ARTICLE IN PRESS

JID:YDSPR AID:2368 /FLA [m5G; v1.241; Prn:8/08/2018; 9:37] P.1 (1-19)

Digital Signal Processing ••• (••••) •••–•••

Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

Group Importance Sampling for particle filtering and MCMC

Luca Martino [a,b,*], Víctor Elvira [c], Gustau Camps-Valls [a]

[a] Image Processing Laboratory, Universitat de València, València, Spain
[b] Dep. of Statistical Signal Processing, Universidad Carlos III de Madrid, Madrid, Spain
[c] IMT Lille Douai CRISTAL (UMR 9189), Villeneuve d'Ascq, France

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Importance Sampling
Markov Chain Monte Carlo (MCMC)
Particle filtering
Particle Metropolis–Hastings
Multiple Try Metropolis
Bayesian inference

ABSTRACT

Bayesian methods and their implementations by means of sophisticated Monte Carlo techniques have become very popular in signal processing over the last years. Importance Sampling (IS) is a well-known Monte Carlo technique that approximates integrals involving a posterior distribution by means of weighted samples. In this work, we study the assignation of a single weighted sample which compresses the information contained in a population of weighted samples. Part of the theory that we present as Group Importance Sampling (GIS) has been employed implicitly in different works in the literature. The provided analysis yields several theoretical and practical consequences. For instance, we discuss the application of GIS into the Sequential Importance Resampling framework and show that Independent Multiple Try Metropolis schemes can be interpreted as a standard Metropolis–Hastings algorithm, following the GIS approach. We also introduce two novel Markov Chain Monte Carlo (MCMC) techniques based on GIS. The first one, named Group Metropolis Sampling method, produces a Markov chain of sets of weighted samples. All these sets are then employed for obtaining a unique global estimator. The second one is the Distributed Particle Metropolis–Hastings technique, where different parallel particle filters are jointly used to drive an MCMC algorithm. Different resampled trajectories are compared and then tested with a proper acceptance probability. The novel schemes are tested in different numerical experiments such as learning the hyperparameters of Gaussian Processes, two localization problems in a wireless sensor network (with synthetic and real data) and the tracking of vegetation parameters given satellite observations, where they are compared with several benchmark Monte Carlo techniques. Three illustrative Matlab demos are also provided.

1. Introduction

Bayesian signal processing, which has become very popular over the last years in statistical signal processing, requires the study of complicated distributions of variables of interested conditioned on observed data [1,2]. Unfortunately, the computation of statistical features related to these posterior distributions (such as moments or credible intervals) is analytically impossible in many real-world applications. Monte Carlo methods are state-of-the-art tools for approximating complicated integrals involving sophisticated multidimensional densities [3,1,2]. The most popular classes of MC methods are the Importance Sampling (IS) techniques and the Markov chain Monte Carlo (MCMC) algorithms [3,2]. IS schemes produce a random discrete approximation of the posterior distribution by a population of weighted samples [4,5,1,2]. 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 [3,1]. Both families are widely used in the signal processing community. Several exhaustive overviews regarding the application of Monte Carlo methods in statistical signal processing, communications and machine learning can be found in the literature: some of them specifically focused on MCMC algorithms [6–9], others specifically focused on IS techniques (and related methods) [10,4,11,12] or with a broader view [13–17].

In this work, we introduce theory and practice of a novel approach, called Group Importance Sampling (GIS), where the information contained in different sets of weighted samples is compressed by using only one, yet properly selected, particle, and one suitable weight.[1] This general idea supports the validity of differ-

* Corresponding author at: Image Processing Laboratory, Universitat de València, València, Spain.
E-mail address: lmartino@ing.uc3m.es (L. Martino).

https://doi.org/10.1016/j.dsp.2018.07.007

[1] A preliminary version of this work has been published in [18]. With respect to that paper, here we provide a complete theoretical support of the Group Importance Sampling (GIS) approach (and of the derived methods), given in the main body of

Please cite this article in press as: L. Martino et al., Group Importance Sampling for particle filtering and MCMC, Digit. Signal Process. (2018), https://doi.org/10.1016/j.dsp.2018.07.007

Table 1
Main notation of the work.

$\mathbf{x} = [x_1, \ldots, x_D]$	Variable of interest, $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^{D\times\xi}$, with $x_d \in \mathbb{R}^{\xi}$ for all d
$\bar{\pi}(\mathbf{x})$	Normalized posterior pdf, $\bar{\pi}(\mathbf{x}) = p(\mathbf{x}\|\mathbf{y})$
$\pi(\mathbf{x})$	Unnormalized posterior function, $\pi(\mathbf{x}) \propto \bar{\pi}(\mathbf{x})$
$\widehat{\pi}(\mathbf{x}\|\mathbf{x}_{1:N})$	Particle approximation of $\bar{\pi}(\mathbf{x})$ using the set of samples $\mathbf{x}_{1:N} = \{\mathbf{x}_n\}_{n=1}^N$
$\widetilde{\mathbf{x}}$	Resampled particle, $\widetilde{\mathbf{x}} \sim \widehat{\pi}(\mathbf{x}\|\mathbf{x}_{1:N})$ (note that $\widetilde{\mathbf{x}} \in \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$)
$w_n = w(\mathbf{x}_n)$	Unnormalized standard IS weight of the particle $\mathbf{x}_n$
$\bar{w}_n = \bar{w}(\mathbf{x}_n)$	Normalized weight associated to $\mathbf{x}_n$
$\widetilde{w}_m = \widetilde{w}(\widetilde{\mathbf{x}}_m)$	Unnormalized proper weight associated to the resampled particle $\widetilde{\mathbf{x}}_m$
W_m	Summary weight of m-th set $\mathcal{S}_m$
$\widehat{I}_N$	Standard self-normalized IS estimator using N samples
$\widetilde{I}_N$	Self-normalized estimator using N samples and based on GIS theory
Z	Marginal likelihood; normalizing constant of $\pi(\mathbf{x})$
$\widehat{Z}, \overline{Z}$	Estimators of the marginal likelihood Z

ent Monte Carlo algorithms in the literature: interacting parallel particle filters [19–21], particle island schemes and related techniques [22–24], particle filters for model selection [25–27], nested Sequential Monte Carlo (SMC) methods [28–30] are some examples. We point out some consequences of the application of GIS in Sequential Importance Resampling (SIR) schemes, allowing partial resampling procedures and the use of different marginal likelihood estimators. Then, we show that the Independent Multiple Try Metropolis (I-MTM) techniques and the Particle Metropolis–Hastings (PMH) algorithm can be interpreted as a classical Independent Metropolis–Hastings method by the application of GIS.

Furthermore, we present two novel techniques based on GIS. The first one is the *Group Metropolis Sampling* (GMS) algorithm that generates a Markov chain of sets of weighted samples. All these resulting sets of samples are jointly exploited to obtain a unique particle approximation of the target distribution. On the one hand, GMS can be considered an MCMC method since it produces a Markov chain of sets of samples. On the other hand, the GMS can be also considered as an iterated importance sampler where different estimators are finally combined in order to build a unique IS estimator. This combination is obtained *dynamically* through random repetitions given by MCMC-type acceptance tests. GMS is closely related to Multiple Try Metropolis (MTM) techniques and Particle Metropolis–Hastings (PMH) algorithms [31–36], as we discuss below. The GMS algorithm can be also seen as an extension of the method in [37], for recycling auxiliary samples in a MCMC method.

The second novel algorithm based on GIS is the Distributed PMH (DPMH) technique where the outputs of several parallel particle filters are compared by an MH-type acceptance function. The proper design of DPMH is a direct application of GIS. The benefit of DPMH is twofold: different type of particle filters (for instance, with different proposal densities) can be jointly employed, and the computational effort can be distributed in several machines speeding up the resulting algorithm. As the standard PMH method, DPMH is useful for filtering and smoothing the estimation of the trajectory of a variable of interest in a state–space model. Furthermore, the marginal version of DPMH can be used for the joint estimation of dynamic and static parameters. When the approximation of only one specific moment of the posterior is required, like GMS, the DPMH output can be expressed as a chain of IS estimators. The novel schemes are tested in different numerical experiments: hyperparameter tuning for Gaussian Processes, two localization problems in a wireless sensor network (one with real data), and finally a filtering problem of Leaf Area Index (LAI), which is a parameter widely used to monitor vegetation from satellite observations. The comparisons with other benchmark Monte Carlo methods show the benefits of the proposed algorithms.[2]

The remainder of the paper has the following structure. Section 2 recalls some background material. The basis of the GIS theory is introduced in Section 3. The applications of GIS in particle filtering and Multiple Try Metropolis algorithms are discussed in Section 4. In Section 5, we introduce the novel techniques based on GIS. Section 6.1 provides the numerical results and in Section 7 we discuss some conclusions.

2. Problem statement and background

In many applications, the goal is to infer a variable of interest, $\mathbf{x} = x_{1:D} = [x_1, x_2, \ldots, x_D] \in \mathcal{X} \subseteq \mathbb{R}^{D\times\xi}$, where $x_d \in \mathbb{R}^{\xi}$ for all $d = 1, \ldots, D$, given a set of related observations or measurements, $\mathbf{y} \in \mathbb{R}^{d_Y}$. In the Bayesian framework all the statistical information is summarized by the posterior probability density function (pdf), i.e.,

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

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, $Z \equiv Z(\mathbf{y})$ is unknown and difficult to estimate in general, so we assume to be able to evaluate the unnormalized target function,

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

The computation of integrals involving $\bar{\pi}(\mathbf{x}) = \frac{1}{Z}\pi(\mathbf{x})$ is often intractable. We consider the Monte Carlo approximation of complicated integrals involving the target $\bar{\pi}(\mathbf{x})$ and an integrable function $h(\mathbf{x})$ with respect to $\bar{\pi}$, i.e.,

$$I = E_{\bar{\pi}}[h(\mathbf{X})] = \int_{\mathcal{X}} h(\mathbf{x})\bar{\pi}(\mathbf{x})d\mathbf{x}, \quad (3)$$

where we denote $\mathbf{X} \sim \bar{\pi}(\mathbf{x})$. The basic Monte Carlo (MC) procedure consists in drawing N independent samples from the target pdf, i.e., $\mathbf{x}_1, \ldots, \mathbf{x}_N \sim \bar{\pi}(\mathbf{x})$, so that $\widehat{I}_N = \frac{1}{N}\sum_{n=1}^N h(\mathbf{x}_n)$ is an unbiased estimator of I [1,2]. However, in general, direct methods for drawing samples from $\bar{\pi}(\mathbf{x})$ do not exist so that alternative procedures are required. Below, we describe the most popular approaches. Table 1 summarizes the main notation of the work. Note that the

the text (Sections 3 and 4) and in five additional appendices. Moreover, we provide an additional method based on GIS in Section 5.2 and a discussion regarding particle Metropolis schemes and the standard Metropolis–Hastings method in Section 4.2. We also provide several additional numerical studies, one considering real data. Related Matlab software is also given at https://github.com/lukafree/GIS.git.

[2] Three illustrative Matlab demos are also provided at https://github.com/lukafree/GIS.git.