Martino *et al. EURASIP Journal on Advances in Signal Processing* (2018) 2018:5
DOI 10.1186/s13634-017-0524-6

EURASIP Journal on Advances in Signal Processing

RESEARCH Open Access

Adaptive independent sticky MCMC algorithms

Luca Martino[1*], Roberto Casarin[2], Fabrizio Leisen[3] and David Luengo[4]

Abstract

Monte Carlo methods have become essential tools to solve complex Bayesian inference problems in different fields, such as computational statistics, machine learning, and statistical signal processing. In this work, we introduce a novel class of adaptive Monte Carlo methods, called adaptive independent sticky Markov Chain Monte Carlo (MCMC) algorithms, to sample efficiently from any bounded target probability density function (pdf). The new class of algorithms employs adaptive non-parametric proposal densities, which become closer and closer to the target as the number of iterations increases. The proposal pdf is built using interpolation procedures based on a set of support points which is constructed iteratively from previously drawn samples. The algorithm's efficiency is ensured by a test that supervises the evolution of the set of support points. This extra stage controls the computational cost and the convergence of the proposal density to the target. Each part of the novel family of algorithms is discussed and several examples of specific methods are provided. Although the novel algorithms are presented for univariate target densities, we show how they can be easily extended to the multivariate context by embedding them within a Gibbs-type sampler or the hit and run algorithm. The ergodicity is ensured and discussed. An overview of the related works in the literature is also provided, emphasizing that several well-known existing methods (like the adaptive rejection Metropolis sampling (ARMS) scheme) are encompassed by the new class of algorithms proposed here. Eight numerical examples (including the inference of the hyper-parameters of Gaussian processes, widely used in machine learning for signal processing applications) illustrate the efficiency of sticky schemes, both as stand-alone methods to sample from complicated one-dimensional pdfs and within Gibbs samplers in order to draw from multi-dimensional target distributions.

Keywords: Bayesian inference, Monte Carlo methods, Adaptive Markov chain Monte Carlo (MCMC), Adaptive rejection Metropolis sampling (ARMS), Gibbs sampling, Metropolis-within-Gibbs, Hit and run algorithm

1 Introduction

Markov chain Monte Carlo (MCMC) methods [1, 2] are very important tools for Bayesian inference and numerical approximation, which are widely employed in signal processing [3–7] and other related fields [1, 8]. A crucial issue in MCMC is the choice of a proposal probability density function (pdf), as this can strongly affect the mixing of the MCMC chain when the target pdf has a complex structure, e.g., multimodality and/or heavy tails. Thus, in the last decade, a remarkable stream of literature focuses on adaptive proposal pdfs, which allow for self-tuning procedures of the MCMC algorithms, flexible movements within the state space, and improved acceptance rates [9, 10].

Adaptive MCMC algorithms are used in many statistical applications and several schemes have been proposed in the literature [8–11]. There are two main families of methods: parametric and non-parametric. The first strategy consists in adapting the parameters of a parametric proposal pdf according to the past values of the chain [10]. However, even if the parameters are perfectly adapted, a discrepancy between the target and the proposal pdfs remains. A second strategy attempts to adapt the entire shape of the proposal density using non-parametric procedures [12, 13]. Most authors have payed more attention to the first family, designing local adaptive random-walk

*Correspondence: lukatotal@gmail.com
[1]Image Processing Lab., University of Valencia, Valencia, Spain
Full list of author information is available at the end of the article

algorithms [9, 10], due to the difficulty of approximating the full target distribution by non-parametric schemes with any degree of generality.

In this work, we describe a general framework to design suitable adaptive MCMC algorithms with non-parametric proposal densities. After describing the different building blocks and the general features of the novel class, we introduce two specific algorithms. Firstly, we describe the adaptive independent sticky Metropolis (AISM) algorithm to draw efficiently from any bounded univariate target distribution.[1] Then, we also propose a more efficient scheme that is based on the multiple try Metropolis (MTM) algorithm: the adaptive independent sticky Multiple Try Metropolis (AISMTM) method. The ergodicity of the adaptive sticky MCMC methods is ensured and discussed. The underlying theoretical support is based on the approach introduced in [14]. The new schemes are particularly suitable for sampling from complicated full-conditional pdfs within a Gibbs sampler [5–7].

Moreover, the new class of methods encompasses different well-known algorithms available in literature: the *griddy Gibbs sampler* [15], the *adaptive rejection Metropolis sampler* (ARMS) [12, 16], and the *independent doubly adaptive Metropolis sampler* (IA^{2}RMS) [13, 17]. Other related or similar approaches are also discussed in Section 6. The main contributions of this paper are the following:

1. A very general framework, that allows practitioners to design proper adaptive MCMC methods by employing a non-parametric proposal.
2. Two algorithms (AISM and AISMTM), that can be used off-the-shelf in signal processing applications.
3. An exhaustive overview of the related algorithms proposed in the literature, showing that several well-known methods (such as ARMS) are encompassed by the proposed framework.
4. A theoretical analysis of the AISM algorithm, proving its ergodicity and the convergence of the adaptive proposal to the target.

The structure of the paper is the following. Section 2 introduces the generalities of the class of sticky MCMC methods and the AISM scheme. Sections 3 and 4 present the general properties, altogether with specific examples, of the proposal constructions and the update control tests. Section 5 introduces some theoretical results. Section 6 discusses several related works and highlights some specific techniques belonging to the class of sticky methods. Section 7 introduces the AISMTM method. Section 8 describes the range of applicability of the proposed framework, including its use within other Monte Carlo methods (like the Gibbs sampler or the hit and run algorithm) to sample from multivariate distributions. Eight numerical examples (including the inference of hyper-parameters of Gaussian processes) are then provided in Section 9.[2] Finally, Section 10 contains some conclusions and possible future lines.[3]

2 Adaptive independent sticky MCMC algorithms

Let $\widetilde{\pi}(x) \propto \pi(x) > 0$, with $x \in \mathcal{X} \subseteq \mathbb{R}$, be a bounded[4] target density known up to a normalizing constant, $c_\pi = \int_{\mathcal{X}} \pi(x)dx$, from which direct sampling is unfeasible. In order to draw from it, we employ an MCMC algorithm with an independent adaptive proposal,

$$\widetilde{q}_t(x|\mathcal{S}_t) \propto q_t(x|\mathcal{S}_t) > 0, \quad x \in \mathcal{X},$$

where t is the iteration index of the corresponding MCMC algorithm, and $\mathcal{S}_t = \{s_1, \ldots, s_{m_t}\}$ with $m_t > 0$ is the set of support points used for building $\widetilde{q}_t$. At the t-th iteration, an adaptive independent sticky MCMC method is conceptually formed by three stages (see Fig. 1):

1. *Construction of the non-parametric proposal*: given the nodes in $\mathcal{S}_t$, the function q_t is built using a suitable non parametric procedure that provides a function which is closer and closer to the target as the number of points m_t increases. Section 3 describes the general properties that must be fulfilled by a suitable proposal construction, as well as specific procedures to build this proposal.

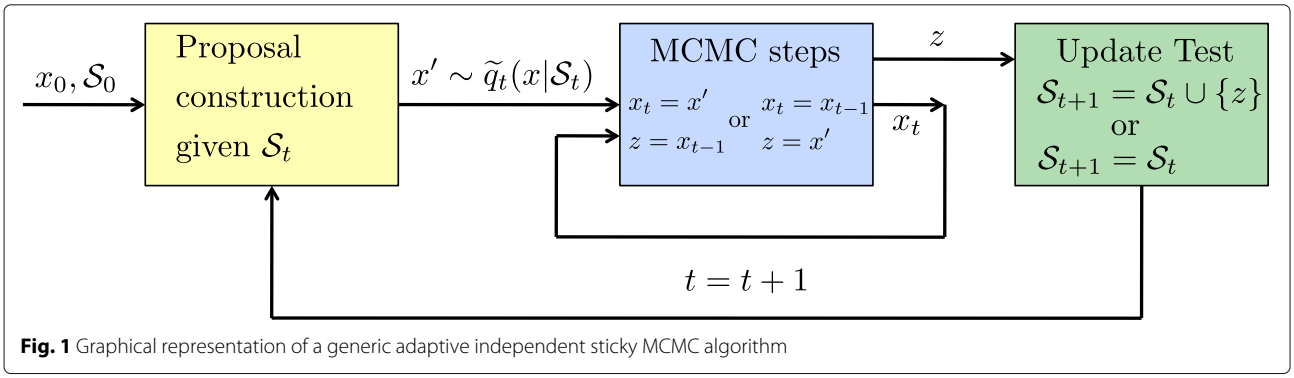


Fig. 1 Graphical representation of a generic adaptive independent sticky MCMC algorithm

2. *MCMC stage*: some MCMC method is applied in order to produce the next state of the chain, x_t, employing $\widetilde{q}_t(x|\mathcal{S}_t)$ as (part of the) proposal pdf. This stage produces the next state of the chain, x_{t+1}, and an auxiliary variable z (see Tables 1 and 4), used in the following update stage.
3. *Update stage*: A statistical test on the auxiliary variable z is performed in order to decide whether to increase the number of points in $\mathcal{S}_t$ or not, defining a new support set, $\mathcal{S}_{t+1}$, which is used to construct the proposal at the next iteration. The update stage controls the computational cost and ensures the ergodicity of the generated chain (see Appendix A). Section 4 is devoted to the design of different suitable update rules.

In the following section, we describe the simplest possible sticky method, obtained by using the MH algorithm, whereas in Section 7 we consider a more sophisticated technique that employs the MTM scheme.[5]

2.1 Adaptive independent sticky Metropolis

The simplest adaptive independent sticky method is the *adaptive independent sticky Metropolis* (AISM) technique, outlined in Table 1. In this case, the proposal pdf $\widetilde{q}_t(x|\mathcal{S}_t)$ changes along the iterations (see step 1 of Table 1) following an adaptation scheme that relies upon a suitable interpolation given the set of support points $\mathcal{S}_t$ (see Section 3). Step 3 of Table 1 applies a statistical control to update the set $\mathcal{S}_t$. The point z, rejected at the current iteration of the algorithm in the MH test, is added to $\mathcal{S}_t$ with probability

$$P_a(z) = \eta_t(z, d_t(z)), \tag{1}$$

Table 1 Adaptive independent sticky Metropolis (AISM)

For $t = 0, \ldots, T-1$:

1 Construction of the proposal: Build a proposal function $q_t(x|\boldsymbol{\mathcal{S}}_t)$ via a suitable procedure using a set of support points $\boldsymbol{\mathcal{S}}_t$ (see Section 3).
2 MH step:

2.1 Draw $x' \sim \widetilde{q}_t(x|\boldsymbol{\mathcal{S}}_t) \propto q_t(x|\boldsymbol{\mathcal{S}}_t)$.
2.2 Set $x_t = x'$ and $z = x_{t-1}$ with probability

$$\alpha = \min\left[1, \frac{\pi(x')q_t(x_{t-1}|\boldsymbol{\mathcal{S}}_t)}{\pi(x_{t-1})q_t(x'|\boldsymbol{\mathcal{S}}_t)}\right].$$

Otherwise, set $x_t = x_{t-1}$ and $z = x'$.

3 Test to update $\boldsymbol{\mathcal{S}}_t$: Let $\eta_t(z,d) : \boldsymbol{\mathcal{X}} \times \mathbb{R}^+ \to [0,1]$ be an increasing function w.r.t. the variable d, such that $\eta_t(z,0) = 0$ and $\lim_{d\to\infty} \eta_t(z,d) = 1$. Then, set

$$\boldsymbol{\mathcal{S}}_{t+1} = \begin{cases} \boldsymbol{\mathcal{S}}_t \cup \{z\}, & \text{with prob. } P_a(z) = \eta_t(z, d_t(z)), \\ \boldsymbol{\mathcal{S}}_t, & \text{with prob. } 1 - P_a(z), \end{cases}$$

where $d_t(z) = |\pi(z) - q_t(z|\boldsymbol{\mathcal{S}}_t)|$.

where $\eta_t(z,d) : \mathcal{X} \times \mathbb{R}^+ \to [0,1]$is an increasing test function w.r.t. the variable d, such that $\eta_t(z,0) = 0$, and $d = d_t(z) = \left|\pi(z) - q_t(z|\mathcal{S}_t)\right|$. is the point distance between π and q_t at z. The rationale behind this test is to use information from the target density in order to include in the support set only those points where the proposal pdf differs substantially from the target value at z. Note that, since z is always different from the current state (i.e., $z \neq x_t$ for all t), then the proposal pdf is *independent* from the current state according to Holden's definition [14], and thus the theoretical analysis is greatly simplified.

3 Construction of the sticky proposals

There are many alternatives available for the construction of a suitable *sticky proposal* (SP). However, in order to be able to provide some theoretical results in Section 5, let us define precisely what we understand here by a sticky proposal.

Definition 1 (Valid Adaptive Proposal) *Let us consider a target density, $\widetilde{\pi}(x) \propto \pi(x) > 0$ for any $x \in \mathcal{X} \subseteq \mathbb{R}$ (the target's support), and a set of $m_t = |\mathcal{S}_t|$ support points, $\mathcal{S}_t = \{s_1, \ldots, s_{m_t}\}$ with $s_i \in \mathcal{X}$ for all $i = 1, \ldots, m_t$. An adaptive proposal built using $\mathcal{S}_t$ via some non-parametric interpolation approach is considered valid if the following four properties are satisfied:*

1. *The proposal function is positive, i.e., $q_t(x|\mathcal{S}_t) > 0$ for all $x \in \mathcal{X}$ and for all possible sets $\mathcal{S}_t$ with $t \in \mathbb{N}$.*
2. *Samples can be drawn directly and easily from the resulting proposal, $\widetilde{q}_t(x|\mathcal{S}_t) \propto q_t(x|\mathcal{S}_t)$, using some exact sampling procedure.*
3. *For any bounded target, $\pi(x)$, the resulting proposal function, $q_t(x|\mathcal{S}_t)$, is also bounded. Furthermore, defining $\mathcal{I}_t = (s_1, s_{m_t}]$, we have*

$$\max_{x\in\mathcal{I}_t} q_t(x|\mathcal{S}_t) \leq \max_{x\in\mathcal{I}_t} \pi(x).$$

4. *The proposal function, $q_t(x|\mathcal{S}_t)$, has heavier tails than the target, i.e., defining $\mathcal{I}_t^c = (-\infty, s_1] \cup (s_{m_t}, \infty)$, we have*

$$q_t(x|\mathcal{S}_t) \geq \pi(x) \qquad \forall x \in \mathcal{I}_t^c.$$

Condition 1 guarantees that the function $q_t(x|\mathcal{S}_t)$ leads to a valid pdf, $\widetilde{q}_t(x|\mathcal{S}_t)$, that covers the entire support of the target. Condition 2 is required from a practical point of view to obtain efficient algorithms. Finally, conditions 3 and 4 are required by the proofs of Theorems 3 and 1, respectively, and also make sense from a practical point of view: if the target is bounded, we would expect the proposal learnt from it to be also bounded and this proposal should be heavier tailed than the target in order to avoid under-sampling the tails. Now we can define precisely what we understand by a "sticky" proposal.

Definition 2 (Sticky Proposal (SP)) *Let us consider a valid proposal pdf according to Definition* 1. *Let us assume also that the i-th support point is distributed according to* $p_i(x)$ *(i.e.,* $s_i \sim p_i(x)$*) such that* $p_i(x) > 0$ *for any* $x \in \mathcal{X}$ *and* $i = 1, \ldots, m_t$. *Then, a sticky proposal is any valid proposal pdf s.t. the* L_1 *distance between* $q_t(x)$ *and* $\pi(x)$ *vanishes to zero when the number of support points increases, i.e., if* $m_t \to \infty$,

$$D_1(\pi, q_t) = \|\pi - q_t\|_1 = \int_{\mathcal{X}} |\pi(z) - q_t(z|\mathcal{S}_t)| dz = \int_{\mathcal{X}} d_t(z) dz \to 0, \quad (2)$$

where $d_t(z) = |\pi(z) - q_t(z|\mathcal{S}_t)|$ *is the* L_1 *distance between* $\pi(x)$ *and* $q_t(x)$ *evaluated at* $z \in \mathcal{X}$, *and* (2) *implies almost everywhere (a.k.a., almost surely) convergence of* $q_t(x)$ *to* $\pi(x)$.

In the following, we provide some examples of constructions that fulfill all the conditions in Definitions 1 and 2. All of them approximate the target pdf by interpolating points that belong to the graph of the target function π.

3.1 Examples of constructions

Given $\mathcal{S}_t = \{s_1, \ldots, s_{m_t}\}$ at the t-th iteration, let us define a sequence of $m_t + 1$ intervals: $\mathcal{I}_0 = (-\infty, s_1]$, $\mathcal{I}_j = (s_j, s_{j+1}]$ for $j = 1, \ldots, m_t - 1$, and $\mathcal{I}_{m_t} = (s_{m_t}, +\infty)$. The simplest possible procedure uses piecewise constant (uniform) pieces in $\mathcal{I}_i$, $1 \leq i \leq m_t - 1$, with two exponential tails in the first and last intervals [13, 15, 18]. Mathematically,

$$q_t(x|\mathcal{S}_t) = \begin{cases} E_0(x), & x \in \mathcal{I}_0, \\ \max\{\pi(s_i), \pi(s_{i+1})\}, & x \in \mathcal{I}_i, \\ E_{m_t}(x), & x \in \mathcal{I}_{m_t}, \end{cases} \quad (3)$$

where $1 \leq i \leq m_t - 1$ and $E_0(x)$, $E_{m_t}(x)$ represent two exponential pieces. These two exponential tails can be obtained simply constructing two straight lines in the log-domain as shown in [12, 13, 19]. For instance, defining $V(x) = \log[\pi(x)]$, we can build the straight line $w_0(x)$ passing through the points $(s_1, V(s_1))$ and $(s_2, V(s_2))$, and the straight line $w_{m_t}(x)$ passing through the points $(s_{m_t-1}, V(s_{m_t-1}))$ and $(s_{m_t}, V(s_{m_t}))$. Hence, the proposal function is defined as $E_0(x) = \exp(w_0(x))$ for $x \in \mathcal{I}_0$ and $E_{m_t}(x) = \exp(w_{m_t}(x))$ for $x \in \mathcal{I}_{m_t}$. Other kinds of tails can be built, e.g., using Pareto pieces as shown in Appendix E.2. Alternatively, we can use piecewise linear pieces [20]. The basic idea is to build straight lines, $L_{i,i+1}(x)$, passing through the points $(s_i, \pi(s_i))$ and $(s_{i+1}, \pi(s_{i+1}))$ for $i = 1, \ldots, m_t - 1$, and two exponential pieces, $E_0(x)$ and $E_{m_t}(x)$, for the tails:

$$q_t(x|\mathcal{S}_t) = \begin{cases} E_0(x), & x \in \mathcal{I}_0, \\ L_{i,i+1}(x), & x \in \mathcal{I}_i, \\ E_{m_t}(x), & x \in \mathcal{I}_{m_t}, \end{cases} \quad (4)$$

with $i = 1, \ldots, m_t - 1$. Note that drawing samples from these trapezoidal pdfs inside $\mathcal{I}_i = (s_i, s_{i+1}]$ is straightforward [20, 21]. Figure 2 shows examples of the construction of $q_t(x|\mathcal{S}_t)$ using Eq. (3) or (4) with different number of points, $m_t = 6, 8, 9, 11$. See Appendix A for further considerations.

A more sophisticated and costly construction has been proposed for the ARMS method in [12]. However, note that this construction does not fulfill Condition 3 in Definition 1. A similar construction based on B-spline interpolation methods has been proposed in [22, 23] to build a non-adaptive random walk proposal pdf for an MH algorithm. Other alternative procedures can also be found in the literature [13, 16, 18–20].

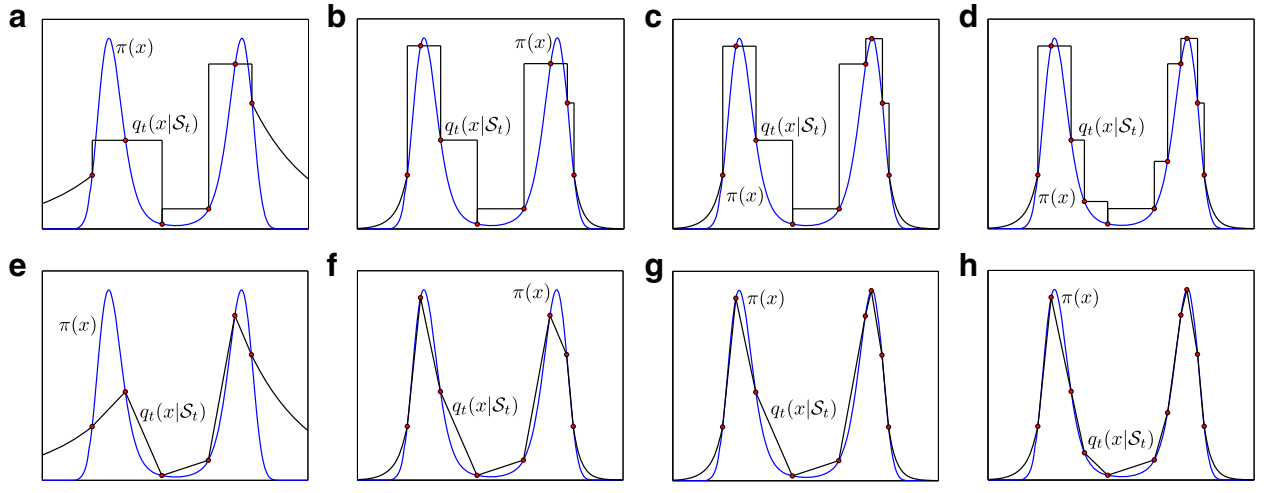


Fig. 2 Examples of the proposal construction q_t considering a bimodal target π, using the procedures described in Eq. (3) for **a**–**d** and in Eq. (4) for **e**–**h** with $m_t = 6, 8, 9, 11$ support points, respectively

4 Update of the set of support points

In AISM, a suitable choice of the function $\eta_t(z,d)$ is required. Although more general functions could be employed, we concentrate on test functions that fulfill the conditions provided in the following definition.

Definition 3 (Test Function) *Let us denote the L_1 distance between the target and the proposal at the t-th iteration, for any $z \in \mathcal{X}$, as $d = d_t(z) = |\pi(z) - q_t(z)|$. A valid test function, $\eta_t(z,d)$, is any function that fulfills all of the following properties:*

1. $\eta_t(z,d) : \mathcal{X} \times \mathbb{R}^+ \to [0,1]$.
2. $\eta_t(z,0) = 0$ *for all* $z \in \mathcal{X}$ *and* $t \in \mathbb{N}$.
3. $\lim_{d\to\infty} \eta_t(z,d) = 1$ *for all* $z \in \mathcal{X}$ *and* $t \in \mathbb{N}$.
4. $\eta_t(z,d)$ *is a strictly increasing function w.r.t. d, i.e.,* $\eta_t(z,d_2) > \eta_t(z,d_1)$ *for any* $d_2 > d_1$.

The first condition ensures that we obtain a valid probability for the addition of new support points, $P_a(z) = \eta_t(z,d)$, whereas the remaining three conditions imply that support points are more likely to be added in those areas where the proposal is further away from the target, with a non-null probability of adding new points in places where $d > 0$. In particular, Condition 4 is required by several theoretical results provided in the Appendix. However, update rules that do not fulfill this condition can also be useful, as discussed in the following. Figure 3 depicts an example of function η_t when $\eta_t(z,d) = \eta_t(d)$. Note that, for a given value of z, η_t satisfies all the properties of a continuous distribution function (cdf) associated to a positive random variable. Therefore, any pdf for positive random variables can be used to define a valid test function η_t through its corresponding cdf.

4.1 Examples of update rules

Below, we provide three different possible update rules. First of all, we consider the simplest case: $\eta_t(z,d) = \eta(d)$. As a first example, we propose

$$\eta_t(d) = 1 - e^{-\beta d}, \tag{5}$$

where $\beta > 0$ is a constant parameter. Note that this is the cdf associated to an exponential random variable.

A second possibility is

$$\eta_t(d) = \begin{cases} 1, & \text{if} \quad d > \varepsilon_t, \\ 0, & \text{if} \quad d \le \varepsilon_t; \end{cases} \tag{6}$$

where $0 < \varepsilon_t < M_\pi$, with $M_\pi = \max_{z\in\mathcal{X}}\{\pi(z)\}$,[6] is some appropriate time-varying threshold that can either follow some user pre-defined rule or be updated automatically.[7] Alternatively, we can also set this threshold to a fixed value, $\varepsilon_t = \varepsilon$, as done in the simulations. In this case, setting $\varepsilon \ge M_\pi$ implies that the update of $\mathcal{S}_t$ never happens (i.e., new support points are never added to the support set), whereas candidate nodes would be incorporated to $\mathcal{S}_t$ almost surely by setting $\varepsilon \to 0$. For any other value of ε (i.e., $0 < \varepsilon < M_\pi$), the adaptation would eventually stop and no support points would be added after some random number of iterations. Note that this update rule does not fulfill Condition 4 in Definition 3, implying that some of the theoretical results of Section 5 (e.g., Conjecture 1) are not applicable. However, we have included it here because it is a very simple rule that has shown a good performance in practice and can be useful to limit the number of support points by using a fixed value of ϵ. Finally, note also that Eq. (6) corresponds to the cdf associated to a Dirac's delta located at ε_t.

A third alternative is

$$\eta_t(z,d) = \frac{d}{\max\{\pi(z), q_t(z|\mathcal{S}_t)\}}. \tag{7}$$

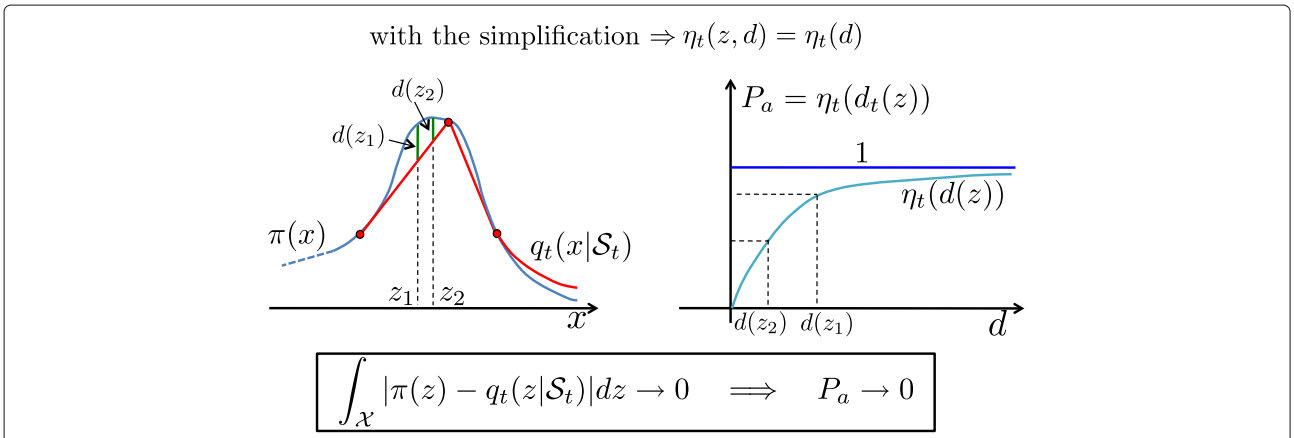


Fig. 3 Graphical representation of the underlying idea behind the update control test. For simplicity, in this figure, we have assumed $\eta_t(z,d) = \eta_t(d)$. As the proposal function q_t becomes closer and closer to π, the probability of adding a new node to $\mathcal{S}_t$ decreases

for $z \in \mathcal{X}$ and $0 \leq d \leq \max\{\pi(z), q_t(z|\mathcal{S}_t)\}$, since

$$\begin{aligned} d &= |\pi(z) - q_t(z|\mathcal{S}_t)|, \\ &= \max\{\pi(z), q_t(z|\mathcal{S}_t)\} - \min\{\pi(z), q_t(z|\mathcal{S}_t)\}, \\ &\leq \max\{\pi(z), q_t(z|\mathcal{S}_t)\}, \end{aligned} \tag{8}$$

This rule appears in other related algorithms, as discussed in Section 6.1. Furthermore, it corresponds to the cdf of a uniform random variable defined in the interval $[0, \max\{\pi(z), q_t(z|\mathcal{S}_t)\}]$. Hence, for a given value of z, the update test can be implemented as follows: (a) draw a samples v' uniformly distributed in the interval $\big[0, \max\{\pi(z), q_t(z|\mathcal{S}_t)\}\big]$; (b) if $v' \leq d_t(z)$, add z to the set of support points. A graphical representation of this rule is given in Fig. 4, whereas Table 2 summarizes all the previously described update rules.

5 Theoretical results

In this section, we provide some theoretical results regarding the ergodicity of the proposed approach, the convergence of a sticky proposal to the target, and the expected growth of the number of support points of the proposal. First of all, regarding the ergodicity of the AISM, we have the following theorem.

Theorem 1 (Ergodicity of AISM) *Let $x_1, x_2, \ldots, x_{T-1}$ be the set of states generated by the AISM algorithm in Table* 1*, using a valid adaptive proposal function, $\widetilde{q}_t(x|\mathcal{S}_t) = \frac{1}{c_t} q_t(x|\mathcal{S}_t)$, constructed according to Definition* 1*, and a test rule fulfilling the conditions in Definition* 3*. The pdf of x_t, $p_t(x)$, converges geometrically in total variation (TV) norm to the target, $\widetilde{\pi}(x) = \frac{1}{c_\pi}\pi(x)$, i.e.,*

$$\|p_t(x) - \widetilde{\pi}(x)\|_{TV} \leq 2 \prod_{\ell=1}^{t} (1 - a_\ell), \tag{9}$$

where

$$a_\ell = \min\left\{1, \frac{c_\pi}{c_\ell} \min_{x \in \mathcal{X}} \left\{\frac{q_\ell(x|\mathcal{S}_\ell)}{\pi(x)}\right\}\right\}. \tag{10}$$

with c_π and c_ℓ denoting the normalizing constants of $\pi(x)$ and $q_\ell(x|\mathcal{S}_\ell)$, respectively.

Proof See Appendix A. □

Theorem 1 ensures that the pdf of the states of the Markov chain becomes closer and closer to the target pdf as t increases, since $0 \leq 1 - a_t \leq 1$ and thus the product in the right hand side of (9) is a decreasing function of t. This theorem is a direct consequence of Theorem 2 in [14], and ensures the ergodicity of the proposed adaptive MCMC approach. Regarding the convergence of a sticky proposal to the target, we consider the following conjecture.

Conjecture 1 (Convergence of SP to the target) *Let $\widetilde{\pi}(x) = \frac{1}{c_\pi}\pi(x)$ be a continuous and bounded target pdf that has bounded first and second derivatives for all $x \in \mathcal{X}$. Let $\widetilde{q}_t(x|\mathcal{S}_t) = \frac{1}{c_t} q_t(x|\mathcal{S}_t)$ be a sticky proposal pdf, constructed according to Definition* 1 *by using either a piecewise constant (PWC) or piecewise linear (PWL) approximation (given by Eqs.* (3) *and* (4)*, respectively). Let us also assume that the support points have been obtained by applying a test rule according to Definition* 3 *within the AISM algorithm described in Table* 1*. Then, it is reasonable to assume that $q_t(x|\mathcal{S}_t)$ converges in L_1 distance to $\pi(x)$ as t increases (i.e., as the number of support points grows), i.e., as $t \to \infty$*

$$D_1(\pi, q_t) = \|\pi - q_t\|_1 = \int_{\mathcal{X}} |\pi(z) - q_t(z|\mathcal{S}_t)| dz \to 0.$$

An intuitive argumentation is provided in Appendix A.

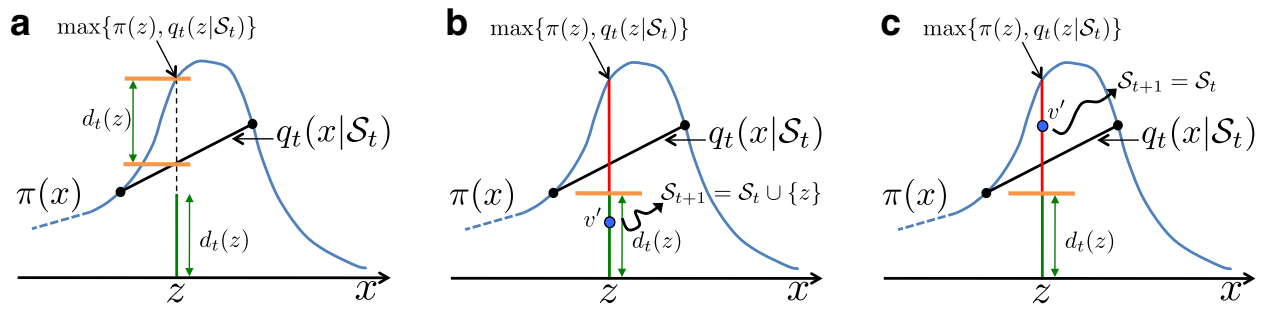


Fig. 4 Graphical interpretation of the third rule in Eq. (7) for the update control test. Given a point z, this test can be implemented as following: (1) draw a sample $v' \sim \mathcal{U}([0, \max\{\pi(z), q_t(z|\boldsymbol{\mathcal{S}}_t)\}])$, (2) then if $v' \leq d_t(z)$, add z to the set of support points, i.e., $\boldsymbol{\mathcal{S}}_{t+1} = \boldsymbol{\mathcal{S}}_t \cup \{z\}$. **a** The interval $[0, \max\{\pi(z), q_t(z|\boldsymbol{\mathcal{S}}_t)\}]$ and the distance $d_t(z)$. **b** The case when $v' \leq d_t(z)$ so that the point is incorporated in the set of support points whereas **c** illustrates the case when $v' > d_t(z)$; hence, $\boldsymbol{\mathcal{S}}_{t+1} = \boldsymbol{\mathcal{S}}_t$. Note that as the proposal function q_t becomes closer and closer to π (i.e., $d_t(z)$ decreases for any z), the probability of adding a new node to $\boldsymbol{\mathcal{S}}_t$ decreases

Table 2 Examples of test function $\eta_t(z,d)$ for different update rules (recall that $d = d_t(z) = |q_t(z|\boldsymbol{\mathcal{S}}_t) - \pi(z)|$)

Rule 1	$\eta_t(d) = 1 - e^{-\beta d}$
Rule 2	$\eta_t(d) = \begin{cases} 1, \text{if } d > \varepsilon_t, \\ 0, \text{if } d \leq \varepsilon_t \end{cases}$
Rule 3	$\eta_t(z,d) = \frac{d}{\max\{\pi(z), q_t(z\|\boldsymbol{\mathcal{S}}_t)\}}$

In the first and second cases, we have $\eta_t(z,d) = \eta_t(d)$

Note that Conjecture 1 essentially shows that the "sticky" condition is fulfilled for PWC and PWL proposals and continuous, bounded targets with bounded first and second derivatives. Note also that this conjecture implies that $q_t(x|\mathcal{S}_t) \to \pi(x)$ almost everywhere. Combining Theorem 1 and Conjecture 1 we get the following corollary.

Corollary 2 *Let $x_1, x_2, \ldots, x_{T-1}$ be the set of states generated by the AISM algorithm in Table* 1*, using either a PWC or a PWL sticky proposal function, $\widetilde{q}_t(x|\mathcal{S}_t) = \frac{1}{c_t} q_t(x|\mathcal{S}_t)$, constructed according to Definition* 2 *and a test rule fulfilling the conditions in Definition* 3*. Let $\widetilde{\pi}(x) = \frac{1}{c_\pi}\pi(x)$ be a continuous and bounded target pdf that has bounded first and second derivatives for all $x \in \mathcal{X}$. Then,*

$$\|\pi(x) - q_t(x)\|_{TV} \to 0 \qquad as \quad t \to \infty.$$

Proof By Theorem 1 we have

$$\|p_t(x) - \widetilde{\pi}(x)\|_{TV} \leq 2 \prod_{\ell=1}^{t} (1 - a_\ell),$$

with the a_ℓ given by Eq. (10). Now, since $q_\ell(x|\mathcal{S}_\ell) \to \pi(x)$ almost everywhere by Conjecture 1, we have $c_\ell \to c_\pi$ and thus $a_\ell \to 1$ as $\ell \to \infty$. Consequently, $\|\pi(x) - q_t(x)\|_{TV} \to 0$ as $t \to \infty$. □

Finally, we also have a bound on the expected growth of the number of support points, as provided by the following theorem.

Theorem 3 (Expected rate of growth of the number of support points) *Let $d_t(z) = |\pi(z) - q_t(z)|$ be the L_1 distance between the bounded target, $\pi(x)$, and an arbitrary sticky proposal function, $q_t(x)$, constructed according to Definition* 2*. Let also $\eta_t(z,d) = \eta_t(d)$ be an acceptance function that only depends on z through $d = d_t(z)$ and fulfills the conditions in Definition* 3*. The expected probability of adding a new support point in the AISM algorithm of Table* 1 *at the t-th iteration is*

$$E\left[P_a|x_{t-1}, \mathcal{S}_t\right] \leq \eta_t(d_t(x_{t-1}) + C \cdot D_1(\pi, q_t)), \qquad (11)$$

where $D_1(\pi, q_t) = \int_{\mathcal{X}} d_t(z)dz$, and $C = \max_{z\in\mathcal{X}} \widetilde{q}_t(z|\mathcal{S}_t)$ is a constant that depends on the sticky proposal used. Furthermore, under the conditions of Conjecture 1*, $E[P_a|x_{t-1}, \mathcal{S}_t] \to 0$ as $t \to \infty$.*

Proof See Appendix C.1. □

Theorem 3 sets a bound on the expected probability of adding new support points, and thus on the expected rate of growth of the number of support points. Furthermore, under certain smoothness assumptions for the target (i.e., that $\pi(x)$ is twice continuously differentiable), it also guarantees that this expectation tends to zero as the number of iterations increases, hence implying that less points are added as the algorithm evolves. Finally, note that Theorem 3 has been derived only for $\eta_t(z,d) = \eta_t(d)$. However, under certain mild assumptions, it can be easily extended to more general test functions, as stated in the following corollary.

Corollary 4 *Let $\eta_t(z, d_t(z)) = \eta_t(\widetilde{d}_t(z))$, where $\widetilde{d}_t(z) = \widetilde{d}_t(\pi(z), q_t(z))$ is some valid semi-metric and $\eta_t(\widetilde{d}_t(z))$ is a concave function of $\widetilde{d}_t(z)$. Then, if the rest of the conditions in Theorem* 3 *are satisfied, the expected probability of adding a new support point in the AISM algorithm of Table* 1 *at the t-th iteration is*

$$E[P_a|x_{t-1}, \mathcal{S}_t] \leq \eta_t\left(\widetilde{d}_t(x_{t-1}) + C \cdot \widetilde{D}_1(\pi, q_t)\right), \qquad (12)$$

where $\widetilde{D}_1(\pi, q_t) = \int_{\mathcal{X}} \widetilde{d}_t(z)dz$ and $C = \max_{z\in\mathcal{X}} \widetilde{q}_t(z|\mathcal{S}_t)$. Furthermore, under the conditions of Conjecture 1*, $E[P_a|x_{t-1}, \mathcal{S}_t] \to 0$ as $t \to \infty$.*

Proof See Appendix C.2. □

Note that Corollary 4 allows us to extend the results of Theorem 3 to update rule 3, which corresponds to $\eta_t(z, d_t(z)) = \widetilde{d}_t(z)$ with $\widetilde{d}_t(z) = \frac{d}{\max\{\pi(z), q_t(z|\mathcal{S}_t)\}}$ and d denoting the L_1 norm.

6 Related works

6.1 Other examples of sticky MCMC methods

The novel class of adaptive independent MCMC methods encompasses several existing algorithms already available in the literature, as shown in Table 3. We denote the proposal pdf employed in these methods as $p_t(x)$ and, for simplicity, we have removed the dependence on $\mathcal{S}_t$ in the function $q_t(x)$. The Griddy Gibbs Sampler [15] builds a proposal pdf as in Eq. (3), which is never adapted later. ARMS [12] and IA^{2}RMS [13] use as proposal density

$$p_t(x) \propto \min\left\{q_t(x), \pi(x)\right\},$$

where $q_t(x)$ is built using different alternative methods [12, 13, 16, 18]. Note that it is possible to draw easily

Table 3 Special cases of sticky MCMC algorithms

Features	Griddy Gibbs	ARMS	IA^{2}RMS
Main reference	[15]	[12]	[13]
Proposal pdf $p_t(x)$	$p_t(x) = \widetilde{q}_t(x)$	$p_t(x) \propto \min\{q_t(x), \pi(x)\}$	$p_t(x) \propto \min\{q_t(x), \pi(x)\}$
Proposal Constr.	Eq. (3)	[12],[16]	Eqs. (3)-(4), [13]
Update rule or $P_a(z)$	Never update, i.e.,	If $q_t(z) \geq \pi(x)$ then Rule 3,	Rule 3
	Rule 2	If $q_t(z) < \pi(x)$ then	
	with $\epsilon = \infty$, i.e.,	no update, i.e.,	
	$P_a(z) = 0$ for all z.	Rule 2 with $\epsilon = \infty$, i.e.,	
		$P_a(z) = \max\left[1 - \frac{\pi(z)}{q_t(z)}, 0\right]$	

The ARS method in [19] is a special case of ARMS and IA^{2}RMS, so that ARS can be considered also belonging to the new class of techniques

from $p_t(x) \propto \min\{q_t(x), \pi(x)\}$ using the rejection sampling principle [24, 25], i.e., using the following procedure (in order to draw one sample x_a):

1. Draw $x' \sim \widetilde{q}_t(x) \propto q_t(x)$ and $u' \sim \mathcal{U}([0, 1])$.
2. If $u' \leq \frac{\pi(x')}{q_t(x')}$, then set $x_a = x'$.
3. Otherwise, if $u' > \frac{\pi(x')}{q_t(x')}$, repeat from 1.

The accepted sample x_a has pdf $p_t(x) \propto \min\{q_t(x), \pi(x)\}$. Moreover, ARMS adds new points to $\mathcal{S}_t$ using the update Rule 3, only when $q_t(z) \geq \pi(z)$, so that

$$P_a(z) = 1 - \frac{\pi(z)}{q_t(z)}$$

Otherwise, if $q_t(z) < \pi(z)$, ARMS does not add new nodes (see the discussion in [13] about the issues in ARMS mixing). Then, the update rule for ARMS can be written as

$$P_a(z) = \max\left[1 - \frac{\pi(z)}{q_t(z)}, 0\right]. \tag{13}$$

Furthermore, the double update check used in IA^{2}RMS coincides exactly with Rule 3 when

$$p_t(x) \propto \min\{q_t(x), \pi(x)\}$$

is employed as proposal pdf. Finally, note that ARMS and IA^{2}RMS contain ARS in [19] as special case when $q_t(x) \geq \pi(x)$, $\forall x \in \mathcal{X}$ and $\forall t \in \mathbb{N}$. Hence, ARS can be considered also a special case of the new class of algorithms.

6.2 Related algorithms

Other related methods, using non-parametric proposals, can be found in the literature. Samplers for drawing from univariate pdfs, using similar proposal constructions, has been proposed in [20] but the sequence of adaptive proposals does not converge to the target. Interpolation procedures for building the proposal pdf are also employed in [22, 23]. The authors in [22, 23] suggest to build the proposal by b-spline procedures. However, in this case, the resulting proposal is a random walk-type (not independent) and the resulting algorithm is not adaptive. Furthermore, there is not a convergence of the shape of proposal to the shape to target, but only local approximations via b-spline interpolation. The methods [12, 13, 15] are included in the sticky class of algorithms, as pointed out in Section 6.1. In [16], the authors suggest an alternative proposal construction considering pieces of second order polynomial, in order to be used with the ARMS structure [12].

The adaptive rejection sampling (ARS) method [19, 26] is not an MCMC technique, but it is strongly related to the sticky approach, since it also employs an adaptive non-parametric proposal pdf. ARS needs to be able to build a proposal such that $q_t(x) \geq \pi(x)$, $\forall x \in \mathcal{X}$ and $\forall t \in \mathbb{N}$. This is possible only when more requirements about the target are assumed (for instance, log-concavity). For this reason, several extensions of the standard ARS have been also proposed [25, 27, 28], for tackling wider classes of target distributions. In [29], the non-parametric proposal is still adapted by in this case the number of support points remains constant, fixed in advance by the user. Different construction non-parametric procedures in order to address multivariate distributions have been also presented [21, 30, 31].

Other techniques have been developed to be applied specifically for Monte Carlo-within-in-Gibbs scenario when it is possible to draw directly from the full-conditional pdfs. In [32], an importance sampling approximation of the univariate target pdf is employed and a resampling step is performed in order to provide an "approximate" sample from the full-conditional. In [18], the authors suggest a non-adaptive strategy for building a suitable non-parametric proposal via interpolation. In this work, the interpolation procedure is first performed using a huge amount of nodes and then many of them are discarded, according to a suitable criteria. Several other alternatives involving MH-type algorithms have been used for sampling efficiently from the full-conditional pdfs within a Gibbs sampler [5–7, 15, 33–35].

7 Adaptive independent sticky MTM

In this section, we consider an alternative MCMC structure for the second stage described in Section 2: using a multiple-try Metropolis (MTM) approach [36, 37]. The resulting technique, Adaptive Independent Sticky MTM (AISMTM), is an extension of AISM that considers multiple candidates as possible new state, at each iteration. This improves the ability of the chain to explore the state space [37]. At iteration t, AISMTM builds the proposal density $q_t(x|\mathcal{S}_t)$ (step 1 of Table 4) using the current set of support points $\mathcal{S}_t$. Let $x_t = x$ be the current state of the chain and x'_j ($j = 1, \ldots, M$) a set of i.i.d. candidates simulated from $q_t(x|\mathcal{S}_t)$ (see step 2 of Table 4). Note that AISMTM uses an independent proposal [2], just like AISM. As a consequence, the auxiliary points in step 2.3 of Table 4 can be deterministically set ([1], pp. 119-120), [37].

In step 2, a sample x' is selected among the set of candidates $\{x'_1, \ldots, x'_M\}$, with probability proportional to the importance sampling weights,

$$w_t(z) = \frac{\pi(z)}{q_t(z|\mathcal{S}_t)}, \qquad \forall j \in \{1, \ldots, M\}.$$

The selected candidate is then accepted or rejected according to the acceptance probability α given in step 2. Finally, step 3 updates the set $\mathcal{S}_t$,including a new point

$$z' \in \mathcal{Z} = \{z_1, \ldots, z_M\},$$

with probability $P_a(z') = \eta_t(z', d_t(z'))$. Note that $x_t \notin \mathcal{Z}$, and thus AISMTM is an independent MCMC algorithm according to Holden's definition [14]. For the sake of simplicity, we only consider the case where a single point can be added to $\mathcal{S}_t$ at each iteration. However, this update step can be easily extended to allow for more than one sample to be included into the set of support points. Note also that AISMTM becomes AISM for $M = 1$.

AISMTM provides a better choice of the new support points than AISM (see Section 9). The price to pay for this increased efficiency is the higher computational cost per iteration. However, since the proposal quickly approaches the target, it is possible to design strategies with a *decreasing* number of tries ($M_1 \geq M_2 \geq \cdots \geq M_t \geq \cdots \geq M_T$) in order to reduce the computational cost.

Table 4 Adaptive independent sticky Multiple-try Metropolis

For $t = 0, \ldots, T-1$:

1 Construction of the proposal: Build a proposal function $q_t(x|\boldsymbol{\mathcal{S}}_t)$ via a suitable interpolation procedure using the set of support points $\boldsymbol{\mathcal{S}}_t$ (see Section 3).

2 MTM step:

- 2.1 Draw $x'_1, \ldots, x'_M \sim \widetilde{q}_t(x|\boldsymbol{\mathcal{S}}_t) \propto q_t(x|\boldsymbol{\mathcal{S}}_t)$ and compute the weights $w_t(x'_i) = \frac{\pi(x'_i)}{q_t(x'_i|\boldsymbol{\mathcal{S}}_t)}$.
- 2.2 Select $x' = x'_j$ among the M tries with probability proportional to $w_t(x'_i)$, for $i = 1, \ldots, M$.
- 2.3 Set the auxiliary point $x_i^* = x'_i$ and $z_i = x'_i$ for $i \neq j$. Moreover, set $x_j^* = x_{t-1}$.
- 2.4 Set $x_t = x'$ and $z_j = x_{t-1}$ with probability

$$\alpha = \min\left[1, \frac{w_t(x'_1) + \cdots + w_t(x'_M)}{w_t(x_1^*) + \cdots + w_t(x_M^*)}\right].$$

Otherwise, set $x_t = x_{t-1}$ and $z_j = x'$.

3 Test to update $\boldsymbol{\mathcal{S}}_t$: (see Section 7.1) Select a point z' within the set $\{z_1, \ldots, z_M\}$, with probability proportional to some suitable weights $\boldsymbol{\varphi}_t(z_i)$, for $i = 1, \ldots, M$, and set

$$\boldsymbol{\mathcal{S}}_{t+1} = \begin{cases} \boldsymbol{\mathcal{S}}_t \cup \{z'\}, & \text{with prob. } P_a(z) = \eta_t(z', d_t(z')), \\ \boldsymbol{\mathcal{S}}_t, & \text{with prob. } 1 - P_a(z), \end{cases}$$

where $d_t(z) = |\pi(z) - q_t(z|\boldsymbol{\mathcal{S}}_t)|$. For further information see Section 7.1.

7.1 Update rules for AISMTM

The update rules presented above require changes that take into account the multiple samples available, when used in AISMTM. As an example, let us consider the update scheme in Eq. (7). Considering for simplicity that only a single point can be incorporated to $\mathcal{S}_t$, the update step for $\mathcal{S}_t$ can be split in two parts: choose a "bad" point in $\mathcal{Z} \in \{z_1, \ldots, z_M\}$ and then test whether it should be added or not. Thus, first a $z' = z_i$ is selected among the samples in $\mathcal{Z}$ with probability proportional to

$$\begin{aligned} \varphi_t(z_i) &= \max\left\{w_t(z_i), \frac{1}{w_t(z_i)}\right\} \\ &= \frac{\max\{\pi(z_i), q_t(z_i|\mathcal{S}_t)\}}{\min\{\pi(z_i), q_t(z_i|\mathcal{S}_t)\}}, \\ &= \frac{d_t(z_i)}{\min\{\pi(z_i), q_t(z_i|\mathcal{S}_t)\}} + 1, \end{aligned} \tag{14}$$

for $i = 1, \ldots, M$.[8] This step selects (with high probability) a sample where the proposal value is far from the target. Then, the point z' is included in $\mathcal{S}_t$ with probability

$$\begin{aligned} P_a(z') = \eta_t(z', d_t(z')) &= 1 - \frac{1}{\varphi_t(z')}, \\ &= \frac{d_t(z')}{\max\{\pi(z'), q_t(z'|\mathcal{S}_t)\}}, \end{aligned}$$

exactly as in Eq. (7). Therefore, the probability of adding a point z_i to $\mathcal{S}_t$ is

$$\begin{aligned} P_{\mathcal{Z}}(z_i) &= \varphi_t(z_i)\eta_t(z_i, d_t(z_i)), \\ &= \varphi_t(z_i)P_a(z_i) = \frac{\varphi_t(z_i) - 1}{\sum_{j=1}^{M} \varphi_t(z_j)}, \end{aligned}$$

that is a probability mass function defined over $M + 1$ elements: $z_1, \ldots, z_M$ and the event {*no addition*} that, for simplicity, we denote with the empty set symbol $\emptyset$. Thus, the update rule in step 3 of Table 4 can be rewritten as a unique step,

$$\mathcal{S}_{t+1} = \begin{cases} \mathcal{S}_t \cup \{z_1\}, & \text{with prob. } P_{\mathcal{Z}}(z_1) = \frac{\varphi_t(z_1)-1}{\sum_{j=1}^{M}\varphi_t(z_j)}, \\ \quad\vdots & \\ \mathcal{S}_t \cup \{z_M\}, & \text{with prob. } P_{\mathcal{Z}}(z_M) = \frac{\varphi_t(z_M)-1}{\sum_{j=1}^{M}\varphi_t(z_j)}, \\ \mathcal{S}_t, & \text{with prob. } P_{\mathcal{Z}}(\emptyset) = \frac{M}{\sum_{j=1}^{M}\varphi_t(z_j)}, \end{cases} \tag{15}$$

where we have used $1 - \sum_{i=1}^{(r)} P_{\mathcal{Z}}(z_i) = \frac{M}{\sum_{j=1}^{M}\varphi_t(z_j)}$.

8 Range of applicability and multivariate generation

The range of applicability of the sticky MCMC methods is briefly discussed below. On the one hand, sticky MCMC methods can be employed as stand-alone algorithms. Indeed, in many applications, it is necessary to draw samples from complicated univariate target pdf (as example in signal processing, see [38]). In this case, the sticky schemes provide virtually independent samples (i.e., with correlation close to zero) very efficiently. It is also important to remark that AISM and AISMTM also provide automatically an estimation of the normalizing constant of the target (a.k.a. *marginal likelihood* or *Bayesian evidence*) (since, with a suitable choice of the update test, the proposal approaches the target pdf almost everywhere). This is usually a hard task using MCMC methods [1, 2, 11].

AISM and AIMTM can be also applied directly to draw from a multivariate distribution if a suitable construction procedure of the multivariate sticky proposal is designed (e.g, see [30, 31, 39, 40] and ([21], Chapter 11)). However, devising and implementing such procedures in high dimensional state spaces are not easy tasks. Therefore, in this paper, we focus on the use of the sticky schemes within other Monte Carlo techniques (such as Gibbs sampling or the hit and run algorithm) to draw from multivariate densities. More generally, Bayesian inference often requires drawing samples from complicated multivariate posterior pdfs, $\widetilde{\pi}(\mathbf{x}|\mathbf{y})$ with

$$\mathbf{x} = [x_1, \ldots, x_L] \in \mathbb{R}^L, \quad L > 1.$$

For instance, this happens in blind equalization and source separation, or spectral analysis [3, 4]. For simplicity, in the following we denote the target pdf as $\widetilde{\pi}(\mathbf{x})$. When direct sampling from $\widetilde{\pi}(\mathbf{x})$ in the space $\mathbb{R}^L$ is unfeasible, a common approach is the use of *Gibbs-type samplers* [2]. This type of methods split the complex sampling problem into simpler univariate cases. Below we briefly summarize some well-known Gibbs-type algorithms.

Gibbs sampling. Let us denote as $\mathbf{x}^{(0)}$ a randomly chosen starting point. At iteration $k \geq 1$, a Gibbs sampler obtains the ℓ-th component ($\ell = 1, \ldots, L$) of $\mathbf{x}$, x_ℓ, drawing from the full conditional $\widetilde{\pi}_\ell\left(x|\mathbf{x}_{1:\ell-1}^{(k)}, \mathbf{x}_{\ell+1:L}^{(k-1)}\right)$ given all the information available, namely:

1. Draw $x_\ell^{(k)} \sim \widetilde{\pi}_\ell\left(x|\mathbf{x}_{1:\ell-1}^{(k)}, \mathbf{x}_{\ell+1:L}^{(k-1)}\right)$ for $\ell = 1, \ldots, L$.
2. Set $\mathbf{x}^{(k)} = \left[x_1^{(k)}, \ldots, x_L^{(k)}\right]^\top$.

The steps above are repeated for $k = 1, \ldots, N_G$, where N_G is the total number of Gibbs iterations. However, even sampling from $\widetilde{\pi}_\ell$ can often be complicated. In some specific situations, rejection samplers [41–45] and their adaptive versions, adaptive rejection sampling (ARS) algorithms, are employed to generate (one) sample from $\widetilde{\pi}_\ell$ [12, 19, 25, 27–29, 40, 46, 47]. The ARS algorithms are very appealing techniques since they construct a non-parametric proposal in order to mimic the shape of the target pdf, yielding in general excellent performance (i.e., independent samples from $\widetilde{\pi}_\ell$ with an high acceptance rate). However, their range of application is limited to some specific classes of densities [19, 47].

More generally, it is impossible to draw from a full-conditional pdf $\widetilde{\pi}_\ell$ (neither a rejection sampler can be applied), an additional MCMC sampler is required in order to draw from $\widetilde{\pi}_\ell$ [33]. Thus, in many practical scenarios, we have an MCMC (e.g., an MH sampler) inside another MCMC scheme (i.e., the Gibbs sampler). In the so-called *MH-within-Gibbs* approach, only one MH step is often performed within each Gibbs iteration, in order to draw from each complicated full-conditionals. This hybrid approach preserves the ergodicity of the Gibbs sampler and provides good performance in many cases. On the other hand, several authors have noticed that using a single MH step for the internal MCMC is not always the best solution in terms of performance (cf. [48]). Other approximated approaches have been also proposed, considering the application of the importance sampling within the Gibbs sampler [32].

Using a larger number of iterations for the MH algorithm, there is more probability of avoiding the "burn-in" period so that the last sample be distributed as the full-conditional [33–35]. Thus, this case is closer to the ideal situation, i.e., sampling directly from the full-conditional pdf. However, unless the proposal is very well tailored to the target, a properly designed adaptive MCMC algorithm should provide less correlated samples than a standard MH algorithm. Several more sophisticated (adaptive or not) MH schemes for the application "within-Gibbs" have been proposed in literature [12, 13, 16, 18, 20, 23, 49, 50]. In general, these techniques employ a non-parametric proposal pdf in the same fashion of the ARS schemes (and as the sticky MCMC methods). It is important to remark that performing more steps of a standard or adaptive MH within a Gibbs sampler can provide better performance than performing a longer Gibbs chain applying only one MH step (see, e.g., [12, 13, 16, 17]).

Recycling Gibbs sampling. Recently, an alternative Gibbs scheme, called *Recycling Gibbs (RG) sampler*, has been proposed in literature [51]. The combined use of RG with a sticky algorithm is particularly interesting since RG recycles and employs all the samples drawn from each full-conditional pdfs in the final estimators. Clearly, this scheme fits specially well for the use of a adaptive sticky MCMC algorithm where different MCMC steps are performed for each full-conditional pdfs.

Hit and Run. The Gibbs sampler only allows movements along the axes. In certain scenarios, e.g., when the variables x_ℓ are highly correlated, this can be an important limitation that slows down the convergence of the chain to the stationary distribution. The *Hit and Run sampler* is a valid alternative. Starting from $\mathbf{x}^{(0)}$, at the k-th iteration, it applies the following steps:

1. Choose uniformly a direction $\mathbf{d}^{(k)}$ in $\mathbb{R}^L$. For instance, it can be done drawing L samples v_ℓ from a standard Gaussian $\mathcal{N}(0,1)$, and setting
$$\mathbf{d}^{(k)} = \frac{\mathbf{v}}{\sqrt{\mathbf{v}\mathbf{v}^\top}},$$
where $\mathbf{v} = [v_1, \ldots, v_L]$.
2. Set $\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \lambda^{(k)}\mathbf{d}^{(k)}$ where $\lambda^{(k)}$ is drawn from the univariate pdf
$$p(\lambda) \propto \widetilde{\pi}\left(\mathbf{x}^{(k-1)} + \lambda\mathbf{d}^{(k)}\right),$$
where $\widetilde{\pi}\left(\mathbf{x}_\ell^{(k-1)} + \lambda\mathbf{d}^{(k)}\right)$ is a slice of the target pdf along the direction $\mathbf{d}^{(k)}$.

Also in this case, we need to be able to draw from the univariate pdf $p(\lambda)$ using either some direct sampling technique or another Monte Carlo method (e.g., see [50]).

There are several methods similar to the Hit and Run where drawing from a univariate pdf is required; for instance, the most popular one is the *Adaptive Direction Sampling* [52].

Sampling from univariate pdfs is also required inside other types of MCMC methods. For instance, this is the case of *exchange-type MCMC* algorithms [53] for handling models with intractable partition functions. In this case, efficient techniques for generating artificial observations are needed. Techniques which generalize the ARS method, using non-parametric proposals, have been applied for this purpose (see [54]).

9 Numerical simulations

In this section, we provide several numerical results comparing the sticky methods with several well-known MCMC schemes, such as the ARMS technique [12], the adaptive MH method in [10], and the slice sampler [55].[9] The first two experiments (which can be easily reproduced by interested users) correspond to bi-modal one-dimensional and two-dimensional targets, respectively, and are used as benchmarks to compare different variants of the AISM and AISMTM methods with other techniques. They allow us to show the benefits of the non-parametric proposal construction, even in these two simple experiments. Then, in the third example, we approximate the hyper-parameters of a Gaussian process (GP) [56], which is often used for regression purposes in machine learning for signal processing applications.

9.1 Multimodal target distribution

We study the ability of different algorithms to simulate multimodal densities (which are clearly non-log-concave). As an example, we consider a mixture of Gaussians as target density,

$$\widetilde{\pi}(x) = 0.5\mathcal{N}(x; 7, 1) + 0.5\mathcal{N}(x; -7, 0.1),$$

where $\mathcal{N}\left(x; \mu, \sigma^2\right)$ denotes the normal distribution with mean μ and variance σ^2. The two modes are so separated that ordinary MCMC methods fail to visit one of the modes or remains indefinitely trapped in one of them. The goal is to approximate the expected value of the target ($E[X] = 0$ with $X \sim \widetilde{\pi}(x)$) via Monte Carlo. We test the ARMS method [12] and the proposed AISM and AISMTM algorithms. For AISM and AISMTM, we consider different construction procedures for the proposal pdf:

- **P1**: the construction given in [12] formed by exponential pieces, specifically designed for ARMS.
- **P2**: alternative construction formed by exponential pieces obtained by a linear interpolation in the log-pdf domain, given in [13].
- **P3**: the construction using uniform pieces in Eq. (3).
- **P4**: the construction using linear pieces in Eq. (4).

Furthermore, for AISM and AISMTM, we consider the Update Rule 1 (R1) with different values of the parameter β, the Update Rule 2 (R2) with different value of the parameter ε, and the Update Rule 3 (R3) for the inclusion of a new node in the set $\mathcal{S}_t$ (see Section 4). More specifically, we first test AISM and AISMTM with all the construction procedures P1, P2, P3, and P4 jointly with the rule R3. Then, we test AISM with the construction P4 and the update test R2 with $\varepsilon \in \{0.005, 0.01, 0.1, 0.2\}$. For Rule 1 we consider $\beta \in \{0.3, 0.5, 0.7, 2, 3, 4\}$. All the algorithms start with $\mathcal{S}_0 = \{-10, -8, 5, 10\}$ and initial state $x_0 = -6.6$. For AISMTM, we have set $M \in \{10, 50\}$. For each independent run, we perform $T = 5000$ iterations of the chain.

The results given in Table 5 are the averages over 2000 runs, without removing any sample to account for the initial burn-in period. Table 5 shows the Mean Square Error

(MSE) in the estimation $E[X]$, the auto-correlation function $\rho(\tau)$ at different lags, $\tau \in \{1, 10, 50\}$ (normalized, i.e., $\rho(0) = 1$), the approximated effective sample size (ESS) of the produced chain ([57], Chapter 4)

$$ESS \approx \frac{T}{1 + 2\sum_{\tau=1}^{\infty} \rho(\tau)}, \tag{16}$$

(clearly, $ESS \leq T$), the final number of support points m_T and the computing time normalized with respect to the time spent by ARMS [12]. For simplicity, in Table 5, we have reported only the case of R2 with $\varepsilon \in \{0.005, 0.01\}$; however, other results are shown in Fig. 5.

AISM and AIMTM outperform ARMS, providing a smaller MSE and correlation (both close to zero). This is because ARMS does not allow a complete adaptation of the proposal pdf as highlighted in [13]. The adaptation in AISM and AIMTM provides a better approximation of the target than ARMS, as also indicated by the ESS which is substantially higher in the proposed methods. ARMS is in general slower than AISM for two main reasons. Firstly, the construction P1 (used by ARMS) is more costly since it requires the computation of several intersection points [12]. It is not required for the procedures P2, P3, and P4. Secondly, the effective number of iterations in ARMS is higher than $T = 5000$ (the averaged value is ≈ 5057.83) due to the discarded samples in the rejection step (in this case, the chain is not moved forward).

Figure 6a–d depicts the averaged autocorrelation function $\rho(\tau)$ for $\tau = 1, \ldots, 100$ for the different techniques and constructions. Figure 6e–h shows the average acceptance probability (AAP; the value of α of the MH-type techniques) of accepting a new state as function of the iterations t. We can see that, with AISM and AIMTM, AAP approaches 1 since q_t becomes closer and closer to π. Figure 7 shows the evolutions of the number of support points, m_t, as function of $t = 1, \ldots, T = 5000$, again for the different techniques and constructions. Note that, with AIMTM and P3-P4, AAP approaches 1 so quickly and the correlation is so small (virtually zero) that it is difficult to recognize the corresponding curves which are almost constant close to one or zero, respectively. The constructions P3 and P4 provide the better results. In this experiment, P4 seems to provide the best compromise between performance and computational cost. We also test AISM with update R2 for different values of ε (and different constructions). The number of nodes m_t and AAP as function of t for these cases are shown in Fig. 5. These figures and the results given in Table 5 show that AISM-P4-R2 provides extremely good performance with a small computational cost (e.g., the final number of

Table 5 (Ex-Sect-9.1). For each algorithm, the table shows the mean square error (MSE), the autocorrelation ($\rho(\tau)$) at different lags, the effective sample size (ESS), the final number of support points (m_T), the computing times normalized w.r.t. ARMS (Time)

Algorithm	MSE	$\rho(1)$	$\rho(10)$	$\rho(50)$	ESS	m_T	Time
ARMS [12]	10.04	0.4076	0.3250	0.2328	89.12	118.19	1.00
AISM-P1-R3	3.0277	0.1284	0.1099	0.0934	235.76	152.63	1.23
AISM-P2-R3	2.9952	0.1306	0.1125	0.0929	235.01	71.14	0.27
AISM-P3-R3	0.0290	0.0535	0.0165	0.0077	609.05	279.65	0.65
AISM-P4-R3	0.0354	0.0354	0.0195	0.0086	608.76	84.87	0.33
AISMTM-P1 ($M = 10$)	0.6720	0.0726	0.0696	0.0624	336.84	159.01	2.35
R3 ($M = 50$)	0.1666	0.0430	0.0395	0.0316	617.10	160.75	5.45
AISMTM-P2 ($M = 10$)	0.5632	0.0588	0.0525	0.0443	440.23	72.16	1.13
R3 ($M = 50$)	0.1156	0.0345	0.0303	0.0231	746.45	72.53	4.38
AISMTM-P3 ($M = 10$)	0.0105	0.0045	0.0001	0.0001	4468.10	315.78	2.60
R3 ($M = 50$)	0.0099	0.0041	0.0001	0.0001	4843.81	360.73	10.59
AISMTM-P4 ($M = 10$)	0.0108	0.0036	0.0011	0.0014	3678.79	92.67	1.86
R3 ($M = 50$)	0.0098	0.0001	0.0001	0.0001	4912.07	101.78	7.25
AISM-P4-R2 ($\varepsilon = 0.01$)	0.0412	0.0407	0.0213	0.0074	604.95	35.01	0.11
($\varepsilon = 0.005$)	0.0321	0.0360	0.0181	0.0072	610.01	43.32	0.20
AISM-P4-R1 ($\beta = 0.3$)	0.1663	0.2710	0.1368	0.0593	216.75	25.56	0.08
($\beta = 0.7$)	0.1046	0.1781	0.0866	0.0441	356.21	33.55	0.11
($\beta = 2$)	0.0824	0.0947	0.0408	0.0204	677.73	46.81	0.21
($\beta = 3$)	0.0371	0.0720	0.0281	0.0099	714.90	52.76	0.23
($\beta = 4$)	0.0310	0.0621	0.0253	0.0096	802.18	58.66	0.24

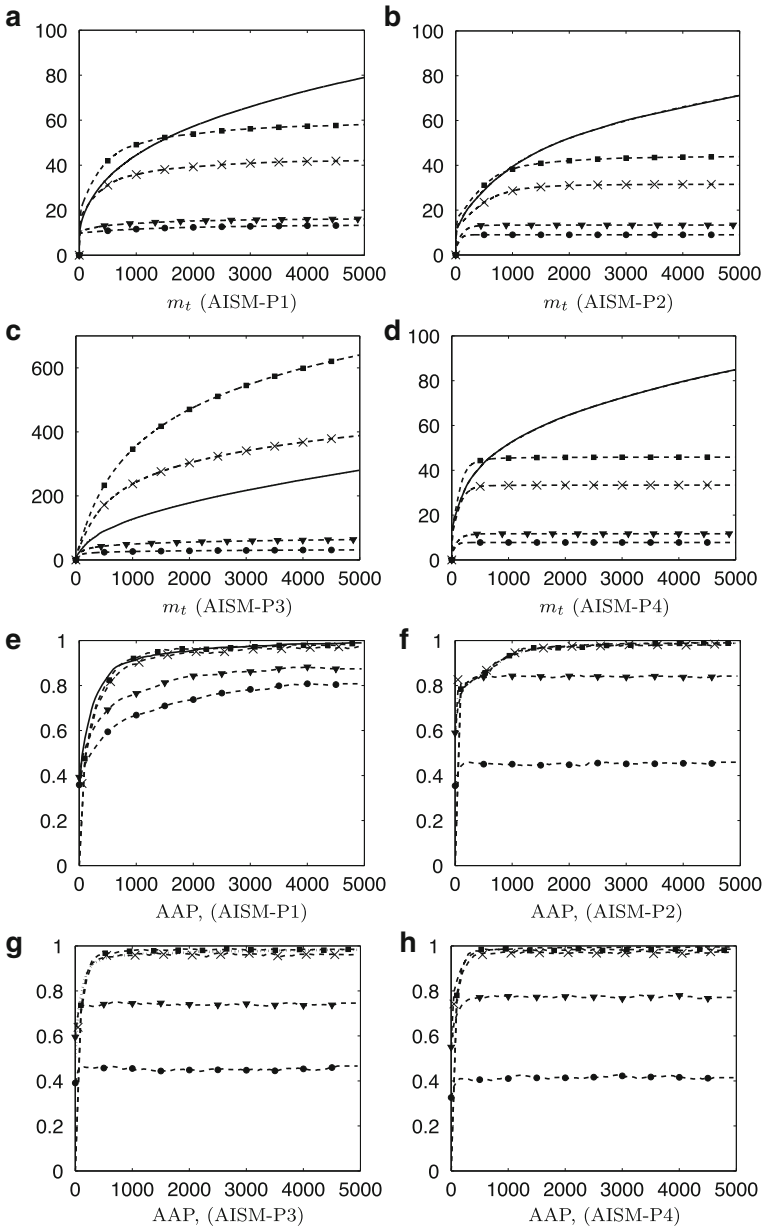


Fig. 5 (Ex-Sect-9.1). Evolution of the number of support points m_t and average acceptance probability (AAP), as function of $t = 1, \ldots, T$ for AISM, for different constructions, and update rule R2 with $\varepsilon = 0.005$ (square), $\varepsilon = 0.01$ (cross), $\varepsilon = 0.1$ (triangle) and $\varepsilon = 0.2$ (circle). Moreover, in **a**–**d** the evolution of m_t of AISM with the update rule R3 is also shown with solid line. Note that the range of values in **a**–**d** is different. (**e**)-(**f**)-(**g**)-(**h**) Acceptance Rate as function of the iteration t

points is only $m_T \approx 43$ with $\epsilon = 0.005$). This shows that the update rule R2 is a very promising choice given the obtained results. Moreover, we can observe that the update rule R1 is very parsimonious in adding new points even considering a great range of values of β, from 0.3 to 4. The results are good also in this case with R1, so that this rule seems to be a more robust interesting alternative to R2 (which seems more dependent on the choice of β). Finally, Fig. 8 shows the histograms of the 5000 samples obtained by one run of AISM-P3-R1 with $\beta = 0.1$ and $\beta = 3$. The target pdf is depicted in solid line and the final construction proposal pdf is shown in dashed line.

9.2 Missing mode experiment

Let us consider again the previous bimodal target pdf,

$$\widetilde{\pi}(x) = 0.5\mathcal{N}(x; 7, 1) + 0.5\mathcal{N}(x; -7, 0.1),$$

shown in Fig. 8. Here, we consider a bad choice of the initial support points, such as $\mathcal{S}_0 = \{5, 6, 10\}$ cutting out

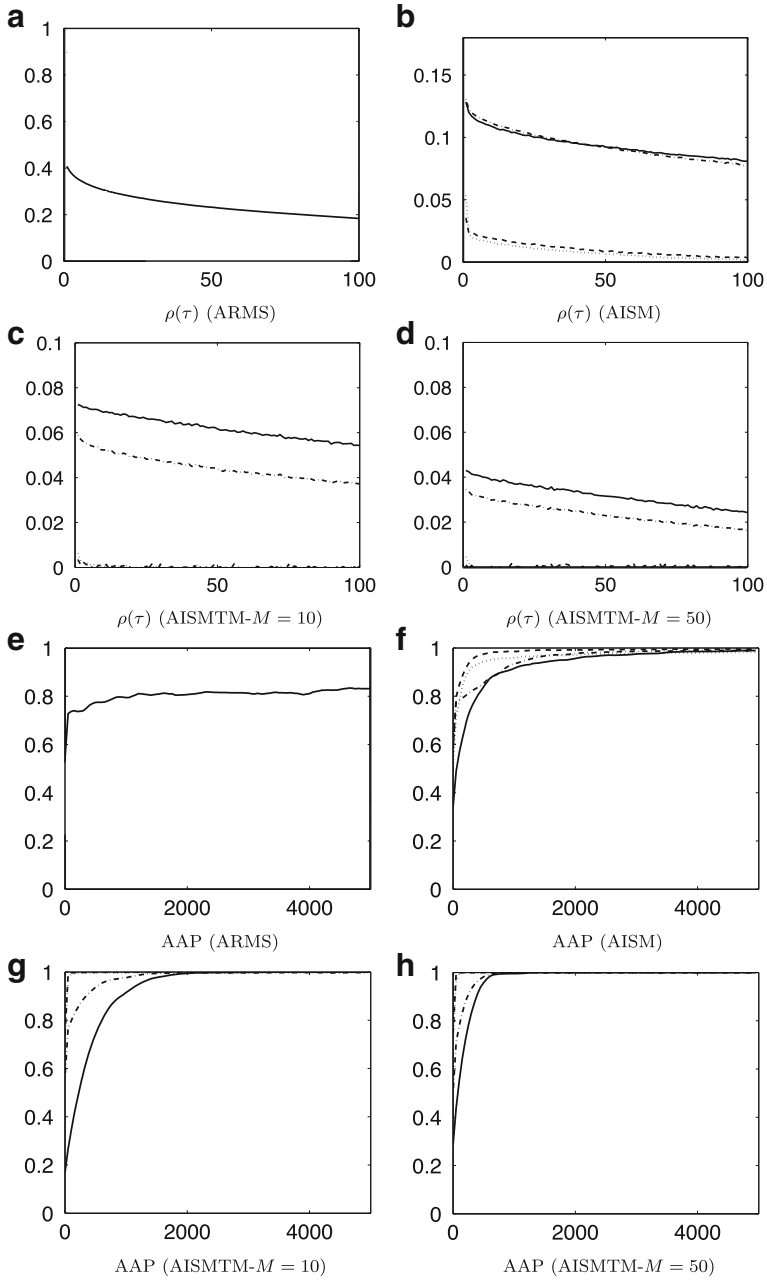


Fig. 6 (Ex-Sect-9.1). (**a**)-(**b**)-(**c**)-(**d**) Autocorrelation Function $\rho(\tau)$ at lags from 1 to 100 and (**e**)-(**f**)-(**g**)-(**h**) Averaged Acceptance Probability (AAP) as function of t, for the different methods. In each plot: P1 (solid line), P2 (dashed-dotted line), P3 (dotted line), and P4 (dashed line). Note the different range of values of $\rho(\tau)$

one of the two modes (we consider that no information about the range of the target pdf is provided). We test the robust implementation described in Appendix E.1, i.e., we employ the proposal density defined

$$\widetilde{q}(x) = \alpha_t \widetilde{q}_1(x) + (1-\alpha_t)\widetilde{q}_2(x|\mathcal{S}_t), \tag{17}$$

where $\widetilde{q}_1(x) = \mathcal{N}\left(x; 0, \sigma_p^2\right)$ and $\widetilde{q}_2(x|\mathcal{S}_t)$ is a sticky proposal constructed using the procedure P3 in Eq. (3) (we use the update rule 1 with $\beta = 0.1$). We consider the most defensive strategy defining $\alpha_t = \alpha_0 = 0.5$ for all t. We test $\sigma_p \in \{2, 3, 8, 10\}$. We compute the mean absolute error (MAE), in estimating the variance $\mathrm{Var}[X] = 49.55$ where $X \sim \widetilde{\pi}(x)$, with different MCMC methods generating chains of length $T = 10^4$. We compare this Robust AISM-P3-R1 scheme with a standard MH method using $\widetilde{q}_1(x)$ as proposal pdf and the Adaptive MH technique where the scale parameter $\sigma_p^{(t)}$ is adapted online [10] (starting with $\sigma_p^{(0)} \in \{2, 3, 8, 10\}$). The

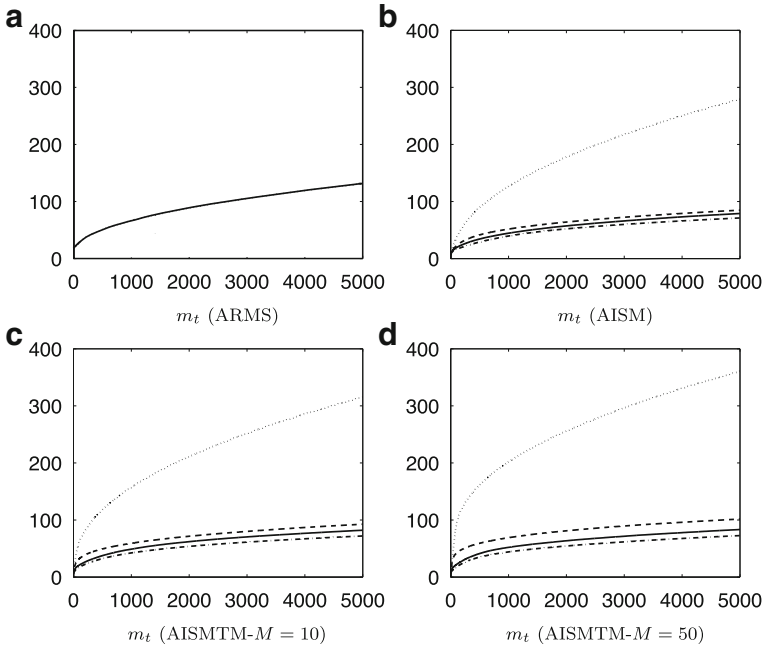


Fig. 7 (Ex-Sect-9.1). (**a**)-(**b**)-(**c**)-(**d**) Evolution of the number of support points m_t as function of $t = 1, \ldots, T$, for the different methods. In each plot: construction P1 (solid line), P2 (dashed-dotted line), P3 (dotted line) and P4 (dashed line)

results, averaged over 10^3 independent runs, are given in Table 6.

9.3 Heavy-tailed target distribution

In this section, we test the AISM method from drawing with a target heavy tails. We show that the sticky MCMC schemes can be applied in this scenario, even by using a proposal pdf with exponential (i.e., "light") tails. However, we recall that an alternative construction of the tails is always possible, as suggested in Appendix E.2 using Pareto tails, for instance. More specifically, we consider the Lévy density, i.e.

$$\bar{\pi}(x) \propto \pi(x) = \frac{1}{(x-\lambda)^{3/2}} \exp\left(-\frac{\nu}{2(x-\lambda)}\right), \qquad (18)$$

$\forall x \geq \lambda$. Given a random variable $X \sim \bar{\pi}(x)$, we have that $E[X] = \infty$ and $\mathrm{Var}[X] = \infty$ due to the heavy-tail of the Lévy distribution. However, the normalizing constant, $\frac{1}{c_\pi}$, such that $\bar{\pi}(x) = \frac{1}{c_\pi}\pi(x)$ integrates to one, can be determined analytically, and is given by $\frac{1}{c_\pi} = \sqrt{\frac{\nu}{2\pi}}$.

Our goal is estimating the normalizing constant $\frac{1}{c_\pi}$ via Monte Carlo simulation, when $\lambda = 0$ and $\nu = 2$. In general, it is difficult to estimate a normalizing constant using MCMC outputs [2, 58, 59]. However, in the sticky MCMC algorithms (with update rules as R1 and R3 in Table 2), the normalizing constant of the adaptive non-parametric proposal approaches the normalizing constant of the target. We compare AISM-P4-R3 and different Multiple-try

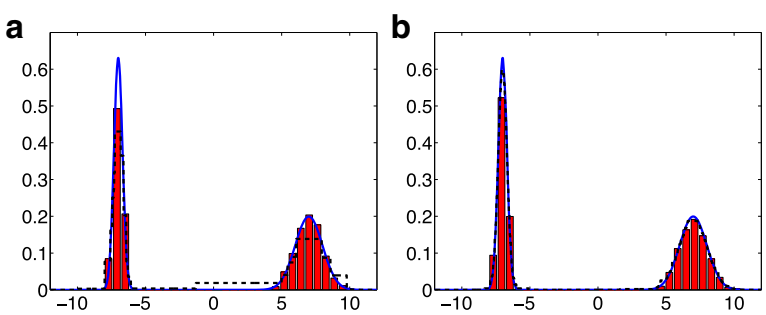


Fig. 8 (Ex-Sect-9.1). **a** Histogram of the 5000 samples obtained by one run of AISM-P3-R1 with $\beta = 0.1$ (28 final points). **b** Histogram of the 5000 samples obtained by one run of AISM-P3-R1 with $\beta = 3$ (79 final points). The target pdf, $\widetilde{\pi}(x)$, is depicted in solid line and the final construction proposal pdf, $\widetilde{q}_T(x)$, is shown in dashed line

Table 6 (Ex-Sect-9.2). Mean absolute error (MAE) in the estimation of the $var[X] = 49.55$, for different techniques and different scale parameters σ_p ($T = 10^4$)

Algorithm	$\sigma_p = 2$	$\sigma_p = 3$	$\sigma_p = 8$	$\sigma_p = 10$
Standard MH	13.51	0.94	0.27	0.35
Adaptive MH	3.28	0.29	0.24	0.28
Robust AISM	1.79	0.16	0.13	0.14

Metropolis (MTM) schemes. For the MTM schemes, we use the following procedure: given the MTM outputs obtained in one run, we use these samples as nodes, then construct the approximated function using the construction P4 (considering these nodes), and finally compute the normalizing constant of this approximated function. Note that we use the same construction procedure P4, for a fair comparison.

For AISM, we start with only $m_0 = 3$ support points, $\mathcal{S}_0 = \{s_1 = 0, s_2, s_3\}$, where two nodes are randomly chosen at each run, i.e., $s_2, s_3 \sim \mathcal{U}([1, 10])$ with $s_2 < s_3$. We also test three different MTM techniques, two of them using an independent proposal pdf (MTM-ind) and the last one a random walk proposal pdf (MTM-rw). For the MTM schemes, we set $M = 1000$ tries and importance weights designed again to choose the best candidate in each step [37]. We set $T = 5000$ for all the methods. Note that, the total number of target valuation E of AISM is only $E = T = 5000$ whereas we $E = MT = 5 \cdot 10^6$ for the MTM-ind schemes and $E = 2MT = 10^7$ for the MTM-rw algorithm (see [37] for further details). For the MTM-ind methods, we use an independent proposal $\widetilde{q}(x) \propto \exp(-(x-\mu)^2/(2\sigma^2))$ with $\mu \in \{10, 100\}$ and $\sigma^2 = 2500$. In MTM-rw, we have a random walk proposal $\widetilde{q}(x|x_{t-1}) \propto \exp\left(-(x - x_{t-1})^2 \Big/ \left(2\sigma^2\right)\right)$ with $\sigma^2 = 2500$. Note that we need to choose huge values of σ^2 due to the heavy-tailed feature of the target.

The results, averaged over 2000 runs, are summarized in Table 7. Note that the real value of $\frac{1}{c_p}$ when $\nu = 2$ is $\frac{1}{\sqrt{\pi}} = 0.5642$. The AISM-P4-R3 provides better results than all of the MTM approaches tested with only a fraction of their computational cost. Furthermore, AISM-P4-R3 avoids the critical issue of parameter selection (selecting a small value of σ^2 in this case can easily lead to very poor performance).

Table 7 Estimation of the normalizing constant $\frac{1}{c_\pi} = 0.5642$ for the Lévy distribution ($T = 5000$)

Technique	MSE	Target evaluation
AISM-P4-R3	0.0015	$E = T = 5000$
MTM-ind	0.0028	$E = MT = 5 \cdot 10^6$
	0.0024	
MTM-rw	0.0054	$E = 2MT = 10^7$

9.4 Sticky MCMC methods within Gibbs sampling

9.4.1 Example 1: comparing different MCMC-within-Gibbs schemes

In this example we show that, even in a simple bivariate scenario, AISM schemes can be useful within a Gibbs sampler. Let us consider the bimodal target density

$$\widetilde{\pi}(x_1, x_2) \propto \exp\left(-\frac{(x_1^2 - A + Bx_2)^2}{4} - \frac{x_1^2}{2\sigma_1^2} - \frac{x_2^2}{2\sigma_2^2}\right),$$

with $A = 16$, $B = 10^{-2}$, and $\sigma_1^2 = \sigma_2^2 = \frac{10^4}{2}$. Densities with this non-linear analytic form have been used in the literature (cf. [10]) to compare the performance of different Monte Carlo algorithms. We apply N_G steps of a Gibbs sampler to draw from $\widetilde{\pi}(x_1, x_2)$, using ARMS [12], AISM-P4-R3, and AISMTM-P4-R3 within of the Gibbs sampler to generate samples from the full-conditionals, starting always with the initial support set $\mathcal{S}_0 = \{-10, -6, -4.3, 0, 3.2, 3.8, 4.3, 7, 10\}$. From each full-conditional pdf, we draw T samples and take the last one as the output from the Gibbs sampler. We also apply a standard MH algorithm with a random walk proposal $q\left(x_{\ell,t}|x_{\ell,t-1}\right) \propto \exp\left((x_{\ell,t} - x_{\ell,t-1})^2 \Big/ \left(2\sigma_p^2\right)\right)$ for $\ell \in \{1, 2\}$, $\sigma_p \in \{1, 2, 10\}$, $1 \le t \le T$. Furthermore, we test an adaptive parametric approach (as suggested in [8]). Specifically, we apply the adaptive MH method in [10] where the scale parameter of $q(x_{\ell,t}|x_{\ell,t-1})$ is adapted online, i.e., $\sigma_{p,t}$ varies with t (we set $\sigma_{p,0} = 3$). We also consider the application of the slice sampler [55] and the Hamiltonian Monte Carlo (HMC) method [60]. For the standard MH and the slice samplers we have used the function `mhsample.m` and `slicesample.m` directly provided by MATLAB. For HMC, we consider the code provided in [61] with $\epsilon_d = 0.01$ as discretization parameter and $L = 1$ as length of the trajectory.[10] We recall that a preliminary code of AISM is also available at `Matlab-FileExchange` webpage.

We consider two initializations for all the methods-within-Gibbs: (**In1**) $x_{\ell,0}^{(k)} = 1$; (**In2**) $x_{\ell,0}^{(k)} = 1$ and $x_{\ell,0}^{(k)} = x_{\ell,T}^{(k-1)}$ for $k = 1, \ldots, N_G$. We use all the samples to estimate four statistics that involve the first four moments of the target: mean, variance, skewness, and kurtosis. Table 8 provides the mean absolute error (MAE; averaged over 500 independent runs) for each of the four statistics estimated, and the time required by the Gibbs sampler (normalized by considering 1.0 to be the time required by ARMS with $T = 50$).

The results are provided in Table 8. First of all, we notice that AISM outperforms ARMS and the slice sampler for

Table 8 (Ex-Sect-9.4.1). Mean absolute error (MAE) in the estimation of four statistics (first component) and normalized computing time

Technique	T	N_G	Init.	MAE				Avg. MAE	Time
				Mean	Variance	Skewness	Kurtosis		
Panel I									
AISM-P4	3	2000	In1	0.878	0.781	0.437	0.223	0.579	0.066
	5			0.749	0.576	0.389	0.160	0.468	0.098
	10			0.266	0.057	0.136	0.020	0.120	0.178
	50			0.101	0.041	0.051	*0.003*	0.049	0.741
AISMTM-P4	3	2000	In1	0.251	0.056	0.128	0.017	0.113	0.202
($M = 5$)	10			0.096	*0.031*	0.048	*0.003*	0.044	0.642
ARMS	3	2000	In1	3.408	11.580	3.384	11.572	7.486	0.077
	5			3.151	9.839	2.650	7.079	5.679	0.116
	10			2.798	7.665	2.024	4.124	4.152	0.223
	50			1.918	3.407	1.134	1.292	1.937	1.000
MH ($\sigma_p = 1$)	100	2000	In1	3.509	12.308	3.671	13.666	8.288	0.602
MH ($\sigma_p = 2$)				1.756	3.077	0.978	0.963	1.693	0.602
MH ($\sigma_p = 10$)				*0.075*	0.037	*0.036*	*0.002*	*0.038*	0.602
MH ($\sigma_p = 1$)	1000	2000	In1	3.508	12.302	3.665	13.624	8.274	4.052
MH ($\sigma_p = 2$)				1.601	2.560	0.874	0.769	1.451	4.052
MH ($\sigma_p = 10$)				*0.074*	0.036	*0.036*	*0.002*	*0.037*	4.052
MH ($\sigma_p = 10$)	1	2000	In1	0.697	11.598	0.883	3.622	4.200	*0.033*
		10000		0.493	9.881	0.611	2.905	3.472	0.162
	3	2000		0.352	6.510	0.290	0.927	2.019	0.042
	10			0.085	1.411	0.043	0.160	0.424	0.081
Adaptive MH	100	2000	In1	0.415	0.304	0.234	0.068	0.255	0.634
	1000			*0.075*	0.038	0.037	*0.002*	*0.038*	4.107
HMC	10	2000	In1	0.091	1.509	0.042	0.123	0.441	0.092
	100			0.078	0.037	0.039	*0.002*	0.039	0.630
Slice	3	2000	In1	0.810	1.174	0.415	0.231	0.658	0.156
	10			0.607	0.372	0.306	0.096	0.345	0.463
	50			0.156	0.043	0.077	0.007	0.071	2.311
Panel II									
AISM-P4	3	2000	In2	0.138	0.055	0.070	0.006	0.067	0.066
	5			0.112	0.050	0.057	0.004	0.056	0.098
	10			0.093	0.045	0.046	*0.002*	0.046	0.178
	3	10000		0.095	*0.023*	0.050	*0.002*	0.042	0.335
AISMTM-P4	3	2000	In2	0.085	0.036	0.043	*0.002*	0.042	0.202
($M = 5$)		4000		0.083	*0.028*	0.042	*0.002*	*0.038*	0.400
($M = 10$)		2000		*0.073*	0.031	*0.036*	*0.002*	*0.035*	0.316

Table 8 (Ex-Sect-9.4.1). Mean absolute error (MAE) in the estimation of four statistics (first component) and normalized computing time (*Continued*)

Technique	T	N_G	Init.	MAE				Avg. MAE	Time
				Mean	Variance	Skewness	Kurtosis		
MH ($\sigma_p = 10$)		10000	In2	0.178	0.126	0.091	0.012	0.102	0.162
	1	20000		0.151	0.112	0.090	0.008	0.090	0.331
		30000		0.138	0.063	0.068	0.007	0.069	0.492
	2	10000		0.130	0.062	0.066	0.006	0.066	0.196
	3			0.125	0.066	0.063	0.006	0.065	0.223
	10	2000		0.149	0.083	0.075	0.009	0.079	0.081
Adaptive MH	10	2000	In2	0.158	0.082	0.087	0.012	0.084	0.090
	100			0.146	0.076	0.073	0.010	0.076	0.634
HMC	10	2000	In2	0.152	0.092	0.079	0.015	0.084	0.092
	100			0.148	0.081	0.070	0.012	0.077	0.630
Slice	3	2000		0.204	0.105	0.103	0.022	0.108	0.156
	10		In2	0.188	0.091	0.095	0.018	0.098	0.463
	3	10000		0.132	0.051	0.066	0.007	0.064	0.783

All the techniques are used within a Gibbs sampler: N_G is the number of iterations of the Gibbs sampler whereas T is is the number of iterations of the technique within Gibbs (so that $T \times N_G$ is the global number of MCMC iterations). The best results (in each column, and in each panel) are highlighted with italics

all values of T and N_G, in terms of performance and computational time. Regarding the use of the MH algorithm within Gibbs, the results depend largely on the choice of the variance of the proposal, σ_p^2, and the initialization, showing the need for adaptive MCMC strategies. For a fixed value of $T \times N_G$, the AISM schemes provide results close to the smallest averaged MAE for **In1** and the best results for **In2** with a slight increase in the computing time, w.r.t. the standard MH algorithm. Finally, Table 8 shows the advantage of the non-parametric adaptive independent sticky approach w.r.t. the parametric adaptive approach [8, 10].

9.4.2 Example 2: comparison with an ideal Gibbs sampler

The ideal scenario for the Gibbs sampling scheme is that we are able to draw samples from the full-conditional pdfs (using a transformation or a direct method). In this section, we compare the performance of MH and AISM-within-Gibbs schemes with the ideal case. Let us consider two Gaussian full-conditional densities,

$$\widetilde{\pi}_1(x_1|x_2) \propto \exp\left(-\frac{(x_1 - 0.5x_2)^2}{2\xi_1^2}\right), \tag{19}$$

$$\widetilde{\pi}_2(x_2|x_1) \propto \exp\left(-\frac{(x_2 - 0.5x_1)^2}{2\xi_2^2}\right), \tag{20}$$

with $\xi_1 = 1$ and $\xi_2 = 0.2$. The joint pdf is a bivariate Gaussian pdf with mean vector $\boldsymbol{\mu} = [0, 0]^\top$ and covariance matrix $\boldsymbol{\Sigma} = [1.08\ 0.54;\ 0.54\ 0.31]$. We apply a Gibbs sampler with N_G iterations to estimate both the mean and the covariance of the joint pdf. Then, we calculate the average MSE in the estimation of all the elements in $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, averaged over 2000 independent runs. We use this simple case, where we can draw directly from the full-conditionals, to check the performance of MH and AISM-P3-R3 within Gibbs as a function of T and N_G. For the MH scheme, we use a Gaussian random walk proposal, $\widetilde{q}\left(x_{\ell,t}^{(k)} \left| x_{\ell,t-1}^{(k)}\right.\right) \propto \exp\left(-\left(x_{\ell,t}^{(k)} - 0.5x_{\ell,t-1}^{(k)}\right)^2 \Big/ \left(2\sigma_p^2\right)\right)$ for $\ell \in \{1, 2\}$, $1 \leq t \leq T$ and $1 \leq k \leq N_G$. For AISM-P3-R3, we start with $\mathcal{S}_0 = \{-2, 0, 2\}$.

We set $N_G = 10^3$ and $x_{\ell,0}^{(i)} = 1$ (both for MH and AISM-P3-R3), and increase the value of T. The results can be seen in Fig. 9. AISM-within-Gibbs easily reaches the same performance as the ideal case (sampling directly from the full conditionals) even for small values of T, whereas the MH-within-Gibbs needs a substantially larger value of T (up to $T = 500$ for $\sigma_p = 0.1$) to attain a similar performance. Note the importance of using a proper parameter σ_p for attaining good performance. This observation shows the importance of employing an adaptive technique within-Gibbs.

9.5 Sticky MCMC methods within Recycling Gibbs sampling

In this section, we test the sticky MCMC methods within the Recycling Gibbs (RG) sampling scheme where the intermediate samples drawn from each full-conditional pdf are sued in the final estimator [51]. We consider a simple numerical simulation (easily reproducible by any practitioner) involving a bi-dimensional target pdf

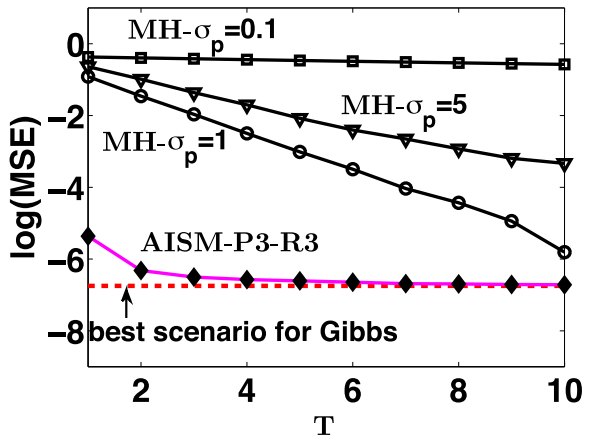


Fig. 9 MSE as function of the number of iterations in the internal chain (T), and $N_G = 1000$. The constant dashed line is the MSE (≈ 0.0012) obtained drawing directly from the full-conditionals (ideal Gibbs scenario drawing directly from the full-conditional pdfs)

$$\widetilde{\pi}(x_1, x_2) \propto \exp\left(-\frac{\left(x_1^2 - \mu_1\right)^2}{2\delta_1^2} - \frac{(x_2 - \mu_2)^2}{2\delta_2^2}\right),$$

where $\mu_1 = 4$, $\mu_2 = 1$, $\delta_1 = \sqrt{\frac{5}{2}}$ and $\delta_2 = 1$. Note that $\widetilde{\pi}(x_1, x_2)$ is bimodal and is not Gaussian. The goal is to approximate via Monte Carlo the expected value, $\mathbb{E}[\mathbf{X}]$ where $\mathbf{X} = [X_1, X_2] \sim \widetilde{\pi}(x_1, x_2)$.

We test different Gibbs techniques: the MH [2] and AISM-P3-R3 algorithm (with update rule 3 and proposal construction in Eq. (3)), within the Standard Gibbs (SG) and within the RG sampling schemes. For the MH method, we use a Gaussian random walk proposal,

$$q\left(x_{\ell,t}^{(k)} \middle| x_{\ell,t-1}^{(k)}\right) \propto \exp\left(-\frac{\left(x_{\ell,t}^{(k)} - x_{\ell,t-1}^{(k)}\right)^2}{2\sigma^2}\right),$$

for $\sigma > 0$, $\ell \in \{1, 2\}$, $1 \leq k \leq N_G$ and $1 \leq t \leq T$. We set $x_{\ell,0}^{(k)} = 1$ and $x_{\ell,0}^{(k)} = x_{\ell,T}^{(k-1)}$ for $k = 1, \ldots, N_G$, for all schemes.

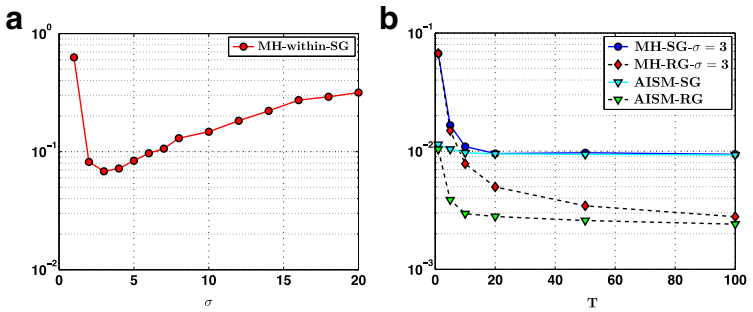
9.5.1 Optimal scale parameter for MH

First of all, we obtain the MSE in estimation of $E[\mathbf{X}]$ for different values of the σ parameter for MH-within-SG (with $T = 1$ and $N_G = 1000$). Figure 10a shows the results averaged over 10^5 independent runs. The performance of the Standard Gibbs (SG) sampler depends strongly on the choice of σ of the *internal* MH method. We can observe that there exists an optimal value $\sigma^* \approx 3$. This shows the need of using an adaptive scheme for drawing from the full-conditional pdfs. In the following, we compare the performance of AISM with the performance of this *optimized* MH using the optimal scale parameter $\sigma^* = 3$, in order to show the capability of the non-parametric adaptation employed in AISM, with respect to a standard adaptation procedure [10].

9.5.2 Comparison among different schemes

For AISM-P3-R3, we start with the set of support points $\mathcal{S}_0 = \{-10, -6, -2, 2, 6, 10\}$. We have averaged the MSE values over 10^5 independent runs for each Gibbs scheme.

In Fig. 10b (represented in log-scale), we fix $N_G = 1000$ and vary T. As T grows, when a standard Gibbs (SG) sampler is used, the curves show an horizontal asymptote since the internal chains converge after some value $T \geq T^*$. Considering an RG scheme, the increase of T yield lower MSE since now we recycle the internal samples. Figure 10b shows the advantage of using AISM-R3-P3 even when compared with the optimized MH method. The advantage of AISM-R3-P3 is clearer with small T values ($10 < T < 30$; recall that in this experiment $N_G = 1000$ is kept fixed). The performance of AISM-R3-P3 and optimized MH (within Gibbs) becomes more similar as T increases. This is due to the fact that,



MSE as function of σ. MSE as function of T ($N_G = 1000$).

Fig. 10 (Ex-Sect-9.5). **a** MSE (log-scale) as function of σ for MH-within-SG ($T = 1$ and $N_G = 1000$). **b** MSE (log-scale) as function of T for different MCMC-within-Gibbs schemes (we keep fixed $N_G = 1000$). Note the MH is using the optimal scale value $\sigma^* = 3$ for the (Gaussian) parametric proposal density

in this case, with a high enough value of T, the MH chain is able to exceed its burn-in period and eventually converges.

9.6 Tuning of the hyper-parameters of a Gaussian process (GP)

9.6.1 Exponential Power kernel function

Let assume to observe the pairs of data $\{y_j, \mathbf{z}_j\}_{j=1}^{P}$, with $y_j \in \mathbb{R}$ and $\mathbf{z}_j \in \mathbb{R}^{d_Z}$, and denote the corresponding vectors $\mathbf{y} = \left[y_1, \ldots, y_P\right]$ and $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_P]$. We address the regression problem of inferring the hidden function $y = f(z)$, linking the variable y and z. For this goal, we assume the model

$$y = f(\mathbf{z}) + e, \tag{21}$$

where $e \sim N\left(e; 0, \sigma^2\right)$. For simplicity, we set $d_Z = 1$. We consider the f is a Gaussian process (GP) [56], i.e., we assume a GP prior over f, so $f \sim \mathrm{GP}(\mu(z), \kappa(z, r))$ where $\mu(z) = 0$, and the kernel function is

$$\kappa(z, r) = \exp\left(-\frac{|z - r|^{\beta}}{2\delta^2}\right), \qquad \beta, \delta > 0. \tag{22}$$

Therefore, the vector $\mathbf{f} = \left[f(z_1), \ldots, f(z_P)\right]$ is distributed as $p(\mathbf{f}|\mathbf{Z}, \kappa, \beta, \delta) = \mathcal{N}(\mathbf{f}; \mathbf{0}, \mathbf{K})$ where $\mathbf{0}$ is a $1 \times P$ vector, $\mathbf{K} := \kappa(z_i, z_j)$, for all $i, j = 1, \ldots, P$ is a $P \times P$ matrix, and we have expressed explicitly the dependence on the choice of the kernel family κ in Eq. (22). Moreover, we denote the hyper-parameters of the model as $\boldsymbol{\theta} = [\theta_1 = \sigma, \theta_2 = \beta, \theta_3 = \delta]$, i.e., the standard deviation of the observation noise and the two parameters of the kernel $\kappa(z, r)$. We assume a prior with independent truncated positive Gaussian components for the hyper-parameters $p(\boldsymbol{\theta}) = p(\sigma, \beta, \delta) = \mathcal{N}(\sigma; 0, 5)\mathcal{N}(\beta; 0, 5)\mathcal{N}(\delta; 0, 5)\mathbb{I}_{\sigma}\mathbb{I}_{\beta}\mathbb{I}_{\gamma}$ where $\mathbb{I}_{\nu} = 1$ if $\nu > 0$, and $\mathbb{I}_{\nu} = 0$ if $\nu \leq 0$. To simplify the expression of the posterior pdf, let us focus on the filtering problem and the tune of the parameters, namely we desire to infer $\mathbf{f}$ and $\boldsymbol{\theta}$. Hence, the posterior pdf is given by

$$p(\mathbf{f}, \boldsymbol{\theta}|\mathbf{y}, \mathbf{Z}, \kappa) = \frac{p(\mathbf{y}|\mathbf{f}, \mathbf{Z}, \boldsymbol{\theta}, \kappa)p(\mathbf{f}|\mathbf{Z}, \boldsymbol{\theta}, \kappa)p(\boldsymbol{\theta})}{p(\mathbf{y}|\mathbf{Z}, \kappa)}, \tag{23}$$

with $p(\mathbf{y}|\mathbf{f}, \mathbf{Z}, \boldsymbol{\theta}, \kappa) = \mathcal{N}\left(\mathbf{y}; \mathbf{0}, \sigma^2\mathbf{I}\right)$ and $p(\mathbf{f}|\mathbf{y}, \mathbf{Z}, \boldsymbol{\theta}, \kappa) = \mathcal{N}(\mathbf{f}; \boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$, with mean $\boldsymbol{\mu}_p = \mathbf{K}\left(\mathbf{K} + \sigma^2\mathbf{I}\right)^{-1}\mathbf{y}^{\top}$ and covariance matrix $\boldsymbol{\Sigma}_p = \mathbf{K} - \mathbf{K}\left(\mathbf{K} + \sigma^2\mathbf{I}\right)^{-1}\mathbf{K}^{\top}$, representing the solution of the GP given the specific choice of the hyper-parameters $\boldsymbol{\theta}$. The marginal posterior of the hyper-parameters [56] is

$$p(\boldsymbol{\theta}|\mathbf{y}, \mathbf{Z}, \kappa) = \int p(\mathbf{f}, \boldsymbol{\theta}|\mathbf{y}, \mathbf{Z}, \kappa)d\mathbf{f} = \frac{p(\mathbf{y}|\mathbf{Z}, \boldsymbol{\theta}, \kappa)p(\boldsymbol{\theta})}{p(\mathbf{y}|\mathbf{Z}, \kappa)}. \tag{24}$$

where

$$p(\mathbf{y}|\mathbf{Z}, \boldsymbol{\theta}, \kappa) = \int p(\mathbf{y}|\mathbf{Z}, \mathbf{f}, \boldsymbol{\theta}, \kappa)p(\mathbf{f}|\mathbf{Z}, \boldsymbol{\theta}, \kappa)d\mathbf{f}. \tag{25}$$

Hence, the log-marginal posterior is

$$\log\left[p(\boldsymbol{\theta}|\mathbf{y}, \mathbf{Z}, \kappa)\right] \propto -\frac{1}{2}\mathbf{y}(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y}^{\top} - \frac{1}{2}\log[\det[\mathbf{K} + \sigma^2\mathbf{I}]] - \frac{1}{10}\sum_{i=1}^{3}\theta_i^2, \tag{26}$$

for $\theta_1, \theta_2, \theta_3 > 0$, where clearly $\mathbf{K}$ depends on $\theta_1 = \sigma$, $\theta_2 = \beta$ and $\theta_3 = \delta$.[11] We apply a Gibbs sampler from drawing from $p(\boldsymbol{\theta}|\mathbf{y}, \mathbf{Z}, \kappa)$. We fix $\mathbf{Z} = [-10 : 0.1 : 10]$ (i.e., a grid between -10 and 10 with step 0.1); hence, $P = 201$, and the data $\mathbf{y}$ are artificially generated according to the model (21) considering the values $\boldsymbol{\theta}^* = [\sigma^* = 1, \beta^* = 0.5, \delta^* = 3]$. We average the results using 10^3 independent runs. At each run, we generate new data $\mathbf{y}$ according to the model with $\boldsymbol{\theta}^*$, and run the Gibbs sampler in order to approximate $p(\boldsymbol{\theta}|\mathbf{y}, \mathbf{Z}, \kappa)$ considering $N_G = 2000$ samples (without removing any burn-in period). We approximate the expected value of the posterior $\widehat{\boldsymbol{\theta}} \approx E_p[\boldsymbol{\theta}]$ using these N_G samples and compare with $\boldsymbol{\theta}^*$ (with enough number of data, it can be considered the ground-truth). For drawing from the full-conditional pdfs, we set $T = 10$, we employ a standard MH with Gaussian random proposal a $q(x_{\ell,t}|x_{\ell,t-1}) \propto \exp\left(-(x_{\ell,t} - x_{\ell,t-1})^2/\left(2\sigma_p^2\right)\right)$ for $\ell \in \{1, 2, 3\}$, and we test different values of $\sigma_p \in \{1, 2, 3\}$. Moreover, we apply AISM-P4-R3 with $T = 10$ and the initial support points $\mathcal{S}_0 = \{0.01, 0.2, 0.5, 1, 2, 4, 7, 10\}$. We also test the IA^{2}RMS method [13] which is a special case of AISM technique (see Section 6.1). For IA^{2}RMS, we use the construction procedure P4 as in AISM (both methods employ the update rule R3). The initializations for all techniques is set $x_{\ell,0}^{(k)} = 1$ and $x_{\ell,0}^{(k)} = x_{\ell,T}^{(k-1)}$ for $\ell = 1, 2, 3$ and $k = 1, \ldots, N_G$. The mean square error (MSE) in the estimation of θ^*, averaged over 10^3 runs, is shown in Table 9. AISM outperforms the MH methods. IA^{2}RMS provides better results w.r.t. AISM since it uses a better equivalent proposal $p_t(x) \propto \min\{q_t(x), \pi(x)\}$. However, IA^{2}RMS is slower than AISM due to its rejection step (necessary in order to produce samples from the equivalent proposal $p_t(x) \propto \min\{q_t(x), \pi(x)\}$). We recall that IA^{2}RMS is a special case of AISM technique. Finally, Table 9 shows the MSE in the estimation of the hyper-parameters $\boldsymbol{\theta}^*$ employing a Riemann quadrature, i.e., using a grid approximation $[0, A]^3$ with $A = 100$ and with step $\epsilon_g \in \{0.1, 0.2, 0.5, 1, 2\}$ (note this method excludes the possibility that the hyper-parameters are greater than A). The

Table 9 (Ex-Sect-9.6.1). MSE in the estimation of the hyper-parameters $\boldsymbol{\theta}^*$ with $N_G = 2000$

Algorithm	MH ($\sigma_p = 1$)	MH ($\sigma_p = 2$)	MH ($\sigma_p = 3$)	IA^{2}RMS-P4	AISM-P4-R3
MSE	6.21	5.08	6.83	3.12	3.46
Time	1	1	1	1.64	1.42

Note that IA^{2}RMS is a special case of AISM which employs the equivalent proposal $p_t(x) \propto \min\{q_t(x), \pi(x)\}$, and the rule R3 (see Section 6.1). In IA^{2}RMS, we have used the construction procedure P4 in order to build $q_t(x)$. The computing times are normalized w.r.t. the time spent by MH

computing times are normalized w.r.t. the time spent by MH in Tables 9 and 10.

9.6.2 Automatic Relevant Determination kernel function

Here we consider the estimation of the hyper-parameters of the Automatic Relevance Determination (ARD) covariance ([62], Chapter 6). Let us assume again the P observed data pairs $\{y_j, \mathbf{z}_j\}_{j=1}^P$, with $y_j \in \mathbb{R}$ and

$$\mathbf{z}_j = \left[z_{j,1}, z_{j,2}, \ldots, z_{j,d_Z}\right]^\top \in \mathbb{R}^{d_Z},$$

where d_Z is the dimension of the input features. We also denote the corresponding $P \times 1$ output vector as $\mathbf{y} = [y_1, \ldots, y_P]^\top$ and the $d_Z \times P$ input matrix $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_P]$. We again address the regression problem of inferring the unknown function f which links the variable y and $\mathbf{z}$. Thus, the assumed model is $y = f(\mathbf{z}) + e$, where $e \sim N\left(e; 0, \sigma^2\right)$, and that $f(\mathbf{z})$ is a realization of a Gaussian process (GP) [56]. Hence $f(\mathbf{z}) \sim \mathcal{GP}(\mu(\mathbf{z}), \kappa(\mathbf{z}, \mathbf{r}))$ where $\mu(\mathbf{z}) = 0$, $\mathbf{z}, \mathbf{r} \in \mathbb{R}^{d_Z}$, and we consider the ARD kernel function

$$\kappa(\mathbf{z}, \mathbf{r}) = \exp\left(-\sum_{\ell=1}^{d_Z} \frac{(z_\ell - r_\ell)^2}{2\delta_\ell^2}\right), \text{ with } \delta_\ell > 0, \quad (27)$$

for $\ell = 1, \ldots, d_Z$. Note that we have a different hyper-parameter δ_ℓ for each input component z_ℓ; hence, we also define $\boldsymbol{\delta} = \delta_{1:d_Z} = [\delta_1, \ldots, \delta_{d_Z}]$. Unlike in the previous section, note that here β is assumed known ($\beta = 2$). This type of kernel function is often employed to perform an *automatic relevance determination* (ARD) of the input components with respect the output variable ([62], Chapter 6). Namely, using ARD allows us to infer the relative importance of different components of inputs: a small value of δ_ℓ means that a variation of the ℓ-component z_ℓ impacts the output more, while a high value of δ_ℓ shows virtually independence between the ℓ-component and the output. Therefore, the complete vector containing all the hyper-parameters of the model is

$$\boldsymbol{\theta} = \left[\theta_{1:d_Z} = \delta_{1:d_Z}, \theta_{d_Z+1} = \sigma\right],$$
$$\boldsymbol{\theta} = [\boldsymbol{\delta}, \sigma] \in \mathbb{R}^{d_Z+1},$$

i.e., all the parameters of the kernel function in Eq. (22) and standard deviation σ of the observation noise. We assume $p(\boldsymbol{\theta}) = \prod_{\ell=1}^{d_Z+1} \frac{1}{\theta_\ell^\alpha} \mathbb{I}_{\theta_\ell}$ where $\alpha = 1.3$, $\mathbb{I}_\nu = 1$ if $\nu > 0$, and $\mathbb{I}_\nu = 0$ if $\nu \leq 0$. We desire to compute the expected value $\mathbb{E}[\boldsymbol{\Theta}]$ with $\boldsymbol{\Theta} \sim p(\boldsymbol{\theta}|\mathbf{y}, \mathbf{Z}, \kappa)$, via Monte Carlo quadrature.

More specifically, we apply a AISM-P4-R3 within-Gibbs (with $\mathcal{S}_0 = \{0.01, 0.5, 1, 2, 5, 8, 10, 15\}$) and the Single Component Adaptive Metropolis (SCAM) algorithm [63] within-Gibbs to draw from $\pi(\boldsymbol{\theta}) \propto p(\boldsymbol{\theta}|\mathbf{y}, \mathbf{Z}, \kappa)$. Note that dimension of the problem is $D = d_X + 1$ since $\boldsymbol{\theta} \in \mathbb{R}^D$. For SCAM, we use the Gaussian random walk proposal $q(x_{\ell,t}|x_{\ell,t-1}) \propto \exp\left(-(x_{\ell,t} - x_{\ell,t-1})^2 / \left(2\gamma_{\ell,t}^2\right)\right)$. In SCAM, the scale parameters $\gamma_{\ell,t}$ are adapted (one for each component) considering all the previous corresponding samples (starting with $\gamma_{\ell,0} = 1$).

We generated the $P = 500$ pairs of data, $\{y_j, \mathbf{z}_j\}_{j=1}^P$, drawing $\mathbf{z}_j \sim \mathcal{U}\left([0, 10]^{d_Z}\right)$ and $\mathbf{y}_j$ according to the model in Eq. (21), considered $d_Z \in \{1, 3, 5, 7, 9\}$ so that $D \in \{2, 4, 6, 8, 10\}$, and set $\sigma^* = \frac{1}{2}$ and $\delta_\ell^* = 2$, $\forall \ell$, for all the experiments (recall that $\boldsymbol{\theta}^* = [\boldsymbol{\delta}^*, \sigma^*]$). We consider $\boldsymbol{\theta}^*$ as ground truth and compute the MSE obtained by the different Monte Carlo techniques.

We have averaged the results using 10^3 independent runs. We consider $N_G = 1000$ and $T = 20$ for both schemes, AISM-within-Gibbs and SCAM-within-Gibbs. The results are provided in Table 11. We can see that

Table 10 (Ex-Sect-9.6.1). MSE in the estimation of the hyper-parameters $\boldsymbol{\theta}^*$ employing a Riemann quadrature, i.e., using a grid approximation $[0, 100]^3$ with step ϵ_g

MSE	10.52	8.72	4.09	2.67	1.01
ϵ_g	2	1	0.5	0.2	0.1
Time	0.11	0.36	1.25	7.31	20.71

The computing times are normalized w.r.t. the time spent by MH in Table 9

Table 11 (Ex-Sect-9.6.2). MSE for different techniques and different dimensions $D = d_Z + 1$ of the inference problem (number of hyper-parameters), with $T = 20$ and $N_G = 1000$ for both schemes

Algorithm	$D = 2$	$D = 4$	$D = 6$	$D = 8$	$D = 10$
SCAM within-Gibbs	0.0452	0.3013	1.61	2.87	4.68
AISM-P4-R3 within-Gibbs	0.0170	0.1521	0.5821	1.33	2.67

AISM-P4-R3 provides the better performance and the difference increases with the dimension $D = d_Z + 1$ of the problem.

10 Conclusions

In this work, we have introduced a new class of adaptive MCMC algorithms for any-purpose stochastic simulation. We have discussed the general features of the novel family, describing the different parts which form a generic sticky adaptive MCMC algorithm. The proposal density used in the new class is adapted on-line, constructed by employing non-parametric procedures. The name "sticky" remarks that the proposal pdf becomes progressively more and more similar to the target. Namely, a complete adaptation of the shape of the proposal is obtained (unlike using parametric proposals). The role of the update control test for the inclusion of new support points has been investigated. The design of this test is extremely important, since it controls the trade-off between computational cost and the efficiency of the resulting algorithm. Moreover, we have discussed how the combined design of a suitable proposal construction and a proper update test ensures the ergodicity of the generated chain.

Two specific sticky schemes, AISM and ASMTM, have been proposed and tested exhaustively in different numerical simulations. The numerical results show the efficiency of the proposed algorithms with respect to other state-of-the-art adaptive MCMC methods. Furthermore, we have showed that other well-known algorithms already introduced in the literature are encompassed by the novel class of methods proposed. A detailed description of the related works in the literature and their range of applicability are also provided, which is particularly useful for the interested practitioners and researchers. The novel methods can be applied both as a stand-alone algorithm or within any Monte Carlo approach that requires sampling from univariate densities (e.g., the Gibbs sampler, the hit-and-run algorithm or adaptive direction sampling). A promising future line is designing suitable constructions of the proposal density in order to allow the direct sampling from multivariate target distributions (similarly as [21, 30, 31, 39, 40]). However, we remark that the structure of the novel class of methods is valid regardless of the dimension of the target.

Endnotes

[1] The adjective "sticky" highlights the ability of the proposed schemes to generate a sequence of proposal densities that progressively "stick" to the target.

[2] The purpose of this work is to provide a family of methods applicable to a wide range of signal processing problems. A generic Matlab code (not focusing on any specific application) is provided at http://www.lucamartino.altervista.org/STICKY.zip.

[3] A preliminary version of this work has been published in [64]. With respect to that paper, the following major changes have been performed: we discuss exhaustively the general structure of the new family (not just a particular algorithm); we perform a complete theoretical analysis of the AISM algorithm; we extend substantially the discussion about related works; we introduce the AISMTM algorithm; we show how sticky methods can be used to sample from multi-variate pdfs by embedding them within a Gibbs sampler or the hit and run algorithm; and we provide additional numerical simulations (including comparisons with other benchmark sampling algorithms and the estimation of the hyper parameters of a Gaussian processes).

[4] For simplicity, we assume that $\pi(x)$ is bounded. However, the case of unbounded target pdfs can also be tackled by designing a suitable proposal construction that takes into account the vertical asymptotes of the target function. Similarly, we consider a target function defined in a continuous space $\mathcal{X}$ for the sake of simplicity, although the support domain could also be discrete.

[5] Note that any other MCMC technique could be used.

[6] Note that $d_t(z) \leq \max\{\pi(z), q_t(z|\mathcal{S}_t)\} \leq M_\pi$, since $M_t = \max_{z\in\mathcal{X}} q_t(z|\mathcal{S}_t) \leq M_\pi$ for all of the constructions described in Section 3 for the proposal function. Therefore, all the $\varepsilon_t \geq M_\pi$ lead to equivalent update rules.

[7] Regarding the definition of ε_t, this threshold should decrease over time (to guarantee that new support points can always be added), but not too fast (to avoid adding too many points and thus increasing the computational cost). Selecting the optimum threshold can be very challenging, but many of the rules that have been used in the area of stochastic filtering for the update parameter could be used here. For instance, good update rules could be $\varepsilon_t = \kappa M_\pi \cdot e^{-\gamma t}$ or $\varepsilon_t = \frac{\kappa M_\pi}{t+1}$ for some appropriate values of $0 < \kappa < 1$ and $\gamma > 0$. Exploring this issue is out of the scope of this paper, but we plan to address this in future works.

[8] We have used the equality $d_t(z_i) = |\pi(z_i) - q_t(z_i|\mathcal{S}_t)| = \max\{\pi(z_i), q_t(z_i|\mathcal{S}_t)\} - \min\{\pi(z_i), q_t(z_i|\mathcal{S}_t)\}$.

[9] Preliminary Matlab code for the AISM algorithm, with the constructions described in Section 3.1 and the update control rule R3, is provided at https://www.mathworks.com/matlabcentral/fileexchange/

`54701-adaptive-independent-sticky-metropolis-aism-algorithm`.

[10] Other related codes can be also found at `http://mc-stan.org`.

[11] Recall that if $\theta_1\theta_2\theta_3 \leq 0$ then $p(\boldsymbol{\theta}|\mathbf{y},\mathbf{Z},\kappa) = 0$.

[12] Note that we can always guarantee that $q_t(x|\mathcal{S}_t)$ is heavier tailed than $\pi(x)$ by using an appropriate construction for the tails of the proposal, as discussed in Section 3 and Appendix E.2.

[13] If we consider the complementary case (i.e., $\pi(s_{i+1}) \geq \pi(s_i)$ and thus $q_t(x) = \pi(s_{i+1})\ \forall x \in \mathcal{I}_{t,i}$) we obtain exactly the same bound following an identical procedure.

[14] The same conclusion is obtained if we consider a point $s' \in (s_{m_t}, \infty)$.

[15] Note that the proposals are assumed to be uniformly heavier tailed than the target by Condition 4 of Definition 1. Therefore, we can guarantee that enough candidate samples are generated in the tails.

Appendix A: Proof of Theorem 1

Note that Eq. (9) in Theorem 1 is a direct consequence of Theorem 2 in [14], which requires $x_t \sim q(x|\mathcal{S}_t)$ to be independent of the current state, x_{t-1}, and the satisfaction of the strong Doeblin condition. Regarding the first issue, x_t is independent of x_{t-1} by construction of the algorithm, so we only need to focus on the second issue. The *strong Doeblin condition* is satisfied if, given a proposal pdf, $\widetilde{q}_t(x|\mathcal{S}_t) = \frac{1}{c_t} q_t(x|\mathcal{S}_t)$, and a target, $\widetilde{\pi}(x) = \frac{1}{c_\pi}\pi(x)$ with support $\mathcal{X} \subseteq \mathbb{R}$, there exists some $a_t \in (0, 1]$ such that, for all $x \in \mathcal{X}$ and $t \in \mathbb{N}$,

$$\frac{1}{a_t}\widetilde{q}_t(x|\mathcal{S}_t) \geq \widetilde{\pi}(x). \tag{28}$$

First of all, note that Eq. (28) can be rewritten as

$$a_t \leq \frac{c_\pi}{c_t}\frac{q_t(x|\mathcal{S}_t)}{\pi(x)} \qquad \forall x \in \mathcal{X} \quad \text{and} \quad \forall t \in \mathbb{R}. \tag{29}$$

Then, note also that

$$\begin{aligned}\frac{c_\pi}{c_t}\frac{q_t(x|\mathcal{S}_t)}{\pi(x)} &\geq \frac{c_\pi}{c_t}\min_{x\in\mathcal{X}}\left\{\frac{q_t(x|\mathcal{S}_t)}{\pi(x)}\right\} \\ &\geq \min\left\{1, \frac{c_\pi}{c_t}\min_{x\in\mathcal{X}}\left\{\frac{q_t(x|\mathcal{S}_t)}{\pi(x)}\right\}\right\},\end{aligned}$$

where the last inequality is due to the fact that $\min\{1, x\} \leq x$. Therefore, a possible value of a_t that allows us to satisfy Eq. (29) is

$$a_t = \min\left\{1, \frac{c_\pi}{c_t}\min_{x\in\mathcal{X}}\left\{\frac{q_t(x|\mathcal{S}_t)}{\pi(x)}\right\}\right\}. \tag{30}$$

From Eq. (30) it is clear that $a_t \leq 1$, so all that remains to be shown is that $a_t > 0$. Let us recall that $\mathcal{I}_t = (s_1, s_{m_t}]$, where s_1 and s_{m_t} are the smallest and largest support points in $\mathcal{S}_t = \{s_1, \ldots, s_{m_t}\}$, respectively. Then, since $q_t(x|\mathcal{S}_t) > 0$ for all $x \in \mathcal{X}$ (condition 1 in Definition 1) and $t \in \mathbb{N}$, and $\pi(x)$ is assumed to be bounded, we have

$$\min\left\{1, \frac{c_\pi}{c_t}\min_{x\in\mathcal{I}_t}\left\{\frac{q_t(x|\mathcal{S}_t)}{\pi(x)}\right\}\right\} > 0.$$

And regarding the tails, note that $q_t(x|\mathcal{S}_t)$ must be uniformly heavier tailed by construction (condition 4 in Definition 1),[12] so $q_t(x|\mathcal{S}_t) \geq \pi(x)$ for all $x \in \mathcal{I}_t^c = (-\infty, s_1] \cup (s_{m_t}, \infty)$ and we also have

$$\min\left\{1, \frac{c_\pi}{c_t}\min_{x\in\mathcal{I}_t^c}\left\{\frac{q_t(x|\mathcal{S}_t)}{\pi(x)}\right\}\right\} > 0.$$

Therefore, we conclude that $0 < a_t \leq 1$, the strong Doeblin condition is satisfied and thus all the conditions for Theorem 2 in [14] are fulfilled.

Appendix B: Argumentation for Conjecture 1

Let us define $\mathcal{I}_t = (s_1, s_{m_t}]$ and $\mathcal{I}_t^c = (-\infty, s_1] \cup (s_{m_t}, \infty)$, where s_1 and s_{m_t} are the smallest and largest points of the set of support points at time step t, $\mathcal{S}_t = \{s_1, \ldots, s_{m_t}\}$ with $s_1 < \ldots < s_{m_t}$. Then, the L_1 distance between the target and the proposal can be expressed as $D_1(\pi, q_t) = D_{\mathcal{I}_t}(\pi, q_t) + D_{\mathcal{I}_t^c}(\pi, q_t)$, where $D_{\mathcal{I}_t}(\pi, q_t) = \int_{\mathcal{I}_t} d_t(x)\, dx$ and $D_{\mathcal{I}_t^c}(\pi, q_t) = \int_{\mathcal{I}_t^c} d_t(x)\, dx$ with $d_t(x) = |\pi(x) - q_t(x)|$. Let us focus first on $D_{\mathcal{I}_t}(\pi, q_t)$. Since $q_t(x)$ is constructed as a piecewise polynomial approximation on the intervals $\mathcal{I}_{t,i} = (s_i, s_{i+1}]$,

$$D_{\mathcal{I}_t}(\pi, q_t) = \sum_{i=1}^{m_t-1} D_{\mathcal{I}_{t,i}}(\pi, q_t), \tag{31}$$

where

$$D_{\mathcal{I}_{t,i}}(\pi, q_t) = \int_{\mathcal{I}_{t,i}} d_t(x)dx$$

is the L_1 distance between the target and the proposal in the i-th interval. Now, using Theorem 3.1.1 in [65] we can easily bound $d_t(x)$ for the ℓ-th order interpolation polynomial (with $\ell \in \{0, 1\}$ in this case) used within the i-th interval. For $\ell = 0$ and assuming that $\pi(s_i) \geq \pi(s_{i+1})$ (and thus $q_t(x) = \pi(s_i)\ \forall x \in \mathcal{I}_{t,i}$) without loss of generality,[13]

$$\begin{aligned} d_t(x) &= |\pi(x) - q_t(x)| \\ &= |x - s_i||\dot{\pi}(\xi)| \\ &\leq (s_{i+1} - s_i)\max_{x\in\mathcal{I}_{t,i}}|\dot{\pi}(x)| < \infty, \end{aligned}$$

where $\dot{\pi}(\xi)$ denotes the first derivative of $\pi(x)$ evaluated at $x = \xi$, $\xi \in (s_i, s_{i+1}]$ is some point inside the interval whose value depends on x, x_i and $\pi(x)$, and this bound is finite since we assume that the first derivative of $\pi(x)$ is bounded. Therefore, for the PWC approximation we have

$$D_{\mathcal{I}_t}(\pi, q_t) \leq \sum_{i=1}^{m_t-1} (s_{i+1}-s_i)^2 \max_{x\in\mathcal{I}_{t,i}} |\dot{\pi}(x)| \leq \max_{x\in\mathcal{I}_t} |\dot{\pi}(x)| \cdot \sum_{i=1}^{m_t-1} (s_{i+1}-s_i)^2 < \infty. \quad (32)$$

Similarly, for $\ell = 1$ we have

$$d_t(x) = |\pi(x) - q_t(x)| = \frac{|x-s_i||x-s_{i+1}|}{2}|\ddot{\pi}(\xi)| \leq \frac{(s_{i+1}-s_i)^2}{2} \max_{x\in\mathcal{I}_{t,i}} |\ddot{\pi}(x)| < \infty,$$

where $\ddot{\pi}(\xi)$ denotes the second derivative of $\pi(x)$ evaluated at $x = \xi$, $\xi \in (s_i, s_{i+1}]$ is some point inside the interval, and this bound is again finite since we assume that the second derivative of $\pi(x)$ is also bounded. And the L_1 distance for the PWL approximation can thus be bounded as

$$D_{\mathcal{I}_t}(\pi, q_t) \leq \sum_{i=1}^{m_t-1} \frac{(s_{i+1}-s_i)^3}{2} \max_{x\in\mathcal{I}_{t,i}} |\ddot{\pi}(x)| \leq \frac{1}{2} \max_{x\in\mathcal{I}_t} |\ddot{\pi}(x)| \cdot \sum_{i=1}^{m_t-1} (s_{i+1}-s_i)^3 < \infty. \quad (33)$$

Note that the two cases can be summarized in a single expression:

$$D_{\mathcal{I}_t}(\pi, q_t) \leq L_t^{(\ell)}, \quad (34)$$

where

$$L_t^{(\ell)} = C_t^{(\ell)} \cdot \sum_{i=1}^{m_t-1} (s_{i+1}-s_i)^{\ell+1}, \quad (35)$$

with $C_t^{(0)} = \max_{x\in\mathcal{I}_t} |\dot{\pi}(x)|$ and $C_t^{(1)} = \frac{1}{2}\max_{x\in\mathcal{I}_t} |\ddot{\pi}(x)|$.

Now, let us assume that a new point, $s' \in \mathcal{I}_{t,k} = [s_k, s_{k+1}]$ for $1 \leq k \leq m_t - 1$, is added at some iteration $t' > t$ using the mechanism described in the AISM algorithm (see Table 1) and that no other points have been incorporated to the support set for $t+1, \ldots, t'-1$. In this case, the construction of the proposal function changes only inside the interval $\mathcal{I}_{t,k}$, which splits now into $\mathcal{I}_{t',k} = [s_k, s']$ and $\mathcal{I}_{t',k+1} = [s', s_{k+1}]$. Then, the new bound for the distance inside $\mathcal{I}_{t'} = \mathcal{I}_t$ is $D_{\mathcal{I}_{t'}}(\pi, q_{t'}) \leq L_{t'}^{(\ell)}$, with

$$L_{t'}^{(\ell)} = L_t^{(\ell)} + C_t^{(\ell)} \left[(s'-s_k)^{\ell+1} + (s_{k+1}-s')^{\ell+1} - (s_{k+1}-s_k)^{\ell+1} \right] < L_t^{(\ell)}, \quad (36)$$

where the last inequality is obtained by applying Newton's binomial theorem, which states that $A^{\ell+1} + B^{\ell+1} < (A+B)^{\ell+1}$ for any $A, B > 0$, using $A = s' - s_k > 0$ and $B = s_{k+1} - s' > 0$. Hence, the bound in Eq. (36) can never increase when a new support point is incorporated and indeed tends to decrease as new points are added to the support set.

Note that we could still have $L_t^{(\ell)} \to K > 0$ as $t \to \infty$. However, the conditions of Definition 1 ensure that the support of the proposal always contains the support of the target (i.e., $q_t(x|\mathcal{S}_t) > 0$ whenever $\pi(x) > 0$ for any t and $\mathcal{S}_t$) and it has uniformly heavier tails (implying that $q_t(x|\mathcal{S}_t) \to 0$ slower than $\pi(x)$ as $x \to \pm\infty$). Consequently, support points can be added anywhere inside the support of the target, $\mathcal{X} \subseteq \mathbb{R}$. This implies that $L_t^{(\ell)} \to 0$ as $t \to \infty$, since $(s_{i+1} - s_i) \to 0$ as more points are added inside $\mathcal{I}_t$, and thus also $D_{\mathcal{I}_t}(\pi, q_t) \to 0$ as $t \to \infty$. Let us focus now on $D_{\mathcal{I}_t^c}(\pi, q_t)$. Let us assume, without loss of generality, that a new point, $s' \in (-\infty, s_1]$,[14] is added at some iteration $t' > t$ using the mechanism described in the AISM algorithm (see Table 1) and that no other points have been incorporated to the support set for $t+1, \ldots, t'-1$. In this case, it is clear that the distance in the tails decreases (i.e., $D_{\mathcal{I}_{t'}^c}(\pi, q_t) < D_{\mathcal{I}_t^c}(\pi, q_t)$) at the expense of increasing the distance in the central part of the target (i.e., $D_{\mathcal{I}_{t'}}(\pi, q_t) > D_{\mathcal{I}_t}(\pi, q_t)$). However, even if this leads to a momentary increase in the overall distance, note that we still have $D_{\mathcal{I}_{t'}}(\pi, q_t) \to 0$ as $t' \to \infty$ as long as new support points can be added inside $\mathcal{I}_{t'}$, something which is guaranteed by the AISM algorithm. Finally, since there is always a non-null probability of incorporating points in the tails,[15] thus implying that $D_{\mathcal{I}_t^c}(\pi, q_t) \to 0$ as $t \to \infty$, since $\mathcal{I}_t^c$ becomes smaller and smaller as t increases.

Therefore, we can guarantee that using the AISM algorithm in Table 1, with a valid proposal that fulfills Definition 1 and an acceptance rule according to Definition 3, we obtain a sticky proposal that fulfills Definition 2.

Appendix C: Support points

In this appendix we provide the proofs of Theorem 3 and Corollary 4, which bound the expected growth of the number of support points.

C.1 Proof of Theorem 3

Given the support set $\mathcal{S}_t$ and the state x_{t-1}, the expected probability of adding a new point to $\mathcal{S}_t$ at the t-th iteration is given by

$$E[P_a(z)|x_{t-1}, \mathcal{S}_t] = \int_{\mathcal{X}} P_a(z) p_t(z|x_{t-1}, \mathcal{S}_t)\, dz, = \int_{\mathcal{X}} \eta_t(z, d_t(z)) p_t(z|x_{t-1}, \mathcal{S}_t)\, dz, \quad (37)$$

where $d_t(z) = \left|\pi(z) - q_t(z|\mathcal{S}_t)\right|$ and

$$p_t(z|x_{t-1},\mathcal{S}_t) = \int_{\mathcal{X}} p_t\left(z|x',x_{t-1},\mathcal{S}_t\right) p_t\left(x'|x_{t-1},\mathcal{S}_t\right)\, dx', \tag{38}$$

represents the kernel function of AISM given x_{t-1} and $\mathcal{S}_t$. Since candidate points $x' \in \mathcal{X}$ are directly drawn from the proposal pdf, we have $p_t\left(x'|x_{t-1},\mathcal{S}_t\right) = \widetilde{q}_t\left(x'|\mathcal{S}_t\right)$, and from the structure of the AISM in Table 1 it is straightforward to see that

$$\begin{aligned} p_t(z|x',x_{t-1},\mathcal{S}_t) =& \alpha(x_{t-1},x')\delta(z-x_{t-1}) \\ &+ \left[1-\alpha(x_{t-1},x')\right]\delta(z-x'), \end{aligned}$$

where $\alpha(x_{t-1},x') = \min\left[1, \frac{\pi(x')q_t(x_{t-1}|\mathcal{S}_t)}{\pi(x_{t-1})q_t(x'|\mathcal{S}_t)}\right]$. Inserting these two expressions in Eq. (38), the kernel function of AISM becomes

$$\begin{aligned} p_t(z|x_{t-1},\mathcal{S}_t) =& \left[\int_{\mathcal{X}} \alpha(x_{t-1},x')\, \widetilde{q}_t(x'|\mathcal{S}_t)\, dx'\right] \\ &\times \delta(z-x_{t-1}) \\ &+ \left[1-\alpha(x_{t-1},z)\right]\, \widetilde{q}_t(z|\mathcal{S}_t) \end{aligned} \tag{39}$$

Let us recall now the integral form of Jensen's inequality for a concave function $\varphi(x)$ with support $\mathcal{X} \subseteq \mathbb{R}$ [66]:

$$\int_{\mathcal{X}} \varphi(x) f(x)\, dx \leq \varphi\left(\int_{\mathcal{X}} x f(x)\, dx\right),$$

which is valid for any non-negative function $f(x)$ such that $\int_{\mathcal{X}} f(x)\, dx = 1$. Then, since we assume that $\eta_t(z,d) = \eta_t(d)$, $\eta_t(d)$ is a concave function of d by condition 4 of Definition 3, and $\int_{\mathcal{X}} p_t(z|x_{t-1},\mathcal{S}_t)dz = 1$, we have

$$\begin{aligned} E[P_a(z)|x_{t-1},\mathcal{S}_t] &= \int_{\mathcal{X}} \eta_t(d_t(z)) p_t(z|x_{t-1},\mathcal{S}_t)\, dz \\ &\leq \eta_t\left(E\left[d_t(z)|x_{t-1},\mathcal{S}_t\right]\right), \end{aligned} \tag{40}$$

with

$$\begin{aligned} E\left[d_t(z)|x_{t-1},\mathcal{S}_t\right] &= \eta_t\left(\int_{\mathcal{X}} d_t(z) p_t(z|x_{t-1},\mathcal{S}_t)\, dz\right) \\ &= \left[\int_{\mathcal{X}} \alpha(x_{t-1},x')\, \widetilde{q}_t(x'|\mathcal{S}_t)\, dx'\right] d_t(x_{t-1}) \\ &\quad + \int_{\mathcal{X}} \left[1-\alpha(x_{t-1},z)\right] d_t(z)\widetilde{q}_t(z|\mathcal{S}_t)\, dz, \end{aligned} \tag{41}$$

where we have used (39) to obtain the final expression in (41). Now, for the first term in the right hand side of (41), note that $\left[\int_{\mathcal{X}} \alpha(x_{t-1},x')\, \widetilde{q}_t(x'|\mathcal{S}_t)\, dx'\right] \leq 1$, since $0 \leq \alpha(x_{t-1},x') \leq 1$ and $\int_{\mathcal{X}} \widetilde{q}_t(x'|\mathcal{S}_t)\, dx' = 1$. And for the second term, we have

$$\begin{aligned} &\int_{\mathcal{X}} \left[1-\alpha(x_{t-1},z)\right] d_t(z)\widetilde{q}_t(z|\mathcal{S}_t)\, dz \\ &\leq \int_{\mathcal{X}} d_t(z)\widetilde{q}_t(z|\mathcal{S}_t)\, dz \\ &\leq C \cdot D_1(\pi,q_t), \end{aligned}$$

where we recall that $D_1(\pi,q_t) = \int_{\mathcal{X}} d_t(z)\, dz = \int_{\mathcal{X}} |\pi(z) - q_t(z|\mathcal{S}_t)|\, dz$ and $C = \max_{z\in\mathcal{X}} \widetilde{q}_t(z|\mathcal{S}_t) < \infty$, since we have assumed that $\pi(x)$ is bounded and thus, by condition 4 in Definition 1, $\widetilde{q}_t(z|\mathcal{S}_t)$ is also bounded. Therefore, we obtain

$$E[\, d_t(z)|x_{t-1},\mathcal{S}_t] \leq d_t(x_{t-1}) + C \cdot D_1(\pi,q_t), \tag{42}$$

and inserting (42) into (40) we have the following bound for the expected probability of adding a support point at the t-the iteration,

$$E[\, P_a(z)|x_{t-1},\mathcal{S}_t] \leq \eta_t\Big(d_t(x_{t-1}) + C \cdot D_1(\pi,q_t)\Big). \tag{43}$$

Finally, noting $C < \infty$, that both $d_t(x_{t-1}) \to 0$ and $D_1(q_t,\pi) \to 0$ as $t \to \infty$ by Conjecture 1, and that $\eta_t(0) = 0$ by condition 2 in Definition 3, we have $E[\, P_a(z)|x_{t-1},\mathcal{S}_t] \to 0$ as $t \to \infty$.

C.2 Proof of Corollary 4

First of all, recall that a semi-metric fulfills all the properties of a metric except for the triangle inequality. Therefore, we have $\widetilde{d}_t(\pi(z),q_t(z)) \geq 0$, $\widetilde{d}_t(\pi(z),q_t(z)) = 0 \iff \pi(z) = q_t(z)$ and $\widetilde{d}_t(\pi(z),q_t(z)) = \widetilde{d}_t(q_t(z),\pi(z))$. Now, from the proof of Theorem 3 (see Appendix C.1) we can see that η_t is not used until Eq. (40). Since $\eta_t(\widetilde{d}_t(z))$ is a concave function of $\widetilde{d}_t(z)$, we can still use Jensen's inequality and this equation becomes

$$E[\, P_a(z)|x_{t-1},\mathcal{S}_t] \leq \eta_t\Big(E[\, \widetilde{d}_t(z)|x_{t-1},\mathcal{S}_t]\, \Big),$$

where, following the same procedure as in Appendix C.1 (which is still valid due to the fact that $\widetilde{d}_t(\pi(z),q_t(z))$ is a semi-metric), the term inside η_t can be now bounded by

$$E\big[\widetilde{d}_t(z)|x_{t-1},\mathcal{S}_t\big] \leq \widetilde{d}_t(x_{t-1}) + C \cdot \widetilde{D}_t(\pi,q_t),$$

with $\widetilde{D}_t(\pi,q_t) = \int_{\mathcal{X}} \widetilde{d}_t(z)\, dz$. Therefore, we have

$$E\left[P_a(z)|x_{t-1},\mathcal{S}_t\right] \leq \eta_t\left(\widetilde{d}_t(x_{t-1}) + C \cdot \widetilde{D}_t(\pi,q_t)\right),$$

with $E[\, P_a(z)|x_{t-1},\mathcal{S}_t] \to 0$ as $t \to \infty$ under the conditions of Conjecture 1.

Appendix D: Variate generation

The proposal density $\widetilde{q}_t(x|\mathcal{S}_t) \propto q_t(x|\mathcal{S}_t)$, built using one of the interpolation procedures in Section 3.1, is composed of $m_t + 1$ pieces (including the two tails). More specifically, the function $q_t(x|\mathcal{S}_t)$ can be seen as a finite mixture

$$\widetilde{q}_t(x|\mathcal{S}_t) = \sum_{i=0}^{m} \eta_i \phi_i(x),$$

with $\sum_{i=0}^{m_t} \eta_i = 1$, whereas $\phi_i(x)$ is a linear pdf or a uniform pdf (depending on the employed construction; see Eqs. (3)-(4)) defined in the interval $\mathcal{I}_i$, and $\phi_i(x) = 0$ for $x \notin \mathcal{I}_i$. The tails, $\phi_0(x)$ and $\phi_{m_t}(x)$, are truncated exponential pdfs (or Pareto tails see Appendix E.2). Hence, in order to draw a sample from $\widetilde{q}_t(x|\mathcal{S}_t) \propto q_t(x|\mathcal{S}_t)$, it is necessary to perform the following steps:

1. Compute the area A_i below each piece composing $q_t(x|\mathcal{S}_t)$, $i = 0, \ldots, m_t$. This is straightforward for the construction procedures in Eqs. (3)-(4) since the function $q_t(x|\mathcal{S}_t)$ is formed by linear or constant pieces, so that it can be easily done analytically. Moreover, since the tails are exponential functions also in this case we compute the areas below A_0 and A_{m_t} analytically. Then, we need to normalize them,
$$\eta_i = \frac{A_i}{\sum_{j=1}^{m} A_j}, \quad \text{for} \quad i = 0, \ldots, m.$$
2. Choose a piece (i.e., an index $j^* \in \{0, \ldots, m_t\}$) according to the weights η_i for $i = 0, \ldots, m_t$.
3. Given the index j^*, draw a sample x' in the interval $\mathcal{I}_{j^*}$ with pdf $\phi_{j^*}(x)$, i.e., $x' \sim \phi_{j^*}(x)$.

Appendix E: Robust algorithms

In this appendix, we briefly discuss how to increase the robustness of the method, both with respect to a bad choice of the initial set $\mathcal{S}_0$ (e.g., when information about the range of the target pdf is not available) and w.r.t. the heavy tails that appear in many target pdfs.

E.1 Mixture of proposal densities

Let us define a proposal density as

$$\widetilde{q}(x) = \alpha_t \widetilde{q}_1(x) + (1 - \alpha_t)\widetilde{q}_2(x|\mathcal{S}_t), \tag{44}$$

where $\widetilde{q}_2(x|\mathcal{S}_t)$ is a sticky proposal pdf built as described in Section 3. The density $\widetilde{q}_1(x)$ is a generic proposal function with an explorative task. The explorative behavior of $\widetilde{q}_1$ can be controlled by its scale parameter. The weight α_t can be kept constant $\alpha_t = \alpha_0 = 0.5$ for all t (this is the most defensive strategy), or it can be decreased with the iteration t, i.e., $\alpha_t \to 0$ as $t \to \infty$. The joint adaptation of the weight α_t, the scale parameter of $\widetilde{q}_1$ and $\widetilde{q}_2$ using a sticky procedure needs and deserves additional studies.

E.2 Heavy tails

The choice of the tails for the proposal is important for two reasons: (a) to accelerate the convergence of the chain to the target (especially for heavy-tailed target distributions) and (b) to increase the robustness of the method w.r.t. the initial choice of the set $\mathcal{S}_0$. Indeed, often the construction of tails with a bigger area below them can reduce the dependence on a specific choice of the set of initial support points. For heavy tailed constructions, there are several possibilities. For instance, here we propose to use Pareto pieces, which have the following analytic form

$$q_t(x|\mathcal{S}_t) = e^{\rho_0} \frac{1}{|x - \mu_0|^{\gamma_0}}, \ \forall x \in \mathcal{I}_0,$$
$$q_t(x|\mathcal{S}_t) = e^{\rho_{m_t}} \frac{1}{|x - \mu_{m_t}|^{\gamma_{m_t}}}, \forall x \in \mathcal{I}_{m_t},$$

with $\gamma_j > 1, j \in \{0, m_t\}$. In the log-domain, this results in

$$w_0(x) = \rho_0 - \gamma_0 \log(|x - \mu_0|), \text{for} x \in \mathcal{I}_0,$$
$$w_{m_t}(x) = \rho_{m_t} - \gamma_{m_t} \log(|x - \mu_{m_t}|), \text{for} x \in \mathcal{I}_{m_t},$$

i.e., $q_t(x|\mathcal{S}_t) = \exp(w_i(x))$ with $i \in \{0, m_t\}$. Let us denote $V(x) = \log[\pi(x)]$. Fixing the parameters $\mu_j, j \in \{0, m_t\}$, the remaining parameters, ρ_j and γ_j, are set in order to satisfy the passing conditions through the points $(s_1, V(s_1))$ and $(s_2, V(s_2))$, and through the points $(s_{m_t-1}, V(s_{m_t-1}))$ and $(s_{m_t}, V(s_{m_t}))$, respectively. The parameters μ_j can be arbitrarily chosen by the user, as long as they fulfill the following inequalities:

$$\mu_0 > s_2, \quad \mu_{m_t} < s_{m_t-1}.$$

Values of μ_j such that $\mu_0 \approx s_2$ and $\mu_{m_t} \approx s_{m_t-1}$ yield small values of γ_j (close to 1) and, as a consequence, fatter tails. Larger differences in $|\mu_0 - s_2|$ and $|\mu_{m_t} - s_{m_t-1}|$ yield $\gamma_j \to +\infty$, i.e., lighter tails. Note that we can compute analytically the integral of $q_t(x)$ in $\mathcal{I}_0$ and $\mathcal{I}_{m_t}$:

$$A_0 = \frac{e^{\rho_0}}{\gamma_0 - 1} \frac{1}{(\mu_0 - s_1)^{\gamma_0 - 1}},$$
$$A_{m_t} = \frac{e^{\rho_{m_t}}}{\gamma_{m_t} - 1} \frac{1}{(s_{m_t} - \mu_{m_t})^{\gamma_{m_t} - 1}}.$$

Moreover, we can also draw samples easily from each Pareto tail using the inversion method [2].

Funding

This work has been supported by the Spanish Ministry of Economy and Competitiveness (MINECO) through the MIMOD-PLC (TEC2015-64835-C3-3-R) and KERMES (TEC2016-81900-REDT/AEI) projects; by the Italian Ministry of Education, University and Research (MIUR); by PRIN 2010-11 grant; and by the European Union (Seventh Framework Programme FP7/2007-2013) under grant agreement no:630677.

Authors' contributions

All the authors have participated in writing the manuscript and have revised the final version. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Author details

[1] Image Processing Lab., University of Valencia, Valencia, Spain. [2] Department of Economics, University Ca' Foscari of Venice, Venice, Italy. [3] School of Mathematics, Statistics and Actuarial Sciences, University of Kent, Canterbury, UK. [4] Department of Signal Theory and Communications, Universidad Politécnica de Madrid, Madrid, Spain.

Received: 27 February 2017 Accepted: 20 December 2017
Published online: 11 January 2018

References

1. JS Liu, *Monte Carlo Strategies in Scientific Computing*. (Springer-Verlag, 2004)
2. CP Robert, G Casella, *Monte Carlo Statistical Methods*. (Springer, 2004)
3. WJ Fitzgerald, Markov chain Monte Carlo methods with applications to signal processing. Signal Process. **81**, 3–18 (2001)
4. A Doucet, X Wang, Monte Carlo methods for signal processing: a review in the statistical signal processing context. IEEE Signal Process. Mag. **22**, 152–17 (2005)
5. M Davy, C Doncarli, JY Tourneret, Classification of chirp signals using hierarchical Bayesian learning and MCMC methods. IEEE Trans. Signal Process. **50**, 377–388 (2002)
6. N Dobigeon, JY Tourneret, CI Chang, Semi-supervised linear spectral unmixing using a hierarchical Bayesian model for hyperspectral imagery. IEEE Trans. Signal Process. **56**, 2684–2695 (2008)
7. T Elguebaly, N Bouguila, Bayesian learning of finite generalized Gaussian mixture models on images. Signal Process. **91**, 801–820 (2011)
8. GO Roberts, JS Rosenthal, Examples of adaptive MCMC. J. Comput. Graph. Stat. **18**, 349–367 (2009)
9. C Andrieu, J Thoms, A tutorial on adaptive MCMC. Stat. Comput. **18**, 343–373 (2008)
10. H Haario, E Saksman, J Tamminen, An adaptive Metropolis algorithm. Bernoulli. **7**, 223–242 (2001)
11. F Liang, C Liu, R Caroll, *Advanced Markov Chain Monte Carlo Methods: Learning from Past Samples*. (Wiley Series in Computational Statistics, England, 2010)
12. WR Gilks, NG Best, KKC Tan, Adaptive rejection Metropolis sampling within Gibbs sampling. Appl.Stat. **44**, 455–472 (1995)
13. L Martino, J Read, D Luengo, Independent doubly adaptive rejection Metropolis sampling within Gibbs sampling. IEEE Trans. Signal Process. **63**, 3123–3138 (2015)
14. L Holden, R Hauge, M Holden, Adaptive independent Metropolis-Hastings. Ann. Appl. Probab. **19**, 395–413 (2009)
15. C Ritter, MA Tanner, Facilitating the Gibbs sampler: The Gibbs stopper and the griddy-Gibbs sampler. J. Am. Stat. Assoc. **87**, 861–868 (1992)
16. R Meyer, B Cai, F Perron, Adaptive rejection Metropolis sampling using Lagrange interpolation polynomials of degree 2. Comput. Stat. Data Anal. **52**, 3408–3423 (2008)
17. L Martino, J Read, D Luengo, Independent doubly adaptive rejection Metropolis sampling. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) (2014)
18. L Martino, H Yang, D Luengo, J Kanniainen, J Corander, A fast universal self-tuned sampler within Gibbs sampling. Digital Signal Process. **47**, 68–83 (2015)
19. WR Gilks, P Wild, Adaptive rejection sampling for Gibbs sampling. Appl. Stat. **41**, 337–348 (1992)
20. B Cai, R Meyer, F Perron, Metropolis-Hastings algorithms with adaptive proposals. Stat. Comput. **18**, 421–433 (2008)
21. W Hörmann, J Leydold, G Derflinger, *Automatic nonuniform random variate generation*. (Springer, 2003)
22. G Krzykowski, W Mackowiak, Metropolis Hastings simulation method with spline proposal kernel. An Isaac Newton Institute Workshop (2006)
23. W Shao, G Guo, F Meng, S Jia, An efficient proposal distribution for Metropolis-Hastings using a b-splines technique. Comput. Stat. Data Anal. **53**, 465–478 (2013)
24. L Tierney, Markov chains for exploring posterior distributions. Ann. Stat. **22**, 1701–1728 (1994)
25. L Martino, J Míguez, Generalized rejection sampling schemes and applications in signal processing. Signal Process. **90**, 2981–2995 (2010)
26. WsR Gilks, Derivative-free adaptive rejection sampling for Gibbs sampling. Bayesian Stat. **4**, 641–649 (1992)
27. D Görür, YW Teh, Concave convex adaptive rejection sampling. J. Comput. Graph. Stat. **20**, 670–691 (2011)
28. W Hörmann, A rejection technique for sampling from T-concave distributions. ACM Trans. Math. Softw. **21**, 182–193 (1995)
29. L Martino, F Louzada, Adaptive rejection sampling with fixed number of nodes. (to appear) Communications in Statistics - Simulation and Computation, 1–11 (2017). doi:10.1080/03610918.2017.1395039
30. J Leydold, A rejection technique for sampling from log-concave multivariate distributions. ACM Trans. Model. Comput. Simul. **8**, 254–280 (1998)
31. J Leydold, W Hörmann, A sweep plane algorithm for generating random tuples in a simple polytopes. Math. Comput. **67**, 1617–1635 (1998)
32. KR Koch, Gibbs sampler by sampling-importance-resampling. J. Geodesy. **81**, 581–591 (2007)
33. AE Gelfand, TM Lee, Discussion on the meeting on the Gibbs sampler and other Markov Chain Monte Carlo methods. J. R. Stat. Soc. Ser. B. **55**, 72–73 (1993)
34. C Fox, A Gibbs sampler for conductivity imaging and other inverse problems. Proc. SPIE Image Reconstruction Incomplete Data VII. **8500**, 1–6 (2012)
35. P Müller, *A generic approach to posterior integration and, Gibbs sampling. Technical Report 91-09*. (Department of Statistics of Purdue University, 1991)
36. JS Liu, F Liang, WH Wong, The multiple-try method and local optimization in Metropolis sampling. J. Am. Stat. Assoc. **95**, 121–134 (2000)
37. L Martino, J Read, On the flexibility of the design of multiple try Metropolis schemes. Comput. Stat. **28**, 2797–2823 (2013)
38. D Luengo, L Martino, Almost rejectionless sampling from Nakagami-m distributions ($m \geq 1$). IET Electron. Lett. **48**, 1559–1561 (2012)
39. R Karawatzki, The multivariate Ahrens sampling method. Technical Report 30, Department of Statistics and Mathematics (2006)
40. W Hörmann, A universal generator for bivariate log-concave distributions. Computing. **52**, 89–96 (1995)
41. BS Caffo, JG Booth, AC Davison, Empirical supremum rejection sampling. Biometrika. **89**, 745–754 (2002)
42. W Hörmann, A note on the performance of the Ahrens algorithm. Computing. **69**, 83–89 (2002)
43. J W Hörmann, G Leydold, *Derflinger, Inverse transformed density rejection for unbounded monotone densities. Research Report Series/ Department of Statistics and Mathematics (Economy and Business)*. (Vienna University, 2007)
44. G Marrelec, H Benali, Automated rejection sampling from product of distributions. Comput Stat. **19**, 301–315 (2004)
45. H Tanizaki, On the nonlinear and non-normal filter using rejection sampling. IEEE Trans. Automatic Control. **44**, 314–319 (1999)
46. M Evans, T Swartz, Random variate generation using concavity properties of transformed densities. J. Comput. Graph. Stat. **7**, 514–528 (1998)
47. L Martino, J Míguez, A generalization of the adaptive rejection sampling algorithm. Stat. Comput. **21**, 633–647 (2011)
48. M Brewer, C Aitken, Discussion on the meeting on the Gibbs sampler and other Markov Chain Monte Carlo methods. J. R. Stat. Soc. Ser. B. **55**, 69–70 (1993)
49. F Lucka, Fast Gibbs sampling for high-dimensional Bayesian inversion (2016). arXiv:1602.08595
50. H Zhang, Y Wu, L Cheng, I Kim, Hit and run ARMS: adaptive rejection Metropolis sampling with hit and run random direction. J. Stat. Comput. Simul. **86**, 973–985 (2016)
51. L Martino, V Elvira, G Camps-Valls, Recycling Gibbs sampling. 25th European Signal Processing Conference (EUSIPCO), 181–185 (2017)
52. WR Gilks, NGO Robert, EI George, Adaptive direction sampling. The Statistician. **43**, 179–189 (1994)
53. I Murray, Z Ghahramani, DJC MacKay, in *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*. MCMC for doubly-intractable distributions, (2006), pp. 359–366
54. D Rohde, J Corcoran, in *Statistical Signal Processing (SSP), 2014 IEEE Workshop on*. MCMC methods for univariate exponential family models with intractable normalization constants, (2014), pp. 356–359
55. RM Neal, Slice sampling. Ann. Stat. **31**, 705–767 (2003)
56. CE Rasmussen, CKI Williams, *Gaussian processes for machine learning*, (2006)
57. D Gamerman, *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*. (Chapman and Hall/CRC, 1997)
58. BP Carlin, S Chib, Bayesian model choice via markov chain monte carlo methods. J. R. Stat. Soc. Series B (Methodological). **3**, 473–484 (1995)
59. S Chib, I Jeliazkov, Marginal likelihood from the Metropolis-Hastings output. J. Am. Stat. Assoc. **96**, 270–281 (2001)
60. R Neal, *Chapter 5 of the Handbook of Markov Chain Monte Carlo*. (S Brooks, A Gelman, G Jones, X-L Meng, eds.) (Chapman and Hall/CRC Press, 2011)

61. IT Nabney, *Netlab: Aalgorithms for Pattern Recognition*. (Springer, 2008)
62. C Bishop, *Pattern Recognition and Machine Learning*. (Springer, 2006)
63. H Haario, E Saksman, J Tamminen, Component-wise adaptation for high dimensional MCMC. Comput. Stat. **20**, 265–273 (2005)
64. L Martino, R Casarin, D Luengo, Sticky proposal densities for adaptive MCMC methods. IEEE Workshop on Statistical Signal Processing (SSP) (2016)
65. PJ Davis, *Interpolation and approximation*. (Courier Corporation, 1975)
66. GH Hardy, JE Littlewood, G Pólya, *Inequalities*. (Cambridge Univ. Press, 1952)