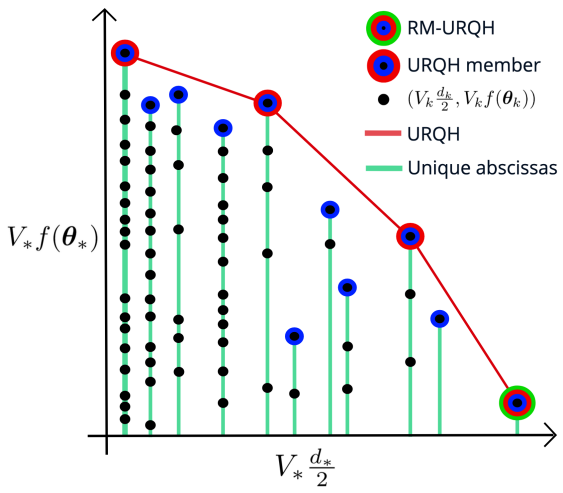


Figure 5. Partitions and their associated density values are mapped to points in a 2D space for which the only partitions that can fulfil CR1 are also a member of the upper right quadrant of the convex hull, shown as the red line. Moreover, the only partitions whose points needs to be a part of the convex hull calculation are the ones that have the highest ordinate $V_j \cdot f(\boldsymbol{\theta}_j)$ (mass) per unique abscissa (green), shown with blue rings, which significantly reduces the needed computation.

The second statement (Equation 4) is checked by considering the *upper bound* of the $\bar{K}$ that a given partition Ω_k used to fulfil the first statement. This would be the $\bar{K}$ that, were it any larger, would result in the right hand neighbour on the URQH having a larger upper bound of the mass $V_{i+1} \cdot (f(\boldsymbol{\theta}_{i+1}) + \bar{K}\frac{d_{i+1}}{2}) > V_i \cdot (f(\boldsymbol{\theta}_{i+1}) + \bar{K}\frac{d_i}{2})$, which would violate Equation 3. In other words, we have $\bar{K}_i^{\text{upper}} := 2\frac{V_{i+1} f(\boldsymbol{\theta}_{i+1}) - V_i f(\boldsymbol{\theta}_i)}{V_i d_i - V_{i+1} d_{i+1}}$, except for the right-most member which will have positively infinite upper-bound on $\bar{K}$. As such, statement two is fulfilled if $V_i \cdot (f(\boldsymbol{\theta}_i) + \bar{K}_i^{\text{upper}} \frac{d_i}{2}) \geq \beta \frac{\hat{Z}}{N_t + 1}$ is true.

Fulfilment of asymptotic guarantee As of Equation 3 and 4, it is clear that the right-most member of URQH (RM-URQH) will always be divided as its upper bound is positively infinite. Note that this is true for any (finite) choice of β. Furthermore, as the RM-URQH will always be among the partitions with the largest diameter, the largest diameter will asymptotically tend to zero. As a consequence, all partitions will eventually be divided, which in turn guarantees asymptotic convergence to f as of a partitioning of the whole domain constitutes a Riemann sum.

4.1.2. EXPLOITING CHARACTERISTICS OF DENSITY FUNCTIONS (CR2 AND CR3)

We now add search behaviour to exploit desirable heuristics targeting both locally linear correlations and spatial proximity in Ω.

Set of high mass partitions We define H as a set of high (relative) mass partitions at the current iteration. A partition Ω_j will be a member of this set when

$$\hat{Z}_j \geq \hat{Z}_M \quad \text{and} \quad \hat{Z}_j \geq \alpha \frac{\hat{Z}}{N_t + 1} \tag{5}$$

subject to the constraint that $1 < |H| \leq M$. The value $\hat{Z}_M$ is defined to be the mass of the partition of the M^{th} highest mass, $\hat{Z}_* = V_* \cdot f(\boldsymbol{\theta}_*)$, and α is a positive parameter specifying what constitutes a *large* (outlier) mass ratio relative to the average partition. In practice, we set $M = \min(5, D)$ and fix $\alpha = 20$ for all experiments in this work [6].

CR2: Sufficient partition division criterion 2 See Figure 3 for intuition. Existing partitions with a high estimated mass, relative to other partitions, are grouped in the set H and denoted with red centroids. Were we to assume (at least some) linear correlation between the centroids in H, we may also assume that high mass is more likely to concentrate along an affine subspace defined by linear combinations of the centroids denoted Φ_{linear}, illustrated by the maroon line. Partitions intersecting this $(|H| - 1)$-dimensional hyperplane are candidates for division.

Given the set H of high mass partitions, we construct a finite set of *representer points* $\mathbf{R}_{\text{linear}}$ based on the affine subspace constructed from linear combinations of the centroids of the partitions in H that we denote Φ_{linear}. We address the representer points in the supplement. A partition Ω_k will be divided if any point $\mathbf{r} \in \mathbf{R}_{\text{linear}}$ falls within the partition. That is, the set $\{\mathbf{r} \mid \mathbf{r} \in \Omega_k, \mathbf{r} \in \mathbf{R}_{\text{linear}}\}$ is not empty.

CR3: Sufficient partition division criterion 3 We will now address the neighbourhoods of the H partitions, illustrated in Figure 3. The figure shows that we consider spatial proximity to elements of H through sets of D-dimensional balls Φ_{balls} centred on elements of H. We define Φ_{balls} as the union of the interiors of D-dimensional balls centred at respective centroid of the partitions H. The diameter of a given D-ball corresponding to partition $\Omega_j \in H$ is $\phi\, d_h$, where ϕ is constrained to be $\phi > 1$ to guarantee that positions *outside* of Ω_j exists in all directions from $\boldsymbol{\theta}_j$. A given ϕ leads to a volume ratio ν of Ω_j relative to the ball as $\nu = V_j \cdot \Gamma(D/2+1) / (\sqrt{\pi} \frac{\phi d_j}{2})^D$. In practice we fix $\phi = 1.2$ in all experiments in this work, which we found works well empirically with little to no benefit of targeted tuning.

We take the set H of high mass partitions and construct a discrete set $\mathbf{R}_{\text{balls}}$ of additional representer points based on spatial proximity to the centroids in H, denoted Φ_{balls}, and check the condition analogously to CR2.

Summary We have now addressed the criteria CR1 to CR3, describing the full algorithm Algorithm 1. The combined time complexity of a step t is $\mathcal{O}(\log N_t + U \log U)$, where U is the number of *unique* abscissas (see Section 4.1.1). For an average number of steps proportional to N_T this results in a total average time complexity of the algorithm as $\mathcal{O}(N_T(\log N_T + \bar{U} \log \bar{U}))$. The space complexity (including storage of partitions) of the algorithm is linear, i.e. remains proportional to the number of observations. See the supplement for the analysis. We will show empirically that $\bar{U}$ is sufficiently small, and close to constant with respect to N_T, leading to a fast and scalable algorithm.

5. Experiments

As discussed in Section 1, the focus of this work is to produce approximations to density functions with support over the whole sample space, which in turn can be used for a number of down-stream tasks. However, it is crucial that the presented algorithm produces approximations that have good quality whilst being computationally efficient. We will begin this section by assessing this using comparison to modern approximate inference methods, and we will later provide examples of the larger set of down-stream tasks we may use the new method for. For an ablation study of criterion (CR) 1 to 3, and further setup or problem details, see the supplement.

We choose dynamic nested sampling (DNS) (Higson et al., 2019), slice sampling (Neal, 2003), and parallel-tempering MCMC (PTMCMC) (Ellis & van Haasteren, 2017) as the baselines because they are easy-to-use or able to handle multi-modality and work in the absence of gradient information. As of the capability of DNS to also estimate evidence, and to be robust to multi-modal and degenerate posteriors with relatively little tuning, it has the largest overlap in features with DEFER whilst being a recent and strong baseline. We will therefore make more elaborate comparisons with this method. Note, however, that DNS does not support proxy density queries, nor other still missing features to be addressed later. PTMCMC and slice sampling neither support mass integration nor proxy density queries (see Section 2), but is compared with when applicable for reference.

[6] This setting we found to work well empirically, but we did not observe a large sensitivity to the choice of α. We also observed that $\hat{Z}_j$ is typically either similar to the average (as of CR1 aiming to keep true masses roughly equal), or very much larger than the average. Thus $\alpha = 20$ is mainly set to not trigger the rule when CR1 already manages to prioritise among the partitions effectively.

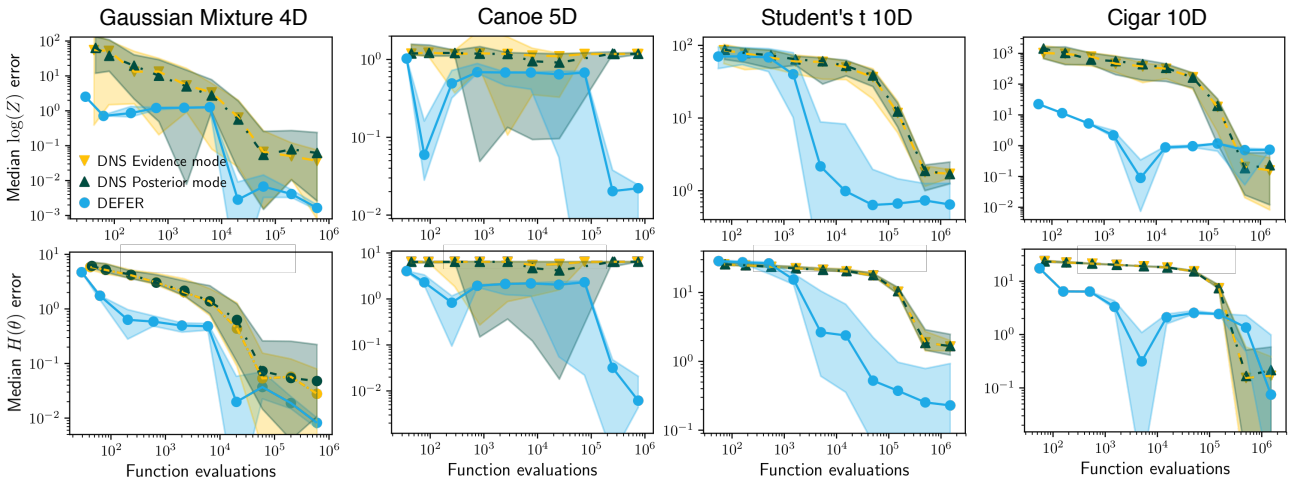


Figure 6. Upper row: Median absolute error of $\log Z$. Lower row: Median absolute error of the entropy $H(\boldsymbol{\theta})$. For both metrics, shaded areas are the 95% CI of the median.

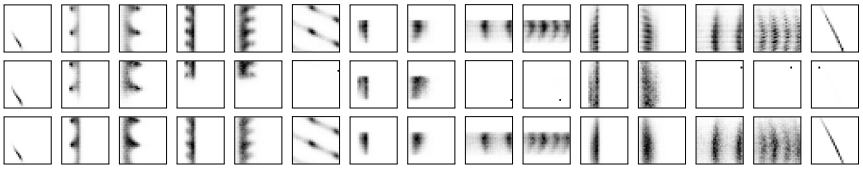


Figure 7. Parameter estimation for gravitational-wave physics. Shown are histograms of the two-dimensional marginals of a 6D parameter inference problem (Ashton et al., 2019), with samples produced after 5M density function evaluations. Upper row: from the DEFER approximation. Second row: using PTMCMC. Bottom row: using DNS.

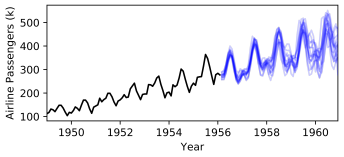


Figure 8. Time-series forecasting using Gaussian Processes with the Spectral Mixture kernel. The data is shown in black, and posterior GP samples in blue.

5.1. Approximation quality and sample-efficiency

We first test the robustness of the inference methods against challenging properties (see Figure 4) including multi-modality, strong correlations, and discontinuities, on low dimensional examples that are easy to visualize. We illustrate the asymptotic guarantees of DEFER in the presence of such properties in the density surface. We also note its sample-efficiency in representing all modes when compared to MCMC, and in capturing detail when compared to both DNS and the MCMC methods. In one of the shown examples DNS fail to recover the ground truth, which may be related to the algorithm's determination of likelihood contours (Higson et al., 2019).

We compare with DNS quantitatively both for evidence and parameter estimation. For evidence estimation, we measure the median absolute error in log evidence (or $\log Z$) over 20 runs with various budgets. For parameter estimation, we measure the error in estimated differential entropy, due to lack of summary statistics for multi-modal distributions. Results for a few functions with various challenging characteristics are shown in Figure 6, such as very small or elongated typical sets (see supplement for the functions). We remind the reader that although the surface of the approximation always tends towards the true surface, integral estimates can oscillate slightly as local overestimations and underestimations can cancel out to some degree. DEFER is able to match and typically significantly surpass the performance of DNS. Especially in getting close to the solution already at low N_T budgets, sometimes needing order of magnitudes fewer function evaluations, but it also performs better using the higher N_T budgets. The gaps between the methods are noticeably larger for estimating differential entropy. Estimation of entropy is more sensitive to the matching of the shape of a distribution, and not only its integral. On Canoe, DNS completely misses the concentrated mass, but DEFER finds it after 100k evaluations, which leads to a sharp drop in the error of both evidence and differential entropy.

5.2. Real-world density sufaces

Synthetic distributions like the ones addressed are convenient to use for assessment as they have known properties. However, it is important to confirm that DEFER can be applied to real-world distributions and density functions. To do this, we will apply DEFER on a parameter estimation task from gravitional-wave (GW) physics research (Abbott et al., 2019; Collaboration et al., 2020), and a hyperparameter inference task from (Wilson & Adams, 2013).

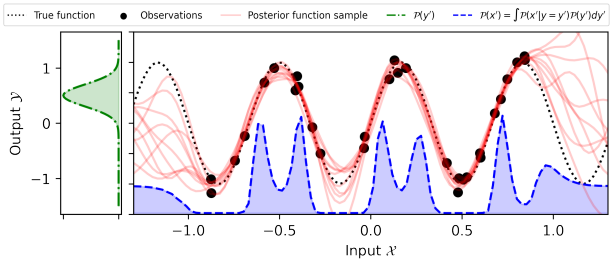


Figure 9. Input distributions. Shown in pink are function (f) samples from posterior Gaussian Processes, with hyperparameters $\boldsymbol{\theta}$ ancestrally sampled from the posterior $\mathcal{P}(\boldsymbol{\theta}|\mathcal{D})$. $\mathcal{P}(x')$ represent the distribution of the input which generated y', where $y' := f(x')$ and $y' \sim \mathcal{P}(y')$. DEFER makes it possible to derive an approximation to this input distribution by a combination of capabilities. The free-form approximation produced is flexible enough to *represent* the multi-modal input distribution. Furthermore, we are able to estimate the intractable integral $\mathcal{P}(x') \propto \int \mathcal{P}(y = y'|x)\mathcal{P}(x)\mathcal{P}(y')dxdy'$ for each $x' \in \mathcal{X}$. $\mathcal{P}(x)$ is here set to be uniform over the range of the figure.

In GW research physically motivated likelihood functions and priors (Kalaghatgi et al., 2020) are used, often without gradients available, and the induced density surfaces are typically complicated, multi-modal, or discontinuous. Inference on these problems is often prohibitively slow, warranting actions such as re-sampling, density re-weighting and local density integration. DEFER outputs a density function approximation with support for all these tasks making use of the domain-indexed search tree over partitions. We apply DEFER to a simulated signal example from (Ashton et al., 2019). Figure 7 shows all the 2D marginals of a 6D problem using the 'IMRPhenomPv2' waveform approximant. Inferred parameters are, for example, the luminosity distance, and the spin magnitudes of binary black-holes. We note the complicated interactions between parameters, showing the importance of handling multi-modality and strong correlations. Importantly, DEFER is able to handle the surface well without any tuning parameters. As PTMCMC, DEFER asymptotically approaches the unknown ground truth surface. See the supplement for additional GW experiments.

We apply DEFER to a Gaussian Process time-series regression model with a spectral mixture kernel (SMK) (Wilson & Adams, 2013) to infer the posterior of hyperparameters of the kernel, which is known to be heavily multi-modal and complicated. With a budget of 50k function evaluations, the negative log-likelihoods on the test data are 377.66, 365.97, 236.89 and **205.50**, for slice sampling, PTMCMC, DNS and DEFER, respectively. For predictions using DEFER, see Figure 8. See supplement for further details and plots.

5.3. Runtime performance

In the supplement we confirm both that DEFER has a similar algorithmic cost to the other methods and that the DEFER has a near-constant cost per function evaluation with respect to N_T. With algorithmic cost, we refer to the cost per 'decision' of where to evaluate the density function, which is computed from the total (wall-clock) time of inference minus the total function evaluation time, divided by the number of function evaluations made. In practice, using our implementation of DEFER and setup, the decision time per evaluation were around 0.3 to 0.5 milliseconds also after millions of density evaluations.

5.4. Down-stream task and application examples

We are now finished with the empirical comparisons, confirming the quality of the resulting approximations and the efficiency of DEFER to construct them. Importantly, we have treated density functions in a way that lets us be agnostic to the problem giving rise to the density function. The approximation produced is a tree from which we can produce a piecewise constant function defined over the domain or any axis-aligned hyper-rectangular subdomain. This way we can transform a myriad of problems involving intractable integrals, such as density marginalisations, or deriving conditional distributions, into queries of the approximation and summation.

For example, consider the problem of estimating mutual information $I(\boldsymbol{\theta}_a; \boldsymbol{\theta}_b) = \mathbb{E}_{\mathcal{P}(\boldsymbol{\theta}_a, \boldsymbol{\theta}_b)}[\log \frac{\mathcal{P}(\boldsymbol{\theta}_a, \boldsymbol{\theta}_b)}{\mathcal{P}(\boldsymbol{\theta}_a)\mathcal{P}(\boldsymbol{\theta}_b)}]$. Normally this would require a specialised algorithm, but we may now instead estimate it by using DEFER to approximate the joint $\mathcal{P}(\boldsymbol{\theta}_a, \boldsymbol{\theta}_b)$, and then use the capabilities of the approximation. Another example is to propagate uncertainty, such as deriving the distribution of the input to an uncertain function producing an uncertain output, see Figure 9. We may also apply DEFER to multiple distributions, allowing us to estimate divergences.

We provide code, and examples for how to use DEFER for these applications and more, at https://github.com/bodin-e/defer.

6. Conclusion

We have presented a new approach to general Bayesian computation and approximate inference based on recursive partitioning. The approach is shown to be competitive to state-of-the-art methods on black-box density functions, with high sample-efficiency and scalability, allowing complicated surfaces to be estimated with high precision. The algorithm produces a function representation that can be used for a diverse set of tasks beyond the computation of evidence and sampling, and is flexible and easy to use.

Acknowledgments Supported by the Hans Werthn Fund at The Royal Swedish Academy of Engineering Sciences, UKRI CAMERA project (EP/M023281/1, EP/T022523/1) and the Royal Society.

References

Abbott, B., Abbott, R., Abbott, T., Abraham, S., Acernese, F., Ackley, K., Adams, C., Adhikari, R., Adya, V., Affeldt, C., et al. Gwtc-1: A gravitational-wave transient catalog of compact binary mergers observed by ligo and virgo during the first and second observing runs. *Physical Review X*, 9(3):031040, 2019.

Acerbi, L. Variational bayesian monte carlo. In *Advances in Neural Information Processing Systems*, pp. 8213–8223, 2018.

Ai, X., Wen, J., Wu, T., and Lee, W.-J. A discrete point estimate method for probabilistic load flow based on the measured data of wind power. *IEEE Transactions on Industry Applications*, 49(5):2244–2252, 2013.

Ashton, G., Hübner, M., Lasky, P. D., Talbot, C., Ackley, K., Biscoveanu, S., Chu, Q., Divakarla, A., Easter, P. J., Goncharov, B., et al. Bilby: A user-friendly bayesian inference library for gravitational-wave astronomy. *The Astrophysical Journal Supplement Series*, 241(2):27, 2019.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: a review for statisticians. *CoRR*, 2016. URL http://arxiv.org/abs/1601.00670v9.

Collaboration, L. S., Collaboration, V., et al. Gw190412: Observation of a binary-black-hole coalescence with asymmetric masses. *arXiv preprint arXiv:2004.08342*, 2020.

DeVuyst, E. A. and Preckel, P. V. Gaussian cubature: A practitioner's guide. *Mathematical and Computer Modelling*, 45(7-8):787–794, 2007.

Ellis, J. and van Haasteren, R. jellis18/ptmcmcsampler: Official release, October 2017. URL https://doi.org/10.5281/zenodo.1037579.

Farmer, L. E. and Toda, A. A. Discretizing nonlinear, non-gaussian markov processes with exact conditional moments. *Quantitative Economics*, 8(2):651–683, 2017.

Gander, W. and Gautschi, W. Adaptive quadrature—revisited. *BIT Numerical Mathematics*, 40(1):84–101, 2000.

Gerstner, T. and Griebel, M. Numerical integration using sparse grids. *Numerical Algorithms*, 18:209–232, 1998.

Geyer, C. J. Practical markov chain monte carlo. *Statistical science*, pp. 473–483, 1992.

Hewitt, J. and Hoeting, J. A. Approximate bayesian inference via sparse grid quadrature evaluation for hierarchical models. *CoRR*, 2019. URL http://arxiv.org/abs/1904.07270v1.

Higson, E., Handley, W., Hobson, M., and Lasenby, A. Dynamic nested sampling: an improved algorithm for parameter estimation and evidence calculation. *Statistics and Computing*, 29(5):891–913, 2019.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. Stochastic variational inference. 14(1):1303–1347, 2013.

Jarvenpaa, M., Vehtari, A., and Marttinen, P. Batch simulations and uncertainty quantification in gaussian process surrogate approximate bayesian computation. In *Conference on Uncertainty in Artificial Intelligence*, pp. 779–788. PMLR, 2020.

Jones, D. R., Perttunen, C. D., and Stuckman, B. E. Lipschitzian optimization without the lipschitz constant. *Journal of optimization Theory and Applications*, 79(1):157–181, 1993.

Kalaghatgi, C., Hannam, M., and Raymond, V. Parameter estimation with a spinning multimode waveform model. *Phys. Rev. D*, 101:103004, May 2020. doi: 10.1103/PhysRevD.101.103004. URL https://link.aps.org/doi/10.1103/PhysRevD.101.103004.

Kronmal, R. A. and Peterson Jr, A. V. On the alias method for generating random variables from a discrete distribution. *The American Statistician*, 33(4):214–218, 1979.

Li, D., Yang, K., and Wong, W. H. Density estimation via discrepancy based adaptive sequential partition. *Advances in neural information processing systems*, 29:1091, 2016.

Lu, L., Jiang, H., and Wong, W. H. Multivariate density estimation by bayesian sequential partitioning. *Journal of the American Statistical Association*, 108(504):1402–1410, 2013.

McCool, M. D. and Harwood, P. K. Probability trees. In *Graphics Interface*, volume 97, pp. 37–46. Citeseer, 1997.

Miller III, A. C. and Rice, T. R. Discrete approximations of probability distributions. *Management science*, 29(3):352–362, 1983.

Neal, R. M. Slice sampling. *Annals of statistics*, pp. 705–741, 2003.

Nguyen-Hong, N., Nguyen-Duc, H., and Nakanishi, Y. Optimal sizing of energy storage devices in isolated wind-diesel systems considering load growth uncertainty. *IEEE Transactions on Industry Applications*, 54(3):1983–1991, 2018.

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

Smolyak, S. A. Quadrature and interpolation formulas for tensor products of certain classes of functions. 148:1042–1045, 1963.

Tanaka, K. and Toda, A. A. Discrete approximations of continuous distributions by maximum entropy. *Economics Letters*, 118(3):445–450, 2013.

Tanaka, K. and Toda, A. A. Discretizing distributions with exact moments: Error estimate and convergence analysis. *SIAM Journal on Numerical Analysis*, 53(5):2158–2177, 2015.

Williams, J. W. J. Algorithm 232: heapsort. *Commun. ACM*, 7:347–348, 1964.

Wilson, A. and Adams, R. Gaussian process kernels for pattern discovery and extrapolation. In *International conference on machine learning*, pp. 1067–1075, 2013.