A Survey of Stochastic Simulation and Optimization Methods in Signal Processing

Marcelo Pereyra, *Member, IEEE*, Philip Schniter, *Fellow, IEEE*, Émilie Chouzenoux, *Member, IEEE*, Jean-Christophe Pesquet, *Fellow, IEEE*, Jean-Yves Tourneret, *Senior Member, IEEE*, Alfred O. Hero, III, *Fellow, IEEE*, and Steve McLaughlin, *Fellow, IEEE*

***Abstract*—Modern signal processing (SP) methods rely very heavily on probability and statistics to solve challenging SP problems. SP methods are now expected to deal with ever more complex models, requiring ever more sophisticated computational inference techniques. This has driven the development of statistical SP methods based on stochastic simulation and optimization. Stochastic simulation and optimization algorithms are computationally intensive tools for performing statistical inference in models that are analytically intractable and beyond the scope of deterministic inference methods. They have been recently successfully applied to many difficult problems involving complex statistical models and sophisticated (often Bayesian) statistical inference techniques. This survey paper offers an introduction to stochastic simulation and optimization methods in signal and image processing. The paper addresses a variety of high-dimensional Markov chain Monte Carlo (MCMC) methods as well as deterministic surrogate methods, such as variational Bayes, the Bethe approach, belief and expectation propagation and approximate message passing algorithms. It also discusses a range of optimization methods that have been adopted to solve stochastic problems, as well as stochastic methods for deterministic optimization. Subsequently, areas of overlap between simulation and optimization, in particular optimization-within-MCMC and MCMC-driven optimization are discussed.**

***Index Terms*—Bayesian inference, Markov chain Monte Carlo, proximal optimization algorithms, variational algorithms for approximate inference.**

Manuscript received April 30, 2015; revised August 29, 2015; accepted October 09, 2015. Date of publication November 02, 2015; date of current version February 11, 2016. This work was supported in part by the SuSTaIN program under EPSRC Grant EP/D063485/1 at the Department of Mathematics, University of Bristol., in part by the CNRS Imag'in project under Grant 2015OPTIMISME, in part by the HYPANEMA ANR Project under Grant ANR-12-BS03-003, in part by the BNPSI ANR Project ANR-13- BS-03-0006-01, and in part by the project ANR-11-L ABX-0040-CIMI as part of the program ANR-11-IDEX-0002-02 within the thematic trimester on image processing. The work of S. McLaughlin was supported by the EPSRC under Grant EP/J015180/1. The work of M. Pereyra was supported by a Marie Curie Intra-European Fellowship for Career Development. The work of P. Schniter was supported by the National Science Foundation under Grants CCF-1218754 and CCF-1018368. The work of A. Hero was supported by the Army Research Office under Grant W911NF-15-1-0479. The Editor-in-Chief coordinating the review of this manuscript and approving it for publication was Dr. Fernando Pereira.

M. Pereyra is with the School of Mathematics, University of Bristol, Bristol BS8 1TW, U.K. (e-mail: marcelo.pereyra@bristol.ac.uk).

P. Schniter is with the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210 USA (e-mail: schniter@ece.osu.edu).

É. Chouzenoux and J.-C. Pesquet are with the Laboratoire d'Informatique Gaspard Monge, Université Paris-Est, 77454 Marne la Vallée, France (e-mail: emilie.chouzenoux@univ-mlv.fr; pesquet@univ-mlv.fr).

J.-Y. Tourneret is with the INP-ENSEEIHT-IRIT-TeSA, University of Toulouse, 31071 Toulouse, France (e-mail: jean-yves.tourneret@enseeiht.fr).

A. O. Hero III is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109-2122 USA (e-mail: hero@eecs.umich.edu).

S. McLaughlin is with Engineering and Physical Sciences, Heriot Watt University, Edinburgh EH14 4AS, U.K. (e-mail: s.mclaughlin@hw.ac.uk).

Digital Object Identifier 10.1109/JSTSP.2015.2496908

I. Introduction

MODERN signal processing (SP) methods, (we use SP here to cover all relevant signal and image processing problems), rely very heavily on probabilistic and statistical tools; for example, they use stochastic models to represent the data observation process and the prior knowledge available, and they obtain solutions by performing statistical inference (e.g., using maximum likelihood or Bayesian strategies). Statistical SP methods are, in particular, routinely applied to many and varied tasks and signal modalities, ranging from resolution enhancement of medical images to hyperspectral image unmixing; from user rating prediction to change detection in social networks; and from source separation in music analysis to speech recognition.

However, the expectations and demands on the performance of such methods are constantly rising. SP methods are now expected to deal with challenging problems that require ever more complex models, and more importantly, ever more sophisticated computational inference techniques. This has driven the development of computationally intensive SP methods based on stochastic simulation and optimization. Stochastic simulation and optimization algorithms are computationally intensive tools for performing statistical inference in models that are analytically intractable and beyond the scope of deterministic inference methods. They have been recently successfully applied to many difficult SP problems involving complex statistical models and sophisticated (often Bayesian) statistical inference analyses. These problems can generally be formulated as inverse problems involving partially unknown observation processes and imprecise prior knowledge, for which they delivered accurate and insightful results. These stochastic algorithms are also closely related to the randomized, variational Bayes and message passing algorithms that are pushing forward the state of the art in approximate statistical inference for very large-scale problems. The key thread that makes stochastic simulation and optimization methods appropriate for these applications is the complexity and high dimensionality involved. For example in the case of hyperspectral imaging the data being processed can involve images

of 2048 by 1024 pixels across up to hundreds or thousands of wavelengths.

This survey paper offers an introduction to stochastic simulation and optimization methods in signal and image processing. The paper addresses a variety of high-dimensional Markov chain Monte Carlo (MCMC) methods as well as deterministic surrogate methods, such as variational Bayes, the Bethe approach, belief and expectation propagation and approximate message passing algorithms. It also discusses a range of stochastic optimization approaches. Subsequently, areas of overlap between simulation and optimization, in particular optimization-within-MCMC and MCMC-driven optimization are discussed. Some methods such as sequential Monte Carlo methods or methods based on importance sampling are not considered in this survey mainly due to space limitations.

This paper seeks to provide a survey of a variety of the algorithmic approaches in a tutorial fashion, as well as to highlight the state of the art, relationships between the methods, and potential future directions of research. In order to set the scene and inform our notation, consider an unknown random vector of interest $\boldsymbol{x} = [x_1, \ldots, x_N]^T$ and an observed data vector $\boldsymbol{y} = [y_1, \ldots, y_M]^T$, related to $\boldsymbol{x}$ by a statistical model with likelihood function $p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})$ potentially parametrized by a deterministic vector of parameters $\boldsymbol{\theta}$. Following a Bayesian inference approach, we model our prior knowledge about $\boldsymbol{x}$ with a prior distribution $p(\boldsymbol{x};\boldsymbol{\theta})$, and our knowledge about $\boldsymbol{x}$ after observing $\boldsymbol{y}$ with the posterior distribution

$$p(\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\theta}) = \frac{p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})p(\boldsymbol{x};\boldsymbol{\theta})}{Z(\boldsymbol{\theta})} \tag{1}$$

where the normalizing constant

$$Z(\boldsymbol{\theta}) = p(\boldsymbol{y};\boldsymbol{\theta}) = \int p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})p(\boldsymbol{x};\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{x} \tag{2}$$

is known as the "evidence", model likelihood, or the partition function. Although the integral in (2) suggests that all x_j are continuous random variables, we allow any random variable x_j to be either continuous or discrete, and replace the integral with a sum as required.

In many applications, we would like to evaluate the posterior $p(\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\theta})$ or some summary of it, for instance point estimates of $\boldsymbol{x}$ such as the conditional mean (i.e., MMSE estimate) $\mathrm{E}\{\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\theta}\}$, uncertainty reports such as the conditional variance $\mathrm{var}\{\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\theta}\}$, or expected log statistics as used in the expectation maximization (EM) algorithm [1]–[3]

$$\boldsymbol{\theta}^{(i+1)} = \arg\max_{\boldsymbol{\theta}} \mathrm{E}\{\ln p(\boldsymbol{x},\boldsymbol{y};\boldsymbol{\theta})|\boldsymbol{y};\boldsymbol{\theta}^{(i)}\} \tag{3}$$

where the expectation is taken with respect to $p(\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\theta}^{(i)})$. However, when the signal dimensionality N is large, the integral in (2), as well as those used in the posterior summaries, are often computationally intractable. Hence, the interest in computationally efficient alternatives. An alternative that has received a lot of attention in the statistical SP community is maximum-a-posteriori (MAP) estimation. Unlike other posterior summaries, MAP estimates can be computed by finding the value of $\boldsymbol{x}$ maximizing $p(\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\theta})$, which for many models is significantly more computationally tractable than numerical integration. In the sequel, we will suppress the dependence on $\boldsymbol{\theta}$ in the notation, since it is not of primary concern.

The paper is organized as follows. After this brief introduction where we have introduced the basic notation adopted, Section II discusses stochastic simulation methods, and in particular a variety of MCMC methods. In Section III we discuss deterministic surrogate methods, such as variational Bayes, the Bethe approach, belief and expectation propagation, and provide a brief summary of approximate message passing algorithms. Section IV discusses a range of optimization methods for solving stochastic problems, as well as stochastic methods for solving deterministic optimization problems. Subsequently, in Section V we discuss areas of overlap between simulation and optimization, in particular the use of optimization techniques within MCMC algorithms and MCMC-driven optimization, and suggest some interesting areas worthy of exploration. Finally, in Section VI we draw together thoughts, observations and conclusions.

II. Stochastic Simulation Methods

Stochastic simulation methods are sophisticated random number generators that allow samples to be drawn from a user-specified target density $\pi(\boldsymbol{x})$, such as the posterior $p(\boldsymbol{x}|\boldsymbol{y})$. These samples can then be used, for example, to approximate probabilities and expectations by Monte Carlo integration [4, Ch. 3]. In this section we will focus on MCMC methods, an important class of stochastic simulation techniques that operate by constructing a Markov chain with stationary distribution π. In particular, we concentrate on Metropolis-Hastings (MH) algorithms for high-dimensional models (see [5] for a more general recent review of MCMC methods). It is worth emphasizing, however, that we do not discuss many other important approaches to simulation that also arise often in signal processing applications, such as "particle filters" or sequential Monte Carlo methods [6], [7], and approximate Bayesian computation [8].

A cornerstone of MCMC methodology is the MH algorithm [4, Ch. 7], [9], [10], a universal machine for constructing Markov chains with stationary density π. Given some generic proposal distribution $\boldsymbol{x}^* \sim q(\cdot|\boldsymbol{x})$, the generic MH algorithm proceeds as follows.

Algorithm 1 Metropolis-Hastings algorithm (generic version)

Set an initial state $\boldsymbol{x}^{(0)}$
for $t = 1$ to T **do**
  Generate a candidate state $\boldsymbol{x}^*$ from a proposal $q(\cdot|\boldsymbol{x}^{(t-1)})$
  Compute the acceptance probability

$$\rho^{(t)} = \min\left(1, \frac{\pi(\boldsymbol{x}^*)}{\pi(\boldsymbol{x}^{(t-1)})}\frac{q(\boldsymbol{x}^{(t-1)}|\boldsymbol{x}^*)}{q(\boldsymbol{x}^*|\boldsymbol{x}^{(t-1)})}\right)$$

  Generate $u_t \sim \mathcal{U}(0,1)$
  if $u_t \leq \rho^{(t)}$ **then**
    Accept the candidate and set $\boldsymbol{x}^{(t)} = \boldsymbol{x}^*$
  else
    Reject the candidate and set $\boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t-1)}$
  end if
end for

Under mild conditions on q, the chains generated by Algo. 1 are ergodic and converge to the stationary distribution π [11], [12]. An important feature of this algorithm is that computing the acceptance probabilities $\rho^{(t)}$ does not require knowledge of the normalization constant of π (which is often not available in Bayesian inference problems). The intuition for the MH algorithm is that the algorithm proposes a stochastic perturbation to the state of the chain and applies a carefully defined decision rule to decide if this perturbation should be accepted or not. This decision rule, given by the random accept-reject step in Algo. 1, ensures that at equilibrium the samples of the Markov chain have π as marginal distribution.

The specific choice of q will of course determine the efficiency and the convergence properties of the algorithm. Ideally one should choose $q = \pi$ to obtain a perfect sampler (i.e., with candidates accepted with probability 1); this is of course not practically feasible since the objective is to avoid the complexity of directly simulating from π. In the remainder of this section we review strategies for specifying q for high-dimensional models, and discuss relative advantages and disadvantages. In order to compare and optimize the choice of q, a performance criterion needs to be chosen. A natural criterion is the stationary integrated autocorrelation time for some relevant scalar summary statistic $g : \mathbb{R}^N \to \mathbb{R}$, i.e.,

$$\tau_g = 1 + 2\sum_{t=1}^{\infty} \mathrm{Cor}\{g(\boldsymbol{x}^{(0)}), g(\boldsymbol{x}^{(t)})\} \tag{4}$$

with $\boldsymbol{x}^{(0)} \sim \pi$, and where $\mathrm{Cor}(\cdot,\cdot)$ denotes the correlation operator. This criterion is directly related to the effective number of independent Monte Carlo samples produced by the MH algorithm, and therefore to the mean squared error of the resulting Monte Carlo estimates. Unfortunately drawing conclusions directly from (4) is generally not possible because τ_g is highly dependent on the choice of g, with different choices often leading to contradictory results. Instead, MH algorithms are generally studied asymptotically in the infinite-dimensional model limit. More precisely, in the limit $N \to \infty$, the algorithms can be studied using diffusion process theory, where the dependence on g vanishes and all measures of efficiency become proportional to the diffusion speed. The "complexity" of the algorithms can then be defined as the rate at which efficiency deteriorates as $N \to \infty$, e.g., $\mathcal{O}(N)$ (see [13] for an introduction to this topic and details about the relationship between the efficiency of MH algorithms and their average acceptance probabilities or acceptance rates)[1].

Finally, it is worth mentioning that despite the generality of this approach, there are some specific models for which conventional MH sampling is not possible because the computation of $\rho^{(t)}$ is intractable (e.g., when π involves an intractable function of $\boldsymbol{x}$, such as the partition function of the Potts-Markov random field). This issue has received a lot of attention in the recent MCMC literature, and there are now several variations of the MH construction for intractable models [8], [14]–[16].

[1]Notice that this measure of complexity of MCMC algorithms does not take into account the computational costs related to generating candidate states and evaluating their Metropolis-Hastings acceptance probabilities, which typically also scale at least linearly with the problem dimension N.

A. Random Walk Metropolis-Hastings Algorithms

The so-called *random walk Metropolis-Hastings* (RWMH) algorithm is based on proposals of the form $\boldsymbol{x}^* = \boldsymbol{x}^{(t-1)} + \boldsymbol{w}$, where typically $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$ for some positive-definite covariance matrix $\boldsymbol{\Sigma}$ [4, Ch. 7.5]. This algorithm is one of the most widely used MCMC methods, perhaps because it has very robust convergence properties and a relatively low computational cost per iteration. It can be shown that the RWMH algorithm is geometrically ergodic under mild conditions on π [17]. Geometric ergodicity is important because it guarantees a central limit theorem for the chains, and therefore that the samples can be used for Monte Carlo integration. However, the myopic nature of the random walk proposal means that the algorithm often requires a large number of iterations to explore the parameter space, and will tend to generate samples that are highly correlated, particularly if the dimension N is large (the performance of this algorithm generally deteriorates at rate $\mathcal{O}(N)$, which is worse than other more advanced stochastic simulation MH algorithms [18]). This drawback can be partially mitigated by adjusting the matrix $\boldsymbol{\Sigma}$ to approximate the covariance structure of π, and some adaptive versions of RWHM perform this adaptation automatically. For sufficiently smooth target densities, performance is further optimized by scaling $\boldsymbol{\Sigma}$ to achieve an acceptance probability of approximately 20% – 25% [18].

B. Metropolis Adjusted Langevin Algorithms

The Metropolis adjusted Langevin algorithm (MALA) is an advanced MH algorithm inspired by the Langevin diffusion process on $\mathbb{R}^N$, defined as the solution to the stochastic differential equation [19]

$$dX(t) = \frac{1}{2}\nabla \log \pi\left(X(t)\right) dt + dW(t), \quad X(0) = \boldsymbol{x}_0 \tag{5}$$

where W is the Brownian motion process on $\mathbb{R}^N$ and $\boldsymbol{x}_0 \in \mathbb{R}^N$ denotes some initial condition. Under appropriate stability conditions, $X(t)$ converges in distribution to π as $t \to \infty$, and is therefore potentially interesting for drawing samples from π. Since direct simulation from $X(t)$ is only possible in very specific cases, we consider a discrete-time forward Euler approximation to (5) given by

$$X^{(t+1)} \sim \mathcal{N}\left(X^{(t)} + \frac{\delta}{2}\nabla \log \pi\left(X^{(t)}\right), \delta \mathbb{I}_N\right) \tag{6}$$

where the parameter δ controls the discrete-time increment. Under certain conditions on π and δ, (6) produces a good approximation of $X(t)$ and converges to a stationary density which is close to π. In MALA this approximation error is corrected by introducing an MH accept-reject step that guarantees convergence to the correct target density π. The resulting algorithm is equivalent to Algo. 1 above, with proposal

$$q(\boldsymbol{x}^*|\boldsymbol{x}^{(t-1)}) = \frac{1}{(2\pi\delta)^{N/2}} \exp\left(-\frac{\|\boldsymbol{x}^* - \boldsymbol{x}^{(t-1)} - \frac{\delta}{2}\nabla \log \pi\left(\boldsymbol{x}^{(t-1)}\right)\|^2}{2\delta}\right). \tag{7}$$

By analyzing the proposal (7) we notice that, in addition to the Langevin interpretation, MALA can also be interpreted as an MH algorithm that, at each iteration t, draws a candidate from

a local quadratic approximation to $\log \pi$ around $\boldsymbol{x}^{(t-1)}$, with $\delta^{-1}\mathbb{I}_N$ as an approximation to the Hessian matrix.

In addition, the MALA proposal can also be defined using a matrix-valued time step $\boldsymbol{\Sigma}$. This modification is related to redefining (6) in an Euclidean space with the inner product $\langle \boldsymbol{w}, \boldsymbol{\Sigma}^{-1}\boldsymbol{x}\rangle$ [20]. Again, the matrix $\boldsymbol{\Sigma}$ should capture the correlation structure of π to improve efficiency. For example, $\boldsymbol{\Sigma}$ can be the spectrally-positive version of the inverse Hessian matrix of $\log \pi$ [21], or the inverse Fisher information matrix of the statistical observation model [20]. Note that, in a similar fashion to preconditioning in optimization, using the exact full Hessian or Fisher information matrix is often too computationally expensive in high-dimensional settings and more efficient representations must be used instead. Alternatively, adaptive versions of MALA can learn a representation of the covariance structure online [22]. For sufficiently smooth target densities, MALA's performance can be further optimized by scaling $\boldsymbol{\Sigma}$ (or δ) to achieve an acceptance probability of approximately 50% – 60% [23].

Finally, there has been significant empirical evidence that MALA can be very efficient for some models, particularly in high-dimensional settings and when the cost of computing the gradient $\nabla \log \pi(\boldsymbol{x})$ is low. Theoretically, for sufficiently smooth π, the complexity of MALA scales at rate $\mathcal{O}(N^{1/3})$ [23], comparing very favorably to the $\mathcal{O}(N)$ rate of RWMH algorithms. However, the convergence properties of the conventional MALA are not as robust as those of the RWMH algorithm. In particular, MALA may fail to converge if the tails of π are super-Gaussian or heavy-tailed, or if δ is chosen too large [19]. Similarly, MALA might also perform poorly if π is not sufficiently smooth, or multi-modal. Improving MALA's convergence properties is an active research topic. Many limitations of the original MALA algorithm can now be avoided by using more advanced versions [20], [24]–[27].

C. Hamiltonian Monte Carlo

The Hamiltonian Monte Carlo (HMC) method is a very elegant and successful instance of an MH algorithm based on auxiliary variables [28]. Let $\boldsymbol{w} \in \mathbb{R}^N$, $\boldsymbol{\Sigma} \in \mathbb{R}^{N\times N}$ positive definite, and consider the augmented target density $\pi(\boldsymbol{x}, \boldsymbol{w}) \propto \pi(\boldsymbol{x}) \exp(-(1/2)\boldsymbol{w}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{w})$, which admits the desired target density $\pi(\boldsymbol{x})$ as marginal. The HMC method is based on the observation that the trajectories defined by the so-called *Hamiltonian dynamics* preserve the level sets of $\pi(\boldsymbol{x}, \boldsymbol{w})$. A point $(\boldsymbol{x}_0, \boldsymbol{w}_0) \in \mathbb{R}^{2N}$ that evolves according to the set of differential equations (8) during some simulation time period $(0, t]$

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} &= -\nabla_{\boldsymbol{w}} \log \pi(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{\Sigma}^{-1}\boldsymbol{w} \\ \frac{\mathrm{d}\boldsymbol{w}}{\mathrm{d}t} &= \nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{x}, \boldsymbol{w}) = \nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{x})\end{aligned} \tag{8}$$

yields a point $(\boldsymbol{x}_t, \boldsymbol{w}_t)$ that verifies $\pi(\boldsymbol{x}_t, \boldsymbol{w}_t) = \pi(\boldsymbol{x}_0, \boldsymbol{w}_0)$. In MCMC terms, the deterministic proposal (8) has $\pi(\boldsymbol{x}, \boldsymbol{w})$ as invariant distribution. Exploiting this property for stochastic simulation, the HMC algorithm combines (8) with a stochastic sampling step, $\boldsymbol{w} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$, that also has invariant distribution $\pi(\boldsymbol{x}, \boldsymbol{w})$, and that will produce an ergodic chain. Finally, as with the Langevin diffusion (5), it is generally not possible to solve the Hamiltonian (8) exactly. Instead, we use a *leap-frog* approximation detailed in [28]

$$\begin{aligned}\boldsymbol{w}^{(t+\delta/2)} &= \boldsymbol{w}^{(t)} + \frac{\delta}{2}\nabla_{\boldsymbol{x}} \log \pi\left(\boldsymbol{x}^{(t)}\right) \\ \boldsymbol{x}^{(t+\delta)} &= \boldsymbol{x}^{(t)} + \delta\boldsymbol{\Sigma}^{-1}\boldsymbol{w}^{(t+\delta/2)} \\ \boldsymbol{w}^{(t+\delta)} &= \boldsymbol{w}^{(t+\delta/2)} + \frac{\delta}{2}\nabla_{\boldsymbol{x}} \log \pi\left(\boldsymbol{x}^{(t+\delta)}\right)\end{aligned} \tag{9}$$

where again the parameter δ controls the discrete-time increment. The approximation error introduced by (9) is then corrected with an MH step targeting $\pi(\boldsymbol{x}, \boldsymbol{w})$. This algorithm is summarized in Algo. 2 below (see [28] for details about the derivation of the acceptance probability).

Algorithm 2 Hamiltonian Monte Carlo (with leap-frog)

Set an initial state $\boldsymbol{x}^{(0)}$, $\delta > 0$, and $L \in \mathbb{N}$.
for $t = 1$ to T' **do**
  Refresh the auxiliary variable $\boldsymbol{w} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$.
  Generate a candidate $(\boldsymbol{x}^*, \boldsymbol{w}^*)$ by propagating the current state $(\boldsymbol{x}^{(t-1)}, \boldsymbol{w})$ with L leap-frog steps of length δ defined in **(9)**.
  Compute the acceptance probability

$$\rho^{(t)} = \min\left(1, \frac{\pi(\boldsymbol{x}^*, \boldsymbol{w}^*)}{\pi(\boldsymbol{x}^{(t-1)}, \boldsymbol{w})}\right).$$

  Generate $u_t \sim \mathcal{U}(0, 1)$.
  if $u_t \leq \rho^{(t)}$ **then**
    Accept the candidate and set $\boldsymbol{x}^{(t)} = \boldsymbol{x}^*$.
  else
    Reject the candidate and set $\boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t-1)}$.
  end if
end for

Note that to obtain samples from the marginal $\pi(\boldsymbol{x})$ it is sufficient to project the augmented samples $(\boldsymbol{x}^{(t)}, \boldsymbol{w}^{(t)})$ onto the original space of $\boldsymbol{x}$ (i.e., by discarding the auxiliary variables $\boldsymbol{w}^{(t)}$). It is also worth mentioning that under some regularity condition on $\pi(\boldsymbol{x})$, the leap-frog approximation (9) is time-reversible and volume-preserving, and that these properties are key to the validity of the HMC algorithm [28].

Finally, there has been a large body of empirical evidence supporting HMC, particularly for high-dimensional models. Unfortunately, its theoretical convergence properties are much less well understood [29]. It has been recently established that for certain target densities the complexity of HMC scales at rate $\mathcal{O}(N^{1/4})$, comparing favorably with MALA's rate $\mathcal{O}(N^{1/3})$ [30]. However, it has also been observed that, as with MALA, HMC may fail to converge if the tails of π are super-Gaussian or heavy-tailed, or if δ is chosen too large. HMC may also perform poorly if π is multi-modal, or not sufficiently smooth.

Of course, the practical performance of HMC also depends strongly on the algorithm parameters $\boldsymbol{\Sigma}$, L and δ [29]. The covariance matrix $\boldsymbol{\Sigma}$ should be designed to model the correlation structure of $\pi(\boldsymbol{x})$, which can be determined by performing pilot runs, or alternatively by using the strategies described in [20], [21], [31]. The parameters δ and L should be adjusted to obtain

an average acceptance probability of approximately 60% – 70% [30]. Again, this can be achieved by performing pilot runs, or by using an adaptive HMC algorithm that adjusts these parameters automatically [32], [33].

D. Gibbs Sampling

The Gibbs sampler (GS) is another very widely used MH algorithm which operates by updating the elements of $\boldsymbol{x}$ individually, or by groups, using the appropriate conditional distributions [4, Ch. 10]. This divide-and-conquer strategy often leads to important efficiency gains, particularly if the conditional densities involved are "simple", in the sense that it is computationally straightforward to draw samples from them. For illustration, suppose that we split the elements of $\boldsymbol{x}$ in three groups $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3)$, and that by doing so we obtain three conditional densities $\pi(\boldsymbol{x}_1|\boldsymbol{x}_2, \boldsymbol{x}_3)$, $\pi(\boldsymbol{x}_2|\boldsymbol{x}_1, \boldsymbol{x}_3)$, and $\pi(\boldsymbol{x}_3|\boldsymbol{x}_1, \boldsymbol{x}_2)$ that are "simple" to sample. Using this decomposition, the GS targeting π proceeds as in Algo. 3. The Markov kernel resulting from concatenating the component-wise kernels admits $\pi(\boldsymbol{x})$ as joint invariant distribution, and thus the chain produced by Algo. 3 has the desired target density (see [4, Ch. 10] for a review of the theory behind this algorithm). This fundamental property holds even if the simulation from the conditionals is done by using other MCMC algorithms (e.g., RWMH, MALA or HMC steps targeting the conditional densities), though this may result in a deterioration of the algorithm convergence rate. Similarly, the property also holds if the frequency and order of the updates is scheduled randomly and adaptively to improve the overall performance.

Algorithm 3 Gibbs sampler algorithm

Set an initial state $\boldsymbol{x}^{(0)} = (\boldsymbol{x}_1^{(0)}, \boldsymbol{x}_2^{(0)}, \boldsymbol{x}_3^{(0)})$
for $t = 1$ to T **do**
  Generate $\boldsymbol{x}_1^{(t)} \sim \pi\left(\boldsymbol{x}_1|\boldsymbol{x}_2^{(t-1)}, \boldsymbol{x}_3^{(t-1)}\right)$
  Generate $\boldsymbol{x}_2^{(t)} \sim \pi\left(\boldsymbol{x}_2|\boldsymbol{x}_1^{(t)}, \boldsymbol{x}_3^{(t-1)}\right)$
  Generate $\boldsymbol{x}_3^{(t)} \sim \pi\left(\boldsymbol{x}_3|\boldsymbol{x}_1^{(t)}, \boldsymbol{x}_2^{(t)}\right)$
end for

As with other MH algorithms, the performance of the GS depends on the correlation structure of π. Efficient samplers seek to update simultaneously the elements of $\boldsymbol{x}$ that are highly correlated with each other, and to update "slow-moving" elements more frequently. The structure of π can be determined by pilot runs, or alternatively by using an adaptive GS that learns it online and that adapts the updating schedule accordingly as described in [34]. However, updating elements in parallel often involves simulating from more complicated conditional distributions, and thus introduces a computational overhead. Finally, it is worth noting that the GS is very useful for SP models, which typically have sparse conditional independence structures (e.g., Markovian properties) and conjugate priors and hyper-priors from the exponential family. This often leads to simple one-dimensional conditional distributions that can be updated in parallel by groups [16], [35].

E. Partially Collapsed Gibbs Sampling

The partially collapsed Gibbs sampler (PCGS) is a recent development in MCMC theory that seeks to address some of the limitations of the conventional GS [36]. As mentioned previously, the GS performs poorly if strongly correlated elements of $\boldsymbol{x}$ are updated separately, as this leads to chains that are highly correlated and to an inefficient exploration of the parameter space. However, updating several elements together can also be computationally expensive, particularly if it requires simulating from difficult conditional distributions. In collapsed samplers, this drawback is addressed by carefully replacing one or several conditional densities by *partially collapsed*, or marginalized conditional distributions.

For illustration, suppose that in our previous example the subvectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ exhibit strong dependencies, and that as a result the GS of Algo. 3 performs poorly. Assume that we are able to draw samples from the marginalized conditional density $\pi(\boldsymbol{x}_1|\boldsymbol{x}_3) = \int \pi(\boldsymbol{x}_1, \boldsymbol{x}_2|\boldsymbol{x}_3)\mathrm{d}\boldsymbol{x}_2$, which does not depend on $\boldsymbol{x}_2$. This leads to the PCGS described in Algo. 4 to sample from π, which "partially collapses" Algo. 3 by replacing $\pi(\boldsymbol{x}_1|\boldsymbol{x}_2, \boldsymbol{x}_3)$ with $\pi(\boldsymbol{x}_1|\boldsymbol{x}_3)$.

Algorithm 4 Partially collapsed Gibbs sampler

Set an initial state $\boldsymbol{x}^{(0)} = (\boldsymbol{x}_1^{(0)}, \boldsymbol{x}_2^{(0)}, \boldsymbol{x}_3^{(0)})$
for $t = 1$ to T' **do**
  Generate $\boldsymbol{x}_1^{(t)} \sim \pi\left(\boldsymbol{x}_1|\boldsymbol{x}_3^{(t-1)}\right)$
  Generate $\boldsymbol{x}_2^{(t)} \sim \pi\left(\boldsymbol{x}_2|\boldsymbol{x}_1^{(t)}, \boldsymbol{x}_3^{(t-1)}\right)$
  Generate $\boldsymbol{x}_3^{(t)} \sim \pi\left(\boldsymbol{x}_3|\boldsymbol{x}_1^{(t)}, \boldsymbol{x}_2^{(t)}\right)$
end for

Van Dyk and Park [36] established that the PCGS is always at least as efficient as the conventional GS, and it has been observed that the PCGS is remarkably efficient for some statistical models [37], [38]. Unfortunately, PCGSs are not as widely applicable as GSs because they require simulating exactly from the partially collapsed conditional distributions. In general, using MCMC simulation (e.g., MH steps) within a PCGS will lead to an incorrect MCMC algorithm [39]. Similarly, altering the order of the updates (e.g., by permuting the simulations of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ in Algo. 4) may also alter the target density [36].

III. Surrogates for Stochastic Simulation

A. Variational Bayes

In the *variational Bayes* (VB) approach described in [40], [41], the true posterior $p(\boldsymbol{x}|\boldsymbol{y})$ is approximated by a density $q_\star(\boldsymbol{x}) \in \mathcal{Q}$, where $\mathcal{Q}$ is a subset of valid densities on $\boldsymbol{x}$. In particular,

$$q_\star(\boldsymbol{x}) = \arg\min_{q\in\mathcal{Q}} D\big(q(\boldsymbol{x})\big\|p(\boldsymbol{x}|\boldsymbol{y})\big) \tag{10}$$

where $D(q\|p)$ denotes the Kullback-Leibler (KL) divergence between p and q. As a result of the optimization in (10) over a function, this is termed "variational Bayes" because of the relation to the calculus of variations [42]. Recalling that $D(q\|p)$ reaches its minimum value of zero if and only if $p = q$ [43],

we see that $q_\star(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y})$ when $\mathcal{Q}$ includes all valid densities on $\boldsymbol{x}$. However, the premise is that $p(\boldsymbol{x}|\boldsymbol{y})$ is too difficult to work with, and so $\mathcal{Q}$ is chosen as a balance between fidelity and tractability.

Note that the use of $D(q\|p)$, rather than $D(p\|q)$, implies a search for a $q_\star$ that agrees with the true posterior $p(\boldsymbol{x}|\boldsymbol{y})$ over the set of $\boldsymbol{x}$ where $p(\boldsymbol{x}|\boldsymbol{y})$ is large. We will revisit this choice when discussing expectation propagation in Section III-E.

Rather than working with the KL divergence directly, it is common to decompose it as follows

$$D\big(q(\boldsymbol{x})\big\|p(\boldsymbol{x}|\boldsymbol{y})\big) = \int q(\boldsymbol{x}) \ln \frac{q(\boldsymbol{x})}{p(\boldsymbol{x}|\boldsymbol{y})}\,\mathrm{d}\boldsymbol{x} = \ln Z + F(q) \quad (11)$$

where

$$F(q) \triangleq \int q(\boldsymbol{x}) \ln \frac{q(\boldsymbol{x})}{p(\boldsymbol{x},\boldsymbol{y})}\,\mathrm{d}\boldsymbol{x} \quad (12)$$

is known as the *Gibbs free energy* or *variational free energy*. Rearranging (11), we see that

$$-\ln Z = F(q) - D\big(q(\boldsymbol{x})\big\|p(\boldsymbol{x}|\boldsymbol{y})\big) \le F(q) \quad (13)$$

as a consequence of $D\big(q(\boldsymbol{x})\big\|p(\boldsymbol{x}|\boldsymbol{y})\big) \ge 0$. Thus, $F(q)$ can be interpreted as an upper bound on the negative log partition. Also, because $\ln Z$ is invariant to q, the optimization (10) can be rewritten as

$$q_\star(\boldsymbol{x}) = \arg\min_{q\in\mathcal{Q}} F(q), \quad (14)$$

which avoids the difficult integral in (2). In the sequel, we will discuss several strategies to solve the variational optimization problem (14).

B. The Mean-Field Approximation

A common choice of $\mathcal{Q}$ is the set of fully factorizable densities, resulting in the *mean-field* approximation [44], [45]

$$q(\boldsymbol{x}) = \prod_{j=1}^{N} q_j(x_j). \quad (15)$$

Substituting (15) into (12) yields the mean-field free energy

$$F_{\mathsf{MF}}(q) \triangleq \int \left[\prod_j q_j(x_j)\right] \ln \frac{1}{p(\boldsymbol{x},\boldsymbol{y})}\,\mathrm{d}\boldsymbol{x} - \sum_{j=1}^{N} h(q_j) \quad (16)$$

where $h(q_j) \triangleq -\int q_j(x_j)\ln q_j(x_j)\,\mathrm{d}x_j$ denotes the differential entropy. Furthermore, for $j = 1,\dots,N$, (16) can be written as

$$F_{\mathsf{MF}}(q) = D\big(q_j(x_j)\big\|g_j(x_j,\boldsymbol{y})\big) - \sum_{i\neq j} h(q_i) \quad (17)$$

$$g_j(x_j,\boldsymbol{y}) \triangleq \exp\int \Big[\prod_{i\neq j} q_i(x_i)\Big] \ln p(\boldsymbol{x},\boldsymbol{y})\,\mathrm{d}\boldsymbol{x}_{\setminus j} \quad (18)$$

where $\boldsymbol{x}_{\setminus j} \triangleq [x_1,\dots,x_{j-1},x_j,\dots,x_N]^T$ for $j = 1,\dots,N$. Equation (17) implies the optimality condition

$$q_{j,\star}(x_j) = \frac{g_{j,\star}(x_j,\boldsymbol{y})}{\int g_{j,\star}(x'_j,\boldsymbol{y})\,\mathrm{d}x'_j} \quad \forall j = 1,\dots,N \quad (19)$$

where $q_\star(\boldsymbol{x}) = \prod_{j=1}^N q_{j,\star}(x_j)$ and where $g_{j,\star}(x_j,\boldsymbol{y})$ is defined as in (18) but with $q_{i,\star}(x_i)$ in place of $q_i(x_i)$. Equation (19) suggests an iterative coordinate-ascent algorithm: update each component $q_j(x_j)$ of $q(\boldsymbol{x})$ while holding the others fixed. But this requires solving the integral in (18). A solution arises when $\forall j$ the conditionals $p(x_j,\boldsymbol{y}|\boldsymbol{x}_{\setminus j})$ belong to the same exponential family of distributions [46], i.e.,

$$p(x_j,\boldsymbol{y}\,|\,\boldsymbol{x}_{\setminus j}) \propto h(x_j)\exp\big(\boldsymbol{\eta}(\boldsymbol{x}_{\setminus j},\boldsymbol{y})^T\boldsymbol{t}(x_j)\big) \quad (20)$$

where the sufficient statistic $\boldsymbol{t}(x_j)$ parameterizes the family. The exponential family encompasses a broad range of distributions, notably jointly Gaussian and multinomial $p(\boldsymbol{x},\boldsymbol{y})$. Plugging $p(\boldsymbol{x},\boldsymbol{y}) = p(x_j,\boldsymbol{y}\,|\,\boldsymbol{x}_{\setminus j})p(\boldsymbol{x}_{\setminus j},\boldsymbol{y})$ and (20) into (18) immediately gives

$$g_j(x_j,\boldsymbol{y}) \propto h(x_j)\exp\big(\mathrm{E}\{\boldsymbol{\eta}(\boldsymbol{x}_{\setminus j},\boldsymbol{y})\}^T\boldsymbol{t}(x_j)\big) \quad (21)$$

where the expectation is taken over $\boldsymbol{x}_{\setminus j} \sim \prod_{i\neq j} q_{i,\star}(x_i)$. Thus, if each q_j is chosen from the same family, i.e., $q_j(x_j) \propto h(x_j)\exp\big(\boldsymbol{\gamma}_j^T\boldsymbol{t}(x_j)\big)\ \forall j$, then (19) reduces to the moment-matching condition

$$\boldsymbol{\gamma}_{j,\star} = \mathrm{E}\big\{\boldsymbol{\eta}(\boldsymbol{x}_{\setminus j},\boldsymbol{y})\big\} \quad (22)$$

where $\boldsymbol{\gamma}_{j,\star}$ is the optimal value of $\boldsymbol{\gamma}_j$.

C. The Bethe Approach

In many cases, the fully factored model (15) yields too gross of an approximation. As an alternative, one might try to fit a model $q(\boldsymbol{x})$ that has a similar dependency structure as $p(\boldsymbol{x}|\boldsymbol{y})$. In the sequel, we assume that the true posterior factors as

$$p(\boldsymbol{x}|\boldsymbol{y}) = Z^{-1}\prod_\alpha f_\alpha(\boldsymbol{x}_\alpha) \quad (23)$$

where $\boldsymbol{x}_\alpha$ are subvectors of $\boldsymbol{x}$ (sometimes called *cliques* or *outer clusters*) and f_α are non-negative potential functions. Note that the factorization (23) defines a Gibbs random field when $p(\boldsymbol{x}|\boldsymbol{y}) > 0$. When a collection of variables $\{x_n\}$ always appears together in a factor, we can collect them into $\boldsymbol{x}_\beta$, an *inner cluster*, although it is not necessary to do so. For simplicity we will assume that these $\boldsymbol{x}_\beta$ are non-overlapping (i.e., $\boldsymbol{x}_\beta \cap \boldsymbol{x}_{\beta'} = 0\ \forall\beta \neq \beta'$), so that $\{\boldsymbol{x}_\beta\}$ represents a partition of $\boldsymbol{x}$. The factorization (23) can then be drawn as a *factor graph* to help visualize the structure of the posterior, as in Fig. 1.

We now seek a tractable way to build an approximation $q(\boldsymbol{x})$ with the same dependency structure as (23). But rather than designing $q(\boldsymbol{x})$ as a whole, we design the cluster marginals, $\{q_\alpha(\boldsymbol{x}_\alpha)\}$ and $\{q_\beta(\boldsymbol{x}_\beta)\}$, which must be non-negative, normalized, and consistent

$$0 \le q_\alpha(\boldsymbol{x}_\alpha),\ \ 0 \le q_\beta(\boldsymbol{x}_\beta)\ \ \forall\alpha,\beta,\boldsymbol{x}_\alpha,\boldsymbol{x}_\beta \quad (24)$$

$$1 = \int q_\alpha(\boldsymbol{x}_\alpha)\,\mathrm{d}\boldsymbol{x}_\alpha\ \ 1 = \int q_\beta(\boldsymbol{x}_\beta)\,\mathrm{d}\boldsymbol{x}_\beta\ \ \forall\alpha,\beta \quad (25)$$

$$q_\beta(\boldsymbol{x}_\beta) = \int q_\alpha(\boldsymbol{x}_\alpha)\,\mathrm{d}\boldsymbol{x}_{\alpha\setminus\beta}\ \ \forall\alpha,\beta\in\mathfrak{N}_\alpha,\boldsymbol{x}_\beta \quad (26)$$

where $\boldsymbol{x}_{\alpha\setminus\beta}$ gathers the components of $\boldsymbol{x}$ that are contained in the cluster α and not in the cluster β, and $\mathfrak{N}_\alpha$ denotes the neigh-

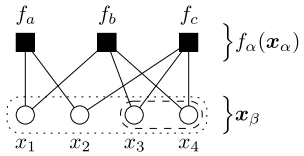

Fig. 1. An example of a factor graph, which is a bipartite graph consisting of variable nodes, (circles/ovals), and factor nodes, (boxes). In this example, $\boldsymbol{x}_a = \{x_1, x_2\}$, $\boldsymbol{x}_b = \{x_1, x_3, x_4\}$, $\boldsymbol{x}_c = \{x_2, x_3, x_4\}$. There are several choices for the inner clusters $\boldsymbol{x}_\beta$. One is the full factorization $\boldsymbol{x}_1 = x_1$, $\boldsymbol{x}_2 = x_2$, $\boldsymbol{x}_3 = x_3$, and $\boldsymbol{x}_4 = x_4$. Another is the partial factorization $\boldsymbol{x}_1 = x_1$, $\boldsymbol{x}_2 = x_2$, and $\boldsymbol{x}_3 = \{x_3, x_4\}$, which results in the "super node" in the dashed oval. Another is no factorization: $\boldsymbol{x}_1 = \{x_1, x_2, x_3, x_4\}$, resulting in the "super node" in the dotted oval. In the latter case, we redefine each factor f_α to have the full domain $\boldsymbol{x}$ (with trivial dependencies where needed).

borhood of the factor α (i.e., the set of inner clusters β connected to α).

In general, it is difficult to specify $q(\boldsymbol{x})$ from its cluster marginals. However, in the special case that the factor graph has a tree structure (i.e., there is at most one path from one node in the graph to another), we have [47]

$$q(\boldsymbol{x}) = \frac{\prod_\alpha q_\alpha(\boldsymbol{x}_\alpha)}{\prod_\beta q_\beta(\boldsymbol{x}_\beta)^{N_\beta - 1}} \tag{27}$$

where $N_\beta = |\mathfrak{N}_\beta|$ is the neighborhood size of the cluster β. In this case, the free energy (12) simplifies to

$$F(q) = \underbrace{\sum_\alpha D\big(q_\alpha \big\| f_\alpha\big) + \sum_\beta (N_\beta - 1) h(q_\beta)}_{\triangleq F_{\mathsf{B}}\left(\{q_\alpha\}, \{q_\beta\}\right)} + \text{const}, \tag{28}$$

where F_{B} is known as the *Bethe free energy* (BFE) [47].

Clearly, if the true posterior $\{f_\alpha\}$ has a tree structure, and no constraints beyond (24)–(26) are placed on the cluster marginals $\{q_\alpha\}, \{q_\beta\}$, then minimization of $F_{\mathsf{B}}\big(\{q_\alpha\}, \{q_\beta\}\big)$ will recover the cluster marginals of the true posterior. But even when $\{f_\alpha\}$ is not a tree, $F_{\mathsf{B}}\big(\{q_\alpha\}, \{q_\beta\}\big)$ can be used as an *approximation* of the Gibbs free energy $F(q)$, and minimizing F_{B} can be interpreted as designing a q that *locally* matches the true posterior.

The remaining question is how to efficiently minimize $F_{\mathsf{B}}\big(\{q_\alpha\}, \{q_\beta\}\big)$ subject to the (linear) constraints (24)–(26). Complicating matters is the fact that $F_{\mathsf{B}}\big(\{q_\alpha\}, \{q_\beta\}\big)$ is the sum of convex KL divergences and concave entropies. One option is direct minimization using a "double loop" approach like the concave-convex procedure (CCCP) [48], where the outer loop linearizes the concave term about the current estimate and the inner loop solves the resulting convex optimization problem (typically with an iterative technique). Another option is belief propagation, which is described below.

D. Belief Propagation

Belief propagation (BP) [49], [50] is an algorithm for computing (or approximating) marginal probability density functions (pdfs)[2] like $q_\beta(\boldsymbol{x}_\beta)$ and $q_\alpha(\boldsymbol{x}_\alpha)$ by propagating messages on a factor graph. The standard form of BP is given by the sum-product algorithm (SPA) [51], which computes the following messages from each factor node f_α to each variable (super) node $\boldsymbol{x}_\beta$ and vice versa

$$m_{\alpha\to\beta}(\boldsymbol{x}_\beta) \leftarrow \int f_\alpha(\boldsymbol{x}_\alpha) \prod_{\beta' \in \mathfrak{N}_\alpha \backslash \beta} m_{\beta'\to\alpha}(\boldsymbol{x}_{\beta'})\, \mathrm{d}\boldsymbol{x}_{\alpha\backslash\beta} \tag{29}$$

$$m_{\beta\to\alpha}(\boldsymbol{x}_\beta) \leftarrow \prod_{\alpha' \in \mathfrak{N}_\beta \backslash \alpha} m_{\alpha'\to\beta}(\boldsymbol{x}_\beta). \tag{30}$$

These messages are then used to compute the beliefs

$$q_\beta(\boldsymbol{x}_\beta) \propto \prod_{\alpha \in \mathfrak{N}_\beta} m_{\alpha\to\beta}(\boldsymbol{x}_\beta) \tag{31}$$

$$q_\alpha(\boldsymbol{x}_\alpha) \propto f_\alpha(\boldsymbol{x}_\alpha) \prod_{\beta \in \mathfrak{N}_\alpha} m_{\beta\to\alpha}(\boldsymbol{x}_\beta) \tag{32}$$

which must be normalized in accordance with (25). The messages (29)-(30) do not need to be normalized, although it is often done in practice to prevent numerical overflow.

When the factor graph $\{f_\alpha\}$ has a tree structure, the BP-computed marginals coincide with the true marginals after one round of forward and backward message passes. Thus, BP on a tree-graph is sometimes referred to as the *forward-backward* algorithm, particularly in the context of hidden Markov models [52]. In the tree case, BP is akin to a dynamic programming algorithm that organizes the computations needed for marginal evaluation in a tractable manner.

When the factor graph $\{f_\alpha\}$ has cycles or "loops," BP can still be applied by iterating the message computations (29)-(30) until convergence (not guaranteed), which is known as *loopy BP* (LBP). However, the corresponding beliefs (31)-(32) are in general only approximations of the true marginals. This suboptimality is expected because exact marginal computation on a loopy graph is an NP-hard problem [53]. Still, the answers computed by LBP are in many cases very accurate [54]. For example, LBP methods have been successfully applied to communication and SP problems such as: turbo decoding [55], LDPC decoding [49], [56], inference on Markov random fields [57], multiuser detection [58], and compressive sensing [59], [60].

Although the excellent performance of LBP was at first a mystery, it was later established that LBP minimizes the constrained BFE. More precisely, the fixed points of LBP coincide with the stationary points of $F_{\mathsf{B}}\big(\{q_\alpha\}, \{q_\beta\}\big)$ from (28) under the constraints (24)-(26) [47]. The link between LBP and BFE can be established through the Lagrangian formalism, which converts constrained BFE minimization to an unconstrained minimization through the incorporation of additional variables known as Lagrange multipliers [61]. By setting the derivatives of the Lagrangian to zero, one obtains a set of equations that are equivalent to the message updates (29)-(30) [47]. In particular, the stationary-point versions of the Lagrange multipliers equal the fixed-point versions of the loopy SPA log-messages.

Note that, unlike the mean-field approach (15), the cluster-based nature of LBP does not facilitate an explicit description of the joint-posterior approximation $q \in \mathcal{Q}$ from (10). The reason is that, when the factor graph is loopy, there is no straightforward relationship between the joint posterior $q(\boldsymbol{x})$ and the cluster marginals $\{q_\alpha(\boldsymbol{x}_\alpha)\}, \{q_\beta(\boldsymbol{x}_\beta)\}$, as explained

[2]Note that another form of BP exists to compute the maximum a posteriori (MAP) estimate $\arg\max_{\boldsymbol{x}} p(\boldsymbol{x}|\boldsymbol{y})$ known as the "max-product" or "min-sum" algorithm [50]. However, this approach does not address the problem of computing surrogates for stochastic methods, and so is not discussed further.

before (27). Instead, it is better to interpret LBP as an efficient implementation of the Bethe approach from Section III-C, which aims for a local approximation of the true posterior.

In summary, by constructing a factor graph with low-dimensional $\{\boldsymbol{x}_\beta\}$ and applying BP or LBP, we trade the high-dimensional integral $q_\beta(\boldsymbol{x}_\beta) = \int p(\boldsymbol{x}|\boldsymbol{y})\,\mathrm{d}\boldsymbol{x}_{\backslash\beta}$ for a sequence of low-dimensional message computations (29)-(30). But (29)-(30) are themselves tractable only for a few families of $\{f_\alpha\}$. Typically, $\{f_\alpha\}$ are limited to members of the exponential family closed under marginalization (see [62]), so that the updates of the message pdfs (29)-(30) reduce to updates of the natural parameters (i.e., $\boldsymbol{\eta}$ in (20)). The two most common instances are multivariate Gaussian pdfs and multinomial probability mass functions (pmfs). For both of these cases, when LBP converges, it tends to be much faster than double-loop algorithms like CCCP (see, e.g., [63]). However, LBP does not always converge [54].

E. Expectation Propagation

Expectation propagation (EP) [64] (see also the overviews [62], [65]) is an iterative method of approximate inference that is reminiscent of LBP but has much more flexibility with regards to the modeling distributions. In EP, the true posterior $p(\boldsymbol{x}|\boldsymbol{y})$, which is assumed to factorize as in (23), is approximated by $q(\boldsymbol{x})$ such that

$$q(\boldsymbol{x}) \propto \prod_\alpha m_\alpha(\boldsymbol{x}_\alpha) \tag{33}$$

where $\boldsymbol{x}_\alpha$ are the same as in (23) and m_α are referred to as "site approximations." Although no constraints are imposed on the true-posterior factors $\{f_\alpha\}$, the approximation $q(\boldsymbol{x})$ is restricted to a factorized exponential family. In particular,

$$q(\boldsymbol{x}) = \prod_\beta q_\beta(\boldsymbol{x}_\beta) \tag{34}$$

$$q_\beta(\boldsymbol{x}_\beta) = \exp\big(\boldsymbol{\gamma}_\beta^T \boldsymbol{t}_\beta(\boldsymbol{x}_\beta) - c_\beta(\boldsymbol{\gamma}_\beta)\big),\ \forall\beta, \tag{35}$$

with some given base measure. We note that our description of EP applies to arbitrary partitions $\{\boldsymbol{x}_\beta\}$, from the trivial partition $\boldsymbol{x}_\beta = \boldsymbol{x}$ to the full partition $\boldsymbol{x}_\beta = x_\beta$.

The EP algorithm iterates the following updates over all factors α until convergence (not guaranteed)

$$q^{\backslash\alpha}(\boldsymbol{x}) \leftarrow \frac{q(\boldsymbol{x})}{m_\alpha(\boldsymbol{x}_\alpha)} \tag{36}$$

$$\widehat{q}^{\backslash\alpha}(\boldsymbol{x}) \leftarrow q^{\backslash\alpha}(\boldsymbol{x}) f_\alpha(\boldsymbol{x}_\alpha) \tag{37}$$

$$q^{\mathsf{new}}(\boldsymbol{x}) \leftarrow \arg\min_{q\in\mathcal{Q}} D\big(\widehat{q}^{\backslash\alpha}(\boldsymbol{x})\big\|q(\boldsymbol{x})\big) \tag{38}$$

$$m_\alpha^{\mathsf{new}}(\boldsymbol{x}_\alpha) \leftarrow \frac{q^{\mathsf{new}}(\boldsymbol{x})}{q^{\backslash\alpha}(\boldsymbol{x})} \tag{39}$$

$$q(\boldsymbol{x}) \leftarrow q^{\mathsf{new}}(\boldsymbol{x}) \tag{40}$$

$$m_\alpha(\boldsymbol{x}_\alpha) \leftarrow m_\alpha^{\mathsf{new}}(\boldsymbol{x}_\alpha) \tag{41}$$

where $\mathcal{Q}$ in (38) refers to the set of $q(\boldsymbol{x})$ obeying (34)-(35). Essentially, (36) removes the αth site approximation m_α from the posterior model q, and (37) replaces it with the true factor f_α. Here, $q^{\backslash\alpha}$ is known as the "cavity" distribution. The quantity $\widehat{q}^{\backslash\alpha}$ is then projected onto the exponential family in (38). The site approximation is then updated in (39), and the old quantities are overwritten in (40)-(41). Note that the right side of (39) depends only on $\boldsymbol{x}_\alpha$ because $q^{\mathsf{new}}(\boldsymbol{x})$ and $q^{\backslash\alpha}(\boldsymbol{x})$ differ only over $\boldsymbol{x}_\alpha$. Note also that the KL divergence in (38) is reversed relative to (10).

The EP updates (37)–(41) can be simplified by leveraging the factorized exponential family structure in (34)-(35). First, for (33) to be consistent with (34)-(35), each site approximation must factor into exponential-family terms, i.e.,

$$m_\alpha(\boldsymbol{x}_\alpha) = \prod_{\beta\in\mathfrak{N}_\alpha} m_{\alpha,\beta}(\boldsymbol{x}_\alpha) \tag{42}$$

$$m_{\alpha,\beta}(\boldsymbol{x}_\alpha) = \exp\big(\boldsymbol{\mu}_{\alpha,\beta}^T \boldsymbol{t}_\beta(\boldsymbol{x}_\beta)\big). \tag{43}$$

It can then be shown [62] that (36)–(38) reduce to

$$\boldsymbol{\gamma}_\beta^{\mathsf{new}} \leftarrow \arg\max_{\boldsymbol{\gamma}} \Big[\boldsymbol{\gamma}^T \mathrm{E}_{\widehat{q}^{\backslash\alpha}}\{\boldsymbol{t}_\beta(\boldsymbol{x}_\beta)\} - c_\beta(\boldsymbol{\gamma})\Big] \tag{44}$$

for all $\beta \in \mathfrak{N}_\alpha$, which can be interpreted as the moment matching condition $\mathrm{E}_{q^{\mathsf{new}}}\{\boldsymbol{t}_\beta(\boldsymbol{x}_\beta)\} = \mathrm{E}_{\widehat{q}^{\backslash\alpha}}\{\boldsymbol{t}_\beta(\boldsymbol{x}_\beta)\}$. Furthermore, (39) reduces to

$$\boldsymbol{\mu}_{\alpha,\beta}^{\mathsf{new}} \leftarrow \boldsymbol{\gamma}_\beta^{\mathsf{new}} - \sum_{\alpha'\in\mathfrak{N}_\beta\backslash\alpha} \boldsymbol{\mu}_{\alpha',\beta} \tag{45}$$

for all $\beta \in \mathfrak{N}_\alpha$. Finally, (40) and (41) reduce to $\boldsymbol{\gamma}_\beta \leftarrow \boldsymbol{\gamma}_\beta^{\mathsf{new}}$ and $\boldsymbol{\mu}_{\alpha,\beta} \leftarrow \boldsymbol{\mu}_{\alpha,\beta}^{\mathsf{new}}$, respectively, for all $\beta \in \mathfrak{N}_\alpha$.

Interestingly, in the case that the true factors $\{f_\alpha\}$ are members of an exponential family closed under marginalization, the version of EP described above is equivalent to the SPA up to a change in message schedule. In particular, for each given factor node α, the SPA updates the outgoing message towards one variable node β per iteration, whereas EP simultaneously updates the outgoing messages in all directions, resulting in m_α^{new} (see, e.g., [41]). By restricting the optimization in (39) to a single factor q_β^{new}, EP can be made equivalent to the SPA. On the other hand, for generic factors $\{f_\alpha\}$, EP can be viewed as a tractable approximation of the (intractable) SPA.

Although the above form of EP iterates serially through the factor nodes α, it is also possible to perform the updates in parallel, resulting in what is known as the *expectation consistent* (EC) approximation algorithm [66].

EP and EC have an interesting BFE interpretation. Whereas the fixed points of LBP coincide with the stationary points of the BFE (28) subject to (24)-(25) and *strong* consistency (26), the fixed points of EP and EC coincide with the stationary points of the BFE (28) subject to (24)-(25) and the *weak* consistency (i.e., moment-matching) constraint [67]

$$\mathrm{E}_{\widehat{q}^{\backslash\alpha}}\{\boldsymbol{t}(\boldsymbol{x}_\beta)\} = \mathrm{E}_{q_\beta}\{\boldsymbol{t}(\boldsymbol{x}_\beta)\}, \quad \forall\alpha,\beta\in\mathfrak{N}_\alpha. \tag{46}$$

EP, like LBP, is not guaranteed to converge. Hence, provably convergence double-loop algorithms have been proposed that directly minimize the weakly constrained BFE, e.g., [67].

F. Approximate Message Passing

So-called *approximate message passing* (AMP) algorithms [59], [60] have recently been developed for the separable generalized linear model

$$p(\boldsymbol{x}) = \prod_{j=1}^N p_{\mathsf{x}}(x_j),\ \ p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{m=1}^M p_{\mathsf{y}|\mathsf{z}}(y_m|\boldsymbol{a}_m^T\boldsymbol{x}) \tag{47}$$

where the prior $p(\boldsymbol{x})$ is fully factorizable, as is the conditional pdf $p(\boldsymbol{y}|\boldsymbol{z})$ relating the observation vector $\boldsymbol{y}$ to the (hidden) transform output vector $\boldsymbol{z} \triangleq \boldsymbol{A}\boldsymbol{x}$, where $\boldsymbol{A} \triangleq [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_M]^T \in \mathbb{R}^{M\times N}$ is a known linear transform. Like EP, AMP allows tractable inference under *generic*[3] p_{x} and $p_{\mathsf{y}|\mathsf{z}}$.

AMP can be derived as an approximation of LBP on the factor graph constructed with inner clusters $\boldsymbol{x}_\beta = x_\beta$ for $\beta = 1, \ldots, N$, with outer clusters $\boldsymbol{x}_\alpha = \boldsymbol{x}$ for $\alpha = 1, \ldots, M$ and $\boldsymbol{x}_\alpha = x_{\alpha-M}$ for $\alpha = M+1, \ldots, M+N$, and with factors

$$f_\alpha(\boldsymbol{x}_\alpha) = \begin{cases} p_{\mathsf{y}|\mathsf{z}}(y_\alpha|\boldsymbol{a}_\alpha^T\boldsymbol{x}) & \alpha = 1, \ldots, M \\ p_{\mathsf{x}}(x_{\alpha-M}) & \alpha = M+1, \ldots, M+N. \end{cases} \quad (48)$$

In the large-system limit (LSL), i.e., $M, N \to \infty$ for fixed ratio M/N, the LBP beliefs $q_\beta(x_\beta)$ simplify to

$$q_\beta(x_\beta) \propto p_{\mathsf{x}}(x_\beta)\mathcal{N}(x_\beta; \widehat{r}_\beta, \tau) \quad (49)$$

where $\{\widehat{r}_\beta\}_{\beta=1}^N$ and τ are iteratively updated parameters. Similarly, for $m = 1, \ldots, M$, the belief on z_m, denoted by $q_{\mathsf{z},m}(\cdot)$, simplifies to

$$q_{\mathsf{z},m}(z_m) \propto p_{\mathsf{y}|\mathsf{z}}(y_m|z_m)\mathcal{N}(z_m; \widehat{p}_m, \nu) \quad (50)$$

where $\{\widehat{p}_m\}_{m=1}^M$ and ν are iteratively updated parameters. Each AMP iteration requires only one evaluation of the mean and variance of (49)–(50), one multiplication by $\boldsymbol{A}$ and $\boldsymbol{A}^T$, and relatively few iterations, making it very computationally efficient, especially when these multiplications have fast implementations (e.g., using fast Fourier transforms and discrete wavelet transforms).

In the LSL under i.i.d sub-Gaussian $\boldsymbol{A}$, AMP is fully characterized by a scalar state evolution (SE). When this SE has a unique fixed point, the marginal posterior approximations (49)–(50) are known to be exact [68], [69].

For generic $\boldsymbol{A}$, AMP's fixed points coincide with the stationary points of an LSL version of the BFE [70], [71]. When AMP converges, its posterior approximations are often very accurate (e.g., [72]), but AMP does not always converge. In the special case of Gaussian likelihood $p_{\mathsf{y}|\mathsf{z}}$ and prior p_{x}, AMP convergence is fully understood: convergence depends on the ratio of peak-to-average squared singular values of $\boldsymbol{A}$, and convergence can be guaranteed for any $\boldsymbol{A}$ with appropriate damping [73]. For generic p_{x} and $p_{\mathsf{y}|\mathsf{z}}$, damping greatly helps convergence [74] but theoretical guarantees are lacking. A double-loop algorithm to directly minimize the LSL-BFE was recently proposed and shown to have global convergence for strictly log-concave p_{x} and $p_{\mathsf{y}|\mathsf{z}}$ under generic $\boldsymbol{A}$ [75].

IV. Optimization Methods

A. Optimization Problem

The Monte Carlo methods described in Section II provide a general approach for estimating reliably posterior probabilities and expectations. However, their high computational cost often makes them unattractive for applications involving very high dimensionality or tight computing time constraints. One alternative strategy is to perform inference approximately by using deterministic surrogates as described in Section III. Unfortunately, these faster inference methods are not as generally applicable, and because they rely on approximations, the resulting inferences can suffer from estimation bias. As already mentioned, if one focuses on the MAP estimator, efficient optimization techniques can be employed, which are often more computationally tractable than MCMC methods and, for which strong guarantees of convergence exist. In many SP applications, the computation of the MAP estimator of $\boldsymbol{x}$ can be formulated as an optimization problem having the following form

$$\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}}\ \varphi(\boldsymbol{H}\boldsymbol{x}, \boldsymbol{y}) + g(\boldsymbol{D}\boldsymbol{x}) \quad (51)$$

where $\varphi: \mathbb{R}^M \times \mathbb{R}^M \to]-\infty, +\infty]$, $g: \mathbb{R}^P \to]-\infty, +\infty]$, $\boldsymbol{H} \in \mathbb{R}^{M\times N}$, and $\boldsymbol{D} \in \mathbb{R}^{P\times N}$ with $P \in \mathbb{N}^*$. For example, $\boldsymbol{H}$ may be a linear operator modeling a degradation of the signal of interest, $\boldsymbol{y}$ a vector of observed data, φ a least-squares criterion corresponding to the negative log-likelihood associated with an additive zero-mean white Gaussian noise, g a sparsity promoting measure, e.g., an ℓ_1 norm, and $\boldsymbol{D}$ a frame analysis transform or a gradient operator.

Often, φ is an additively separable function, i.e.,

$$\varphi(\boldsymbol{z}, \boldsymbol{y}) = \frac{1}{M}\sum_{i=1}^M \varphi_i(z_i, y_i) \quad \forall (\boldsymbol{z}, \boldsymbol{y}) \in (\mathbb{R}^M)^2 \quad (52)$$

where $\boldsymbol{z} = [z_1, \ldots, z_M]^T$. Under this condition, the previous optimization problem becomes an instance of the more general stochastic one

$$\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}}\ \Phi(\boldsymbol{x}) + g(\boldsymbol{D}\boldsymbol{x}) \quad (53)$$

involving the expectation

$$\Phi(\boldsymbol{x}) = \mathsf{E}\{\varphi_j(\boldsymbol{h}_j^T\boldsymbol{x}, y_j)\} \quad (54)$$

where j, $\boldsymbol{y}$, and $\boldsymbol{H}$ are now assumed to be random variables and the expectation is computed with respect to the joint distribution of $(j, \boldsymbol{y}, \boldsymbol{H})$, with $\boldsymbol{h}_j^T$ the j-th line of $\boldsymbol{H}$. More precisely, when (52) holds, (51) is then a special case of (53) with j uniformly distributed over $\{1, \ldots, M\}$ and $(\boldsymbol{y}, \boldsymbol{H})$ deterministic. Conversely, it is also possible to consider that j is deterministic and that for every $i \in \{2, \ldots, M\}$, $\varphi_i = \varphi_1$, and $(y_i, \boldsymbol{h}_i)_{1\le i\le M}$ are identically distributed random variables. In this second scenario, because of the separability condition (52), the optimization problem (51) can be regarded as a proxy for (53), where the expectation $\Phi(\boldsymbol{x})$ is approximated by a sample estimate (or stochastic approximation under suitable mixing assumptions). All these remarks illustrate the existing connections between problems (51) and (53).

Note that the stochastic optimization problem defined in (53) has been extensively investigated in two communities: machine learning, and adaptive filtering, often under quite different practical assumptions on the forms of the functions $(\varphi_j)_{j\ge1}$ and g. In machine learning [76]–[78], $\boldsymbol{x}$ indeed represents the vector of parameters of a classifier which has to be learnt, whereas in adaptive filtering [79], [80], it is generally the impulse response of an unknown filter which needs to be identified and possibly tracked. In order to simplify our presentation, in the rest of this

[3]More precisely, the AMP algorithm [59] handles Gaussian $p_{\mathsf{y}|\mathsf{z}}$ while the generalized AMP (GAMP) algorithm [60] handles arbitrary $p_{\mathsf{y}|\mathsf{z}}$.

section, we will assume that the functions $(\varphi_j)_{j\geq 1}$ are convex and Lipschitz-differentiable with respect to their first argument (for example, they may be logistic functions).

B. Optimization Algorithms for Solving Stochastic Problems

The main difficulty arising in the resolution of the stochastic optimization problem (53) is that the integral involved in the expectation term often cannot be computed in practice since it is generally high-dimensional and the underlying probability measure is usually unknown. Two main computational approaches have been proposed in the literature to overcome this issue. The first idea is to approximate the expected loss function by using a finite set of observations and to minimize the associated empirical loss (51). The resulting deterministic optimization problem can then be solved by using either deterministic or stochastic algorithms, the latter being the topic of Section IV-C. Here, we focus on the second family of methods grounded in stochastic approximation techniques to handle the expectation in (54). More precisely, a sequence of identically distributed samples $(y_j, \boldsymbol{h}_j)_{j\geq 1}$ is drawn, which are processed sequentially according to some update rule. The iterative algorithm aims to produce a sequence of random iterates $(\boldsymbol{x}_j)_{j\geq 1}$ converging to a solution to (53).

We begin with a group of *online* learning algorithms based on extensions of the well-known *stochastic gradient descent* (SGD) approach. Then we will turn our attention to stochastic optimization techniques developed in the context of adaptive filtering.

1) Online Learning Methods Based on SGD: Let us assume that an estimate $\boldsymbol{u}_j \in \mathbb{R}^N$ of the gradient of Φ at $\boldsymbol{x}_j$ is available at each iteration $j \geq 1$. A popular strategy for solving (53) in this context leverages the gradient estimates to derive a so-called *stochastic forward-backward* (SFB) scheme, (also sometimes called *stochastic proximal gradient algorithm*)

$$\begin{aligned}(\forall j \geq 1) \quad \boldsymbol{z}_j &= \operatorname{prox}_{\gamma_j g\circ \boldsymbol{D}}(\boldsymbol{x}_j - \gamma_j \boldsymbol{u}_j)\\ \boldsymbol{x}_{j+1} &= (1-\lambda_j)\boldsymbol{x}_j + \lambda_j \boldsymbol{z}_j \end{aligned} \tag{55}$$

where $(\gamma_j)_{j\geq 1}$ is a sequence of positive stepsize values and $(\lambda_j)_{j\geq 1}$ is a sequence of relaxation parameters in $]0,1]$. Hereabove, $\operatorname{prox}_{\psi}(\boldsymbol{v})$ denotes the proximity operator at $\boldsymbol{v} \in \mathbb{R}^N$ of a lower-semicontinuous convex function $\psi\colon \mathbb{R}^N \to]-\infty, +\infty]$ with nonempty domain, i.e., the unique minimizer of $\psi + (1/2)\|\cdot - \boldsymbol{v}\|^2$ (see [81] and the references therein), and $g \circ \boldsymbol{D}(\boldsymbol{x}) = g(\boldsymbol{D}\boldsymbol{x})$. A convergence analysis of the SFB scheme has been conducted in [82]–[85], under various assumptions on the functions Φ, g, on the stepsize sequence, and on the statistical properties of $(\boldsymbol{u}_j)_{j\geq 1}$. For example, if $\boldsymbol{x}_1$ is set to a given (deterministic) value, the sequence $(\boldsymbol{x}_j)_{j\geq 1}$ generated by (55) is guaranteed to converge almost surely to a solution of Problem (53) under the following technical assumptions [84]

(i) Φ has a β^{-1}-Lipschitzian gradient with $\beta \in]0, +\infty[$, g is a lower-semicontinuous convex function, and $\Phi + g \circ \boldsymbol{D}$ is strongly convex.

(ii) For every $j \geq 1$,

$$\begin{aligned}&\mathrm{E}\{\|\boldsymbol{u}_j\|^2\} < +\infty,\ \mathrm{E}\{\boldsymbol{u}_j \mid \boldsymbol{\mathcal{X}}_{j-1}\} = \nabla\Phi(\boldsymbol{x}_j),\\ &\mathrm{E}\{\|\boldsymbol{u}_j - \nabla\Phi(\boldsymbol{x}_j)\|^2 \mid \boldsymbol{\mathcal{X}}_{j-1}\} \leq \sigma^2(1+\alpha_j\|\nabla\Phi(\boldsymbol{x}_j)\|^2)\end{aligned}$$

where $\boldsymbol{\mathcal{X}}_j = (y_i, \boldsymbol{h}_i)_{1\leq i\leq j}$, and α_j and σ are positive values such that $\gamma_j \leq (2-\epsilon)\beta(1+2\sigma^2\alpha_j)^{-1}$ with $\epsilon > 0$.

(iii) We have

$$\sum_{j\geq 1}\lambda_j\gamma_j = +\infty \quad \text{and} \quad \sum_{j\geq 1}\chi_j^2 < +\infty$$

where, for every $j \geq 1$, $\chi_j^2 = \lambda_j\gamma_j^2(1+2\alpha_j\|\nabla\Phi(\overline{\boldsymbol{x}})\|^2)$ and $\overline{\boldsymbol{x}}$ is the solution of Problem (53).

When $g \equiv 0$, the SFB algorithm in (55) becomes equivalent to SGD [86]–[89]. According to the above result, the convergence of SGD is ensured as soon as $\sum_{j\geq 1}\lambda_j\gamma_j = +\infty$ and $\sum_{j\geq 1}\lambda_j\gamma_j^2 < +\infty$. In the unrelaxed case defined by $\lambda_j \equiv 1$, we then retrieve a particular case of the decaying condition $\gamma_j \propto j^{-1/2-\delta}$ with $\delta \in]0, 1/2]$ usually imposed on the stepsize in the convergence studies of SGD under slightly different assumptions on the gradient estimates $(\boldsymbol{u}_j)_{j\geq 1}$ (see [90], [91] for more details). Note also that better convergence properties can be obtained, if a Polyak-Ruppert averaging approach is performed, i.e., the averaged sequence $(\overline{\boldsymbol{x}}_j)_{j\geq 1}$, defined as $\overline{\boldsymbol{x}}_j = (1/j)\sum_{i=1}^j \boldsymbol{x}_i$ for every $j \geq 1$, is considered instead of $(\boldsymbol{x}_j)_{j\geq 1}$ in the convergence analysis [90], [92].

We now comment on approaches related to SFB that have been proposed in the literature to solve (53). It should first be noted that a simple alternative strategy to deal with a possibly nonsmooth term g is to incorporate a subgradient step into the previously mentioned SGD algorithm [93]. However, this approach, like its deterministic version, may suffer from a slow convergence rate [94]. Another family of methods, close to SFB, adopt the *regularized dual averaging* (RDA) strategy, first introduced in [94]. The principal difference between SFB and RDA methods is that the latter rely on iterative averaging of the stochastic gradient estimates, which consists of replacing in the update rule (55), $(\boldsymbol{u}_j)_{j\geq 1}$ by $(\overline{\boldsymbol{u}}_j)_{j\geq 1}$ where, for every $j \geq 1$, $\overline{\boldsymbol{u}}_j = (1/j)\sum_{i=1}^j \boldsymbol{u}_i$. The advantage is that it provides convergence guarantees for nondecaying stepsize sequences. Finally, the so-called *composite mirror descent* methods, introduced in [95], can be viewed as extended versions of the SFB algorithm where the proximity operator is computed with respect to a non Euclidean distance (typically, a Bregman divergence).

In the last few years, a great deal of effort has been made to modify SFB when the proximity operator of $g \circ \boldsymbol{D}$ does not have a simple expression, but when g can be split into several terms whose proximity operators are explicit. We can mention the stochastic proximal averaging strategy from [96], the stochastic *alternating direction method of mutipliers* (ADMM) from [97]–[99] and the alternating block strategy from [100] suited to the case when $g \circ \boldsymbol{D}$ is a separable function.

Another active research area addresses the search for strategies to improve the convergence rate of SFB. Two main approaches can be distinguished in the literature. The first, adopted for example in [83], [101]–[103], relies on subspace acceleration. In such methods, usually reminiscent of Nesterov's acceleration techniques in the deterministic case, the convergence rate is improved by using information from previous iterates for the construction of the new estimate. Another efficient way to accelerate the convergence of SFB is to incorporate in the update rule second-order information one may have on the cost

functions. For instance, the method described in [104] incorporates quasi-Newton metrics into the SFB and RDA algorithms, and the *natural gradient* method from [105] can be viewed as a preconditioned SGD algorithm. The two strategies can be combined, as for example, in [106].

2) Adaptive Filtering Methods: In adaptive filtering, stochastic gradient-like methods have been quite popular for a long time [107], [108]. In this field, the functions $(\varphi_j)_{j\geq 1}$ often reduce to a least squares criterion

$$(\forall j \geq 1) \quad \varphi_j(\boldsymbol{h}_j^T \boldsymbol{x}, y_j) = (\boldsymbol{h}_j^T \boldsymbol{x} - y_j)^2 \tag{56}$$

where $\boldsymbol{x}$ is the unknown impulse response. However, a specific difficulty to be addressed is that the designed algorithms must be able to deal with dynamical problems the optimal solution of which may be time-varying due to some changes in the statistics of the available data. In this context, it may be useful to adopt a multivariate formulation by imposing, at each iteration $j \geq Q$

$$\boldsymbol{y}_j \simeq \boldsymbol{H}_j \boldsymbol{x} \tag{57}$$

where $\boldsymbol{y}_j = [y_j, \ldots, y_{j-Q+1}]^T$, $\boldsymbol{H}_j = [\boldsymbol{h}_j, \ldots, \boldsymbol{h}_{j-Q+1}]^T$, and $Q \geq 1$. This technique, reminiscent of mini-batch procedures in machine learning, constitutes the principle of *affine projection algorithms*, the purpose of which is to accelerate the convergence speed [109]. Our focus now switches to recent work which aims to impose some sparse structure on the desired solution.

A simple method for imposing sparsity is to introduce a suitable adaptive preconditioning strategy in the stochastic gradient iteration, leading to the so-called *proportionate least mean square* method [110], [111], which can be combined with affine projection techniques [112], [113]. Similarly to the work already mentioned that has been developed in the machine learning community, a second approach proceeds by minimizing penalized criteria such as (53) where g is a sparsity measure and $\boldsymbol{D} = \boldsymbol{I}_N$. In [114], [115], *zero-attracting* algorithms are developed which are based on the stochastic subgradient method. These algorithms have been further extended to affine projection techniques in [116]–[118]. Proximal methods have also been proposed in the context of adaptive filtering, grounded on the use of the forward-backward algorithm [119], an accelerated version of it [120], or primal-dual approaches [121]. It is interesting to note that *proportionate affine projection algorithms* can be viewed as special cases of these methods [119]. Other types of algorithms have been proposed which provide extensions of the *recursive least squares* method, which is known for its fast convergence properties [106], [122], [123]. Instead of minimizing a sparsity promoting criterion, it is also possible to formulate the problem as a feasibility problem where, at iteration $j \geq Q$, one searches for a vector $\boldsymbol{x}$ satisfying both $\sup_{j\leq i\leq j-Q+1} |y_i - \boldsymbol{h}_i^T \boldsymbol{x}| \leq \eta$ and $\|\boldsymbol{x}\|_1 \leq \rho$, where $\|\cdot\|_1$ denotes the (possibly weighted) ℓ_1 norm and $(\eta, \rho) \in]0, +\infty[^2$. Over-relaxed projection algorithms allow such kind of problems to be solved efficiently [124], [125].

C. Stochastic Algorithms for Solving Deterministic Optimization Problems

We now consider the deterministic optimization problem defined by (51) and (52). Of particular interest is the case when the dimensions N and/or M are very large (for instance, in [126], $M = 2500000$ and in [127], $N = 100250$).

1) Incremental Gradient Algorithms: Let us start with incremental methods, which are dedicated to the solution of (51) when M is large, so that one prefers to exploit at each iteration a single term φ_j, usually through its gradient, rather than the global function φ. There are many variants of incremental algorithms, which differ in the assumptions made on the functions involved, on the stepsize sequence, and on the way of activating the functions $(\varphi_i)_{1\leq i\leq M}$. This order could follow either a deterministic [128] or a randomized rule. However, it should be noted that the use of randomization in the selection of the components presents some benefits in terms of convergence rates [129] which are of particular interest in the context of machine learning [130], [131], where the user can only afford few full passes over the data. Among randomized incremental methods, the SAGA algorithm [132], presented below, allows the problem defined in (51) to be solved when the function g is not necessarily smooth, by making use of the proximity operator introduced previously. The n-th iteration of SAGA reads as

$$\begin{aligned}
\boldsymbol{u}_n &= \boldsymbol{h}_{j_n} \nabla \varphi_{j_n}(\boldsymbol{h}_{j_n}^T \boldsymbol{x}_n, y_{j_n}) - \boldsymbol{h}_{j_n} \nabla \varphi_{j_n}(\boldsymbol{h}_{j_n}^T \boldsymbol{z}_{j_n,n}, y_{j_n}) \\
&\quad + \frac{1}{M} \sum_{i=1}^{M} \boldsymbol{h}_i \nabla \varphi_i(\boldsymbol{h}_i^T \boldsymbol{z}_{i,n}, y_i) \\
\boldsymbol{z}_{j_n,n+1} &= \boldsymbol{x}_n \\
\boldsymbol{z}_{i,n+1} &= \boldsymbol{z}_{i,n} \quad \forall\, i \in \{1, \ldots, M\} \setminus \{j_n\} \\
\boldsymbol{x}_{n+1} &= \operatorname{prox}_{\gamma g \circ \boldsymbol{D}}(\boldsymbol{x}_n - \gamma \boldsymbol{u}_n)
\end{aligned} \tag{58}$$

where $\gamma \in]0, +\infty[$, for all $i \in \{1, \ldots, M\}$, $\boldsymbol{z}_{i,1} = \boldsymbol{x}_1 \in \mathbb{R}^N$, and j_n is drawn from an i.i.d. uniform distribution on $\{1, \ldots, M\}$. Note that, although the storage of the variables $(\boldsymbol{z}_{i,n})_{1\leq i\leq M}$ can be avoided in this method, it is necessary to store the M gradient vectors $\big(\boldsymbol{h}_i \nabla \varphi_i(\boldsymbol{h}_i^T \boldsymbol{z}_{i,n}, y_i)\big)_{1\leq i\leq M}$. The convergence of Algorithm (58) has been analyzed in [132]. If the functions $(\varphi_i)_{1\leq i\leq M}$ are β^{-1}-Lipschitz differentiable and μ-strongly convex with $(\beta, \mu) \in]0, +\infty[^2$ and the stepsize γ equals $\beta/(2(\mu\beta M + 1))$, then $(\mathrm{E}\{\|\boldsymbol{x}_n - \overline{\boldsymbol{x}}\|^2\})_{n\in\mathbb{N}}$ goes to zero geometrically with rate $1 - \gamma$, where $\overline{\boldsymbol{x}}$ is the solution to Problem (51). When only convexity is assumed, a weaker convergence result is available.

The relationship between Algorithm (58) and other stochastic incremental methods existing in the literature is worthy of comment. The main distinction arises in the way of building the gradient estimates $(\boldsymbol{u}_n)_{n\geq 1}$. The standard incremental gradient algorithm, analyzed for instance in [129], relies on simply defining, at iteration n, $\boldsymbol{u}_n = \boldsymbol{h}_{j_n} \nabla \varphi_{j_n}(\boldsymbol{h}_{j_n}^T \boldsymbol{x}_n, y_{j_n})$. However, this approach, while leading to a smaller computational complexity per iteration and to a lower memory requirement, gives rise to suboptimal convergence rates [91], [129], mainly due to the fact that its convergence requires a stepsize sequence $(\gamma_n)_{n\geq 1}$ decaying to zero. Motivated by this observation, much

recent work [126], [130]–[134] has been dedicated to the development of *fast incremental gradient methods*, which would benefit from the same convergence rates as batch optimization methods, while using a randomized incremental approach. A first class of methods relies on a *variance reduction approach* [130], [132]–[134] which aims at diminishing the variance in successive estimates $(\boldsymbol{u}_n)_{n\geq 1}$. All of the aforementioned algorithms are based on iterations which are similar to (58). In the *stochastic variance reduction gradient* method and the *semi-stochastic gradient descent* method proposed in [133], [134], a full gradient step is made at every K iterations, $K \geq 1$, so that a single vector $\widetilde{\boldsymbol{z}}_n$ is used instead of $(\boldsymbol{z}_{i,n})_{1\leq i\leq M}$ in the update rule. This so-called mini-batch strategy leads to a reduced memory requirement at the expense of more gradient evaluations. As pointed out in [132], the choice between one strategy or another may depend on the problem size and on the computer architecture. In the *stochastic average gradient* algorithm (SAGA) from [130], a multiplicative factor $1/M$ is placed in front of the gradient differences, leading to a lower variance counterbalanced by a bias in the gradient estimates. It should be emphasized that the work in [130], [133] is limited to the case when $g \equiv 0$. A second class of methods, closely related to SAGA, consists of applying the proximal step to $\overline{\boldsymbol{z}}_n - \gamma \boldsymbol{u}_n$, where $\overline{\boldsymbol{z}}_n$ is the average of the variables $(\boldsymbol{z}_{i,n})_{1\leq i\leq M}$ (which thus need to be stored). This approach is retained for instance in the Finito algorithm [131] as well as in some instances of the *minimization by incremental surrogate optimization* (MISO) algorithm, proposed in [126]. These methods are of particular interest when the extra storage cost is negligible with respect to the high computational cost of the gradients. Note that the MISO algorithm relying on the majoration-minimization framework employs a more generic update rule than Finito and has proven convergence guarantees even when g is nonzero.

2) Block Coordinate Approaches: In the spirit of the Gauss-Seidel method, an efficient approach for dealing with Problem (51) when N is large consists of resorting to block coordinate alternating strategies. Sometimes, such a block alternation can be performed in a deterministic manner [135], [136]. However, many optimization methods are based on fixed point algorithms, and it can be shown that with deterministic block coordinate strategies, the contraction properties which are required to guarantee the convergence of such algorithms are generally no longer satisfied. In turn, by resorting to stochastic techniques, these properties can be retrieved in some probabilistic sense [85]. In addition, using stochastic rules for activating the different blocks of variables often turns out to be more flexible.

To illustrate why there is interest in block coordinate approaches, let us split the target variable $\boldsymbol{x}$ as $[\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_K^T]^T$, where, for every $k \in \{1, \ldots, K\}$, $\boldsymbol{x}_k \in \mathbb{R}^{N_k}$ is the k-th block of variables with reduced dimension N_k (with $N_1 + \cdots + N_K = N$). Let us further assume that the regularization function can be blockwise decomposed as

$$g(\boldsymbol{D}\boldsymbol{x}) = \sum_{k=1}^{K} g_{1,k}(\boldsymbol{x}_k) + g_{2,k}(\boldsymbol{D}_k \boldsymbol{x}_k) \tag{59}$$

where, for every $k \in \{1, \ldots, K\}$, $\boldsymbol{D}_k$ is a matrix in $\mathbb{R}^{P_k \times N_k}$, and $g_{1,k}\colon \mathbb{R}^{N_k} \to]-\infty, +\infty]$ and $g_{2,k}\colon \mathbb{R}^{P_k} \to]-\infty, +\infty]$ are proper lower-semicontinuous convex functions. Then, the stochastic primal-dual proximal algorithm allowing us to solve Problem (51) is given by

Algorithm 5 Stochastic primal-dual proximal algorithm

for $n = 1, 2, \ldots$ **do**
  for $k = 1$ to K **do**
    with probability $\varepsilon_k \in (0, 1]$ **do**

$$\boldsymbol{v}_{k,n+1} = (\mathrm{Id} - \mathrm{prox}_{\tau^{-1} g_{2,k}})(\boldsymbol{v}_{k,n} + \boldsymbol{D}_k \boldsymbol{x}_{k,n})$$

$$\boldsymbol{x}_{k,n+1} = \mathrm{prox}_{\gamma g_{1,k}}\Bigg(\boldsymbol{x}_{k,n} - \gamma\Bigg(\tau \boldsymbol{D}_k^T (2\boldsymbol{v}_{k+1,n} - \boldsymbol{v}_{k,n}) + \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{h}_{i,k} \nabla \varphi_i\Big(\sum_{k\text{ß}=1}^{K} h_{i,k\text{ß}}^T x_{k\text{ß},n}, y_i\Big).\Bigg)\Bigg)$$

    otherwise

$$\boldsymbol{v}_{k,n+1} = \boldsymbol{v}_{k,n},\ \boldsymbol{x}_{k,n+1} = \boldsymbol{x}_{k,n}.$$

  end for
end for

In the algorithm above, for every $i \in \{1, \ldots, M\}$, the scalar product $\boldsymbol{h}_i^T \boldsymbol{x}$ has been rewritten in a blockwise manner as $\sum_{k'=1}^{K} \boldsymbol{h}_{i,k'}^T \boldsymbol{x}_{k'}$. Under some stability conditions on the choice of the positive step sizes τ and γ, $\boldsymbol{x}_n = [\boldsymbol{x}_{1,n}^T, \ldots, \boldsymbol{x}_{K,n}^T]^T$ converges almost surely to a solution of the minimization problem, as $n \to +\infty$ (see [137] for more technical details). It is important to note that the convergence result was established for arbitrary probabilities $\boldsymbol{\varepsilon} = [\varepsilon_1, \ldots, \varepsilon_K]^T$, provided that the block activation probabilities ε_k are positive and independent of n. Note that the various blocks can also be activated in a dependent manner at a given iteration n. Like its deterministic counterparts (see [138] and the references therein), this algorithm enjoys the property of not requiring any matrix inversion, which is of paramount importance when the matrices $(\boldsymbol{D}_k)_{1\leq k\leq K}$ are of large size and do not have some simple forms.

When $g_{2,k} \equiv 0$, the random block coordinate forward-backward algorithm is recovered as an instance of Algorithm 5 since the dual variables $(\boldsymbol{v}_{k,n})_{1\leq k\leq K, n\in\mathbb{N}}$ can be set to 0 and the constant τ becomes useless. An extensive literature exists on the latter algorithm and its variants. In particular, its almost sure convergence was established in [85] under general conditions, whereas some worst case convergence rates were derived in [139]–[143]. In addition, if $g_{1,k} \equiv 0$, the *random block coordinate descent* algorithm is obtained [144].

When the objective function minimized in Problem (51) is strongly convex, the random block coordinate forward-backward algorithm can be applied to the dual problem, in a similar fashion to the dual forward-backward method used in the deterministic case [145]. This leads to so-called dual ascent strategies which have become quite popular in machine learning [146]–[149].

Random block coordinate versions of other proximal algorithms such as the Douglas-Rachford algorithm and ADMM have also been proposed [85], [150]. Finally, it is worth emphasizing that asynchronous distributed algorithms can be deduced from various randomly activated block coordinate methods [137], [151]. As well as dual ascent methods, the latter algorithms can also be viewed as incremental methods.

V. Areas of Intersection: Optimization-Within-MCMC and MCMC-Driven Optimization

There are many important examples of the synergy between stochastic simulation and optimization, including global optimization by simulated annealing, stochastic EM algorithms, and adaptive MCMC samplers [4]. In this section we highlight some of the interesting new connections between modern simulation and optimization that we believe are particularly relevant for the SP community, and that we hope will stimulate further research in this community.

A. Riemannian Manifold MALA and HMC

Riemannian manifold MALA and HMC exploit differential geometry for the problem of specifying an appropriate proposal covariance matrix $\boldsymbol{\Sigma}$ that takes into account the geometry of the target density π [20]. These new methods stem from the observation that specifying $\boldsymbol{\Sigma}$ is equivalent to formulating the Langevin or Hamiltonian dynamics in an Euclidean parameter space with inner product $\langle \boldsymbol{w}, \boldsymbol{\Sigma}^{-1}\boldsymbol{x}\rangle$. Riemannian methods advance this observation by considering a smoothly-varying position dependent matrix $\boldsymbol{\Sigma}(\boldsymbol{x})$, which arises naturally by formulating the dynamics in a Riemannian manifold. The choice of $\boldsymbol{\Sigma}(\boldsymbol{x})$ then becomes the more familiar problem of specifying a metric or distance for the parameter space [20]. Notice that the Riemannian and the canonical Euclidean gradients are related by $\tilde{\nabla} g(\boldsymbol{x}) = \boldsymbol{\Sigma}(\boldsymbol{x})\nabla g(\boldsymbol{x})$. Therefore this problem is also closely related to gradient preconditioning in gradient descent optimization discussed in Section IV.B. Standard choices for $\boldsymbol{\Sigma}$ include for example the inverse Hessian matrix [21], [31], which is closely related to Newton's optimization method, and the inverse Fisher information matrix [20], which is the "natural" metric from an information geometry viewpoint and is also related to optimization by natural gradient descent [105]. These strategies have originated in the computational statistics community, and perform well in inference problems that are not too high-dimensional. Therefore, the challenge is to design new metrics that are appropriate for SP statistical models (see [152], [153] for recent work in this direction).

B. Proximal MCMC Algorithms

Most high-dimensional MCMC algorithms rely particularly strongly on differential analysis to navigate vast parameter spaces efficiently. Conversely, the potential of convex calculus for MCMC simulation remains largely unexplored. This is in sharp contrast with modern high-dimensional optimization described in Section IV, where convex calculus in general, and proximity operators [81], [154] in particular, are used extensively. This raises the question as to whether convex calculus and proximity operators can also be useful for stochastic simulation, especially for high-dimensional target densities that are log-concave, and possibly not continuously differentiable.

This question was studied recently in [24] in the context of Langevin algorithms. As explained in Section II.B, Langevin MCMC algorithms are derived from discrete-time approximations of the time-continuous Langevin diffusion process (5). Of course, the stability and accuracy of the discrete approximations determine the theoretical and practical convergence properties of the MCMC algorithms they underpin. The approximations commonly used in the literature are generally well-behaved and lead to powerful MCMC methods. However, they can perform poorly if π is not sufficiently regular, for example if π is not continuously differentiable, if it is heavy-tailed, or if it has lighter tails than the Gaussian distribution. This drawback limits the application of MCMC approaches to many SP problems, which rely increasingly on models that are not continuously differentiable or that involve constraints.

Using proximity operators, the following proximal approximation for the Langevin diffusion process (5) was recently proposed in [24]

$$X^{(t+1)} \sim \mathcal{N}\left(\operatorname{prox}_{(-\delta/2)\log\pi}\left(X^{(t)}\right), \delta \mathbb{I}_N\right) \tag{60}$$

as an alternative to the standard forward Euler approximation $X^{(t+1)} \sim \mathcal{N}\left(X^{(t)} + (\delta/2)\nabla \log \pi\left(X^{(t)}\right), \delta \mathbb{I}_n\right)$ used in MALA[4]. Similarly to MALA, the time step δ can be adjusted online to achieve an acceptance probability of approximately 50%. It was established in [24] that when π is log-concave, (60) defines a remarkably stable discretization of (5) with optimal theoretical convergence properties. Moreover, the "proximal" MALA resulting from combining (60) with an MH step has very good geometric ergodicity properties. In [24], the algorithm efficiency was demonstrated empirically on challenging models that are not well addressed by other MALA or HMC methodologies, including an image resolution enhancement model with a total-variation prior. Further practical assessments of proximal MALA algorithms would therefore be a welcome area of research.

Proximity operators have also been used recently in [155] for HMC sampling from log-concave densities that are not continuously differentiable. The experiments reported in [155] show that this approach can be very efficient, in particular for SP models involving high-dimensionality and non-smooth priors. Unfortunately, theoretically analyzing HMC methods is difficult, and the precise theoretical convergence properties of this algorithm are not yet fully understood. We hope future work will focus on this topic.

C. Optimization-Driven Gaussian Simulation

The standard approach for simulating from a multivariate Gaussian distribution with precision matrix $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ is to perform a Cholesky factorization $\boldsymbol{Q} = \boldsymbol{L}^T\boldsymbol{L}$, generate an auxiliary Gaussian vector $\boldsymbol{w} \sim \mathcal{N}(0, \mathbb{I}_N)$, and then obtain the desired sample $\boldsymbol{x}$ by solving the linear system $\boldsymbol{L}\boldsymbol{x} = \boldsymbol{w}$ [156]. The computational complexity of this approach generally scales at a prohibitive rate $\mathcal{O}(N^3)$ with the model dimension N, making it impractical for large problems, (note however that there are specific cases with lower complexity, for instance when $\boldsymbol{Q}$ is Toeplitz [157], circulant [158] or sparse [156]).

[4]Recall that $\operatorname{prox}_{\varphi}(\boldsymbol{v})$ denotes the proximity operator of φ evaluated at $\boldsymbol{v} \in \mathbb{R}^N$ [81], [154].

Optimization-driven Gaussian simulators arise from the observation that the samples can also be obtained by minimizing a carefully designed stochastic cost function [159], [160]. For illustration, consider a Bayesian model with Gaussian likelihood $\boldsymbol{y}|\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{H}\boldsymbol{x}, \boldsymbol{\Sigma}_{\boldsymbol{y}})$ and Gaussian prior $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{x}_0, \boldsymbol{\Sigma}_{\boldsymbol{x}})$, for some linear observation operator $\boldsymbol{H} \in \mathbb{R}^{N\times M}$, prior mean $\boldsymbol{x}_0 \in \mathbb{R}^N$, and positive definite covariance matrices $\boldsymbol{\Sigma}_{\boldsymbol{x}} \in \mathbb{R}^{N\times N}$ and $\boldsymbol{\Sigma}_{\boldsymbol{y}} \in \mathbb{R}^{M\times M}$. The posterior distribution $p(\boldsymbol{x}|\boldsymbol{y})$ is Gaussian with mean $\boldsymbol{\mu} \in \mathbb{R}^N$ and precision matrix $\boldsymbol{Q} \in \mathbb{R}^{N\times N}$ given by

$$\begin{aligned}\boldsymbol{Q} &= \boldsymbol{H}^T\boldsymbol{\Sigma}_{\boldsymbol{y}}^{-1}\boldsymbol{H} + \boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1}\\ \boldsymbol{\mu} &= \boldsymbol{Q}^{-1}\left(\boldsymbol{H}^T\boldsymbol{\Sigma}_{\boldsymbol{y}}^{-1}\boldsymbol{y} + \boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1}\boldsymbol{x}_0\right).\end{aligned}$$

Simulating samples $\boldsymbol{x}|\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{Q}^{-1})$ by Cholesky factorization of $\boldsymbol{Q}$ can be computationally expensive when N is large. Instead, optimization-driven simulators generate samples by solving the following "random" minimization problem

$$\begin{aligned}\boldsymbol{x} = \underset{\boldsymbol{u}\in\mathbb{R}^N}{\operatorname{argmin}}\ &\left(\boldsymbol{w}_1 - \boldsymbol{H}\boldsymbol{u}\right)^T\boldsymbol{\Sigma}_{\boldsymbol{y}}^{-1}\left(\boldsymbol{w}_1 - \boldsymbol{H}\boldsymbol{u}\right)\\ &+ \left(\boldsymbol{w}_2 - \boldsymbol{u}\right)^T\boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1}\left(\boldsymbol{w}_2 - \boldsymbol{u}\right) \quad (61)\end{aligned}$$

with random vectors $\boldsymbol{w}_1 \sim \mathcal{N}(\boldsymbol{y}, \boldsymbol{\Sigma}_y)$ and $\boldsymbol{w}_2 \sim \mathcal{N}(\boldsymbol{x}_0, \boldsymbol{\Sigma}_x)$. It is easy to check that if (61) is solved exactly, then $\boldsymbol{x}$ is a sample from the desired posterior distribution $p(\boldsymbol{x}|\boldsymbol{y})$. From a computational viewpoint, however, it is significantly more efficient to solve (61) approximately, for example by using a few linear conjugate gradient iterations [160]. The approximation error can then be corrected by using an MH step [161], at the expense of introducing some correlation between the samples and therefore reducing the total effective sample size. Fortunately, there is an elegant strategy to determine automatically the optimal number of conjugate gradient iterations that maximizes the overall efficiency of the algorithm [161].

VI. Conclusions and Observations

In writing this paper we have sought to provide an introduction to stochastic simulation and optimization methods in a tutorial format, but which also raised some interesting topics for future research. We have addressed a variety of MCMC methods and discussed surrogate methods, such as variational Bayes, the Bethe approach, belief and expectation propagation, and approximate message passing. We also discussed a range of recent advances in optimization methods that have been proposed to solve stochastic problems, as well as stochastic methods for deterministic optimization. Subsequently, we highlighted new methods that combine simulation and optimization, such as proximal MCMC algorithms and optimization-driven Gaussian simulators. Our expectation is that future methodologies will become more flexible. Our community has successfully applied computational inference methods, as we have described, to a plethora of challenges across an enormous range of application domains. Each problem offers different challenges, ranging from model dimensionality and complexity, data (too much or too little), inferences, accuracy and computation times. Consequently, it seems not unreasonable to speculate that the different computational methodologies discussed in this paper will evolve to become more adaptable, with their boundaries becoming less well defined, and with the development of algorithms that make use of simulation, variational approximations and optimization simultaneously. Such an approach is more likely to be able to handle an even wider range of models, datasets, inferences, accuracies and computing times in a computationally efficient way.

References

[1] A. Dempster, N. M. Laird, and D. B. Rubin, "Maximum-likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc.*, vol. 39, pp. 1–17, 1977.

[2] R. Neal and G. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants," in *Learning in Graphical Models*, M. I. Jordan, Ed. Cambridge, MA, USA: MIT Press, 1998, pp. 355–368.

[3] H. Attias, "A variational Bayesian framework for graphical models," in *Proc. Neural Inf. Process. Syst. Conf.*, Denver, CO, USA, 2000, pp. 209–216.

[4] C. Robert and G. Casella, *Monte Carlo Statistical Methods*, 2nd ed. Berlin, Germany: Springer-Verlag, 2004.

[5] P. Green, K. Latuszyński, M. Pereyra, and C. P. Robert, "Bayesian computation: A summary of the current state, and samples backwards and forwards," *Statistics and Comput.*, 2015, to be published.

[6] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," in *In The Oxford Handbook of Nonlinear Filtering*, D. Crisan and B. Rozovsky, Eds. Oxford, U.K.: Oxford Univ. Press, 2011.

[7] A. Beskos, A. Jasra, E. A. Muzaffer, and A. M. Stuart, "Sequential Monte Carlo methods for Bayesian elliptic inverse problems," *Statist. Comput.*, vol. 25, 2015, to be published.

[8] J. Marin, P. Pudlo, C. Robert, and R. Ryder, "Approximate Bayesian computational methods," *Statist. Comput.*, pp. 1–14, 2011.

[9] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equations of state calculations by fast computational machine," *J. Chem. Phys.*, vol. 21, no. 6, pp. 1087–1091, 1953.

[10] W. Hastings, "Monte Carlo sampling using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[11] S. Chib and E. Greenberg, "Understanding the Metropolis-Hastings algorithm," *Ann. Math. Statist.*, vol. 49, pp. 327–335, 1995.

[12] M. Bédard, "Weak convergence of Metropolis algorithms for non-i.i.d. target distributions," *Ann. Appl. Probab.*, vol. 17, no. 4, pp. 1222–1244, 2007.

[13] G. O. Roberts and J. S. Rosenthal, "Optimal scaling for various Metropolis-Hastings algorithms," *Statist. Sci.*, vol. 16, no. 4, pp. 351–367, Nov. 2001.

[14] C. Andrieu and G. Roberts, "The pseudo-marginal approach for efficient Monte Carlo computations," *Ann. Statist.*, vol. 37, no. 2, pp. 697–725, 2009.

[15] C. Andrieu, A. Doucet, and R. Holenstein, "Particle Markov chain Monte Carlo (with discussion)," *J. Roy. Statist. Soc. Ser. B*, vol. 72, no. 2, pp. 269–342, 2011.

[16] M. Pereyra, N. Dobigeon, H. Batatia, and J.-Y. Tourneret, "Estimating the granularity coefficient of a Potts-Markov random field within an MCMC algorithm," *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2385–2397, Jun. 2013.

[17] S. Jarner and E. Hansen, "Geometric ergodicity of Metropolis algorithms," *Stochast. Process. Applicat.*, vol. 85, no. 2, pp. 341–361, 2000.

[18] A. Beskos, G. Roberts, and A. Stuart, "Optimal scalings for local Metropolis-Hastings chains on nonproduct targets in high dimensions," *Ann. Appl. Probab.*, vol. 19, no. 3, pp. 863–898, 2009.

[19] G. Roberts and R. Tweedie, "Exponential convergence of Langevin distributions and their discrete approximations," *Bernoulli*, vol. 2, no. 4, pp. 341–363, 1996.

[20] M. Girolami and B. Calderhead, "Riemann manifold Langevin and Hamiltonian Monte Carlo methods," *J. Roy. Statist. Soc.: Ser. B (Statist. Methodol.)*, vol. 73, pp. 123–214, 2011.

[21] M. Betancourt, F. Nielsen and F. Barbaresco, Eds., "A general metric for Riemannian manifold Hamiltonian Monte Carlo," in *Proc. Nat. Conf. Geometric Sci. Inf.*, 2013, vol. 8085, Lecture Notes in Computer Science, pp. 327–334.

[22] Y. Atchadé, "An adaptive version for the Metropolis adjusted Langevin algorithm with a truncated drift," *Methodol. Comput. Appli. Probabil.*, vol. 8, no. 2, pp. 235–254, 2006.

[23] N. S. Pillai, A. M. Stuart, and A. H. Thiéry, "Optimal scaling and diffusion limits for the Langevin algorithm in high dimensions," *Annals Appl. Probabil.*, vol. 22, no. 6, pp. 2320–2356, 2012.

[24] M. Pereyra, "Proximal Markov chain Monte Carlo algorithms," *Statist. Comput.*, 2015, to be published.

[25] T. Xifara, C. Sherlock, S. Livingstone, S. Byrne, and M. Girolami, "Langevin diffusions and the Metropolis-adjusted Langevin algorithm," *Statist. Probabil. Lett.*, vol. 91, pp. 14–19, 2014.

[26] B. Casella, G. Roberts, and O. Stramer, "Stability of partially implicit Langevin schemes and their MCMC variants," *Methodol. Comput. Appl. Probabil.*, vol. 13, no. 4, pp. 835–854, 2011.

[27] A. Schreck, G. Fort, S. L. Corff, and E. Moulines, "A shrinkage-thresholding metropolis adjusted Langevin algorithm for Bayesian variable selection ArXiv, 2013 [Online]. Available: arXiv:1312.5658, to be published

[28] R. Neal, "MCMC using Hamiltonian dynamics," in *Handbook of Markov Chain Monte Carlo*, S. Brooks, A. Gelman, G. Jones, and X. Meng, Eds. Boca Raton, FL, USA: Chapman & Hall/CRC Press, 2013, pp. 113–162.

[29] M. J. Betancourt, S. Byrne, S. Livingstone, and M. Girolami, "The geometric foundations of Hamiltonian Monte Carlo," *ArXiv E-Prints*, 2014.

[30] A. Beskos, N. Pillai, G. Roberts, J.-M. Sanz-Serna, and A. Stuart, "Optimal tuning of the hybrid Monte Carlo algorithm," *Bernoulli*, vol. 19, no. 5A, pp. 1501–1534, 2013.

[31] Y. Zhang and C. A. Sutton, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds., "Quasi-Newton methods for Markov chain Monte Carlo," in *Adv. Neural Inf. . Process. Syst. 24 (NIPS'11)*, 2011, pp. 2393–2401.

[32] M. D. Hoffman and A. Gelman, "The no-u-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo," *J. Mach. Learn. Res.*, vol. 15, pp. 1593–1623, 2014.

[33] M. Betancourt, S. Byrne, and M. Girolami, Optimizing the Integrator Step Size for Hamiltonian Monte Carlo ArXiv, 2014 [Online]. Available: arXiv:1411.6669, to be published

[34] K. Latuszyński, G. Roberts, and J. Rosenthal, "Adaptive Gibbs samplers and related MCMC methods," *Ann. Appl. Probab.*, vol. 23, no. 1, pp. 66–98, 2013.

[35] C. Bazot, N. Dobigeon, J.-Y. Tourneret, A. K. Zaas, G. S. Ginsburg, and A. O. Hero, "Unsupervised Bayesian linear unmixing of gene expression microarrays," *BMC Bioinformatics*, vol. 14, no. 99, Mar. 2013.

[36] D. A. van Dyk and T. Park, "Partially collapsed Gibbs samplers: Theory and methods," *J. Amer. Statistical Association*, vol. 103, pp. 790–796, 2008.

[37] T. Park and D. A. van Dyk, "Partially collapsed Gibbs samplers: Illustrations and applications," *J. Comput. Graph. Statist.*, vol. 103, pp. 283–305, 2009.

[38] G. Kail, J.-Y. Tourneret, F. Hlawatsch, and N. Dobigeon, "Blind deconvolution of sparse pulse sequences under a minimum distance constraint: A partially collapsed Gibbs sampler method," *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 2727–2743, Jun. 2012.

[39] D. A. van Dyk and X. Jiao, "Metropolis-Hastings within partially collapsed Gibbs samplers," *J. of Computational and Graphical Statistics*, to be published.

[40] M. Jordan, Z. Gharamani, T. Jaakkola, and L. Saul, "An introduction to variational methods for graphical models," *Mach. Learn.*, vol. 37, pp. 183–233, 1999.

[41] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2007.

[42] R. Weinstock, *Calculus of Variations*. New York, NY, USA: Dover, 1974.

[43] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. New York, NY, USA: Wiley, 2006.

[44] C. Peterson and J. R. Anderson, "A mean field theory learning algorithm for neural networks," *Complex Syst.*, vol. 1, pp. 995–1019, 1987.

[45] G. Parisi, *Statistical Field Theory*. Reading, MA, USA: Addison-Wesley, 1988.

[46] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Found. Trends Mach. Learn.*, vol. 1, May 2008.

[47] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2282–2312, Jul. 2005.

[48] A. L. Yuille and A. Rangarajan, "The concave-convex procedure (CCCP)," in *Proc. Neural Inf. Process. Syst. Conf.*, Vancouver, BC, Canada, 2002, vol. 2, pp. 1033–1040.

[49] R. G. Gallager, "Low density parity check codes," *IRE Trans. Inf. Theory*, vol. 8, pp. 21–28, Jan. 1962.

[50] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. San Mateo, CA, USA: Morgan Kaufman, 1988.

[51] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

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

[53] G. F. Cooper, "The computational complexity of probabilistic inference using Bayesian belief networks," *Artif. Intell.*, vol. 42, pp. 393–405, 1990.

[54] K. P. Murphy, Y. Weiss, and M. I. Jordan, "Loopy belief propagation for approximate inference: An empirical study," in *Proc. Uncert. Artif. Intell.*, 1999, pp. 467–475.

[55] R. J. McEliece, D. J. C. MacKay, and J.-F. Cheng, "Turbo decoding as an instance of Pearl's 'belief propagation' algorithm," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 140–152, Feb. 1998.

[56] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. New York, NY, USA: Cambridge Univ. Press, 2003.

[57] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, "Learning low-level vision," *Int. J. Comput. Vis.*, vol. 40, no. 1, pp. 25–47, Oct. 2000.

[58] J. Boutros and G. Caire, "Iterative multiuser joint decoding: Unified framework and asymptotic analysis," *IEEE Trans. Inf. Theory*, vol. 48, no. 7, pp. 1772–1793, Jul. 2002.

[59] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: I. Motivation and construction," in *Proc. Inf. Theory Workshop*, Cairo, Egypt, Jan. 2010, pp. 1–5.

[60] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," in *Proc. IEEE Int. Symp. Inf. Theory*, St. Petersburg, Russia, Aug. 2011, pp. 2168–2172.

[61] D. Bertsekas, *Nonlinear Programming*, 2nd ed. Nashua, NH, USA: Athena Scientific, 1999.

[62] T. Heskes, M. Opper, W. Wiegerinck, O. Winther, and O. Zoeter, "Approximate inference techniques with expectation constraints," *J. Statist. Mech.*, vol. P11015, 2005.

[63] T. Heskes, O. Zoeter, and W. Wiegerinck, "Approximate expectation maximization," in *Proc. Neural Inf. Process. Syst. Conf.*, Vancouver, BC, Canada, 2004, pp. 353–360.

[64] T. Minka, "A family of approximate algorithms for Bayesian inference," Ph.D. dissertation, MIT, Cambridge, MA, 2001.

[65] M. Seeger, "Expectation propagation for exponential families," EPFL, 2005, Tech. Rep. 161464.

[66] M. Opper and O. Winther, "Expectation consistent free energies for approximate inference," in *Proc. Neural Inf. Process. Syst. Conf.*, Vancouver, BC, Canada, 2005, pp. 1001–1008.

[67] T. Heskes and O. Zoeter, "Expectation propagation for approximate inference in dynamic Bayesian networks," in *Proc. Uncert. Artif. Intell.*, Edmonton, AB, Canada, 2002, pp. 313–320.

[68] M. Bayati and A. Montanari, "The dynamics of message passing on dense graphs, with applications to compressed sensing," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 764–785, Feb. 2011.

[69] A. Javanmard and A. Montanari, "State evolution for general approximate message passing algorithms, with applications to spatial coupling," *Inf. Inference*, vol. 2, no. 2, pp. 115–144, 2013.

[70] S. Rangan, P. Schniter, E. Riegler, A. Fletcher, and V. Cevher, "Fixed points of generalized approximate message passing with arbitrary matrices," in *Proc. IEEE Int. Symp. Inf. Theory*, Istanbul, Turkey, Jul. 2013, pp. 664–668.

[71] F. Krzakala, A. Manoel, E. W. Tramel, and L. Zdeborová, "Variational free energies for compressed sensing," in *Proc. IEEE Int. Symp. Inf. Theory*, Honolulu, HI, USA, Jul. 2014, pp. 1499–1503.

[72] J. P. Vila and P. Schniter, "An empirical-Bayes approach to recovering linearly constrained non-negative sparse signals," *IEEE Trans. Signal Process.*, vol. 62, no. 18, pp. 4689–4703, Sep. 2014.

[73] S. Rangan, P. Schniter, and A. Fletcher, "On the convergence of generalized approximate message passing with arbitrary matrices," in *Proc. IEEE Int. Symp. Inf. Theory*, Honolulu, HI, USA, Jul. 2014, pp. 236–240.

[74] J. Vila, P. Schniter, S. Rangan, F. Krzakala, and L. Zdeborová, "Adaptive damping and mean removal for the generalized approximate message passing algorithm," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Brisbane, Australia, 2015, pp. 2021–2025.

[75] S. Rangan, A. K. Fletcher, P. Schniter, and U. Kamilov, "Inference for feneralized linear models via alternating directions and Bethe free energy minimization," 2015 [Online]. Available: arXiv:1501:01797

[76] L. Bottou and Y. LeCun, "Large scale online learning," in *Proc. Annu. Conf. Neur. Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 13–18, 2004, pp. x–x+8.

[77] *Optimization for Machine Learning*, S. Sra, S. Nowozin, and S. J. Wright, Eds. Cambridge, MA, USA: MIT Press, 2011.

[78] S. Theodoridis, *Machine Learning: A Bayesian and Optimization Perspective*. San Diego, CA, USA: Academic, 2015.

[79] S. O. Haykin, *Adaptive Filter Theory*, 4th ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2002.

[80] A. H. Sayed, *Adaptive Filters*. Hoboken, NJ, USA: Wiley, 2011.

[81] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, H. H. Bauschke, R. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, Eds. New York, NY, USA: Springer-Verlag, 2010, pp. 185–212.

[82] J. Duchi and Y. Singer, "Efficient online and batch learning using forward backward splitting," *J. Mach. Learn. Res.*, vol. 10, pp. 2899–2934, 2009.

[83] Y. F. Atchadé, G. Fort, and E. Moulines, "On stochastic proximal gradient algorithms," 2014 [Online]. Available: http://arxiv.org/abs/1402.2365

[84] L. Rosasco, S. Villa, and B. C. Vũ, "A stochastic forward-backward splitting method for solving monotone inclusions in Hilbert spaces," 2014 [Online]. Available: http://arxiv.org/abs/1403.7999

[85] P. L. Combettes and J.-C. Pesquet, "Stochastic quasi-Fejér block-coordinate fixed point iterations with random sweeping," *SIAM J. Optim.*, vol. 25, no. 2, pp. 1221–1248, 2015.

[86] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, pp. 400–407, 1951.

[87] J. M. Ermoliev and Z. V. Nekrylova, "The method of stochastic gradients and its application," in *Seminar: Theory of Optimal Solutions. No. 1 (Russian)*, Kiev, Ukraine, 1967, pp. 24–47.

[88] O. V. Guseva, "The rate of convergence of the method of generalized stochastic gradients," *Kibernetika (Kiev)*, no. 4, pp. 143–145, 1971.

[89] D. P. Bertsekas and J. N. Tsitsiklis, "Gradient convergence in gradient methods with errors," *SIAM J. Optim.*, vol. 10, no. 3, pp. 627–642, 2000.

[90] F. Bach and E. Moulines, "Non-asymptotic analysis of stochastic approximation algorithms for machine learning," in *Proc. Ann. Conf. Neur. Inf. Proc. Syst.*, Granada, Spain, Dec. 12–15, 2011, pp. 451–459.

[91] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM J. Optim.*, vol. 19, no. 4, pp. 1574–1609, 2008.

[92] B. T. Polyak and A. B. Juditsky, "Acceleration of stochastic approximation by averaging," *SIAM J. Control Optim.*, vol. 30, no. 4, pp. 838–855, Jul. 1992.

[93] S. Shalev-Shwartz, Y. Singer, and N. Srebro, "Pegasos: Primal estimated sub-gradient solver for SVM," in *Proc. 24th Int. Conf. Mach. Learn.*, Corvalis, OR, USA, Jun. 20–24, 2007, pp. 807–814.

[94] L. Xiao, "Dual averaging methods for regularized stochastic learning and online optimization," *J. Mach. Learn. Res.*, vol. 11, pp. 2543–2596, Dec. 2010.

[95] J. Duchi, S. Shalev-Shwartz, Y. Singer, and A. Tewari, "Composite objective mirror descent," in *Proc. Conf. Learn. Theory*, Haifa, Israel, Jun. 27–29, 2010, pp. 14–26.

[96] L. W. Zhong and J. T. Kwok, "Accelerated stochastic gradient method for composite regularization," *J. Mach. Learn. Rech.*, vol. 33, pp. 1086–1094, 2014.

[97] H. Ouyang, N. He, L. Tran, and A. Gray, "Stochastic alternating direction method of multipliers," in *Proc. 30th Int. Conf. Mach. Learn.*, Atlanta, GA, USA, Jun. 16–21, 2013, pp. 80–88.

[98] T. Suzuki, "Dual averaging and proximal gradient descent for online alternating direction multiplier method," in *Proc. 30th Int. Conf. Mach. Learn.*, Atlanta, GA, USA, Jun. 16–21, 2013, pp. 392–400.

[99] W. Zhong and J. Kwok, "Fast stochastic alternating direction method of multipliers," Tech. Rep. Beijing, China, 2014.

[100] Y. Xu and W. Yin, "Block stochastic gradient iteration for convex and nonconvex optimization," 2014 [Online]. Available: http://arxiv.org/pdf/1408.2597v2.pdf

[101] C. Hu, J. T. Kwok, and W. Pan, "Accelerated gradient methods for stochastic optimization and online learning," in *Proc. Ann. Conf. Neur. Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 11–12, 2009, pp. 781–789.

[102] G. Lan, "An optimal method for stochastic composite optimization," *Math. Program.*, vol. 133, no. 1-2, pp. 365–397, 2012.

[103] Q. Lin, X. Chen, and J. Peña, "A sparsity preserving stochastic gradient method for composite optimization," *Comput. Optim. Appl.*, vol. 58, no. 2, pp. 455–482, Jun. 2014.

[104] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Jul. 2011.

[105] S.-I. Amari, "Natural gradient works efficiently in learning," *Neural Comput.*, vol. 10, no. 2, pp. 251–276, Feb. 1998.

[106] E. Chouzenoux, J.-C. Pesquet, and A. Florescu, "A stochastic 3MG algorithm with application to 2D filter identification," in *Proc. 22nd Eur. Signal Process. Conf.*, Lisbon, Portugal, Sep. 1–5, 2014, pp. 1587–1591.

[107] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1985.

[108] H. J. Kushner and G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*, 2nd ed. New York, NY, USA: Springer-Verlag, 2003.

[109] S. L. Gay and S. Tavathia, "The fast affine projection algorithm," in *Proc. Int. Conf. Acoust., Speech Signal Process.*, Detroit, MI, USA, May 9–12, 1995, vol. 5, pp. 3023–3026.

[110] A. W. H. Khong and P. A. Naylor, "Efficient use of sparse adaptive filters," in *Proc. Asilomar Conf. Signal, Systems and Computers*, Pacific Grove, CA, USA, Oct.-Nov. 29–1, 2006, pp. 1375–1379.

[111] C. Paleologu, J. Benesty, and S. Ciochina, "An improved proportionate NLMS algorithm based on the l_0 norm," in *Proc. Int. Conf. Acoust., Speech Signal Process.*, Dallas, TX, USA, Mar. 14–19, 2010.

[112] O. Hoshuyama, R. A. Goubran, and A. Sugiyama, "A generalized proportionate variable step-size algorithm for fast changing acoustic environments," in *Proc. Int. Conf. Acoust., Speech Signal Process.*, Montreal, QC, Canada, May 17–21, 2004, vol. 4, pp. iv–161–iv–164.

[113] C. Paleologu, J. Benesty, and F. Albu, "Regularization of the improved proportionate affine projection algorithm," in *Proc. Int. Conf. Acoust., Speech Signal Process.*, Kyoto, Japan, Mar. 25–30, 2012, pp. 169–172.

[114] Y. Chen, Y. Gu, and A. O. Hero, "Sparse LMS for system identification," in *Proc. Int. Conf. Acoust., Speech Signal Process.*, Taipei, Taiwan, Apr. 19–24, 2009, pp. 3125–3128.

[115] Y. Chen, Y. Gu, and A. O. Hero, "Regularized least-mean-square algorithms," 2010 [Online]. Available: http://arxiv.org/abs/1012.5066

[116] R. Meng, R. C. De Lamare, and V. H. Nascimento, "Sparsity-aware affine projection adaptive algorithms for system identification," in *Proc. Sensor Signal Process. Defence*, London, U.K., Sep. 27–29, 2011, pp. 1–5.

[117] L. V. S. Markus, W. A. Martins, and P. S. R. Diniz, "Affine projection algorithms for sparse system identification," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Vancouver, BC, Canada, May 26–31, 2013, pp. 5666–5670.

[118] L. V. S. Markus, T. N. Tadeu, N. Ferreira, W. A. Martins, and P. S. R. Diniz, "Sparsity-aware data-selective adaptive filters," *IEEE Trans. Signal Process.*, vol. 62, no. 17, pp. 4557–4572, Sep. 2014.

[119] Y. Murakami, M. Yamagishi, M. Yukawa, and I. Yamada, "A sparse adaptive filtering using time-varying soft-thresholding techniques," in *Proc. Int. Conf. Acoust., Speech Signal Process.*, Dallas, TX, USA, Mar. 14–19, 2010, pp. 3734–3737.

[120] M. Yamagishi, M. Yukawa, and I. Yamada, "Acceleration of adaptive proximal forward-backward splitting method and its application to sparse system identification," in *Proc. Int. Conf. Acoust., Speech Signal Process.*, Prague, Czech Republic, May 22–27, 2011, pp. 4296–4299.

[121] S. Ono, M. Yamagishi, and I. Yamada, "A sparse system identification by using adaptively-weighted total variation via a primal-dual splitting approach," in *Proc. Int. Conf. Acoust., Speech Signal Process.*, Vancouver, BC, Canada, May 26–31, 2013, pp. 6029–6033.

[122] D. Angelosante, J. A. Bazerque, and G. B. Giannakis, "Online adaptive estimation of sparse signals: Where RLS meets the ℓ_1-norm," *IEEE Trans. Signal Process.*, vol. 58, no. 7, pp. 3436–3447, Jul. 2010.

[123] B. Babadi, N. Kalouptsidis, and V. Tarokh, "SPARLS: The sparse RLS algorithm," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4013–4025, Aug. 2010.

[124] Y. Kopsinis, K. Slavakis, and S. Theodoridis, "Online sparse system identification and signal reconstruction using projections onto weighted ℓ_1 balls," *IEEE Trans. Signal Process.*, vol. 59, no. 3, pp. 936–952, Mar. 2011.

[125] K. Slavakis, Y. Kopsinis, S. Theodoridis, and S. McLaughlin, "Generalized thresholding and online sparsity-aware learning in a union of subspaces," *IEEE Trans. Signal Process.*, vol. 61, no. 15, pp. 3760–3773, Aug. 2013.

[126] J. Mairal, "Incremental majorization-minimization optimization with application to large-scale machine learning," *SIAM J. Optim.*, vol. 25, no. 2, pp. 829–855, 2015.

[127] A. Repetti, E. Chouzenoux, and J.-C. Pesquet, "A parallel block-coordinate approach for primal-dual splitting with arbitrary random block selection," in *Proc. Eur. Signal Process. Conf. (EUSIPCO'15)*, Nice, France, Aug. – Sep. 31 – 4, 2015.
[128] D. Blatt, A. O. Hero, and H. Gauchman, "A convergent incremental gradient method with a constant step size," *SIAM J. Optim.*, vol. 18, no. 1, pp. 29–51, 1998.
[129] D. P. Bertsekas, "Incremental gradient, subgradient, and proximal methods for convex optimization: A survey," 2010 [Online]. Available: http://www.mit.edu/~dimitrib/Incremental_Survey_LIDS.pdf
[130] M. Schmidt, N. Le Roux, and F. Bach, 'Minimizing finite sums with the stochastic average gradient," 2013 [Online]. Available: http://arxiv.org/abs/1309.2388
[131] A. Defazio, J. Domke, and T. Cartano, "Finito: A faster, permutable incremental gradient method for big data problems," in *Proc. 31st Int. Conf. Mach. Learn.*, Beijing, China, Jun. 21–26, 2014, pp. 1125–1133.
[132] A. Defazio, F. Bach, and S. Lacoste, "SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives," in *Proc. Ann. Conf. Neur. Inf. Proc. Syst.*, Montreal, QC, Canada, Dec. 8–11, 2014, pp. 1646–1654.
[133] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Proc. Annu. Conf. Neur. Inf. Process. Syst.*, Lake Tahoe, NV, USA, Dec. 5–10, 2013, pp. 315–323.
[134] J. Konečný, J. Liu, P. Richtárik, and M. Takáč, "mS2GD: Mini-batch semi-stochastic gradient descent in the proximal setting," 2014 [Online]. Available: http://arxiv.org/pdf/1410.4744v1.pdf
[135] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *J. Optim. Theory Appl.*, vol. 109, no. 3, pp. 475–494, 2001.
[136] E. Chouzenoux, J.-C. Pesquet, and A. Repetti, "A block coordinate variable metric forward-backward algorithm," 2014 [Online]. Available: http://www.optimization-online.org/DB_HTML/2013/12/4178.html
[137] J.-C. Pesquet and A. Repetti, "A class of randomized primal-dual algorithms for distributed optimization," *J. Nonlinear Convex Anal.*, 2015, to be published.
[138] N. Komodakis and J.-C. Pesquet, "Playing with duality: An overview of recent primal-dual approaches for solving large-scale optimization problems," *IEEE Signal Process. Mag.*, vol. 32, no. 6, pp. 31–54, Nov. 2014.
[139] P. Richtárik and M. Takáč, "Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function," *Math. Program.*, vol. 144, pp. 1–38, 2014.
[140] I. Necoara and A. Patrascu, "A random coordinate descent algorithm for optimization problems with composite objective function and linear coupled constraints," *Comput. Optim. Appl.*, vol. 57, pp. 307–337, 2014.
[141] Z. Lu and L. Xiao, "On the complexity analysis of randomized block-coordinate descent methods," *Math. Program.*, 2015, to be published.
[142] O. Fercoq and P. Richtárik, "Accelerated, parallel and proximal coordinate descent," 2014 [Online]. Available: http://arxiv.org/pdf/1312.5799v2.pdf
[143] Z. Qu and P. Richtárik, "Coordinate descent with arbitrary sampling I: Algorithms and complexity," 2014 [Online]. Available: http://arxiv.org/abs/1412.8060
[144] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems," *SIAM J. Optim.*, pp. 341–362, 2012.
[145] P. L. Combettes, D. Dũng, and B. C. Vũ, "Dualization of signal recovery problems," *Set-Valued Var. Anal.*, vol. 18, pp. 373–404, Dec. 2010.
[146] S. Shalev-Shwartz and T. Zhang, "Stochastic dual coordinate ascent methods for regularized loss minimization," *J. Mach. Learn. Res..*, vol. 14, pp. 567–599, 2013.
[147] S. Shalev-Shwartz and T. Zhang, "Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization," *Math. Program. Ser. A*, Nov. 2014, to be published.
[148] M. Jaggi, V. Smith, M. Takáč, J. Terhorst, S. Krishnan, T. Hofmann, and M. I. Jordan, "Communication-efficient distributed dual coordinate ascent," in *Proc. Annu Conf. Neur. Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 8–11, 2014, pp. 3068–3076.
[149] Z. Qu, P. Richtárik, and T. Zhang, "Randomized dual coordinate ascent with arbitrary sampling," 2014 [Online]. Available: http://arxiv.org/abs/1411.5873
[150] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem, "Asynchronous distributed optimization using a randomized alternating direction method of multipliers," in *Proc. 52nd Conf. Decision Control*, Florence, Italy, Dec. 10–13, 2013, pp. 3671–3676.
[151] P. Bianchi, W. Hachem, and F. Iutzeler, "A stochastic coordinate descent primal-dual algorithm and applications to large-scale composite optimization," 2014 [Online]. Available: http://arxiv.org/abs/1407.0898
[152] S. Allassonniere and E. Kuhn, "Convergent stochastic expectation maximization algorithm with efficient sampling in high dimension. Application to deformable template model estimation," *ArXiv E-Prints*, Jul. 2012.
[153] Y. Marnissi, A. Benazza-Benyahia, E. Chouzenoux, and J.-C. Pesquet, "Majorize-minimize adapted Metropolis Hastings algorithm. Application to multichannel image recovery," in *Proc. Eur. Signal Process. Conf. (EUSIPCO'14)*, Lisbon, Portugal, Sep. 1–5, 2014, pp. 1332–1336.
[154] J. J. Moreau, "Fonctions convexes duales et points proximaux dans un espace Hilbertien," *C. R. Acad. Sci. Paris Sér. A*, vol. 255, pp. 2897–2899, 1962.
[155] L. Chaari, J.-Y. Tourneret, C. Chaux, and H. Batatia, "A Hamiltonian Monte Carlo method for non-smooth energy sampling," *ArXiv E-Prints*, Jan. 2014.
[156] H. Rue, "Fast sampling of Gaussian Markov random fields," *J. Roy. Statist. Soc. Ser. B*, vol. 63, no. 2, pp. 325–338, 2001.
[157] W. F. Trench, "An algorithm for the inversion of finite Toeplitz matrices," *J. Soc. Ind. Appl. Math.*, vol. 12, no. 3, pp. 515–522, 1964.
[158] D. Geman and C. Yang, "Nonlinear image recovery with half-quadratic regularization," *IEEE Trans. Image Process.*, vol. 4, no. 7, pp. 932–946, Jul. 1995.
[159] G. Papandreou and A. L. Yuille, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds., "Gaussian sampling by local perturbations," in *Adv. Neural Inf. Process. Syst. 23 (NIPS'10)*, 2010, pp. 1858–1866.
[160] F. Orieux, O. Feron, and J.-F. Giovannelli, "Sampling high-dimensional Gaussian distributions for general linear inverse problems," *IEEE Signal Process. Lett.*, vol. 19, no. 5, pp. 251–254, May 2012.
[161] C. Gilavert, S. Moussaoui, and J. Idier, "Efficient Gaussian sampling for solving large-scale inverse problems using MCMC," *IEEE Trans. Signal Process.*, vol. 63, no. 1, pp. 70–80, Jan. 2015.

Marcelo Pereyra (S'09–M'13) received the M.Eng. degree in electronic engineering from both ITBA, Argentina, and INSA Toulouse, France, in June 2009, the M.Sc. degree in electronic engineering and control theory from INSA Toulouse in September 2009, and the Ph.D. degree in signal processing from the Institut National Polytechnique de Toulouse, University of Toulouse, in July 2012. He currently holds a Marie Curie Intra-European Fellowship for Career Development at the School of Mathematics of the University of Bristol. He is also the recipient of a Brunel Postdoctoral Research Fellowship in Statistics, a Postdoctoral Research Fellowship from French Ministry of Defence, a Leopold Escande Ph.D. Thesis excellent award from the University of Toulouse (2012), an INFOTEL R&D excellent award from the Association of Engineers of INSA Toulouse (2009), and an ITBA R&D excellence award (2007). His research activities are centred around statistical image processing, with a particular interest in Bayesian analysis and computation for high-dimensional inverse problems.

Philip Schniter (F'14) received the B.S. and M.S. degrees in electrical engineering from the University of Illinois at Urbana-Champaign in 1992 and 1993, respectively, and the Ph.D. degree in electrical engineering from Cornell University in Ithaca, NY, in 2000. From 1993 to 1996, he was employed by Tektronix Inc. in Beaverton, OR, as a Systems Engineer. After receiving the Ph.D. degree, he joined the Department of Electrical and Computer Engineering at The Ohio State University, Columbus, where he is currently a Professor and a member of the Information Processing Systems (IPS) Lab. In 2008–2009 he was a Visiting Professor at Eurecom, Sophia Antipolis, France, and Supélec, Gif-sur-Yvette, France. In 2003, Dr. Schniter received the National Science Foundation CAREER Award. His areas of interest currently include statistical signal processing, wireless communications and networks, and machine learning.

Émilie Chouzenoux (S'08–M'11) received the engineering degree from École Centrale, Nantes, France, in 2007 and the Ph.D. degree in signal processing from the Institut de Recherche en Communications et Cybernétique, Nantes in 2010. Since 2011, she has been an Assistant Professor at the University of Paris-Est Marne-la-Vallée, Champs-sur-Marne, France (LIGM, UMR CNRS 8049). Her research interests are in convex and nonconvex optimization algorithms for large scale inverse problems of image and signal processing.

Jean-Christophe Pesquet (S'89–M'91–SM'99–F'12) received the engineering degree from Supélec, Gif-sur-Yvette, France, in 1987, the Ph.D. degree from the Université Paris-Sud (XI), Paris, France, in 1990, and the Habilitation à Diriger des Recherches from the Université Paris-Sud in 1999.

From 1991 to 1999, he was a Maître de Conférences at the Université Paris-Sud, and a Research Scientist at the Laboratoire des Signaux et Systèmes, Centre National de la Recherche Scientifique (CNRS), Gif-sur-Yvette. He is currently a Professor (classe exceptionnelle) with Université de Paris-Est Marne-la-Vallée, France, and the Deputy Director of the Laboratoire d'Informatique of the university (UMR-CNRS 8049).

His research interests include multiscale analysis, statistical signal processing, inverse problems, and optimization methods with applications to imaging.

Jean-Yves Tourneret (SM'08) received the ingénieur degree in electrical engineering from the Ecole Nationale Supérieure d'Electronique, d'Electrotechnique, d'Informatique, d'Hydraulique et des Télécommunications (ENSEEIHT) de Toulouse in 1989 and the Ph.D. degree from the National Polytechnic Institute from Toulouse in 1992. He is currently a Professor in the University of Toulouse (ENSEEIHT) and a member of the IRIT Laboratory (UMR 5505 of the CNRS). His research activities are centered around statistical signal and image processing with a particular interest to Bayesian and Markov chain Monte Carlo (MCMC) methods. He has been involved in the organization of several conferences including the European conference on signal processing EUSIPCO'02 (program chair), the international conference ICASSP'06 (plenaries), the statistical signal processing workshop SSP'12 (international liaisons), the International Workshop on Computational Advances in Multi-Sensor Adaptive Processing CAMSAP 2013 (local arrangements), the statistical signal processing workshop SSP2014 (special sessions), the workshop on machine learning for signal processing MLSP2014 (special sessions). He has been the general chair of the CIMI workshop on optimization and statistics in image processing hold in Toulouse in 2013 (with F. Malgouyres and D. Kouamé) and of the International Workshop on Computational Advances in Multi-Sensor Adaptive Processing CAMSAP 2015 (with P. Djuric). He has been a member of different technical committees including the Signal Processing Theory and Methods (SPTM) committee of the IEEE Signal Processing Society (2001–2007, 2010–present). He has been serving as an associate editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING (2008–2011, 2015–present) and for the *EURASIP Journal on Signal Processing* (2013–present).

Alfred O. Hero, III (S'79–M'84–SM'98–F'98) received the B.S. (summa cum laude) from Boston University (1980) and the Ph.D. from Princeton University (1984), both in electrical engineering. Since 1984, he has been with the University of Michigan, Ann Arbor, where he is the R. Jamison and Betty Williams Professor of Engineering and co-director of the Michigan Institute for Data Science (MIDAS) . His primary appointment is in the Department of Electrical Engineering and Computer Science and he also has appointments, by courtesy, in the Department of Biomedical Engineering and the Department of Statistics. From 2008 to 2013, he held the Digiteo Chaire d'Excellence, sponsored by Digiteo Research Park in Paris, located at the Ecole Superieure d'Electricite, Gif-sur-Yvette, France. He has held other visiting positions at LIDS Massachusetts Institute of Technology (2006), Boston University (2006), I3S University of Nice, Sophia-Antipolis, France (2001), Ecole Normale Supérieure de Lyon (1999), Ecole Nationale Supérieure des Télécommunications, Paris (1999), Lucent Bell Laboratories (1999), Scientific Research Labs of the Ford Motor Company, Dearborn, Michigan (1993), Ecole Nationale Superieure des Techniques Avancees (ENSTA), Ecole Superieure d'Electricite, Paris (1990), and M.I.T. Lincoln Laboratory (1987–1989). He is a Fellow of the Institute of Electrical and Electronics Engineers (IEEE). He received the University of Michigan Distinguished Faculty Achievement Award (2011). He has been plenary and keynote speaker at several workshops and conferences. He has received several best paper awards including: an IEEE Signal Processing Society Best Paper Award (1998), a Best Original Paper Award from the Journal of Flow Cytometry (2008), a Best Magazine Paper Award from the IEEE Signal Processing Society (2010), a SPIE Best Student Paper Award (2011), an IEEE ICASSP Best Student Paper Award (2011), an AISTATS Notable Paper Award (2013), and an IEEE ICIP Best Paper Award (2013). He received an IEEE Signal Processing Society Meritorious Service Award (1998), an IEEE Third Millennium Medal (2000), an IEEE Signal Processing Society Distinguished Lecturership (2002), and an IEEE Signal Processing Society Technical Achievement Award (2014). He was President of the IEEE Signal Processing Society (2006–2007). He was a member of the IEEE TAB Society Review Committee (2009), the IEEE Awards Committee (2010–2011), and served on the Board of Directors of the IEEE (2009–2011) as Director of Division IX (Signals and Applications). He served on the IEEE TAB Nominations and Appointments Committee (2012–2014). He is currently a member of the Big Data Special Interest Group (SIG) of the IEEE Signal Processing Society. Since 2011 he has been a member of the Committee on Applied and Theoretical Statistics (CATS) of the US National Academies of Science.

His recent research interests are in the data science of high dimensional spatio-temporal data, statistical signal processing, and machine learning. Of particular interest are applications to networks, including social networks, multi-modal sensing and tracking, database indexing and retrieval, imaging, biomedical signal processing, and biomolecular signal processing.

Steve McLaughlin (F'11) was born in Clydebank Scotland, in 1960. He received the B.Sc. degree in electronics and electrical engineering from the University of Glasgow in 1981 and the Ph.D. degree from the University of Edinburgh in 1990. From 1981 to 1984, he was a Development Engineer in industry involved in the design and simulation of integrated thermal imaging and fire control systems. From 1984 to 1986, he worked on the design and development of high-frequency data communication systems. In 1986, he joined the Dept. of Electronics and Electrical Engineering at the University of Edinburgh as a Research Fellow where he studied the performance of linear adaptive algorithms in high noise and non-stationary environments. In 1988, he joined the academic staff at Edinburgh, and from 1991 until 2001, he held a Royal Society University Research Fellowship to study nonlinear signal processing techniques. In 2002, he was awarded a personal Chair in Electronic Communication Systems at the University of Edinburgh. In October 2011, he joined Heriot-Watt University as a Professor of Signal Processing and Head of the School of Engineering and Physical Sciences. His research interests lie in the fields of adaptive signal processing and nonlinear dynamical systems theory and their applications to biomedical , energy and communication systems. Prof. McLaughlin is a Fellow of the Royal Academy of Engineering, of the Royal Society of Edinburgh, of the Institute of Engineering and Technology and of the IEEE.