Recycling Gibbs Sampling

Luca Martino*, Víctor Elvira†, Gustau Camps-Valls*
* Image Processing Laboratory, Universitat de València (Spain).
† IMT Lille Douai CRISTAL (UMR 9189), Villeneuve d'Ascq (France).

***Abstract*—Gibbs sampling is a well-known Markov chain Monte Carlo (MCMC) algorithm, extensively used in signal processing, machine learning and statistics. The key point for the successful application of the Gibbs sampler is the ability to draw samples from the full-conditional probability density functions efficiently. In the general case this is not possible, so in order to speed up the convergence of the chain, it is required to generate auxiliary samples. However, such intermediate information is finally disregarded. In this work, we show that these auxiliary samples can be recycled within the Gibbs estimators, improving their efficiency with no extra cost. Theoretical and exhaustive numerical comparisons show the validity of the approach.**

Keywords: Bayesian inference, Markov Chain Monte Carlo (MCMC), Gibbs sampling, Gaussian Processes (GP)

I. Introduction

Many applications in statistical signal processing, machine learning and statistics, demand fast and accurate procedures for drawing samples from probability distributions that exhibit arbitrary, non-standard forms [1]–[5]. One of the most popular approaches are the Markov chain Monte Carlo (MCMC) algorithms [1], [6]. MCMC techniques generate a Markov chain (i.e., a sequence of correlated samples) with a pre-established target probability density function (pdf) as invariant density [7].

The Gibbs sampling technique is a well-known MCMC algorithm, extensively used in the literature in order to generate samples from multivariate target densities, drawing each component of the samples from the full-conditional densities [8]–[13]. In order to draw samples from a multivariate target distribution, the key point for the successful application of the standard Gibbs sampler is the ability to draw efficiently from the univariate conditional pdfs [6], [7]. The best scenario for Gibbs sampling occurs when specific direct samplers are available for each full-conditional, e.g. inversion method or, more generally, some transformation of a random variable [6], [14]. Otherwise, other Monte Carlo techniques, such as rejection sampling (RS) and different flavors of the Metropolis-Hastings (MH) algorithms, are typically used *within* the Gibbs sampler to draw from the complicated full-conditionals. The performance of the resulting Gibbs sampler depends on the employed internal technique, as pointed out for instance in [15]–[18].

In this context, some authors have suggested using more steps of the MH method within the Gibbs sampler [19]–[21]. Moreover, other different algorithms have been proposed as alternatives to the MH technique [9], [15], [22], [23]. For instance, several automatic and self-tuning samplers have been designed to be used primarily *within-Gibbs*: the adaptive rejection sampling (ARS) [24], [25], the griddy Gibbs sampler [26], the FUSS sampler [18], the Adaptive Rejection Metropolis Sampling (ARMS) method [13], [16], [27], [28], and the Independent Doubly Adaptive Rejection Metropolis Sampling (IA^{2}RMS) technique [17], just to name a few.

Most of the previous solutions require performing several MCMC steps for each full-conditional in order to improve the performance, although only one of them is considered to produce the resulting Markov chain because the rest of samples play the mere role of auxiliary variables. Strikingly, they require an increase in the computational cost that is not completely paid off: several samples are drawn from the full-conditionals, but only a subset of these generated samples is employed in the final estimators. In this work, we show that the rest of generated samples can be directly incorporated within the corresponding Gibbs estimator. We call this approach the *Recycling Gibbs (RG) sampler* since all the samples drawn from each full-conditional can be used also to provide a better estimation, instead of discarding them.

The consistency of the proposed RG estimators is guaranteed, as will be noted after considering the connection between the Gibbs scheme and the chain rule for sampling purposes [6], [14]. RG fits particularly well combined with adaptive MCMC schemes where different internal steps are performed also for adapting the proposal density, see e.g. [13], [16], [17], [28]. The novel RG scheme allows us to obtain better performance without adding any extra computational cost as shown by numerical simulations. We test RG for learning the hyperparameters of a Gaussian Process with automatic relevance determination (ARD) kernel [29].

II. Problem Statement and Background

In many applications, the goal is to infer a variable of interest, $\mathbf{x} = [x_1, \ldots, x_D] \in \mathbb{R}^D$, given a set of observations or measurements, $\mathbf{y} \in \mathbb{R}^P$. In Bayesian inference, we obtain the posterior pdf

$$\bar{\pi}(\mathbf{x}) = p(\mathbf{x}|\mathbf{y}) = \frac{\ell(\mathbf{y}|\mathbf{x})g(\mathbf{x})}{Z(\mathbf{y})}, \qquad (1)$$

where $\ell(\mathbf{y}|\mathbf{x})$ is the likelihood function, $g(\mathbf{x})$ is the prior pdf and $Z(\mathbf{y})$ is the marginal likelihood (a.k.a., Bayesian evidence). In general, $Z(\mathbf{y})$ is unknown and difficult to estimate then we assume that we are able to evaluate the unnormalized target function,

$$\pi(\mathbf{x}) = \ell(\mathbf{y}|\mathbf{x})g(\mathbf{x}). \qquad (2)$$

ISBN 978-0-9928626-7-1 © EURASIP 2017

The analytical study of the posterior density $\bar{\pi}(\mathbf{x}) \propto \pi(\mathbf{x})$ is often unfeasible and integrals involving $\bar{\pi}(\mathbf{x})$ are typically intractable. For instance, one might be interested in computing

$$I = \int_{\mathbb{R}^D} f(\mathbf{x})\bar{\pi}(\mathbf{x})d\mathbf{x}, \tag{3}$$

where $f(\mathbf{x})$ is an integrable function with respect to $\bar{\pi}$. In order to compute the intractable integral I, numerical approximations are typically required. Our goal here is to approximate this integral by using a Monte Carlo (MC) quadrature [6], [7]. Namely, considering T independent samples from the posterior target pdf, i.e., $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(T)} \sim \bar{\pi}(\mathbf{x})$, we build the estimator

$$\widehat{I}_T = \frac{1}{T}\sum_{t=1}^{T} f(\mathbf{x}^{(t)}) \xrightarrow{p} I. \tag{4}$$

This means that for the weak law of large numbers, $\widehat{I}_T$ converges in probability to I. In general, a direct method for drawing independent samples from $\bar{\pi}(\mathbf{x})$ is not available, and alternative approaches, e.g., MCMC algorithms, are needed.

A. The Standard Gibbs (SG) sampler

The Gibbs sampler is arguably the most used MCMC algorithm in signal processing, statistics and machine learning [6], [8], [9], [11]. Let us define $\mathbf{x}_{\neg d} := [x_1, \ldots, x_{d-1}, x_{d+1}, \ldots, x_D]$ and introduce the following equivalent notations

$$\bar{\pi}_d(x_d|x_{1:d-1}, x_{d+1:D}) \quad \equiv \quad \bar{\pi}_d(x_d|\mathbf{x}_{\neg d}).$$

In order to denote the unidimensional full-conditional pdf of the component $x_d \in \mathbb{R}$, $d \in \{1, \ldots, D\}$, given the rest of variables $\mathbf{x}_{\neg d}$, i.e.

$$\bar{\pi}_d(x_d|\mathbf{x}_{\neg d}) = \frac{\bar{\pi}(\mathbf{x})}{\bar{\pi}_{\neg d}(\mathbf{x}_{\neg d})} = \frac{\bar{\pi}(\mathbf{x})}{\int_{\mathbb{R}} \bar{\pi}(\mathbf{x})dx_d}. \tag{5}$$

The density $\bar{\pi}_{\neg d}(\mathbf{x}_{\neg d}) = \int_{\mathbb{R}} \bar{\pi}(\mathbf{x})dx_d$ is the joint pdf of all variables, except x_d. The Gibbs algorithm generates a sequence of T samples, and is formed by the steps in Alg. 1.

Algorithm 1 The Standard Gibbs (SG) algorithm

```
Fix $T$, $D$
for $t = 1, \ldots, T$ do
    for $d = 1, \ldots, D$ do
        Draw $x_d^{(t)} \sim \bar{\pi}_d(x_d|x_{1:d-1}^{(t)}, x_{d+1:D}^{(t-1)})$
    end for
    Set $\mathbf{x}^{(t)} = [x_1^{(t)}, x_2^{(t)}, \ldots, x_D^{(t)}]$
end for
```

B. Monte Carlo-within-Gibbs sampling

The main assumption for the application of Gibbs sampling is the ability to draw efficiently from these univariate full-conditional pdfs $\bar{\pi}_d$, which is not possible in general. Thus, other Monte Carlo techniques are needed for drawing from $\bar{\pi}_d$. For instance, depending on the specific scenario, the alternatives are: the adaptive rejection samplers (ARS) [25], [30]–[32] when they can be applied, and additional MCMC samplers as the standard Metropolis-Hastings (MH) method or its adaptive/automatic versions [15]–[17], [22], [26], [28], [33], [34]. The application of other MCMC method *within Gibbs* can require the generation of intermediate points but only one of them is used for the next iteration of the Gibbs sampler [16], [19]–[21]. In this work, we show that these auxiliary samples can employed inside the final estimators.

III. Chain Rule and the Gibbs Sampling

For the sake of simplicity, let us consider a bivariate target pdf that can be factorized according to the chain rule,

$$\begin{aligned} \bar{\pi}(x_1, x_2) &= \bar{\pi}_2(x_2|x_1)p_1(x_1) \\ &= \bar{\pi}_1(x_1|x_2)p_2(x_2), \end{aligned}$$

where p_1 and p_2 denote the marginal pdfs and, $\bar{\pi}_2$ and $\bar{\pi}_1$, are the conditional pdfs. Let us consider the first equality. Clearly, if we are able to draw from the marginal pdf $p_1(x_1)$ and from the conditional pdf $\bar{\pi}_2(x_2|x_1)$, we can draw samples from $\bar{\pi}(x_1, x_2)$ following the chain rule procedure in Alg. 2. Note that, consequently, the T independent random vectors $[x_1^{(t)}, x_2^{(t)}]$, with $t = 1, \ldots, T$, are all distributed as $\bar{\pi}(x_1, x_2)$.

Algorithm 2 Chain rule method

```
for $t = 1, \ldots, T$ do
    Draw $x_1^{(t)} \sim p_1(x_1)$ and $x_2^{(t)} \sim \bar{\pi}_2(x_2|x_1^{(t)})$
end for
```

A. Standard Gibbs sampler as the chain rule

Considering the previous bivariate case, the standard Gibbs sampler consists in the following two steps (a) $x_2^{(t)} \sim \bar{\pi}_1(x_2|x_1^{(t-1)})$, (b) $x_1^{(t)} \sim \bar{\pi}_2(x_1|x_2^{(t)})$ and then set $\mathbf{x}^{(t)} = [x_1^{(t)}, x_2^{(t)}]$. After the burn-in period, i.e., $t \geq t_b$, we have $\mathbf{x}^{(t)} \sim \bar{\pi}(\mathbf{x})$. Therefore, recalling that $\bar{\pi}(x_1, x_2) = \bar{\pi}_2(x_2|x_1)p_1(x_1) = \bar{\pi}_1(x_1|x_2)p_2(x_2)$ for $t \geq t_b$, each component of the vector $\mathbf{x}^{(t)} = [x_1^{(t)}, x_2^{(t)}]$ is distributed as the corresponding marginal pdf, i.e., $x_1^{(t)} \sim p_1(x_1)$ and $x_2^{(t)} \sim p_2(x_2)$. Therefore, after t_b iterations, the standard Gibbs sampler can be interpreted as the application of the chain rule procedure of Alg. 2.

B. Alternative chain rule procedure

An alternative procedure is shown in Alg. 3. This chain rule draws M samples from the full conditional $\bar{\pi}_2(x_2|x_1)$ at each t-th iteration, and generates samples from the joint pdf $\bar{\pi}(x_1, x_2)$.

Algorithm 3 An alternative chain rule procedure

```
for $t = 1, \ldots, T$ do
    Draw $x_1^{(t)} \sim p_1(x_1)$
    Draw $x_{2,m}^{(t)} \sim \bar{\pi}_2(x_2|x_1^{(t)})$, with $m = 1, \ldots. M$
end for
```

Note that all the TM vectors, $[x_1^{(t)}, x_{2,m}^{(t)}]$, with $t = 1, \ldots, T$ and $m = 1, \ldots, M$, are samples from $\bar{\pi}(x_1, x_2)$. This scheme is valid and, in some cases, it can present some benefits w.r.t. the traditional scheme in terms of performance, depending on certain statistical features of the joint pdf $\bar{\pi}(x_1, x_2)$. For instance, the correlation between variables x_1 and x_2, and the variances of the marginal pdfs $p_1(x_1)$ and $p_2(x_2)$.

At this point, a natural question arises: is it possible to design a Gibbs sampling scheme equivalent to the alternative chain rule scheme described before? The answer is in the next section.

IV. The Multiple Recycling Gibbs sampler

Based on the previous considerations, we design the *Multiple Recycling Gibbs* (MRG) sampler which draws $M > 1$ samples from each full conditional pdf, as shown in Alg. 4.

Algorithm 4 Multiple Recycling Gibbs (MRG) sampler

```
1:  Choose a starting point $[z_1^{(0)}, \dots, z_D^{(0)}]$
2:  for $t = 1, \dots, T$ do
3:      for $d = 1, \dots, D$ do
4:          for $m = 1, \dots, M$ do
5:              Draw $x_{d,m}^{(t)} \sim \bar{\pi}_d(x_d | z_{1:d-1}^{(t)}, z_{d+1:D}^{(t-1)})$
6:              Set $\mathbf{x}_{d,m}^{(t)} = [z_{1:d-1}^{(t)}, x_{d,m}^{(t)}, z_{d+1:D}^{(t-1)}]$
7:          end for
8:          Set $z_d^{(t)} = x_{d,M}^{(t)}$
9:      end for
10: end for
11: return $\{\mathbf{x}_{d,m}^{(t)}\}$ for all $d$, $m$ and $t$
```

For a given test function $f(\mathbf{x})$ in the integral of Eq. (3), the MRG estimator is eventually formed by TDM samples, i.e., without removing any burn-in period, as

$$\widehat{I}_T = \frac{1}{TDM} \sum_{t=1}^{T} \sum_{d=1}^{D} \sum_{m=1}^{M} f(\mathbf{x}_{d,m}^{(t)}). \tag{6}$$

Observe that in order to go forward to sampling from the next full-conditional, we only consider the last generated component, i.e., $z_d^{(t)} = x_{d,M}^{(t)}$. However, an alternative to step 8 of Algorithm 4 is: (a) draw $j \sim \mathcal{U}(1, \dots, M)$ and (b) set $z_d^{(t)} = x_{d,j}^{(t)}$. Note that choosing the last sample $x_{d,M}^{(t)}$ is more convenient for an MCMC-within-MRG scheme.

The MRG sampler is equivalent to the alternative chain rule scheme described in the previous section, so that the consistency of the MRG estimators is guaranteed. The ergodicity of the generated chain is also ensured since the dynamics of the MRG scheme is identical to the dynamics of the SG sampler, although they differ in the construction of final estimators.

The MRG approach is convenient in terms of accuracy and computational efficiency, as also confirmed by the numerical results in Section V. MRG is particularly advisable if an adaptive MCMC is employed to draw from the full-conditional pdfs, i.e., when several MCMC steps are performed for sampling from each full-conditional and adapting the proposal. We can use all the sequence of samples generated by the internal MCMC algorithm in the resulting estimator.

V. Numerical Example: Learning Hyperparameters in Gaussian Processes

In section, we test the proposed approach for the estimation of hyperparameters of the Automatic Relevance Determination (ARD) kernel function for Gaussian processes (GPs) [29, Chapter 6], [35]. The MATLAB code of this numerical example is provided at http://isp.uv.es/code/RG.zip.

Let us assume observed data pairs $\{y_j, \mathbf{z}_j\}_{j=1}^{P}$, with $y_j \in \mathbb{R}$ and

$$\mathbf{z}_j = [z_{j,1}, z_{j,2}, \dots, z_{j,L}]^\top \in \mathbb{R}^L,$$

where L is the dimension of the input features. We also denote the corresponding $P \times 1$ output vector as $\mathbf{y} = [y_1, \dots, y_P]^\top$ and the $L \times P$ input matrix as $\mathbf{Z} = [\mathbf{z}_1, \dots, \mathbf{z}_P]$. We address the regression problem of inferring the unknown function f which links the variable y and $\mathbf{z}$. Thus, the assumed model is

$$y = f(\mathbf{z}) + e, \tag{7}$$

where $e \sim N(e; 0, \sigma^2)$, and that $f(\mathbf{z})$ is a realization of a GP [35]. Hence $f(\mathbf{z}) \sim \mathcal{GP}(\mu(\mathbf{z}), \kappa(\mathbf{z}, \mathbf{r}))$ where $\mu(\mathbf{z}) = 0$, $\mathbf{z}, \mathbf{r} \in \mathbb{R}^L$, and we consider the ARD kernel function

$$\kappa(\mathbf{z}, \mathbf{r}) = \exp\left(-\sum_{\ell=1}^{L} \frac{(z_\ell - r_\ell)^2}{2\delta_\ell^2} \right), \quad \text{with} \quad \delta_\ell > 0 \tag{8}$$

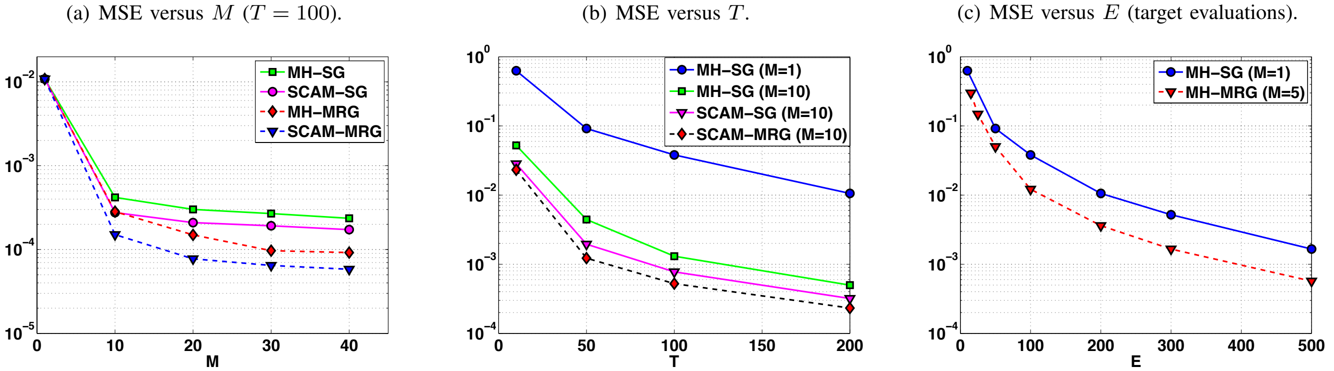
and $\ell = 1, \dots, L$. Note that we have a different hyper-parameter δ_ℓ for each input component z_ℓ, hence we also define $\boldsymbol{\delta} = \delta_{1:L} = [\delta_1, \dots, \delta_L]$. Using ARD allows us to infer the relative importance of different components of inputs: a small value of δ_ℓ means that a variation of the ℓ-component z_ℓ impacts the output more, while a high value of δ_ℓ shows virtually independence between the ℓ-component and the output [29, Chapter 6]. Given these assumptions, the vector $\mathbf{f} = [f(\mathbf{z}_1), \dots, f(\mathbf{z}_P)]^\top$ is distributed as

$$p(\mathbf{f} | \mathbf{Z}, \boldsymbol{\delta}, \kappa) = \mathcal{N}(\mathbf{f}; \mathbf{0}, \mathbf{K}), \tag{9}$$

where $\mathbf{0}$ is a $P \times 1$ null vector, and $\mathbf{K}_{ij} := \kappa(\mathbf{z}_i, \mathbf{z}_j)$, for all $i, j = 1, \dots, P$, is a $P \times P$ matrix. Note that in Eq. (9) we have expressed explicitly the dependence on the input matrix $\mathbf{Z}$, on the vector $\boldsymbol{\delta}$ and on the choice of the kernel family κ. Therefore, the vector containing all the hyper-parameters of the model is $\boldsymbol{\theta} = [\theta_{1:L} = \delta_{1:L}, \theta_{L+1} = \sigma] = [\boldsymbol{\delta}, \sigma]$, i.e., all the parameters of the kernel function in Eq. (8) and standard deviation σ of the observation noise. Considering the filtering scenario and the tuning of the parameters (i.e., inferring the vectors $\mathbf{f}$ and $\boldsymbol{\theta}$), the full Bayesian solution addresses the study of the full posterior pdf involving $\mathbf{f}$ and $\boldsymbol{\theta}$,

$$p(\mathbf{f}, \boldsymbol{\theta} | \mathbf{y}, \mathbf{Z}, \kappa) = \frac{p(\mathbf{y} | \mathbf{f}, \mathbf{Z}, \boldsymbol{\theta}, \kappa) p(\mathbf{f} | \mathbf{z}, \boldsymbol{\theta}, \kappa) p(\boldsymbol{\theta})}{p(\mathbf{y} | \mathbf{Z}, \kappa)}, \tag{10}$$

where $p(\mathbf{y} | \mathbf{f}, \mathbf{Z}, \boldsymbol{\theta}, \kappa) = \mathcal{N}(\mathbf{y}; \mathbf{0}, \sigma^2 \mathbf{I})$ given the observation model in Eq. (7), $p(\mathbf{f} | \mathbf{z}, \boldsymbol{\theta}, \kappa)$ is given in Eq. (9), and $p(\boldsymbol{\theta})$ is the prior over the hyper-parameters. We assume $p(\boldsymbol{\theta}) = \prod_{\ell=1}^{L+1} \frac{1}{\theta_\ell^\beta} \mathbb{I}_{\theta_\ell}$ where $\beta = 1.3$, and $\mathbb{I}_v = 1$ if $v > 0$, whereas $\mathbb{I}_v = 0$ otherwise. Note that the posterior in Eq. (10) is analytically intractable but, given a fixed vector $\boldsymbol{\theta}'$, the marginal



(a) MSE versus M ($T = 100$).

(b) MSE versus T.

(c) MSE versus E (target evaluations).

Figure 1. MSE (log-scale) of different MCMC-within-Gibbs schemes **(a)** as function of M ($T = 100$ and $D = 2$), **(b)** as function of T for different techniques (in this case, $D = 4$), with $M = 1$ for the MH-within-SG method depicted with a solid line and circles, whereas $M = 10$ for the remaining curves, **(c)** as function of the total number of target evaluations $E = MT$ ($D = 4$). Namely, for MH-within-SG we have $M = 1$ and $T \in \{10, 50, 100, 200, 300, 500\}$, whereas for MH-within-MRG we have $M = 5$ and $T \in \{3, 5, 10, 20, 40, 60, 100\}$. The MRG approaches, shown with dashed lines, always outperform the corresponding standard Gibbs (SG) schemes, shown with solid lines.

posterior of $p(\mathbf{f}|\mathbf{y}, \mathbf{Z}, \boldsymbol{\theta}', \kappa) = \mathcal{N}(\mathbf{f}; \boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$ is known in closed-form: it is Gaussian with mean $\boldsymbol{\mu}_p = \mathbf{K}(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y}$ and covariance matrix $\boldsymbol{\Sigma}_p = \mathbf{K} - \mathbf{K}(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{K}$ [35]. For the sake of simplicity, in this experiment we focus on the marginal posterior density of the hyperparameters,

$$\bar{p}(\boldsymbol{\theta}|\mathbf{y}, \mathbf{Z}, \kappa) \propto p(\boldsymbol{\theta}|\mathbf{y}, \mathbf{Z}, \kappa) = p(\mathbf{y}|\boldsymbol{\theta}, \mathbf{Z}, \kappa)p(\boldsymbol{\theta}),$$

which can be evaluated analytically, but we cannot compute integrals involving it. Actually, since $p(\mathbf{y}|\boldsymbol{\theta}, \mathbf{Z}, \kappa) = \mathcal{N}(\mathbf{y}; \mathbf{0}, \mathbf{K} + \sigma^2\mathbf{I})$ and $p(\boldsymbol{\theta}|\mathbf{y}, \mathbf{Z}, \kappa) \propto p(\mathbf{y}|\boldsymbol{\theta}, \mathbf{Z}, \kappa)p(\boldsymbol{\theta})$, we have

$$\begin{aligned}\log\left[p(\boldsymbol{\theta}|\mathbf{y}, \mathbf{Z}, \kappa)\right] = &-\frac{1}{2}\mathbf{y}^\top(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y} \\ &-\frac{1}{2}\log\left[\det\left[\mathbf{K} + \sigma^2\mathbf{I}\right]\right] - \beta\sum_{\ell=1}^{L+1}\log\theta_\ell,\end{aligned}$$

with $\theta_\ell > 0$, where clearly $\mathbf{K}$ depends on $\theta_{1:L} = \delta_{1:L}$ and recall that $\theta_{L+1} = \sigma$ [35]. However, the moments of this marginal posterior cannot be computed analytically. Then, in order to compute the Minimum Mean Square Error (MMSE) estimator, i.e., the expected value $\mathbb{E}[\boldsymbol{\Theta}]$ with $\boldsymbol{\Theta} \sim p(\boldsymbol{\theta}|\mathbf{y}, \mathbf{Z}, \kappa)$, we approximate $\mathbb{E}[\boldsymbol{\Theta}]$ via Monte Carlo quadrature. More specifically, we apply a Gibbs-type samplers to draw from $\pi(\boldsymbol{\theta}) \propto p(\boldsymbol{\theta}|\mathbf{y}, \mathbf{Z}, \kappa)$. Note that dimension of the problem is $D = L + 1$ since $\boldsymbol{\theta} \in \mathbb{R}^D$.

We generated the $P = 500$ pairs of data, $\{y_j, \mathbf{z}_j\}_{j=1}^P$, drawing $\mathbf{z}_j \sim \mathcal{U}([0, 10]^L)$ and $\mathbf{y}_j$ according to the model in Eq. (7), considered $L \in \{1, 3\}$ so that $D \in \{2, 4\}$, and set $\sigma^* = \frac{1}{2}$ for both cases, $\delta^* = 1$ and $\boldsymbol{\delta}^* = [1, 3, 1]$, respectively (recall that $\boldsymbol{\theta}^* = [\boldsymbol{\delta}^*, \sigma^*]$). Keeping fixed the generated data for each scenario, we then computed the ground-truths using an exhaustive and costly Monte Carlo approximation, in order to be able to compare the different techniques. We tested the standard MH within SG and MRG, and also the Single Component Adaptive Metropolis (SCAM) algorithm [33] within SG and MRG. SCAM is a component-wise version of the adaptive MH method [36] where the covariance matrix of the proposal is automatically adapted. In SCAM, the covariance matrix of the proposal is diagonal and each element is adapted considering only the corresponding component: that is, the variances of the marginal densities of the target pdf are estimated and used as a scale parameter of the proposal pdf in the corresponding component. We averaged the results using 10^3 independent runs. Figure 1(a) shows the MSE curves (in log-scale) of the different schemes as function of $M \in \{1, 10, 20, 30, 40\}$, while keeping fixed $T = 100$ (in this case, $D = 2$). Figure 1(b) depicts the MSE curves ($D = 4$) as function of T considering in one case $M = 1$ and $M = 10$ for the rest. In both figures, we notice that (1) using an $M > 1$ is advantageous in any case (SG or MRG), (2) using a procedure to adapt the proposal improves the results, and (3) MRG, i.e., recycling all the generated samples, always outperforms the SG schemes.

Figure 1(c) compares the MH-within-SG with the MH-within-MRG, showing the MSE as function of the total number of target evaluations $E = MT$. We set $M = 5$, $T \in \{3, 5, 10, 20, 40, 60, 100\}$ for MH-within-MRG, whereas we have $M = 1$ and $T \in \{10, 50, 100, 200, 300, 500\}$ for MH-within-SG. Namely, we used a longer Gibbs chain for MH-within-SG. Note that the MH-within-MRG provides always smaller MSEs, considering the same total number of evaluations E of the target density.

VI. Conclusions

In this work, we have shown that the efficiency of the Gibbs estimators can be improved including some generated auxiliary samples, without any extra computational cost. The consistency of the resulting estimators is ensured since the novel MRG scheme is equivalent to an alternative formulation of the well-known chain rule method.

Acknowledgements

This work has been supported by the European Research Council (ERC) through the ERC Consolidator Grant SEDAL ERC-2014-CoG 647423.

References

[1] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan, "An introduction to MCMC for machine learning," *Machine Learning*, vol. 50, no. 1, pp. 5–43, 2003.

[2] W. J. Fitzgerald, "Markov chain Monte Carlo methods with applications to signal processing," *Signal Processing*, vol. 81, no. 1, pp. 3–18, January 2001.

[3] L. Martino, J. Read, V. Elvira, and F. Louzada, "Cooperative parallel particle filters for on-line model selection and applications to urban mobility," *Digital Signal Processing*, vol. 60, no. 3, pp. 172–185, 2017.

[4] V. Elvira, L. Martino, D. Luengo, and M. Bugallo, "Efficient multiple importance sampling estimators," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1757–1761, 2015.

[5] ——, "Heretical multiple importance sampling," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1474–1478, 2016.

[6] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. Springer, 2004.

[7] J. S. Liu, *Monte Carlo Strategies in Scientific Computing*. Springer-Verlag, 2004.

[8] Y. Chen, L. Bornn, N. De Freitas, M. Eskelin, J. Fang, and M. Welling, "Herded Gibbs sampling," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 263–291, 2016.

[9] K. R. Koch, "Gibbs sampler by sampling-importance-resampling," *Journal of Geodesy*, vol. 81, no. 9, pp. 581–591, 2007.

[10] J. Kotecha and P. M. Djurić, "Gibbs sampling approach for generation of truncated multivariate Gaussian random variables," *Proceedings of Acoustics, Speech, and Signal Processing, (ICASSP)*, 1999.

[11] R. J. B. Goudie and S. Mukherjee, "A Gibbs sampler for learning DAGs," *Journal of Machine Learning Research*, vol. 17, no. 2, pp. 1–39, 2016.

[12] F. Lucka, "Fast Gibbs sampling for high-dimensional Bayesian inversion," *arXiv:1602.08595*, 2016.

[13] H. Zhang, Y. Wu, L. Cheng, and I. Kim, "Hit and run ARMS: adaptive rejection Metropolis sampling with hit and run random direction," *Journal of Statistical Computation and Simulation*, vol. 86, no. 5, pp. 973–985, 2016.

[14] L. Devroye, *Non-Uniform Random Variate Generation*. Springer, 1986.

[15] B. Cai, R. Meyer, and F. Perron, "Metropolis-Hastings algorithms with adaptive proposals," *Statistics and Computing*, vol. 18, pp. 421–433, 2008.

[16] W. R. Gilks, N. G. Best, and K. K. C. Tan, "Adaptive rejection Metropolis sampling within Gibbs sampling," *Applied Statistics*, vol. 44, no. 4, pp. 455–472, 1995.

[17] L. Martino, J. Read, and D. Luengo, "Independent doubly adaptive rejection Metropolis sampling within Gibbs sampling," *IEEE Transactions on Signal Processing*, vol. 63, no. 12, pp. 3123–3138, June 2015.

[18] L. Martino, H. Yang, D. Luengo, J. Kanniainen, and J. Corander, "A fast universal self-tuned sampler within Gibbs sampling," *Digital Signal Processing*, vol. 47, pp. 68 – 83, 2015.

[19] P. Müller, "A generic approach to posterior integration and Gibbs sampling," *Technical Report 91-09, Department of Statistics of Purdue University*, 1991.

[20] A. E. Gelfand and T. M. Lee, "Discussion on the meeting on the Gibbs sampler and other Markov Chain Monte Carlo methods," *Journal of the Royal Statistical Society. Series B*, vol. 55, no. 1, pp. 72–73, 1993.

[21] C. Fox, "A Gibbs sampler for conductivity imaging and other inverse problems," *Proc. of SPIE, Image Reconstruction from Incomplete Data VII*, vol. 8500, pp. 1–6, 2012.

[22] W. Shao, G. Guo, F. Meng, and S. Jia, "An efficient proposal distribution for Metropolis-Hastings using a b-splines technique," *Computational Statistics and Data Analysis*, vol. 53, pp. 465–478, 2013.

[23] L. Martino, R. Casarin, and D. Luengo, "Sticky proposal densities for adaptive MCMC methods," *IEEE Workshop on Statistical Signal Processing (SSP)*, 2016.

[24] W. R. Gilks, "Derivative-free adaptive rejection sampling for Gibbs sampling," *Bayesian Statistics*, vol. 4, pp. 641–649, 1992.

[25] W. R. Gilks and P. Wild, "Adaptive rejection sampling for Gibbs sampling," *Applied Statistics*, vol. 41, no. 2, pp. 337–348, 1992.

[26] C. Ritter and M. A. Tanner, "Facilitating the Gibbs sampler: The Gibbs stopper and the griddy-Gibbs sampler," *Journal of the American Statistical Association*, vol. 87, no. 419, pp. 861–868, 1992.

[27] W. R. Gilks, R. Neal, N. G. Best, and K. K. C. Tan, "Corrigidum: Adaptive rejection Metropolis sampling within Gibbs sampling," *Applied Statistics*, vol. 46, no. 4, pp. 541–542, 1997.

[28] R. Meyer, B. Cai, and F. Perron, "Adaptive rejection Metropolis sampling using Lagrange interpolation polynomials of degree 2," *Computational Statistics and Data Analysis*, vol. 52, no. 7, pp. 3408–3423, March 2008.

[29] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[30] W. Hörmann, "A rejection technique for sampling from T-concave distributions," *ACM Transactions on Mathematical Software*, vol. 21, no. 2, pp. 182–193, 1995.

[31] D. Görür and Y. W. Teh, "Concave convex adaptive rejection sampling," *Journal of Computational and Graphical Statistics*, vol. 20, no. 3, pp. 670–691, September 2011.

[32] L. Martino and J. Míguez, "A generalization of the adaptive rejection sampling algorithm," *Statistics and Computing*, vol. 21, no. 4, pp. 633–647, October 2011.

[33] H. Haario, E. Saksman, and J. Tamminen, "Component-wise adaptation for high dimensional MCMC," *Computational Statistics*, vol. 20, no. 2, pp. 265–273, 2005.

[34] R. A. Levine, Z. Yu, W. G. Hanley, and J. J. Nitao, "Implementing component-wise Hastings algorithms," *Computational Statistics and Data Analysis*, vol. 48, no. 2, pp. 363–389, 2005.

[35] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. MIT Press, 2006.

[36] H. Haario, E. Saksman, and J. Tamminen, "An adaptive Metropolis algorithm," *Bernoulli*, vol. 7, no. 2, pp. 223–242, April 2001.