Entropy-based convergence analysis for (A)MCMC algorithms in high dimension

Didier Chauveau, Pierre Vandekerkhove

Entropy-based convergence analysis for (A)MCMC algorithms in high dimension

Didier Chauveau* Pierre Vandekerkhove†

June 4, 2020

Abstract

Many recent and often (Adaptive) Markov Chain Monte Carlo (A)MCMC methods are associated in practice to unknown rates of convergence. We propose a simulation-based methodology to estimate and compare MCMC's performance, using a Kullback divergence criterion requiring an estimate of the entropy of the algorithm densities at each iteration, computed from iid simulated chains. In previous works, we proved some consistency results in MCMC setup for an entropy estimate based on Monte-Carlo integration of a kernel density estimate proposed by Györfi and Van Der Meulen (1989), and we investigate an alternative Nearest Neighbor (NN) entropy estimate from Kozachenko and Leonenko (1987). This estimate has been used mostly in univariate situations until recently when entropy estimation in higher dimensions has been considered in other fields like neuroscience or system biology. Unfortunately, in higher dimensions, both estimators converge slowly with a noticeable bias. The present work goes several steps further, with bias reduction and automatic (A)MCMC convergence criterion in mind. First, we apply in our situation a recent, "crossed NN-type" nonparametric estimate of the Kullback divergence between two densities, based on iid samples from each, introduced by Wang et al. (2006, 2009). We prove the consistency of these entropy estimates under recent uniform control conditions, for the successive densities of a generic class of MCMC algorithm to which most of the methods proposed in the recent literature belong. Secondly, we propose an original solution based on a PCA for reducing relevant dimension and bias in even higher dimensions. 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.

*Institut Denis Poisson, Université d'Orléans, Université de Tours, CNRS, Route de Chartres, BP 6759, 45067 Orléans cedex 2, FRANCE. didier.chauveau@univ-orleans.fr

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

The Metropolis-Hastings (MH) algorithm (Hastings, 1970; Metropolis et al., 1953) is one of 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", 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 these last two decades, 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 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 algorithm 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 tools or numerical methods for MCMC comparisons,

showing that these questions are crucial in nowadays MCMC application and research; for instance 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 for two different situations:

(S1) For researchers to numerically better understand which methods (MCMC or AMCMC) perform best among a set of given and fully tractable circumstances.

(S2) For end users to decide, given a practical Bayesian problem, which sampler among a list of possible candidates, with one of them identified as a convergent benchmark algorithm, performs the best.

The common feature of the two above situations is that a sample directly drawn from the target density, see **(S1)**, or a very close approximation of it, see **(S2)**, is available to proceed to our study. 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 $\mathbb{R}^d$, 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 (with t) of the Kullback-Leibler divergence between p^t and f,

$$\mathcal{K}(p^t, f) := \int p^t \log\left(\frac{p^t}{f}\right) = \mathcal{H}(p^t) - \int p^t \log f. \tag{2}$$

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 satisfies a uniform minorization condition implying its geometric convergence as in Holden (1998), then $\mathcal{K}(p^t, f)$ also decreases geometrically. Our criterion requires the estimation of two terms: the entropy $\mathcal{H}(p^t)$ and the "external entropy" $\int p^t \log f = \mathbb{E}_{p^t}(\log f)$, where p^t is not available explicitly.

We start in Section 2, by a presentation of our approach which is grounded on parallel chains methods, and a review of our successive developments related to this objective. In particular, we mention the two estimates of $\mathcal{H}(p^t)$ we considered, and the reasons for which this methodology requires more work, motivating the present article. In Section 3, we present in detail the entropy and Kullback estimates we propose, historically based on Nearest Neighbor (NN) from Kozachenko and Leonenko (1987) adapted to our adaptive MCMC setup, and recently improved by NN entropy and Kullback estimates from Wang et al. (2006, 2009). In Section 4, we prove the consistency of these entropy estimates based on new conditions rigorously established by Bulinski and Dimitrov (2019), see also Berrett et al. (2019) for the k-NN entropy estimate case, for a generic class of Adaptive Metropolis-Hastings (AMH) algorithms to which most of the AMCMC strategies proposed in the recent literature belong. Section 5 illustrates the good behavior of our criterion on synthetic and actual multi-dimensional examples. These examples also allow us to show that the bias coming from the curse of dimension in the nonparametric statistical estimation of $\mathcal{H}(p^t)$ can lead to wrong decisions. In Section 6 we consequently introduce a novel solution for handling that bias problem, based on Principal Components Analysis (PCA) and projections, in such a way that our approach is still usable

even in large dimension, in practice for Bayesian models with dozens of parameters. Some technical details are finally given in the Appendix.

It is important to mention that our method, which requires intensive simulations of parallel chains, is in the scope of the current evolution of statistical computing that uses more and more parallel computing. All our algorithms are progressively implemented in the EntropyMCMC (Chauveau and Alrachid, 2019) package for the R statistical software (R Core Team, 2018) that precisely takes advantage of these recent advances, and can exploit multicore computers, networked workstations and genuine computer clusters. The first version of this package is already available online[1]

2 Problem review and objectives

Motivations for estimation of the entropy $\mathcal{H}(p)$ for a multivariate density p over $\mathbb{R}^d$ appeared recently in the literature, in other fields like molecular science (see, e.g. Singh et al., 2003) or Biology (Charzyńska and Gambin, 2015). 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 MCMC kernel. In addition, these approaches could be used to estimate $\mathcal{H}(f)$ but cannot estimate $\mathcal{H}(p^t)$ for each t with a same desired precision (sample size).

Our approach is consequently based on the simulation of N parallel (iid) copies of (eventually Adaptive) Markov chains started from a (preferably) diffuse initial distribution p^0 and using the transition kernel defined by the (A)MCMC strategy under investigation. The N chains, started from $\theta_1^0, \ldots, \theta_N^0$ 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 a (eventually non-homogeneous) Markovian move. At "time" (iteration) t, the locations of the N simulated chains $\boldsymbol{\theta}^t = (\theta_1^t, \ldots, \theta_N^t)$ form a N-sample iid$\sim p^t$.

In an experimental framework where one wants to evaluate a new (A)MCMC algorithm – situation **(S1)** – the target often corresponds to a benchmark model, for which f 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 i.e., the Strong Law of Large Numbers,

$$\hat{p}_N^t(\log f) = \frac{1}{N}\sum_{i=1}^{N} \log f(\theta_i^t) \to \int p^t \log f \quad \text{when } N \to \infty \tag{3}$$

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, say $f(\cdot) = C\phi(\cdot)$ where ϕ is the product of the prior and likelihood, the normalization constant C is not known – situation **(S2)**. In this case only $\hat{p}_N^t(\log \phi)$ is accessible. This is not really a problem since ϕ itself retains all the specificity (shape, modes, tails,...) of f, and since we are mostly interested in the stabilization in t of $\mathcal{K}(p^t, f)$,

[1] https://CRAN.R-project.org/package=EntropyMCMC

not necessarily in knowing its limiting value (which, as we will see, can also be biased in large dimensions). 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

$$\mathcal{K}(p_1^t, f) - \mathcal{K}(p_2^t, f) = \mathcal{H}(p_1^t) - \mathcal{H}(p_2^t) + \mathbb{E}_{p_2^t}[\log \phi] - \mathbb{E}_{p_1^t}[\log \phi]. \tag{4}$$

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. We start here by recalling the building blocks of our methodology, since our early results, and the problems that have risen to the challenges addressed in the present work.

Step 1 In Chauveau and Vandekerkhove (2013), we first proposed an estimate of $\mathcal{K}(p^t, f)$ based on the simulation of parallel (iid) chains started from a same initial distribution. In this work, the estimation of $\mathcal{H}(p^t)$ was built on a Kernel density estimate (KDE), and the estimation of $\int p^t \log f$ was based on Monte-Carlo integration (SLLN) of f or ϕ using (3). This KDE following Györfi and Van Der Meulen (1989) was interesting since it requires mild regularity condition for p^t. We proved the consistency of our estimate as $N \to \infty$ in some generic MCMC situations. A difficulty in this approach was the requirement of tuning parameters: as always, the kernel bandwidth for the KDE, but more importantly a trimming parameter difficult to tune appropriately. We did not investigate high dimension situations in that work.

Step 2 In Chauveau and Vandekerkhove (2014), we showed that the KDE of $\mathcal{H}(p^t)$ deteriorates as dimension increases, a phenomenon known as the curse of dimension that was actually noticed in literature from other fields requiring entropy estimation in moderate to high dimensions (see e.g., Stowell and Plumbley, 2009; Sricharan et al., 2013). This and the question of tuning the trimming parameter motivated us to investigate an alternative estimation technique based on Nearest Neighbors (NN), to estimate the first term $\mathcal{H}(p^t)$ in (2). This family of estimators initiated by Kozachenko and Leonenko (1987) reveals itself particularly attractive since it allows for a straightforward implementation whatever the dimension and without any tuning parameters. We did not investigate its theoretical properties in MCMC situation in this article. The NN estimate of $\mathcal{H}(p)$ also shows on our numerical experiments a better stability in moderate dimension (say $d \leq 10$). We also proposed a first solution to get rid of the curse of dimension. This methodology was based on the assumed availability of a "benchmark" convergent MCMC, with an approximately known convergence time t^*. Using the stabilization value of the estimate $\mathcal{K}(p^{t^*}, f)$ for this benchmark as the desired target value for any converging MCMC (namely an approximation of bias $+\log(C)$), we compared competing MCMC algorithms based on their stabilization towards this reference.

The present work goes several steps further: We propose to use in this (A)MCMC situation a recent nonparametric estimate of the Kullback divergence between any two densities,

$\mathcal{K}(g_1, g_2)$, based on crossed NN-type estimates based on iid samples from both densities. This estimate has been introduced by Wang et al. (2006, 2009). We will refer to it as a "2-samples-NN" or 2-NN estimate here, see Section 3.1, to insist on its difference with the previous estimate of $\mathcal{K}(p^t, f)$ that was using a NN estimate for $\mathcal{H}(p^t)$ (from one sample) and Monte-Carlo integration of the analytical form of f (or ϕ) for $\mathbb{E}_{p^t}(\log f)$ as in (3). This 2-NN estimate hence does not require any analytical form of the target density, but it has an obvious drawback: an iid sample from f is required and is not directly available in MCMC situation. Using this version has nevertheless several advantages and possibilities:

1. An experimental study we did conclude that, when $g_1 \equiv g_2$, the bias in the estimation of $\mathcal{K}(g_1, g_2)$ eliminates since each term has a bias (in the density estimation) of the same magnitude, even in moderate to large dimension (at least up to $d = 50$ and for Gaussian, Student, and Gaussian mixture distributions). Hence, when $p^t \approx f$, we can expect the bias in $\mathcal{K}(p^t, f)$ estimation to be approximately negligible, so that stabilization of the sequence $t \mapsto \mathcal{K}(p^t, f)$ towards zero indicates convergence; this eliminates the requirement of a reference value (unlike in Step 2 above) and provides a better readability and automatic decision rule for our criterion.

2. The estimation of the sequence $\mathcal{K}(p^t, f)$ for $t = 1, \ldots, n$ does not require $N*n$ evaluations of $\log f$ that needs computing effort in situations where the target is complex. The expression of f or ϕ is not evaluated in the estimation step.

3. Our present experiments show that, when the dimension gets large (typically $d > 50$), even the 2-NN estimate of $\mathcal{K}(p^t, f)$ for $p^t \approx f$ may be slightly biased, resulting in possible wrong decision despite its better behavior as explained in (1) above. We thus propose an alternative: monitoring the convergence of the processes obtained by projection of the original chain paths in optimal, lower dimension sub-spaces obtained by a Principal Components Analysis (PCA), where the 2-NN estimate is unbiased so that our criterion is reliable. This will be detailed in Section 6.

3 Entropy and Kullback estimation in MCMC context

As detailed in the introduction, our approach is based on the simulation of N iid copies of simulated chains, sharing the same initial distribution and (A)MCMC kernel. 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, $t = 1, \dots, n$, so that the sequence of $(\mathcal{H}(p^t))_{t=1,\dots,n}$ 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.

3.1 Estimates based on Nearest Neighbor (NN) 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 choosing an 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). Here, based on the sample $\boldsymbol{\theta}^t$ iid$\sim p^t$ in dimension d, this NN estimate is

$$\hat{\mathcal{H}}_N(p^t) = \frac{1}{N}\sum_{i=1}^{N} \log(\rho_i^d) + \log(N-1) + \log(C_1(d)) + C_E, \tag{5}$$

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

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

is the (Euclidean) distance $\rho(\cdot,\cdot)$ from the ith point to its nearest neighbor in the sample $\boldsymbol{\theta}^t$. The term involving the Euler constant comes for a correction for asymptotic unbiasedness, see Kozachenko and Leonenko (1987), Equation (10). We also provide an intuitive understanding of this estimate in Appendix B.

In our setup, the external entropy term $\int p^t \log f$ in $\mathcal{K}(p^t, f)$ can be estimated following two methods. Remember that the target f is fully known in situation **(S1)** so that a N-sample iid $\sim f$ is available, and that f is only known up to a multiplicative constant in **(S2)**, and only a N-sample iid $\sim f_\varepsilon$ is available (using a benchmark MCMC) in this latter case. Thus in **(S1)** the natural estimate is $\hat{p}_N^t(\log f)$, the Monte-Carlo (MC) integration of $\log f$, consistent by the SLLN, given by Equation (3). We denote the estimate of $\mathcal{K}(p^t, f)$ using this MC term

$$\hat{\mathcal{K}}_{N,1}(p^t, f) := \hat{\mathcal{H}}_N(p^t) - \hat{p}_N^t(\log f),$$

where the subscript "1" indicates that it is based on a single sample iid $\sim p^t$ contrarily to the two-sample based approach (7) under **(S2)**. In **(S2)**, a similar estimate can be used, up to the multiplicative constant, i.e. by using the MC integration of the available expression of $\log \phi$, where $f(\cdot) \propto \phi(\cdot)$.

However, previous studies and experiments as, e.g., Stowell and Plumbley (2009) , Sricharan et al. (2013), have shown that the available entropy estimates — including the NN estimate $\hat{\mathcal{H}}_N(p^t)$ used here — are suffering from a non neglictible bias arising when the dimension gets large (noticeable already for $d > 10$, see Chauveau and Vandekerkhove (2014) Figure 2). Since the Monte-Carlo term is not suffering from the same bias, the Kullback estimate $\hat{\mathcal{K}}_{N,1}(p, f)$ based on the NN + MC terms is biased. Chauveau and Vandekerkhove (2014) proposed a methodological approach using some prior information from a benchmark MCMC known to converge, to handle this bias.

A novelty of the present work, as stated in Section 2, consists in introducing in our (A)MCMC context another estimate of $\mathcal{K}(p^t, f)$ involving the NN machinery for both terms,

with bias reduction in mind. Indeed, the external entropy can also be estimated very similarly using a two-sample NN approach, see Wang et al. (2006) or Wang et al. (2009). For that we have to consider an iid sample $\boldsymbol{\theta}^{\varepsilon} = (\theta_1^{\varepsilon}, \ldots, \theta_N^{\varepsilon})$ drawn from f_{ε} (where $f_{\varepsilon} \equiv f$ directly in **(S1)**) and define

$$\hat{\mathcal{H}}_N(p^t, f_{\varepsilon}) = \frac{1}{N}\sum_{i=1}^{N} \log(\nu_i^d) + \log(N) + \log(C_1(d)) + C_E, \tag{6}$$

where

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

is the distance from the ith point in $\boldsymbol{\theta}^t$ to its nearest neighbor in the sample $\boldsymbol{\theta}^{\varepsilon}$. This leads to the two-sample estimate of $\mathcal{K}(p^t, f)$,

$$\hat{\mathcal{K}}_{N,2}(p^t, f) := \hat{\mathcal{H}}_N(p^t) - \hat{\mathcal{H}}_N(p^t, f_{\varepsilon}), \tag{7}$$

where now the subscript "2" indicates that this estimator uses the two N-samples $\boldsymbol{\theta}^t$ iid$\sim p^t$ and $\boldsymbol{\theta}^{\varepsilon}$ iid$\sim f$ in **(S1)** or iid $\sim f^{\varepsilon}$ in **(S2)**. This estimate will be called "2-NN" in the experiment Section and figures.

4 Peak and tail type conditions

We propose in this section to briefly introduce the technical assumptions established recently by Bulinski and Dimitrov (2019) in order to rigorously prove the L^2-consistency of the entropy estimator (5) and check that they are satisfied for the successive densities p^t of 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). The asymptotic analysis of the external entropy estimator (6) being very similar to the entropy estimator (5), we propose, for simplicity matters, to present and derive our theory only on this last case (the extension would require extra tedious work not of great importance for application).

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. For easier notations, we denote in this section θ^t by x^t, to omit the connection with Bayesian MCMC setup, and just focus on the stochastic process involved. In the case of a generic Adaptive MH processes $(X^t)_{t\geq 0}$ valued in $\mathbb{R}^d$, 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. Set $G(t) = t \log t \mathbb{I}_{t \geq 1}$ and for any pdf h defined on $\mathbb{R}^d$, $x \in \mathbb{R}^d$, $r > 0$ and $R > 0$ introduce the functions:

$$I_h(x,r) = \frac{\int_{B(x,r)} h(y)dy}{r^d V_d}, \quad M_h(x,R) = \sup_{r \in (0,R]} I_h(x,r), \quad m_h(x,R) = \inf_{r \in (0,R]} I_h(x,r),$$

where $B(x,r)$ denotes the ball in $\mathbb{R}^d$ with radius r centered at point $x \in \mathbb{R}^d$, see also Section B for further notations.

Assumptions. For positive ε_i, $i = 0,1,2$ and R_j, $j = 1,2$, we have :

(A1) $K_h(\varepsilon_0) = \int \left(\int G\left(|\log \rho(x,y)| \, h(y)dy \right) \right)^{1+\varepsilon_0} h(x)dx < +\infty.$

(A1') $K_{h,2}(\varepsilon_0) = \int \left(\int G\left(\log^2 \rho(x,y) h(y)dy \right) \right)^{1+\varepsilon_0} h(x)dx < +\infty.$

(A2) $Q_h(\varepsilon_1, R_1) = \int M_h^{\varepsilon_1}(x, R_1) h(x)dx < +\infty.$

(A3) $T_h(\varepsilon_2, R_2) = \int m_h^{-\varepsilon_2}(x, R_2) h(x)dx < +\infty.$

If conditions **(A1–3)** are satisfied, Theorem 1 in Bulinski and Dimitrov (2019) establishes that the estimate (5) is asymptotically unbiased. If in addition **(A1')** is also satisfied, Theorem 2 in Bulinski and Dimitrov (2019) establishes the L^2 consistency of the estimate (5).

Proposition 1. *Suppose that there exist nonnegative functions* (φ_1, φ_2) *both defined on* $\mathbb{R}^d$, *a constant* $a \in (0,1)$ *and positive constants* ε_i, $i = 0,1,2$ *such that:*

(C1) $C_1 = \int \varphi_1(x)dx < \infty$, $C_2 = \int \frac{\varphi_1^2(x)}{\varphi_2(x)} dx < \infty$, *and* $C_3 = \int \varphi_2(x)dx < \infty.$

(C2) $\varphi_1 \leq p^0 \leq \varphi_2$, *and* $\varphi_1 \leq f$.

(C3) $af \leq q_\vartheta \leq \varphi_2$ *for all* $\vartheta \in \Theta$.

(C4) $K_{\varphi_2}(\varepsilon_0) < +\infty.$

(C4') $K_{\varphi_2,2}(\varepsilon_0) < +\infty.$

(C5) $\int M_{\varphi_2}^{\varepsilon_1}(x, R_1) \varphi_2(x)dx < +\infty.$

(C6) $\int m_{\varphi_1^2/\varphi_2}^{-\varepsilon_2}(x, R_2) \varphi_2(x)dx < +\infty.$

Under **(C1-3)**, *the successive densities of the Adaptive MH algorithm described above then satisfy respectively* **(A1-3)** *and* **(A1')** *if* **(C4-6)** *and* **(C4')** *respectively hold.*

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 (13) and (14) using conditions **(C1-3)** of Proposition 1:

$$\begin{aligned} A^t(x) &:= a^{2t} C_1 C_2^{t-1} \frac{\varphi_1^2(x)}{\varphi_2(x)} \\ B^t(x) &:= 2(C_3+1)^{t-1} \varphi_2(x) \\ A^t(x) &\leq p^t(x) \leq B^t(x), \quad t \geq 1. \end{aligned} \tag{8}$$

For **(A1)** we can notice, since $G(\cdot) \geq 0$, that from (8) we have:

$$\begin{aligned} K_{p^t}(\varepsilon_0) &= \int \left(\int G\left(|\log \rho(x,y)| \right) p^t(y) dy \right)^{1+\varepsilon_0} p^t(x) dx \\ &\leq \int \left(\int G\left(|\log \rho(x,y)| \right) B^t(y) dy \right)^{1+\varepsilon_0} B^t(x) dx \\ &\leq \left(2(C_3+1)^{t-1} \right)^{2+\varepsilon_0} \int \left(\int G\left(|\log \rho(x,y)| \, \varphi_2(y) dy \right) \right)^{1+\varepsilon_0} \varphi_2(x) dx. \end{aligned}$$

For **(A2)**, using (8) and the fact that

$$M_{p^t}^{\varepsilon_1}(x, R_1) = \sup_{r \in (0,R]} \frac{\int_{B(x,r)} p^t(y) dy}{r^d V^d} \leq 2(C_3+1)^{t-1} \sup_{r \in (0,R]} \frac{\int_{B(x,r)} \varphi_2(y) dy}{r^d V^d},$$

we have:

$$Q_{p^t}(\varepsilon_1, R_1) \leq \left(2(C_3+1)^{t-1} \right)^2 \int M_{\varphi_2}^{\varepsilon_1}(x, R_1) \varphi_2(x) dx < +\infty.$$

For **(A3)**, using again (8) and the fact that

$$m_{p^t}(x, R) = \inf_{r \in (0,R]} I_{p^t}(x, r) \geq \inf_{r \in (0,R]} I_{A^t}(x, r) = a^{2t} C_1 C_2^{t-1} \inf_{r \in (0,R]} I_{\varphi_1^2/\varphi_2}(x, r)$$

we have

$$T_{p^t}(\varepsilon_2, R_2) = \int m_{p^t}^{-\varepsilon_2}(x, R_2) p^t(x) dx \leq \frac{2(C_3+1)^{t-1}}{\left(a^{2t} C_1 C_2^{t-1} \right)^{\varepsilon_2}} \int m_{\varphi_1^2/\varphi_2}^{-\varepsilon_2}(x, R_2) \varphi_2(x) dx < +\infty,$$

which conclude the proof. □

5 Numerical Experiments and simulations

Before experimenting our approach on actual or synthetic MCMC and Bayesian models, we have checked that $\hat{\mathcal{K}}_{N,2}(f, f) \approx 0$, for some known f for various dimensions $2 \leq d \leq 50$ and several parametric families (multivariate Gaussian, Student with heavy tails, and mixtures of these). We have also checked that $\hat{\mathcal{K}}_{N,2}(f, f_\varepsilon) \approx 0$, where the sample iid$\sim f_\varepsilon$ comes from a convergent (bench) MCMC ran up to a large enough number of iterations, with **(S2)** in mind. These preliminary experiments (not detailed in this article for brevity) give a numerical evidence that the 2-NN estimate is approximately unbiased, at least when the two samples come from approximately the same distribution, in comparison with the one-sample estimate. This will be illustrated in the first two examples below.

5.1 MCMC vs. AMCMC in situation (S1)

We illustrate this situation with a well-known synthetic model called the *banana-shaped* example, a benchmark already used in several MCMC articles. We compare here standard Random-Walk Hastings-Metropolis (RWHM) algorithms available in the EntropyMCMC package (Chauveau and Alrachid, 2019), vs. an Adaptive Metropolis (AM) sampler from Haario et al. (2001), documented in Roberts and Rosenthal (2009) and for which a C code simulating a single chain, is available online[2]. Our motivations for using this example are twofold:

1. illustrate the employment of the R package EntropyMCMC (Chauveau and Alrachid, 2019) [3] in an actual context where an external code for a AMCMC algorithm is available online, and needs only to be run repeatedly to obtain N iid copies of simulated Markov Chains, to which the Kullback criterion can be applied in a black-box manner after importing into R the array (n, d, N) of simulated chains.

2. illustrate the real improvement of our new two-samples estimate $\hat{\mathcal{K}}_{N,2}(p^t, f)$ in term of bias elimination, in comparison with the MC + NN estimate $\hat{\mathcal{K}}_{N,1}(p^t, f)$. This bias elimination gives, at least in situation **(S1)**, a practical tool for both convergence assessment and MCMC's comparisons.

Our experiment for this model has been set to dimension $d = 20$, and simulation of $N = 600$ chains for $n = 30,000$ iterations each. One important point is the choice of the initial distribution p^0 used to draw the starting position of the N chains. The code associated to Roberts and Rosenthal (2009) sets $\theta^0 = 100\mathbb{I}_d \in \mathbb{R}^d$, i.e. $p^0 \equiv \delta_{100}$. This can be viewed as a penalty for two reasons. First, the domain of interest of the banana-shaped pdf is similar to that of the product of d normals $\mathcal{N}(0, 1)$, except for the first two coordinates that are (non-linearly) transformed to get the banana shape. Hence a chain started from $\theta^0 = (100, \ldots, 100)$ has to accept big jumps to "escape" from θ^0 and reach the area of interest. A standard RWHM is penalized in the sense that if its variance is large enough to escape from θ^0, then it will be too large then to explore the domain (smaller in comparison) efficiently. Conversely, if its variance is calibrated small enough to explore the domain, it will not allow the chain to escape from θ^0. These situations can be highlighted by running a couple of RWHM algorithms and look at their convergence and empirical rates of acceptation in the EntropyMCMC package. Here the adaptation helps the AM by updating the variance matrix of the random walk, but to compare with RWHM's we choose instead $p^0 \equiv \delta_0$, so that all chains are started from the center of the domain. Secondly, even if a Dirac distribution is natural in a single chain experiment, a diffuse initial distribution would have been preferable in our parallel chain setup. However, we kept the Dirac initial distribution for simplicity, allowing us to use directly the public code for the AM.

We compare three algorithms: two RWHM with variance parameters 1 (RW1) and 0.02 (RW2), and the AM from Haario et al. (2001) discussed in Roberts and Rosenthal (2009) using the available code and tuning parameters. For these, we compared our two Kullback estimates $\hat{\mathcal{K}}_{N,k}(p^t, f)$ for $k = 1, 2$. Remember that here **(S1)** the asymptotic N-sample involved in $\hat{\mathcal{K}}_{N,2}(p^t, f)$ is drawn from the exact target pdf, i.e. $\boldsymbol{\theta}^{\varepsilon}$ iid $\sim f$.

Figure 1 shows the plots of the criterion for both estimates. We can see that the two RWHM's stabilize at the same level, corresponding to the bias of the estimate $\hat{\mathcal{K}}_N(p^t)$ indicated on Fig. 1, top. The RW with variance parameter set to 1 (RW1) can here play the role of

[2] from J. Rosenthal, http://probability.ca/jeff/comp/

[3] https://CRAN.R-project.org/package=EntropyMCMC

the benchmark, but we have to assume (e.g. by running more iterations, or using external information like here because we actually know the target) that this is acceptable. Then the bias can be estimated from a batch of last iterations. This RW1 converges (i.e. stabilizes at the bias level) after about $15,000$ iterations, whereas RW2 with too small variance converges more slowly, after about $25,000$ iterations. The important fact is that we can deduce the same convergence and comparison properties much more directly buy looking at the criterion based on the 2-NN estimate $\hat{\mathcal{K}}_{N,2}(p^t, f)$, since it is unbiased so that a convergent algorithm should stabilizes around 0, without the requirement of a benchmark MCMC. The AM does not even converges during these $n = 30,000$ iterations, for both estimates, i.e. the biased or the unbiased. This is not really surprising since the advantage of this AMCMC is its ability to found the area of interest where the mass of the target is, when started from far away. As mentioned in Roberts and Rosenthal (2009) in their experiments, *it takes many iterations for the algorithm to learn this information*; they announce $n = 400,000$ for $d = 100$.

5.1.1 An automated convergence criterion

The fact that the new version of the convergence criterion (the 2-NN estimate) stabilizes around 0 instead of a strongly biased value depending on a benchmark MCMC and the target particularities, suggests to monitor the convergence in a automatic manner based on the sequence $t \mapsto \hat{\mathcal{K}}_{N,2}(p^t, f)$, in a way to produce a convergence time comparing MCMC's. After some investigations, we choose a numerical procedure based on moving averages and numerical derivatives, both required to be sufficiently smaller than some $\epsilon > 0$ simultaneously. An example of the result of this procedure is displayed in Figure 2. The ordering of the three MCMC's is as we suggest by looking at Figure 1, i.e. RW1 converges faster than RW2, and AM does not satisfy the convergence criterion in these $30,000$ iterations (despite a short "visit" near 0 for $4000 \leq t \leq 5000$). The tuning parameters of the procedure (size of the moving window, lag for differences, ϵ), together with the resulting convergence times, are given in the plots.

5.2 AMCMC in situation (S2): Real data and Bayesian model

In this section we propose to compare various (adaptive)-MCMC strategies on the James-Stein Bayesian model for baseball data studied in Rosenthal (1996). In that model, a sample $(Y_1, \ldots, Y_K)$ is observed where the $Y_i|\theta_i \sim \mathcal{N}(\theta_i, V)$, $i = 1, \ldots, K$, and are conditionally independent. The θ_i's are unknown parameters to be estimated and the variance V is supposed to be known. In addition $\theta_i|\mu, A \sim \mathcal{N}(\mu, A)$, $i = 1, \ldots, K$, and are conditionally independent. The James-Stein estimator for θ_i is obtained, see Efron and Morris (1975), as the posterior mean $\mathbb{E}(\theta_i|Y_i)$, where μ is a prior guess on θ's and where $(1 + A^2)^{-1}$ is replaced by its unbiased estimator $(K-2)/\sum_{i=1}^{K}(Y_i - \mu)^2$. To integrate μ and A as further parameters to be estimated, Rosenthal (1996) introduce prior distributions $\mu \sim \mathcal{N}(\mu_0, \sigma_0^2)$ and $A \sim IG(a, b)$, the inverse Gamma distribution with pdf proportional to $\exp(-b/x)x^{-(a+1)}$. This leads for $(A, \mu, \theta_1, \ldots, \theta_K)$ to a posterior distribution on the $(K + 2)$-dimensional state space $\mathcal{X} =$

$[0, +\infty) \times \mathbb{R}^{K+1}$ with pdf

$$\begin{aligned} f(A, \mu, \theta_1, \ldots, \theta_k) &= \mathcal{N}(\mu_0, \sigma_0^2; \mu) IG(a_1, b_1; A) \times \prod_{i=1}^{K} [\mathcal{N}(\mu, A; \theta_i) \mathcal{N}(\theta_i, V; Y_i)] \\ &\propto \exp(-(\mu - \mu_0)^2 / 2\sigma_0^2) \exp(-b_1/A) / A^{a_1+1}) \\ &\quad \times \prod_{i=1}^{K} [A^{-1/2} \exp(-(\theta_i - \mu)^2 / 2A) V^{-1/2} \exp(-(Y_i - \theta_i)^2 / 2V)]. \end{aligned} \tag{9}$$

This Bayesian posterior distribution is applied on a real baseball dataset $(y_1, \ldots, y_{18})$ with $K = 18$ observations, see Table 1 in Efron and Morris (1975). The state space is $\mathcal{X} \subseteq \mathbb{R}^{20}$ and they use prior parameters $\mu_0 = 0$, $\sigma_0^2 = 1$, $a_1 = -1$ and $b_1 = 2$. V is replaced by its fixed empirical Bayes estimate.

In Roberts and Rosenthal (2009), the authors run a Regional Adaptive Metropolis Algorithm (RAMA) on this example. The RAMA methods consists in partitioning the state space $\mathcal{X}$ into a finite number of disjoint regions, *i.e.* $\mathcal{X} = \mathcal{X}_1 \dot{\cup} \ldots \dot{\cup} \mathcal{X}_m$. The algorithm then runs a Metropolis algorithm with proposal $Q(x, \cdot) = \mathcal{N}(x, \exp(2r_i))$, whenever $x \in \mathcal{X}_i$, $i = 1, \ldots, m$. Now if $x \in \mathcal{X}_i$ and $y \in \mathcal{X}_j$, then $\sigma_x^2 = \exp(2r_i)$ and $\sigma_y = \exp(2r_j)$ the Metropolis acceptance ratio for the move $x \to y$ is:

$$\alpha(x, y) = \min \left[1, \frac{\pi(y)}{\pi(x)} \exp \left(d(r_i - r_j) - \frac{1}{2}(x - y)^2 [\exp(-2r_j) - exp(-2r_i)] \right) \right].$$

In Roberts and Rosenthal (2009) the adaptation is calibrated in order to get an acceptance probability close to 0.234 in each region. That acceptance rate was proved to be optimal in certain high-dimensional settings, see references therein.

Usage of the entropy criterion with the RAMA online code As in Section 5.1, we illustrate with this actual Bayesian model the relevance of our approach for evaluating and tuning new AMCMC strategies, using only (repeatedly) the C code available online for this RAMA algorithm[4]. In this code, the algorithm is designed for two regions, defined by a (so-called) cutoff parameter c, which consequently acts as a tuning parameter for the RAMA adaptive strategy. Indeed, several settings are proposed in the code, from $c = 0.07$ up to $c = 1000$, but Roberts and Rosenthal (2009) claim themselves in Remark 3 that *user specify the regions by hand*. We thus illustrate the usage of our methodology for selecting the cutoff parameter. Specifically we:

- simulate $N = 500$ iid copies of RAMA chains for $n = 10^5$ iterations, and for three cutoff values proposed in this code $(0.07, 1.6, 3.08)$;
- define in the **EntropyMCMC** package the target pdf f proportional to the posterior, as given by Equation(9);
- generate an "asymptotic sample" iid$\sim f_\varepsilon$, as required in situation **(S2)**: in the present case we ran N iid copies of one RAMA algorithm and retain the last "slice" at time $n = 10^5$ as the N-sample approximately f-distributed;
- for each cutoff value, compute the sequences of estimates $t \mapsto \hat{\mathcal{K}}_{N,1}(p^t, f)$ (NN + MC terms), and $t \mapsto \hat{\mathcal{K}}_{N,2}(p^t, f_\varepsilon)$ (2-samples NN estimates).

[4] from J. Rosenthal, `http://probability.ca/jeff/comp/`

- compare the behavior of the three RAMA w.r.t. convergence.

Results are displayed in Figure 3. Surprisingly, there are significative differences (up to $20,000$ iterations) in convergence times due to just these apparently slightly different cutoff values. First, like in **(S1)**, results illustrate the fact that (as expected and shown before) the NN+MC estimate is biased but the three algorithms converge to the same stabilization value that depends on the unknown normalization constant for f, plus the unknown bias due to dimension. The biased estimate thus give insurance that, after $n = 10^5$ iteration, the asymptotic sample is approximately f-distributed, validating the 2-NN estimation.

Secondly, both estimate types (NN+MC and 2-NN) deliver the same ranking between the three competing AMCMC. Fortunately, as in **(S1)**, the 2-NN estimate is approximately unbiased, so that the target stabilization value of 0 can be used to produce a convergence criterion in an automatic manner, as already illustrated in Figure 2.

6 PCA for unbiased Kullback estimation in large dimension

The automated convergence criterion we used in Section 5.1 is based on the fact that, for small to moderate dimensions like, say, $d \leq 20$, the 2-NN estimate $\hat{\mathcal{K}}_{N,2}(p^t, f)$ is "almost unbiased", so that stabilization around the target value of zero delivers a right answer. However, we noticed in further experiments that increasing the dimension up to larger values, this estimate itself reveals some negative bias. Again, this bias is neglictible in most cases, in comparison with the NN+MC estimate, but it may impact a numerical (automatic) detection of what "stabilization near zero" means. Hence we developed a complementary approach to reduce the dimension of the stochastic process under study.

Recently various authors considered that the convergence rate of MCMCs in high dimension, which basically degenerates as the dimension grows to infinity, could be considered/scaled with more or enough accuracy by taking into account the convergence over domains of significant interest with respect to the target density, basically with a significant amount of information related to the landscape related to f, (see, e.g. Atchadé, 2019; Yang and Rosenthal, 2019; Maire and Vandekerkhove, 2018).

We propose in this Section to develop these intuitions by numerically showing that the rates of convergence of MCMC algorithms in high dimension could be approximated with enough precision and virtually no bias if they are looked on a lower dimension linear subspace provided by a Principal Component Analysis (PCA) method (main percentage of inertia) instead of the whole space. The main advantage then is that our criterion based on the stabilization of $\mathcal{K}(p^t, f)$ around zero can be derived in an automatic manner, even if the dimension d of the state space is large. Let us denote by

$$\boldsymbol{\theta}^{\varepsilon} = (\boldsymbol{\theta}^{\varepsilon}(1), \ldots, \boldsymbol{\theta}^{\varepsilon}(d)) := \begin{pmatrix} \theta_1^{\varepsilon}(1) & \ldots & \theta_1^{\varepsilon}(d) \\ \vdots & & \vdots \\ \theta_N^{\varepsilon}(1) & \ldots & \theta_N^{\varepsilon}(d) \end{pmatrix}, \tag{10}$$

the indifferently **(S1)** or **(S2)** $N \times d$ benchmark data matrix, corresponding to a N-sample iid $\sim f$ or f^{ε}. In this notation $\theta(1), \ldots, \theta(d)$ are the d coordinates of any $\theta \in \mathbb{R}^d$, and the sample mean of each column is supposed to be shifted to zero. The PCA method is as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by some projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate,

and so on. The transformation is defined by a set of d-dimensional vectors of weights or loadings $(u_1^\varepsilon, \ldots, u_d^\varepsilon)$ corresponding to the eigenvectors of the covariance or correlation matrix based on the data matrix $\boldsymbol{\theta}^\varepsilon$ (depending on the metric used), and associated to the sorted eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_d$. The projection of the i-th benchmark individual on the q-th principal axis is obtained by

$$\psi_i(q) =< \theta_i^\varepsilon, u_q^\varepsilon >= \sum_{j=1}^{d} \theta_i^\varepsilon(j) u_q^\varepsilon(j), \quad q = 1, \ldots, d, \tag{11}$$

which defines the (simulated) data according to this new coordinate system: for all $1 \leq q \leq d$, the N-coordinates vector $\psi(q) = (\psi_1(q), \ldots, \psi_N(q))^T$ denotes the so-called q-th principal component of the PCA. The next step consists then in determining a reduced number $d' < d$ of principal axes such that the percentage of variability explained by this subset, that is $\sum_{j=1}^{d'} \lambda_j / \sum_{j=1}^{d} \lambda_j$, is considered as acceptable.

Turning back to the statistical viewpoint, each "individual" $\psi_i = (\psi_i(1), \ldots, \psi_i(d))$ is distributed as the image of f^ε by the $\boldsymbol{\theta}^\varepsilon$-PCA transformation (enhancing the spreading on subspaces that matter in the multidimensional composition of the targeted density f^ε). Unfortunately, the images of the N simulated realizations $\boldsymbol{\theta}^\varepsilon$ from the benchmark process are not iid anymore after this transformation, since the loadings depend on this whole dataset itself. This is true also for their d' first coordinates so that the $N \times d'$ dataset $\overline{\psi}$ cannot be used directly for statistical purposes, namely NN entropy estimation. One way of correcting this is to actually simulate the benchmark data matrix for $2N$ realization (at the expand of some additional computing time in situation **(S2)**), and:

1. retain a first N-sample for building the $\boldsymbol{\theta}^\varepsilon$-PCA loadings $(u_1^\varepsilon, \ldots, u_d^\varepsilon)$. In common PCA analysis, the N (simulated) data from the benchmark process that have been used to compute the PCA axis are called the "active" dataset,

2. use the second N-sample, say $\boldsymbol{\theta}_S^\varepsilon$ for the needed statistical estimates based on projection of these simulated data as in (11). Again in PCA analysis, the individuals that do not participate to the principal axes determination, but are simply projected after the PCA using the loadings, are called the "supplementary" dataset, hence the notation.

It is clear then that the d' first principal components provide a N-sample in this reduced dimension, iid from the d'-dimensional marginal of the image of f^ε by the PCA. The transformed simulated realizations is a $N \times d'$ matrix, the rows of which form an iid sample $(\overline{\psi}_1, \ldots, \overline{\psi}_N)$.

Once this preliminary $\boldsymbol{\theta}^\varepsilon$-PCA analysis is done, we propose to compute for the MCMC process of interest $(\theta^t)_{t\geq 0}$ the projection of its N realizations on the first d' principal axis of the $\boldsymbol{\theta}^\varepsilon$-PCA, for every time t. More precisely this step leads to consider for each t the $N \times d$ matrix Ξ^t of the N simulations in $\mathbb{R}^d$, with rows $\xi_1^t, \ldots, \xi_N^t$, and to compute the d' first projections with the analog of Equation (11). We obtain d' vectors of N-coordinates, each one being the projection using the $\boldsymbol{\theta}^\varepsilon$-PCA, of the i-th simulated chain from the MCMC process of interest at time t,

$$\overline{\xi}_q^t = \Xi^t u_q^\varepsilon \quad q = 1, \ldots, d'.$$

Note that the iid property of the N simulated and projected individuals $\overline{\xi}_1^t, \ldots, \overline{\xi}_N^t$ is preserved for the same reason given above.

Finally, let us denote by h^t and h^ε the d'-dimensional pdf of the iid samples $(\overline{\xi}_1^t, \ldots, \overline{\xi}_N^t)$ and $(\overline{\psi}_1, \ldots, \overline{\psi}_N)$ respectively. These are image distributions, by the PCA transformation,

of p^t and f^ε respectively. The bottom line of our approach lies in the study of the Kullback divergence $t \to \mathcal{K}(h^t, h^\varepsilon)$ when t increases, which can be estimated using the two-sample estimate $\hat{\mathcal{K}}_{N,2}(h^t, h^\varepsilon)$ as in Equation (7). As said before, this is the advantage of this estimate which does not require any analytical expression of these two densities, that are not available here.

6.1 An example for PCA-based entropy estimation

To illustrate the approach detailed above, we have built a synthetic model (situation **(S1)**), designed so that the PCA is efficient, i.e. most of the inertia of the full space (simulated) data $\boldsymbol{\theta}^\varepsilon$ is kept in the three first principal axis. The target pdf corresponds to a multivariate Gaussian in dimension $d = 50$ separated in 3 independent blocs with high within-block correlation, as detailed in Table 1.

Table 1: Block structure for a synthetic $d = 50$ model resulting in efficient PCA.

	dimension	mean	variance	within-block correlation
block 1	30	0	100	0.95
block 2	15	1	4	0.90
block 3	5	2	1	0.80

The idea is that each block is essentially summarized by one principal component (PC), these PC's being ordered by decreasing variance corresponding to the decreasing block dimensions and magnitude of within-block correlation. The $d - 3$ remaining PC's correspond to coordinates remaining from each block (merely like "noise"). Note that block means and variances have no impact on the PCA, but are chosen to impact the convergence property of our candidate MCMC, a RWHM with variance 1 started from $p^0 \equiv \mathcal{N}(0, 1)^{\otimes d}$. To investigate the effect of the number of chains over the remaining bias, we set the experiment size to $N = 500, 1000$ and 2000 parallel chains, ran for $n = 100,000$ iterations.

Figure 4 summarizes typical results we obtain. The top-left panel shows the strong bias of the NN+MC estimate $\hat{\mathcal{K}}_{N,1}(p^t, f)$, and also the slow impact of N in decreasing this bias. The three other plots are using the same y scale. The top-right panel shows the obvious improvement brought by the 2-NN estimate $\hat{\mathcal{K}}_{N,2}(p^t, f)$ for this $d = 50$ example: the Kullback divergence estimates are almost unbiaised, but a small negative bias is nevertheless noticeable for $N = 500$ iid chains, and also slightly for $N = 1000$ chains. In this case, an automatic detection of stabilization near zero based on the $N = 500$ chains run could lead to a wrong decision. The bottom panels shows the 2-NN estimates $\hat{\mathcal{K}}_{N,2}(h^t, h^\varepsilon)$ as detailed in Section 6, computed after projections of the chain's paths on the optimal subspaces of dimension $d' = 2$ (left) and $d' = 8$ (right). Thanks to the efficient PCA for this model, the $d' = 2$ subspace already retains 84% of the total inertia. The important conclusion is that one can see that the estimates are unbiased, even for $N = 500$ chains (the number of chains have an effect on the variance, as always), but also that they deliver the *same* conclusion concerning the convergence time, since the decays and stabilization around zero are similar.

7 Conclusion

In this paper, we have proposed a methodological approach to evaluate (A)MCMC efficiency and control of convergence on the basis of simulation of parallel chains only. No theoretical — and often unavailable — study of the MCMC kernels regarding their convergence rate is needed. We started with a review of our preliminar developments, the first estimates we considered, and the difficulty these first approaches were suffering due to a noticeable bias slowly decreasing in high dimensions. We have then defined for this MCMC and parallel chains framework a way to use a more recent estimate for the Kullback divergence between any two densities, based on crossed NN-type estimates and iid samples from both densities. Our numerical experiments show an efficient bias reduction in moderate dimensions, permitting and automatic (A)MCMC convergence diagnostic based on a practical, easy-to-understand graphical criterion. We also proved here the consistency of these entropy estimates under recent uniform control conditions, for the successive densities of a generic class of MCMC algorithm to which most of the methods proposed in the recent literature belong. Finally, for larger dimensions problems, we also proposed an original solution based on PCA projections of the simulated chains, for monitoring MCMC convergence in lower dimension where the automatic criterion above is usable and reliable.

Since our methods require intensive simulations that may be computationally demanding, all our algorithms are progressively implemented in the R package EntropyMCMC (Chauveau and Alrachid, 2019) for the R statistical software (R Core Team, 2018). The first version of this package is already available on the *CRAN* mirrors[5], and takes advantage of recent advances in high performance (parallel) computing. Most of the examples shown in this paper have been ran with the development version of it, that will be made available in future updates.

A further interesting topic could be the generalization and adaptation of the Leonenko and Kosachenko entropy/Kullback estimator to the online mixing and stationary case. This step could allow to detect, with a much less computing effort, some asymptotic bias between a known to converge benchmark algorithm (in the stationary regime) and an (Adaptive)-algorithm to be tested.

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 key inequality (8) in the proof of 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 **(C1-3)** 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,

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

[5] https://CRAN.R-project.org/package=EntropyMCMC

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

$$p^2(y) \geq q(y)\int p^1(x)\alpha(x,y)dx \geq a^4C_1\left[\int \frac{\varphi_1^2(x)}{\varphi_2(x)}dx\right]\frac{\varphi_1^2(y)}{\varphi_2(y)}. \tag{12}$$

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 4. 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* **(C1-3)** *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 $x_0^t \in (\mathbb{R}^d)^{t+1}$. Then, for all $y \in \mathbb{R}^d$,*

$$a^{2t}C_1C_2^{t-1}\frac{\varphi_1^2(y)}{\varphi_2(y)} \leq p^t(y) \tag{13}$$

and

$$p^t(y) \leq 2(C_3+1)^{t-1}\varphi_2(y), \tag{14}$$

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 $x_0^{t-1} = (x^0, \ldots, x^{t-1})$:

$$\begin{aligned} P_{x_0^{t-1}}(x^{t-1}, dy) &= q_{x_0^{t-1}}(y)\alpha_{x_0^{t-1}}(x^{t-1}, y)dy \\ &+ \int q_{x_0^{t-1}}(z)\left[1 - \alpha_{x_0^{t-1}}(x^{t-1}, z)\right] dz\, \delta_{x^{t-1}}(y)\, dy \end{aligned} \tag{15}$$

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 (13). 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:

$$\begin{aligned} p^t(y)\, dy &= \int p^0(x^0)\, dx^0 P_{x^0}(x^0, dx^1)P_{x_0^1}(x^1, dx^2)\ldots P_{x_0^{t-1}}(x^{t-1}, dy) \\ &\geq a^{2t}\int \varphi_1(x^0)\frac{\varphi_1^2(x^1)}{\varphi_2^2(x^1)}\cdots\frac{\varphi_1^2(x^{t-1})}{\varphi_2(x^{t-1})}dx_0^{t-1}\frac{\varphi_1^2(y)}{\varphi_2(y)}\, dy \\ &= a^{2t}C_1C_2^{t-1}\frac{\varphi_1^2(y)}{\varphi_2(y)}\, dy. \end{aligned}$$

To obtain the majorization (14) of the densities, we notice from (15) 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

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

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. □

B Entropy formula intuition

For a given pdf f in $\mathbb{R}^d$ the associated differential entropy is $H(f) = -\int_{\mathbb{R}^d} f(x)\log f(x)dx$. To estimate this entropy from a sample $(X_1,\ldots,X_N)$ drawn from f, Kozachenko and Leonenko (1987) introduced the following estimator:

$$H_N = \frac{1}{N}\sum_{i=1}^{N}\xi_i(N), \quad \text{where} \quad \xi_i(N) = \log\left(\rho_i^d V_d \tilde{\gamma}(N-1)\right) \tag{16}$$

where $\rho_i = \min\left(\rho(X_i,X_j) : j\in\{1,\ldots,N\}\setminus\{i\}\right)$, $\tilde{\gamma} = e^{\gamma}$ where $\gamma = -\int_0^{\infty} e^{-t}\log t dt$ is the Euler constant, and $V_d = \pi^{d/2}/\Gamma(d/2+1)$ is the volume of the unit ball in $\mathbb{R}^d$. In the sequel we will denote $B(x,r) = \left\{y\in\mathbb{R}^d :\ \rho(x,y)\leq r\right\}$ the ball with radius $r\geq 0$ and centered at point $x\in\mathbb{R}^d$.

The consistency proof for this estimator relies on two main steps: i) proving first that $E(|\xi_1(N)|) < +\infty$ and $E(\xi_1(N))\to H$ (asymptotic unbiasedness); ii) the risk of H_N is asymptotically null. The construction of H_N itself is directly connected to step i). To understand this point the key technical argument is that

$$E(\xi_1(N)|X_1 = x) \to -\log f(x), \quad \text{as} \quad N\to+\infty, \tag{17}$$

the Monte Carlo average in (16) converging somehow towards $\lim_{N\to+\infty} E(E(\xi_1(N)|X_1)) = E(E(\lim_{N\to+\infty}\xi_1(N)|X_1)) = E(-\log f(X)) = H(f)$. Let us explain now briefly how the

convergence result (17) happens. Considering the conditional cdf of $e^{\xi_1(N)}$ given $\{X_1 = x\}$ it comes:

$$\begin{aligned} F_{N,x}(u) &= P(e^{\xi_1(N)} \leq u | X_1 = x) \\ &= P(\xi_{N,x} \leq u), \quad \text{where } \xi_{N,x} = \min_{j=2,\ldots,N} \rho^d(x, X_j) V_d \tilde{\gamma}(N-1) \\ &= P(\min_{j=2,\ldots,N} \rho(x, X_j) \leq r_N(u)), \quad \text{where } r_N(u) = (u/V_d\tilde{\gamma}(N-1))^{1/d} \\ &= 1 - (1 - P(X \in B(x, r_N(u)))^{N-1} \\ &= 1 - \left(1 - \int_{B(x,r_N(u))} f(y)dy\right)^{N-1}. \end{aligned}$$

Noticing that $\text{Vol}(B(x, r_N(u)) = V_d(r_N(u))^d = u/(\tilde{\gamma}(N-1))$, and that according to the Lebesgue differentiation theorem $\int_{B(x,r_N(u))} f(y)dy/\text{Vol}(B(x, r_N(u)) = f(x) + \alpha_N(x, u)$ where $\alpha_N(x, u) \to 0$ as $r_N \to +\infty$, it comes that

$$\begin{aligned} \lim_{N\to+\infty} F_{N,x}(u) &= 1 - \lim_{N\to+\infty} \left(1 - \frac{u}{\tilde{\gamma}(N-1)} \times \frac{\int_{B(x,r_N(u))} f(y)dy}{\text{Vol}(B(x, r_N(u))}\right)^{N-1} \\ &= 1 - \exp\left(-\frac{f(x)u}{\tilde{\gamma}}\right). \end{aligned}$$

This last results shows in particular that the random variable $\xi_{N,x}$ converges in law towards a ξ_x random variable with $\mathcal{E}(f(x)/\tilde{\gamma})$ distribution, where $\mathcal{E}$ denotes the exponential distribution. As a consequence $\log \xi_{N,x}$ converges also in law towards $\log \xi_x$ as $N \to +\infty$. Noticing that for any random variable η such that $\eta \sim \mathcal{E}(\lambda)$, for $\lambda > 0$, we have:

$$E(\log(\eta)) = \int_0^{+\infty} \log u \lambda e^{-\lambda u} du = \underbrace{\int_0^{+\infty} e^{-t} \log t dt}_{-\gamma = -\log(\tilde{\gamma})} - \log \lambda = -\log(\lambda\tilde{\gamma}). \tag{18}$$

Considering now $\lambda = f(x)/\tilde{\gamma}$ we obtain the wanted result: $E(\log \xi_x) = -\log f(x)$, the final convergence in expectation of $E(\log \xi_{N,x}) \to E(\log \xi_x) = -\log f(x)$, as $n \to +\infty$ being handled using the Theorem 3.5 in Billingsley (1995) under some uniform integrability condition on the random variables family $\{\log \xi_{N,x};\ N \geq N_0(x)\}$, where $N_0(x)$ is suitable rank to be determined.

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.

Atchadé, Y. (2019). Approximate spectral gaps for Markov chains mixing times in high-dimension. *ArXiv e-prints*.

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.

Berrett, T. B., Samworth, R. J., and Yuan, M. (2019). Efficient multivariate entropy estimation via k-nearest neighbour distances. *Ann. Statist.*, 47:288–318.

Bulinski, A. and Dimitrov, D. (2019). Statistical estimation of the shannon entropy. *Acta Math. Sinica*, 35:17–46.

Charzyńska, A. and Gambin, A. (2015). Improvement of the k-NN entropy estimator with applications in systems biology. *Entropy*, 18:1–19.

Chauveau, D. and Alrachid, H. (2019). *EntropyMCMC: An R Package for MCMC Simulation and Convergence Evaluation using Entropy and Kullback Divergence Estimation*.

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

Chauveau, D. and Vandekerkhove, P. (2014). Simulation based nearest neighbor entropy estimation for (adaptive) MCMC evaluation. In *JSM Proceedings, Statistical Computing Section*, pages 2816–2827. American Statistical Association, Alexandria, VA.

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

Efron, B. and Morris, C. (1975). Data analysis using Stein's estimator and its generalizations. *J. Amer. Stat. Assoc.*, 70(350):311–319.

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.

Haario, H., Saksman, E., and Tamminen, J. (2001). An adaptive Metropolis algorithm. *Bernoulli*, 7(2):223–242.

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.

Maire, F. and Vandekerkhove, P. (2018). On Markov chain Monte Carlo for sparse and filamentary distributions. *ArXiv e-prints*.

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 (2018). *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.

Roberts, G. O. and Rosenthal, J. S. (2009). Examples of Adaptive MCMC. *Journal of Computational and Graphical Statistics*, 18(2):349–367.

Rosenthal, J. (1996). Analysis of the Gibbs sampler for a model related to James-Stein estimators. *Stat. and Comp.*, 6:269–275.

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., Kulkarni, S. R., and Verdú, S. (2006). A nearest-neighbor approach to estimating divergence between continuous random vectors.

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

Yang, J. and Rosenthal, J. S. (2019). Complexity results for MCMC derived from quantitative bounds. *ArXiv e-prints*.

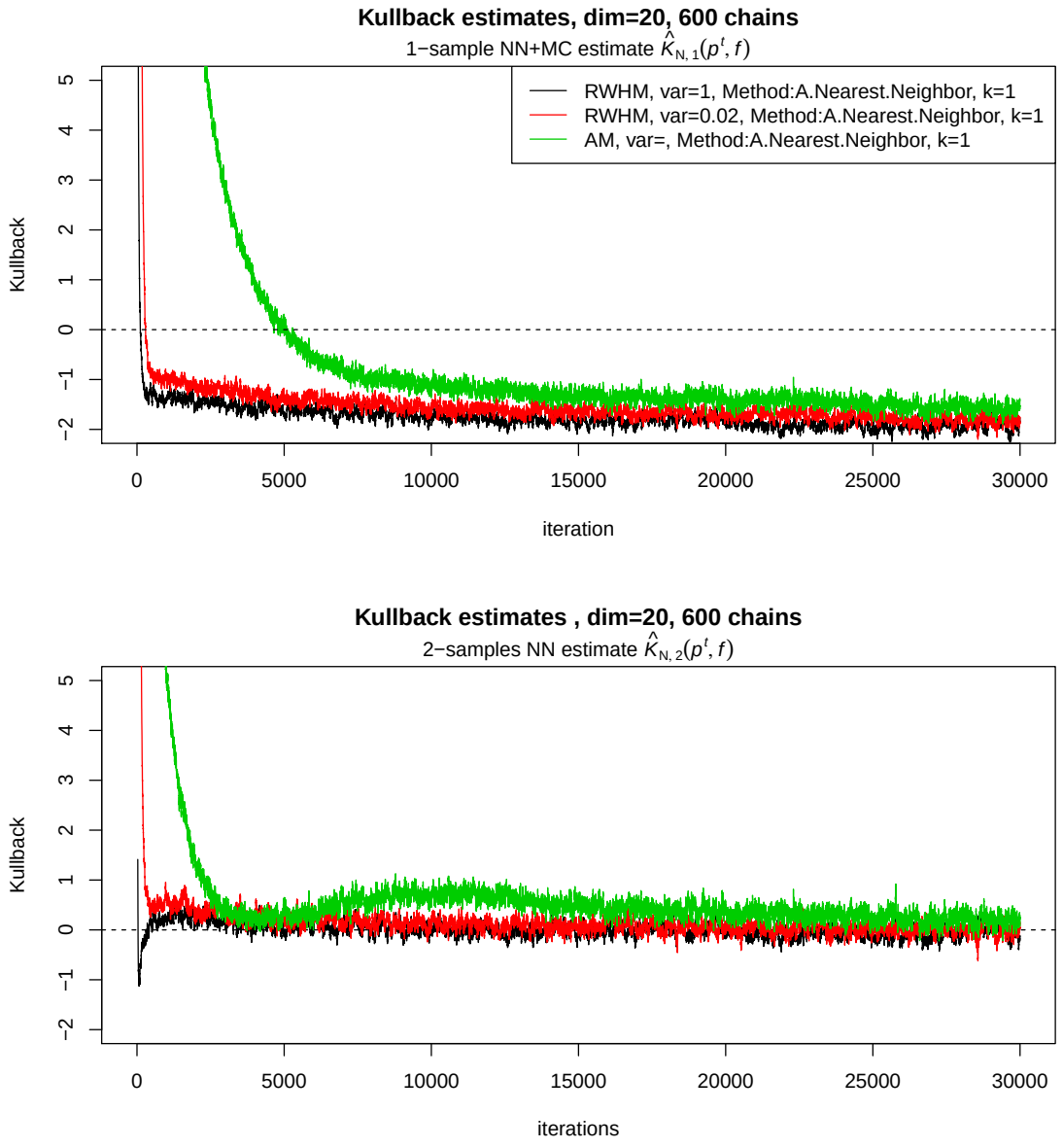

Figure 1: Banana-shaped target pdf, $d = 20$. Top: default plot from the EntropyMCMC package for MCMC's comparisons using the NN+MC biased estimate $\hat{\mathcal{K}}_{N,1}(p^t, f)$. Bottom: the two-sample estimate $\hat{\mathcal{K}}_{N,2}(p^t, f)$.

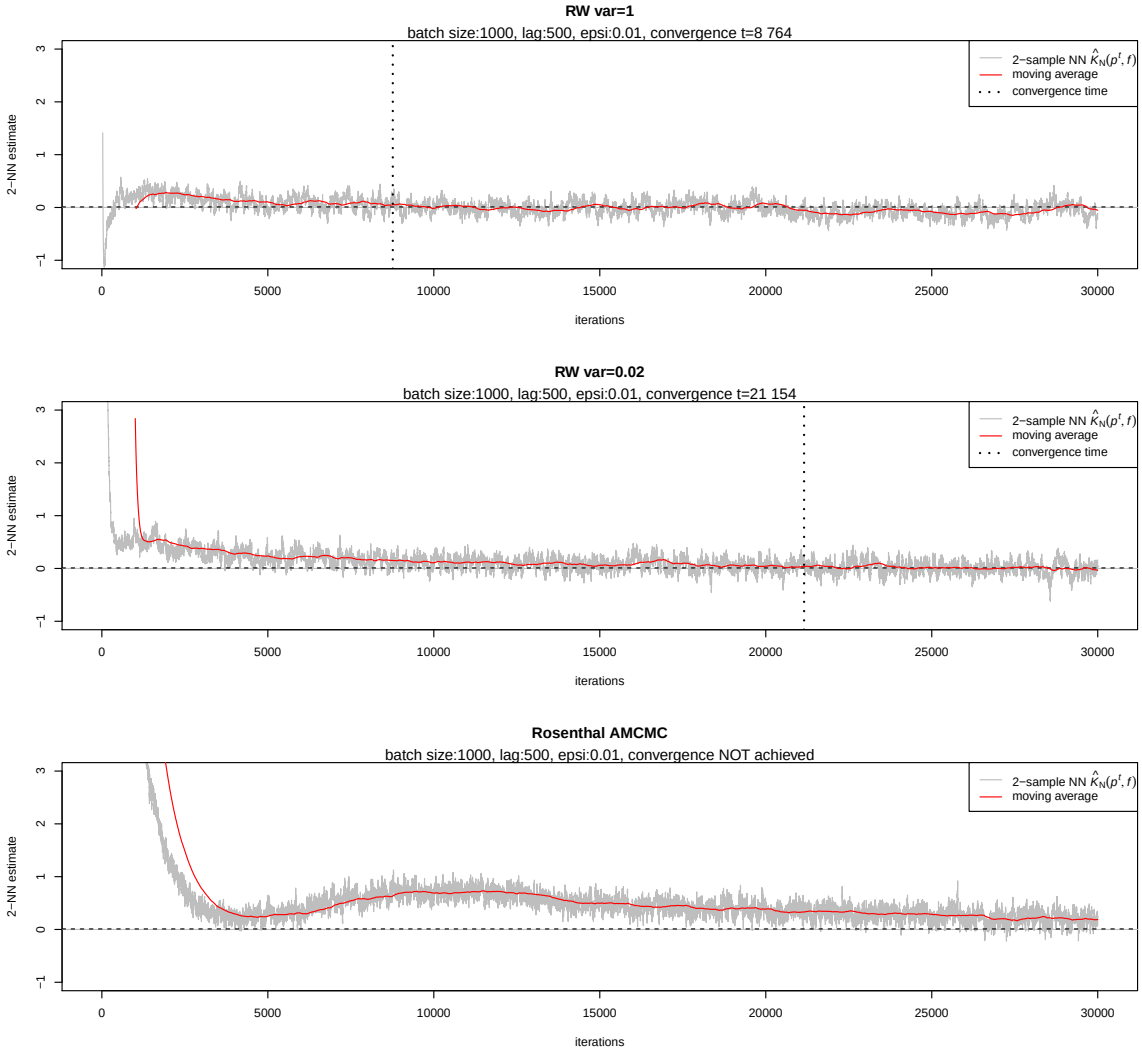

Figure 2: Banana-shaped target pdf, $d = 20$: Automatic convergence criterion for RW1, RW2 and AM based on stabilization near 0 of the sequences of two-sample estimates $t \mapsto \hat{\mathcal{K}}_{N,2}(p^t, f)$.

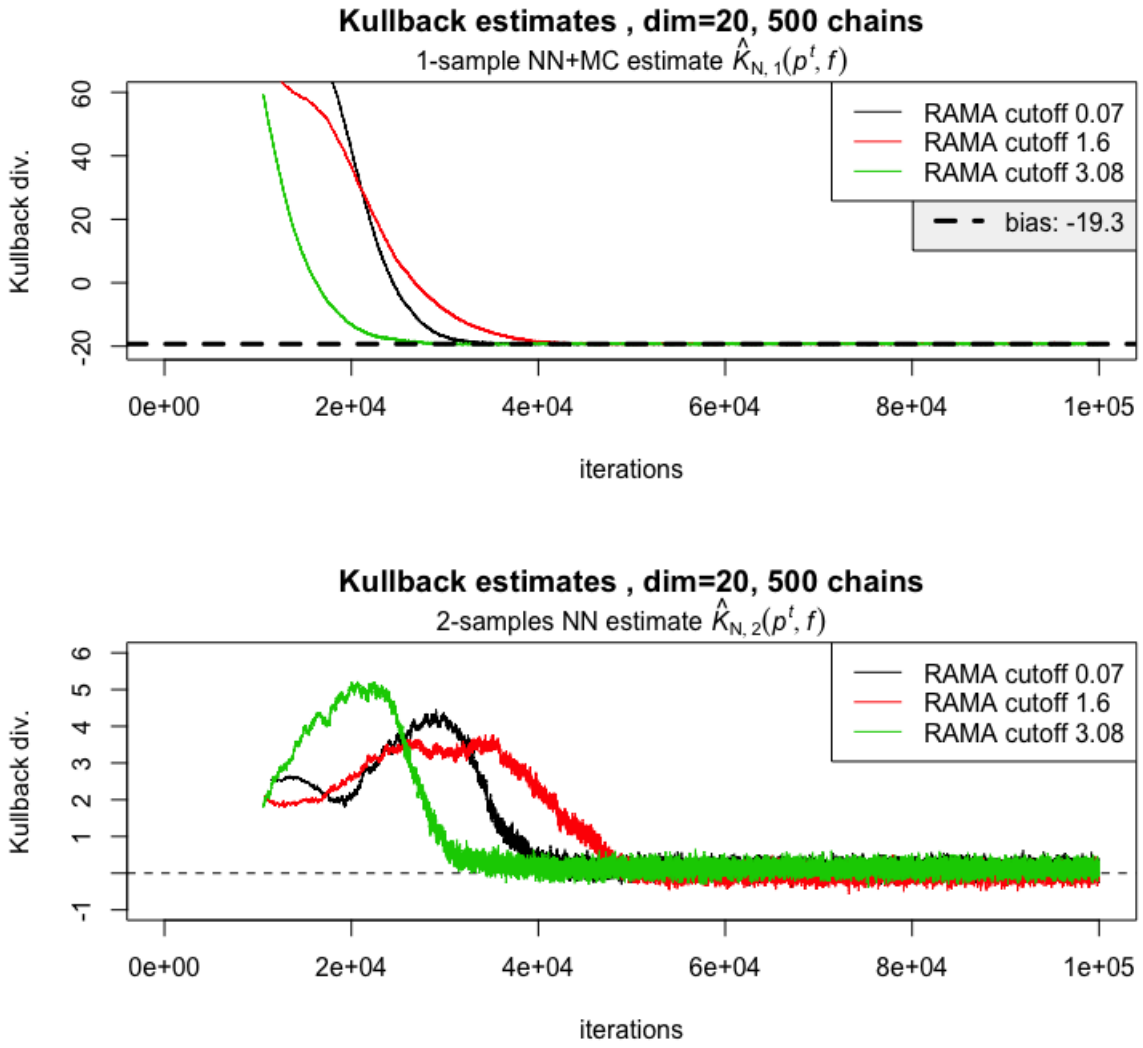

Figure 3: Baseball data and RAMA algorithm, $d = 20$. Top: default plot from the Entropy-MCMC package for three values of cutoff between regions, using the NN+MC biased estimate $\hat{\mathcal{K}}_{N,1}(p^t, f)$. Bottom: Same setting with the two-sample estimates $\hat{\mathcal{K}}_{N,2}(p^t, f)$.

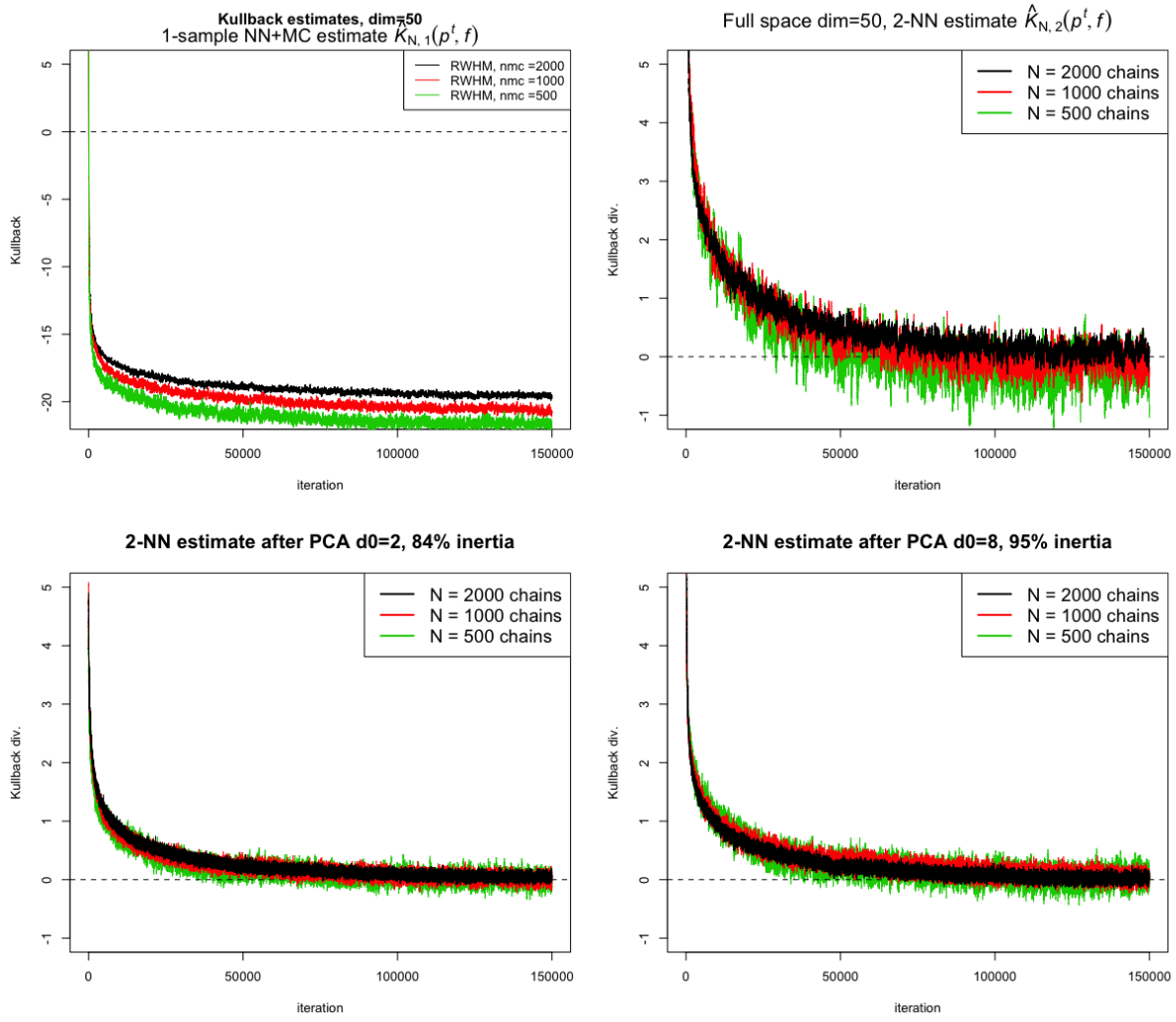

Figure 4: Gaussian target for efficient PCA, $d = 20$. Top left: default plot from the EntropyMCMC package for three values of N, using the NN+MC biased estimate $\hat{\mathcal{K}}_{N,1}(p^t, f)$. Top right: 2-NN estimate $\hat{\mathcal{K}}_{N,2}(p^t, f)$ on the full space $d = 50$. Bottom: 2-NN estimates $\hat{\mathcal{K}}_{N,2}(h^t, h^\varepsilon)$ as detailed in Section 6, for two projections in dimensions $d' = 2$ and $d' = 8$.