The Canadian Journal of Statistics
Vol. xx, No. yy, 20??, Pages 1–25
La revue canadienne de statistique

On-line partitioning of the sample space in the Regional Adaptive algorithm

Nicolas Grenon-Godbout [1] and Mylène Bédard [1*]

[1] *Département de mathématiques et de statistique, Université de Montréal, Montréal, QC, Canada*

Key words and phrases: adaptive sampler; bimodal distribution; ergodicity; Mahalanobis distance; Markov chain Monte Carlo

MSC 2010: Primary 65C40; secondary 62P10

Abstract: The Regional Adaptive (RAPT) algorithm is particularly useful in sampling from multimodal distributions. We propose an adaptive partitioning of the sample space, to be used in conjunction with the RAPT sampler and its variants. The adaptive partitioning consists in defining a hyperplane that is orthogonal to the line joining averaged coordinates in two separate regions, and that goes through a point such that both averaged coordinates are equally Mahalanobis-distant from this point. This yields an adaptive process that is robust to the choice of initial partition, stabilizes rapidly, and is implemented at a marginal computational cost. The ergodicity of the sampler is verified through the *Simultaneous Uniform Ergodicity* and *Diminishing Adaptation* conditions. The approach is compared to the RAPT algorithm with fixed regions and to the RAPT with online recursion (RAPTOR) through various examples, including a real data application. In short, our main contribution is the development of an alternative version of RAPTOR that seems to have no obvious downside and runs 15 to 35 percent faster in the examples considered.

The Canadian Journal of Statistics xx: 1–25; 20??

Résumé: L'algorithme adaptatif régional (RAPT) est particulièrement utile pour échantillonner de distributions cibles multimodales. Nous proposons un processus adaptatif de partitionnement de l'espace d'états destiné à être utilisé avec l'algorithme RAPT et ses variantes. Le partitionnement adaptatif consiste à définir un hyperplan orthogonal au segment joignant la moyenne des coordonnées comprises dans deux régions distinctes, et passant par un point tel que les coordonnées moyennes de chacune des deux régions sont à égale distance de ce point selon le critère de Mahalanobis. Ceci mène à un algorithme robuste au choix de partition initiale, dont le processus de partitionnement se stabilise rapidement, et qui est implémenté à un coût supplémentaire marginal. L'ergodicité de l'algorithme est vérifiée à l'aide des conditions d'*ergodicité uniforme simultanée* et d'*adaptation dissipante*. La nouvelle méthode est comparée à l'algorithme RAPT avec régions fixes, ainsi qu'au RAPT avec récursion dynamique (RAPTOR) à travers différents exemples, incluant une application sur des données réelles. En résumé, notre contribution principale est le développement d'une alternative à l'algorithme RAPTOR qui ne semble avoir aucun désavantage évident et qui roule de 15% à 35% plus vite dans les exemples considérés. *La revue canadienne de statistique* xx: 1–25; 20??

1. INTRODUCTION

In the past few decades, statistical models to study real-world phenomena have been increasing both in terms of complexity and dimensionality. Such models generally produce densities that cannot be treated analytically; Markov chain Monte Carlo (MCMC) methods have thus become a device of choice to obtain samples from these complicated probability distributions.

The Metropolis-Hastings sampler (Metropolis et al. [1953]; Hastings [1970]) is at the core of the MCMC toolbox and has spurred the development of countless specialized algorithms. The

* *Author to whom correspondence may be addressed.*
E-mail: mylene.bedard@umontreal.ca

idea is to build a Markov chain with invariant distribution π (the d-dimensional distribution of interest) on a state space $\mathcal{S}$ by proposing candidates to be included in the process according to some acceptance probability. Let an initial value X_0 for the process be drawn from an arbitrary distribution μ. Then, at iteration $t+1$, the Metropolis-Hastings (MH) sampler generates a candidate $Y_{t+1} = y$ from a proposal distribution $Q(y|X_t)$ with density $q(y|X_t)$. This candidate is accepted as the next state X_{t+1} of the Markov chain with probability $\alpha(X_t, y) = \min\{1, \frac{\pi(y)q(X_t|y)}{\pi(X_t)q(y|X_t)}\}$, otherwise we set $X_{t+1} = X_t$.

A pragmatic choice is to draw candidates from a $\mathcal{N}(X_t, s_d\, \Sigma_{d\times d})$, where $\Sigma_{d\times d}$ is a positive definite covariance matrix and $s_d > 0$ is a scalar; this yields a random walk version of the MH sampler (RWMH). For the Markov chain to rapidly explore its state space $\mathcal{S}$, careful tuning of the parameters of Q is required. In traditional MCMC, these parameters are either fixed or depend on information collected from the process at time t, preserving the Markovian property of the algorithm. For instance, Roberts and Rosenthal [2001] demonstrate that $s_d\, \Sigma_{d\times d} = 2.38^2\Sigma_{d\times d}/d$ is the optimal choice for sampling from a d-dimensional normal target with covariance $\Sigma_{d\times d}$ (with d large). Nevertheless, optimally tuned samplers may still fail, in practice, to appropriately explore the distribution of interest; this is often the case with distributions formed of highly correlated components or the notorious bimodal distributions.

In this paper, we consider a more flexible form of tuning based on information available from the sample $X_0, \ldots, X_t$ and updated at every iteration. These adaptive MCMC algorithms rely on theoretical foundations of their own as they violate the Markovian property, which is the building block of traditional MCMC techniques. The celebrated Adaptive Metropolis (AM) sampler of Haario et al. [2001] updates, at every iteration, the covariance matrix of the Gaussian proposal used in the MH. The empirical estimate of $\Sigma_{d\times d}$ is obtained at a low computation cost through a recursive formula that uses all past realizations of the chain. In the wake of this contribution, an interesting collection of adaptive samplers was introduced in the statistical literature; see, for instance, Haario et al. [2005], Haario et al. [2006], Roberts and Rosenthal [2009], Solonen et al. [2012].

Convergence properties of adaptive algorithms have also been studied by several authors (*e.g.* Andrieu and Robert [2001], Atchadé and Rosenthal [2005], Andrieu et al. [2006]). In particular, Roberts and Rosenthal [2007] introduce the *Diminishing Adaptation* and *Simultaneous Uniform Ergodicity* conditions, which together guarantee the ergodicity of adaptive samplers. The first condition states that an algorithm should adapt less and less as it proceeds, while the second is a technical condition (usually satisfied in practice) ensuring that the process does not wander off; these are stated formally in Section 5.

Despite improved performances of adaptive methods by comparison to classical ones, sampling from distributions involving multimodality or strong asymmetry remains a challenge. In some cases, proposal distributions adapted using past samples might have learned the geography of π over some regions, but might require a different adaptation on features not yet explored, in alternate regions of $\mathcal{S}$. Samplers such as the Regional Adaptive MCMC (RAPT) proposed by Craiu et al. [2009] and the Regional Adaptive algorithm with online recursion (RAPTOR) by Bai et al. [2011] aim at sampling from the usually problematic bimodal distributions (with possible generalization to multimodal ones) by targeting the adaptation on various regions of the state space. RAPT assumes that users can propose a decent partition of $\mathcal{S}$; RAPTOR makes no such assumption, but the resulting sampler is more demanding computationally.

In this paper, we introduce a robust and computationally affordable adaptive process for partitioning the state space $\mathcal{S}$, to be used in conjunction with the RAPT of Craiu et al. [2009]. The idea is to partition the space with hyperplanes that are orthogonal to the lines joining pairs of regional sample averages. In proving the ergodicity of the new sampler, we show that the angle between successive hyperplanes converges to 0. This leads to an interesting compromise

between RAPT and RAPTOR, with performances that compare favorably to those of Bai et al. [2011]. In our illustrations, regions rapidly adapt to nearly ideal partitions, even with extremely poor initialization settings. The proposed sampler also performs well in our real data application, where RAPT requires a separate optimization step to obtain a quality partition. In summary, our newly developed OPRA acts similarly to RAPTOR but runs in about 65 to 85 percent of the time. Naturally, if target computations were so demanding that adaptation became insignifiant in comparison, then OPRA would take as long as RAPTOR to run.

2. MULTIMODALITY IN THE LITERATURE

Regional adaptive samplers are not the only way of tackling multimodality; a classical approach in this context is to use multiple parallel chains started according to an over-dispersed distribution with respect to π (Gelman et al. [1992]). Information from these parallel chains may also be used to recursively update the parameters of the proposal distributions; this is known as inter-chain adaptation (INCA, see Craiu et al. [2009]).

Another popular avenue is to rely on tempering (Neal [1996]; Geyer and Thompson [1995]). A temperature parameter T is included in the target and as T increases (*i.e.* as the temperature rises), the target π_T becomes flatter and therefore easier to explore. Implementing several parallel algorithms with increasing temperatures and exchanging states among chains of different temperatures is known as parallel tempering. The equi-energy sampler of Kou et al. [2006] runs several Metropolis-Hastings chains at different temperatures but also builds energy rings, *i.e.* sets containing states of similar densities. At every iteration, a given chain can either propose a local move or sample a state of similar energy from a neighbouring chain, which helps the process crossing low-energy barriers.

Lately, several approaches based on the concept of free energy have been proposed. The Wang-Landau sampler (Landau et al. [2004]) partitions $\mathcal{S}$ and then artificially increases the energy of visited states. In other words, the density of a state is decreased by a predetermined factor every time it is visited, so this state (and the region to which it belongs) has fewer chances of being visited again, favoring the exploration of regions that have yet to be visited. Repeating this with finer factors (*i.e.* factors closer to 1) in subsequent sweeps leads to a thorough exploration of the space. Another method, introduced by Chopin et al. [2012], uses the free energy of a reaction coordinate to build a bias for the MCMC sampler; this allows to move more freely between the different modal regions of the initial target distribution. In their paper, Bornn et al. [2013] automate the Wang-Landau algorithm by adapting its proposal distributions and space partition.

These methods all have their pros and cons. Energy-based samplers are quite effective at crossing low-density barriers but require extensive user input. Accordingly, the implementation of these methods is often difficult for practitioners, both in terms of coding and tuning. Our goal is to introduce a method that is as simple and user-friendly as possible, with virtually no tuning left to users. The method we study for tackling multimodal targets is regional adaptation, which can be combined to other tools such as INCA or parallel tempering if desired. For now, the context in which the sampler is described is kept as plain as possible.

3. FRAMEWORK

To illustrate the need for regional adaptation, let $\mathcal{N}\left(x;\mu,\sigma^2\right)$ denote the normal density with mean μ and variance σ^2; now consider the density $\pi(x)=\frac{1}{2}\mathcal{N}\left(x;-6,4\right)+\frac{1}{2}\mathcal{N}\left(x;6,\frac{1}{4}\right)$, whose left mode is more spread out than the right one. To sample from this target, we could use a RWMH with a normal proposal distribution. In that context, the proposal variance $\sigma^2=140$ gives rise to the fastest exploration of $\mathcal{S}$ (*i.e.* optimizes the average quadratic variation defined below), and is large enough for the sampler to jump from one mode to the other. It however generates

several candidates that are located in low-density areas, so only 17% of candidates turn out to be accepted.

There exist, in MCMC theory, different notions of efficiency. Hereafter, the term efficiency is used as a measure of how rapidly the Markov chain explores its state space. For d-dimensional RWMH chains, this might be measured by the (standardized) average quadratic variation,

$$\text{AQV} = \frac{1}{nd}\sum_{t=1}^{n}(X_t - X_{t-1})^T D^{-1}(X_t - X_{t-1})\,, \tag{1}$$

where n is the number of iterations and D is a $d \times d$ diagonal matrix whose entries are the variances of the d components, $(\sigma_1^2, \ldots, \sigma_d^2)$. The optimization of this measure encourages a trade-off between large and frequently accepted moves, without penalizing components featuring smaller scales.

Because of the geography of π, $\sigma^2 = 140$ is obviously too large for individually exploring each of the modes. It would be preferable to use different proposal variances on different regions of $\mathcal{S}$; we could use $\sigma_L^2 = 22.6$ and $\sigma_R^2 = 1.4$, the variances that optimize the AQV given that the chain is restricted to the left or right mode, respectively. To know which of σ_L^2 or σ_R^2 to use, we however need a frontier that partitions $\mathcal{S} = \mathbb{R}$. Here a good partition is easy to find as we clearly want each region to be unimodal, but examples are often not that simple. In asymmetrical and correlated contexts for instance, quality partitions are often difficult to find.

In order to define a notion of partition quality, let us suppose that we have access, for each partition $\{\mathcal{S}_1, \ldots, \mathcal{S}_K\}$ of $\mathcal{S}$, to the corresponding K optimal proposal variances (meaning that the i-th variance optimizes the AQV of a chain whose target is the restriction of π to $\mathcal{S}_i$). Although several good partitions of $\mathcal{S}$ may exist (possibly based on various efficiency criteria), in the current framework, we think of an ideal partition as one that maximizes the spectral gap, and thus that optimizes the mixing time of the chain. In most contexts, it is intuitively clear that the overall mixing of the sampler is limited by the slowest mixing region.

More formally, let us define the standardized AQV of Region i as

$$AQV_i = \frac{1}{d\sum_{t'=1}^{n}\mathbb{1}_{\mathcal{S}_i}(X_{t'-1})}\sum_{t=1}^{n}\mathbb{1}_{\mathcal{S}_i}(X_{t-1})(X_t - X_{t-1})^T D_i^{-1}(X_t - X_{t-1})\,,$$

where D_i is the diagonal variance matrix of π restricted to Region i and where again we suppose that for each partition, we have access to the corresponding K optimal proposal variances. Following the above intuition, an ideal partition is one that maximizes $\min_{i\in\{1,\ldots,K\}} AQV_i$. If several partitions feature the same maximum, our choice narrows to the partition that maximizes $\sum_{i=1}^{K} AQV_i/K$. In what follows, any reference to partition quality thus refers to this upper-level, inter-region notion of optimality, as opposed to the lower-level, within-region optimality of the proposal variances.

For now, suppose that a decent partition $\{\mathcal{S}_1, \ldots, \mathcal{S}_K\}$ of $\mathcal{S}$ is available. In that situation, one could draw a candidate Y_{t+1} from $\sum_{i=1}^{K}\mathbb{1}_{\mathcal{S}_i}(X_t)Q_i(Y_{t+1}|X_t)$, where $\mathbb{1}_A(x)$ is the indicator function that x belongs to A and Q_i is the proposal distribution used in Region i. A RWMH sampler would then be carried as usual for moves within each region, and by carefully computing the acceptance function for moves crossing the frontier (*e.g.* $x \in \mathcal{S}_1$ and $Y_{t+1} \in \mathcal{S}_2$).

The performance of the above sampler heavily relies on the quality of the partition. In Craiu et al. [2009] and Bai et al. [2011], two different avenues are explored to alleviate the effect of suboptimal partitions, namely the RAPT and RAPTOR samplers. The remainder of this section describes the idea behind these samplers. The adaptive partitioning process and its generalization to $K > 2$ regions are introduced in Section 4, and theoretically justified in Section 5. Using sim-

ulation studies and a real data application, the performance and robustness of the new algorithm are compared to those of RAPTOR in Sections 6 and 7. Although valid more generally, discussions about RAPT and the adaptive partitioning process will be restricted to the RWMH version with Gaussian proposal distributions.

3.1. RAPT Algorithm

This sampler makes the assumption that users have access to a decent partition of $\mathcal{S}$. To account for suboptimal partitions $\{\mathcal{S}_1, \ldots, \mathcal{S}_K\}$ (which could cause Q_2 to be better than Q_1 for some $x \in \mathcal{S}_1$, say), candidates are generated from a mixture of the distributions $Q_1, \ldots, Q_K$. Specifically,

$$Q(y, t|x) = \sum_{i=1}^{K} \mathbb{1}_{\mathcal{S}_i}(x) \sum_{j=1}^{K} \lambda_{i,j}(t)\, Q_j(y|x) \,,$$

where mixture weights vary according to the region of the current state x and $\sum_j \lambda_{i,j}(t) = 1$, $\forall i, t$. Ideal weights $\lambda_{i,j}(t)$ reflect the extent to which Q_j is more or less appropriate than the other proposal distributions in Region i. It is naturally difficult to select appropriate weights, so these parameters are adapted as the sampler proceeds to account for newly gained information. With Gaussian proposals, the AM sampler may also be used to update the K covariance matrices; Q_j is then updated using sample values from Region j exclusively, $j = 1, \ldots, K$.

Various approaches may be used to compute the mixture weights. For a RWMH version of the RAPT algorithm define, for $i, j = 1, \ldots, K$,

$$W_{i,j}(t) = \{0 \leq s \leq t-1 : x_s \in \mathcal{S}_i \text{ and } y_{s+1} \text{ is generated from } Q_j\} \,, \tag{2}$$

and $|W_{i,j}(t)|$ the number of elements in the set. To estimate time-t weights, the AQV of moves from $x_s \in \mathcal{S}_i$ to a new state x_{s+1} generated from Q_j is used:

$$d_{i,j}(t) = \frac{\sum_{s \in W_{i,j}(t)} \|x_{s+1} - x_s\|^2}{|W_{i,j}(t)|} \,, \quad i, j = 1, \ldots, K \,,$$

where $\|\cdot\|$ is the Euclidean norm. For $x \in \mathcal{S}_i$, time-t mixture weights are chosen proportional to $d_{i,j}(t)$,

$$\lambda_{i,j}(t) = \begin{cases} \frac{d_{i,j}(t)}{\sum_{l=1}^{K} d_{i,l}(t)} \,, & \text{if } \sum_{l=1}^{K} d_{i,l}(t) > 0 \,, \\ \frac{1}{K} \,, & \text{otherwise} \,, \end{cases} \qquad \text{for} \quad i, j = 1, \ldots, K \,.$$

Proposal distributions Q_j giving rise to larger $d_{i,j}(t)$ will then be attributed heavier mixture weights. It should be mentioned that RAPT relies on a pre-adaptation period; therefore, events such as $W_{i,j}(t) = \emptyset$ will not occur if the pre-adaptation is long enough for moves from each i to each j to be proposed at least once.

To enable a good flow between regions, Craiu et al. [2009] add a global adaptive component to the mixture. The proposal distribution then becomes

$$Q(y, t|x) = (1-\beta) \sum_{i=1}^{K} \mathbb{1}_{\mathcal{S}_i}(x) \sum_{j=1}^{K} \lambda_{i,j}(t) Q_j(y, t|x) + \beta\, Q_S(y, t|x) \,,$$

where $0 < \beta < 1$ and Q_S adapts using all past sample values. The weight of this component is kept fixed to guarantee a positive probability, at any t, of crossing the frontier between regions

(usually $\beta = 0.3$, see Guan et al. [2007]). This sampler then accepts candidates with probability

$$\alpha_t^*(x,y) = \begin{cases} \frac{\pi(y)}{\pi(x)}\ , & \text{if } x,y \in \mathcal{S}_k\ , \\ \frac{\pi(y)\left((1-\beta)\sum_{i=1}^K \lambda_{\ell,i}(t) q_i(x,t|y) + \beta q_S(x,t|y)\right)}{\pi(x)\left((1-\beta)\sum_{j=1}^K \lambda_{k,j}(t) q_j(y,t|x) + \beta q_S(y,t|x)\right)}\ , & \text{if } x \in \mathcal{S}_k, y \in \mathcal{S}_\ell, k \neq \ell\ . \end{cases} \tag{3}$$

RAPT is generally implemented with two regions. If the initial partition of a version with K regions is harder to initialize, it is however not significantly more expensive in terms of computational effort. In fact, when the geography of π calls for an additional region, the resulting efficiency gain usually outweighs the computational overhead introduced.

The principal challenge in implementing RAPT is to determine a good partition of $\mathcal{S}$ without extensively studying π. A poorly chosen partition may slow mixing or contribute to overlooking a mode during preadaptation. Given that RAPT recursively updates regional means and covariance matrices, it would make sense to also adapt the regions $\mathcal{S}_i$, $i = 1, \ldots, K$.

To recursively approximate the ideal partition we could, at any $t \geq 0$, make it equally difficult for all K sub-samplers to travel from the center of their region to the frontier. Regions producing volatile observations will tend to push the frontier away, while those producing concentrated ones will pull it closer; eventually, the frontier should stabilize when an equilibrium between means, covariances, and partition is reached.

We note that it is usually impossible to get analytical expressions for the regional (lower-level) optimal proposal variances mentioned above. As in Craiu et al. [2009], we rely on time-t proposal scalings $2.38^2\Sigma_i(t)/d$ for $i = 1, \ldots, K$, which are proportional to the updated regional covariance matrices $\Sigma_i(t)$. These scalings are not necessarily optimal in our context (*e.g.* for hopping between regions), but they each are as close to optimality (over a given region) as theory brings us.

3.2. RAPTOR Algorithm

We now summarize the regional adaptive algorithm with online recursion. For more details on its implementation, we refer the reader to Bai et al. [2011].

RAPTOR is an appropriate choice of sampler when the target π is well approximated by $\tilde{\pi}(x) = \sum_{i=1}^K \omega_i\ \mathcal{N}(x; \mu_i, \Sigma_i)$. The idea is to update the parameters $\{(\omega_i(t), \mu_i(t), \Sigma_i(t))\,; i = 1, \ldots, K\}$ on the fly, using recursion formulas based on the EM algorithm and initially developed by Andrieu et al. [2006].

At every iteration, the updated mixture $\tilde{\pi}_t$ is then used to adaptively define a partition of the state space, $\mathcal{S} = \bigcup_{i=1}^K \mathcal{S}_i(t)$. At time t, the region $\mathcal{S}_i(t)$ is the set in which the i-th component of $\tilde{\pi}_t$ dominates the others:

$$\mathcal{S}_i(t) = \left\{ x : \arg\max_{i'} \mathcal{N}(x; \mu_{i'}(t), \Sigma_{i'}(t)) = i \right\}\ .$$

Once the time-t partition is updated, a candidate y_{t+1} is generated from

$$\begin{aligned} Q(y,t|x) = (1-\beta) \sum_{i=1}^K \mathbb{1}_{\mathcal{S}_i(t)}(x)\, \mathcal{N}\left(y; x, s_d\left(\Sigma_i(t) + \epsilon I_d\right)\right) \\ +\ \beta\, \mathcal{N}\left(y; x, s_d\left(\Sigma_S(t) + \epsilon I_d\right)\right)\ , \end{aligned}$$

where $s_d = 2.38^2/d$, $\epsilon > 0$, and $\beta \in (0,1)$ is a fixed weight that controls the flow between regions. The sample covariance matrix $\Sigma_S(t)$ is updated using all past observations, while the terms ϵI_d are added to ensure that covariance matrices are positive definite. If $x \in \mathcal{S}_i(t)$, a can-

didate is thus generated using the dominant component of the mixture with probability $1-\beta$, or the global proposal distribution with probability β. The candidate is then accepted according to the usual Metropolis-Hastings rule, $\alpha_t(x,y) = \min\left\{1, \frac{\pi(y)q(x,t|y)}{\pi(x)q(y,t|x)}\right\}$, where $q(y,t|x)$ is the density of $Q(y,t|x)$.

4. ON-LINE PARTITIONING AND REGIONAL ADAPTATION (OPRA)

We present a modification of RAPT in which we partition $\mathcal{S}$ adaptively. For this discussion, we focus on $K=2$ and denote the time-$(t-1)$ partition $\mathcal{S}_1(t-1) \cup \mathcal{S}_2(t-1) = \mathcal{S}$ with $\mathcal{S}_1(t-1) \cap \mathcal{S}_2(t-1) = \emptyset$. For the new algorithm to be appealing from a practical point of view, the adaptive partitioning should remain as simple as possible and avoid extra computationally intensive calculations.

In multidimensional settings, an ideal partition of $\mathcal{S}$ is usually difficult to obtain. Most of the time, a carefully chosen hyperplane leads to a very good approximation of the sought-after frontier. Moreover, the mixture proposal distribution of RAPT protects the user, to some extent, against suboptimal partitions; this will be particularly useful before the partitioning process stabilizes. Define, for $t \geq 1$,

$$W_i(t) = \{0 < s \leq t : x_s \in \mathcal{S}_i(s-1)\}\,, \quad i = 1,2\,, \tag{4}$$

the sets containing indices of values that were progressively added to each of the two regions, up to current time t. These sets are used to keep track of observations in each region. Intuitively, we should in fact write $\{0 < s \leq t : x_s \in \mathcal{S}_i(t-1)\}$, but we would need to reevaluate the region of all observations every time the partition is updated. As will be explained in Remark 2, computational gains from progressively classifying observations are worth the minor loss in precision.

To determine the hyperplane used at time t, we need the time-t sample averages in each region,

$$\overline{x}_i(t) = \frac{1}{|W_i(t)|} \sum_{s \in W_i(t)} x_s\,, \quad i = 1,2\,,$$

with $|W_i(t)|$ the number of values in the set $W_i(t)$ for $i = 1,2$. We also need estimates of the covariance matrices in each region, $\Sigma_1(t), \Sigma_2(t)$; these quantities are readily available from the adaptation of the proposal distributions Q_1, Q_2 through the AM algorithm, and are computed recursively.

At time t, we wish to define a hyperplane that is orthogonal to the segment joining the sample averages $\overline{x}_i(t)$, $i = 1,2$. If the target densities over $\mathcal{S}_1(t-1)$ and $\mathcal{S}_2(t-1)$ possess relatively similar shapes and scales, then it will be optimal to require that the hyperplane dividing the space go through the middle point of this segment. It will then be expressed as $a_t^T X = b_t$, with

$$a_t = \overline{x}_1(t) - \overline{x}_2(t) \quad , \quad b_t = a_t^T \left(\frac{\overline{x}_1(t) + \overline{x}_2(t)}{2}\right)\,.$$

This hyperplane represents the frontier between the updated regions $\mathcal{S}_1(t)$ and $\mathcal{S}_2(t)$. We determine the region to which a value x belongs according to

$$x \in \begin{cases} \mathcal{S}_1(t) & \text{if } a_t^T x \geq b_t\,, \\ \mathcal{S}_2(t) & \text{if } a_t^T x < b_t\,. \end{cases} \tag{5}$$

To account for possible shape and scale discrepancies of the target density between regions, we may use a weighted average of $\overline{x}_1(t), \overline{x}_2(t)$ to obtain the parameter b_t defining the position of the hyperplane (by opposition to its orientation). To gain intuition about this weighted average, we reexpress the problem in terms of Mahalanobis distance, a multi-dimensional generalization of the z-score that measures the number of standard deviations a point is away from the mean of a distribution. The Mahalanobis distance of an observation $x = (x_1, \ldots, x_d)^T$ from a distribution with mean $\mu = (\mu_1, \ldots, \mu_d)^T$ and covariance matrix Σ is

$$D_M(x) = \sqrt{(x-\mu)^T \Sigma^{-1} (x-\mu)} \, .$$

By imagining that the hyperplane should go through a coordinate $r(t)$ on the segment joining $\overline{x}_1(t), \overline{x}_2(t)$ such that Mahalanobis distances between $\overline{x}_i(t)$ and $r(t)$ are equal for $i = 1, 2$, then we have

$$\begin{aligned} &(r(t) - \overline{x}_1(t))^T \{\Sigma_1(t)\}^{-1} (r(t) - \overline{x}_1(t)) \\ &\quad = \; (r(t) - \overline{x}_2(t))^T \{\Sigma_2(t)\}^{-1} (r(t) - \overline{x}_2(t)) \, . \end{aligned}$$

The coordinate $r(t)$ thus satisfies $r(t) = \overline{x}_1(t) + k(t) \, (\overline{x}_2(t) - \overline{x}_1(t))$ for a certain value $k(t) \in (0, 1)$ such that

$$\begin{aligned} &k^2(t) \, (\overline{x}_2(t) - \overline{x}_1(t))^T \, \Sigma_1^{-1}(t) \, (\overline{x}_2(t) - \overline{x}_1(t)) = \\ &\quad (1 - k(t))^2 \, (\overline{x}_2(t) - \overline{x}_1(t))^T \, \Sigma_2^{-1}(t) \, (\overline{x}_2(t) - \overline{x}_1(t)) \; . \end{aligned}$$

If $\Sigma_1(t) = \Sigma_2(t)$ then $k(t) = 1/2$ and we are back to the situation described above (hyperplane going through the middle point of the segment); otherwise,

$$k(t) = \frac{\sqrt{z_1 z_2} - z_2}{z_1 - z_2} \; = \; \frac{\sqrt{z_2}}{\sqrt{z_1} + \sqrt{z_2}} \, , \tag{6}$$

with $z_i = (\overline{x}_2(t) - \overline{x}_1(t))^T \, \Sigma_i^{-1}(t) \, (\overline{x}_2(t) - \overline{x}_1(t))$. Hence,

$$\begin{aligned} a_t &= \overline{x}_1(t) - \overline{x}_2(t) \, , \\ b_t = a_t^T r(t) \; &= \; a_t^T \, \{(1 - k(t)) \, \overline{x}_1(t) + k(t) \overline{x}_2(t)\} \; . \end{aligned} \tag{7}$$

An observation x is then classified according to (5) as before.

In practice, the determination of this weighted hyperplane requires slightly more effort than the version going through the middle point. It turns out that a decomposition of the matrices $\Sigma_i(t)$, $i = 1, 2$ is readily available at every step where a change of region occurs (since it is then needed to compute α_t^* in (3)), which facilitates computations involving the inverse of those matrices.

The proposed adaptive partitioning uses quantities that are already updated in RAPT and is implemented at a marginal computational cost. In the unlikely event where one region is not visited during pre-adaptation (which suggests that the initial partition is poorly selected, or that pre-adaptation is not long enough), then one of the sample averages would be undefined ($\overline{x}_2$, say). We could then fit a hyperplane going through $\overline{x}_1$, with arbitrary orientation, say $a = (1, \ldots, 1)$. In practice, we try to avoid this situation by relying on parallel chains with initial values belonging to different initial regions. Experimental results show that the RAPT algorithm with adaptive partitioning is robust to the initial partition, in the sense that the hyperplane evolves rapidly and stabilizes near the ideal hyperplane.

4.1. Implementation

We now describe, step by step, how to implement OPRA. We first initialize the sampler: $X_0 = x_0$, $\beta = 0.3$, $\lambda_{i,j}(0) = 1/2\ \forall (i,j)$, $\Sigma_1(0) = \Sigma_2(0) = I_d$, and $\Sigma_S(0) = MI_d$, where I_d is the $d \times d$ identity matrix and M yields a Gaussian density that is over-dispersed with respect to $\mathcal{S}$. Initial regions $\mathcal{S}_i(0)$ $(i = 1, 2)$ are separated by an hyperplane $a_0^T X = b_0$ with $b_0 = x_0$ and arbitrary a_0.

Given $X_t = x$, $\Sigma_S(t)$, $\mathcal{S}_i(t)$, $\lambda_{i,j}(t)$, $\Sigma_i(t)$ for $i, j = 1, 2$, then at time $t + 1$

1. Generate a candidate Y_{t+1} from the proposal distribution

$$Y_{t+1} \sim (1-\beta) \sum_{i=1}^{2} \mathbb{1}_{\mathcal{S}_i(t)}(x) \left[\lambda_{i,1}(t) Q_1(Y_{t+1}, t|x) + \lambda_{i,2}(t) Q_2(Y_{t+1}, t|x)\right] + \beta\, Q_S(Y_{t+1}, t|x)\ ,$$

   where $Q_j(Y_{t+1}, t|x) \sim \mathcal{N}(x, \Sigma_j(t))$, $j = 1, 2$. In other words, assuming that $x \in \mathcal{S}_k(t)$, generate $\ell' \in \{1, 2\} \cup \{S\}$ according to the probabilities $\{(1-\beta)\lambda_{k,1}(t), (1-\beta)\lambda_{k,2}(t), \beta\}$, then generate $Y_{t+1} \sim \mathcal{N}(x, \Sigma_{\ell'}(t))$.
2. Define the time-$(t + 1)$ value of the Markov chain using $\alpha_t^*(x, y)$ in (3):

$$X_{t+1} = \begin{cases} Y_{t+1}, & \text{with probability } \alpha_t^*(x, Y_{t+1}) \\ x, & \text{with probability } 1 - \alpha_t^*(x, Y_{t+1}) \end{cases} .$$

3. If $\ell' = S$, go to Step 4; otherwise, (2) implies that $t \in W_{k,\ell'}(t + 1)$, hence

$$|W_{k,\ell'}(t+1)| = |W_{k,\ell'}(t)| + 1\ ,$$
$$d_{k,\ell'}(t+1) = \frac{1}{|W_{k,\ell'}(t+1)|} \left(|W_{k,\ell'}(t)| d_{k,\ell'}(t) + \|X_{t+1} - x\|^2\right)\ .$$

   For any other pair $(i, j) \neq (k, \ell')$, the quantities remain unchanged, *i.e.* $W_{i,j}(t + 1) = W_{i,j}(t)$ and $d_{i,j}(t + 1) = d_{i,j}(t)$. Finally,

$$\lambda_{k,j}(t+1) = \frac{d_{k,j}(t+1)}{d_{k,1}(t+1) + d_{k,2}(t+1)}\ , \quad j = 1, 2$$

   and $\lambda_{i,j}(t + 1)$ remains unchanged for $i \neq k$, $j = 1, 2$.
4. Now, let ℓ be such that $X_{t+1} \in \mathcal{S}_\ell(t)$; then

$$|W_\ell(t+1)| = |W_\ell(t)| + 1 \quad , \quad |W_i(t+1)| = |W_i(t)|\ ,\ \ i \neq \ell\ ,$$

   with $W_i(t)$ as in (4), and

$$\overline{x}_\ell(t+1) = \frac{1}{|W_\ell(t+1)|} \left(|W_\ell(t)| \overline{x}_\ell(t) + X_{t+1}\right)\ ,$$
$$\overline{x}_i(t+1) = \overline{x}_i(t)\ ,\ \ i \neq \ell\ .$$

   Similarly, $\Sigma_\ell(t + 1)$ and $\Sigma_S(t + 1)$ are updated according to the AM sampler, while $\Sigma_i(t + 1) = \Sigma_i(t)$ $(i \neq \ell)$. We finally update the hyperplane equation as

$$a_{t+1} = \overline{x}_1(t+1) - \overline{x}_2(t+1) \quad , \quad b_{t+1} = a_{t+1}^T r(t+1)\ ,$$

where $r(t+1)$ follows from (7) and (6); X_{t+1} then belongs to the new region

$$X_{t+1} \in \begin{cases} \mathcal{S}_1(t+1) & \text{if } a_{t+1}^T X_{t+1} \geq b_{t+1} \,, \\ \mathcal{S}_2(t+1) & \text{if } a_{t+1}^T X_{t+1} < b_{t+1} \,. \end{cases}$$

Remark 1. *In a given iteration, we do not reevaluate the region of belonging of every sample point, but merely that of the most recent observation. If we did, some values close to the frontier would likely move from one region to another, due to the adaptive nature of the partitioning. Such extra computations could be interesting during the first few iterations, when the partition is still very unstable. A subprocess that alternately evaluates regions and computes a new partition could even be used, which would be equivalent to the K-means algorithm. Once the partition stabilizes, it naturally becomes inefficient to reevaluate the position of all sample values with respect to new partitions. Preliminary simulation studies however suggest that these extra computations do not improve the performance of the sampler, so this avenue has not been explored further.*

Remark 2. *The mixture form of the proposal distribution remains useful in this new version of the sampler. Experimental results show that the adaptive mixture weights greatly contribute to the rapid convergence towards a good partition.*

4.2. Generalization to $K > 2$

The adaptive partitioning process of Section 4 is described in the context of two separate regions. The approach may be generalized to more regions by obtaining a hyperplane for each pair of sample averages, and then considering appropriate intersections arising from these hyperplanes. In the general case, one has to compute $\binom{K}{2}$ hyperplanes in order to define K regions.

Specifically, let $a_{i,j}(t)$, $b_{i,j}(t)$ represent the coefficients of the hyperplanes dividing $\overline{x}_i(t)$ and $\overline{x}_j(t)$, for $1 \leq i < j \leq K$; these terms are obtained from calculations analogous to (7). An observation x then belongs to the region $\mathcal{S}_j(t)$ such that $a_{j',j}^T(t)x < b_{j',j}(t)\ \forall j' < j$ and $a_{j,j'}^T(t)x \geq b_{j,j'}(t)\ \forall j' > j$. To classify a new observation according to this rule given $\{(a_{i,j}(t), b_{i,j}(t)); 1 \leq i < j \leq K\}$, we conveniently need exactly $K-1$ scalar products, as each carefully selected comparison eliminates one potential region. Also note that only $K-1$ of the $\binom{K}{2}$ hyperplanes need to be updated in a given iteration, since only one sample average and covariance are updated at a time.

If hyperplanes go through the middle points of the segments joining sample averages, then this becomes equivalent to classifying a point in the region containing the closest sample average. When hyperplanes do not go through middle points, we cannot classify an observation by simply computing its K Mahalanobis distances to each sample average. Since hyperplanes are perpendicular to the lines joining pairs of sample averages, then observations that are not located on this line (*i.e.* most observations) are not necessarily located in the region containing the Mahalanobis-closest average. In fact, a frontier dividing two regions so as to ensure that both Mahalanobis distances (from each sample mean to the frontier) are equal generally is a hypersurface more complex than a hyperplane.

Although generalizations to more than 2 regions are easily implemented, K should be kept at a minimum. To efficiently explore $\mathcal{S}$, every region should be visited sufficiently often, which becomes increasingly complicated as K grows.

4.3. Computational complexity

To have a better understanding of how OPRA and RAPTOR compare in terms of computational efficiency, we briefly analyse the complexity of their partitioning process per iteration. To assign

a new point to a region, RAPTOR requires evaluating K normal densities, each with its own $d \times d$ determinant and quadratic form. Since all K densities are updated at every iteration, then K Cholesky decompositions must be computed every time. RAPTOR thus takes $O(Kd^3)$ flops per iteration to assign a new iterate to its region.

OPRA evaluates $O(K)$ $d \times d$ quadratic forms to update the $b_{i,j}(t)$ coefficients, but these forms only involve one new covariance matrix. Assuming that Cholesky decompositions are saved at each step, then only one new decomposition per iteration is required. The sampler then performs $K-1$ well-chosen scalar comparisons to assign a new point to its region. Overall, OPRA thus takes $O(d^3 + Kd^2)$ flops per iteration to classify a new point. Its computational advantage stems from the number of matrix decompositions required (1 for OPRA versus K for RAPTOR).

Different partitioning schemes could also lead to potentially interesting results (as long as they remain strictly less complex than 2 Cholesky decompositions). One could, for instance, update two covariance matrices per iteration as in OPRA (regional and global matrices) but use the partitioning scheme of RAPTOR, for a cost of $O(d^3 + Kd^2)$ per iteration. Alternatively, regions could be divided according to the Mahalanobis distances to each sample mean, $\mathcal{S}_i(t) = \{x : \arg\min_{i'}(x - \overline{x}_{i'}(t))^T \Sigma_{i'}^{-1}(t)(x - \overline{x}_{i'}(t)) = i\}, i = 1, \ldots, K$. Although the computational efficiency of the partitioning process is directly related to the number of matrix decompositions, users should be cautious in defining regions. In our numerical experiments, RAPTOR faced problems with some targets and it is unclear whether this was due to the way of adapting parameters or of defining regions. Partitioning schemes based on Mahalanobis distances should however yield mixing times that are similar to OPRA.

5. ERGODICITY

To prove the ergodicity of OPRA, we make use of the sufficient ergodicity conditions introduced in Theorem 1 of Roberts and Rosenthal [2007]. They state that an adaptive sampler on a state space $\mathcal{S}$ with adaptive transition space $\mathcal{Y}$, for which every possible transition kernel P_γ, $\gamma \in \mathcal{Y}$ has $\pi(\cdot)$ as its stationary distribution, is ergodic for $\pi(\cdot)$ under the following conditions:

1. *Simultaneous Uniform Ergodicity*: For all $\epsilon > 0$, there exists $N \in \mathbb{N}$ such that

$$\|P_\gamma^{(N)}(x, \cdot) - \pi(\cdot)\|_{TV} < \epsilon, \forall x \in \mathcal{S}, \forall \gamma \in \mathcal{Y}, \tag{8}$$

   where $\|\cdot\|_{TV}$ denotes the norm in total variation distance.
2. *Diminishing Adaptation*:

$$\sup_x \|P_{\Gamma_t}(x, \cdot) - P_{\Gamma_{t-1}}(x, \cdot)\|_{TV} \xrightarrow[t \to \infty]{} 0 \text{ , in probability,} \tag{9}$$

   where Γ_t contains the actual (random) adaptive parameters in use at time t.

We impose some conditions on the state space and target density. Specifically, we assume that there is a compact subset $\mathcal{S} \subseteq \mathbb{R}^d$ such that the target density π is continuous on $\mathcal{S}$, positive on the interior of $\mathcal{S}$, and zero outside of $\mathcal{S}$. These assumptions are of technical interest and do not introduce important constraints in practice, as compact sets can be arbitrarily large.

To update hyperplanes, we also require a minimum distance $\delta > 0$ between all pairs of sample means. We thus start by assuming that $\|a_{i,j}(0)\| = \|\overline{x}_i(0) - \overline{x}_j(0)\| \geq \delta$ for all pairs $i \neq j$ $(i, j = 1, \ldots, K)$. Then, if $\|\overline{x}_i(t+1) - \overline{x}_j(t+1)\| < \delta$ at some $t > 0$ for some pair $i \neq j$ $(i, j = 1, \ldots, K)$, the current partition is maintained by letting $a_{t+1} = a_t$ and $b_{t+1} = b_t$. This means that we update the usual quantities of Section 4.1 to the exception of a_t and b_t, which

remain constant until $\|\overline{x}_i(t+1) - \overline{x}_j(t+1)\| \geq \delta$ for $i, j = 1, \ldots, K$, $i \neq j$. Choosing δ appropriately small will have a negligible effect on the sampler in practice.

Theorem 1. *Consider a compact sample space $\mathcal{S}$ and a continuous, strictly positive target density $\pi(\cdot)$ on $\mathcal{S}$. The algorithm OPRA described in Section 4 and equipped with the above δ-requirement satisfies conditions (8) and (9), which are sufficient for guaranteeing ergodicity of adaptive algorithms.*

Proof. The proof of (8) is identical to that of RAPT in Craiu et al. [2009] and is thus omitted. The new adaptive parameters for partitioning $\mathcal{S}$ do not jeopardize the existence of lower and upper bounds on the proposal and target densities or their positivity, which are the sole necessary elements for proving this condition.

We thus focus on verifying (9). This portion of the proof is also similar to what is done in Craiu et al. [2009], with the extra complication that integration regions now vary with t. We focus on the case $K = 2$ and then provide indications for the generalization of this proof to any finite $K > 2$.

Let the time-t proposal density be expressed as

$$f_{\Gamma_t}(y|x) = (1-\beta) \sum_{i=1}^{K} \mathbb{1}_{\mathcal{S}_i(t)}(x) \sum_{j=1}^{K} \lambda_{i,j}(t) q_j(y, t|x) + \beta \, q_S(y, t|x) \, ,$$

where $q_j(y, t|x)$ is the normal density with mean x and covariance $\Sigma_j(t)$ for $j \in \{1, \ldots, K, S\}$. The vector $\Gamma_t = (\lambda_{1,1}(t), \ldots, \lambda_{K,K}(t), \Sigma_1(t), \ldots, \Sigma_K(t), \Sigma_S(t))$ represents the set adaptive parameters at a given time t. Let $\mathbb{M}(c_1, c_2)$ be the set of all $k \times k$ positive definite matrices M such that $c_1 I_k \leq M \leq c_2 I_k$. From (14) in the proof of Theorem 1 in Haario et al. [2001], there exist $c_1, c_2 > 0$ such that all $\Sigma_i(t)$ $(i = 1, \ldots, K)$ and $\Sigma_S(t)$ are in $\mathbb{M}(c_1, c_2)$. Since $\mathcal{S}$ is compact, then

$$M_1 \equiv \max \left\{ \sup_{x,y \in \mathcal{S}} q_1(y, t|x), \ldots, \sup_{x,y \in \mathcal{S}} q_K(y, t|x), \sup_{x,y \in \mathcal{S}} q_S(y, t|x) \right\} < \infty \, .$$

Define $g_{\Gamma_t}(x, y) := f_{\Gamma_t}(y|x)\alpha_t^*(x, y)$, with f as above and α_t^* as in (3). For any $x \in \mathcal{S}_1(t)$ and $A \in \mathcal{B}(\mathcal{S})$ (the Borel sets of $\mathcal{S}$), the time-t probability of moving from x to a state in A is

$$\begin{aligned} P_{\Gamma_t}(x, A) = & \sum_{i=1}^{K} \int_{A \cap \mathcal{S}_i(t)} f_{\Gamma_t}(y|x) \alpha_t^*(x, y) \mathrm{d}y \\ & + \mathbb{1}_A(x) \sum_{i=1}^{K} \int_{\mathcal{S}_i(t)} f_{\Gamma_t}(y|x)(1 - \alpha_t^*(x, y)) \mathrm{d}y \, . \end{aligned} \tag{10}$$

Denote each term in (10) by $I_{i,t}(x, A)$, $i = 1, \ldots, 2K$. Equation (9) satisfies

$$\sup_x \|P_{\Gamma_{t+1}}(x, \cdot) - P_{\Gamma_t}(x, \cdot)\|_{TV} \leq \sup_x \sup_A \sum_{i=1}^{2K} |I_{i,t+1}(x, A) - I_{i,t}(x, A)| \, . \tag{11}$$

Hereafter, we let $K = 2$ and focus on the convergence of $|I_{2,t+1}(x, A) - I_{2,t}(x, A)|$ (the proof for the other terms is very similar). We have

$$|I_{2,t+1}(x, A) - I_{2,t}(x, A)| = \left| \int_{A \cap \mathcal{S}_2(t+1)} g_{\Gamma_{t+1}}(x, y) \mathrm{d}y - \int_{A \cap \mathcal{S}_2(t)} g_{\Gamma_t}(x, y) \mathrm{d}y \right| ;$$

splitting both integration regions and using the fact that $g_{\Gamma_t}(x, y)$ is uniformly bounded by M_1 leads to

$$\begin{aligned}
&\sup_x \sup_A |I_{2,t+1}(x, A) - I_{2,t}(x, A)| \\
&\leq \sup_x \sup_A \int_{A\cap\mathcal{S}_2(t)\cap\mathcal{S}_2(t+1)} \left|g_{\Gamma_{t+1}}(x, y) - g_{\Gamma_t}(x, y)\right| \mathrm{d}y \\
&\quad + M_1 \int_{\mathcal{S}_1(t)\cap\mathcal{S}_2(t+1)} \mathrm{d}y + M_1 \int_{\mathcal{S}_1(t+1)\cap\mathcal{S}_2(t)} \mathrm{d}y \,. \qquad (12)
\end{aligned}$$

We now show that each term on the right converges to 0 as $t \to \infty$. The convergence to 0 of the first term is proved in Lemma 4.2 of Craiu et al. [2009]. For the second and third terms, it suffices to show that the subset of $\mathcal{S}$ switching regions from t to $t + 1$ has a measure converging to 0. We suppose that $d \geq 2$, the case $d = 1$ being trivial.

At any time t, for two successive hyperplanes to be exactly parallel, X_{t+1} has to be generated directly on the segment joining $\overline{x}_1(t)$ and $\overline{x}_2(t)$, which happens with probability 0. Successive hyperplanes thus intersect, and the intersection is a $(d-2)$-dimensional hyperplane. The acute angle between the normal vectors a_t and a_{t+1} of these hyperplanes is $\theta_t := \arccos\left(a_t^T/\|a_t\| \cdot a_{t+1}/\|a_{t+1}\|\right)$, with a_t as in (7). Letting $i(t)$ represent the region of x_{t+1},

$$a_{t+1} = \overline{x}_1(t) - \overline{x}_2(t) + \frac{(-1)^{i(t)-1}(x_{t+1} - \overline{x}_{i(t)}(t))}{|W_{i(t)}(t+1)|} =: a_t + c_t \,. \qquad (13)$$

Since $\mathcal{S}$ is compact, then every component in the numerator of c_t is bounded. Furthermore, $|W_{i(t)}(t+1)| \to \infty$ when $t \to \infty$; indeed, even if one of the regions (say $\mathcal{S}_1$) is only visited a finite number of times, then for t_0 large enough, $i(t) = 2\ \forall t \geq t_0$, implying that $|W_2(t+1)| \to \infty$. Hence, $c_t \to 0$ term by term. Using (13) along with the triangle inequality at the denominator, we find

$$1 \geq \frac{a_t^T a_{t+1}}{\|a_t\| \cdot \|a_{t+1}\|} \geq \frac{\|a_t\|^2 + a_t^T c_t}{\|a_t\|^2 + \|a_t\| \cdot \|c_t\|} \xrightarrow[t\to\infty]{} 1 \,. \qquad (14)$$

We note that the compacity of $\mathcal{S}$ and the design of the sampler, specifically the δ-requirement, imply that there exist δ and M_0 such that $0 < \delta \leq \|a_t\| \leq M_0 < \infty$ for all t. Continuity of the function $\arccos$ implies that $\theta_t \to 0$ as $t \to \infty$.

Since the angle between successive hyperplanes goes to 0, then the volume between those hyperplanes inside $\mathcal{S}$ also goes to 0 as $t \to \infty$, as long as the intersection between hyperplanes do not get arbitrarily far from $\mathcal{S}$. In fact, the point z^* that is simultaneously contained in two successive hyperplanes and closest to origin can be obtained by minimizing $h(z) = \|z\|^2 + 2\lambda_1\left(a_t^T z + b_t\right) + 2\lambda_2\left(a_{t+1}^T z + b_{t+1}\right)$, using Lagrange multipliers. This leads to

$$z^* = \frac{\left(\|a_{t+1}\|^2 b_t - a_t^T a_{t+1} b_{t+1}\right) a_t + \left(\|a_t\|^2 b_{t+1} - a_t^T a_{t+1} b_t\right) a_{t+1}}{\|a_t\|^2\|a_{t+1}\|^2 - \left(a_t^T a_{t+1}\right)^2} \,.$$

Using a decomposition similar to (14), we see that the denominator is bounded below by a positive constant. Hyperplane parameters are also bounded as they are functions of sample averages on a compact $\mathcal{S}$. Each term of z^* is thus bounded (so is $\|z^*\|$), implying that the distance between the intersection of two successive hyperplanes and $\mathcal{S}$ is bounded, as desired. The same

holds for the Mahalanobis version described in Section 4. The only difference lies in the parameter b_t, but it remains bounded as $b_t = a_t \{(1 - k(t))\overline{x}_1(t) + k(t)\overline{x}_2(t)\}$, with $k(t) \in (0, 1)$. We conclude that the second and third terms on the right of (12) converge to 0 as $t \to \infty$, and thus $\sup_x \sup_A |I_{2,t+1}(x, A) - I_{2,t}(x, A)| \to 0$ as $t \to \infty$. Since this reasoning may be repeated for every term in (11), this verifies (9).

To generalize the proof to the case $K > 2$, we use the following upper bound for $|I_{2,t+1}(x, A) - I_{2,t}(x, A)|$

$$\int_{A\cap\mathcal{S}_2(t)\cap\mathcal{S}_2(t+1)} \left|g_{\Gamma_{t+1}}(x, y) - g_{\Gamma_t}(x, y)\right| \mathrm{d}y$$
$$+ \sum_{i\neq 2, 1\leq i\leq K} \left[\int_{\mathcal{S}_i^*(t)\cap\mathcal{S}_2^*(t+1)} |g_{\Gamma_{t+1}}(x, y)|\mathrm{d}y + \int_{\mathcal{S}_i^*(t+1)\cap\mathcal{S}_2^*(t)} |g_{\Gamma_t}(x, y)|\mathrm{d}y\right] ,$$

where $\mathcal{S}_2^*(\cdot)$ and $\mathcal{S}_i^*(\cdot)$ denote the partition assuming that these are the only two regions of $\mathcal{S}$. We may then show that each term on the second line converges to 0 as $t \to \infty$, using arguments similar to those used above when $K = 2$. Indeed, each region of integration only depends on the evolution of one hyperplane. For instance, the area of the region $\mathcal{S}_i^*(t + 1) \cap \mathcal{S}_2^*(t)$ only depends on the evolution of the hyperplane separating $\overline{x}_i$ and $\overline{x}_2$ from t to $t + 1$, and so on. ■

6. SIMULATED EXAMPLES

We now study the behaviour of OPRA and compare it to competitors through simulation studies. Consider the following flexible family of target densities

$$\pi(x) = \sum_{k=1}^{\kappa} p_k \, \mathcal{N}(\phi_\psi(x); \mu_k, \Sigma_k) , \quad x \in \mathbb{R}^d .$$

For $k = 1, \ldots, \kappa$, $\mu_k \in \mathbb{R}^d$, $\Sigma_k \in \mathbb{R}^{d\times d}$ are positive definite matrices, $p_k \in [0, 1]$ with $\sum p_k = 1$, and $\psi \geq 0$. Here, $\mathcal{N}(x; \mu, \Sigma)$ is the normal density function with mean μ and covariance Σ; the function $\phi_\psi(\cdot)$ satisfies $\phi_\psi(x_2) = x_2 + \psi(x_1^2 - 100)$ and $\phi_\psi(x_j) = x_j$ for $j \in \{1, 3 \ldots, d\}$. The parameter ψ introduces a nonlinearity in the exponential term, leading to increasingly twisted distributions and non convex confidence regions as ψ gets larger. This type of distribution has been used in Haario et al. [2001], for instance, in the context of comparative tests.

6.1. Illustrations of the adaptive partitioning

From its construction, it seems clear that the adaptive partitioning of OPRA will be effective in the case of multimodal target distributions, with modes separated by a low density region. It is however interesting to investigate OPRA's behavior in more challenging situations, as may arise from the above family of target densities. We focus here on two-dimensional examples, as the convergence of the separating hyperplane can then be visually illustrated.

In all cases, $K = 2$ and the basic OPRA$_0$ sampler is used (hyperplane going through the middle point). The number of iterations before starting adaptation is $t_0 = 100$ and the weight of the global component is $\beta = 0.3$ in all cases. Two parallel chains started in each of the initial regions are implemented with $\Sigma_1(0) = \Sigma_2(0) = I_2$ and $\Sigma_S(0) = 25I_2$. The graphs in Figures 1 to 3 each present an i.i.d. sample of size 10^4 from the target distribution. The initial hyperplane and starting values are orange, while the final hyperplane and sample means $\overline{x}_1(n), \overline{x}_2(n)$ are green. The blue ellipses correspond to 95% confidence region estimates for $\mathbb{E}\left[X\mathbb{1}_{\mathcal{S}_i(n)}(X)\right]$ $(i = 1, 2)$ and $\mathbb{E}[X]$ (one for each region, as well as a global one), obtained from sample means

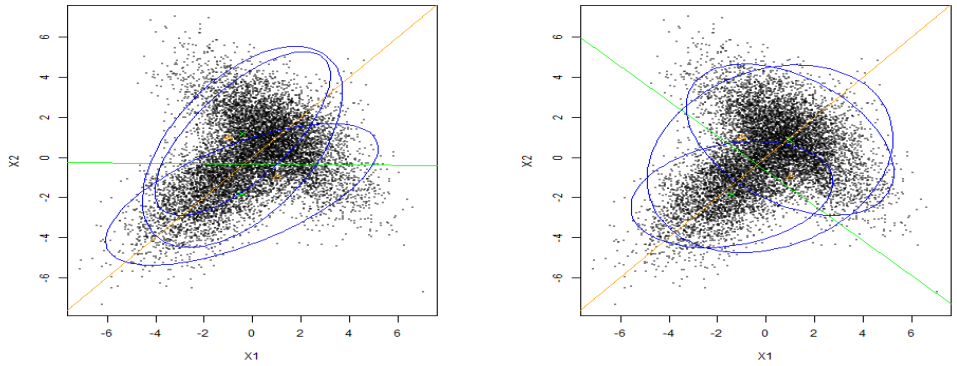

FIGURE 1: Non spherical covariances, $n = 200$ (left), $n = 2,000$ (right). Target parameters: $\mu_1 = (-1,-1)$, $\mu_2 = (1,1)$, $\Sigma_1 = (3,2;2,3)$, $\Sigma_2 = (3,-2;-2,3)$, $p_1 = 0.5$.

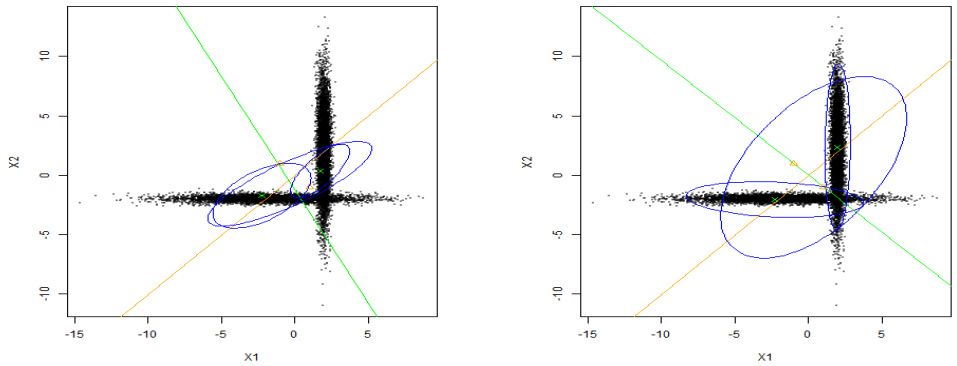

FIGURE 2: Stretched covariances, $n = 200$ (left), $n = 2,000$ (right). Target parameters: $\mu_1 = (-2,-2)$, $\mu_2 = (2,2)$, $\Sigma_1 = diag(10, 0.05)$, $\Sigma_2 = diag(0.05, 10)$, $p_1 = 0.4$.

and empirical covariances. In each figure the left graph illustrates the evolution after a limited number of iterations ($n = 200$), while the right one considers $n = 2,000$.

In Figures 1 to 3, the hyperplane adaptation is fast, even when the initial partition is way off. Naturally, the online partitioning gradually becomes more challenging as the dimension of the target density increases.

6.2. Comparative tests

We now evaluate the performance of RAPT, RAPTOR, OPRA$_0$, and OPRA in sampling from target densities with $d = 50$. Densities are selected so as to cover a variety of challenges, but are simple enough so that specific efficiency and convergence measures may be recorded. In all cases, N runs of n iterations, each with a pre-adaptation period of t_0 are obtained. In each example, $k = 4$ adaptive parallel chains with inter-chain adaptation (as described in Craiu et al. [2009]) are implemented to remain true to what would usually be done in practice; this improves the performance of all four samplers, but particularly that of RAPT. We use $K = 2$ regions and initial conditions that reproduce as closely as possible approaches favored in the literature: small-normed $\Sigma_1(0)$, $\Sigma_2(0)$ that initially lead to high acceptance rates within each region during pre-adaptation and an overly spread out $\Sigma_S(0)$ covering the whole state space.

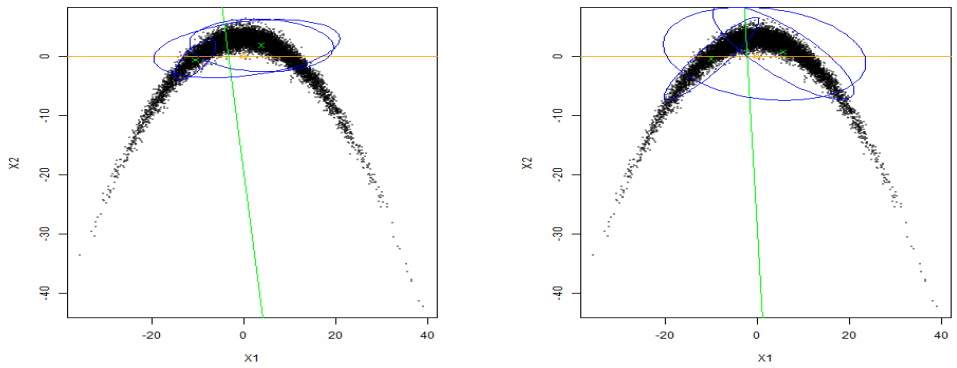

FIGURE 3: Irregular distribution, $n = 200$ (left), $n = 2,000$ (right). Target parameters: $\mu_1 = (0, 0)$, $\Sigma_1 = diag(100, 1)$, $\psi = 0.03$.

To compare the performances of the various samplers we measure ρ, the absolute difference between theoretical and empirical coverage rates at various levels. If $C_{1-\alpha}$ is the $(1-\alpha)\%$-level highest density region associated to $\pi(x)$, then $\rho = |(1-\alpha) - \sum_{i=0}^{n} \mathbb{1}_{C_{1-\alpha}}(x_i)/(n+1)|$. We report $\overline{\rho}$, the average difference over N runs, as well as the associated standard deviation. Quality algorithms should produce small $\overline{\rho}$, along with a small standard deviation. Evaluating average coverage rates at various levels (50%, 90%, 95%, 99%) is ideal in the examples considered, as they measure the extent to which a sampler is able to estimate the tails of a complicated target density. In cases where theoretical regions $C_{1-\alpha}$ cannot be obtained analytically, we compute estimates based on a sample of size 10^6 obtained directly from the target. We also report running times and the acceptance rate $T_a = \sum_{t=1}^{n} \mathbb{1}\{x_t \neq x_{t-1}\}/n$ whenever appropriate.

Below, $X_0^{(i)}$, $i = 1, \ldots, 4$, denote the starting value of each parallel chain. The initial covariance matrices are specified for each example, as well as the initial partition of $\mathcal{S}$ used by RAPT, OPRA$_0$ and OPRA, and the initial means $\mu_1(0)$ and $\mu_2(0)$ for RAPTOR. All other initial parameters are set to their default values as specified in Section 4.1, and in Bai et al. [2011] in the case of RAPTOR. We first discuss a few unimodal examples before turning to a bimodal context.

The first example, a simple unimodal, spherical normal with an appropriate initial hyperplane, leads to interesting observations. In Table 1, RAPTOR and OPRA offer comparable performances that are better than those of RAPT and OPRA$_0$ (themselves similar). Compared to its competitors, RAPTOR however takes 33% more time to complete its task, which makes OPRA's net performance more appealing. In fact OPRA's extra flexibility pushes the separating hyperplane to the side of the density, leading to a proposal distribution that is unimodal, and thus better suited to the target at hand. RAPTOR shows a similar behaviour in relatively high dimensions (one of the proposal modes gets a weight close to 1). Numerical explorations however show that this sampler suffers from significant problems in smaller dimensions (when $d = 5$, the proposal distribution remains bimodal). The designs of RAPT and OPRA$_0$ do not allow detecting unimodality, explaining their somehow inferior performances.

Similar conclusions hold when the target is a two-term mixture of distributions whose modes coincide, *e.g.* $0.6\mathcal{N}(\mathbf{0}_d, I_d) + 0.4\mathcal{N}(\mathbf{0}_d, 4I_d)$. As before, RAPTOR and OPRA often treat such distributions as unimodal. Our numerical experiments show that OPRA does not suffer from working with a single region, offering net performances that are at least as good, and often better than those of RAPT (initialized with a decent partition) and OPRA$_0$. RAPTOR faces some difficulties: in moderate dimension ($d = 20$) net performances are, at best, comparable to those of its competitors; in smaller and larger dimensions ($d = 5, 50$), gross performances remain inferior

Algorithm	RAPT		RAPTOR		OPRA_0		OPRA	
Criteria	mean	s.d.	mean	s.d.	mean	s.d.	mean	s.d.
	$K = 2$	$d = 50$	$n = 2 \cdot 10^5$	$t_0 = 10^4$	$N = 10^3$	$k = 4$		
Time (sec)	32.92	2.80e-02	44.72	6.02e-02	33.80	1.31e-01	33.42	1.23e-02
Cov. 50% (%)	6.51	3.70e-02	5.51	6.67e-02	6.50	3.80e-02	5.53	3.90e-02
Cov. 90% (%)	2.47	1.80e-02	2.03	2.47e-02	2.47	1.77e-02	2.18	1.86e-02
Cov. 95% (%)	1.39	1.17e-02	1.14	1.54e-02	1.40	1.15e-02	1.24	1.20e-02
Cov. 99% (%)	0.33	4.06e-03	0.28	5.05e-03	0.34	4.17e-03	0.31	4.34e-03
T_a	0.25	5.65e-05	0.28	7.72e-04	0.25	5.75e-05	0.25	9.65e-05

TABLE 1: Unimodal, spherical normal target: $\mu_1 = (0, \dots, 0), \Sigma_1 = I_d$. Parameters: $\mu_1(0) = X_0^{(1)} = X_0^{(2)} = (-0.1, 0, \dots, 0), \mu_2(0) = X_0^{(3)} = X_0^{(4)} = (0.1, 0, \dots, 0), \Sigma_1(0) = \Sigma_2(0) = 0.1 I_d$, $\Sigma_S(0) = 2I_d$. Initial hyperplane: $x_1 = 0$.

to its competitors, a phenomenon that is amplified when taking account of computational effort.

As a second example we look at a unimodal, banana-shaped target density similar to that illustrated in Figure 3. According to Table 2, the suboptimal initial partition favors OPRA_0 compared to RAPT, which cannot explore $\mathcal{S}$ as efficiently due to its static partition. RAPTOR yields coverage rates that are comparable to OPRA_0, while OPRA performs significantly better than its competitors for a fixed number of iterations. The gap between RAPTOR and the other samplers widens when taking account of the increased computational effort: almost 30% compared to RAPT, as opposed to an increase of only 2% for OPRA_0 and OPRA. Interestingly, modifying the twisting degree of the target does not appear to affect RAPTOR's performance, contrarily to that of the other samplers. Nonetheless, in our simulation studies, the net performances of OPRA_0 and OPRA remain the most appealing options after adjusting for running time.

Algorithm	RAPT		RAPTOR		OPRA_0		OPRA	
Criteria	mean	s.d.	mean	s.d.	mean	s.d.	mean	s.d.
	$K = 2$	$d = 50$	$n = 2 \cdot 10^5$	$t_0 = 10^4$	$N = 10^3$	$k = 4$		
Time (sec)	44.28	2.73e-02	56.88	1.04e-01	44.46	3.16e-02	45.07	2.75e-02
Cov. 50% (%)	9.78	4.49e-02	8.82	1.09e-01	9.04	4.12e-02	8.58	4.22e-02
Cov. 90% (%)	3.55	1.84e-02	3.28	3.89e-02	3.30	1.82e-02	3.14	1.87e-02
Cov. 95% (%)	1.98	1.17e-02	1.86	2.23e-02	1.86	1.15e-02	1.77	1.19e-02
Cov. 99% (%)	0.47	4.15e-03	0.45	6.16e-03	0.45	4.08e-03	0.42	4.28e-03
T_a	0.23	2.83e-04	0.21	1.41e-03	0.23	1.82e-04	0.22	1.91e-04

TABLE 2: Unimodal, stretched normal target with slight twisting and poor initial partition: $\psi = 0.03$, $\mu_1 = (0, \dots, 0), \Sigma_1 = \text{diag}(100, 1, \dots, 1)$. Parameters: $X_0^{(1)} = \dots = X_0^{(4)} = (0, \dots, 0)$, $\mu_1(0) = -\mu_2(0) = (-0.1, 0, \dots, 0), \Sigma_1(0) = \Sigma_2(0) = 0.1 I_d, \Sigma_S(0) = I_d$. Initial hyperplane: $x_2 = 0$.

We now discuss the bimodal context. In very simple cases where modes are distinct without being too far from each other and of similar shape/scale, RAPTOR is generally the best available option (even when adjusting for running time). If initial parameters are reasonable, without necessarily being ideal, then RAPT offers performances similar to OPRA_0 and OPRA, due to the flexibility of its other adaptive parameters. Although OPRA_0 and OPRA do not lead to significant efficiency gains here, they do not cost much to implement either. As soon as we depart from these generic cases however, we find different conclusions.

We consider a third example in which two modes are blended into a single, asymmetrical density mass. For a fixed number of iterations, RAPTOR loses ground to OPRA and OPRA$_0$ (Table 3). It seems that the pre-adaptation period is not long enough for the sampler to appropriately detect and explore the second mode, leading to a discarded component and a unimodal proposal distribution. A longer pre-adaptation period improves all performances but that of RAPTOR. Our numerical explorations point towards RAPT, OPRA$_0$, and OPRA being more robust than RAPTOR to initial parameters and length of pre-adaptation.

Algorithm	RAPT		RAPTOR		OPRA$_0$		OPRA	
Criteria	mean	s.d.	mean	s.d.	mean	s.d.	mean	s.d.
	$K=2$	$d=50$	$n=2\cdot10^5$	$t_0=10^4$	$N=10^3$	$k=4$		
Time (sec)	52.41	3.87e-01	62.51	4.41e-01	53.93	4.12e-01	53.80	3.99e-01
Cov. 50% (%)	40.80	2.39e-01	46.68	1.27e-01	39.45	2.73e-01	39.14	2.75e-01
Cov. 90% (%)	8.96	3.15e-02	9.88	7.07e-03	8.74	3.71e-02	8.69	3.86e-02
Cov. 95% (%)	4.55	1.44e-02	4.96	2.82e-03	4.45	1.70e-02	4.42	1.79e-02
Cov. 99% (%)	0.93	2.65e-03	1.00	5.31e-04	0.91	3.14e-03	0.91	3.45e-03
T_a	0.26	4.30e-04	0.17	1.44e-03	0.27	4.84e-04	0.26	5.28e-04

TABLE 3: Normal bimodal target with unequal variances: $\mu_1=-\mu_2=-(2.5,\ldots,2.5)/d$, $\Sigma_1=4I_d$, $\Sigma_2=I_d$, $p_1=0.6$. Parameters: $\mu_1(0)=X_0^{(1)}=X_0^{(2)}=(-2,0,\ldots,0)$, $\mu_2(0)=X_0^{(3)}=X_0^{(4)}=(2,0,\ldots,0)$, $\Sigma_1(0)=\Sigma_2(0)=0.1I_d$, $\Sigma_S(0)=100I_d/d$. Initial hyperplane : $x_1=-1$.

The last example studies a 20-dim bimodal distribution in which one mode is narrow and the other widely spread out. Table 4 is one of the few examples where pre-adaptation time has a drastic impact on the performances. With $t_0=10^3$, OPRA does better than its competitors in recovering from a pre-adaptation period that is arguably too short. RAPTOR comes second, but is further penalized by its running time compared to RAPT and OPRA$_0$. With $t_0=10^4$, RAPT, OPRA$_0$, and OPRA offer similar performances. Thanks to the quality of its initial partition, RAPT is able to recover when the pre-adaptation is sufficiently long, while efficiency gains for RAPTOR are less important as for other samplers.

6.3. Comparative tests with three or more regions

Table 5 presents results for the twisted target of Table 2, with $K=3$. For all samplers but RAPTOR, the computational overhead introduced by the third region is negligible. The additional region improves the performances of all four samplers, with RAPTOR offering the best results. When accounting for running time, RAPTOR is however relegated behind OPRA given that the other samplers run in about 60% of the time.

In this example, the initial partition for RAPT, OPRA$_0$, and OPRA is extremely poor. OPRA thus produces a slightly larger $\bar{\rho}$ than RAPTOR as it has to recover from this poor initial setting. Surprisingly, RAPT and OPRA offer seemingly identical performances. This provides a false sense of security in using RAPT, but users should be aware that despite an appealing $\bar{\rho}$-value, every run consistently under-samples one tail of the distribution due to the poor initial partition. A similar behaviour happens with OPRA$_0$ (see the average mean square errors in Table 5). For these algorithms, add-ons such as INCA then become mandatory to improve the quality of samples (already implemented here).

In Table 6, a normal target with 5 spherical modes, which is an ideal density for RAPTOR, is implemented with $K=5$ regions. In that context, OPRA and OPRA$_0$ offer similar performances, and turn out to be much better than RAPTOR when adjusting for computational effort. The initial

Algorithm	RAPT		RAPTOR		OPRA_0		OPRA	
Criteria	mean	s.d.	mean	s.d.	mean	s.d.	mean	s.d.
	$K=2$	$d=20$	$n=10^5$	$t_0=10^3$	$N=10^4$	$k=4$		
Time (sec)	4.20	9.36e-03	5.49	1.18e-02	4.31	9.97e-03	4.43	1.03e-02
Cov. 50% (%)	18.04	1.41e-01	17.64	1.20e-01	18.33	1.42e-01	16.72	1.50e-01
Cov. 90% (%)	6.18	3.70e-02	6.02	3.24e-02	6.26	3.76e-02	5.76	4.01e-02
Cov. 95% (%)	3.20	1.80e-02	3.11	1.59e-02	3.24	1.83e-02	2.99	1.96e-02
Cov. 99% (%)	0.67	3.60e-03	0.65	3.21e-03	0.68	3.64e-03	0.63	3.92e-03
T_a	0.22	3.81e-04	0.23	4.11e-04	0.21	4.01e-04	0.20	3.97e-04
	$K=2$	$d=20$	$n=10^5$	$t_0=10^4$	$N=10^4$	$k=4$		
Time (sec)	3.38	2.87e-03	4.54	3.42e-03	3.40	2.81e-03	3.47	2.93e-03
Cov. 50% (%)	5.34	1.61e-01	8.24	1.55e-01	5.80	1.61e-01	5.61	1.62e-01
Cov. 90% (%)	2.48	4.64e-02	3.35	4.42e-02	2.57	4.65e-02	2.48	4.69e-02
Cov. 95% (%)	1.38	2.30e-02	1.78	2.20e-02	1.42	2.31e-02	1.37	2.33e-02
Cov. 99% (%)	0.31	4.75e-03	0.38	4.57e-03	0.31	4.77e-03	0.30	4.82e-03
T_a	0.19	3.95e-04	0.22	4.55e-04	0.19	3.95e-04	0.18	4.07e-04

TABLE 4: Bimodal normal target with narrow and wide modes: $\mu_1 = -(2.5,\ldots,2.5)/d$, $\mu_2 = (2.5,\ldots,2.5)/d$, $\Sigma_1 = I_d/10$, $\Sigma_2 = I_d$, $p = 0.6$. Parameters: $\mu_1(0) = X_0^{(1)} = X_0^{(2)} = (-2,0,\ldots,0)$, $\mu_2(0) = X_0^{(3)} = X_0^{(4)} = (2,0,\ldots,0)$, $\Sigma_1(0) = \Sigma_2(0) = 0.1I_d$, $\Sigma_S(0) = 100I_d/d$. Initial hyperplane : $x_1 = 0$.

partition is relatively good, so RAPT does not come very far behind the OPRAs.

In general, RAPT's and OPRA_0's running times are nearly identical, and are slightly faster than OPRA (this difference slowly increases with K). The gap between RAPTOR's running time and its competitors' is however much larger and significantly widens with K, as expected. For $K = 5$, RAPTOR takes roughly twice as long as its competitors to perform a fixed number of iterations.

It is interesting to note that OPRA_0 sometimes does better than OPRA, a phenomenon that gradually becomes more frequent as K grows. Indeed having more regions improves the flexibility of the proposal, so obtaining highly precise frontiers might not be as crucial for some targets. As witnessed in our experiments this extra flexibility, when superfluous, might lead to slight performance losses since there is no compensation for this instability in the initial stage of the sampler. Overall, results from examples with $K > 2$ are consistent with our previous findings, and in agreement with the complexity analysis of Section 4.3.

7. REAL DATA EXAMPLE : GENETIC INSTABILITY OF ESOPHAGEAL CANCERS

Loss of heterozygosity (LOH) is one of the genetic changes suffered by cancer cells during disease progression. Of interest in cancer studies are chromosome regions with high rates of LOH, which are hypothesized to contain genes regulating cell behavior. The goal of the Seattle Barrett's Esophagus research project is to locate "Tumor Suppressor Genes" (TSGs). The associated dataset (Barrett et al. [1996]) contains LOH rates from esophageal cancers for 40 regions (under the form of a frequency and sample size for each region). In order to determine the probability of LOH in the background and TSG groups, Desai [2000] proposes to model the frequencies using the following hierarchical mixture:

$$X_i \sim \eta \text{ Binomial}(N_i, \pi_1) + (1-\eta) \text{ Beta-Binomial}(N_i, \pi_2, \gamma) \,,$$

Algorithm	RAPT		RAPTOR		OPRA_0		OPRA	
Criteria	mean	s.d.	mean	s.d.	mean	s.d.	mean	s.d.
	$K=3$	$d=50$	$n=2\cdot10^5$	$t_0=10^4$	$N=10^3$	$k=4$		
Time (sec)	42.61	9.12e-03	73.30	6.48e-02	42.84	1.23e-02	44.70	1.13e-02
Mean MSE	2.94	2.01e-02	1.60	3.31e-02	4.79	1.03e-02	1.97	3.15e-02
Cov. 95% (%)	1.60	1.23e-02	1.26	1.41e-02	1.42	1.30e-02	1.61	1.26e-02
T_a	0.24	2.44e-04	0.23	2.59e-03	0.24	1.60e-04	0.23	2.53e-04

TABLE 5: Unimodal, stretched normal target with slight twisting and poor initial partitions, 3 regions: $\mu_1=(0,\ldots,0)$, $\Sigma_1=\text{diag}(100,1,\ldots,1)$, $\psi=0.03$. Parameters: $X_0^{(1)}=\ldots=X_0^{(4)}=\mu_3(0)=(0,\ldots,0)$, $\mu_1(0)=-\mu_2(0)=(-0.1,0,\ldots,0)$, $\Sigma_1(0)=\Sigma_2(0)=\Sigma_3(0)=0.1I_d$, $\Sigma_S(0)=100I_d/d$. Initial hyperplanes: $x_2=1$, $x_2=0$, $x_2=-2$.

Algorithm	RAPT		RAPTOR		OPRA_0		OPRA	
Criteria	mean	s.d.	mean	s.d.	mean	s.d.	mean	s.d.
	$K=5$	$d=50$	$n=2\cdot10^5$	$t_0=10^4$	$N=10^3$	$k=4$		
Time (sec)	57.01	1.46e-02	116.42	1.54e-01	57.36	1.43e-02	63.71	5.60e-02
Mean MSE	7.53	4.24e-03	7.59	4.69e-03	7.56	4.90e-03	7.57	4.76e-03
Cov. 95% (%)	3.31	1.33e-02	2.69	1.90e-02	3.22	1.41e-02	3.21	1.51e-02
T_a	0.26	1.79e-04	0.28	5.30e-04	0.26	1.61e-04	0.29	4.58e-04

TABLE 6: Normal target with 5 spherical modes: $\mu_1=(0,\ldots,0)$, $\mu_2=-\mu_3=(2.5,\ldots,2.5)/d$, $\mu_4=-\mu_5=(-2.5,2.5,\ldots,2.5)/d$, $\Sigma_1=I_d$, $\Sigma_2=\cdots=\Sigma_5=I_d/2$, $p_1=\cdots=p_5=1/5$. Parameters: $\mu_1(0)=X_0^{(1)}=-\mu_2(0)=-X_0^{(2)}=(2,0,\ldots,0)$, $\mu_3(0)=X_0^{(3)}=X_0^{(4)}=(0,\ldots,0)$, $\mu_4(0)=-\mu_5(0)=(0,2,0\ldots,0)$, $\Sigma_1(0)=\cdots=\Sigma_5(0)=0.1I_d$, $\Sigma_S(0)=100I_d/d$. Initial hyperplanes: $x_1=0,\ldots,x_{10}=0$.

with priors for η,π_1,π_2 each distributed as a Unif$[0,1]$ and $\gamma\sim$ Unif$[-30,30]$. The labeling of the background and TSG groups being unknown, η represents the probability that a region belongs to the binomial group. The parameters π_1,π_2 are the probabilities of LOH in the binomial and beta-binomial groups respectively, while γ controls the variability of the latter.

To simplify the use of MCMC samplers, the beta-binomial is parameterized so that $\gamma\in\mathbb{R}$. Furthermore, a logistic transformation is applied to η,π_1,π_2 so that they each be supported on $\mathbb{R}$; we refer the reader to Craiu et al. [2009] for more details. In practice, restricting the resulting posterior distribution to a sufficiently large, 4-dim compact set then ensures the ergodicity of the regional adaptive MCMC samplers, without significantly affecting the analysis.

In implementing RAPT, Craiu et al. [2009] had to rely on the optimization used in Warnes [2001] to determine a reasonable partition $\mathcal{S}_1\cup\mathcal{S}_2$ of the four-dimensional space $\mathcal{S}$. According to that procedure, the two modes of π are reasonably well separated when we choose $\mathcal{S}_1=\{(\eta,\pi_1,\pi_2,\gamma)\in[0,1]\times[0,1]\times[0,1]\times[-30,30]|\pi_2\geq\pi_1\}$ and $\mathcal{S}_2=\{(\eta,\pi_1,\pi_2,\gamma)\in[0,1]\times[0,1]\times[0,1]\times[-30,30]|\pi_2\leq\pi_1\}$. The implementation of OPRA does not require this extra step, as even poor initial partitions lead to an efficient exploration of $\mathcal{S}$. Instead of relying on the partition proposed by Warnes [2001], we initialize OPRA with the uninformed partition generated by $a_0^TX=b_0$ with $a_0^T=(1,1,1,1)$ and $b_0=0$.

To provide a fair comparison with the RAPT of Craiu et al. [2009] and the RAPTOR of Bai et al. [2011], we use the same initialization parameters as them. We set $\beta=0.3$ and covariance matrices are initialized to $\Sigma_1(0)=\Sigma_2(0)=0.1I_4$, $\Sigma_S(0)=20I_4$. Starting points are randomly selected over the prior range (in the original parameterization). Using OPRA we run

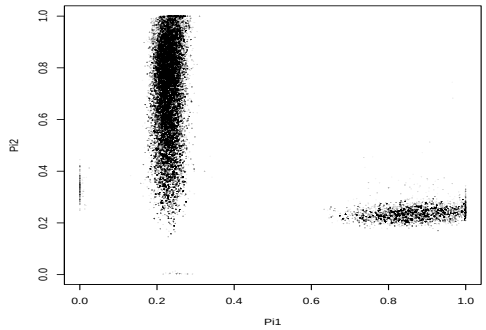

FIGURE 4: Scatterplot of the samples for (π_1, π_2).

five independent parallel chains and perform 800,000 iterations for each chain; Figure 4 shows a two-dimensional scatterplot of (π_1, π_2) based on all the samples obtained. The graph is very similar to that in Craiu et al. [2009] based on RAPT with Warne's partition, and also to that in Bai et al. [2011] obtained with RAPTOR. In particular, the two distinct modes are clearly identifiable.

Table 7 provides a summary of estimates from RAPT (with Warnes' partition) and OPRA, detailed by regions (on the original scales). For results to be comparable to those of Craiu et al. [2009], we use the first 55,000 iterations of each of our five parallel chains, from which we drop the first 5,000 iterations as burn-in. Even though OPRA is at a net disadvantage with its uninformed initial partition, estimates for the whole space are quite close to those of RAPT. This not only testifies of the quality partitioning process proposed by OPRA, but also of the speed at which it stabilizes. While estimates for Region 1 agree under both approaches, estimates for Region 2 are not as similar. We should note that in the current 4-dim space, there are very likely more than one decent partition of $\mathcal{S}$, and OPRA seems to rely on a partition different from that of Warnes [2001].

Algorithm	RAPT			OPRA		
Mean in	Region 1	Region 2	Whole space	Region 1	Region 2	Whole space
η	0.897	0.079	0.838	0.898	0.934	0.901
π_1	0.229	0.863	0.275	0.229	0.233	0.230
π_2	0.714	0.237	0.679	0.720	0.865	0.729
γ	15.661	-14.796	13.435	14.222	-15.149	12.401

TABLE 7: Simulation results for the LOH data (250,000 iterations).

Table 8 contains estimates for the parameters of interest (both regions confounded) from OPRA, RAPTOR, and RAPT. To compare our results with those of Bai et al. [2011] for RAPTOR, we use four of our five available 800,000-iteration parallel chains, from which we drop the first 40,000 iterations as burn-in. While the above uninformed initial partition is used in OPRA, starting mixture estimates for RAPTOR are slightly informed, with $\hat{\mu}_1(0) = (2.2, -1.4, 1.4, 12.2)$, $\hat{\mu}_2(0) = (-2.2, 2.2, -1.15, -13.25)$, and $(0.8, 0.2)$ for the modes weights. RAPT figures are based on the 55,000-iteration runs of Craiu et al. [2009] and are included as a baseline only. Once again, parameter estimates in Table 8 are comparable under all three approaches; OPRA's γ is slightly below the others, but still largely within one standard

deviation according to Bai et al. [2011], who highlight the high variability of this parameter. OPRA runs in 85% of RAPTOR's time and its performance is, overall, convincing and reliable in this real data context.

	η	π_1	π_2	γ	Time (sec)
RAPT	0.838	0.275	0.679	13.435	95.382
OPRA	0.824	0.285	0.671	10.085	94.686
RAPTOR	0.828	0.248	0.614	12.732	112.378

TABLE 8: Simulation results for the LOH data; global parameters means (3,200,000 iterations).

8. DISCUSSION

The adaptive partitioning process proposed in this paper is used in conjunction with RAPT in the context of two separate regions. The approach may be generalized to more regions by obtaining a hyperplane for each pair of sample averages, and then considering appropriate intersections arising from the hyperplanes. The resulting sampler is ergodic in the sense that it satisfies the *Simultaneous Uniform Ergodicity* and *Diminishing Adaptation* conditions for adaptive algorithms.

The simplicity of this new adaptive partitioning of the sample space is what makes it appealing. In the examples considered, the partitioning process has been seen to stabilize rapidly, regardless of the quality of the initial partition. For a fixed number of iterations, it is understood that OPRA produces better results than RAPT alone, and that it often compares favorably to RAPTOR. OPRA's ease of implementation, along with an adaptive partitioning step that is virtually free in terms of running time, consolidates its avantage over the other regional adaptive samplers when accounting for computational effort. According to our extensive numerical explorations, it would seem that OPRA does not suffer the same struggles as RAPTOR in dealing, for instance, with low-dimensional target densities or short pre-adaptation times. According to the results presented in this paper, users that intend to implement RAPT might want to hedge their initial partition choice by relying on the adaptive partitioning process. This will provide some peace of mind at a marginal computational cost.

ACKNOWLEDGEMENTS

We are grateful to the AE and referees, whose comments have contributed to significantly improve the paper. This work has been supported by the Natural Sciences and Engineering Research Council of Canada.

BIBLIOGRAPHY

C. Andrieu, É. Moulines, et al., On the ergodicity properties of some adaptive MCMC algorithms, The Annals of Applied Probability 16 (2006) 1462–1505.

C. Andrieu, C.P. Robert, Controlled MCMC for optimal sampling, 2001. Technical report.

Y.F. Atchadé, J.S. Rosenthal, On adaptive Markov chain Monte Carlo algorithms, Bernoulli 11 (2005) 815–828.

Y. Bai, R.V. Craiu, A.F. Di Narzo, Divide and conquer: a mixture-based approach to regional adaptation for MCMC, Journal of Computational and Graphical Statistics 20 (2011) 63–79.

M. Barrett, P. Galipeau, C. Sanchez, M. Emond, B. Reid, Determination of the frequency of loss of heterozygosity in esophageal adenocarcinoma by cell sorting, whole genome amplification and microsatellite polymorphisms., Oncogene 12 (1996) 1873–1878.

L. Bornn, P.E. Jacob, P. Del Moral, A. Doucet, An adaptive interacting Wang–Landau algorithm for automatic density exploration, Journal of Computational and Graphical Statistics 22 (2013) 749–773.

N. Chopin, T. Lelièvre, G. Stoltz, Free energy methods for Bayesian inference: Efficient exploration of univariate Gaussian mixture posteriors, Statistics and Computing 22 (2012) 897–916.

R.V. Craiu, J.S. Rosenthal, C. Yang, Learn from thy neighbor: Parallel-chain and regional adaptive MCMC, Journal of the American Statistical Association 104 (2009) 1454–1466.

M. Desai, Mixture models for genetic changes in cancer cells, 2000. Ph.D. thesis, University of Washington.

A. Gelman, D.B. Rubin, et al., Inference from iterative simulation using multiple sequences, Statistical science 7 (1992) 457–472.

C.J. Geyer, E.A. Thompson, Annealing Markov chain Monte Carlo with applications to ancestral inference, Journal of the American Statistical Association 90 (1995) 909–920.

Y. Guan, S.M. Krone, et al., Small-world MCMC and convergence to multi-modal distributions: From slow mixing to fast mixing, The Annals of Applied Probability 17 (2007) 284–304.

H. Haario, M. Laine, A. Mira, E. Saksman, DRAM: Efficient adaptive MCMC, Statistics and Computing 16 (2006) 339–354.

H. Haario, E. Saksman, J. Tamminen, An adaptive Metropolis algorithm, Bernoulli 7 (2001) 223–242.

H. Haario, E. Saksman, J. Tamminen, Componentwise adaptation for high dimensional MCMC, Computational Statistics 20 (2005) 265–273.

W.K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, Biometrika 57 (1970) 97–109.

S. Kou, Q. Zhou, W.H. Wong, et al., Equi-energy sampler with applications in statistical inference and statistical mechanics, The annals of Statistics 34 (2006) 1581–1619.

D. Landau, S.H. Tsai, M. Exler, A new approach to Monte Carlo simulations in statistical physics: Wang-Landau sampling, American Journal of Physics 72 (2004) 1294–1302.

N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, Equation of state calculations by fast computing machines, The journal of chemical physics 21 (1953) 1087–1092.

R.M. Neal, Sampling from multimodal distributions using tempered transitions, Statistics and computing 6 (1996) 353–366.

G. Roberts, J. Rosenthal, Optimal scaling for various Metropolis-Hastings algorithms, Statistical Science 16 (2001) 351–367.

G. Roberts, J. Rosenthal, Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms, Journal of Applied Probability 44 (2007) 458–475.

G. Roberts, J. Rosenthal, Examples of adaptive MCMC, Journal of Computational and Graphical Statistics 18 (2009) 349–367.

A. Solonen, P. Ollinaho, M. Laine, H. Haario, J. Tamminen, H. Järvinen, et al., Efficient MCMC for climate model parameter estimation: Parallel adaptive chains and early rejection, Bayesian Analysis 7 (2012) 715–736.

G. Warnes, The normal kernel coupler: an adaptive Markov chain Monte Carlo method for efficiently sampling from multi-modal distributions, 2001. Technical report.

Received 15 April 2019
Accepted 27 February 2020