Posterior Expectation of Regularly Paved Random Histograms

RAAZESH SAINUDIIN, GLORIA TENG, JENNIFER HARLOW, and DOMINIC LEE,
University of Canterbury

We present a novel method for averaging a sequence of histogram states visited by a Metropolis-Hastings Markov chain whose stationary distribution is the posterior distribution over a dense space of tree-based histograms. The computational efficiency of our posterior mean histogram estimate relies on a statistical data-structure that is sufficient for nonparametric density estimation of massive, multidimensional metric data. This data-structure is formalized as statistical regular paving (SRP). A regular paving (RP) is a binary tree obtained by selectively bisecting boxes along their first widest side. SRP augments RP by mutably caching the recursively computable sufficient statistics of the data. The base Markov chain used to propose moves for the Metropolis-Hastings chain is a random walk that data-adaptively prunes and grows the SRP histogram tree. We use a prior distribution based on Catalan numbers and detect convergence heuristically. The performance of our posterior mean SRP histogram is empirically assessed for large sample sizes simulated from several multivariate distributions that belong to the space of SRP histograms.

Categories and Subject Descriptors: E.1 [**Data Structures**]: *Trees*; G.2.2 [**Discrete Mathematics**]: Graph Theory—*Trees*; G.3 [**Probability and Statistics**]: *Probabilistic algorithms (including Monte Carlo); statistical computing*

General Terms: Algorithms, Design, Performance, Theory

Additional Key Words and Phrases: Multivariate histogram, Markov chain Monte Carlo, Bayesian estimation, Catalan Prior, averaging tree-based histograms, statistical regular pavings

ACM Reference Format:
Sainudiin, R., Teng, G., Harlow, J., and Lee, D. 2013. Posterior expectation of regularly paved random histograms. *ACM Trans. Model. Comput. Simul.* 23, 1, Article 6 (January 2013), 20 pages.
DOI: http://dx.doi.org/10.1145/2414416.2414422

1. INTRODUCTION

Histograms are commonly used for non-parametric density estimation but the estimates depend on the selection of the histogram bins, which is essentially a partition of the data space. Stone [1982], as cited by Lugosi and Nobel [1996], showed that data-dependent partitions can provide estimates which are theoretically superior to those using partitions based simply on the number of data points in the data set. The problem, then, is how best to create data-dependent partitions.

A common approach is to use a penalized maximum likelihood estimator, often based on the Akaike information criterion (AIC) [Akaike 1974]. Taylor [1987] minimized

This work was partly supported by R. Sainudiin's Visiting Scientists award to the Indian Statistical Institute, Bangalore. J. Harlow was partly supported by R. Sainudiin's external consulting revenues from the NZ Ministry of Tourism.
Authors' address: R. Sainudiin, G. Teng, J. Harlow, and D. Lee, Department of Mathematics and Statistics, University of Canterbury, Private Bag 4800, Christchurch 8041, New Zealand; email: r.sainudiin@math.canterbury.ac.nz.

DOI: http://dx.doi.org/10.1145/2414416.2414422

AIC to derive the asymptotically optimal bin width for 1-dimensional data. Birgé and Rozenholc [2006] used a nonasymptotic evaluation of the performances of penalized maximum likelihood estimator in some exponential families due to Castellan [1999] and heavy simulations to optimize the form of the penalty function. These methods were data-based partitioning schemes for regular histograms. Hartigan [1996] then compared equal-bin-width "Akaike-histograms" to Bayesian histograms constructed with a subjective smoothing parameter to control the number of elements in his partitions. Castellan [1999] extended this to multivariate data and irregular histograms. Klemelä [2009b] described adaptive density estimation with best basis selection for multi-dimensional data by determining the dimension on which to subdivide some element of the partition. This algorithm grows the tree by bisecting each element successively on each possible dimension until a specified maximum number of bisections in each dimension has been reached, followed by pruning to minimize a complexity penalized L_2 error. Klemelä [2007] also discussed a CART-like [Breiman et al. 1984] methodology for density estimation, involving partitioning using a greedy algorithm to minimize an empirically approximated L_2-based error followed by pruning to minimize the complexity-penalized error.

Some of the problems with the complexity penalized approach are discussed in Birgé and Rozenholc [2006]. This methodology relied on the asymptotically optimal performance of penalized maximum likelihood estimators but was constrained by the best form of the penalty function itself being dependent on the unknown underlying density. Furthermore, Jackson et al. [2005] commented on work by Scargle [1998], which found that greedy algorithms are not guaranteed to find optimal partitions and proposed the use of a Bayesian model. Knuth [2006] describes a Bayesian framework and an algorithm for finding the equal-bin-width solution which maximizes the marginal posterior probability of the number of bins by a brute-force search.

All these methods are not able to computationally cope in high dimensions. Among the space partitioning tree-based methods for computationally efficient kernel density estimation and band-width selection using cross-validation, the multiresolution *kd*-trees and ball trees of Gray and Moore [2003] can cope better in high dimensions, especially if the underlying density is highly structured and nonuniform. Various aspects of statistical regular pavings, our space partitioning tree-based data structure, including cached sufficient statistics at internal nodes, were inspired by Gray and Moore [2003].

A density estimate based on a single partition of the data space, whether this partition is found by a greedy algorithm, by dynamic programming, or by an asymptotically optimal formula, is a point estimate. An alternative approach is to estimate the expectation of the posterior probability distribution over a set of partitions from the average of a number of independent random samples of partition states from the distribution. Klemelä [2009a] has also been involved in work on dealing with the variability of unstable histogram density estimators by averaging a number of estimates produced from bootstrapped subsamples from the original data set in the `delt` package for `R`. However, each sample histogram state is obtained from a particular algorithm (either greedy or CART-pruning) and is not a random sample from the whole sample space.

The main focus in this work is in obtaining computationally efficient Bayesian density estimates from averaging histograms as opposed to finding the best partition using principles such as cross-validation or penalized maximum likelihood. Here we propose an efficient algorithm that allows for easy averaging of multidimensional histograms with regularly paved partitions. This space of regular paving histograms is closed under addition and scalar multiplication. We construct a Metropolis-Hastings Markov chain under a Catalan prior distribution to produce the sample mean estimate of the

Bayesian posterior expectation over this space of histograms. We apply our method to complicated mixtures of uniform densities over a range of partitions from the space of regular pavings and study their L_1 error or integrated absolute error (IAE) for sample sizes as large as 10^7 points in various multivariate settings.

Posterior inference of density estimates based on a Pólya tree [Lavine 1992] and its extensions, such as a mixture of Pólya tree [Lavine 1992] and randomized Pólya tree [Paddock et al. 2003] work from a nested sequence of partitions. The nested partitioning strategy of Pólya trees is different from the one we propose here and our partitioning strategy can be combined with the complementary strengths of Pólya trees as discussed in 5.

Methods for density estimation typically break down in high dimensions because computational complexity grows exponentially with the dimension. The standard arguments given for this breakdown called the *curse of dimensionality*, as remarked by Moore [2000], assume that the data has no underlying structure (e.g., data comes from the uniform density on $[0, 1]^d$). The multiresolution *kd*-trees and ball-trees [Gray and Moore 2003] exploit the nonuniform structure in real-world data to accelerate high dimensional kernel density estimation. Our methods share several aspects of the data structures in Moore [2000] and Gray and Moore [2003] but are complementary in the sense that they work better with highly uniform or mixtures of uniform data. Also, our focus is on arithmetic capabilities of the data structures in a Bayesian context of averaging histograms that are sampled form the posterior over a family of partitions that are not as strictly nested as Pólya trees and their variants, as opposed to the focus of Gray and Moore [2003] on computational efficiency in cross-validation. Finally, we are interested in obtaining the IAE of our estimator for datasets that have been simulated from high-dimensional densities, unlike other high-dimensional density estimation studies that either focus on real-world data and thereby avoid IAE altogether or simply cannot report the IAE exactly in dimensions higher than 1 or 2. Our SRP trees and their associated arithmetic aided by interval analytic approximation methods allow us to obtain IAEs of our estimator in high dimensions.

The remaining sections of this article are organized as follows. We first describe regular pavings and their associated tree spaces in 2.1 and then show their statistical extension in 2.2. The Bayesian context for density estimation is set in 2.3 and 2.4 gives the method for averaging histograms. The Metropolis-Hastings Markov chain and associated convergence heuristics is described in Section 3. The performance of our density estimator is assessed using simulations in Section 4 and we conclude with a discussion in Section 5.

2. STATISTICAL REGULAR PAVING (SRP)

2.1. Regular Paving (RP)

Let $\boldsymbol{x} := [\underline{x}, \overline{x}]$ be a compact real interval with lower bound $\underline{x}$ and upper bound $\overline{x}$ where $\underline{x} \leq \overline{x}$. Let the space of such intervals be $\mathbb{IR}$. We can then define a box of dimension d as an interval vector

$$\boldsymbol{x} := \left[\underline{x}_1, \overline{x}_1\right] \times \ldots \times \left[\underline{x}_d, \overline{x}_d\right] \quad .$$

Let $\mathbb{IR}^d$ be the set of all such boxes. Consider a box $\boldsymbol{x}$ in $\mathbb{IR}^d$. Let the index ι be the first coordinate of maximum width, that is,

$$\iota = \min\left(\underset{i}{\operatorname{argmax}}(\overline{x}_i - \underline{x}_i)\right) \quad .$$

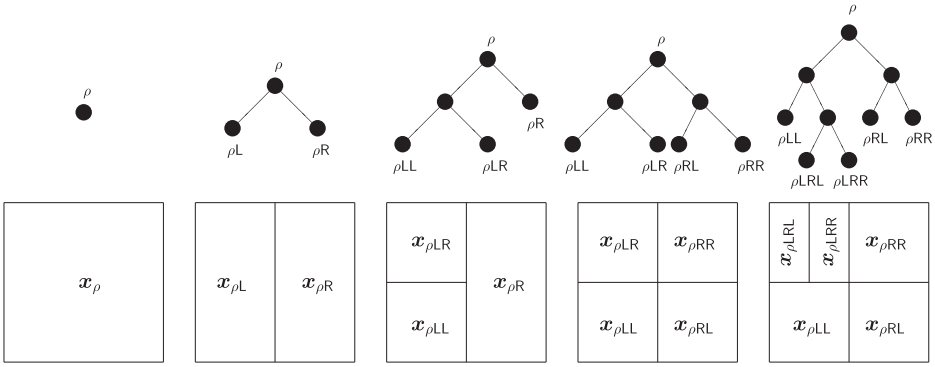


Fig. 1. A sequence of selective bisections of boxes (nodes) along the first widest coordinate, starting from the root box (root node), produces an RP.

A *bisection* or *split* of $\boldsymbol{x}$ at the midpoint along this first widest component gives us the left- and right-child boxes of $\boldsymbol{x}$ as follows:

$$\boldsymbol{x}_{\mathsf{L}} := \left[\underline{x}_1, \overline{x}_1\right] \times \cdots \times \left[\underline{x}_\iota, (\underline{x}_\iota + \overline{x}_\iota)/2\right) \times \left[\underline{x}_{\iota+1}, \overline{x}_{\iota+1}\right] \times \cdots \times \left[\underline{x}_d, \overline{x}_d\right],$$

$$\boldsymbol{x}_{\mathsf{R}} := \left[\underline{x}_1, \overline{x}_1\right] \times \cdots \times \left[(\underline{x}_\iota + \overline{x}_\iota)/2, \overline{x}_\iota\right] \times \left[\underline{x}_{\iota+1}, \overline{x}_\iota + 1\right] \times \cdots \times \left[\underline{x}_d, \overline{x}_d\right].$$

Such a bisection is said to be *regular*. A recursive sequence of selective regular bisections of boxes with possibly open boundaries along the first widest coordinate, starting from the root box $\boldsymbol{x}$ in $\mathbb{IR}^d$ is known as a *regular paving* (RP) [Jaulin et al. 2001] or n-tree [Samet 1990] of $\boldsymbol{x}$. An RP of $\boldsymbol{x}$ can also be seen as a binary tree formed by recursively bisecting the box $\boldsymbol{x}$ at the root node. These trees are known as *plane binary trees* in enumerative combinatorics [Stanley 1999, Ex. 6.19(d), p. 220] and as *finite, rooted binary trees (frb-trees)* in geometric group theory [Meier 2008, Ch. 10]. When the root box $\boldsymbol{x}$ is clear from the context we refer to an RP of $\boldsymbol{x}$ as merely an RP. Each node of an RP is associated with a sub-box of the root box that can be attained by a sequence of selective regular bisections.

Each node in an RP is distinctly labeled by the sequence of child node selections from the root node. We label these nodes and the associated boxes with strings composed of L and R for left and right, respectively. For example, in Figure 1, the root node associated with root box $\boldsymbol{x}_\rho$ is labeled ρ. First, we split ρ into two child nodes and denote it by $\bigtriangledown(\rho) = \{\rho\mathsf{L}, \rho\mathsf{R}\}$. These left child and right child nodes are labeled by $\rho\mathsf{L}$ and $\rho\mathsf{R}$, respectively. The left half of $\boldsymbol{x}_\rho$ that is now associated with node $\rho\mathsf{L}$ is denoted by $\boldsymbol{x}_{\rho\mathsf{L}}$. Similarly, the right half of $\boldsymbol{x}_\rho$ that is associated with the right child node $\rho\mathsf{R}$ is denoted by $\boldsymbol{x}_{\rho\mathsf{R}}$. We say $\rho\mathsf{L}$ and $\rho\mathsf{R}$ are a pair of *sibling nodes* since they share the same parent node ρ. This pair of sibling nodes can be *reunited* or *merged* to its parent node ρ and such a merging operation is denoted by $\bigtriangleup(\rho\mathsf{L}, \rho\mathsf{R}) = \rho$. A node with no child nodes is called a *leaf node*. A *cherry node* is a subterminal node with a pair of leaf nodes that are siblings. These sibling leaf nodes can be reunited or merged back to the cherry node, thereby turning the cherry node into a leaf node in the process. Note that we can only split a leaf node or merge a cherry node. Returning to Figure 1, let us further split the left node $\rho\mathsf{L}$ by bisecting the associated box $\boldsymbol{x}_{\rho\mathsf{L}}$ to get its left and right child nodes $\rho\mathsf{LL}$ and $\rho\mathsf{LR}$ with the associated sub-boxes $\boldsymbol{x}_{\rho\mathsf{LL}}$ and $\boldsymbol{x}_{\rho\mathsf{LR}}$, respectively. Next, we split the right child node $\rho\mathsf{R}$ similarly into its child nodes $\rho\mathsf{RL}$ and $\rho\mathsf{RR}$, respectively. Let us select $\rho\mathsf{LR}$ to do a final split and obtain its child nodes $\rho\mathsf{LRL}$ and $\rho\mathsf{LRR}$. We have obtained a binary tree from four splits of the root node. A graphical representation of the

obtained RP is shown in Figure 1. We denote the label set of all nodes by $\mathbb{V} := \rho \cup \left\{\rho\{\mathsf{L},\mathsf{R}\}^j : j \in \mathbb{N}\right\}$.

Let the jth interval of a box $\boldsymbol{x}_{\rho v}$ be $[\underline{x}_{\rho v,j}, \overline{x}_{\rho v,j}]$. Then the volume of a d-dimensional box $\boldsymbol{x}_{\rho v}$ associated with the node ρv of an RP of $\boldsymbol{x}_\rho$ is the product of the side-lengths of the box, that is,

$$\text{vol}\ (\boldsymbol{x}_{\rho v}) = \prod_{j=1}^{d} \left(\overline{x}_{\rho v,j} - \underline{x}_{\rho v,j}\right) \quad .$$

The volume may also be associated with the *depth* of a node. A node has depth δ if it can be reached by δ splits from the root node. Then, the volume of any d-dimensional box $\boldsymbol{x}_{\rho v}$ associated with node ρv having depth δ is $\text{vol}\,(\boldsymbol{x}_{\rho v}) = 2^{-\delta}\text{vol}\,(\boldsymbol{x}_\rho)$ due to the recursive nature of the bisections and the restriction to only bisect perpendicularly at the mid-point along the first widest coordinate. We use the nodes of the RP in Figure 1 for illustration purposes. Assume that the root box $\boldsymbol{x}_\rho$ is a unit hypercube. Then the root node ρ has depth 0 and $\text{vol}\,(\boldsymbol{x}_\rho) = 1$, the nodes $\rho\mathsf{L}$ and $\rho\mathsf{R}$ have depth 1 and volume 2^{-1}, the nodes $\rho\mathsf{LL}$, $\rho\mathsf{LR}$, $\rho\mathsf{RL}$, $\rho\mathsf{RR}$ have depth 2 and volume 2^{-2}, and finally the nodes $\rho\mathsf{LRL}$, $\rho\mathsf{LRR}$ have depth 3 and volume 2^{-3}.

We can now label each leaf node of a tree by its depth. The leaf nodes of the RP in Figure 1, listed in left-right ordering, is $[\rho\mathsf{LL}, \rho\mathsf{LRL}, \rho\mathsf{LRR}, \rho\mathsf{RL}, \rho\mathsf{RR}]$. Then, this RP has 23322 as its *ordered leaf-depth string*. Each RP can be uniquely identified by its ordered leaf-depth string. Thus, this RP can be denoted by s_{23322} and the set of leaf boxes associated with its leaf nodes by $\ell(s_{23322}) = \{\boldsymbol{x}_{\rho\mathsf{LL}}, \boldsymbol{x}_{\rho\mathsf{LRL}}, \boldsymbol{x}_{\rho\mathsf{LRR}}, \boldsymbol{x}_{\rho\mathsf{RL}}, \boldsymbol{x}_{\rho\mathsf{RR}}\}$. Among these leaf nodes, we can reunite $\rho\mathsf{LRL}$ and $\rho\mathsf{LRR}$ to get $\rho\mathsf{LR}$, and further reunite $\rho\mathsf{RL}$ and $\rho\mathsf{RR}$ to get $\rho\mathsf{R}$. Note that the nodes $\rho\mathsf{LR}$ and $\rho\mathsf{R}$ of s_{23322} are cherry nodes and the set of boxes associated with its cherry nodes is $c(s_{23322}) = \{\boldsymbol{x}_{\rho\mathsf{LR}}, \boldsymbol{x}_{\rho\mathsf{R}}\}$. Each sequence of splits and merges of an RP with root node ρ returns a partition of its root box $\boldsymbol{x}_\rho$ given by the set of its leaf boxes.

Having seen a particular RP s_{23322} let us study the space of all RPs.

Let $\mathbb{S}_k$ be the set of all RPs of $\boldsymbol{x}_\rho$ made of k splits. Note that $|\ell(s)| = k + 1$ if $s \in \mathbb{S}_k$. The number of distinct binary trees with k splits is equal to the Catalan number

$$C_k = \frac{1}{k+1}\binom{2k}{k} = \frac{(2k)!}{(k+1)!(k!)} \quad . \tag{1}$$

For $i,j \in \mathbb{Z}_+$, where $\mathbb{Z}_+ := \{0, 1, 2, \ldots\}$ and $i \leq j$, let $\mathbb{S}_{i:j}$ be the set of RPs with k splits where $k \in \{i, i+1, \ldots, j\}$. The space of all RPs is then $\mathbb{S}_{0:\infty} := \lim_{j\to\infty} \mathbb{S}_{0:j}$. Figure 2 displays the transition diagram over $\mathbb{S}_{0:3}$ where the gray arrows represent the transition from one RP state to another through a split or reunion. There may be more than one way, that is, distinct sequence of splits, to reach an RP in $\mathbb{S}_k$ from the root node by applying exactly k splits. We are interested in randomized algorithms, which can be seen as Markov chains on $\mathbb{S}_{0:\infty}$.

2.2. Statistical Regular Pavings (SRPs)

Suppose n points $X_1, X_2, \ldots, X_n$ have fallen into the bounding root box $\boldsymbol{x}_\rho$ of an RP s. We extend the notion of RP to SRP in order to represent a data-driven partition of $\boldsymbol{x}_\rho$. Recall that each node ρv of an RP has a box $\boldsymbol{x}_{\rho v}$ associated with it. For the purpose of statistical set processing we can further associate each node ρv of an RP with *recursively computable statistics*, such as, (i) $\#\boldsymbol{x}_{\rho v} := \sum_{i=1}^{n} \mathbb{1}_{\boldsymbol{x}_{\rho v}}(X_i)$, the sample count,

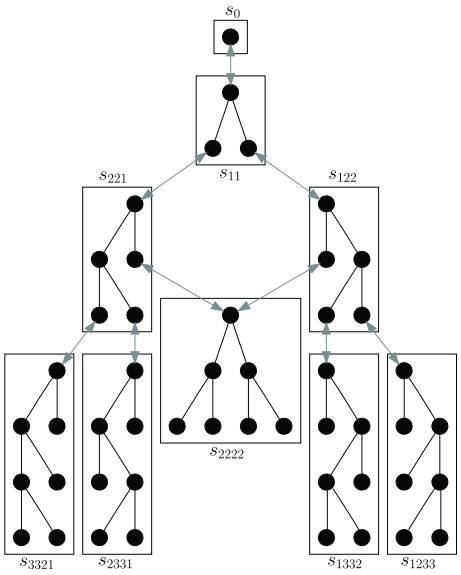


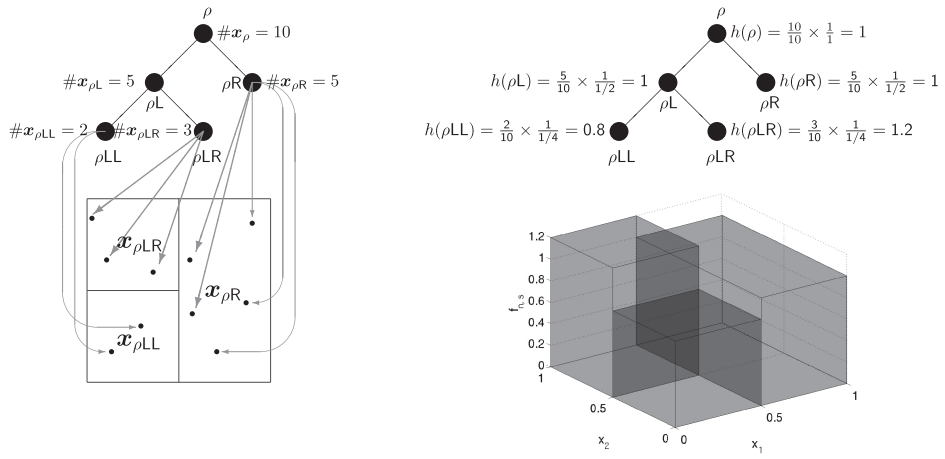
Fig. 2. Transition diagram over $\mathbb{S}_{0:3}$ with split/reunion transitions from one RP state to another.

(ii) the sample mean, (iii) the sample variance-covariance matrix, etc., of the data points that fall into $\boldsymbol{x}_{\rho v}$. This aspect of our method is inspired by the *cached-statistic metric trees* of [Moore 2000]. Each leaf node has associations (via pointers in our C++ implementation) to the data that lie within its leaf box. When a bisection happens, the data falls into the box associated with either the left or right child node of the bisected node, depending on its location. The recursively computable statistics in each child node gets updated in this process. When two sibling nodes are reunited, the recursively computable statistics of the reunited node remain unchanged. We call this information structure a *statistical regular paving* (SRP) since it enhances an RP by recording recursively computable statistics of the data for subsequent statistical set processing. Figure 3(a) depicts the details of an SRP tree, nodes, leaf boxes forming the partition of the root box along with their associations with the data (gray arrows). This gives an *SRP histogram* as depicted in Figure 3(b). By an abuse of notation, we denote an RP as well as an SRP by s and the space of all RPs as well as SRPs by $\mathbb{S}_{0:\infty}$.

2.3. Posterior Mean of SRP Histograms

Suppose $X_1, \ldots, X_n$ are independent and identical random vectors in $\mathbb{R}^d$, each distributed according to μ and having a non-atomic density $f \in L_1(\lambda)$, that is, $P(X_1 \in A) = \mu(A) = \int_A f d\lambda$, where μ is absolutely continuous with respect to d-dimensional Lebesgue measure λ or $\mu \ll \lambda$. Using our SRP information structures, we are interested in producing the sample mean histogram estimate of the Bayes posterior expectation over this space of SRP histograms.

After the arrival of a particular data sample $x_1, \ldots, x_n$ into the root box $\boldsymbol{x}_\rho$ of the current SRP s, where each point x_i is a row vector such that $x_i = (x_{i,1}, \ldots, x_{i,d})$, we get a partition $\ell(s)$. For each leaf box $\boldsymbol{x}_{\rho v} \in \ell(s)$, let $\mu_n(\boldsymbol{x}_{\rho v}) := \#\boldsymbol{x}_{\rho v}/n$ be its empirical measure based on the sample $x_1, \ldots, x_n$ and $\lambda(\boldsymbol{x}_{\rho v}) = \mathrm{vol}\,(\boldsymbol{x}_{\rho v})$ be its Lebesgue measure on $\mathbb{R}^d$. Let $\boldsymbol{x}(x)$ be the leaf box containing x. Then we only need to know the counts and



(a) An SRP tree and its constituents. (b) An SRP histogram and its tree.

Fig. 3. An SRP and its corresponding histogram.

the boxes for the current set of leaf nodes $\ell(s)$ to produce the histogram estimate as follows:

$$f_{n,s}(x) = \begin{cases} \frac{\mu_n(\boldsymbol{x}(x))}{\mathrm{vol}\,(\boldsymbol{x}(x))} = \frac{\#\boldsymbol{x}(x)}{n} \times \frac{1}{\mathrm{vol}\,(\boldsymbol{x}(x))} & \text{if } 0 < \mathrm{vol}\,(\boldsymbol{x}(x)) < \infty, \boldsymbol{x}(x) \in \ell(s), \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Let the height-value of a given node ρv of an SRP histogram be given by $h(\rho v) = \#\boldsymbol{x}_{\rho v}/(n \cdot \mathrm{vol}\,(\boldsymbol{x}_{\rho v}))$. Figure 3(b) shows the corresponding histogram estimate of the ten data points shown in Figure 3(a). Once the SRP has been constructed the associations to data from the leaf nodes can be removed so that only the height statistics at all nodes remain. This is a significant memory-saving intermediary step when comparing batches of massive simulated data during simulation-intensive inference problems or when producing a sequence of density estimates from pulses of massive real-world data in assimilation and online processing problems [Teng et al. 2012].

Let π be the posterior distribution that is proportional to the product of the likelihood of the data given s and the prior probability of s, that is,

$$\begin{aligned} \pi(s) &:= \Pr\{s|x_1, \ldots, x_n\} \\ &\propto \Pr\{x_1, \ldots, x_n|s\} \Pr\{s\} \\ &= \Pr\{x_1|s\}\Pr\{x_2|s\} \cdots \Pr\{x_{n-1}|s\} \Pr\{x_n|s\} \Pr\{s\} \\ &\approx \prod_{i=1}^{n} f_{n,s}\left(x_i\right) \Pr\{s\} \\ &= \prod_{\boldsymbol{x}_{\rho v} \in \ell(s)} \left(\frac{\#\boldsymbol{x}_{\rho v}}{n \cdot \mathrm{vol}\,\left(\boldsymbol{x}_{\rho v}\right)} \right)^{\#\boldsymbol{x}_{\rho v}} \Pr\{s\} \ . \end{aligned}$$

Note how we approximate the likelihood of the data given s by the maximum likelihood value from the histogram on s.

We want our prior distribution $\{\Pr(s)\}$ over $s \in \mathbb{S}_{0:\infty}$ to be proper and uninformative in some natural sense. Moreover, we also want our prior probabilities to decrease as the partition size increases in order to penalize large partitions. With these considerations, we propose a Catalan family of proper priors associated with any

convergent decreasing sequence. Suppose $\{a_k\}$ for $k = 1, 2, \ldots$ is any decreasing sequence of positive real numbers such that $\sum_{k=1}^{\infty} a_k = a < \infty$. Recall from Eq. (1) that the Catalan number C_k gives the number of SRPs in $\mathbb{S}_k$ with k splits and $k+1$ leaves. An $\{a_k\}$-*penalized uninformative proper Catalan prior* that assigns states in $\mathbb{S}_k$ with probability a_k/a and distributes this mass uniformly over $\mathbb{S}_k$ is given by:

$$\Pr\{s\} = \sum_{k=0}^{\infty} \mathbb{1}_{\mathbb{S}_k}(s) \frac{a_k}{a C_k} \quad . \tag{3}$$

In this work, we fix a particular prior obtained from the sequence of $a_k = 1/C_k$ with $a = 2 + 4\pi/3^{5/2} \approx 2.806133050770763$ [McGarvey and Cloitre 2005]. Such a *natural Catalan prior* is given by:

$$\Pr\{s\} = \sum_{k=0}^{\infty} \mathbb{1}_{\mathbb{S}_k}(s) \frac{1}{\left(2 + 4\pi/3^{5/2}\right) C_k^2} \quad . \tag{4}$$

Thus, the posterior distribution on $\mathbb{S}_{0:\infty}$ is given by

$$\pi(s) \propto \prod_{\boldsymbol{x}_{\rho v} \in \ell(s)} \left(\frac{\#\boldsymbol{x}_{\rho v}}{n \cdot \mathrm{vol}\left(\boldsymbol{x}_{\rho v}\right)} \right)^{\#\boldsymbol{x}_{\rho v}} \cdot \sum_{k=0}^{\infty} \mathbb{1}_{\mathbb{S}_k}(s) \frac{1}{\left(2 + 4\pi/3^{5/2}\right) C_k^2} \quad . \tag{5}$$

We can obtain an estimate of the posterior mean from the sample average of the thinned-out post-burn-in sequence of states visited by a discrete time Markov chain $\{S(t)\}$, for $t \in \mathbb{Z}_+ := \{0, 1, 2, \ldots\}$, over the state space $\mathbb{S}_{0:\infty}$ with the posterior distribution π as its stationary distribution. In order to do this, we need to be able to efficiently obtain the average of a number of SRP histograms.

2.4. Averaging SRP Histograms

We can average m histograms if we are able to add any two histograms together and get another histogram on the condition that each histogram has the same root box, and multiply any histogram by a scalar. Since the RP trees, just like frb-trees, are closed under pairwise union or overlay operations, we can extend these operations to SRPs from which the histograms are built. This enables us to perform arithmetic over SRPs in a recursive and efficient manner to obtain averaged histograms.

Consider two SRPs $s^{(1)}$ and $s^{(2)}$ with root nodes $\rho^{(1)}$ and $\rho^{(2)}$, respectively, with the same root box $\boldsymbol{x}_\rho = \boldsymbol{x}_{\rho^{(1)}} = \boldsymbol{x}_{\rho^{(2)}}$. Let the corresponding histograms of SRPs $s^{(1)}$ and $s^{(2)}$ be $f_{n,s^{(1)}}$ and $f_{n,s^{(2)}}$. We can add the two histograms by applying `AddSRPHist`$(\rho^{(1)}, \rho^{(2)})$ described by Algorithm 1.

Figure 4 illustrates how addition is performed on two SRPs to obtain the sum SRP. The sum is then divided by 2 to obtain the average histogram $f_{n,s^{(1)+(2)}} = (f_{n,s^{(1)}} + f_{n,s^{(2)}})/2$. The average of m SRPs is obtained similarly as $\left(\left(\cdots\left(\left(f_{n,s^{(1)}} + f_{n,s^{(2)}}\right) + f_{n,s^{(3)}}\right) + \cdots\right) + f_{n,s^{(m)}}\right)/m$.

3. A METROPOLIS-HASTINGS MARKOV CHAIN OVER SRP HISTOGRAMS

In this section, we will first discuss how the space of all SRPs $\mathbb{S}_{0:\infty}$ can be bounded to make a finite state space $\tilde{\mathbb{S}}_n$. We then show that for a given set of n data points $x_1, x_2, \ldots, x_n$ in the root box $\boldsymbol{x}_\rho$, the base chain $\{Y(t)\}_{t \in \mathbb{Z}_+}$ is irreducible and aperiodic on the machine-representable finite state space $\tilde{\mathbb{S}}_n \subset \mathbb{S}_{0:\infty}$ and then derive a Metropolis-Hastings chain $\{S(t)\}_{t \in \mathbb{Z}_+}$ with the desired stationary distribution. We conclude the section with some heuristics to diagnose and accelerate mixing.

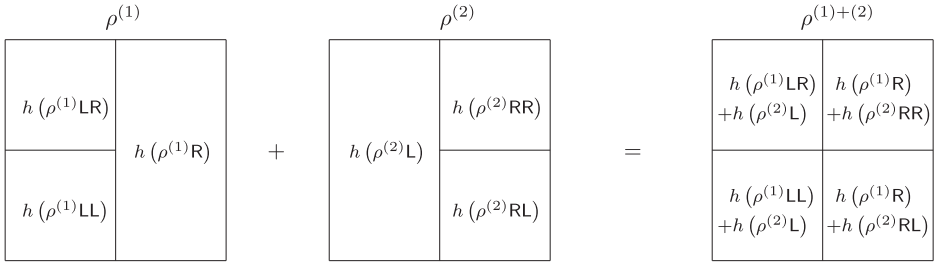


Fig. 4. Adding two SRP histograms.

ALGORITHM 1: `AddSRPHist`

input : two SRP histogram root nodes $\rho^{(1)}$ and $\rho^{(2)}$ with same root box $\boldsymbol{x}_\rho = \boldsymbol{x}_{\rho^{(1)}} = \boldsymbol{x}_{\rho^{(2)}}$.

output : the sum SRP histogram with root node $\rho^{(1)+(2)}$.

Make a new node $\rho^{(1)+(2)}$ with box $\boldsymbol{x}_\rho$
$h(\rho^{(1)+(2)}) \leftarrow h(\rho^{(1)}) + h(\rho^{(2)})$

if ($\rho^{(1)}$ *is a leaf node*) & ($\rho^{(2)}$ *is not a leaf node*) **then**
  Make nodes L', R'
  $\boldsymbol{x}_{\mathsf{L}'} \leftarrow \boldsymbol{x}_{\rho^{(1)}\mathsf{L}}, \boldsymbol{x}_{\mathsf{R}'} \leftarrow \boldsymbol{x}_{\rho^{(1)}\mathsf{R}}$
  $h(\mathsf{L}') \leftarrow h(\rho^{(1)}), h(\mathsf{R}') \leftarrow h(\rho^{(1)})$
  Graft onto $\rho^{(1)+(2)}$ as left child the node `AddSRPHist`$(\rho^{(2)}\mathsf{L}, \mathsf{L}')$
  Graft onto $\rho^{(1)+(2)}$ as right child the node `AddSRPHist`$(\rho^{(2)}\mathsf{R}, \mathsf{R}')$
end

if ($\rho^{(2)}$ *is a leaf node*) & ($\rho^{(1)}$ *is not a leaf node*) **then**
  Make nodes L', R'
  $\boldsymbol{x}_{\mathsf{L}'} \leftarrow \boldsymbol{x}_{\rho^{(2)}\mathsf{L}}, \boldsymbol{x}_{\mathsf{R}'} \leftarrow \boldsymbol{x}_{\rho^{(2)}\mathsf{R}}$
  $h(\mathsf{L}') \leftarrow h(\rho^{(2)}), h(\mathsf{R}') \leftarrow h(\rho^{(2)})$
  Graft onto $\rho^{(1)+(2)}$ as left child the node `AddSRPHist`$(\rho^{(1)}\mathsf{L}, \mathsf{L}')$
  Graft onto $\rho^{(1)+(2)}$ as right child the node `AddSRPHist`$(\rho^{(1)}\mathsf{R}, \mathsf{R}')$
end

if (*both* $\rho^{(1)}$ *and* $\rho^{(2)}$ *are not leaf nodes*) **then**
  Graft onto $\rho^{(1)+(2)}$ as left child the node `AddSRPHist`$(\rho^{(1)}\mathsf{L}, \rho^{(2)}\mathsf{L})$
  Graft onto $\rho^{(1)+(2)}$ as right child the node `AddSRPHist`$(\rho^{(1)}\mathsf{R}, \rho^{(2)}\mathsf{R})$
end
return $\rho^{(1)+(2)}$

State Space. Let us note that on a computer with double-precision floating-point numbers [ANSI/IEEE 754-1985], $\tilde{\mathbb{S}}_n$ is necessarily a finite subset of $\mathbb{S}_{0:\infty}$. For instance, we cannot represent the child boxes with our number screen if the widest side of the parent box to be bisected is already given by two adjacent floating-point numbers. First, let us appreciate how the data-dependent states in $\tilde{\mathbb{S}}_n$ are dictated by the boundary of "unsplittable states" in $\mathbb{S}_{0:\infty}$ in addition to the hard boundaries imposed by the machine's floating-point number screen. There are three natural ways to define the boundary of unsplittable SRP states.

The first and simplest way to define such a boundary in $\mathbb{S}_{0:\infty}$ is by saying that we cannot split an SRP state beyond a maximal number of splits $\overline{m}_n$. We can allow this

$\overline{m}_n = n^\beta$ for some $\beta < 1$ in order to ensure a sub-linear growth of the number of leaves with the number of data points, that is, $\overline{m}_n/n \to 0$ as $n \to \infty$, for instance. Using this hard boundary based on the maximum number of allowed splits our finite state space of the base chain is $\tilde{\mathbb{S}}_n = \mathbb{S}_{0:\overline{m}_n}$. In this case, we know that $|\mathbb{S}_{0:\overline{m}_n}| = \sum_{k=0}^{\overline{m}_n} C_k$. We typically take β close to 1 under this approach to ensure a sufficiently large $\tilde{\mathbb{S}}_n$ provided the density f is sufficiently bounded away from 0 uniformly over $\boldsymbol{x}_\rho$.

The second and less simple way to define such a boundary in $\mathbb{S}_{0:\infty}$ is by saying that we cannot split any leaf of an SRP state beyond a minimum volume, say $\underline{\lambda}_n$. Recall that the volume of a box at a leaf node ρv at depth δ in an SRP s is $\mathrm{vol}(\boldsymbol{x}_{\rho v}) = \mathrm{vol}(\boldsymbol{x}_\rho)/2^\delta$. Thus, the minimum volume boundary constraint is equivalent to a maximal leaf-depth constraint on states in $\tilde{\mathbb{S}}_n$, where we ensure that all states visited by the chain have an $\underline{\lambda}_n$-dependent maximal depth $\overline{\delta}_n = \left\lfloor \log_2 \left(\mathrm{vol}(\boldsymbol{x}_\rho)/\underline{\lambda}_n \right) \right\rfloor$. This gives an upper bound by considering the tip of the parabolic tongue of the unsplittable boundary of leaf-depth constrained SRP states in $\mathbb{S}_{0:\infty}$ over an arrangement similar to the transition diagram of Figure 2. In this case, $\tilde{\mathbb{S}}_n$ is contained in $\mathbb{S}_{0:\tau}$, where $\tau = 2^{\overline{\delta}_n} - 1$. Once again we can allow the maximal depth $\overline{\delta}_n$ to increase appropriately with the number of data points n to ensure that $\underline{\lambda}_n \to 0$ as $n \to \infty$.

The third, more complicated, and more data-driven way to define such a boundary of unsplittable states in $\mathbb{S}_{0:\infty}$ in order to produce a finite data-dependent state space $\tilde{\mathbb{S}}_n$ for our base chain is by looking at the number of points inside the boxes. Here we only allow a leaf box to be split if each resulting child box will have at least $\#_n$ points in it except when one of the child boxes will be empty and the other child box will get all of the $\#_n$ or more points from its parent box. This way, we ensure that all the leaves of a tree in $\tilde{\mathbb{S}}_n$ have at least $\#_n$ points, provided they have any points at all. This splitting procedure is more data-driven than the previous two and also concentrates the base chain over SRP states whose partitions refine on the locations of the given data set $x_1, x_2, \ldots, x_n$. Observe that for a tree s at the boundary of splittable states in $\tilde{\mathbb{S}}_n$ only a subset of its leaf nodes are splittable by this criterion. For instance, a leaf node with no data points in its associated leaf box cannot be split further and a leaf box with at least $\#_n$ many points in its leaf box cannot be split further if the split will result in a nonempty child box with fewer that $\#_n$ points in it. We refer to the set of leaf boxes corresponding to this set of splittable leaf nodes of s as $\hat{\ell}(s)$ and note that $\hat{\ell}(s) \subseteq \ell(s)$ for all $s \in \tilde{\mathbb{S}}_n$. We take $\#_n$ to either be a constant, say 1 or 2, for all values of n in our simulations and thereby ensure that $\tilde{\mathbb{S}}_n$ is one of the largest finite state spaces. We can also allow $\#_n$ to be a sublinear function of n, say $\#_n = n^\alpha$ with an appropriate $\alpha < 1$, in order to parametrically control the size of $\tilde{\mathbb{S}}_n$. In all the simulations carried out here, we use this $\#_n$-specified splitting rule and fix $\#_n = 1$ in order to work conservatively with a large state space $\tilde{\mathbb{S}}_n$. The finite state space $\tilde{\mathbb{S}}_n$ determined by such a $\#_n$-specified splitting rule has advantages due to its data-dependent partitioning nature and thereby better computational performance when compared to the exclusively n-dependent global splitting rules based on $\overline{m}_n$, the maximal number of splits allowed or on $\overline{\delta}_n$, the maximal depth of a leaf node. We sometimes use these two globally determined boundary of unsplittable states in logical conjunction with this $\#_n$-based data-dependent rule in order to study the effects of $\overline{\delta}_n$ and $\overline{m}_n$ on the jointly determined state space $\tilde{\mathbb{S}}_n$. In all simulation experiments of Section 4, we found that the effect of the $\#_n$-determined state space $\tilde{\mathbb{S}}_n$ is minimal on the performance of the posterior mean histogram estimate, provided $\tilde{\mathbb{S}}_n$ is large enough to contain partitions that can represent the underlying unknown density. We observe no significant change in integrated absolute errors when $\#_n = 0, 1, 2, 3$ because states close to these boundaries have the most number of

leaves with either 0 or $\#_n$ points in their leaf boxes and such states are rarely visited by our Markov chain due to the strong penalizing effect of the Catalan prior.

Base Markov Chain. Consider the *stay-split-merge* base Markov chain $\{Y(t)\}_{t\in\mathbb{Z}_+}$ on the state space $\tilde{\mathbb{S}}_n$ with initial state $Y(0) = s_0$. We propose to stay in the current state with positive probability σ and move to another state with probability $1-\sigma$. If a move is chosen, it can be a permitted bisection or a reunion with equally probability $(1-\sigma)/2$. If a bisection is chosen, each splittable leaf node in the current state s has an equal probability $(1-\sigma)/2|\hat{\ell}(s)|$ of being bisected. Similarly, if a reunion is chosen, each cherry node in s has an equal probability $(1-\sigma)/2|c(s)|$ of having its sibling nodes reunited to itself. Then, the transition probabilities between any two states $s, s' \in \tilde{\mathbb{S}}_n$ are:

$$Q(s,s') = \begin{cases} (1-\sigma)/2|\hat{\ell}(s)| & \text{if } s \text{ can be split once to get } s' \\ (1-\sigma)/2|c(s)| & \text{if } s \text{ can be reunited once to get } s' \\ \sigma & \text{if } s = s' \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

The chain $\{Y(t)\}_{t\in\mathbb{Z}_+}$ on the finite state space $\tilde{\mathbb{S}}_n$, that is obtained by the $\#_n$-specified splitting rule with $\#_n = 1$ for instance, is irreducible since we can eventually go from any state s to any other state s' and vice versa by reuniting cherries to reach s_0 from s and then from s_0 to s' by selectively splitting leaves (in effect by reversing the reunion operations that take you from s' to s_0). Note that any SRP state can be reached from the root SRP s_0 by a sequence of selective splits. The chain is also aperiodic since there is a positive probability σ of staying in the current state. Therefore, the base chain has a unique stationary distribution. This base chain $\{Y(t)\}$ is more uniformly distributed than the random walk base chain $\{X(t)\}$ on the stay-split-merge graph of SRP states but not as uniformly distributed as the uniform Metropolis-Hastings walk $\{Z(t)\}$ with base chain $\{X(t)\}$ and uniform stationary distribution on $\tilde{\mathbb{S}}_n$. We find $\{Y(t)\}$ to be better in practice when compared to $\{X(t)\}$ and $\{Z(t)\}$, the other base chains explored here.

Metropolis-Hasting Markov Chain. Using the irreducible and aperiodic base chain $\{Y(t)\}_{t\in\mathbb{Z}_+}$ on the finite state space $\tilde{\mathbb{S}}_n$ with transition matrix Q in Eq. (6) and the posterior distribution π given in Eq. (5), we can now proceed to construct a Metropolis-Hastings chain $\{S(t)\}_{t\in\mathbb{Z}_+}$ on $\tilde{\mathbb{S}}_n$ with π truncated to $\tilde{\mathbb{S}}_n$ as its stationary distribution with the following transition probabilities:

$$P(s,s') = \begin{cases} Q(s,s')a(s,s') & \text{if } s \text{ leads to } s' \text{ by a split or a merge} \\ 1 - \sum_{s\in\{z\in\mathbb{S}:z\neq s\}} Q(s,z)a(s,z) & \text{if } s = s' \\ 0 & \text{otherwise,} \end{cases}$$

where the acceptance probability is

$$a(s,s') := \min\left\{1, \frac{\pi(s')Q(s',s)}{\pi(s)Q(s,s')}\right\} \quad .$$

Gelman-Rubin Diagnostics. One of the difficulties with any MCMC algorithm is to determine when convergence has taken place. We use the heuristic Gelman-Rubin convergence diagnostic statistic [Gelman and Rubin 1992] to automatically stop $\{S(t)\}$ after trying to produce the desired number of post-burn-in thinned-out samples from the desired stationary distribution π over $\tilde{\mathbb{S}}_n$. To obtain this heuristic auto-stopping rule we run C parallel independent Metropolis-Hastings Markov chains $\left(\{S_1(t)\},\ldots,\{S_C(t)\}\right)$ for $t \in \{1,2,\ldots,M\}$ of maximum allowed length M and calculate $\hat{R}(t)$, the Gelman-Rubin convergence diagnostic statistic that gives the ratio

of between-sequence variance to the within-sequence variance for a scalar summary of the histogram states in the C chains up to time t. The heuristic Gelman-Rubin diagnostic method is based on the idea that the sample variance of the scalar summary within a single chain will be less than that in the combined sequences, if convergence has not taken place. Gelman recommends that the sequences be run until $\hat{R}(t)$ is less than 1.1 or 1.2 [Gelman and Rubin 1992]. We try to produce the required samples from the chain after $\hat{R}(t)$ for the number of leaves gets below 1.1. Finally, the algorithm stops, possibly without all the needed samples if all C chains have run for the maximum length of M. We focus on the scalar summary of the SRP state that gives the number of leaves. Our simple GR diagnostics was successful for some densities but can be misleading for others. Therefore, we look at trace plots of log likelihood values as well as the number of leaves of the current state to heuristically assess convergence.

Initial Condition. If the Metropolis-Hastings chain $\{S(t)\}$ is initialized far from the states with high posterior mass then the mixing time can be prohibitively large. Choosing an initial state $S(0) = \check{s}$ with too few splits or too many splits can lead to poor mixing of $\{S(t)\}$. Thus, we want a heuristic strategy to produce good initial states for $\{S(t)\}$. Algorithm 2 is a simple histogram partitioning scheme that traverses through $\tilde{\mathbb{S}}_n$ by a sequence of selective bisections from the root node or some given SRP tree containing the data. The leaf box for the next bisection is drawn uniformly at random from $\text{argmax}_{\boldsymbol{x}_{\rho v} \in \hat{\ell}(s)} \#\boldsymbol{x}_{\rho v}$, the set of current leaf boxes containing the most data points. Such draws are made using a randomized queue of the current leaf boxes prioritized by the number of data points. This data-driven partitioning criterion is based on the statistically equivalent blocks principle [Anderson 1966; Gessaman 1970] and prioritizes the splits on leaf boxes with the most data points. The splitting stops when we leave $\tilde{\mathbb{S}}_n$. Finally, we initialize our Metropolis-Hastings chain $S(t)$ from the state with the maximum log-posterior among all the states visited by our `InitSEBPQ`. This initialization strategy, coupled with monitoring the log-posterior and the number of leaves of $S(t)$ leads to reasonable estimates with small IAEs in all the simulation experiments of Section 4. However, acceleration by `InitSEBPQ` is not necessary for simpler unstructured densities.

ALGORITHM 2: `InitSEBPQ`

input : (i) data $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$;
(ii) s `// initial SRP`
output : $\check{s}$ to initialize the M-H MC $\{S(t)\}$.

initialize: $\mathfrak{S} \leftarrow \emptyset$ `// a list to track SRP histograms`

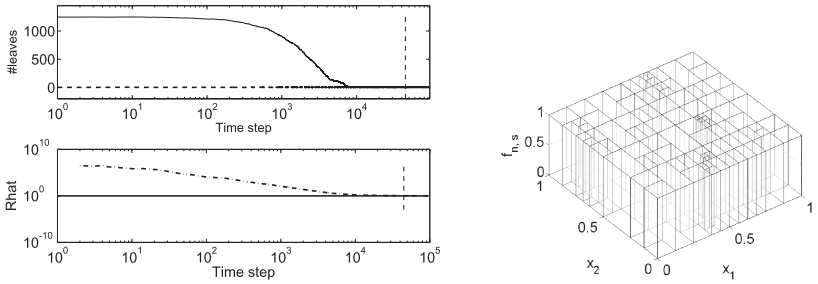
repeat
  $\boldsymbol{x}_{v^*} \leftarrow$ `Uniform`$(\text{argmax}_{\boldsymbol{x}_{\rho v} \in \ell(\mathrm{s})} \#\boldsymbol{x}_{\rho v})$ `// sample a leaf box with most data`
  bisect $\boldsymbol{x}_{v^*}$ of s
  update counts in the child nodes
  $\mathfrak{S}$`.append(s)` `// update list of states visited`
until $s \in \tilde{\mathbb{S}}_n$;
return $\check{s} \leftarrow \text{argmax}_{s \in \mathfrak{S}} \log(\text{posterior}(\mathrm{f_{n,s}}))$

4. SIMULATION RESULTS

We now present the mean integrated absolute error (MIAE) and standard error (std. err.) for various sample sizes, densities and dimensions to empirically evaluate

Table I. MIAE (std. err.) for n Samples from Uniform Density in Various Dimensions. 1000 samples collected with thin-out rate of 50. A dash (-) indicates the lack of RAM to store the sample data

n	1D	2D	10D	100D	1000D
10^2	0.1014 (0.0655)	0.1006 (0.0659)	0.1225 (0.0670)	0.1408 (0.0711)	0.1187 (0.0771)
10^3	0.0380 (0.0231)	0.0333 (0.0221)	0.0294 (0.0178)	0.0330 (0.0204)	0.0386 (0.0231)
10^4	0.0118 (0.0066)	0.0121 (0.0090)	0.0123 (0.0067)	0.0115 (0.0061)	0.0121 (0.0075)
10^5	0.0035 (0.0020)	0.0040 (0.0025)	0.0038 (0.0023)	0.0042 (0.0025)	0.0034 (0.0023)
10^6	0.0011 (0.0006)	0.0012 (0.0006)	0.0013 (0.0006)	0.0011 (0.0007)	0.0012 (0.0010)
10^7	0.0004 (0.0002)	0.0004 (0.0002)	0.0003 (0.0002)	-	-
10^8	0.0001 (0.0001)	0.0001 (0.0001)	-	-	-



(a) Trace plots of the number of leaves for two chains and its $\hat{R}$ statistic.

(b) Posterior mean histogram estimate with 97 leaf nodes with MIAE 0.001.

Fig. 5. A typical run of two Metropolis-Hastings chains based on 10^6 data points from the 2D-uniform density. The chains have been started from the root node and a state with 1247 leaf nodes. 10000 histogram samples were collected with a thin-out of 10 after burn-in at time step 44861.

the performance of our estimator. All of our programs were run on a machine with dual Intel X5670 2.93Ghz 6 core Xeon CPUs, 48GB of RAM, 2 x 320GB 15K SAS hard drives and OpenSuSE 11.2 (x86 64) OS. Mean integrated absolute errors and their standard errors were obtained from 25 replicate data sets for each density. We obtained the posterior mean histogram estimate by averaging over 10000 samples from our Metropolis-Hastings Markov chain $\{S(t)\}$ after burn-in with a thin-out rate of 10 to 50 samples. We determined burn-in time and thin-out rate heuristically from diagnostic and trace plots and emphasize that our method is not averaging perfectly independent samples from the posterior distribution. The two types of densities (unstructured and structured) that we estimate are described in the next two subsections.

4.1. Multivariate Uniform Densities

We first test the performance of our density estimator with data simulated from d dimensional uniform (dD-uniform) density, that is, the uniform density on $[0, 1]^d$. For our simulations summarized in Table I, we worked with the uniform density in dimensions 1D, 2D, 10D, 100D, and 1000D for various sample sizes as shown in Table I. A dash (-) is used in Table I to indicate machine memory (RAM) limitations for the storage of the required data.

We start the two chains with initial states as the root box (only one leaf node) and a state with a large number of leaf nodes respectively, so that we have two starting states that are "far" from each other. Histogram samples from $\{S(t)\}$ are thinned out and collected for averaging after $\hat{R}(t)$ gets below 1.1 Figure 5(b) shows the convergence diagnostic plots for the 2D-uniform density. The trace plots for the number of leaves show that the chains eventually hover around states with similar number of leaves (between 1-10 leaves). The posterior mean histogram estimate is given by Figure 5(a).

From Table I, it is clear that as the sample size n increases from 10^2 to 10^8 in multiples of 10 the MIAE decreases by at least a factor of 2 across all dimensions $d \in \{1, 2, 10, 100, 1000\}$. Observe how the MIAE of the density estimate of the uniform random vector on the hypercube $[0,1]^d$ is independent of the dimension d. This is because the target density being estimated for any d is the SRP histogram $f_{n,s_0}(x) = \mathbb{1}_{[0,1]^d}(x)$ at the root SRP $s_0 \in \tilde{S}_n$. Observe that our posterior mean estimate is so close in MIAE to the true density and the chains initialised far from s_0 quickly converge to it due to the penalizing effect of our Catalan prior on states with unnecessarily high number of leaves. We are not aware of other multivariate density estimators that can handle the sample sizes and dimensions in Table I for such highly unstructured data from this family of uniform densities on $[0,1]^d$. Thus, we seem to have found an estimator that contradicts the following pessimistic view in Moore [2000]: "If there *is* no underlying structure in the data (e.g. if it is uniformly distributed) there will be little or no acceleration in high dimensions no matter what we do." On the other hand, as the two families of structured densities in the next section show, our estimator is not immune to the curse of dimensionality when there is underlying structure in the data.

4.2. Multivariate Gaussian and Rosenbrock Densities

Approximated Functions. We now test the performance of our density estimator with data simulated from a refining family of approximations to multivariate Gaussian and Rosenbrock densities. These multivariate densities were chosen for their underlying non-uniform structures as opposed to the uniform densities in 4.1. The Rosenbrock density in d dimensions over some box $\boldsymbol{x} \in \mathbb{R}^d$ is obtained by appropriately normalizing the Rosenbrock shape given by:

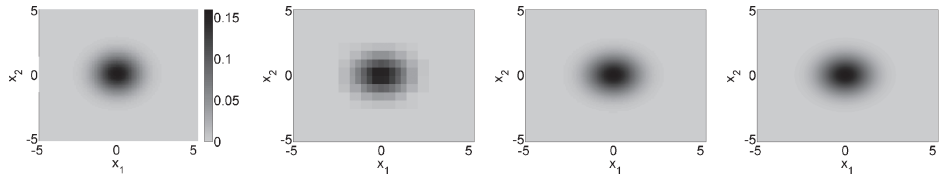
$$r_d(x) = \exp\left(-\sum_{j=2}^{d}\left(100\left(x_j - x_{j-1}^2\right)^2 + \left(1 - x_{j-1}\right)^2\right)\right), \tag{7}$$

for $d \geq 2$. In our simulation studies, the standard 1D-, 2D-, and 5D-Gaussian densities and the 2D- and 5D-Rosenbrock densities are approximated by simple functions in order to simplify the multivariate integrations during absolute error evaluations, especially in higher dimensions.

The multivariate Gaussian and Rosenbrock densities are approximated using simple functions over SRP partitions and simulated from corresponding mixtures of uniform densities using interval analytic methods of Sainudiin and York [2005]. Briefly, we use an RP tree s with root box representing the domain of the target density f. The sub-boxes associated with the leaves $\ell(s)$ of the tree represent a partition of the domain. The range of the target density f over each leaf box $\boldsymbol{x}_\mathsf{v}$ in $\ell(s)$ is enclosed rigorously in an interval $\boldsymbol{y}_\mathsf{v}$, that is, $\boldsymbol{y}_\mathsf{v} := \left[\underline{y}_\mathsf{v}, \overline{y}_\mathsf{v}\right] \supseteq \{f(x) : x \in \boldsymbol{x}_\mathsf{v}\}$, using interval arithmetic [Moore 1967]. Each leaf box $\boldsymbol{x}_\mathsf{v}$ is prioritized for the next bisection on the basis of $\mathrm{vol}(\boldsymbol{x}_\mathsf{v}) \times \left(\overline{y}_\mathsf{v} - \underline{y}_\mathsf{v}\right)$, that is, the uncertainty in the enclosure of $\mu(\boldsymbol{x}_\mathsf{v}) = \int_{\boldsymbol{x}_\mathsf{v}} f d\lambda$. The bisection stops once the partition has exactly Λ leaves and the target density at the midpoint of each leaf box is used to construct a simple function approximation to the target density. Finally, this simple function is normalized to give a weighted mixture of uniform densities over the RP tree with Λ leaf boxes. We can easily produce perfect samples to simulate data from this normalized simple function approximation of the target density. The fundamental theorems of interval analysis guarantee that the simple function converges uniformly to the target density as the mesh approaches 0 and $\Lambda = |\ell(s)|$ approaches ∞, provided

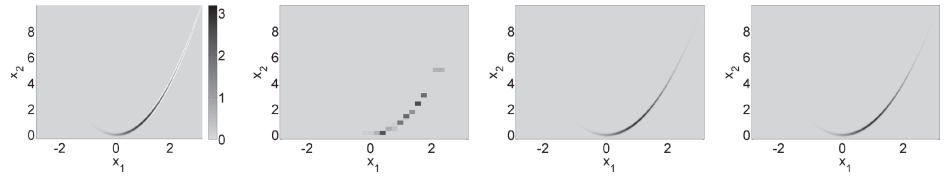
Table II. MIAE (std. err.) for n Samples from Approximated 1D-, 2D- and 5D-Gaussian Densities, 1D- , 2D- and 5D- Rosenbrock Densities. 10000 Samples Were Collected

Λ	n	Standard Gaussian densities			Rosenbrock densities	
		1D	2D	5D	2D	5D
10^2	10^4	0.0568 (0.0057)	0.0826 (0.0076)	0.0587 (0.0015)	0.0322 (0.0053)	0.0054 (0.0077)
	10^5	0.0256 (0.0019)	0.0231 (0.0022)	0.0144 (0.0005)	0.0102 (0.0022)	0.0009 (0.0012)
	10^6	0.0083 (0.0003)	0.0075 (0.0006)	0.0038 (0.0004)	0.0032 (0.0009)	0.0002 (0.0003)
	10^7	0.0025 (0.0002)	0.0023 (0.0002)	0.0011 (0.0001)	0.0023 (0.0002)	0.0001 (0.0001)
10^4	10^4	0.0583 (0.0054)	0.1650 (0.0042)	0.5137 (0.0070)	0.3620 (0.0111)	0.3186 (0.0106)
	10^5	0.0274 (0.0015)	0.0826 (0.0076)	0.2509 (0.0026)	0.1821 (0.0053)	0.0885 (0.0030)
	10^6	0.0128 (0.0004)	0.0508 (0.0005)	0.0771 (0.0009)	0.0797 (0.0006)	0.0265 (0.0008)
	10^7	0.0060 (0.0002)	0.0256 (0.0003)	0.0222 (0.0003)	0.0063 (0.0001)	0.0086 (0.0003)
10^6	10^4	0.0565 (0.0053)	0.1673 (0.0046)	0.6467 (0.0051)	0.3717 (0.0103)	1.0190 (0.0059)
	10^5	0.0274 (0.0011)	0.0932 (0.0002)	0.4655 (0.0020)	0.1982 (0.0067)	0.7250 (0.0011)
	10^6	0.0129 (0.0006)	0.0533 (0.0005)	0.3274 (0.0009)	0.1102 (0.0006)	0.4812 (0.0012)
	10^7	0.0060 (0.0001)	0.0304 (0.0002)	0.2292 (0.0034)	0.0608 (0.0049)	0.3302 (0.0004)



(a) Exact density. (b) $\Lambda = 100$. (c) $\Lambda = 10000$. (d) $\Lambda = 1000000$.

Fig. 6. 2D-Gaussian: Plots of the exact density and its approximations with Λ = 100, 10000, 1000000.



(a) Exact density. (b) $\Lambda = 100$. (c) $\Lambda = 10000$. (d) $\Lambda = 1000000$.

Fig. 7. 2D-Rosenbrock: Plots of the exact density and its approximations with Λ = 100, 10000, 1000000.

the target density is given by a locally Lipschitz arithmetical expression [Neumaier 1990, 2.1.1-3]. The Λ in Table II denotes the number of leaf nodes used to approximate the densities. Figures 6 and 7 show 2D-Gaussian and 2D-Rosenbrock densities and their Λ-specific approximations.

These simple approximating densities were chosen for two fundamental reasons: (i) to keep the true density that the data is simulated from within the class of SRP histograms for easier interpretation of our simulation results and (ii) to compute the exact IAE by taking advantage of SRP histogram arithmetic (for details, see Harlow et al. [2012]).

However, we are not simulating from the Gaussian and Rosenbrock densities any longer but from their Λ-specific approximations. To quantify the extent of these approximations we use an estimated Hellinger distance between the original target density and its Λ-specific approximation. The Hellinger distance can be estimated from the first two sample moments of each distribution. The sample moments for the Rosenbrock densities are obtained from the Moore rejection sampler in Sainudiin and York [2005]. Figure 8 shows the estimated Hellinger distances for various

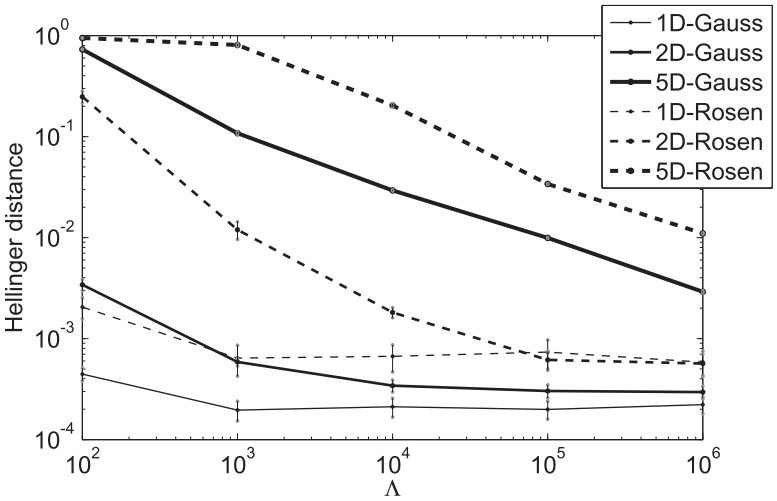
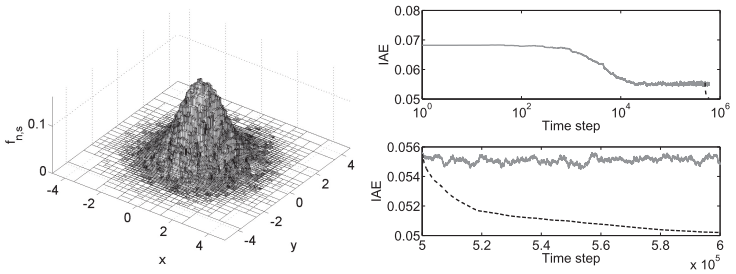


Fig. 8. Hellinger distances for the Gaussian (solid line) and Rosenbrock (dashed lines) densities.

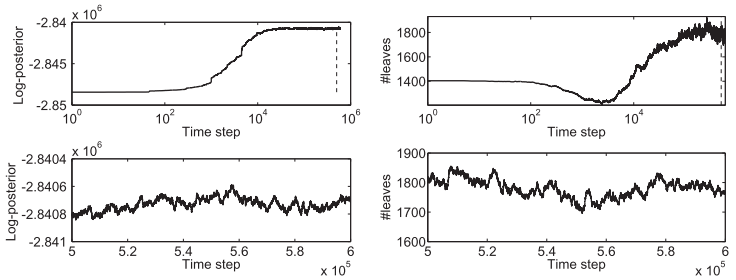
Gaussian (solid lines) and Rosenbrock (dashed lines) densities from their Λ-specific approximations based on 10^7 samples. As the dimension increases, the Hellinger distance increases for each density as well, but decreases and eventually stabilizes as Λ increases. Overall the Hellinger distances for the more complex Rosenbrock densities are higher than that of the centrally concentrated Gaussian densities across dimensions. The curse of dimensionality for a non-uniform density thus manifests itself in terms of the size of Λ needed to approximate it, in a manner that depends on the complexity of its underlying structure. The estimated Hellinger distances in Figure 8 highlight the limitations of using Λ-specific SRP partitions to approximate a desired density, especially as the dimension increases. However, the simulation of data from the approximations allows us to compute the integrated absolute errors exactly in high dimensions using arithmetic over trees representing SRP histograms.

From Table II, it is clear that as the sample size n increases from 10^2 to 10^7 in multiples of 10 the MIAE decreases in each Λ-specific approximation of the multivariate Gaussian and Rosenbrock target densities. Once again, we simulate n data points from each Λ-specific approximation given by an SRP histograms and efficiently compute the IAE of our Bayesian SRP histogram estimate. We get the MIAE and standard error from 25 replications. The Hellinger distance of each Λ-specific SRP histogram from the target (Gaussian or Rosenbrock) density is given in Figure 8. Observe that the Hellinger distance of the λ-specific approximation for the Rosenbrock density is much larger than that for the Gaussian density for each Λ. Thus, the Rosenbrock density has a higher structural complexity than the Gaussian density in terms of requiring an SRP histogram with more leaf boxes or Λ to approximate it to a given Hellinger distance. We ensured that when $\Lambda = 10^6$ the Hellinger distance to Gaussian or Rosenbrock density is well below 0.05 in all our simulation experiments. The MIAE values clearly show that the dimension as well as the structural complexity of the density determines the performance of our estimator. We use trace plots of the log-posterior and the number of leaf nodes shown in Subfigures 9(c) and 9(d) to heuristically determine convergence of the chain. Notice how the IAE for current histogram state of the chain (gray) is worse than that of the current average histogram state (black) in Figure 9(b).



(a) Posterior mean histogram estimate.

(b) IAEs for current (gray) and averaged states (black).



(c) Trace plots for log-posterior.

(d) Trace plots for number of leaves.

Fig. 9. The 2D-Gaussian density and its resulting posterior mean histogram estimate with 6022 boxes with IAE = 0.0502. The starting state has 1404 leaf nodes. Burnin is taken at time step 500000 and 10000 samples are collected with a thin-out of 10.

5. DISCUSSIONS AND CONCLUSIONS

The algorithms described in this article to obtain the posterior mean of adaptive, regularly paved, multivariate histograms are implemented in MRS: a C++ class library for statistical set processing and publicly available under the terms of the GNU General Public License from `http://www.math.canterbury.ac.nz/ r.sainudiin/codes/mrs/`. We have shown that the algorithm performs well for massive data problems in high dimensional settings using simulations from multivariate uniform densities and mixtures of uniform densities that approximate multivariate Gaussian and Rosenbrock densities. We chose mixtures of uniform densities to simplify the multivariate integrations in absolute error evaluations. Given that the space of regular paving histograms with their countably infinite family of partitions in $\mathbb{S}_{0:\infty}$ are dense in the space of continuous densities over the root box, one can get uniformly close to any continuous density when given enough data points from the space of SRP histograms. The rate of uniform convergence, in terms of the number of splits, to a particular density within a given class, from the space of SRP histograms for a given sample size n would shed light on the statistical efficiency of our estimator. However, we are able to achieve low integrated absolute errors because of the large sample sizes we assume in our massive data setting. Such a large number of data points allow our Markov chains to dive as deep as necessary into the SRP space of trees to get to the partitions of the true densities that are generating the data.

We are currently using heuristics for determining when the chain has converged. It is not straightforward to produce perfect samples from the posterior distribution on the state space of adaptive SRP histograms due to its size and combinatorially complicated transition structure. We used one prior distribution in this study and show that the

simulations are reasonably good when the size of the data is large. The effect of other prior distributions on the posterior (especially for smaller sample sizes) and other base chains on mixing time should be further explored.

The methods developed in this project consider only partitions formed by successive bisections of the data space on the widest dimension of the box being bisected. There are two important constraints in this: first, we consider only bisections, or division of a box into two equal-volume halves. Secondly, the choice of basis, or dimension to split on, is not directly data driven. These constraints give the partitions very attractive qualities from the point of view of creating binary tree structures to represent the histograms, and also mean that the process of adding and averaging histograms is relatively straightforward. More crucially, this simple bisecting strategy that only depends on the box as opposed to the structure of the data points it contains allows us to easily obtain integrated absolute errors for our estimates from the Λ-specific approximations to the true density with SRP trees. Once we allow more data-dependent bisecting strategies, which could potentially improve the partitioning scheme, as discussed below, we lose our efficient IAE computations in high dimensional settings – a feature that allows us to study performance measure of our estimators in the universal IAE scale of $[0, 2]$.

For any root box $\boldsymbol{x}_\rho \in \mathbb{IR}^d$, a typical nested sequence of Pólya tree partitions with the uniformly distributed base measure is the sequence

$$\{s_{di\,di\cdots di\,di}\}_{i=0}^{\infty}, \quad s_{di\,di\cdots di\,di} \in \mathbb{S}_{2^{di}-1}, \quad \boldsymbol{x}_\rho \in \mathbb{IR}^d \ .$$

Posterior inference of density estimates based on Pólya trees [Lavine 1992] for example can be strongly influenced by the choice of partition. This is a direct consequence of the fact that all random distributions from a Pólya tree have a common partition in a nested sequence of partitions. Extensions, such as a mixture of Pólya trees [Lavine 1992] and randomized Pólya trees [Paddock et al. 2003], alleviate this problem by allowing the random distributions to have different partitions. Our space of SRP histograms $\tilde{\mathbb{S}}_n \subset \mathbb{S}_{0:\infty}$ is quite different from the partition generated by a nested sequence of Pólya trees because it contains about $\sum_{k=0}^{2^{di}-1} C_k$ many histograms with distinct partitions in $\mathbb{S}_{0:2^{di}-1}$ that are made of splits no larger than $2^{di}-1$ such that they are not all nested in the strict Pólya sense. Our density estimator is the posterior mean over this SRP histogram space, and therefore is not reliant on a single partition of the sample space. Thus, our partitioning scheme and the space of histograms in $\mathbb{S}_{0:\infty}$, that are dense in the space of continuous densities on $\boldsymbol{x}_\rho$, seem not to influence the posterior estimates too much at least for the target densities in our simulation studies. The key motivation for our method is the ability to perform arithmetic over density estimates with tree-based partitions. A thorough study of the effect of prior distributions would be a natural sequel to this work.

A current limitation of our posterior mean density estimate over SRP histograms is the approximation of the likelihood of the data given an s by the maximum likelihood value from the histogram on $\ell(s)$, the leaf boxes of s. A fully Bayesian approach involving a prior distribution on a class of simple functions over leaf boxes of s that are non-negative and integrate to 1 by an adaptation of Dirichlet distributions used in the constructions of classical Pólya trees [Lavine 1992, 1994] would be a natural extension of our estimator. Such a Dirichlet process over SRP histograms would have the arithmetical efficiency of the dense space of SRP histograms as well as the fully Bayesian setting of classical Pólya trees and their mixtures. We hope that research in this integrative direction will continue.

Our density estimator copes well with high dimensional data with little structure, while others [Gray and Moore 2003] cope well with data with underlying structure.

Clearly, getting a density estimator that can cope well with both structured and unstructured data is harder. We suggest some trivial and non-trivial extensions that will help SRPs to better cope with structured high dimensional data. If we use the sample mean and the sample variance in each SRP node, then we can make the bisecting procedure for SRPs much closer to that of the *multiresolution kd-trees* of Gray and Moore [2003] as follows. We can now bisect a leaf box along the first coordinate with largest sample variance either at the mid-point or at the sample mean at this coordinate. Note that plane binary trees still represent the SRPs formed under this new way of bisecting and therefore SRP trees are still closed under union operations. Therefore, averaging operations can be done with minor modifications. However, once we allow bisections to be data-dependent as opposed to being box-dependent, we cannot easily obtain integrated absolute errors for our estimates from the Λ-specific approximations to the true density with SRP trees. For structured data in dimensions larger than 10, the *ball trees* of Gray and Moore [2003, Sec. 3.2], which replace the boxes in kd-trees with centroids and spheres, and built with the *anchor's hierarchy* [Moore 2000], perform significantly better by growing sub-exponentially in the number of dimensions. It would be interesting to extend arithmetic over densities represented by ball trees. Given that plane binary trees can represent ball trees as well as SRP trees, we can see that ball trees just like SRP trees are closed under union operations. However, arithmetic operations with ball trees may need more refined notions than the space partitions used for SRP trees. We hope that new research will also continue in the direction of arithmetic for ball trees.

ACKNOWLEDGMENTS

R. Sainudiin would like to thank Warwick Tucker for an introduction to regular sub-pavings, Andrew W. Moore for discussions on metric trees with cached statistics, Professor R. V. Ramamoorthi and B. V. Rao for encouraging discussions on Bayesian nonparametrics and Rua Murray for discussions on measure theory.

REFERENCES

Akaike, H. 1974. A new look at the statistical model identification. *Automat. Control IEEE Trans. 19,* 6, 716–723.

Anderson, T. W. 1966. Some nonparametric multivariate procedures based on statisticallly equivalent blocks. In *Multivariate Analysis*, P. R. Krisnaiah Ed., Academic Press, New York, 5–27.

ANSI/IEEE 754-1985. Standard for Binary Floating-Point Arithmetic. IEEE, New York.

Birgé, L. and Rozenholc, Y. 2006. How many bins should be put in a regular histogram. *ESAIM: Probab. Stat. 10*, 24–45.

Breiman, L., Friedman, J., Stone, C., and Olshen, R. 1984. *Classification and Regression Trees* 1st Ed. Chapman and Hall/CRC.

Castellan, G. 1999. Modified akaike's criterion for histogram density estimation. Tech. rep., Université Paris XI.

Gelman, A. and Rubin, D. B. 1992. Inference from iterative simulation using multiple sequences. *Stat. Sci. 7,* 4, 457–472.

Gessaman, M. P. 1970. A consistent nonparametric multivariate density estimator based on statistically equivalent blocks. *Ann. Math. Stat. 41,* 4, 1344–1346.

Gray, A. G. and Moore, A. W. 2003. Nonparametric density estimation: toward computational tractability. In *Proceedings of the SIAM International Conference on Data Mining*.

Harlow, J., Sainudiin, R., and Tucker, W. 2012. Mapped regular pavings. *Reliab. Comput. 11*, 252–282.

Hartigan, J. A. 1996. Bayesian histograms. In *Bayesians Statistics 5*, J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith Eds., Claredon Press, Oxford, UK, 211–222.

Jackson, B., Scargle, J. D., Barnes, D., Arabhi, S., Alt, A., Gioumousis, P., Gwin, E., San, P., Tan, L., and Tsai, T. T. 2005. An algorithm for optimal partitioning of data on an interval. *IEEE Sig. Proc. Lett. 12*, 105–108.

Jaulin, L., Kieffer, M., Didrit, O., and Walter, E. 2001. *Applied Interval Analysis with Examples in Parameter and State Estimation, Robust Control and Robotics*. Springer-Verlag, London.

Klemelä, J. 2007. Density estimation with stagewise optimization of the empirical risk. *Mach. Learn. 67,* 3, 169–195.

Klemelä, J. 2009a. DELT: Estimation of multivariate densities with adaptive histograms. University of Oulu, Finland. R package version 0.8.0.

Klemelä, J. 2009b. Multivariate histograms with data-dependent partitions. *Statistica Sinica 19,* 1, 159–176.

Knuth, K. H. 2006. Optimal data-based binning for histograms. arXiv:physics/0605197.

Lavine, M. 1992. Some aspects of polya tree distributions for statistical modelling. *Ann. Stat. 20*, 1222–1235.

Lavine, M. 1994. More aspects of polya tree distributions for statistical modelling. *Ann. Stat. 22*, 1161–1176.

Lugosi, G. and Nobel, A. 1996. Consistency of data-driven histogram methods for density estimation and classification. *Ann. Stat. 24,* 2, 687–706.

McGarvey, G. and Cloitre, B. 2005. Sequence A000108, The On-line Encyclopedia of Integer Sequences. published electronically. (http://oeis.org/A000108).

Meier, J. 2008. *Groups, Graphs and Trees: An Introduction to the Geometry of Infinite Groups*. London Mathematical Society student texts. Cambridge University Press, Cambridge.

Moore, A. W. 2000. The anchors hierarchy: Using the triangle inequality to survive high dimensional data. In *Proceedings of the 12th Annual Conference on Uncertainty in Artificial Intelligence*. AAAI Press, 397–405.

Moore, R. E. 1967. *Interval Analysis*. Prentice-Hall, Englewood Cliffs, NJ.

Neumaier, A. 1990. *Interval Methods for Systems of Equations*. Cambridge University Press, Cambridge.

Paddock, S., Ruggeri, F., Lavine, M., and West, M. 2003. Randomized Pólya tree models for nonparametric Bayesian inference. *Stat. Sinica 13*, 443–460.

Sainudiin, R. and York, T. 2005. An auto-validating rejection sampler. Tech. rep. bu-1661-m, BSCB Department, Cornell University, Ithaca, NY.

Samet, H. 1990. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA.

Scargle, J. D. 1998. Studies in astronomical time series analysis. V. Bayesian blocks, a new method to analyze structure in photon counting data. *Astrophys. J. 504,* 1, 405.

Stanley, R. P. 1999. *Enumerative Combinatorics*. Vol. 2. Cambridge University Press, Cambridge.

Stone, C. 1982. Optimal global rates of convergence for nonparametric regression. *Ann. Stat. 10,* 4, 1040–1053.

Taylor, C. C. 1987. Akaike's information criterion and the histogram. *Biometrika 74,* 3, 636–639.

Teng, G., Kuhn, K., and Sainudiin, R. 2012. Statistical regular pavings to analyze massive data of aircraft trajectories. *J. Aeros. Comput. Info. Comm. 9,* 1, 14–25.

Received October 2011; revised September 2012; accepted October 2012