[Arnaud Doucet and Xiaodong Wang]

Monte Carlo Methods for Signal Processing

[A review in the statistical signal processing context]

© INFINITY 8

In many problems encountered in signal processing, it is possible to accurately describe the underlying statistical model using probability distributions. Statistical inference can then theoretically be performed based on the relevant likelihood function or posterior distribution in a Bayesian framework. However, most problems encountered in applied research require non-Gaussian and/or nonlinear models to correctly account for the observed data. In these cases, it is typically impossible to obtain the required statistical estimates of interest [e.g., maximum likelihood (ML) or conditional expectation] in closed form as it requires integration and/or maximization of complex multidimensional functions. A standard approach consists of making model simplifications or crude analytic approximations to obtain algorithms that can be easily implemented. With the recent availability of high-powered computers, numerical-simulation-based

1053-5888/05/$20.00©2005IEEE

approaches can now be considered and the full complexity of real problems can be addressed.

These integration and/or optimization problems can be tackled using analytic approximation techniques or deterministic numerical integration/optimization methods. These classical methods are often not sufficiently precise or robust, or they are too complex to implement. Monte Carlo algorithms are an attractive alternative. These algorithms are remarkably flexible and extremely powerful. The basic idea is to draw a large number of samples distributed according to some probability distribution(s) of interest so as to obtain consistent simulation-based estimates. These methods first became popular in physics [39] before literally revolutionizing applied statistics and related fields such as bioinformatics and econometrics in the 1990s [8], [27], [36], [44].

THESE ALGORITHMS ARE REMARKABLY FLEXIBLE AND EXTREMELY POWERFUL.

Despite their ability to allow statistical inference to be performed for highly complex models, these flexible and powerful methods are not yet well known in signal processing. This tutorial attempts to provide a simple yet complete review of these methods in a signal processing context. We describe generic Monte Carlo methods that can be used to perform statistical inference in both batch and sequential contexts and demonstrate their efficiency on some complex nonlinear, non-Gaussian signal processing problems.

MOTIVATION

MODEL-BASED SIGNAL PROCESSING

In statistical signal processing, many problems can be formulated as follows. One is interested in obtaining an estimate of an unobserved random variable X taking values in $\mathcal{X}$ given the realization of some statistically related observations $Y = y$. In a model-based context, one has access to the likelihood function giving the probability or probability density function (pdf) $p(y|x)$ of $Y = y$, given $X = x$. In this case, a standard point estimate of X is given by the ML estimate

$$x_{ML} = \arg\max_{x \in \mathcal{X}} \; p(y|x).$$

For simple models, it is possible to compute $p(y|x)$ in closed form, and the maximization of the pdf can be performed easily. However, when the model includes latent variables, or some non-Gaussian and/or nonlinear elements, it is often impossible to compute the likelihood in closed form. It is also difficult to maximize it as it is a multimodal and potentially high-dimensional function. This severely limits the applications of ML approaches for complex models.

The problem appears even more clearly when one is interested in performing Bayesian inference [7], [43]. In this context, one sets a prior distribution on X, say $p(x)$, and all (Bayesian) inference relies on the posterior distribution given by Bayes' theorem

$$p(x|y) = \frac{p(y|x)\,p(x)}{p(y)},$$

where

$$\int p(y|x)\,p(x)\,dx = p(y).$$

For example, the minimum mean-square error (MMSE) estimate of X given $Y = y$ is defined by

$$x_{\mathrm{MMSE}} = \int x p(x|y)\,dx.$$

To compute this estimate, it is necessary to compute two integrals. It is only feasible to perform these calculations analytically for simple statistical models.

EXAMPLES

To illustrate these problems, we will discuss a few standard signal processing applications. For the sake of simplicity, we do not distinguish random variables and their realizations from now on. We will use the notation $z_{i:j} = \left(z_i, z_{i+1}, \ldots, z_j\right)^{\mathrm{T}}$ for any sequence $\{z_n\}$.

BLIND EQUALIZATION

Consider a stream of independent binary symbols $b_{2-L:T}$ $(b_k = \pm 1)$ going through a finite impulse response channel $h = (h_0, \ldots, h_{L-1})^{\mathrm{T}}$ and observed in a Gaussian noise of variance σ^2; i.e.,

$$y_n = \sum_{k=0}^{L-1} h_k b_{n-k} + v_k = h^{\mathrm{T}} b_{n-L+1:n} + v_n, \qquad (1)$$

where $v_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$. Given a set of observations $y_{1:T}$, we are interested in estimating the sequence of unknown bits $b_{2-L:T}$. When h and σ^2 are known, the likelihood function $p(y_{1:T} \mid b_{2-L:T}, h, \sigma^2)$ is given by

$$p(y_{1:T}|b_{2-L:T}, h, \sigma^2) \propto \exp\left(-\frac{1}{2\sigma^2}\sum_{n=1}^{T}(y_n - h^{\mathrm{T}} b_{n-L+1:n})^2\right).$$

It is well known that it is possible in this case to maximize the likelihood $p(y_{1:T} \mid b_{2-L:T}, h, \sigma^2)$ with respect to $b_{2-L:T}$ using the Viterbi algorithm. When the parameters (h, σ^2) are unknown, it is not recommended to estimate $(b_{2-L:T}, h, \sigma^2)$ by maximizing their joint likelihood to obtain the joint maximum likelihood (JML) estimate $(b_{2-L:T,\mathrm{JML}}, h_{\mathrm{JML}}, \sigma^2_{\mathrm{JML}})$ because, even if $T \to \infty$, it is well known that $(h_{\mathrm{JML}}, \sigma^2_{\mathrm{JML}})$ do not converge towards their true values. A standard strategy involves estimating the parameters (h, σ^2) by maximizing the marginal likelihood

$$p(y_{1:T} \mid h, \sigma^2) = \sum_{b_{2-L:T}} p(y_{1:T} \mid b_{2-L:T}, h, \sigma^2 t).$$

This sum can be computed exactly with an algorithm of computational complexity linear with T using the forward-backward formula [42]. Indeed, this problem can be reformulated as a standard hidden Markov model (HMM). To perform the maximization of the marginal likelihood, one typically relies on the expectation-maximization (EM) algorithm. The EM algorithm is a deterministic, gradient-type algorithm that typically converges towards a stationary point of the likelihood. Once the parameter estimates of (h, σ^2), say (h_{ML}, σ^2_{ML}), are obtained, $b_{2-L:T}$ is estimated by maximizing $p(y_{1:T} \mid b_{2-L:T}, h_{ML}, \sigma^2_{ML})$ using the Viterbi algorithm. As a byproduct of the forward-backward formula, we also obtain the marginal probabilities $p(b_k | y_{1:T}, h_{ML}, \sigma^2_{ML})$. In the context of blind equalization, the EM algorithm was first presented in [35].

Let us now consider an alternative Bayesian approach to this problem. First, assume the parameters (h, σ^2) are known and assume the symbols are independent identically distributed (i.i.d.) with $p(b_k = \pm 1) = \frac{1}{2}$. Maximizing the posterior distribution

$$p(b_{2-L:T} | y_{1:T}, h, \sigma^2) = \frac{p(y_{1:T} | b_{2-L:T}, h, \sigma^2) p(b_{2-L:T})}{p(y_{1:T})}$$

is obviously equivalent to maximizing the likelihood $p(y_{1:T} | b_{2-L:T}, h, \sigma^2)$. Now, in the case where the parameters (h, σ^2) are unknown, we consider a full Bayesian approach where (h, σ^2) are considered random and distributed according to the following prior distribution

$$p(h, \sigma^2) = p(h|\sigma^2) p(\sigma^2),$$

where

$$h | \sigma^2 \sim \mathcal{N}(0, \sigma^2 \delta^2 I_L), \quad \sigma^2 \sim \mathcal{IG}\left(\frac{\gamma_0}{2}, \frac{\upsilon_0}{2}\right). \tag{2}$$

Here $\mathcal{N}$ denotes the Gaussian distribution and $\mathcal{IG}$ denotes the inverse-Gamma distribution. If the parameters are such that $\delta^2 \gg 1$, $\gamma_0 \ll 1$, $\upsilon_0 \ll 1$, then this prior distribution is vague (i.e., noninformative). It admits a conjugate form [7], [43], i.e., both $p(h, \sigma^2)$ and $p(h, \sigma^2 | y_{1:T}, b_{2-L:T})$ are of the same functional normal-inverse Gamma form. In this case, it can be easily shown that

$$\begin{aligned} p(b_{2-L:T} | y_{1:T}) &= \int p(b_{2-L:T}, h, \sigma^2 | y_{1:T}) dh d\sigma^2 \\ &= \int p(b_{2-L:T} | y_{1:T}, h, \sigma^2) p(h, \sigma^2) dh d\sigma^2 \\ &\propto \Big(\upsilon_0 + y_{1:T} y_{1:T}^{\mathrm{T}} - m(b_{2-L:T}) \\ &\qquad \times \Sigma^{-1}(b_{2-L:T}) m^{\mathrm{T}}(b_{2-L:T}) \Big)^{-\left(\frac{T+\gamma_0}{2}\right)}, \end{aligned} \tag{3}$$

where

$$\begin{aligned} \Sigma^{-1}(b_{2-L:T}) &= \sum_{n=1}^{T} b_{n-L+1:n} b^{\mathrm{T}}_{n-L+1:n} + \delta^{-2} I_L, \\ m(b_{2-L:T}) &= \Sigma \left(\sum_{n=1}^{T} y_n b_{n-L+1:n} \right). \end{aligned} \tag{4}$$

This discrete probability distribution cannot typically be computed exactly as there are 2^{T+L-1} potential sequences. Practically, it is necessary to perform some approximations.

In this case, we see that the ML approach requires solving a global optimization problem to maximize the marginal likelihood, whereas the Bayesian approach to compute/maximize exactly $p(b_{2-L:T} \mid y_{1:T})$ or $p(b_k \mid y_{1:T})$ requires computing an exponential sum of terms. It is possible to come up with sensible deterministic algorithms to approximately perform these maximizations/integrations because this statistical model is still relatively simple. However, these strategies can be very sensitive to initialization. Moreover, as soon as the models become more complex, they are typically not applicable.

DECONVOLUTION OF IMPULSIVE SEQUENCES

Let us consider a very similar model where the observations are also given by (1). However, in this case, $b_{2-L:T}$ is not a sequence of binary symbols anymore but is such that

$$b_k \overset{\text{i.i.d.}}{\sim} \lambda \mathcal{N}\left(0, \sigma_b^2\right) + (1 - \lambda)\, \delta_0;$$

i.e., with probability $1 - \lambda$, one has $b_k = 0$ and otherwise it is distributed according to $\mathcal{N}(0, \sigma_b^2)$. This model has numerous applications in geophysics and nuclear science [38]. This simple modification of the nature of b_k makes the problem much more difficult. In most applications, we are interested in determining whether an impulse occurred at time k; i.e., if $b_k \sim \mathcal{N}(0, \sigma_b^2)$ or $b_k = 0$. To achieve this, we introduce the set of i.i.d. latent variables $i_{2-L:T}$ such that i_k takes two arbitrary values, say $\{0, 1\}$, $\Pr(i_k = 1) = 1 - \Pr(i_k = 0) = \lambda$ and

$$b_k | \, i_k = 0 \sim \delta_0, \qquad b_k | \, i_k = 1 \sim \mathcal{N}\left(0, \sigma_b^2\right). \tag{5}$$

Assuming the parameters $\theta = (h, \sigma^2, \lambda, \sigma_b^2)$ are known, it is possible to integrate out analytically the variables $b_{2-L:T}$ to compute $p(i_{2-L:T} \mid y_{1:T}, h, \sigma^2, \lambda, \sigma_b^2)$ pointwise up to a normalizing constant

$$p(i_{2-L:T} \mid y_{1:T}, \theta) \propto p(i_{2-L:T} \mid \theta)\; p(y_{1:T} \mid i_{2-L:T}, \theta),$$

where

$$p(y_{1:T} \mid i_{2-L:T}, \theta) = \int p(y_{1:T} \mid b_{2-L:T}, i_{2-L:T}, \theta) \times p(b_{2-L:T} \mid i_{2-L:T}, \theta)\; p(i_{2-L:T} \mid \theta) db_{2-L:T}.$$

However, this discrete posterior distribution takes 2^{T+L-1} values, and some approximations are necessary to approximate/maximize it. If the parameter θ is unknown, the problem becomes even more complex. The likelihood function

$$p\,(\,y_{1:T} \mid \theta) = \sum_{i_{2-L:T}} p(y_{1:T} \mid i_{2-L:T}, \theta)$$

cannot be computed pointwise because it involves the sum of $2^{T+L-1} \gg 1$ terms. There is no simple EM algorithm that can be implemented in this framework, unlike in the previous example.

One can adopt a full Bayesian approach by setting a conjugate prior distribution on θ,

$$p\Big(h, \sigma^2, \lambda, \sigma_b^2\Big) = p(h \mid \sigma^2)\; p(\sigma^2) p(\lambda) p\Big(\sigma_b^2\Big),$$

where

$$\begin{aligned} h \big| \,\sigma^2 &\sim \mathcal{N}(0, \sigma^2 \delta^2 I_L), \quad \sigma^2 \sim \mathcal{IG}\left(\frac{\gamma_0}{2}, \frac{\upsilon_0}{2}\right), \\ \sigma_b^2 &\sim \mathcal{IG}\left(\frac{\gamma_b}{2}, \frac{\upsilon_b}{2}\right), \quad \lambda \sim \mathcal{B}\,(\zeta, \tau)\,. \end{aligned} \tag{6}$$

Here $\mathcal{B}$ denotes the Beta distribution. This prior is vague for $\delta^2 \gg 1$, $\gamma_0, \upsilon_0, \gamma_b, \upsilon_b \ll 1$ and $\zeta = \tau = 1$. It is possible to compute $p(i_{2-L:T}, b_{2-L:T}, \theta | y_{1:T})$ and $p(i_{2-L:T}, \theta | y_{1:T})$ up to a normalizing constant, but their normalizing constants are untractable. For example, the distribution $p(i_{2-L:T} | y_{1:T}, \theta)$ takes 2^{T+L-1} values, and some approximations are also necessary to approximate/maximize it. If we are interested in computing some estimates of the parameter θ, say its MMSE estimate, the marginal posterior distribution of the parameters $p(\theta | y_{1:T})$ does not admit a closed-form expression given that the likelihood requires computing the sum of 2^{T+L-1} terms.

SPECTRAL ANALYSIS

Consider the problem of estimating some sinusoids in noise. Let $y_{1:T}$ be an observed vector of T real data samples. The elements of $y_{1:T}$ may be represented by different models $\mathcal{M}_k$ corresponding either to samples of noise only $(k = 0)$ or to the superposition of k $(k \geq 1)$ sinusoids corrupted by noise; more precisely,

$$\begin{aligned} \mathcal{M}_0 :\; & y_n = v_{n,k} && k = 0 \\ \mathcal{M}_k :\; & y_n = \sum_{j=1}^{k} (a_{c_{j,k}} \cos[\omega_{j,k} n] + a_{s_{j,k}} \sin[\omega_{j,k} n]) \\ & \quad + v_{n,k} && k \geq 1 \end{aligned}$$

where $\omega_{j_1,k} \neq \omega_{j_2,k}$ for $j_1 \neq j_2$ and $a_{c_{j,k}}, a_{s_{j,k}}, \; \omega_{j,k}$ are, respectively, the amplitudes and the radial frequency of the jth sinusoid for the model with k sinusoids. The noise sequence $v_{1:T,k}$ is assumed zero-mean white Gaussian of variance σ_k^2. In vector-matrix form, we have

$$y_{1:T} = D\,(\omega_k)\; a_k + v_{k,1:T},$$

where $a_k = (a_{c_{1,k}}, a_{s_{1,k}}, \ldots, a_{c_{k,k}}, a_{s_{k,k}})^{\mathrm{T}}$ and $\omega_k = (\omega_{1,k}, \ldots, \omega_{k,k})^{\mathrm{T}}$. The $T \times 2k$ matrix $D\,(\omega_k)$ is defined as

$$\begin{aligned} [D\,(\omega_k)]_{i,2j-1} &= \cos[\omega_{j,k} i], \;\; (i = 1, \ldots, T, \;\; j = 1, \ldots, k) \\ [D\,(\omega_k)]_{i,2j} &= \sin[\omega_{j,k} i], \;\; (i = 1, \ldots, T, \;\; j = 1, \ldots, k). \end{aligned}$$

Here we assume that the number k of sinusoids and their parameters $(a_k, \omega_k, \sigma_k^2)$ are unknown. Given $y_{1:T}$, our objective is to estimate $(k, a_k, \omega_k, \sigma_k^2)$. It is standard in signal processing to perform parameter estimation and model selection using a (penalized) ML approach. First, an approximate ML estimate of the parameters is found; we emphasize that, unfortunately, the likelihood is highly nonlinear in its parameters ω_k and admits typically severe local maxima. Model selection is then performed by maximizing an information criterion (IC) such as Akaike IC (AIC), Bayes IC (BIC), or minimum description length (MDL). Note that when the number of observations is small, these criteria can perform poorly. We follow here a Bayesian approach; see [1] for a motivation of this model. One has

$$\begin{aligned} a_k \big| \,\sigma_k^2 &\sim \mathcal{N}\Big(0, \sigma_k^2 \delta^2 (D^{\mathrm{T}}\,(\omega_k)\, D\,(\omega_k))^{-1}\Big), \\ \sigma_k^2 &\sim \mathcal{IG}\left(\frac{\upsilon_0}{2}, \frac{\gamma_0}{2}\right) \end{aligned} \tag{7}$$

and the frequencies ω_k are independent and uniformly distributed over $(0, \pi)$. Finally, we assume that the prior distribution $p(k)$ is a truncated Poisson distribution of intensity Λ, where $k_{\max} \triangleq \lfloor (N-1)/2 \rfloor$. (This constraint is added because otherwise the columns of $D(\omega_k)$ would be linearly dependent.) The terms δ^2 and Λ can be respectively interpreted as an expected signal-to-noise ratio (SNR) and the expected number of sinusoids.

In this case, it can easily be established that the marginal posterior distribution of the frequencies ω_k is proportional on $\Omega = \{0, 1, \ldots, k_{\max}\} \times (0, \pi)^k$ to

$$p(\omega_k, k\,|y_{1:T}) \propto \left(\gamma_0 + y_{1:T}^{\mathrm{T}} P_k y_{1:T}\right)^{-\frac{T+\upsilon_0}{2}} \frac{(\Lambda/(\delta^2+1)\pi))^k}{k!}, \tag{8}$$

where

$$\begin{aligned} M_k^{-1} &= (1+\delta^{-2}) D^{\mathrm{T}}(\omega_k)\, D(\omega_k)\,, \\ m_k &= M_k D^{\mathrm{T}}(\omega_k)\, y_{1:T}, \\ P_k &= I_T - D(\omega_k)\, M_k D^{\mathrm{T}}(\omega_k)\,. \end{aligned}$$

This posterior distribution is highly nonlinear in the parameters ω_k. Moreover, one cannot compute explicitly its normalizing constant $p(y_{1:T}|k)$, so it is impossible to compute the Bayes factors to perform model selection. Standard numerical integration techniques could be used, but they are typically inefficient when the dimension of the space of interest is high.

OPTIMAL FILTERING IN STATE-SPACE MODELS

Consider an unobserved Markov process $\{x_n\}_{n\geq 1}$ of initial density μ and transition density $x_n|\, x_{n-1} \sim f(\cdot|\, x_{n-1})$. The observations $\{y_n\}_{n\geq 1}$ are conditionally independent given $\{x_n\}_{n\geq 1}$ of marginal density $y_n|\, x_n \sim g(\cdot|\, x_n)$. This class of models is extremely wide. For example, it includes

$$x_n = \varphi(x_{n-1}, v_n), \quad y_n = \Psi(x_n, w_n)\,,$$

where φ and Ψ are two nonlinear deterministic mappings and $\{v_n\}_{n\geq 2}$ and $\{w_n\}_{n\geq 2}$ are two independent and mutually independent sequences.

All inference on $x_{1:n}$ based on $y_{1:n}$ is founded on the posterior distribution

$$p(x_{1:n}|\, y_{1:n}) = \frac{p(y_{1:n}|\, x_{1:n})\, p(x_{1:n})}{\int p(y_{1:n}|\, x_{1:n})\, p(x_{1:n})\, dx_{1:n}},$$

where

$$\begin{aligned} p(x_{1:n}) &= p(x_1) \prod_{k=2}^{n} f(x_k\,|x_{k-1}), \\ p(y_{1:n}|\, x_{1:n}) &= \prod_{k=1}^{n} g(y_k|\, x_k)\,. \end{aligned}$$

This posterior distribution satisfies the following recursion:

$$p(x_{1:n}\,|y_{1:n}) = \frac{f(x_n\,|x_{n-1}) g(y_n\,|x_n)}{p(y_n\,|y_{1:n-1})} p(x_{1:n-1}\,|y_{1:n-1}).$$

Unfortunately, except in the case where $\{x_n\}_{n\geq 1}$ takes values in a finite state-space (HMM techniques) or where the model is linear Gaussian (Kalman filtering techniques), it is impossible to come up with a closed-form expression for this sequence of posterior distributions. Many suboptimal methods have been proposed to approximate this sequence; e.g., extended Kalman filter and Gaussian sum approximations. However, these methods tend to be unreliable as soon as the model includes strong nonlinear and/or non-Gaussian models. Deterministic numerical integration techniques have been proposed, but they are complex to implement, inflexible, and realistically can only be applied to models where $\{x_n\}_{n\geq 1}$ takes values in $\mathbb{R}$ or $\mathbb{R}^2$. (Note that all problems described above require computing and/or maximizing high-dimensional probability distributions. It is possible to come up with deterministic techniques to approximate these distributions. However, as soon as the problems get very complex, the performance of these methods typically deteriorates quickly. In this tutorial, we advocate that Monte Carlo methods are a powerful set of techniques that can provide satisfactory answers to all these problems.)

BASICS OF MONTE CARLO METHODS

GENERIC PROBLEMS

Let us consider the pdf $\pi(x)$ where $x \in \mathcal{X}$. We will assume from now on that $\pi(x)$ is known pointwise up to a normalizing constant, i.e.,

$$\pi(x) = Z^{-1} \widetilde{\pi}(x)\,,$$

where $\widetilde{\pi}(x)$ is known pointwise but the normalizing constant

$$Z = \int_{\mathcal{X}} \widetilde{\pi}(x)\, dx$$

is unknown. Note that this assumption is satisfied in all the examples discussed in the previous section if x corresponds to all the unknown variables/parameters.

In most applications of interest, the space $\mathcal{X}$ is typically high dimensional; say $\mathcal{X} = \mathbb{R}^{1000}$ or $\mathcal{X} = \{0, 1\}^{1000}$. We are interested in the following generic problems.

- *Computing integrals.* For any test function $\varphi : \mathcal{X} \to \mathbb{R}$, we want to compute

$$E_\pi(\varphi) = \int_{\mathcal{X}} \varphi(x)\, \pi(x)\, dx. \tag{9}$$

- *Marginal distributions.* Assume $x = (x_1, x_2) \in \mathcal{X}_1 \times \mathcal{X}_2$. Then we want to compute the marginal distribution

$$\pi(x_1) = \int_{\mathcal{X}_2} \pi(x_1, x_2)\, dx_2. \tag{10}$$

- *Optimization.* Given $\pi(x)$, we are interested in finding

$$\arg\max_{x\in\mathcal{X}} \pi(x) = \arg\max_{x\in\mathcal{X}} \widetilde{\pi}(x)\,. \tag{11}$$

- *Integration/Optimization.* Given the marginal distribution (10), we want to compute

$$\arg\max_{x_1 \in \mathcal{X}_1} \pi\left(x_1\right) = \arg\max_{x_1 \in \mathcal{X}_1} \widetilde{\pi}\left(x_1\right). \tag{12}$$

MONTE CARLO METHODS

Assume it is possible to obtain a large number of N independent random samples $\{x^{(i)}\}$ $(i = 1, \ldots, N)$ distributed according to π. The Monte Carlo method approximates π by the following point-mass measure:

$$\widehat{\pi}\left(x\right) = \frac{1}{N}\sum_{i=1}^{N}\delta\left(x - x^{(i)}\right). \tag{13}$$

It follows that an estimate of (9) is given by

$$\widehat{E}_{\pi}\left(\varphi\right) = \int_{\mathcal{X}}\varphi\left(x\right)\widehat{\pi}\left(x\right)dx = \frac{1}{N}\sum_{i=1}^{N}\varphi\left(x^{(i)}\right). \tag{14}$$

Marginal distributions can also be estimated in a straightforward manner as

$$\begin{aligned}\widehat{\pi}\left(x_1\right) &= \int_{\mathcal{X}_2}\widehat{\pi}\left(x_1, x_2\right)dx_2 \\ &= \int_{\mathcal{X}_2}\frac{1}{N}\sum_{i=1}^{N}\delta\left(x_1 - x_1^{(i)}, x_2 - x_2^{(i)}\right)dx_2 \\ &= \frac{1}{N}\sum_{i=1}^{N}\delta\left(x_1 - x_1^{(i)}\right). \end{aligned} \tag{15}$$

Since the samples $\{x^{(i)}\}$ are distributed according to π, a significant proportion of them will be in the vicinity of the mode. Thus, a reasonable estimate of (11) is

$$\arg\max_{\{x^{(i)}\}}\widetilde{\pi}\left(x^{(i)}\right). \tag{16}$$

Optimizing marginal distribution is more difficult; one cannot use $\arg\max_{\{x_1^{(i)}\}}\pi(x_1^{(i)})$ since the marginal distribution cannot be computed even up to a normalizing constant. In the scenario where $\pi(x_1|x_2)$ is known analytically, an alternative to (15) is

$$\begin{aligned}\widehat{\pi}\left(x_1\right) &= \int_{\mathcal{X}_2}\pi\left(x_1|\,x_2\right)\widehat{\pi}\left(x_2\right)dx_2 \\ &= \int_{\mathcal{X}_2}\pi\left(x_1|\,x_2\right)\left(\frac{1}{N}\sum_{i=1}^{N}\delta\left(x_2 - x_2^{(i)}\right)\right)dx_2 \\ &= \frac{1}{N}\sum_{i=1}^{N}\pi\left(x_1|\,x_2^{(i)}\right). \end{aligned} \tag{17}$$

It is then possible to estimate (12) by $\arg\max_{\{x_1^{(i)}\}}\widehat{\pi}(x_1^{(i)})$. Note that the computational complexity of this algorithm is unfortunately very expensive since evaluating (17) pointwise involves $N \gg 1$ terms. Alternative techniques will be discussed later.

A natural question to ask is why the Monte Carlo method is attractive. The typical answer is that, if one considers (14), this estimate has good properties; i.e., it is clearly unbiased and one can easily show that its variance satisfies

$$\operatorname{var}\{\widehat{E}_{\pi}\left(\varphi\right)\} = \frac{\int\varphi^{2}\left(x\right)\pi\left(x\right)dx - E_{\pi}^{2}\left(\varphi\right)}{N}. \tag{18}$$

The truly remarkable property of this estimate is that the rate of convergence to zero of its variance is independent of the space $\mathcal{X}$ (i.e., it can be $\mathbb{R}$ or $\mathrm{R}^{10,000}$), whereas all deterministic integration methods have a rate of convergence of the approximation error that decreases severely as the dimension of the space increases. Note, however, that this does not imply that Monte Carlo methods will always outperform deterministic methods, as the numerator of (18) can be huge. However, Monte Carlo methods tend to be much more flexible and powerful.

Nevertheless, these methods rely on the assumption that we are able to simulate samples $\{x^{(i)}\}$ from π. The next question to address is how we obtain such samples.

CLASSICAL MONTE CARLO METHODS

We will only briefly discuss the topic of classical Monte Carlo methods. It is possible to generate samples from most standard distributions (e.g., uniform, Gaussian, Gamma, Poisson, etc.) using standard techniques. Most of these techniques are based on the inverse cumulative distribution function (cdf) transform and the acceptance-rejection method; see [17] for a thorough treatment of the subject. Note that the inverse cdf method is only applicable to simple probability distributions that are known exactly rather than only up to a normalizing constant. The acceptance-rejection method does not require knowledge of the normalizing constant. However, as discussed below, it is inefficient for high-dimensional distributions.

ACCEPTANCE-REJECTION METHOD

We are interested in sampling from $\pi(x) \propto \widetilde{\pi}(x)$. Assume we are able to sample from another pdf $q(x) \propto \widetilde{q}(x)$ such that $\widetilde{\pi}(x) \leq C\widetilde{q}(x)$ for any $x \in \mathcal{X}$. Then the following procedure allows us to sample from π.

ACCEPTANCE-REJECTION SAMPLING

1) Sample a candidate $x^{*} \sim q\left(\cdot\right)$.
2) Sample $u \sim U\left[0, 1\right]$ (uniform distribution on $[0, 1]$).
3) If $u \leq \widetilde{(}\pi(x^{*}))/(C\widetilde{q}(x^{*}))$, then return x^{*}; otherwise go back to step 1.

The accepted candidate x^{*} is distributed according to π. Indeed, one has

$$
\begin{aligned}
\Pr(x^* \le x, x^*\text{accepted}) &= \int_{u\le x} \frac{\widetilde{\pi}\,(u)}{C\widetilde{q}\,(u)} q\,(u)\,du \\
&= \frac{C^{-1}}{\int_{\mathcal{X}} \widetilde{q}\,(u)\,du} \int_{u\le x} \widetilde{\pi}\,(u)\,du \\
\Pr(x^* \text{ accepted}) &= \int_{\mathcal{X}} \frac{\widetilde{\pi}\,(u)}{C\widetilde{q}\,(u)} q\,(u)\,du \\
&= C^{-1} \frac{\int_{\mathcal{X}} \widetilde{\pi}\,(u)\,du}{\int_{\mathcal{X}} \widetilde{q}\,(u)\,du}.
\end{aligned}
$$

Thus

$$
\begin{aligned}
\Pr(x^* \le x \,|x^*\text{accepted}) &= \frac{\Pr(x^* \le x, x^*\text{accepted})}{\Pr(\mathrm{x^*accepted})} \\
&= \frac{\int_{u\le x} \widetilde{\pi}\,(u)\,du}{\int_{\mathcal{X}} \widetilde{\pi}\,(u)\,du} = \int_{u\le x} \pi\,(u)\,du;
\end{aligned}
$$

i.e., if x^* is accepted, then it is distributed according to π.

Example: Assume we are interested in sampling from the posterior distribution $p(x|y) \propto p(x)p(y|x)$. If it is easy to sample from the prior distribution $p(x)$ (whose normalizing constant $p(y)$ is typically known) and, if the likelihood is upper bounded over $\mathcal{X}$, i.e., say $p(y|x) < M$, then it is possible to use the acceptance-rejection algorithm. Just set $\pi(x) = p(x|y)$, $\widetilde{\pi}(x) = p(x)p(y|x)$, $q(x) = \widetilde{q}(x) = p(x)$, and $\widetilde{\pi}(x) \le M\widetilde{q}(x)$, i.e., $C = M$. Note that this method requires being able to find an upper bound on the likelihood, which can be very difficult.

There are two problems with this approach.

- It can be difficult to find a pdf $q\,(\cdot)$ that is easy to sample such that $\widetilde{\pi}\,(x) \le C\widetilde{q}\,(x)$ for any $x \in \mathcal{X}$.
- The acceptance probability of the candidate is typically extremely small if $\mathcal{X}$ is a high-dimensional space.

IMPORTANCE SAMPLING

A simple alternative to the acceptance-rejection method is importance sampling. In this framework, one also introduces a probability distribution $q\,(x) \propto \widetilde{q}\,(x)$ to sample candidates; this probability distribution is called importance distribution. However, instead of accepting the candidates with a given probability, all the candidates are accepted but are weighted to correct for the discrepancy between $q\,(\cdot)$ and $\pi\,(\cdot)$.

Indeed, assuming that $\pi\,(x) > 0 \Rightarrow q\,(x) > 0$, then the following identities hold trivially

$$
\begin{aligned}
\pi\,(x) &= w\,(x)\,q\,(x) && (19) \\
&= \frac{w\,(x)\,q\,(x)}{\int w\,(u)\,q\,(u)\,du}, && (20)
\end{aligned}
$$

where $w\,(\cdot)$ is the so-called importance weight given by

$$
w\,(x) = \frac{\pi\,(x)}{q\,(x)}. \tag{21}
$$

This suggests that if N samples $\{x^{(i)}\}$ from $q(\cdot)$ are available, then an approximation of this distribution is given by

$$
\widehat{q}\,(x) = \frac{1}{N} \sum_{i=1}^{N} \delta\left(x - x^{(i)}\right). \tag{22}
$$

Plugging this approximation into (19), we obtain

$$
\widehat{\pi}\,(x) = \frac{1}{N} \sum_{i=1}^{N} w\left(x^{(i)}\right) \delta\left(x - x^{(i)}\right). \tag{23}
$$

Plugging it in (20), we obtain the alternative estimate

$$
\widehat{\pi}\,(x) = \sum_{i=1}^{N} w^{(i)} \delta\left(x - x^{(i)}\right), \tag{24}
$$

where the normalized importance weights are given by

$$
w^{(i)} = \frac{w\,(x^{(i)})}{\sum_{j=1}^{N} w\,(x^{(j)})}. \tag{25}
$$

It now follows clearly that if we are interested in $E_{\pi}\,(\varphi)$, then two possible estimates are available. If we use (23), then

$$
\widehat{E}_{\pi}\,(\varphi) = \frac{1}{N} \sum_{i=1}^{N} w\left(x^{(i)}\right) \varphi\left(x^{(i)}\right), \tag{26}
$$

and if we use (24), then

$$
\widehat{E}_{\pi}\,(\varphi) = \sum_{i=1}^{N} w^{(i)} \varphi\left(x^{(i)}\right). \tag{27}
$$

One can check that the estimate (26) is unbiased and converges asymptotically ($N \to \infty$) towards the true value. Unfortunately, it cannot be used in most applications as computing $\{w(x^{(i)})\}$ requires knowing the normalizing constant of π (and q) and this normalizing constant is unknown. In this case, it is possible to use (27) since computing $\{w^{(i)}\}$ does not require knowledge of this normalizing constant. The estimate (27) is biased. However, this bias is of the order $\mathcal{O}\,(1/N)$ and this estimate is also asymptotically consistent; see [25] or [44] for a detailed introduction to importance sampling and its properties.

Example: Assume we are interested in approximating the posterior distribution $p\,(x\,|\,y) \propto p\,(x)\,p\,(y\,|\,x)$. Typically, the normalizing constant $p\,(y)$ is unknown, but it is easy to sample from the prior distribution $p\,(x)$. With the notation given above, this corresponds to $\pi\,(x) = p\,(x\,|\,y)$ and $q\,(x) = p\,(x)$. As the normalizing constant of $p\,(x\,|\,y)$ is typically unknown, one cannot use the importance sampling estimates (23) and (26), but rather only (24) and (27). Given N samples from $p\,(x)$, one obtains

$$\widehat{p}(x\,|\,y) = \sum_{i=1}^{N} w^{(i)} \delta\left(x - x^{(i)}\right),$$

$$\widehat{E}_{p(x\,|y)}(\varphi) = \sum_{i=1}^{N} w^{(i)} \varphi\left(x^{(i)}\right),$$

where

$$w^{(i)} = \frac{p\left(x^{(i)}\,\middle|\,y\right)/p\left(x^{(i)}\right)}{\sum_{j=1}^{N} p\left(x^{(j)}\,\middle|\,y\right)/p\left(x^{(j)}\right)} = \frac{p\left(y\middle|\,x^{(i)}\right)}{\sum_{j=1}^{N} p\left(y\middle|\,x^{(j)}\right)}.$$

This method is very simple to implement.

At first glance, it seems that the importance sampling method does not suffer from the problems of acceptance rejection. However, in practice, one can only expect to obtain estimates with a reasonable variance if the unnormalized weights (21) are upper bounded, i.e., $w(x) \leq C$ for all $x \in \mathcal{X}$. A similar condition exists for acceptance rejection. Moreover, even if this condition is satisfied, the method is typically inefficient if the dimension of $\mathcal{X}$ is large. Thus, it is necessary to develop alternative methods.

MARKOV CHAIN MONTE CARLO METHODS

BASIC IDEA

Markov chain Monte Carlo (MCMC) methods have been introduced in physics in the 1950s by Metropolis et al. [39]. The key idea they introduced is to sample from the target distribution of interest $\pi(x)$ using a Markov chain. More precisely, let us consider an $\mathcal{X}$-valued Markov chain $\{x^{(n)}\}_{n\geq 1}$ of initial distribution μ_1 and transition kernel $K(\cdot|\cdot)$, i.e.,

$$x^{(1)} \sim \mu_1, \quad x^{(n)}\Big| x^{(n-1)} \sim K\left(\cdot|\, x^{(n-1)}\right) \text{ for } n \geq 2.$$

Let us denote μ_n, the marginal distribution of x_n given by

$$\mu_n\left(x^{(n)}\right) = \int \mu_1\left(x^{(1)}\right) \prod_{k=2}^{n} K\left(x^{(k)}\Big|\, x^{(k-1)}\right) dx^{(1:n-1)}.$$

Under regularity conditions on the transition kernel, the sequence of probability distribution $\{\mu_n\}$ converges towards a limiting distribution denoted by π.

Example: Consider the real-valued Markov chain defined by $x^{(1)} \sim \mu_1$ and

$$x^{(n)} = \alpha x^{(n-1)} + v^{(n)},$$

where $|\alpha| < 1$ and $v^{(n)} \sim \mathcal{N}(0, \sigma^2)$. In this case, $K(x^{(n)}|x^{(n-1)}) = \mathcal{N}(x^{(n)}; \alpha x^{(n-1)}, \sigma^2)$ is a Gaussian distribution of argument $x^{(n)}$, mean $\alpha x^{(n-1)}$, and variance σ^2. Let us introduce $\nu(x) = \mathcal{N}(x; 0, (\sigma^2)/(1-\alpha^2))$. It can be shown easily that if $x \sim \nu$, then $x'\big|\, x \sim K(\cdot|\, x)$ is such that $x' \sim \nu$ as well, i.e.,

$$\nu(x') = \int \nu(x)\, K(x'|x)dx.$$

This means that ν is a so-called invariant distribution. It can be further proved that $\mu_n \to \pi = v$. What are the practical consequences of this fact? Assume we are interested in obtaining random samples distributed according to π. Then it is possible to sample a realization of the Markov chain of transition kernel K. Then, whatever being the initialization μ_1, one will have $\mu_n \to \pi$, i.e., for n large enough, the samples $\{x^{(n)}\}$ are approximately distributed according to π. Obviously, in this case the method is useless since one can sample from π exactly. However, we will see further that the same methodology can be used even if π is extremely complex.

We have just seen that, under certain regularity assumptions, a Markov chain of transition kernel K admits a limiting distribution π, i.e., the samples $\{x^{(n)}\}$ are approximately distributed according to π for large n. So we can use a realization of this Markov chain to obtain (approximate) samples from π. In the example described above, we were given the kernel K and then we identified π. Obviously, in practice, we are given π and the question becomes: is it possible to come up with a Markov transition kernel K admitting π as limiting distribution? The answer is yes. There is actually an infinity of such kernels, and the first algorithm we present was created by Metropolis et al. [39] and later refined by Hastings [34].

METROPOLIS-HASTINGS ALGORITHM AND GIBBS SAMPLER

Let us introduce a candidate kernel distribution $q(\cdot|\cdot)$. The Metropolis-Hastings (MH) kernel is given by

$$K_{MH}(x'\,|x) = \alpha(x'\,|x)q(x'\,|x) + \delta(x'-x) \int (1-\alpha(u\,|x))q(u\,|x)du, \qquad (28)$$

where

$$\alpha(x'|x) = \min\left(1, \frac{\pi(x')q(x\,|x')}{\pi(x)q(x'\,|x)}\right). \qquad (29)$$

As potential choices for the proposal kernel $q(\cdot|\cdot)$, one can use $q(x'|x) = q(x')$ [independent proposal] or $q(x'|x) = q(x'-x)$ [random walk proposal]. It can easily be shown that K_{MH} is $\pi-$reversible, i.e.,

$$\pi(x')K_{MH}(x\,|x') = \pi(x)K_{MH}(x'\,|x),$$

which implies directly that K_{MH} admits π as invariant distribution

$$\pi(x') = \int \pi(x) K_{MH}(x'|x)dx$$

as

$$\int K_{MH}(x|x')dx = 1$$

is a pdf for any x'. Algorithmically, one can simulate a realization of the Markov chain of transition kernel K_{MH} by using the following algorithm.

METROPOLIS-HASTINGS ALGORITHM

```
Initialization.
  ■ Select randomly or deterministically $x^{(0)}$.
Iteration $n$ $(n \geq 1)$.
  ■ Sample a candidate $x^* \sim q\left(\cdot \middle| \, x^{(n-1)}\right)$.
  ■ Sample $u \sim U[0,1]$.
  ■ If $u \leq \alpha\left(x^* \middle| \, x^{(n-1)}\right)$, then set $x^{(n)} = x^*$; otherwise $x^{(n)} = x^{(n-1)}$.
```

This algorithm is remarkable. It is very simple and does not require knowledge of the normalizing constant of π, which only appears through a ratio in (29). Under weak assumptions on $q(\cdot|\cdot)$, it can be shown that for any μ_1, $\mu_n \to \pi$ [44]. An estimate of (9) could be computed using

$$\frac{1}{n}\sum_{i=1}^{n} \varphi\left(x^{(i)}\right). \tag{30}$$

This asymptotically converges towards the right value under weak assumptions despite the fact that samples $\{x^{(i)}\}$ are dependent. However, using the estimate (30) is not very sensible since the first simulated samples have a distribution initially far from π. It thus makes sense to discard the first n_0 iterations corresponding to the so-called burn-in period of the chain and then consider

$$\frac{1}{n-n_0}\sum_{i=n_0+1}^{n} \varphi\left(x^{(i)}\right).$$

Determining n_0 automatically is very difficult since obtaining sharp bounds on the convergence rates of MCMC for realistic models is an extremely difficult problem. Also, despite 50 years of research, convergence assessment of Markov chains is still more an art than a science. In practice, we typically monitor some statistics of the Markov chain to decide whether it has reached its stationary regime; see [44] for further details.

The crucial problem of MCMC is actually not to determine the length of the burn-in, but to engineer a so-called fast mixing kernel, i.e., such that $\{\mu_n\}$ converges quickly towards π. To achieve this goal, several guidelines must be followed. It is recommended to update the components of $x = (x_1, \ldots, x_P)$ by subblocks using a mixture of random walks and independent proposals. Indeed, if x belongs to a very high-dimensional space, candidates x^* will only be accepted with a very small probability; the resulting Markov chain mixes too slowly and does not explore the distribution π properly given a reasonable number of iterations.

METROPOLIS-HASTINGS ONE-AT-A-TIME ALGORITHM

```
Initialization.
  ■ Select randomly or deterministically $x^{(0)} = (x_1^{(0)}, \ldots, x_P^{(0)})$.
Iteration $n$ $(n \geq 1)$.
  ■ For $i = 1 : P$
      ■ Sample $x_i^* \sim q_i(\cdot|(x_{1:i-1}^{(n)}, x_{i:P}^{(n-1)}))$.
      ■ Sample $u \sim U[0,1]$.
      ■ If $u \leq \alpha_i((x_{1:i-1}^{(n)}, x_i^*, x_{i+1:P}^{(n-1)})|(x_{1:i-1}^{(n)}, x_i^{(n-1)}, x_{i+1:P}^{(n-1)}))$
        then set $x_i^{(n)} = x_i^*$; otherwise $x_i^{(n)} = x_i^{(n-1)}$.
  End For
```

In the above algorithm, one has

$$\alpha_i\left(\left(x_{1:i-1}^{(n)}, x_i^*, x_{i+1:P}^{(n-1)}\right)\middle|\left(x_{1:i-1}^{(n)}, x_i^{(n-1)}, x_{i+1:P}^{(n-1)}\right)\right) = \min \left(1, \frac{\pi\left(x_{1:i-1}^{(n)}, x_i^*, x_{i+1:P}^{(n-1)}\right) q\left(x_i^{(n-1)} \middle| x_{1:i-1}^{(n)}, x_i^*, x_{i+1:P}^{(n-1)}\right)}{\pi\left(x_{1:i-1}^{(n)}, x_i^{(n-1)}, x_{i+1:P}^{(n-1)}\right) q\left(x_i^* \middle| x_{1:i-1}^{(n)}, x_{i:P}^{(n-1)}\right)}\right). \tag{31}$$

This algorithm simply corresponds to the case where the transition kernel is a composition of P MH kernels; each kernel $K_{\mathrm{MH},i}$ $(i = 1, \ldots, P)$ only uses a proposal kernel updating component x_i. If possible, it is better to include correlated components of the vector x in the same subblock so as to improve the mixing property of the Markov chain.

A particular case of great interest corresponds to the selection of proposal distributions

$$q\left(x_i^* \middle| x_{1:i-1}, x_{i:P}\right) = \pi\left(x_i^* \middle| x_{1:i-1}, x_{i+1:P}\right)$$

referred to as the full conditional distributions. Indeed, even if the joint distribution is only known up to a normalizing constant, it is often possible to express the conditional distributions in closed form. In this case, one can check that

$$\begin{aligned}
&\alpha_i\left(\left(x_{1:i-1}^{(n)}, x_i^*, x_{i+1:P}^{(n-1)}\right)\middle|\left(x_{1:i-1}^{(n)}, x_i^{(n-1)}, x_{i+1:P}^{(n-1)}\right)\right)\\
&= \min\left(1, \frac{\pi\left(x_{1:i-1}^{(n)}, x_i^*, x_{i+1:P}^{(n-1)}\right)\pi\left(x_i^{(n-1)} \middle| x_{1:i-1}^{(n)}, x_{i+1:P}^{(n-1)}\right)}{\pi\left(x_{1:i-1}^{(n)}, x_i^{(n-1)}, x_{i+1:P}^{(n-1)}\right)\pi\left(x_i^* \middle| x_{1:i-1}^{(n)}, x_{i+1:P}^{(n-1)}\right)}\right)\\
&=1;
\end{aligned}$$

that is, all candidates are accepted. The MH one-at-a-time algorithm simplifies to the so-called Gibbs sampler.

```
GIBBS SAMPLER
Initialization.
    ■ Select randomly or deterministically $x^{(0)} = (x_1^{(0)}, \ldots, x_P^{(0)})$.
Iteration $n$ $(n \geq 1)$.
    ■ For $i = 1 : P$
        ■ Sample $x_i^{(n)} \sim \pi(\cdot|(x_{1:i-1}^{(n)}, x_{i+1:P}^{(n-1)}))$.
End For
```

The nice feature of the Gibbs sampler is that it does not require the selection of a proposal distribution; everything is fixed by the target distribution π. However, it is not always possible to sample from the full conditional distributions. Moreover, even if this choice is possible, it might not give the best results.

Many convergence results are available for MCMC based on general state-space Markov chains theory [40]. An introduction to these results and an exhaustive list of references on the subject is given in [44].

REVERSIBLE JUMP MCMC

Consider the case where one is interested in sampling from a distribution π defined on a space that is a union of subspaces, say $\mathcal{X} = \biguplus_{k=0}^{\infty} \{k\} \times \mathcal{X}_k$. We write the distribution as $\pi(k, x_k)$ (i.e., $x_k \in \mathcal{X}_k$). In this case, the associated random variables can take values in subspaces of different dimensions. This is the case when one has to solve inference problems where the number of unknowns is not known. This is, for example, the case in the spectral analysis problem discussed at the beginning of this article. In this case, the number of sinusoids k is unknown and the unknown parameters are defined on a space whose dimension is dependent on k.

For such problems, it is not possible to apply the standard MH algorithm to jump from $\mathcal{X}_k$ to $\mathcal{X}_l$ if $\dim(\mathcal{X}_k) \neq \dim(\mathcal{X}_l)$. A solution proposed by Green [29] is called reversible jump MCMC as it is based on a reversibility constraint on the moves between the different sets $\{\mathcal{X}_k\}$. To jump from $\mathcal{X}_1$ to $\mathcal{X}_2$, one samples an auxiliary variable $u_1 \sim q_{1\to2}(\cdot)$ and sets

$$(x_2, u_2) = \varphi_{1\to2}(x_1, u_1),$$

where $\varphi_{1\to2}$ is a one-to-one deterministic mapping; this implies that the vectors (x_1, u_1) and (x_2, u_2) have the same dimension. Note that if $\dim(\mathcal{X}_1) > \dim(\mathcal{X}_2)$, one typically does not introduce a variable u_1. Similarly, to jump from $\mathcal{X}_2$ to $\mathcal{X}_1$, one samples $u_2 \sim q_{2\to1}(\cdot)$ and sets

$$(x_1, u_1) = \varphi_{2\to1}(x_2, u_2),$$

where $\varphi_{2\to1}(\varphi_{1\to2}(x_1, u_1)) = (x_1, u_1)$.

In this case, the probability of accepting a move from $\mathcal{X}_1$ to $\mathcal{X}_2$ is given by

$$\min\left(1, \frac{\pi(2, x_2)}{\pi(1, x_1)} \frac{p_{2\to1} q_{2\to1}(u_2)}{p_{1\to2} q_{1\to2}(u_1)} \left|\frac{\partial\varphi_{1\to2}(x_1, u_1)}{\partial(x_1, u_1)}\right|\right),$$

where $|(\partial\varphi_{1\to2}(x_1, u_1))/(\partial(x_1, u_1))|$ is the determinant of the Jacobian of the transformation, $p_{i\to j}$ is the probability of choosing to attempt a jump from $\mathcal{X}_i$ to $\mathcal{X}_j$, and $q_{1\to2}(\cdot)$ is the density of u_1. To sum up, the reversible jump algorithm proceeds as follows.

```
Reversible Jump MCMC
Initialization.
    ■ Select randomly or deterministically $(k^{(0)}, x^{(0)})$.
Iteration $n$ $(n \geq 1)$.
    ■ Assume we have $x^{(n-1)} = (k^{(n-1)}, x_{k^{(n-1)}}^{(n-1)})$.
    ■ Propose a move from $\mathcal{X}_{k^{(n-1)}}$ to $\mathcal{X}_l$ with probability $p_{k^{(n-1)}\to l}$.
    ■ Sample $u_{k^{(n-1)}} \sim q_{k^{(n-1)}\to l}(\cdot)$.
    ■ Set $(x_l^*, u_l) = \varphi_{k^{(n-1)}\to l}(x_{k^{(n-1)}}^{(n-1)}, u_{k^{(n-1)}})$.
    ■ With probability
        $\min\left(1, \frac{\pi(l, x_l^*)}{\pi\left(k^{(n-1)}, x_{k^{(n-1)}}^{(n-1)}\right)} \frac{p_{l\to k^{(n-1)}} q_{l\to k^{(n-1)}}(u_l)}{p_{k^{(n-1)}\to l} q_{k^{(n-1)}\to l}(u_{k^{(n-1)}})} \left|\frac{\varphi_{k^{(n-1)}\to l}\left(x_{k^{(n-1)}}^{(n-1)}, u_{k^{(n-1)}}\right)}{\partial\left(x_{k^{(n-1)}}^{(n-1)}, u_{k^{(n-1)}}\right)}\right|\right)$
      set $x^{(n)} = (l, x_l^*)$; otherwise $x^{(n)} = x^{(n-1)}$.
```

PARALLEL TEMPERING

Assume we want to sample from π. In some cases, π admits several distinct modes, and standard MCMC methods might be inefficient in this framework. The idea of parallel tempering is based on the fact that the distribution π_n given by

$$\pi_n(x) = \frac{\pi^{\gamma_n}(x)}{\int \pi^{\gamma_n}(u)\,du} \qquad (32)$$

is more diffuse than π if $\gamma_n < 1$. If a distribution is more diffuse, it is easier to sample from it using an MCMC algorithm.

Parallel tempering consists of building an MCMC algorithm on an extended state-space, say E^P, with a target distribution given by

$$\overline{\pi}(x_1, \ldots, x_P) = \prod_{i=1}^{K} \pi_i(x_i)$$

with $\gamma_1 = 1$ and $\gamma_i < \gamma_{i-1}$ [26]. One can use the standard MCMC algorithm to sample from π_i, where $i = 1, \ldots, K$, and then use swap moves where it is proposed to exchange the states between two different "adjacent" distributions, say π_i and π_{i+1}. Obviously, it is not necessary to propose swap moves at each iteration or only between adjacent distributions. Parallel tempering is a generic method, yet it is pretty robust and efficient. The main practical problem consists of selecting K and the schedule $\{\gamma_i\}$.

SIMULATED ANNEALING FOR GLOBAL OPTIMIZATION

We mentioned previously that it is possible to estimate the global optimum of a distribution π by simply obtaining samples (approximately) distributed according to it and then using (16). Though it is sensible, this method can be somewhat inefficient. In particular, if the distribution is fairly diffuse, then most samples are concentrated away from the global mode(s). The idea behind simulated annealing is to sample at iteration n from a modified target distribution given by (32), where $\{\gamma_n\}_{n\geq 1}$ is an increasing positive sequence such that $\lim_{n\to\infty}\gamma_n = \infty$. The rationale is that, for $\gamma_n > 1$, the distribution (32) is more concentrated around its global maxima than π and eventually the limiting distribution π_∞ concentrates itself on the set of global maxima of π. Indeed, assume a global optimum is given by $x_{\max}$, then for any $x \neq x_{\max}$ such that $\pi(x) < \pi(x_{\max})$

$$\pi_n(x) = \frac{[\pi(x)/\pi(x_{\max})]^{\gamma_n}}{\int [\pi(u)/\pi(x_{\max})]^{\gamma_n}\,du} \to 0, \quad \text{as n} \to \infty.$$

It is typically impossible to sample from $\{\pi_n\}_{n\geq 1}$, so the simulated annealing algorithm approximately samples from π_n by using an MCMC kernel K_n of invariant distribution π_n. Simulated annealing is thus nothing but a nonhomogeneous version of MCMC. For "historical" reasons, simulated annealing is often associated with the simple MH algorithm [47] or with Gibbs sampling [24]. However, it should be clear that any MCMC algorithm (Gibbs sampler, MH, reversible jump) can be used to perform global optimization by substituting π with π_n. To ensure convergence of the resulting simulated Markov chain towards the set of global optima, one needs to ensure that the target distributions are slowly time-varying. Convergence towards this set can be ensured for a logarithmic so-called "cooling schedule," i.e.,

$$\gamma_n = a\ln(n+b);$$

see [2] for a convergence proof in the continuous state-space case. This sequence proceeds too slowly to infinity, so practitioners are using a faster cooling schedule such as

$$\gamma_n = an + b.$$

Note that this method only applies to the case where $\pi(x)$ can be estimated pointwise. In particular, this method does not apply if one is interested in estimating (12) and if (10) is not known up to a normalizing constant. A solution to this problem has been recently proposed in [23]; this is the state augmentation for marginal estimation (SAME) algorithm. The idea is to consider at iteration n the artificial target distribution

$$\pi_n\left(x_1, x_{2,1}, x_{2,2}, \ldots, x_{2,\gamma_n}\right) \propto \prod_{k=1}^{\gamma_n} \pi(x_1, x_{2,k}), \qquad (33)$$

where the set of variables we want to integrate has been artificially replicated γ_n times, γ_n obviously being an integer. It follows that the marginal distribution

$$\begin{aligned}\pi_n(x_1) &= \int \cdots \int \pi_n\left(x_1, x_{2,1:\gamma_n}\right) dx_{2,1:\gamma_n} \\ &= \frac{\pi^{\gamma_n}(x_1)}{\int \pi^{\gamma_n}(u)\,du}.\end{aligned} \qquad (34)$$

Now we can use at iteration n, an MCMC kernel of invariant distribution (33) to sample approximately from it. Marginally, the samples $\{x_1^{(n)}\}$ will be approximately distributed according to (34). Similar to standard simulated annealing, it can be shown under regularity assumptions that convergence towards the set of global maxima can be ensured for a logarithmic cooling schedule. The price to pay is that the algorithm is getting more and more computationally intensive as n increases because the number of variables $x_{2,1:\gamma_n}$ to sample also increases.

APPLICATIONS

BLIND EQUALIZATION

We consider the model described in an earlier section also titled "Blind Equalization." We have established that the posterior distribution $p(b_{2-L:T}|y_{1:T})$ is given by (3). There are 2^{T+L-1} possibilities for the binary sequence $b_{2-L:T}$. To approximate $p(b_{2-L:T}|y_{1:T})$, we next discuss several MCMC algorithms.

In this case, the full conditional distributions can be easily computed as shown in the equation at the bottom of the page. The unknown normalizing constant of $p(b_{2-L:T}|y_{1:T})$ does not need to be known since it appears in both the numerator and denominator of the full conditional distributions. A Gibbs sampler to sample from $p(b_{2-L:T}|y_{1:T})$ proceeds as follows.

ALGORITHM 1—SINGLE-SITE GIBBS SAMPLER

Initialization.

$$\begin{aligned}p\left(b_k = 1 \middle| y_{1:T}, b_{2-L:k-1}, b_{k+1:T}\right) &= 1 - p\left(b_k = 0 \middle| y_{1:T}, b_{2-L:k-1}, b_{k+1:T}\right) \\ &= \frac{p\left(b_{2-L:k-1}, b_{k+1:T}, b_k = 1 \middle| y_{1:T}\right)}{p\left(b_{2-L:k-1}, b_{k+1:T}, b_k = 1 \middle| y_{1:T}\right) + p\left(b_{2-L:k-1}, b_{k+1:T}, b_k = 0 \middle| y_{1:T}\right)}.\end{aligned}$$

```
■ Select randomly or deterministically $b^{(0)}_{2-L:T}$.
Iteration $n$ $(n \geq 1)$.
■ For $k = 2 - L : T$
    – Sample $b^{(n)}_{k} \sim p(\cdot|y_{1:T}, b^{(n)}_{2-L:k-1}, b^{(n-1)}_{k+1:T})$.
End For
```

This is just a particular Gibbs sampler. Instead of sampling the symbols b_k one at a time, it is also possible to sample them by subblocks of length M. For the sake of simplicity, assuming that $K = (T + L - 1)/M$ is an integer, one obtains the following algorithm.

ALGORITHM 2—BLOCK GIBBS SAMPLER

```
Initialization.
■ Select randomly or deterministically $b^{(0)}_{2-L:T}$.
Iteration $n$ $(n \geq 1)$.
■ For $k = 1 : K$
    – Sample $b^{(n)}_{2-L+(k-1)M:2-L+kM-1} \sim$
    $p(\cdot|y_{1:T},\ b^{(n)}_{2-L:(k-1)M-1},\ b^{(n-1)}_{2-L+kM:T})$.
End For
```

This algorithm is expected to converge faster than the previous one as variables are updated in larger blocks. The price to pay is that the computational complexity to compute full conditional distributions increases exponentially with M.

Note that both algorithms will eventually yield some samples approximately distributed according to $p\left(b_{2-L:T}\middle|\, y_{1:T}\right)$. Thus, we can directly estimate $p\left(b_k\middle|\, y_{1:T}\right)$ for any k. If we want to estimate the marginal distribution of the unknown parameters (h, σ^2), it is sufficient to add into the Gibbs sampler loop the following sampling step:

$$\left(h^{(n)}, \sigma^{2(n)}\right) \sim p\left(\cdot\middle|\, y_{1:T}, b^{(n)}_{2-L:T}\right).$$

Given the conjugate prior distribution (2), $p\left(h, \sigma^2\middle|\, y_{1:T}, b_{2-L:T}\right)$ is equal to

$$\begin{aligned}&\sigma^2 \sim \mathcal{IG}\\&\left(\frac{T+\gamma_0}{2}, \frac{\upsilon_0 + y^{\mathrm{T}}_{1:T} y_{1:T} - m^{\mathrm{T}}\left(b_{2-L:T}\right)\Sigma^{-1}\left(b_{2-L:T}\right) m\left(b_{2-L:T}\right)}{2}\right),\\&h\middle|\, \sigma^2 \sim \mathcal{N}\left(m\left(b_{2-L:T}\right), \sigma^2 \Sigma\left(b_{2-L:T}\right)\right).\end{aligned} \tag{35}$$

Clearly, since $b^{(n)}_{2-L:T} \sim p(b_{2-L:T}|y_{1:T})$ (for large n), $(b^{(n)}_{2-L:T}, h^{(n)}, \sigma^{2(n)}) \sim p(b_{2-L:T}, h, \sigma^2|y_{1:T})$ and thus marginally $(h^{(n)}, \sigma^{2(n)}) \sim p(h, \sigma^2|y_{1:T})$.

An alternative method to sample from the joint distribution $p\left(b_{2-L:T}, h, \sigma^2\middle|\, y_{1:T}\right)$ involves using the following Gibbs sampling method. This is known as data augmentation in the literature.

ALGORITHM 3—DATA AUGMENTATION

```
Initialization.
■ Select randomly or deterministically $(h^{(0)}, \sigma^{2(0)})$.
Iteration $n$ $(n \geq 1)$.
■ Sample $b^{(n)}_{2-L:T} \sim p(\cdot|y_{1:T}, h^{(n-1)}, \sigma^{2(n-1)})$ using the
  forward-backward sampling formula.
■ Sample $(h^{(n)}, \sigma^{2(n)}) \sim p(\cdot|y_{1:T}, b^{(n)}_{2-L:T})$.
```

Sampling from $p(b_{2-L:T}|y_{1:T}, h, \sigma^2)$ can be achieved using the forward-filtering backward-sampling formula described in [9]; this is a sampling version of the standard forward-backward algorithm for HMM. Sampling from $p(h, \sigma^2|y_{1:T}, h, \sigma^2)$ can be achieved using (35). This data augmentation algorithm updates all variables in two sampling steps compared to $T + L - 1$ for the single-site Gibbs sampler. Intuitively, one might think that consequently data augmentation is more efficient. However, this is not always the case. In particular, if the SNR is high, then the distributions $p(b_{2-L:T}|y_{1:T}, h, \sigma^2)$ and $p(h, \sigma^2|y_{1:T}, b_{2-L:T})$ are very peaky and the Markov chain does not mix well. It is often preferable to use the single-site or block Gibbs sampler. If the chain does not mix, simulated tempering can be used. Finally, to maximize the joint distribution $p(b_{2-L:T}|y_{1:T})$, one can use a simulated annealing version of the single-site or block Gibbs sampler. To maximize $p(h, \sigma^2|y_{1:T})$, it is possible to use the SAME algorithm.

Note that for this problem, one has $p(b_{2-L:T}|y_{1:T}) = p(-b_{2-L:T}|y_{1:T})$ for any sequence $b_{2-L:T}$, as there is an identifiability problem. It follows that $p(b_k|y_{1:T}) = 0.5$ for any b_k. There are two ways to solve this problem. We can find a maximum $p(b_{2-L:T}|y_{1:T})$ and recover the phase using differential encoding. An alternative is to set a prior on the channel h, enforcing say the positivity of its first coefficient h_0. In this case, it is not possible to compute $p(b_{2-L:T}|y_{1:T})$ up to a normalizing constant anymore, so the single-site and block Gibbs samplers cannot be used, but the data augmentation algorithm can still be applied (suitably modified so as to ensure the constraint).

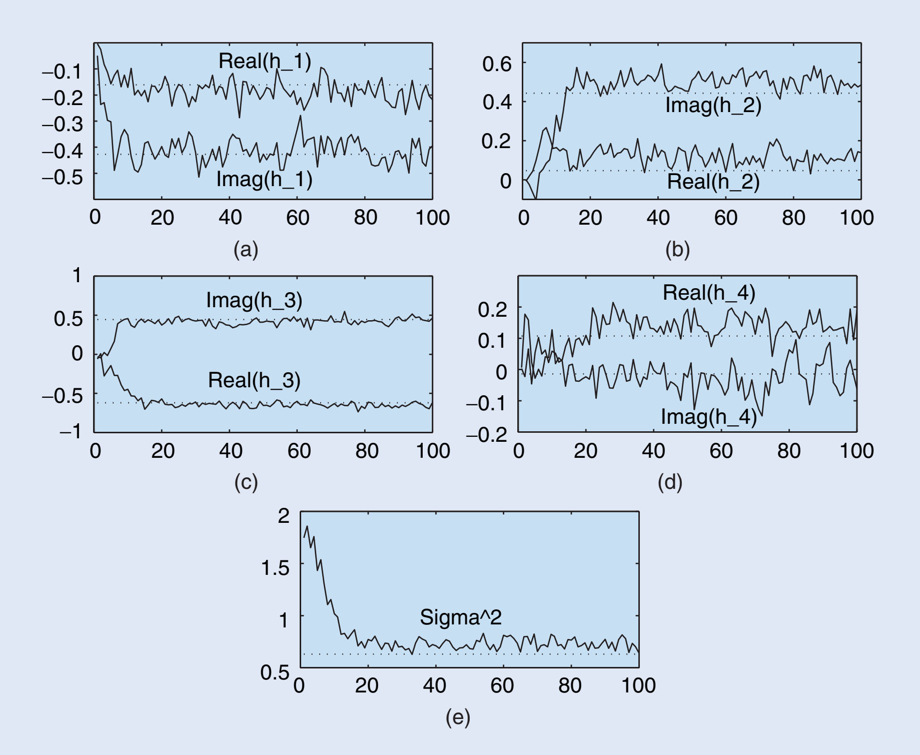
Blind equalization using MCMC was first proposed in [10]. Its application in the context of blind turbo equalization in coded communication systems with either Gaussian or non-Gaussian ambient noise was developed in [49]. In [13], the convergence properties of the various MCMC schemes discussed above are examined for several blind data detection problems found in digital communications, such as blind equalization and blind multiuser detection.

Simulation example: We next provide simulation examples to illustrate the performance of the MCMC blind equalizer based on the single-site Gibbs sampler. We consider a four-tap ISI channel with complex tap coefficients

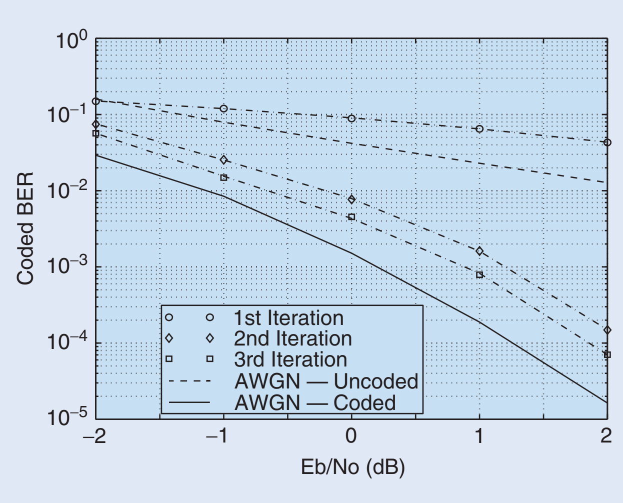
$$\begin{aligned}h = [&-0.1611 - j0.4270,\ 0.0467 + j0.4429,\\&-0.6204 + j0.4436,\ 0.1072 - j0.0140]^T.\end{aligned}$$

(Note that the channel is normalized to have unit norm, i.e., $\|h\| = 1$.) To resolve the delay and phase ambiguities inherent to the blind equalizer, in the Gibbs sampler, we impose the constraints that $|h_3| > |h_l|$ for $l \in \{1, 2, 4\}$ and $\frac{\pi}{2} < \angle h_3 \leq \pi$. The channel code is a rate 1/2 constraint length-5 convolutional code (with generators 23, 35 in octal notation). The interleaver is generated randomly and fixed for all simulations. The block size of the information bits is 128 (i.e., $M = 256$). The code bits are binary phase shift keying (BPSK) modulated, i.e., $b_k \in \{+1, -1\}$. In computing the symbol probabilities, the Gibbs sampler is iterated 100 runs for each data block, with the first 50 iterations as the burn-in period. The following noninformative conjugate prior distributions are used in the Gibbs sampler.

$$h^{(0)} \sim \mathcal{N}(0,\ 1000\,\sigma^{2^{(0)}} I), \quad \sigma^{2^{(0)}} \sim \mathcal{IG}(1,\ 0.1).$$



[FIG1] **Samples drawn by the MCMC blind equalizer in a Gaussian ISI channel.** $E_b/N_o = 2$ dB.



[FIG2] **Bit error rate performance of the MCMC blind turbo equalizer in a Gaussian ISI channel.**

In blind turbo equalization, for the first iteration, the prior symbol probabilities $p(b_k = +1) = 1/2$ for all symbols; in the subsequent iterations, the prior symbol probabilities are provided by the channel decoder. The decoder-assisted convergence assessment is employed. Specifically, if the number of bit corrections made by the decoder exceeds 1/3 of the total number of bits (i.e., $M/3$), then it is decided that convergence is not reached and the Gibbs sampler is applied to the same data block again [49].

We first illustrate the performance of the MCMC blind equalizer in Gaussian ambient noise. In Figure 1, the first 100 samples drawn by the Gibbs sampler for the channel taps (h_1, h_2, h_3, h_4) and the noise variance σ^2 are shown. The corresponding true values of these quantities are also shown in the same figure as the dotted lines. It is seen that the Gibbs sampler reaches convergence rapidly (within about 20 iterations). Figure 2 illustrates the bit error rate performance of the MCMC blind turbo equalizer. The code bit error rate at the output of the blind equalizer is plotted for the first three iterations. The curve corresponding to the first iteration is the uncoded bit error rate at the output of the blind equalizer. The uncoded and coded bit error rate curves in an additive white Gaussian noise (AWGN) ISI-free channel are also shown in the same figure (as, respectively, the dashed and solid lines). It is seen that by incorporating the extrinsic information provided by the channel decoder as the prior symbol probabilities, the MCMC blind equalizer achieves performance that is close to the receiver performance in an ideal AWGN channel in a few iterations.

DECONVOLUTION OF IMPULSIVE SEQUENCES

We consider now the model described in the section "Deconvolution of Impulsive Sequences." First consider the case where the parameters θ are known. We are interested in sampling from $p(i_{2-L:T}, b_{2-L:T} | y_{1:T})$. A simple strategy would consist of using the following Gibbs sampling algorithm.

ALGORITHM 3—DATA AUGMENTATION

Initialization.

- Select randomly or deterministically $(h^{(0)}, \sigma^{2(0)})$.

Iteration n $(n \geq 1)$.

- Sample $b^{(n)}_{2-L:T} \sim p(\cdot | y_{1:T}, h^{(n-1)}, \sigma^{2(n-1)})$ using the forward-backward sampling formula.
- Sample $(h^{(n)}, \sigma^{2(n)}) \sim p(\cdot | y_{1:T}, b^{(n)}_{2-L:T})$.

In this case, it is possible to sample from $p(b_{2-L:T}|y_{1:T}, i_{2-L:T})$ using the forward-filtering backward-sampling formula [9]. To sample from $p\left(i_{2-L:T}\right| y_{1:T}, b_{2-L:T}\right)$, we note that

$$p(i_{2-L:T}|y_{1:T}, b_{2-L:T}) = \prod_{k=2-L}^{T} p\left(i_k | b_k\right).$$

This Markov chain admits $p(i_{2-L:T}, b_{2-L:T}|y_{1:T})$ as invariant distribution but does not converge towards it! Indeed, assume one has $i_k^{(n-1)} = 0$, then it follows that $b_k^{(n-1)} = 0$ because of (5). Now given $b_k^{(n-1)} = 0$, one has $p(i_k = 0|b_k^{(n-1)}) = 1$ so $i_k^{(n)} = 0$. The Markov chain cannot get out of this trapping state and cannot converge towards its invariant distribution. A simple way to avoid this problem is to sample from the marginal distribution $p(i_{2-L:T}|y_{1:T})$, which is known up to a normalizing constant. A single-site Gibbs sampler would proceed as follows.

ALGORITHM 2—SINGLE-SITE GIBBS SAMPLER

```
Initialization.
    ■ Select randomly or deterministically $i_{2-L:T}^{(0)}$.
Iteration $n (n \geq 1)$.
    ■ For $k = 2 - L : T$
        – Sample $i_k^{(n)} \sim p(\cdot| y_{1:T}, i_{2-L:k-1}^{(n)}, i_{k+1:T}^{(n-1)})$.
End For
```

This chain provides samples $\{i_{2-L:T}^{(n)}\}$ asymptotically distributed according to $p(i_{2-L:T}|y_{1:T})$. If we are interested in obtaining samples from the joint distribution $p(i_{2-L:T}, b_{2-L:T}|y_{1:T})$, then we can sample $b_{2-L:T}^{(n)} \sim p(\cdot|y_{1:T}, i_{2-L:T}^{(n)})$ using the simulation smoother.

Now consider the case where the parameters θ are unknown and have selected the prior distribution given by (6). In this case, we could use the following algorithm to sample from $p(i_{2-L:T}, \theta\, |y_{1:T})$.

ALGORITHM 3—MCMC ALGORITHM (GIBBS + MH STEPS)

```
Initialization.
    ■ Select randomly or deterministically $(i_{2-L:T}^{(0)}, \theta^{(0)})$.
Iteration $n\ (n \geq 1)$.
    ■ For $k = 2 - L : T$
        – Sample $i_k^{(n)} \sim p(\cdot| y_{1:T}, \theta^{(n-1)}, i_{2-L:k-1}^{(n)}, i_{k+1:T}^{(n-1)})$.
    ■ Sample $\theta^* \sim q\left(\cdot| \theta^{(n-1)}\right)$. With probability
        $\min\left(1, \frac{p\left(i_{2-L:T}^{(n)}, \theta^* \middle| y_{1:T}\right) q\left(\theta^{(n-1)} \middle| \theta^*\right)}{p\left(i_{2-L:T}^{(n)}, \theta^{(n-1)} \middle| y_{1:T}\right) q\left(\theta^* \middle| \theta^{(n-1)}\right)}\right)$
    set $\theta^{(n)} = \theta^*$; otherwise $\theta^{(n)} = \theta^{(n-1)}$.
```

It appears impossible to sample from $p(\theta|y_{1:T}, i_{2-L:T})$ exactly, so we propose to update the parameter values using an MH step. Unfortunately, the design of a good proposal distribution $q\left(\cdot|\cdot\right)$ might be difficult. An alternative is given by the following (collapsed) Gibbs sampling algorithm.

ALGORITHM 4—COLLAPSED GIBBS SAMPLER

```
Initialization.
    ■ Select randomly or deterministically $i_{2-L:T}^{(0)}$, .
Iteration $n (n \geq 1)$.
    ■ For $k = 2 - L : T$
        – Sample $i_k^{(n)} \sim p(\cdot| y_{1:T}, \theta^{(n-1)}, i_{2-L:k-1}^{(n)}, i_{k+1:T}^{(n-1)})$.
End For
    ■ Sample $b_{2-L:T}^{(n)} \sim p(\cdot| y_{1:T}, i_{2-L:T}^{(n)})$.
    ■ Sample $\theta^{(n)} \sim p(\cdot| y_{1:T}, i_{2-L:T}^{(n)}, b_{2-L:T}^{(n)})$.
```

This algorithm is a so-called collapsed Gibbs sampler since we do not sample the latent variables $i_{2-L:T}$ conditional on $b_{2-L:T}$, but integrate out these variables. After having sampled $i_{2-L:T}^{(n)}$, we sample $b_{2-L:T}^{(n)}$ as the conditional distribution $p(\theta|y_{1:T}, i_{2-L:T}, b_{2-L:T})$ admits a simple form. Indeed, using (6), we obtain

$$\begin{aligned} p\left(\theta| y_{1:T}, i_{2-L:T}, b_{2-L:T}\right) &= p\left(h, \sigma^2, \lambda, \sigma_b^2 \middle| y_{1:T}, i_{2-L:T}, b_{2-L:T}\right) \\ &= p(h, \sigma^2 \big| y_{1:T}, i_{2-L:T}, b_{2-L:T}) \\ &\quad \times p\left(\lambda| i_{2-L:T}\right) p\left(\sigma_b^2 \middle| i_{2-L:T}, b_{2-L:T}\right), \end{aligned}$$

where

$$\sigma^2 \sim \mathcal{IG}\left(\frac{T+\gamma_0}{2}, \frac{\upsilon_0 + y_{1:T}^{\mathrm{T}} y_{1:T} - m^{\mathrm{T}}\left(b_{2-L:T}\right) \Sigma^{-1}\left(b_{2-L:T}\right) m\left(b_{2-L:T}\right)}{2}\right),$$

$$h| \sigma^2 \sim \mathcal{N}\left(m\left(b_{2-L:T}\right), \sigma^2 \Sigma\left(b_{2-L:T}\right)\right),$$

$$\lambda \sim \mathcal{B}(\zeta + n(i_{2-L:T}), \tau + T + L - 1 - n(i_{2-L:T})),$$

$$\sigma_b^2 \sim \mathcal{IG}\left(\frac{\gamma_b + n\left(i_{2-L:T}\right)}{2}, \frac{\upsilon_b + \sum_{k=2-L}^{T} i_k b_k^2}{2}\right).$$

To maximize the marginal distribution $p\left(i_{2-L:T}\right| y_{1:T}\right)$ or $p\left(\theta| y_{1:T}\right)$, one can use the SAME algorithm.

SEQUENTIAL MONTE CARLO METHODS

All the methods we have described thus far enable sampling approximately from a distribution $\pi\left(x\right)$ or a sequence of distributions $\{\pi_n\left(x\right)\}$ varying slowly over time (like in the simulated annealing case). Clearly, this type of algorithm is not well adapted to problems such as optimal filtering; as mentioned earlier, in optimal filtering, we are interested in estimating a sequence of potentially quickly varying distributions whose dimension is increasing over time.

Sequential Monte Carlo (SMC) methods are a class of simulation-based methods solving this type of problem. More precisely, SMC methods approximate a sequence of probability distributions $\{\pi_n\left(x_{1:n}\right)\}$, with each distribution π_n being defined on $\mathcal{X}^n$. These methods have become very popular over the last few years as they help solve numerous optimal filtering problems;

see [22] for a review of the literature illustrated by many applications. These methods are based on a combination of sequential importance sampling and resampling mechanisms.

Assume that at time $(n-1)$ we have a set of N weighted random samples named particles $\{x_{1:n-1}^{(i)}, w_{n-1}^{(i)}\}$ $[w_{n-1}^{(i)} > 0$ and $\sum_{i=1}^{N} w_{n-1}^{(i)} = 1]$approximating the target distribution π_{n-1}, i.e.,

$$\widehat{\pi}_{n-1}(x_{1:n-1}) = \sum_{i=1}^{N} w_{n-1}^{(i)} \delta\left(x_{1:n-1} - x_{1:n-1}^{(i)}\right)$$

and for any test function $\varphi_{n-1} : \mathcal{X}^{n-1} \to \mathbb{R}$

$$\begin{aligned}&\int \varphi_{n-1}(x_{1:n-1})\widehat{\pi}_{n-1}(x_{1:n-1})dx_{1:n-1}\\ &= \sum_{i=1}^{N} w_{n-1}^{(i)} \varphi_{n-1}(x_{1:n-1}) \to E_{\pi_{n-1}}(\varphi_{n-1}), \quad N \to \infty.\end{aligned}$$

At time n, our objective is to derive a simple mechanism to obtain N weighted particles $\{x_{1:n}^{(i)}, w_n^{(i)}\}$ approximating π_n.

SEQUENTIAL IMPORTANCE SAMPLING AND RESAMPLING

Given the weighted particles $\{x_{1:n-1}^{(i)}, w_{n-1}^{(i)}\}$ approximating π_{n-1}, we consider extending each path $x_{1:n-1}^{(i)}$ by sampling

$$x_n^{(i)} \sim q_n\left(\cdot |\, x_{1:n-1}^{(i)}\right).$$

It follows that the weights of each particle should be updated according to

$$w_n^{(i)} \propto w_{n-1}^{(i)} \underbrace{\frac{\pi_n\left(x_{1:n}^{(i)}\right)}{\pi_{n-1}\left(x_{1:n-1}^{(i)}\right) q_n\left(x_n^{(i)} \middle| x_{1:n-1}^{(i)}\right)}}_{\text{incremental weight}}$$

with $\sum_{i=1}^{N} w_n^{(i)} = 1$. Indeed, this method is nothing but a simple instance of importance sampling with importance distribution given at time n by

$$q_n(x_{1:n}) = \mu(x_1) \prod_{k=2}^{n} q_k\left(x_k |\, x_{1:k-1}\right).$$

The importance weight expression follows from

$$\begin{aligned} w_n(x_{1:n}) &= \frac{\pi_n(x_{1:n})}{q_n(x_{1:n})}\\ &= \frac{\pi_{n-1}(x_{1:n-1})}{q_{n-1}(x_{1:n-1})} \frac{\pi_n(x_{1:n})}{\pi_{n-1}(x_{1:n})\, q_n(x_n |\, x_{1:n-1})}\\ &= w_{n-1}(x_{1:n-1}) \frac{\pi_n(x_{1:n})}{\pi_{n-1}(x_{1:n})\, q_n(x_n |\, x_{1:n-1})}, \end{aligned}$$

where $w_n(x_{1:n})$ denotes the unnormalized weight at time n.

The efficiency of this method depends crucially on the selection of the importance distribution $q_n(\cdot|\cdot)$. It can be easily established that the distribution minimizing the variance of the incremental weight conditional on $\{x_{1:n-1}^{(i)}\}$ is given by

$$q_n^{\text{opt}}(x_n |x_{1:n-1}) = \pi_n(x_n |x_{1:n-1}).$$

In this case, the incremental weight is given by

$$\frac{\pi_n(x_{1:n})}{\pi_{n-1}(x_{1:n-1})\pi_n(x_n |x_{1:n-1})} = \frac{\pi_n(x_{1:n-1})}{\pi_{n-1}(x_{1:n-1})}. \quad (36)$$

Typically, it is difficult to sample from $\pi_n(x_n |x_{1:n-1})$, and the associated incremental weight cannot be computed in closed form. So a standard strategy consists of using an approximation of $\pi_n(x_n |x_{1:n-1})$ as an importance distribution [20], [22]. However, whichever importance distribution is used, the variance of the importance weights typically increases over time. Intuitively, this is because typically the discrepancy between the importance distribution $q_n(x_{1:n})$ and $\pi_n(x_{1:n})$ increases over time in real-world applications. Consequently, after a few time steps, all particles but one have a weight close to zero and the remaining one has a weight close to one. This results in worthless Monte Carlo estimates having a huge variance.

To make this method efficient, it is necessary to introduce a resampling step so as to control the variance of the importance weights. If at time n the variance of the importance weights is high, then we resample N times from the current weighted approximation

$$\widehat{\pi}_n(x_{1:n}) = \sum_{i=1}^{N} w_n^{(i)} \delta\left(x_{1:n} - x_{1:n}^{(i)}\right)$$

to obtain a new approximation

$$\frac{1}{N}\sum_{i=1}^{N} \delta\left(x_{1:n} - \widetilde{x}_{1:n}^{(i)}\right),$$

where $\{\widetilde{x}_{1:n}^{(i)}\}$ are the resampled particles. This was first suggested in the optimal filtering context in [28], and this is the key step of SMC methods. In this new approximation, it is possible to have $\widetilde{x}_{1:n}^{(i)} = \widetilde{x}_{1:n}^{(j)}$ for $i \neq j$. The resampling step has the effect of concentrating the computational efforts on the relevant zones of the state-space. Locally (in time), it does introduce some variance, but it is beneficial in the next time steps. A standard measure of variation for the importance weights is the effective sample size (ESS) proposed in [36]

$$\text{ESS}\left(\left\{w_n^{(i)}\right\}\right) = \left(\sum_{i=1}^{N} \left(w_n^{(i)}\right)^2\right)^{-1}.$$

This quantity takes values between 1 and N; the larger, the better. Resampling is typically performed when this measure is below a threshold equal to say $N/2$.

Note that the resampling step consists of performing an approximation of $\widehat{\pi}_n\left(x_{1:n}\right)$ by

$$\sum_{i=1}^{N}\frac{N_n^{(i)}}{N}\delta\left(x_{1:n}-x_{1:n}^{(i)}\right),$$

where $N_n^{(i)}$ is the number of copies of $x_{1:n}^{(i)}$; each being given a weight $1/N$. If one resamples N times from $\widehat{\pi}_n\left(x_{1:n}\right)$, then $\{N_n^{(i)}\}$ are distributed according to a multinomial distribution of parameters $\{w_n^{(i)}\}$ and thus satisfy $E(N_n^{(i)}) = Nw_n^{(i)}$. It is possible to reduce the variance of the resampling scheme. Many methods have been proposed in the literature [22]. To sum up, the sequential importance sampling and resampling (SISR) algorithm proceeds as follows.

SEQUENTIAL IMPORTANCE SAMPLING AND RESAMPLING

```
Initialization.
  - Sample $x_1^{(i)} \sim q_1$ for $i = 1, \ldots, N$.
  - Compute and normalized the weights
      $w_1^{(i)} \propto \frac{\pi_1\left(x_1^{(i)}\right)}{q_1\left(x_1^{(i)}\right)}$.
  - If ESS $(\{w_1^{(i)}\}) <$ Threshold, resample $\{x_1^{(i)}, w_1^{(i)}\}$ to obtain
    $\{x_1^{(i)}, N^{-1}\}$.
Iteration $n (n \geq 2)$.
  - Sample $x_n^{(i)} \Big| x_{n-1}^{(i)} \sim q_n(\cdot | x_{1:n-1}^{(i)})$ for $i = 1, \ldots, N$.
  - Compute and normalized the weights
      $w_n^{(i)} \propto w_{n-1}^{(i)} \frac{\pi_n\left(x_{1:n}^{(i)}\right)}{\pi_{n-1}\left(x_{1:n-1}^{(i)}\right) q_n\left(x_n^{(i)} \Big| x_{1:n-1}^{(i)}\right)}$.
  - If ESS $(\{w_n^{(i)}\}) <$ Threshold, resample $\{x_{1:n}^{(i)}, w_n^{(i)}\}$ to
    obtain $\{x_{1:n}^{(i)}, N^{-1}\}$.
```

To simplify notation, we use $\{x_{1:n}^{(i)}\}$ for the particles before and after resampling rather than using $\{\widetilde{x}_{1:n}^{(i)}\}$. This algorithm has a computational complexity of order $\mathcal{O}\left(N\right)$.

There are multiple variants of this algorithm; further details are given in [22]. Many convergence results are available on SMC algorithms; a complete review of these is presented in [16]. For a more accessible treatment, an introduction to convergence results for engineers is presented in [14] and [15].

APPLICATION TO OPTIMAL FILTERING

FILTERING IN GENERAL STATE-SPACE MODELS

We detail here the application of the SISR methodology to the optimal filtering problem discussed in the section "Optimal Filtering in State-Space Models." In this example, one is interested in estimating an unobserved Markov process $\{x_n\}_{n\geq 1}$ of initial density μ and transition density $x_n | x_{n-1} \sim f(\cdot | x_{n-1})$. The observations $\{y_n\}_{n\geq 1}$ are conditionally independent given $\{x_n\}_{n\geq 1}$ of marginal density $y_n | x_n \sim g\left(\cdot | x_n\right)$. In this case, the sequence of target distributions of interest $\{\pi_n\}_{n\geq 1}$ is given by

$$\begin{aligned}\pi_n\left(x_{1:n}\right) &= p\left(x_{1:n} | y_{1:n}\right)\\ &\propto \mu\left(x_1\right)\prod_{k=2}^{n} f\left(x_k | x_{k-1}\right)\prod_{k=1}^{n} g\left(y_k | x_k\right).\end{aligned}$$

The importance sampling distribution can only depend at time n on the observations until time n, i.e., one has an importance distribution of the form $q_n(x_n | y_{1:n}, x_{1:n-1})$. The optimal importance distribution is given in this case by

$$\begin{aligned}\pi_n(x_{1:n} | x_{1:n-1}) &= \frac{\pi_n(x_{1:n})}{\pi_n(x_{1:n-1})} = \frac{p(x_{1:n} | y_{1:n})}{p(x_{1:n-1} | y_{1:n-1})}\\ &= p(x_n | x_{n-1}, y_n)\\ &= \frac{f(x_n | x_{n-1}) g(y_n | x_n)}{\int f(x_n | x_{n-1}) g(y_n | x_n) dx_n}\end{aligned}$$

and its associated incremental importance weight (36) is proportional to

$$p(y_n | x_{n-1}) = \int f(x_n | x_{n-1}) g(y_n | x_n) dx_n.$$

In practice, we always limit ourselves to distributions of the form $q(x_n | y_n, x_{n-1})$, and in this case the general form of the incremental weight is given by

$$\frac{p\left(x_{1:n} | y_{1:n}\right)}{p\left(x_{1:n-1} | y_{1:n-1}\right) q\left(x_n | y_n, x_{n-1}\right)} \propto \frac{f\left(x_n | x_{n-1}\right) g\left(y_n | x_n\right)}{q\left(x_n | y_n, x_{n-1}\right)}.$$

Typically, one selects $q(x_n | y_n, x_{n-1})$ as an approximation of $p(x_n | y_n, x_{n-1})$ using the extended Kalman filter (EKF) or unscented Kalman filter (UKF) [20], [46]. To sum up, the SISR algorithm for filtering state-space model proceeds as follows.

```
SISR for Optimal Filtering
Initialization.
  - Sample $x_1^{(i)} \sim q_1$ for $i = 1, \ldots, N$.
  - Compute and normalized the weights
      $w_1^{(i)} \propto \frac{\mu\left(x_1^{(i)}\right) g\left(y_1 | x_1^{(i)}\right)}{q\left(x_1^{(i)} \Big| y_1\right)}$.
  - If ESS $(\{w_1^{(i)}\}) <$ Threshold, resample $\{x_1^{(i)}, w_1^{(i)}\}$ to obtain
    $\{x_1^{(i)}, N^{-1}\}$.
Iteration $n$ $(n \geq 1)$.
  - Sample $x_n^{(i)} \sim q(\cdot | y_n, x_{n-1}^{(i)})$ for $i = 1, \ldots, N$.
```

- Compute and normalized the weights

$$w_n^{(i)} \propto w_{n-1}^{(i)} \frac{f\left(x_n^{(i)} \middle| x_{n-1}^{(i)}\right) g\left(y_n \middle| x_n^{(i)}\right)}{q\left(x_n^{(i)} \middle| y_n, x_{1:n-1}^{(i)}\right)}.$$

- If ESS $(\{w_n^{(i)}\}) <$ Threshold, resample $\{x_{1:n}^{(i)}, w_n^{(i)}\}$ to obtain $\{x_{1:n}^{(i)}, N^{-1}\}$.

MIXTURE KALMAN FILTER FOR CONDITIONALLY LINEAR GAUSSIAN STATE-SPACE MODELS

For a few important classes of state-space models, it is possible to come up with more efficient SMC methods. This includes conditionally linear Gaussian state-space models [11], [20], [21] and partially observed Gaussian state-space models [5]. Consider the case where

$$\begin{aligned} x_n \,|\, x_{n-1} &\sim f\left(\cdot \,|\, x_{n-1}\right), \\ z_n &= A\left(x_n\right) z_{n-1} + B\left(x_n\right) v_n, \\ y_n &= C\left(x_n\right) z_n + D\left(x_n\right) w_n, \end{aligned}$$

where $x_1 \sim \mu$, $z_0 \sim \mathcal{N}(m_{0,} \Sigma_0)$, and $v_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, I_{n_v}\right)$ and $w_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, I_{n_w}\right)$ are two mutually independent sequences. In this model, $\{x_n\}$ and $\{z_n\}$ are unobserved whereas $\{y_n\}$ is observed. Clearly, this model is just a particular case of the model discussed in the previous section if one substitutes for x_n the extended state $(x_n, z_n) \in \mathcal{X} \times \mathcal{Z}$ and considers the sequence of posterior distributions $\{p\left(x_{1:n}, z_{1:n} \,|\, y_{1:n}\right)\}$. However, it is possible to come up with a better algorithm. Indeed, one has

WE DESCRIBE GENERIC MONTE CARLO METHODS THAT CAN BE USED TO PERFORM STATISTICAL INFERENCE IN BOTH BATCH AND SEQUENTIAL CONTEXTS.

$$p\left(x_{1:n}, z_{0:n} \,|\, y_{1:n}\right) = p\left(z_{0:n} \,|\, x_{1:n}, y_{1:n}\right) p\left(x_{1:n} \,|\, y_{1:n}\right), \qquad (37)$$

where $p\left(z_{0:n} \,|\, x_{1:n}, y_{1:n}\right)$ is a Gaussian density whose parameters can be computed using Kalman filtering techniques and $p\left(x_{1:n} \,|\, y_{1:n}\right)$ is known up to a normalizing constant

$$p\left(x_{1:n} \,|\, y_{1:n}\right) \propto p\left(y_{1:n} \,|\, x_{1:n}\right) p\left(x_{1:n}\right).$$

Indeed, the likelihood $p\left(y_{1:n} \,|\, x_{1:n}\right)$ is given by

$$p\left(y_{1:n} \,|\, x_{1:n}\right) = p\left(y_1 \,|\, x_1\right) \prod_{k=2}^{n} p(y_k \,|y_{1:k-1}, x_{1:k})$$

and each term $p\left(y_k \,|\, y_{1:k-1}, x_{1:k}\right)$ can be computed using the Kalman filter given $x_{1:k}$.

It follows that it is possible to use SMC methods to estimate the sequence of marginal distributions $\{p\left(x_{1:n} \,|\, y_{1:n}\right)\}$ instead of $\{p\left(x_{1:n}, z_{1:n} \,|\, y_{1:n}\right)\}$. Based on estimates of $\{p\left(x_{1:n} \,|\, y_{1:n}\right)\}$, one can estimate $\{p\left(x_{1:n}, z_{1:n} \,|\, y_{1:n}\right)\}$ using (37). This strategy is more efficient as it does not require sampling particles in $\mathcal{Z}$. The size of the state-space to explore via Monte Carlo simulation is smaller, thus improving the efficiency of the method.

At first glance, it appears that this technique requires storing the entire paths $\{x_{1:n}^{(i)}\}$ at time n. If this was the case, the memory requirements would increase over time and the procedure would be practically useless. Fortunately, this is not the case for all sensible choices of $q\left(x_n \,|\, y_{1:n}, x_{1:n-1}\right)$. Let us, for example, consider the optimal importance distribution given in this case by

$$p(x_n \,|y_{1:n}, x_{1:n-1}) \propto p(y_n \,|y_{1:n-1}, x_{1:n}) f(x_n \,|x_{n-1}).$$

The term $p(y_n \,|y_{1:n-1}, x_{1:n})$ is a Gaussian distribution whose mean $y_{n|n-1}\left(x_{1:n}\right)$ and covariance $S_{n|n-1}\left(x_{1:n}\right)$ can be computed using the Kalman filter. This suggests that all SMC implementations in this context depend on $y_{1:n}$ and $x_{1:n}$ only through a set of fixed-dimensional sufficient statistics. The resulting algorithm corresponds to a random mixture of Kalman filters; see [11], [20], and [21] for details. A similar approach can be adopted to develop an efficient algorithm for conditionally finite state-space HMMs (the Kalman filter being replaced by an HMM filter) [20] and for partially observed linear Gaussian state-space models [5] which have applications for quantized observations.

Simulation Example (Blind detection in fading channel): Suppose we want to transmit binary symbols $x_n \in \{+1, -1\}$, through a fading channel whose input-output relationship is given by

$$y_n = \alpha_n x_n + w_n,$$

where $\{w_n\}$ is a sequence of i.i.d. Gaussian noise. $\{\alpha_n\}$ represents the unobserved Rayleigh fading process, which can be modeled as the output of a low-pass filter of order r driven by white Gaussian noise,

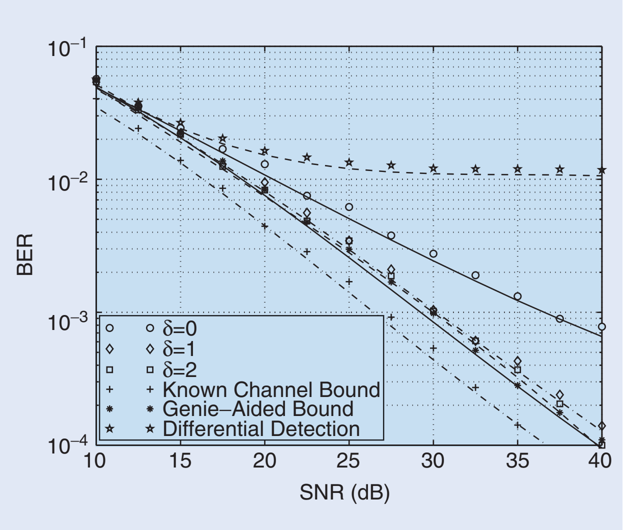
$$\{\alpha_n\} = \frac{\Psi(D)}{\Phi(D)}\{u_n\},$$

where D is the back-shift operator $D^k u_n = u_{n-k}$; $\Phi(z) = \phi_r z^r + \cdots + \phi_1 z + 1$; $\Psi(z) = \psi_r z^r + \cdots + \psi_1 z + \psi_0$; and $\{u_n\}$ is a white complex Gaussian noise sequence with independent real and imaginary components, $u_n \sim \mathcal{N}_c(0, \sigma^2)$. The inference problem is to estimate the transmitted symbols x_n at time n, based on the received signals $\{y_1, \ldots, y_{n+\delta}\}$ for some $\delta \geq 0$, with the knowledge of only the statistics of the fading process. An SMC solution to this problem based on the mixture

Kalman filter was developed in [12]. Here we show a simulation example to demonstrate the performance of such an SMC blind receiver. The fading process is modeled by the output of a Butterworth filter of order $r = 3$ driven by a complex white Gaussian noise process. The cutoff frequency of this filter is 0.05, corresponding to a normalized Doppler frequency (with respect to the symbol rate $(1/T)$ $f_d T = 0.05$, which is a fast-fading scenario. Specifically, the sequence of fading coefficients $\{\alpha_n\}$ is modeled by the following ARMA(3,3) process:

$$\begin{aligned}&\alpha_n - 2.37409\alpha_{n-1} + 1.92936\alpha_{n-2} - 0.53208\alpha_{n-3}\\&= 10^{-2}(0.89409u_n + 2.68227u_{n-1}\\&\quad + 2.68227u_{n-2} + 0.89409u_{n-3}),\end{aligned}$$

where $u_n \sim \mathcal{N}_c(0, 1)$. The filter coefficients are chosen such that $\mathrm{Var}\{\alpha_n\} = 1$. Differential modulation is employed to resolve the phase ambiguity. In the SMC receiver, the number of Monte Carlo samples drawn at each time was empirically set as $N = 50$. The ESS threshold for resampling is set as $N/10$. In Figure 3, the bit error rate (BER) performance versus the signal-to-noise ratio (defined as $\mathrm{Var}\{\alpha_n\}/\mathrm{Var}\{w_n\}$) corresponding to delay values $\delta = 0$ (concurrent estimate), $\delta = 1$, and $\delta = 2$ is plotted. In the same figure, we also plot the known channel lower bound, the genie-aided lower bound, and the BER curve of the differential detector. From this figure, it is seen that the SMC blind receiver does not exhibit an error floor, unlike the differential detector. Moreover, with a delay $\delta = 2$, the SMC receiver essentially achieves the genie-aided lower bound.



[FIG3] BER performance of the SMC receiver in a fading channel.

CONCLUDING REMARKS

In this article, we have introduced two sets of powerful algorithms, MCMC and SMC, to sample and/or maximize high-dimensional probability distributions. These methods enable one to perform likelihood or Bayesian inference for complex nonlinear non-Gaussian models, procedures which were out of reach just a few years ago. It is our belief that these methods have numerous potential applications in signal processing.

We have only presented here a few applications. However, MCMC techniques have recently been applied to solve a number of traditionally "hard" problems found in signal processing and telecommunications. For example, the spectral analysis problem described in the section "Spectral Analysis" is addressed in [1]. Other applications include neural networks [3], target tracking, blind multiuser detection in various CDMA systems (such as DS-CDMA with multipath fading channels [53], nonlinearly modulated CDMA systems [41], and space-time coded multicarrier CDMA systems [51]), blind equalization in various channels (such as impulsive noise channel [49] and systems employing Gaussian minimum-shift-keying modulation [52]), joint synchronization, channel estimation and data detection in OFDM systems [37], and inference of network internal delay and loss characteristics from end-to-end measurements [30].

Similarly, SMC techniques have been used to address various signal processing problems such as chirp tracking [4] and receiver design in fading channels [12], [32], [50]. SMC-based adaptive receivers have also been developed for several other communication systems, such as multiple-antenna systems [19], [31] and OFDM systems [54]. More discussions on the applications of SMC in communications can be found in [18]. Furthermore, SMC-based signal processing methods have also been developed for joint mobility tracking and handoff in cellular wireless networks [55] as well as for target tracking in sensor networks [33], [45], [48]. These methods also have applications in change detection, identification, and control [6].

AUTHORS

Arnaud Doucet graduated from Institut National des Telecommunications in June 1993 and obtained his Ph.D. from University Paris-Sud Orsay in December 1997. He held faculty positions at Melbourne University and Cambridge University. He is an associate professor in the Department of Computer Science and the Department of Statistics of the University of British Columbia. His main research interests are simulation-based methods and their applications to Bayesian statistics.

Xiaodong Wang received the B.S. degree in electrical engineering and applied mathematics (with the highest honor) from Shanghai Jiao Tong University, Shanghai, China, in 1992, an M.S. degree in electrical and computer engineering from Purdue University in 1995, and the Ph.D. in electrical engineering from Princeton University in 1998. From July 1998 to December 2001, he was an assistant professor in the Department of Electrical Engineering, Texas A&M University. In January 2002, he joined the faculty of the Department of Electrical Engineering at Columbia University. His research interests are in the areas of computing, signal processing, and communications. Among his publications is a recent book titled *Wireless Communication Systems: Advanced Techniques for Signal Reception*. He received the 1999 NSF CAREER Award and the 2001 IEEE Communications Society and Information

Theory Society Joint Paper Award. He is an associate editor for *IEEE Transactions on Communications*, *IEEE Transactions on Wireless Communications*, *IEEE Transactions on Signal Processing*, and *IEEE Transactions on Information Theory*.

REFERENCES

[1] C. Andrieu and A. Doucet, "Joint Bayesian detection and estimation of noisy sinusoids via reversible jump MCMC," *IEEE Trans. Signal Processing*, vol. 47, no. 10, pp. 2667–2676, 1999.

[2] C. Andrieu, L. Breyer, and A. Doucet, "Convergence of simulated annealing using Foster-Lyapunov criteria," *J. Applied Probability*, vol. 38, no. 4, pp. 975–994, 2001.

[3] C. Andrieu, N. de Freitas, A. Doucet, "Robust full Bayesian learning for radial basis networks," *Neural Comput.*, vol. 13, no. 10, pp. 2359–2407, 2001.

[4] C. Andrieu, M. Davy, and A. Doucet, "Efficient particle filters for nonlinear jump Markov systems," *IEEE Trans. Signal Processing*, vol. 51, no. 7, pp. 1762–1770, 2003.

[5] C. Andrieu and A. Doucet, "Particle filtering for partially observed Gaussian state space models," *J. Royal Stat. Soc.* B, vol. 64, no. 4, pp. 827–836, 2002.

[6] C. Andrieu, A. Doucet, S.S. Singh and V.B. Tadic, "Particle methods for change detection, system identification and control," *Proc. IEEE*, vol. 92, no. 3, pp. 423–438, 2004.

[7] J.M. Bernardo and A.F.M. Smith, *Bayesian Theory.* New York: Wiley, 1995.

[8] J. Besag, P.J. Green, D. Higdon, and K. Mengersen, "Bayesian computation and stochastic systems," *Stat. Science*, vol. 10, pp. 3–66, 1995.

[9] C.K. Carter and R. Kohn, "On Gibbs sampling for state space models," *Biometrika,* vol. 81, no. 3, pp. 541–553, 1994.

[10] R. Chen and T. Li, "Blind restoration of linearly degraded discrete signals by Gibbs sampling," *IEEE Trans. Signal Processing*, vol. 43, no. 10, pp. 2410–2413, 1995.

[11] R. Chen and J.S. Liu, "Mixture Kalman filters," *J. Roy. Statist. Soc. B*, vol. 62, no. 3, pp. 493–509, 2000.

[12] R. Chen, X. Wang, and J.S. Liu, "Adaptive joint detection and decoding in flat–fading channels via mixture Kalman filtering," *IEEE Trans. Inform. Theory*, vol. 46, no. 6, pp. 2079–2094, 2000.

[13] R. Chen, J.S. Liu, and X. Wang, "Convergence analyses and comparisons of Markov chain Monte Carlo algorithms in digital communications," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 255–270, 2002.

[14] D. Crisan, "Particle filters—A theoretical perspective," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, J.F.G. de Freitas, and N.J. Gordon, Eds. New York: Springer-Verlag, 2001, pp. 17–38.

[15] D. Crisan and A. Doucet, "A survey of theoretical results on particle filtering for practitioners," *IEEE Trans. Signal Processing*, vol. 50, no. 3, pp. 736–746, 2002.

[16] P. Del Moral, *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. New York: Springer-Verlag, 2004.

[17] L. Devroye, *Non-Uniform Random Variate Generation*. New York: Springer, 1986.

[18] P.M. Djuric et al., Particle Filtering, *IEEE Signal Processing Mag.*, vol. 20, no. 5, pp. 19–38, 2003.

[19] B. Dong, X. Wang, and A. Doucet. "A new class of soft MIMO demodulation algorithms," *IEEE Trans. Signal Processing*, vol. 51, no. 11., pp. 2752–2763, 2003.

[20] A. Doucet, S.J. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statist. Comput.*, vol. 10, no. 3, pp. 197–208, 2000.

[21] A. Doucet, N.J. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump Markov linear systems," *IEEE Trans. Signal Processing*, vol. 49, no. 3, pp. 613–624, 2001.

[22] A. Doucet, J.F.G. de Freitas, and N.J. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 2001.

[23] A. Doucet, S.J. Godsill, and C.P. Robert, "Marginal maximum a posteriori parameter estimation using Markov chain Monte Carlo methods," *Statist. Comput.*, vol. 12, pp. 77–84, 2002.

[24] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, no. 11, pp. 721–741, 1994.

[25] J. Geweke, "Bayesian inference in econometrics models using Monte Carlo integration," *Econometrica*, vol. 57, pp. 1317–1339, 1989.

[26] C.J. Geyer and E.A. Thompson, "Annealing Markov chain Monte Carlo with applications to ancestral inference," *J. Am. Statist. Ass.*, vol. 90, pp. 909–920, 1995.

[27] W.R. Gilks, S. Richardson, and D.J. Spiegelhalter, Eds., *Markov Chain Monte Carlo in Practice*. London: Chapman & Hall, 1996.

[28] N.J. Gordon, D.J. Salmond, and A.F.M. Smith, "Novel approach to nonlinear non-Gaussian Bayesian state estimation," *IEE Proc. F*, vol. 140, pp. 107–113, 1993.

[29] P.J. Green, "Reversible jump MCMC computation and Bayesian model determination," *Biometrika*, vol. 82, pp. 711–732, 1995.

[30] D. Guo and X. Wang, "Bayesian inference of network loss and delay characteristics with applications to TCP performance prediction," *IEEE Trans. Signal Processing*, vol. 48, no. 7, pp. 2205–2218, 2003.

[31] D. Guo and X. Wang, "Blind detection in MIMO systems via sequential Monte Carlo," *IEEE J. Select. Areas Commun.*, vol. 21, no. 3, pp. 453–464, 2003.

[32] D. Guo, X. Wang, and R. Chen, "Wavelet-based sequential Monte Carlo blind receivers in fading channels with unknown channel statistics," *IEEE Trans. Signal Processing*, vol. 52, no. 1, pp. 227–239, 2004.

[33] D. Guo and X. Wang, "Dynamic sensor collaboration via sequential Monte Carlo," *IEEE J. Select. Areas Commun*., vol. 22, no. 6, pp. 1037–1047, 2004.

[34] W.K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, pp. 97–109, 1970.

[35] G.K. Kaleh and R. Vallet, "Joint parameter estimation and symbol detection for linear and nonlinear unknown channels," *IEEE Trans. Commun.*, vol. 42, no. 11, pp. 2406–2413, 1994.

[36] J.S. Liu, *Monte Carlo Methods in Scientific Computing*. New York: Springer Verlag, 2001.

[37] B. Lu and X. Wang, "Bayesian blind turbo receiver for coded OFDM systems with frequency offset and frequency-selective fading," *IEEE J. Select. Areas Commun.*, vol. 19, no. 12, pp. 2516–2527, 2001.

[38] J.M. Mendel, *Maximum-Likelihood Deconvolution: A Journey into Model-Based Signal Processing*. New York: Springer-Verlag, 1990.

[39] N. Metropolis, N. Rosenblutt, A.W. Rosenblutt, M.N. Teller, and A.H. Teller, "Equations of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, pp. 1087–1092, 1953.

[40] S.P. Meyn and R.L. Tweedie, *Markov Chains and Stochastic Stability*. New York: Springer-Verlag, 1993.

[41] V.D. Phan and X. Wang, "Bayesian turbo multiuser detection for nonlinearly modulated CDMA," *Signal Processing*, vol. 82, no. 1, pp. 42–68, 2002.

[42] L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[43] C.P. Robert, *The Bayesian Choice*. New York: Springer-Verlag, 1996.

[44] C.P. Robert and G. Casella, *Monte Carlo Statistical Methods*. New York: Springer-Verlag, 1999.

[45] T. Vercauteren, D. Guo, and X. Wang, "Joint multiple target tracking and classification in collaborative sensor networks," *IEEE J. Select. Areas Commun.*, vol. 23, no. 4, pp. 714–723, 2005.

[46] R. van der Merwe, A. Doucet, J.F.G. de Freitas, and E. Wan, "The unscented particle filter,"in *Advances in Neural Information Processing Systems 13*. Cambridge, MA: MIT Press, 2000.

[47] P.J. Van Laarhoven and E.H.L. Arts, *Simulated Annealing: Theory and Applications*. Amsterdam: Reidel, 1987.

[48] B. Vo, S. Singh and A. Doucet, "Sequential Monte Carlo methods for Bayesian multi-target filtering with random finite sets," *IEEE Trans. Aerosp. Electron. Syst.*, to appear in 2005.

[49] X. Wang and R. Chen, "Blind turbo equalization in Gaussian and impulsive noise," *IEEE Trans. Veh. Technol.* vol. 50, no. 4, pp. 1092–1105, 2001.

[50] X. Wang, R. Chen, and D. Guo, "Delayed-pilot sampling for mixture Kalman filter with application in fading channels," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 241–254, 2002.

[51] Z. Yang, B. Lu, and X. Wang, "Bayesian Monte Carlo multiuser receiver for space-time coded multi-carrier CDMA systems," *IEEE J. Select. Areas Commun.*, vol. 19, no. 8, pp. 1625–1637, 2001.

[52] Z. Yang and X. Wang, "Turbo equalization for GMSK signaling over multipath channels based on the Gibbs sampler," *IEEE J. Select. Areas Commun.*, vol. 19, no. 9, pp. 1753–1763, 2001.

[53] Z. Yang and X. Wang, "Blind turbo multiuser detection for long-code multipath CDMA," *IEEE Trans. Commun.*, vol. 50, no. 9, pp. 112–125, 2002.

[54] Z. Yang and X. Wang, "Blind detection of OFDM signals in multipath fading channels via sequential Monte Carlo," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 271–280, 2002.

[55] Z. Yang and X. Wang, "Joint mobility tracking and handoff in cellular networks via sequential Monte Carlo filtering," *IEEE Trans. Signal Processing*, vol. 51, no. 1, pp. 269–281, 2003.