The Nearest Neighbor entropy estimate: an adequate tool for adaptive MCMC evaluation

Didier Chauveau, Pierre Vandekerkhove

The Nearest Neighbor entropy estimate: an adequate tool for adaptive MCMC evaluation

Didier Chauveau* Pierre Vandekerkhove†‡

September 24, 2014

Abstract

Many recent and often adaptive Markov Chain Monte Carlo (MCMC) methods are associated in practice to unknown rates of convergence. We propose a simulation-based methodology to estimate MCMC efficiency, grounded on a Kullback divergence criterion requiring an estimate of the entropy of the algorithm successive densities, computed from iid simulated chains. We recently proved in Chauveau and Vandekerkhove (2013) some consistency results in MCMC setup for an entropy estimate based on Monte-Carlo integration of a kernel density estimate based on Györfi and Van Der Meulen (1989). Since this estimate requires some tuning parameters and deteriorates as dimension increases, we investigate here an alternative estimation technique based on Nearest Neighbor estimates. This approach has been initiated by Kozachenko and Leonenko (1987) but used mostly in univariate situations until recently when entropy estimation has been considered in other fields like neuroscience. Theoretically, we prove that, under certain uniform control conditions, the successive densities of a generic class of Adaptive Metropolis-Hastings algorithms to which most of the strategies proposed in the recent literature belong can be estimated consistently with our method. We then show that in MCMC setup where moderate to large dimensions are common, this estimate seems appealing for both computational and operational considerations, and that the problem inherent to a non neglictible bias arising in high dimension can be overcome. All our algorithms for MCMC simulation and entropy estimation are implemented in an R package taking advantage of recent advances in high performance (parallel) computing.

keywords Adaptive MCMC algorithms, Bayesian model, entropy, Kullback divergence, Metropolis-Hastings algorithm, nearest neighbor estimation, nonparametric statistic.

1 Introduction

A Markov Chain Monte Carlo (MCMC) method generates an ergodic Markov chain for which the stationary distribution is a given probability density function (pdf) f. For common Bayesian inference, f is a posterior distribution of the model parameter θ over a state space $\Theta \subseteq \mathbb{R}^d$. This posterior is typically known only up to a multiplicative normalizing constant, and simulation or integration w.r.t. f are approximated by ergodic averages from the chain. The Metropolis-Hastings (MH) algorithm (Hastings, 1970; Metropolis et al., 1953) is one of

*Univ. Orléans, CNRS, MAPMO, UMR 7349, Orléans, France, didier.chauveau@univ-orleans.fr

†LAMA - CNRS UMR 8050, Université de Marne-la-Vallée, 5, boulevard Descartes, Cité Descartes - Champs-sur-Marne, 77454 Marne-la-Vallée, France.

‡UMI Georgia Tech - CNRS 2958, School of aerospace, Georgia Institute of Technology, 270 Ferst Drive Atlanta GA 30332-0150, USA.

the most popular algorithm used in MCMC methods. Another commonly used method is the Gibbs sampler introduced by Geman and Geman (1984).

Each step of a MH algorithm at a current position θ^t is based on the generation of the proposed next move from a general *proposal density* $q(\cdot|\theta^t)$. Historically, two popular MH strategies used to be (i) the *Independence Sampler* (MHIS), which uses a proposal distribution independent of the current position, and (ii) the Random Walk MH algorithm (RWMH), for which the proposal is a random perturbation of the current position, most often drawn from a Gaussian distribution with a fixed variance matrix that has to be tuned.

To actually implement a MCMC algorithm, many choices for the proposal density are possible, with the goal of improving mixing and convergence properties of the resulting Markov chain. For instance running a RWMH strategy requires the determination of a "good" scaling constant, since the mixing depends dramatically on the variance matrix of the perturbation (Roberts and Rosenthal, 2001). As a consequence, a growing interest in new methods appeared this last decade, which purpose is to optimize in sequence the proposal strategy in MCMC algorithms on the basis of the chain(s) history; see, e.g., Andrieu and Thoms (2008) for a recent survey. These approaches called adaptive Markov Chain Monte Carlo (AMCMC) can be described (not in an entirely general way) as follows: let f be the pdf of interest and suppose that we aim to simulate efficiently from f given a family of Markov kernels $\{P_\vartheta, \vartheta \in \mathcal{E}\}$. This can be done adaptively using a joint process $(\theta^t, \vartheta^t)_{t\geq 0}$ such that the conditional distribution of θ^{t+1} given the information available up to time t is a kernel P_{ϑ^t} where ϑ^t is an Euclidean parameter tuned over time to fit a supposed relevant strategy. Some general sufficient conditions insuring convergence (essentially ergodicity and the strong law of large numbers) of such algorithms have been established by various authors, see Andrieu and Thoms (2008). These conditions are informally based on the two following ideas.
Containment: for any (θ^0, ϑ^0), and any $\varepsilon > 0$, the stochastic process $(M_\varepsilon(\theta^t, \vartheta^t))_{t\geq 0}$ is bounded in probability, where

$$M_\varepsilon(\theta, \vartheta) = \inf\left\{t \geq 1 \ : \ \|P^t_\vartheta(\theta, \cdot) - f(\cdot)\|_{TV} \leq \varepsilon\right\}$$

is the "ε-time to convergence".
Diminishing Adaptation: for any (θ^0, ϑ^0), $\lim_{t\to\infty} D_t = 0$ in $\mathbb{P}_{\theta^0,\vartheta^0}$-probability, where

$$D_t = \sup_{\theta\in\Theta} \|P_{\vartheta^{t+1}}(\theta, \cdot) - P_{\vartheta^t}(\theta, \cdot)\|_{TV},$$

represents the amount of adaptation performed between iterations t and $t + 1$.

Note that in Bai et al. (2008) two examples are provided to show that either Diminishing Adaptation or Containment is not necessary for ergodicity of AMCMC, and diminishing Adaptation alone cannot guarantee ergodicity. See also the very simple four-state Markov Chain Example 1 in Rosenthal and Roberts (2007), which illustrates the fact that ergodicity is not an automatic heritage when adapting a Markov Chain from its past.

These various and sometimes experimental algorithmic choices are associated in general to unknown rates of convergence because of the complexity of the kernel, and the difficulty in computing, when available, the theoretical bounds of convergence. For instance, Bai et al. (2010) compare two AMCMC strategies in dimension $d \leq 5$, and Vrugt et al. (2009) compare two AMCMC's against some benchmark in dimension $d = 10$. More recently Fort et al. (2014) define the best interacting ratio for a simple equi-energy type sampler, by minimizing the corresponding limiting variance involved in the Central Limit Theorem (see Fig. 1 in Fort et al. (2014)). There are also recent works proposing tools or methods for MCMC comparisons, showing that these questions are crucial in nowaday MCMC application and

research. Thompson (2010) proposes the R package SamplerCompare for comparing several MCMC's differing by a single tuning parameter, using standard evaluation criterion.

In this paper, we propose a methodological approach, and corresponding software tool, only based on Monte Carlo simulation (i.e. not requiring a theoretical study typically MCMC and/or target-specific) with two goals: (i) For MCMC users to easily select a good sampler among possible candidates; (ii) For researchers to better understand which (A)MCMC methods perform best in which circumstances. Let

$$\mathcal{H}(p) := \int p \, \log p = \mathbb{E}_p(\log p) \tag{1}$$

be the differential entropy of a probability density p over Θ, and p^t be the marginal density of the (A)MCMC algorithm at "time" (iteration) t. Our approach is grounded on a criterion which is the evolution of the Kullback-Leibler divergence between p^t and f,

$$t \mapsto \mathcal{K}(p^t, f) := \int p^t \log\left(\frac{p^t}{f}\right) = \mathcal{H}(p^t) - \int p^t \, \log f.$$

This Kullback "distance" is indeed a natural measure of the algorithm's quality and has strong connections with ergodicity of Markov chains and rates of convergence, see Harremoes and Holst (2007) for recent results. In MCMC setup, Chauveau and Vandekerkhove (2013) showed that if the proposal density of a Metropolis-Hastings algorithm satifies a uniform minorization condition implying its geometric convergence as in Holden (1998), then $\mathcal{K}(p^t, f)$ also decreases geometrically.

In Section 2, we detail our approach which is methodological but relies on estimation techniques that have been proved to be theoretically consistent in simple cases like Gaussian unidimensional RWMH or independent samplers (Chauveau and Vandekerkhove, 2013). Our estimation of $\mathcal{H}(p^t)$ is grounded on the simulation of N *parallel* (iid) chains. In Section 3, we prove the consistency of our entropy estimate based on Nearest Neighbor (NN) estimate from Kozachenko and Leonenko (1987), adapted to our adaptive MCMC setup. Section 4 illustrates the good behavior of our criterion on synthetic multi-dimensional examples. These examples also allow us to show the difficulty due to a bias coming from the curse of dimension in nonparametric statistical estimation. In Section 5 we detail our methodological solution for handling that bias problem, in such a way that our approach being still usable even in large dimension, in practice for Bayesian models with dozens of parameters.

2 Entropy and Kullback estimation in MCMC context

Recent motivations for entropy estimation in other fields like molecular science appeared recently in the literature (see, e.g. Singh et al., 2003), and are concerned by estimation of $\mathcal{H}(p)$ for multivariate densities p. Most of the estimation techniques proved to be consistent under various conditions are based on iid samples from p. There exists some results about entropy estimation for dependent sequences, but these heavily rely on the mixing properties of these sequences themselves, that are precisely what we want to capture by our simulation-based approach without theoretical investigations concerning mixing properties of the Markov kernel. More importantly, these approaches could be used to estimate $\mathcal{H}(f)$ but cannot estimate $\mathcal{H}(p^t)$ for each t.

2.1 Simulation of iid copies of the (A)MCMC algorithm

Our approach is consequently based on the simulation of N parallel (iid) copies of (Adaptive) Markov chains started from a diffuse initial distribution p^0 and using the transition kernel defined by the MCMC strategy under investigation. The N chains, started from $\theta_1^0, \ldots, \theta_N^0$ $\text{iid}\sim p^0$, are denoted

$$\begin{array}{rcl} \text{chain } \#\ 1 & : & \theta_1^0 \to \theta_1^1 \to \ \cdots \to \theta_1^t \sim p^t \to \cdots \\ \vdots & & \vdots \\ \text{chain } \#\ N & : & \theta_N^0 \to \theta_N^1 \to \ \cdots \to \theta_N^t \sim p^t \to \cdots \end{array}$$

where "$\to$" indicates the (eventually non-homogeneous) Markov dependence. At "time" (iteration) t, the locations of the N simulated chains $\boldsymbol{\theta}^t = (\theta_1^t, \ldots, \theta_N^t)$ forms a N-sample $\text{iid}\sim p^t$.

In an experimental framework where one wants to evaluate a new (A)MCMC algorithm the target f often corresponds to a benchmark example, hence is completely known (as e.g., in Vrugt et al., 2009). In this case a strongly consistent estimate of $\int p^t \log f$ is given by Monte Carlo integration and the Strong Law of Large Numbers,

$$\hat{p}_N^t(\log f) = \frac{1}{N} \sum_{i=1}^{N} \log f(\theta_i^t), \tag{2}$$

so that estimation of $\mathcal{K}(p^t, f)$ is in turn accessible provided $\mathcal{H}(p^t)$ is. However, if the objective is to evaluate an experimental MCMC method for an actual Bayesian model for which f is a posterior density proportional to the likelihood, say $f(\cdot) \propto \phi(\cdot)$ where the normalization constant is not known, only $\hat{p}_N^t(\log \phi)$ is accessible. We will see that this is not really a flaw since ϕ itself retains all the specificity (shape, modes, tails, ...) of f, and since we are mostly interested in the stabilization in t of $\mathcal{K}(p^t, f)$, not necessarily in knowing its limiting value, as will be detailed in Section 5. In addition, the normalization problem can be eliminated by comparing the MCMC under study to a benchmark MCMC algorithm (e.g., a gaussian RWMH) for the same target f. Indeed, considering two MCMC strategies leading to two sequences of marginal densities, say $(p_1^t)_{t\geq 0}$ and $(p_2^t)_{t\geq 0}$ allows the *difference* of the divergences to be accessible to estimation since

$$D(p_1^t, p_2^t, f) = \mathcal{K}(p_1^t, f) - \mathcal{K}(p_2^t, f) = \mathcal{H}(p_1^t) - \mathcal{H}(p_2^t) + \mathbb{E}_{p_2^t}[\log \phi] - \mathbb{E}_{p_1^t}[\log \phi]. \tag{3}$$

The Kullback criterion is the only usual divergence insuring this property and, in addition to its connection with ergodicity, it motivates our choice. Note also that the Kullback divergence is currently used as a criterion in other simulation approaches, see Douc et al. (2007). The choice of this estimate also has the advantage of avoiding numerical integration in moderate or high dimensional spaces (replaced by Monte Carlo integration), in contrary to other criterion such as the L^1-distance.

For estimating the entropy $\mathcal{H}(p^t)$ a classical, plug-in approach, is to build a nonparametric kernel density estimate of p^t, and to compute the Monte Carlo integration of this estimate. Techniques based on this approach have been suggested by Ahmad and Lin (1989), and studied by many authors under different assumptions (see, e.g., the survey paper Beirlant et al., 1997). Several consistency and asymptotic normality results pertaining to this approach have been proved (see references in Eggermont and LaRiccia, 1999). However, most of these are not suitable to estimate $\mathcal{H}(p^t)$ even in the simplest MH cases, either because they do not apply to multivariate densities, or because they require smoothness conditions that are far too

restrictive to be proved for the sequences of densities p^t we have to consider here. Up to our best knowledge, the unique consistency result applicable in this MCMC simulation context is the one proved in Györfi and Van Der Meulen (1989), that essentially requires a Lipschitz type smoothness condition. Indeed, for that approach, Chauveau and Vandekerkhove (2013) have proved that adequate smoothness and tail conditions on the "input ingredients" of the MH algorithm (namely p^0, q and f) propagate a Lipschitz condition to the successive marginals p^t, $t = 1, \ldots, n$, so that the sequence of $(\mathcal{H}(p^t))_{t=1,\ldots,n}$ can be consistently estimated. These technical conditions have been proved to hold in simple univariate IS and RWMH cases, but are not meant to be verified in general, since it would require tedious (and often unfeasible) calculations.

2.2 Estimates based on nearest neighbor distances

The plug-in estimate presented above requires the tuning of several parameters: a certain threshold for truncating the data over the tails of p^t, the choice of the kernel and the difficult issue of the appropriate bandwidth matrix, particularly in high dimensions. All these issues motivated us to find an alternative, and study the behavior of the somehow simpler Nearest Neighbor (NN) estimate initiated by Kozachenko and Leonenko (1987) (see also Beirlant et al., 1997, for a survey on these entropy estimates). In our setup, based on the sample $\boldsymbol{\theta}^t \overset{\text{iid}}{\sim} p^t$ in dimension d, this NN entropy estimate is

$$\hat{\mathcal{H}}_N(p^t) = \frac{1}{N}\sum_{i=1}^{N} \log(\rho_i^d) + \log(N-1) + \log(C_1(d)) + C_E, \tag{4}$$

where $C_E = -\int_0^\infty e^{-u} \log u \, du \approx 0.5772\ldots$ is the Euler constant, $C_1(d) = \frac{\pi^{d/2}}{\Gamma(d/2+1)}$ and where

$$\rho_i = \min\{\rho(\theta_i^t, \theta_j^t), j \in \{1, 2, \ldots, N\}, j \neq i\}$$

is the (Euclidean) distance $\rho(\cdot,\cdot)$ from the ith point to its nearest neighbor in the sample $\boldsymbol{\theta}^t$. Kozachenko and Leonenko (1987) proved the mean square consistency of the NN entropy estimate $\hat{\mathcal{H}}_N(h)$ as in (4) for a density h and any dimension d under the following Peak and Tail (**P&T**) conditions.

Definition 1. *A density h over $\mathbb{R}^d$ satisfies the* **P&T conditions** *if there exists $\alpha > 0$ such that*

$$\int |\log h(x)|^{2+\alpha} h(x)\, dx < +\infty \tag{5}$$

$$\int\int |\log \rho(x,y)|^{2+\alpha} h(x)h(y)\, dx\, dy < +\infty. \tag{6}$$

This NN estimate seems more appealing than kernel density estimates in our situation, both from an operational point of view (no tuning parameters like the threshold and bandwidth), and from a computational point of view (the nearest distance can be computed faster than a multivariate kernel density estimate in high dimension). Until recently, this nearest neighbor approach has been used and studied mostly in univariate or bivariate ($d = 2$) situations, like in image processing. One interest of this study is to investigate its behavior in higher dimensions and for MCMC sequences of marginals.

3 Peak and tail conditions in adaptive MH case

In this section we introduce a generic class of Adaptive Metropolis-Hastings (AMH) algorithms to which most of the AMCMC strategies proposed in the recent literature belong (see, e.g., Andrieu and Thoms, 2008). We prove that, under certain uniform control conditions, the P&T conditions from Definition 1 required to estimate the entropy of successive densities p^t by the NN approach from Kozachenko and Leonenko (1987), hold for each fixed iteration $t \in \mathbb{N}$.

As recalled in the introduction, an AMCMC algorithm relies on a family of Markov kernels based on a joint process $(\theta^t, \vartheta^t)_{t\geq 0}$ such that the conditional distribution of θ^{t+1} given the information available up to time t is a kernel P_{ϑ^t} where ϑ^t is a Euclidean parameter depending on the past. In the case of a generic Adaptive MH processes $(X^t)_{t\geq 0}$ valued in $\Omega \subseteq \mathbb{R}^d$, each MH step at time t is based on the generation of the proposed next move y from an adapted *proposal density* $q_{\vartheta_t}(y) \in \mathcal{F} := \{q_\vartheta | \vartheta \in \Theta\}$, where $\vartheta_t := \vartheta(x_0^t)$ is a strategically tuned parameter possibly integrating the whole past trajectory denoted $x_0^t = (x^0, \ldots, x^t)$.

For a starting value $x^0 \sim p^0$, the t-th step $x^t \to x^{t+1}$ of the AMH algorithm is as follows:

`1. generate` $y \sim q_{\vartheta_t}(\cdot)$

`2. compute` $\alpha_{\vartheta_t}(x^t, y) = \min\left\{1, \dfrac{f(y)q_{\vartheta_t}(x^t)}{f(x^t)q_{\vartheta_t}(y)}\right\}$

`3. take` $x^{t+1} = \begin{cases} y & \texttt{with probability } \alpha_{\vartheta_t}(x^t, y) \\ x^t & \texttt{with probability } 1 - \alpha_{\vartheta_t}(x^t, y). \end{cases}$

The proposition below gives the convergence of our NN entropy estimates for the successive AMH marginal densities.

Proposition 1. *If there exist nonnegative functions (φ_1, φ_2) both defined on Ω, a constant $a \in (0,1)$ and $\alpha > 0$ such that:*

i) $C_1 = \int \varphi_1(x)dx < \infty$, $C_2 = \displaystyle\int \frac{\varphi_1^2(x)}{\varphi_2(x)}dx < \infty$, *and* $C_3 = \int \varphi_2(x)dx < \infty$

ii) $\varphi_1 \leq p^0 \leq \varphi_2$, *and* $\varphi_1 \leq f$

iii) $af \leq q_\vartheta \leq \varphi_2$ *for all* $\vartheta \in \Theta$

iv) $\displaystyle\int \left|\log\left(C\frac{\varphi_1^2(x)}{\varphi_2(x)}\right)\right|^{2+\alpha} \varphi_2(x)dx < +\infty$ *for any constant* $C > 0$

v) $\displaystyle\int |\log\left(C\varphi_2(x)\right)|^{2+\alpha} \varphi_2(x)dx < +\infty$ *for any constant* $C > 0$

vi) $\displaystyle\int\int |\log \rho(x,y)|^{2+\alpha} \varphi_2(x)\varphi_2(y)dxdy < +\infty$,

then the successive densities of the Adaptive MH algorithm described above satisfy the P&T conditions (5)–(6).

Note that the constant a introduced above can be understood as the minorization constant used in, e.g., Holden (1998) and Mengersen and Tweedie (1996) conditions of geometric ergodicity.

Proof. For all $t \geq 0$, we define $P_{\vartheta_t}(x^t, \cdot)$, the generic adaptive transition kernel depending on $\vartheta_t = \vartheta(x_0^t)$:

$$\begin{aligned} P_{\vartheta_t}(x^t, dy) &= q_{\vartheta_t}(y)\alpha_{\vartheta_t}(x^t, y))dy \\ &\quad + \left[1 - \int q_{\vartheta_t}(z)\alpha_{\vartheta_t}(x^t, z)dz\right] \delta_{x^t}(dy). \end{aligned}$$

We denote as before by p^t the marginal density of the AMH algorithm at iteration t. Define first the two nonnegative functions controlling p^t from Lemma 1 in Appendix A.2, Equations (15) and (16) using conditions (i)–(iii) of Proposition 1:

$$\begin{aligned} A^t(x) &:= a^{2t} C_1 C_2^{t-1} \frac{\varphi_1^2(x)}{\varphi_2(x)} \\ B^t(x) &:= 2(C_3 + 1)^{t-1} \varphi_2(x) \\ A^t(x) &\leq p^t(x) \leq B^t(x), \quad t \geq 1. \end{aligned}$$

To prove that p^t, for a fixed step t, satisfies the first P&T condition (5) from Definition 1, set the domain $D^t = \{x \in \mathbb{R}^d : p^t(x) < 1\}$, $\bar{D}^t = \mathbb{R}^d \setminus D^t$, and let $\beta = (2 + \alpha)$. We have

$$\int |\log p^t(x)|^\beta \, p^t(x) \, dx = \int_{D^t} |\log p^t(x)|^\beta \, p^t(x) \, dx + \int_{\bar{D}^t} |\log p^t(x)|^\beta \, p^t(x) \, dx. \tag{7}$$

Since for $x \in D^t$, $0 \leq |\log p^t(x)| = -\log p^t(x) \leq -\log A^t(x) = |\log A^t(x)|$,

$$\int_{D^t} |\log p^t(x)|^\beta \, p^t(x) \, dx \leq \int_{D^t} |\log A^t(x)|^\beta \, B^t(x) \, dx.$$

This last integral is finite from condition (iv) since

$$\int_{D^t} |\log A^t(x)|^\beta \, B^t(x) \, dx \leq \tilde{C} \int \left|\log\left(C \frac{\varphi_1^2(x)}{\varphi_2(x)}\right)\right|^\beta \, \varphi_2(x) \, dx < \infty,$$

where $C = a^{2t} C_1 C_2^{t-1}$ and $\tilde{C} = 2(C_3 + 1)^{t-1}$ are both finite from condition (i). For the rightmost integral in (7), it suffices to note that

$$\int_{\bar{D}^t} |\log p^t(x)|^\beta \, p^t(x) \, dx \leq \int_{\bar{D}^t} |\log B^t(x)|^\beta B^t(x) \, dx \leq \tilde{C} \int |\log(\tilde{C}\varphi_2(x))|^\beta \varphi_2(x) \, dx,$$

where the rightmost integral is finite from condition (v). The P&T condition (6) for p^t is obtained straightforwardly from condition (vi). □

4 Experiments and simulations

All the estimation techniques and MCMC evaluation criterion presented in the previous Sections are based on intensive simulations and computations for which we provide a software tool implemented in the EntropyMCMC package for the R statistical software (R Core Team, 2013), taking advantage of recent advances in High Performance Computing, that will be publicly available in a near futur. This package includes some predefined target distributions and standard MCMC samplers, easy definition of additional ones, functions for running simulations, estimating entropy and Kullback divergences, results visualization and sampler comparison. For instance, Figs 2 and 3 have been done using simply a default `plot()` command from this package. The parallel simulations can be done from inside the package, or imported from external files.

4.1 Multidimensional Gaussian target density and iid sampler

Several authors, mostly from biology, statistics and information theory have recently shown some evidence that estimation of functionals like the entropy suffers from the curse of dimensionality (see for instance, Stowell and Plumbley (2009) and Sricharan et al. (2013)). In these studies, the bias seems much more affected by the dimension than the variance.

To confirm this and also validate our software tool, we ran some experiments in Gaussian situations. Let us denote by $\mathcal{N}_d^{\mu,\Sigma}$ the pdf of a d-multivariate Gaussian $\mathcal{N}_d(\mu, \Sigma)$ with mean vector μ and covariance matrix Σ. One reason for choosing Gaussian targets is that the true entropy is known in this case,

$$\mathcal{H}(\mathcal{N}_d^{\mu,\Sigma}) = -\frac{d(\log(2\pi) + 1) + \log(\det(\Sigma))}{2}. \tag{8}$$

In this section and in Section 5, we use centered Gaussian distributions with spherical covariances matrices, and then simply denote by $\mathcal{N}_d^\sigma$ the pdf of $\mathcal{N}_d(0\mathbf{1}_d, \sigma^2\mathbb{I}_d)$, where $\mathbf{1}_d$ is the unit column vector of size d, and $\mathbb{I}_d$ is the $d \times d$ identity matrix. For these experiments we coded a "fake" MCMC algorithm, i.e. we defined as a MCMC algorithm a simple iid sampler from a completely known target such as $\mathcal{N}_d^\sigma$. In other words, $p^t \equiv \mathcal{N}_d^\sigma$ for $t \geq 1$, so that running n iterations of N iid "chains" from this algorithm corresponds exactly to simulating n replications of N iid observations $\sim \mathcal{N}_d^\sigma$, since there is no dependence with time.

Here, we try our entropy estimate on the simplest possible target: the pdf of the standard Gaussian $\mathcal{N}_d^1$. This allowed us to compare the entropy estimate based on the Monte-Carlo (MC) integration of the known $\log(\mathcal{N}_d^1)$ from (2) and the NN estimates of $\mathcal{H}(\mathcal{N}_d^1)$ from (4). In addition to knowing the true entropy from (8), another advantage is that we can reasonably assume that the gaussian density $p^t \equiv \mathcal{N}_d^1$ satisfies the P&T conditions required by the NN entropy estimates (see a proof in Appendix A.3 for the univariate case). Our results also show numerical evidence of consistency.

We ran this model for dimensions between 3 and 30, and N ranging from 500 to $30,000$ observations. Examples of typical results for $d = 20$ are provided in Table 1. We can see that the bias of the NN estimation is much more affected by the dimension than the variance. Remember that it is "unfair" to compare the MC estimator with the NN estimator, since the former is a simple application of the Law of Large Numbers for a known $\log(\mathcal{N}_d^1)$, whereas in the latter the density itself is estimated from the sample. We also observed that in this case, the bias is negative for large d, leading to the under-estimation of $\mathcal{H}(\mathcal{N}_d^1)$. Bias considerations in actual (A)MCMC setups will be discussed more deeply in Section 5.

Table 1: 100×Bias and standard dev's for estimation of $\mathcal{H}(\mathcal{N}_d^1)$ over $n = 100$ replications of samples of size N, for the $d = 20$ Gaussian target. Here the true $\mathcal{H}(\mathcal{N}_d^1) = -28.38$.

N	MC bias	MC sd	NN bias	NN sd
500	1.6296	13.4395	-101.0047	15.2934
1000	-0.6200	10.4458	-88.8441	12.1195
5000	-0.0637	4.4483	-60.6133	5.4353
10,000	-0.4023	3.0028	-52.0266	3.3988
20,000	-0.0048	2.2053	-44.0156	2.6230
30,000	0.3701	1.9592	-39.2813	2.2468

4.2 Multidimensional Gaussian mixtures

We illustrate here our methodology on a synthetic but more complex target density: a 3-components mixture of multivariate, d-dimensional Gaussian distributions,

$$f(\boldsymbol{x}) = \sum_{j=1}^{3} \lambda_j \mathcal{N}_d^{\mu_j, \Sigma_j}(\boldsymbol{x}),$$

The three weights are set to $\lambda_j = 1/3$, the mean vectors and (spherical) covariance matrices are set to (with the notations introduced in Section 4.1)

$$\mu_1 = 0\mathbf{1}_d, \quad \mu_2 = 4\mathbf{1}_d, \quad \mu_3 = -4\mathbf{1}_d, \quad \Sigma_j = j\mathbb{I}_d, \quad j = 1, 2, 3.$$

The advantage of such a synthetic model is that it is defined for any dimension, but the complexity of the target increases with d (the distance between modes increases and the modes get more and more separated and spiked). Note that here the normalizing constant of f is known, so that theoretically the sequence of marginals p^t from a proper (converging) MCMC satisfies $\mathcal{K}(p^t, f) \to 0$ as $t \to \infty$. We compare several standard, well known MCMC algorithms for recovering this target:

- Two RWMH with Gaussian proposals $\mathcal{N}_d(\theta^t; \sigma^2\mathbb{I}_d)$ resulting in slow or fast MCMC's depending on the magnitude σ^2 of their (spherical) variance matrix that we set to $\sigma^2 = 1$ (algorithm called RW1 in the sequel) and 4 (RW4).
- Three Independence Samplers (IS) with Gaussian proposals $\mathcal{N}_d^{\sigma}$ and the choices $\sigma^2 = 2$ (IS2 in the sequel), $\sigma^2 = 9$ (IS9) and 16 (IS16).

The idea driving the choices above for the tuning parameters of the candidate MCMC's is that we want to compare fast, slow and even MCMC's not converging in practice (i.e., in a feasible amount of iterations). For this Gaussian mixture target in small dimensions ($d \leq 2$), it is easy to figure out how to obtain such algorithms. For instance, Fig. 1 displays the target and five proposal densities for the one-dimensional case. For the IS's, the practical support of the proposal density, within which each proposed next move lies, is suppose to include the region of interest of the target. This is definitely not the case for $\sigma^2 = 2$, hence IS2 can be viewed in practice as a non converging algorithm. Larger variances should lead to better algorithms, but it is not easy to tell which value of σ^2 is the best choice. Also, the IS converges geometrically if the proposal have heavier tails than the target, which is not the case here.

Our method then shows how these strategies behave in higher dimension, where the three components get more and more separated and spiked. We detail some results of our experiments, associated to Figs 2 and 3.

Fig.2: In small dimensions like $d = 2$ (top panel), our criterion delivers the right answer straightforwardly, since there is no noticeable bias in the estimates, even with $N = 500$ chains. All the convergent MCMC's stabilized well before $n = 1000$ iterations. IS2 is not converging and is almost stabilized to a non-zero value in these $n = 1000$ iterations. Similar runs for more iterations show clearly its non convergence, and increasing N up to say 1000 reduces the variance, resulting in easier-to-read diagnostics. The bottom panel displays a similar experiment, but now in dimension $d = 10$, and for $N = 5000$ chains to illustrate the variance reduction resulting is smooth curves (so many chains are not needed to get a readable diagnostic). The two RW's converge, but note that RW4 is less performant than RW1 in this

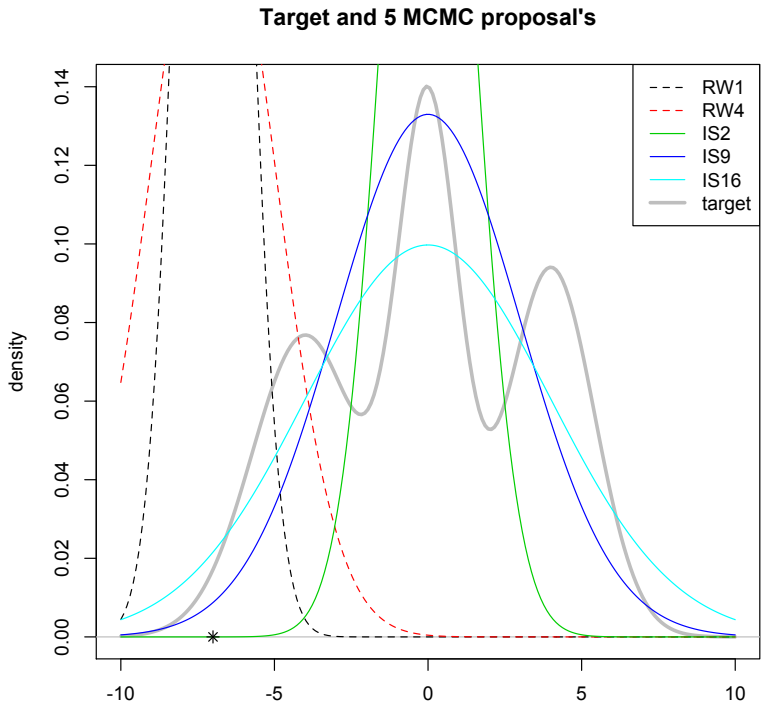

Figure 1: *Target density (bold grey), IS proposal densities (solid lines), and RWHM proposal densities (dashed lines, located on the arbitrary current position* $*$*, for* $d = 1$*.*

higher dimension. IS2 is stabilized away from zero, indicating non convergence. It is hard to tell in 1000 iterations wether IS9 and IS16 will converge at some point, but 1000 iterations are sufficient to tell that these are slow, comparing with the RW's fast stabilizations. Actually running the experiment up to $n = 10,000$ show that IS16 is slow but converging, whereas IS9 convergence is not clear. This is because the proposal "almost covers" the region of interest of the mixture.

Fig.3: As expected, the scale of the number of iterations required to detect stabilization increases with the difficulty associated to the dimension, hence in this figure two simulations in the $d = 20$ dimensional case have been ran up to $n = 10,000$ iterations. The difference between top and bottom panels is just the number of iid chains, $N = 500$ (top) and $N = 10,000$ (bottom). This last number has been chosen intentionnally huge to illustrate the variance reduction, bias difficulty, and the fact that running so many chains is feasible but not needed to obtain a meaningful criterion. Note that this example with $d = 20$, $n = 10,000$ and $N = 500$ only requires about about 25mn of CPU time per algorithm on a 12-cores single workstation. The same example in dimension $d = 30$ required 32mn of CPU time.

One purpose of Fig.3 is precisely to illustrate the bias problem. Indeed, if we look at the top panel and the three IS's only, we conclude that IS16 and IS9 are non or very slowly converging, and that IS2 quickly stabilizes near (and above) 0. Hence if we were only comparing algorithms for stabilization near 0, we would falsely conclude that IS2 is the preferable algorithm. But we already know that IS2 is not a convergent MCMC in this synthetic example. This bizarre

results is due to a bias in $\mathcal{H}(p^t)$ estimation. Looking now at the curves for the RW's, we see that these stabilize on a common negative value, which is theoretically impossible since $\mathcal{K}(p^t, f) \geq 0$. Hence there is a negative bias, more prominent in the top panel with just $N = 500$ chains. Looking at the bottom panel, we see that estimating the entropy from samples of size $N = 10,000$ has the effect of reducing the variance giving more accurate curves, but only slightly reducing the bias: all the curves are just shifted from a small amount (enough to tell that IS2 is actually not convergent). The difference between these two plots illustrate the fact that the bias reduces dramatically slowly with N, and that since all the stabilization values are biased, we cannot rely on stabilization near 0 to assess convergence or lack of convergence.

There are two reasons allowing us to evaluate the competing algorithms in view of Fig.3 top: (i) the property of the Kullback divergence says that $\mathcal{K}(p^t, f)$ decreases when $p^t \to f$, so that if the bias is of the same order in all the algorithms, the faster decay and smaller stabilization value is associated to the best algorithm; (ii) we have a prior knowledge that the RW's are convergent MCMC's, without knowing their rates of convergence. The two RW's can be viewed here as convergent *benchmarks*, and since they stabilize at the same value, it means that their biases are quite identical. To summarize, we can conclude that RW1 is preferable, RW4 is convergent but slower (probably as a consequence of the more spiked component modes so that its proposal variance is comparatively getting too large), IS2 is non convergent since it stabilizes above the benchmark RW1, and IS9 and IS16 may be convergent but even much slower. This diagnostic comes from the analysis of the top panel only, i.e. $N = 500$ is enough to get a conclusion. The next section details all these methodological questions.

5 How to handle the bias in large dimension

The experiments from Section 4 show that some care shoud be taken in the analysis of the plots of the estimates of $t \mapsto \mathcal{K}(p^t, f)$ delivered by our techniques, particularly in high dimension (say $d \geq 10$) where the bias becomes visible. This effect of the dimension on the bias has actually been already noticed in recent literature since nowadays applications of entropy estimation in other fields require moderate to high dimensions. Our results are in accordance with, for instance, Stowell and Plumbley (2009) and Sricharan et al. (2013). These studies show that in $\mathcal{H}(p)$ estimation, one can expect the variance to decreases as $\mathcal{O}(N^{-1})$ whereas the bias only decreases as $\mathcal{O}(N^{-1/(d+1)})$, which these authors called a "glacially slow" rate, and this phenomenon occurs both for Kernel density and NN-based estimates. These behaviors has been confirmed in our case using an iid sampler for a Gaussian target of known entropy (Section 4.1). Hence, trying to achieve in practice the asymptotic unbiasedness guaranteed by the theory by "just" increasing N is hopeless when d gets large.

Another difficulty comes from the unknown normalizing constant. In our experiment the target f in the Monte-Carlo estimate (2) was entirely known, but in practical situations like Bayesian inference for a parameter θ, f is a posterior distribution only known up to a multiplicative constant, $f(\theta) = C\phi(\theta)$ where $C = (\int \phi(\theta)\, d\theta)^{-1}$ is the (unknown) normalizing constant. Hence what can be actually estimated by our method is $\mathcal{K}(p^t, \phi)$, and we have

$$\mathcal{K}(p^t, \phi) = \mathcal{H}(p^t) - \mathbb{E}_{p^t}(\log \phi) = \mathcal{K}(p^t, f) + \log C.$$

This is why in (3) we noticed that in the (estimate of the) difference between the Kullback divergences issued from two MCMC strategies with marginal densities p_1^t and p_2^t, the unknown $\log C$ cancels out and $D(p_1^t, p_2^t, f) \to 0$ as $t \to \infty$ if both strategies are converging.

More precisely, consider first the asymptotic in t, i.e. along the iterations of the algorithm. Define by p_b^t the successive marginals of a converging MCMC algorithm (e.g., a benchmark algorithm), and by p_n^t the successive marginals of a non converging MCMC. This means that, as $t \to \infty$, $p_b^t \to f$ and $p_n^t \to g \neq f$ so that

$$\begin{aligned} \mathcal{K}(p_b^t, \phi) &\to \log C & (9) \\ \mathcal{K}(p_n^t, \phi) &\to \mathcal{K}(g, f) + \log C > \log C. & (10) \end{aligned}$$

Consider now the estimation $\hat{\mathcal{K}}_N(\cdot, \phi)$ based on a N iid copies of each MCMC. For d small (say $d < 10$), the consistency result applies so that, for any fixed $t \geq 1$,

$$\hat{\mathcal{K}}_N(p^t, \phi) \underset{N \to \infty}{\to} \mathcal{K}(p^t, \phi).$$

Thus (for N large enough) our criterion provides a graphical tool comparing convergent MCMC's from the decays of their $\hat{\mathcal{K}}_N(p^t, \phi)$'s. But it can also be used as a convergence assessment tool due to (9)-(10) which imply that a non-converging MCMC will stabilize *above* a converging one.

For large d, each Kullback estimate is biased. As seen previously this bias, which comes from the estimation of the entropy $\mathcal{H}(p)$ of any d-dimensional pdf p, depends on p and d and will be denoted $\text{bias}_N(p, d)$. We can sketch the behavior of the estimates for fixed but large enough N so that the variance becomes neglictible but the bias still remains, which is what is achievable in practice for d large. Informally, for fixed t, the estimate stabilizes at the theoretical value plus this bias,

$$\hat{\mathcal{K}}_N(p^t, \phi) \approx \mathcal{K}(p^t, f) + \log C + \text{bias}_N(p^t, d).$$

We assume that $\text{bias}_N(p^t, d) \to \text{bias}_N(f, d)$ when $p^t \to f$, which seems fairly reasonnable and implies that $\text{bias}_N(p^t, d) \approx \text{bias}_N(f, d)$ for p^t and f "close enough". This is needed in order to compare decays of $t \mapsto \hat{\mathcal{K}}(p^t, f)$ before convergence, and is supported by numerical evidence from the experiments in Section 4 and other experiments detailed below. Intuitively, even a slow MCMC, if converging, should never been too far from its target, leading to similar biases. Finally, for our two strategies above, we have *for both N and t large enough*:

$$\begin{aligned} \hat{\mathcal{K}}_N(p_b^t, \phi) &\approx \log C + \text{bias}_N(f, d) & (11) \\ \hat{\mathcal{K}}_N(p_n^t, \phi) &\approx \mathcal{K}(g, f) + \log C + \text{bias}_N(g, d). & (12) \end{aligned}$$

Unfortunately, we cannot assume in this case that $\mathcal{K}(g, f) + \text{bias}_N(g, d) > \text{bias}_N(f, d)$ without additionnal assumptions on g and f; in particular the biases can be of any sign. We thus provide below an additional experiment which purpose is to evaluate, in a Gaussian situation and for several choices of distinc distributions f and g, the typical biases and $\mathcal{K}(g, f)$ we can expect. The idea is to check wether, for $f \neq g$, typical biases may be such that $\text{bias}_N(g, d) \approx \text{bias}_N(f, d)$, and both smaller in comparison with $\mathcal{K}(g, f)$.

Let $\mathcal{N}_d^\sigma$ denote the pdf of the d-dimensional centered multivariate Gaussian with spherical covariance matrix of diagonal variances σ^2 as in Section 4.1. For $\sigma = 1$, the target density $\mathcal{N}_d^1$ corresponds to the standard multivariate Gaussian for which $\text{bias}_N(\mathcal{N}_d^1, d)$ has been investigated in Table 1 from a Monte-Carlo experiment only using the code provided in the EntropyMCMC package. The same code can be used similarly to estimate any $\mathcal{H}(\mathcal{N}_d^\sigma)$. Moreover, the Kullback divergence between two multivariate Gaussian distribution is known and

available in closed form: For $\mathcal{N}_d^{\mu_j,\Sigma_j} = \mathcal{N}_d(\mu_j, \Sigma_j)$, $j = 1, 2$, we have

$$\mathcal{K}(\mathcal{N}_d^{\mu_2,\Sigma_2}, \mathcal{N}_d^{\mu_1,\Sigma_1}) = \frac{1}{2}\left[\mathrm{Tr}\left(\Sigma_1^{-1}\Sigma_2\right) + (\mu_1-\mu_2)^T\,\Sigma_1^{-1}\,(\mu_1-\mu_2) - d - \log\left(\frac{\det(\Sigma_2)}{\det(\Sigma_1)}\right)\right]. \tag{13}$$

Our experiments only involve centered Gaussian for simplicity, and compare $\mathcal{K}(g, f)+\mathrm{bias}_N(g, d)$ against $\mathrm{bias}_N(f, d)$. Since the entropy $\mathcal{H}(\mathcal{N}_d^{\mu,\Sigma})$ is not affected by the mean, but the Kullback divergence is, differences between means would result in even larger $\mathcal{K}(g, f)$'s.

Tables 2 and 3 show the typical results we obtain in the Gaussian case, for dimensions 20 and 40. In these tables, the case $\sigma = 1$ corresponds to the target $f \equiv \mathcal{N}_d^1$, and the other Gaussian pdf's with larger variances to densities $g \neq f$. The column rel.bias gives the relative bias of the entropy estimates

$$\frac{(\hat{\mathcal{H}}_N(\mathcal{N}_d^\sigma) - \mathcal{H}(\mathcal{N}_d^\sigma))}{\mathcal{H}(\mathcal{N}_d^\sigma)}.$$

It is clear that for any fixed dimension, the biases are similar and neglictible in comparison with the Kullback distances. For instance, in the case $d = 40$, $g \equiv \mathcal{N}_d^5$ and running our criterion for $N = 5000$, we obtain (Table 3, row 5) $\mathrm{bias}_N(f, d) = -3.3139$ almost equal to $\mathrm{bias}_N(g, d) = -3.3152$, whereas $\mathcal{K}(g, f) = 23.9$. To summarize, in all the experiments we did, $\mathrm{bias}_N(g, d) \approx \mathrm{bias}_N(f, d)$, and both smaller in comparison with $\mathcal{K}(g, f)$ even for small differences between f and g's.

To summarize, in large dimension, our experiments provide numerical evidence that biases in the entropy estimates are of the same order for different densities with roughly the same shape (this means for convergent $p_b^t \approx f$ but also for non convergent $p_n^t \approx g \neq f$) and neglictible in comparison with $\mathcal{K}(g, f)$'s. Our criterion based ultimately on (11)–(12) can thus compare efficiency of convergent (A)MCMC's, and detect non-convergent algorithms.

Table 2: True entropies for some target pdf $\mathcal{N}_d^\sigma$ and Kullback distances to the reference $\mathcal{N}_d^1$, together with estimated biases and standard deviations (sd) of the NN entropy estimate based on $n = 100$ replications of N parallel chains, in dimension $d = 20$.

N	σ^2	$\mathcal{H}(\mathcal{N}_d^\sigma)$	$\mathcal{K}(\mathcal{N}_d^\sigma, \mathcal{N}_d^1)$	$\widehat{\mathrm{bias}}_N(\mathcal{N}_d^\sigma, d)$	rel.bias(%)	NN sd
1000	1	-28.4	0	-0.8652	3.0489	0.1097
	5	-44.5	23.9	-0.8685	1.9528	0.1160
	10	-51.4	66.97	-0.8913	1.7338	0.1069
5000	1	-28.4	0	-0.6061	2.1356	0.0518
	5	-44.5	23.9	-0.6099	1.3713	0.0558
	10	-51.4	66.97	-0.6019	1.1709	0.0537
10,000	1	-28.4	0	-0.5084	1.7914	0.0376
	5	-44.5	23.9	-0.5138	1.1553	0.0397
	10	-51.4	66.97	-0.5207	1.0129	0.0364

Table 3: True entropies for some target pdf $\mathcal{N}_d^\sigma$ and Kullback distances to the reference $\mathcal{N}_d^1$, together with estimated biases and standard deviations (sd) of the NN entropy estimate based on $n = 100$ replications of N parallel chains, in dimension $d = 40$.

N	σ^2	$\mathcal{H}(\mathcal{N}_d^\sigma)$	$\mathcal{K}(\mathcal{N}_d^\sigma, \mathcal{N}_d^1)$	$\widehat{\text{bias}}_N(\mathcal{N}_d^\sigma, d)$	relbias(%)	NN sd
1000	1	-56.8	0	-4.0243	7.09	0.1823
	5	-88.9	47.8	-3.9921	4.49	0.1644
	10	-103	133.9	-4.0222	3.91	0.1636
5000	1	-56.8	0	-3.3139	5.8387	0.0781
	5	-88.9	47.8	-3.3152	3.7272	0.0765
	10	-103	133.9	-3.3147	3.2242	0.0749
10,000	1	-56.8	0	-3.0610	5.3932	0.0500
	5	-88.9	47.8	-3.0784	3.4610	0.0448
	10	-103	133.9	-3.0574	2.9739	0.0524

6 Conclusion

We have proposed a methodological approach to evaluate (A)MCMC efficiency and control of convergence on the basis of intensive simulation only. The diagnostic is based on a practical, easy-to-understand graphical criterion. To evacuate the difficulty induced by the bias in high dimensions we have introduced a benchmark convergent MCMC which indicates when stabilization means convergence.

Since our method requires intensive simulations that may be computationally demanding, all our algorithms have been implemented in a package named EntropyMCMC for the R statistical software (R Core Team, 2013) that will be publicly available in a near futur. This package takes advantage of recent advances in parallel, High Performance Computing (HPC) using the Rmpi package (Yu, 2012). All the examples shown in this paper have been ran with this package on multicore workstations and the regional cluster CCSC[1]. These simulations (or part of it) from the best sampler are recyclable after comparisons, or can be re-used in the fly for statistical inference.

Theoretically, we have shown that the peak and tail conditions required for the NN-estimates consistency are satisfied by the successive densities of a generic class of MH including Adaptive MH MCMC algorithms, under uniform conditions controlling the adaptation. Like in Chauveau and Vandekerkhove (2013) for a Kernel density based entropy estimate, the difficulty often comes from the fact that a MH kernel has a point mass at the chain's current position. We have then assumed that these conditions were satisfied for the MCMC algorithms used in our experiment, and this have been supported by numerical evidence of consistency.

Recent researchs extend the NN idea to a k-th nearest neighbor distance estimate (Singh et al., 2003), see also Wang and Kulkarni (2009). There are some hope that these extensions together with recent computing strategies for computing approximate k-NN estimates reduce the bias in entropy estimation. However, it also brings back a tuning parameter (how to choose k) that plays somehow the role of the bandwidth in the kernel density estimation approach. These considerations are perspective for futur work.

[1] *Centre de Calcul Scientifique en région Centre*

A Appendix: Controlling the successive marginals

We provide in this Appendix some technical results which allow us to control the successive marginals of some generic AMH algorithms, in order to prove the P&T conditions (5)–(6) in Proposition 1.

A.1 The MH independence sampler case

To help intuition, we start here by showing how successive marginals of a simple independent MH sampler can be controlled using the assumptions (i)–(iii) of Proposition 1 (where q does also simply not depends on the past). Recall that "independent" here means that the proposal density q does not depend on the current position of the chain. Let us denote the probability of accepting the move y from x,

$$\alpha(x,y) = \min\left(1, \frac{f(y)q(x)}{f(x)q(y)}\right).$$

Then

$$\begin{aligned} p^1(y) &= \int p^0(x)q(y)\alpha(x,y)dx + \int p^0(y)q(z)(1-\alpha(y,z))dz \\ &\geq a\varphi_1(y)\left[\int \varphi_1(x)\alpha(x,y)dx\right]. \end{aligned}$$

We have also

$$\alpha(x,y) \geq \min\left(1, \frac{af(y)}{q(y)}\right) \geq \min\left(1, \frac{a\varphi_1(y)}{\varphi_2(y)}\right) = a\frac{\varphi_1(y)}{\varphi_2(y)}$$

since $a\varphi_1 \leq q \leq \varphi_2$. This leads to

$$p^1(y) \geq a^2\frac{\varphi_1^2(y)}{\varphi_2(y)}\int \varphi_1(x)dx = a^2\frac{\varphi_1^2(y)}{\varphi_2(y)}C_1.$$

Iterating, we have

$$p^2(y) \geq q(y)\int p^1(x)\alpha(x,y)dx \geq a^4C_1\left[\int \frac{\varphi_1^2(x)}{\varphi_2(x)}dx\right]\frac{\varphi_1^2(y)}{\varphi_2(y)}. \tag{14}$$

By induction we prove that

$$\begin{aligned} p^t(y) &\geq q(y)\int p^{t-1}(x)\alpha(x,y)dx \\ &\geq a^{2t}\left[\int \varphi_1(x)dx\right].\left[\int \frac{\varphi_1^2(x)}{\varphi_2(x)}dx\right]^{t-1}\frac{\varphi_1^2(y)}{\varphi_2(y)} \\ &= a^{2t}C_1C_2^{t-1}\frac{\varphi_1^2(y)}{\varphi_2(y)}. \end{aligned}$$

To majorize $p^1(y)$ we can simply notice that $p^1(y) \leq q(y) + p^0(y) \leq 2\varphi_2(y)$ and iterate to get $p^t(y) \leq (t+1)\varphi_2(y)$. However this will not hold in the adaptive case.

A.2 The Adaptive MH (AMH) case

We turn now to the case of the AMH generic algorithm defined in Section 3. For more obvious notations, we will not use the common description of an adaptive MCMC algorithm through a Markov kernel indexed by $\vartheta_t = \vartheta(x_0^t)$ as we did previously, but directly by the trajectory from all the past x_0^t to indicate dependence.

Lemma 1. *Let (φ_1, φ_2) be nonnegative functions satisfying conditions (i)–(iii) of Proposition 1, and $q_{x_0^t}(y)$ be an adaptive proposal density depending on the past such that $af \leq q_{x_0^t} \leq \varphi_2$ for any $x_0^t \in \Omega^{t+1}$. Then, for all $y \in \Omega$,*

$$a^{2t} C_1 C_2^{t-1} \frac{\varphi_1^2(y)}{\varphi_2(y)} \leq p^t(y) \tag{15}$$

and

$$p^t(y) \leq 2(C_3+1)^{t-1} \varphi_2(y), \tag{16}$$

where the constants a, C_1, C_2, C_3, are defined in Proposition 1.

Proof. For all $t \geq 1$, we define the generic AMH transition a kernel depending on the past $x_0^{t-1} = (x^0, \ldots, x^{t-1})$:

$$\begin{aligned} P_{x_0^{t-1}}(x^{t-1}, dy) &= q_{x_0^{t-1}}(y) \alpha_{x_0^{t-1}}(x^{t-1}, y) dy \\ &+ \int q_{x_0^{t-1}}(z) \left[1 - \alpha_{x_0^{t-1}}(x^{t-1}, z)\right] dz \, \delta_{x^{t-1}}(y) \, dy \end{aligned} \tag{17}$$

where

$$\alpha_{x_0^{t-1}}(x^{t-1}, y) = \min\left(1, \frac{f(y) q_{x_0^{t-1}}(x^{t-1})}{f(x^{t-1}) q_{x_0^{t-1}}(y)}\right)$$

is the probability of accepting the move y from x^{t-1} in the MH step.

We handle first the minorization part (15). The technique is similar to the simplest independence sampler case of Appendix A.1, except that here we need to minorize the transition kernel itself as follows:

$$P_{x_0^{t-1}}(x^{t-1}, dy) \geq q_{x_0^{t-1}}(y) \alpha_{x_0^{t-1}}(x^{t-1}, y) \, dy.$$

Similarly to the independent sampler case we have:

$$\alpha_{x_0^{t-1}}(x^{t-1}, y) \geq \min\left(1, \frac{af(y)}{q_{x_0^{t-1}}(y)}\right) \geq a \frac{\varphi_1(y)}{\varphi_2(y)},$$

which implies

$$P_{x_0^{t-1}}(x^{t-1}, dy) \geq a^2 \frac{\varphi_1^2(y)}{\varphi_2(y)} \, dy.$$

Proceeding in that way we have the following minorization for the densities:

$$\begin{aligned} p^t(y) \, dy &= \int p^0(x^0) \, dx^0 P_{x^0}(x^0, dx^1) P_{x_0^1}(x^1, dx^2) \ldots P_{x_0^{t-1}}(x^{t-1}, dy) \\ &\geq a^{2t} \int \varphi_1(x^0) \frac{\varphi_1^2(x^1)}{\varphi_2^2(x^1)} \ldots \frac{\varphi_1^2(x^{t-1})}{\varphi_2(x^{t-1})} dx_0^{t-1} \frac{\varphi_1^2(y)}{\varphi_2(y)} \, dy \\ &= a^{2t} C_1 C_2^{t-1} \frac{\varphi_1^2(y)}{\varphi_2(y)} \, dy. \end{aligned}$$

To obtain the majorization (16) of the densities, we notice from (17) that:

$$P_{x_0^{t-1}}(x^{t-1}, dy) \leq q_{x_0^{t-1}}(y)dy + \delta_{x^{t-1}}(y)\, dy \leq \varphi_2(y)dy + \delta_{x^{t-1}}(y)\, dy = \Phi(x^{t-1}, dy),$$

where

$$\Phi(x, dy) := \varphi_2(y)dy + \delta_x(y)\, dy$$

is a non-normalized transition kernel, i.e. $\int \Phi(x, dy) = C_3 + 1$. This leads to

$$\begin{aligned} p^t(y)\, dy &= \int p^0(x^0)\, dx^0 P_{x^0}(x^0, dx^1) P_{x_0^1}(x^1, dx^2) \dots P_{x_0^{t-1}}(x^{t-1}, dy) \\ &\leq \int p^0(x^0)\, dx^0 \Phi(x^0, dx^1)\Phi(x^1, dx^2) \dots \Phi(x^{t-1}, dy). \end{aligned}$$

We can now study separately the right hand side term of the above inequality. For the first step we have:

$$\begin{aligned} p^1(x^1)\, dx^1 &= \int p^0(x^0)\, dx^0 P_{x^0}(x^0, dx^1) \leq \int p^0(x^0)\, dx^0 \Phi(x^0, dx^1) \\ &= \int p^0(x^0) \left[\varphi_2(x^1)dx^1 + \delta_{x^0}(x^1)\, dx^1\right]\, dx^0 \\ &= \varphi_2(x^1) \left[\int p^0(x^0)dx^0\right] dx^1 + \left[\int p^0(x^0)\mathbb{I}_{\{x^1\}}(x^0)\, dx^0\right]\, dx^1 \\ &\leq \varphi_2(x^1)\, dx^1 + p^0(x^1)dx^1 \\ &\leq 2\varphi_2(x^1)\, dx^1. \end{aligned}$$

Similarly for the second step (the integrals being w.r.t. dx^0 and dx^1),

$$\begin{aligned} p^2(x^2)\, dx_2 &= \int \left[\int p^0(x^0) P_{x^0}(x^0, dx^1)\, dx^0\right] P_{x_0^1}(x^1, dx^2) \\ &\leq \int \left[2\varphi_2(x^1)\, dx^1\right]\, \Phi(x^1, dx^2) \\ &\leq \left(\int \varphi_2(x^1)\, dx^1\right) 2\varphi_2(x^2)\, dx^2 + 2\varphi_2(x^2)\, dx^2 \\ &= 2(C_3 + 1)\varphi_2(x^2)\, dx^2. \end{aligned}$$

So that, by induction,

$$p^t(x^t)\, dx^t \leq 2(C_3 + 1)^{t-1}\varphi_2(x^t)\, dx^t.$$

This bound degenerates as $t \to +\infty$ but it is finite for each fixed iteration t.

□

A.3 P&T conditions in the Gaussian case

We check here that the P&T conditions are satisfied when f is the pdf of the standard normal $\mathcal{N}(0, 1)$. P&T condition (5) is obvious. For P&T condition (6), let

$$I := \int_{\mathbb{R}\times\mathbb{R}} \log(|x - y|)^{2+\alpha} f(x)f(y)\, dx\, dy. \tag{18}$$

We remark that

$$\int_{\mathbb{R}\times\mathbb{R}} |\log(|x-y|)|^{2+\alpha} e^{-\frac{1}{2}(|x|^2+|y|^2)} dxdy = I_1 + I_2,$$

where

$$I_1 := \int_{\mathbb{R}} \left(\int_{|x-y|<\varepsilon} |\log(|x-y|)|^{2+\alpha} e^{-\frac{1}{2}(|x|^2+|y|^2)} dy \right) dx,$$

and

$$I_2 := \int_{\mathbb{R}} \left(\int_{|x-y|\geq\varepsilon} |\log(|x-y|)|^{2+\alpha} e^{-\frac{1}{2}(|x|^2+|y|^2)} \right) dx.$$

I_2 is trivially convergent. For I_1, by the change of variable $u = y-x$ and a symmetry argument we have:

$$\begin{aligned} I_1 &= \int_{\mathbb{R}} \left(\int_{|u|<\varepsilon} |\log(|u|)|^{2+\alpha} e^{-\frac{1}{2}(|x|^2+|x-u|^2)} du \right) dx \\ &\leq \int_{\mathbb{R}} \left(\int_{|u|<\varepsilon} |\log(|u|)|^{2+\alpha}\, du \right) e^{-\frac{1}{2}(|x|^2)}\, dx < \infty, \end{aligned}$$

where the last integral in convergent because, for $\varepsilon < 1/e$,

$$\int_{|u|<\varepsilon} |\log(|u|)|^{2+\alpha}\, du = 2\int_0^{\varepsilon} |\log(u)|^{2+\alpha} du \leq 2\int_0^{\varepsilon} \log(u)^4\, du < \infty.$$

References

Ahmad, I. A. and Lin, P. E. (1989). A nonparametric estimation of the entropy for absolutely continuous distributions. *IEEE Trans. Inform. Theory*, 36:688–692.

Andrieu, C. and Thoms, J. (2008). A tutorial on adaptive MCMC. *Stat. Comput.*, 18:343–373.

Bai, Y., Craiu, R. V., and Di Narzo, A. F. (2010). Divide and conquer: A mixture-based approach to regional adaptation for mcmc. *J. Comp. Graph. Stat.*, pages 1–17.

Bai, Y., Roberts, G. O., and Rosenthal, J. S. (2008). On the containment condition for adaptive markov chain monte carlo algorithms. Technical report, Dept. Satist. Univ. Toronto.

Beirlant, J., Dudewicz, E. J., Györfi, L., and van der Meulen, E. C. (1997). Nonparametric entropy estimation, an overview. *Int. J. Math. Stat. Sci.*, 6:17–39.

Chauveau, D. and Vandekerkhove, P. (2013). Smoothness of Metropolis-Hastings algorithm and application to entropy estimation. *ESAIM: Probability and Statistics*, 17:419–431.

Douc, R., Guillin, A., Marin, J., and Robert, C. (2007). Convergence of adaptive mixtures of importance sampling schemes. *Ann. Statist.*, 35(1):420–448.

Eggermont, P. P. B. and LaRiccia, V. N. (1999). Best asymptotic normality of the kernel density entropy estimator for smooth densities. *IEEE trans. Inform. Theory*, 45(4):1321–1326.

Fort, G., Moulines, E., Priouret, P., and Vandekerkhove, P. (2014). A central limit theorem for adaptive and interacting markov chains. *Bernoulli*, 20:457–485.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6:721–741.

Györfi, L. and Van Der Meulen, E. C. (1989). An entropy estimate based on a kernel density estimation. *Colloquia Mathematica societatis János Bolyai 57, Limit Theorems in Probability and Statistics Pécs*, pages 229–240.

Harremoes, P. and Holst, K. K. (2007). Convergence of Markov chains in information divergence. Technical report, Center for Mathematics and Computer Science, Amsterdam.

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

Holden, L. (1998). Geometric convergence of the Metropolis-Hastings simulation algorithm. *Statistics and Probabilitiy Letters*, 39.

Kozachenko, L. and Leonenko, N. N. (1987). Sample estimate of entropy of a random vector. *Problems of Information Transmission*, 23:95–101.

Mengersen, K. and Tweedie, R. (1996). Rates of convergence of the Hastings and Metropolis algorithms. *Ann. Statist.*, 24:101–121.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953). Equations of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1092.

R Core Team (2013). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

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

Rosenthal, J. S. and Roberts, G. O. (2007). Coupling and ergodicity of adaptive mcmc. *J. Appl. Prob*, 44(458–475).

Singh, H., Misra, N., Hnizdo, V., Fedorowicz, A., and Demchuk, E. (2003). Nearest neighbor estimate of entropy. *American Journal of Mathematical and Management Sciences*, 23(3):301–321.

Sricharan, K., Wei, D., and Hero III, A. O. (2013). Ensemble estimators for multivariate entropy estimation. http://arxiv.org/abs/1203.5829v3.

Stowell, D. and Plumbley, M. D. (2009). Fast multidimensional entropy estimation by k-d partitioning. *IEEE Signal Processing Letters*, 16(6):537–540.

Thompson, M. (2010). *SamplerCompare: A framework for comparing the performance of MCMC samplers*. R package version 1.0.1.

Vrugt, J. A., Braak, C., Diks, C., Robinson, B. A., Hyman, J. M., and Higdon, D. (2009). Accelerating markov chain monte carlo simulation by differential evolution with self-adaptive randomized subspace sampling. *International Journal of Nonlinear Sciences & Numerical Simulation*, 10(3):271–288.

Wang, Q. and Kulkarni, S. R. (2009). Divergence estimation for multidimensional densities via k-nearest-neighbor distances. *IEEE Transactions on Information Theory*, 55(5):2392–2405.

Yu, H. (2012). *Rmpi: Interface (Wrapper) to MPI (Message-Passing Interface)*.

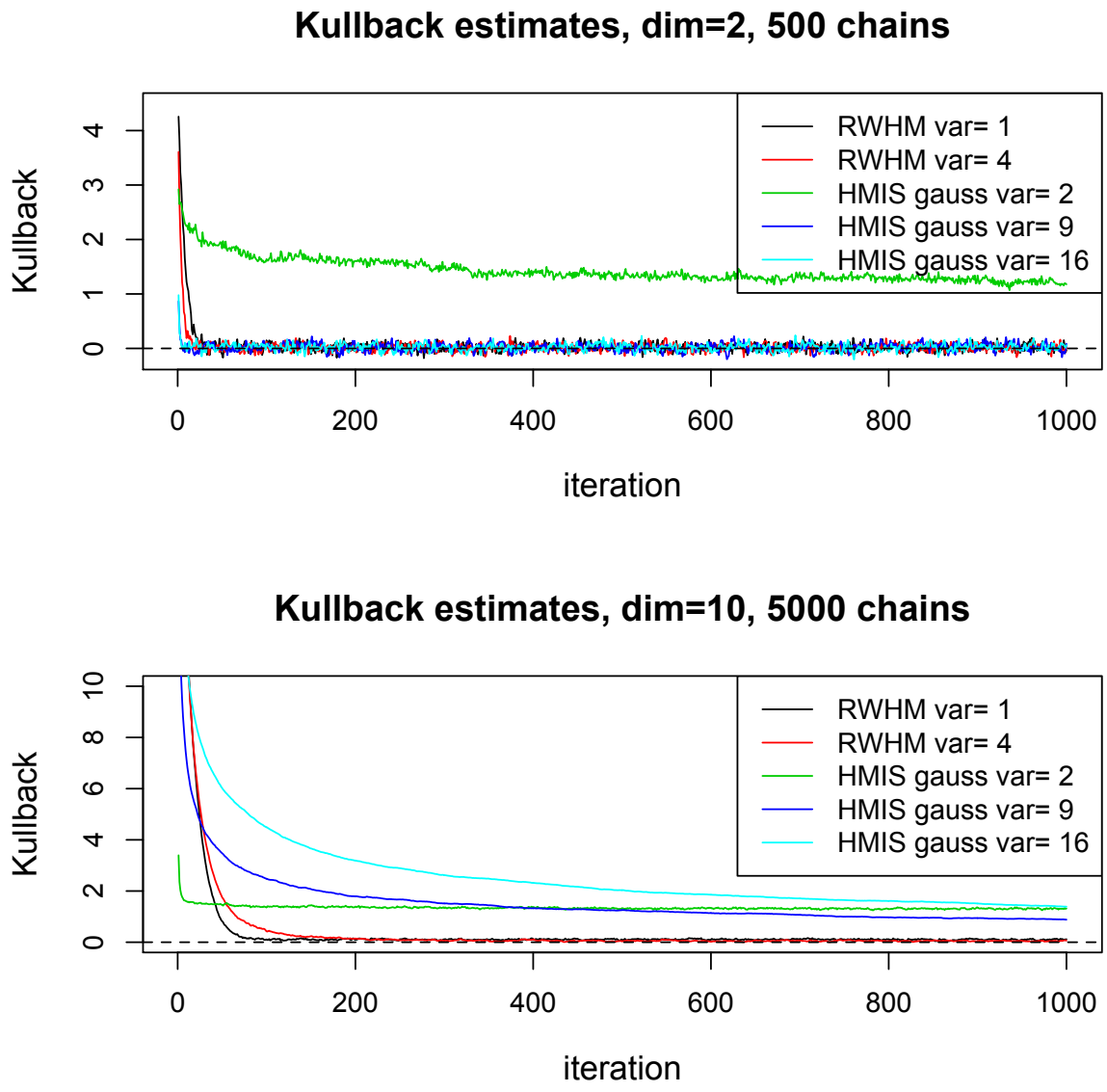

Figure 2: *Kullback estimates* $t \mapsto \hat{K}_N(p^t, f)$ *for* $n = 1000$ *iterations of the 5 MCMC strategies RW1 (black), RW4 (red), IS2 (green), IS9 (blue), IS16 (light blue). Top:* $d = 2$ *and* $N = 500$ *chains; bottom:* $d = 10$ *and* $N = 5000$ *chains.*

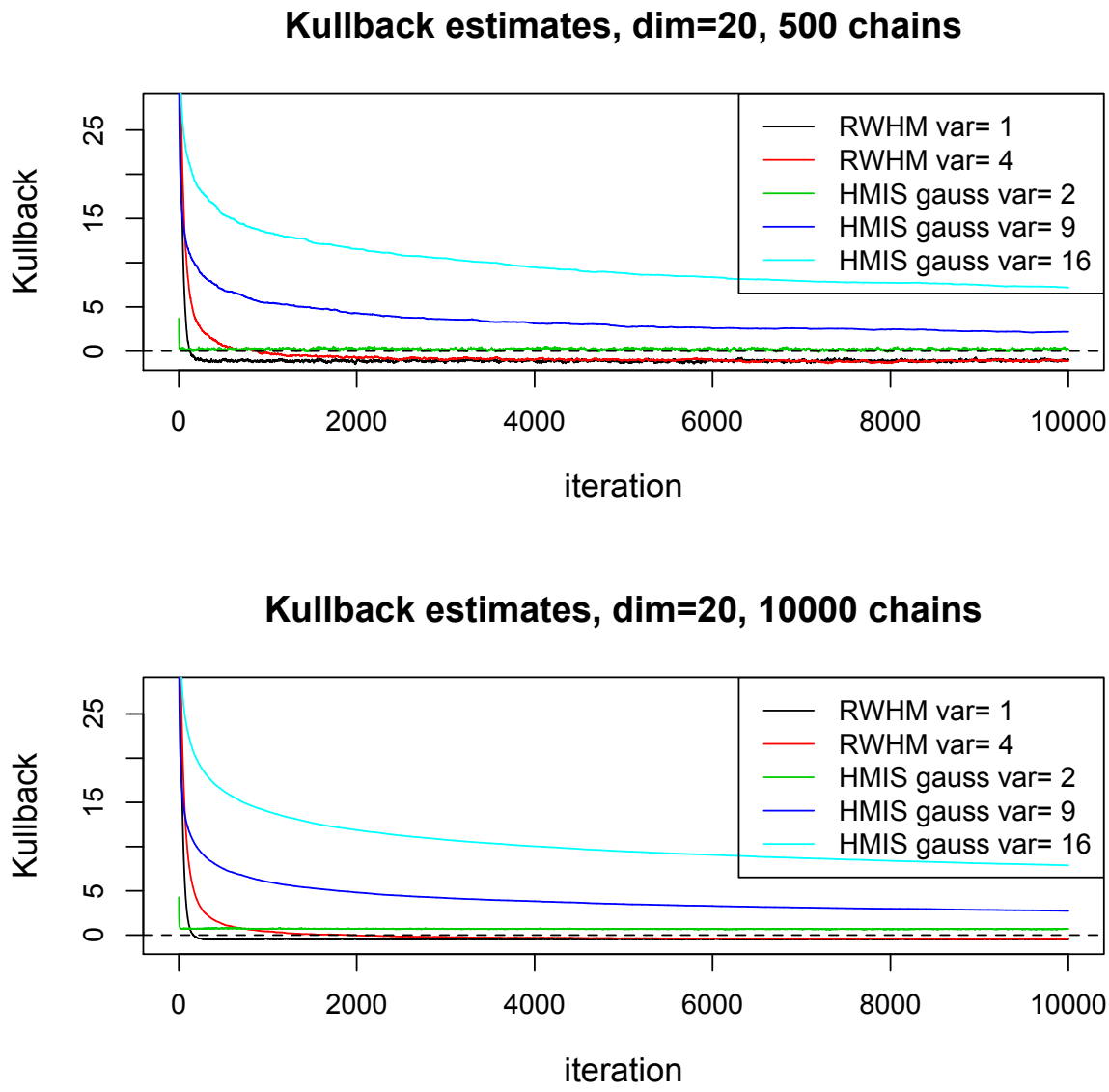

Figure 3: *Kullback estimates* $t \mapsto \hat{K}_N(p^t, f)$ *for* $d = 20$ *and* $n = 10,000$ *iterations of the 5 MCMC strategies RW1 (black), RW4 (red), IS2 (green), IS9 (blue), IS16 (light blue). Top:* $N = 500$ *chains; bottom:* $N = 10,000$ *chains.*