IEEE
SIGNAL PROCESSING LETTERS
A PUBLICATION OF THE IEEE SIGNAL PROCESSING SOCIETY
www.ieee.org/sp/index.html
IEEE Xplore® Digital Library
http://ieeexplore.ieee.org/Xplore

Stochastic Gradient Population Monte Carlo

Yousef El-Laham and Mónica Bugallo
Department of Electrical & Computer Engineering, Stony Brook University, Stony Brook (USA)
{yousef.ellaham, monica.bugallo}@stonybrook.edu

***Abstract*—The population Monte Carlo (PMC) algorithm is a powerful adaptive importance sampling (AIS) methodology used for estimating expected values of random quantities w.r.t. some target probability distribution. At each iteration, a Markov transition kernel is used to propagate a set of particles. Importance weights of the particles are computed and then used to resample the particles that are most representative of the target distribution. At the end of the algorithm, the set of all particles and weights can be used to perform estimation. The resampling step is an adaptive mechanism of the PMC algorithm that allows for particles to locate the most significant regions of the sampling space. In this paper, we generalize the adaptation procedure of PMC sampling by providing a perspective based on stochastic optimization rather than resampling. The proposed method is more flexible than standard PMC as it allows the parameter adaptation to be resolved using any stochastic optimization method. We show that under certain conditions, the standard PMC algorithm is a special case of the proposed approach.**

I. Introduction

Monte Carlo (MC) methods are computational schemes that use random sampling to approximate intractable integrals [1]. While these methods can be used in a variety of settings, they are mostly employed as tools in Bayesian statistical inference, where the goal is to approximate expectations of functions w.r.t. the posterior probability distribution of some unknown parameter in a probabilistic model (also called *target distribution*) [2], [3]. Markov chain Monte Carlo (MCMC) sampling schemes are arguably the most popular and well-studied MC methods for Bayesian posterior inference [4], [5], [6]. MCMC generates samples from the target distribution by iteratively constructing a Markov chain whose stationary distribution is the target [7].

Importance sampling (IS) is an alternative MC method that can be used to approximate posterior expectations [1]. It generates samples from a different distribution, called the *proposal distribution*, and then weighs those samples according to the ratio of the target and the proposal. The set of weighted samples are used to approximate the desired posterior expectations. Unfortunately, IS estimators have high variance whenever there is large mismatch between the target and proposal distribution [8]. This makes the choice of the proposal distribution the key challenge for implementation of IS methods. For this reason, there have been large efforts devoted to the design of adaptive importance sampling (AIS) schemes [9], which are iterative implementations of IS that adapt the proposal distribution at each iteration.

This work was supported by the National Science Foundation (NSF) (CCF-1617986). The Research Computing and Cyberinfrastructure and the Institute for Advanced Computational Science at Stony Brook University allowed for use of the high-performance SeaWulf system (NSF OAC-1531492).

Digital Object Identifier 10.1109/LSP.2019.2954048

Population Monte Carlo (PMC) sampling is a popular AIS method [10], where samples called *particles* are generated from a set of proposal distributions and then weighted using a weighting scheme [11]. At the end of each iteration, a new set of particles is resampled from the discrete random measure formed by the set of weighted particles drawn in that iteration. These resampled particles are used to determine the parameters of the population of proposals in the following iteration. In its standard implementation [10], a single particle is drawn from each proposal, and the resampled particles correspond to the locations of the population of proposals in the next iteration.

Over the years, there have been many advances in the theory and implementation of PMC sampling schemes. In [12], the mixture PMC (M-PMC) was proposed, which treats the population of proposals as a mixture whose parameters (means, covariances, and mixing weights) are adapted in a way to minimize Kullback-Leibler divergence. In nonlinear PMC (N-PMC) [13], a nonlinear transformation is applied to the weights before adaptation. The nonlinear weighting allows for the particles to avoid the so-called *path degeneracy* problem that can be caused by resampling. In [11], alternative weighting and resampling schemes are proposed which improve the performance of PMC for high-dimensional distributions.

In this work, we generalize the adaptation step of PMC by using stochastic optimization, where the goal of the stochastic optimization algorithm is to minimize some function of the parameters of the set of proposal distributions. This is done using stochastic gradients, which in this work are approximated using resampled particles, unlike similar schemes proposed in the adaptive MCMC literature [14]. We show that when the goal is to satisfy a moment matching criteria, we obtain an algorithm where the standard resampling scheme of PMC sampling can be viewed as a special case.

II. Problem Formulation

Consider an unknown vector $\mathbf{x} \in \mathbb{R}^{d_x}$ and a set of observations $\mathcal{Y} \triangleq \{\mathbf{y}_n \in \mathbb{R}^{d_y}\}_{n=1}^N$, where $\mathbf{y}_n \sim p(\mathbf{y}|\mathbf{x})$ for $n = 1, \dots, N$. The posterior distribution of $\mathbf{x}$ given the set of observations $\mathcal{Y}$ can be determined by Bayes' theorem as

$$\pi(\mathbf{x}) \triangleq p(\mathbf{x}|\mathcal{Y}) = \frac{\ell(\mathcal{Y}|\mathbf{x})p(\mathbf{x})}{p(\mathcal{Y})}, \tag{1}$$

where $\ell(\mathcal{Y}|\mathbf{x}) \triangleq \prod_{n=1}^N p(\mathbf{y}_n|\mathbf{x})$ is the *likelihood* of $\mathbf{x}$, $p(\mathbf{x})$ is the *prior* of $\mathbf{x}$, and $Z \triangleq p(\mathcal{Y}) = \int_{\mathbb{R}^{d_x}} \ell(\mathcal{Y}|\mathbf{x})p(\mathbf{x})d\mathbf{x}$ is called the *evidence* or *normalization constant*. Typically, Z is intractable and we can only evaluate the unnormalized posterior

$$\tilde{\pi}(\mathbf{x}) \triangleq \ell(\mathcal{Y}|\mathbf{x})p(\mathbf{x}). \tag{2}$$

Our goal is to estimate expectations w.r.t. $\pi(\mathbf{x})$:

$$\mathcal{H} \triangleq \mathbb{E}_\pi\left[h(\mathbf{x})\right] = \int_{\mathbb{R}^{d_x}} h(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}, \tag{3}$$

where $h(\mathbf{x})$ is assumed to be integrable w.r.t. $\pi(\mathbf{x})$.

III. Brief Review

AIS methods are a class of iterative MC algorithms that can estimate integrals of the form in (3) [9]. The three basic steps of an AIS sampler at each iteration ($i = 1, \dots, I$) are: *sampling*, *weighting*, and *adaptation*. In the sampling step, a set of M particles $\{\mathbf{x}_i^{(m)}\}_{m=1}^M$ is drawn from a proposal $q(\mathbf{x}; \boldsymbol{\theta}_i)$, where $\boldsymbol{\theta}_i$ are the proposal parameters and i denotes the iteration index. The particles are then weighted according to their respective importance ratio

$$\tilde{w}_i^{(m)} = \frac{\tilde{\pi}(\mathbf{x}_i^{(m)})}{q(\mathbf{x}_i^{(m)}; \boldsymbol{\theta}_i)}, \tag{4}$$

where $\tilde{\pi}(\mathbf{x})$ is evaluated according to (2). Finally, the parameters of the proposal distribution are adapted using some update, i.e., $\boldsymbol{\theta}_i \to \boldsymbol{\theta}_{i+1}$. These steps are repeated for I iterations. At each iteration, we can obtain an approximation of (3) as

$$\hat{\mathcal{H}}_i = \sum_{\tau=1}^{i}\sum_{m=1}^{M} \bar{w}_\tau^{(m)} h(\mathbf{x}_\tau^{(m)}), \tag{5}$$

where $\bar{w}_\tau^{(m)} = \frac{\tilde{w}_\tau^{(m)}}{\sum_{k=1}^{i}\sum_{j=1}^{M}\tilde{w}_k^{(j)}}$ is the weight of the mth particle generated at the τth iteration normalized over all iterations.

One popular AIS scheme is PMC. At each iteration of PMC, R particles are drawn from L proposal distributions for a total of $M = LR$ particles. In the standard implementation, a single particle ($R = 1$) is drawn from each of the $L = M$ proposals. The parameters of the proposal are adapted using a resampling scheme. Typically, each proposal is parameterized by a mean vector $\boldsymbol{\mu}_i^{(m)}$ and covariance matrix $\boldsymbol{\Sigma}_i^{(m)}$, where only the mean parameter is adapted [10]. The standard PMC algorithm is summarized in Algorithm 1.

Algorithm 1 Population Monte Carlo (PMC)

1: **Initialization:** Set the initial means $\boldsymbol{\mu}_1^{(1)}, \dots, \boldsymbol{\mu}_1^{(M)}$. Set the covariance matrices $\boldsymbol{\Sigma}_1^{(1)}, \dots, \boldsymbol{\Sigma}_1^{(M)}$.
2: **for** $i = 1, \dots, I$ **do**
3: **Sampling**: Draw a particle from each proposal

$$\mathbf{x}_i^{(m)} \sim q(\mathbf{x}; \boldsymbol{\mu}_i^{(m)}, \boldsymbol{\Sigma}_i^{(m)}), \quad m = 1, \dots, M.$$

4: **Weighting**: Compute the importance weights

$$\tilde{w}_i^{(m)} = \frac{\tilde{\pi}(\mathbf{x}_i^{(m)})}{q(\mathbf{x}_i^{(m)}; \boldsymbol{\mu}_i^{(m)}, \boldsymbol{\Sigma}_i^{(m)})}, \quad m = 1, \dots, M,$$

and normalize them as

$$w_i^{(m)} = \frac{\tilde{w}_i^{(m)}}{\sum_{j=1}^{M}\tilde{w}_i^{(j)}}, \quad m = 1, \dots, M.$$

5: **Adaptation:** Draw M indices from the multinomial distribution

$$s_m \sim \text{Multi}\left(M, \{w_i^{(j)}\}_{j=1}^M\right), \quad m = 1, \dots, M.$$

Set $\tilde{\mathbf{x}}_i^{(m)} = \mathbf{x}_i^{(s_m)}$, and adapt parameters as

$$\boldsymbol{\mu}_{i+1}^{(m)} = \tilde{\mathbf{x}}_i^{(m)}; \ \boldsymbol{\Sigma}_{i+1}^{(m)} = \boldsymbol{\Sigma}_i^{(m)}, \quad m = 1, \dots, M.$$

6: **end for**
7: **Output**: Return $\mathcal{X}_i = \{\mathbf{x}_i^{(m)}, \tilde{w}_i^{(m)}\}_{m=1}^M$ for $i = 1, \dots, I$.

IV. Proposed Method

We generalize the adaptive mechanism of PMC by considering the update to be an instance of stochastic gradient descent (SGD). We begin by deriving the SGD update for minimizing the expected value of some function f w.r.t. the parameters of the proposal distribution. We show that for a certain choice of f, we can obtain an adaptation procedure for PMC where the standard resampling technique can be viewed as a special case. This novel stochastic optimization framework of PMC allows for the design of a wider class of adaptation procedures, where parameters beyond just the mean can be adapted. We also discuss how the algorithm can benefit from using alternative stochastic optimization methods, such as implicit SGD [15].

A. Stochastic Gradient Population Monte Carlo (SG-PMC)

Consider a proposal distribution $q(\mathbf{x}; \boldsymbol{\theta})$, where $\boldsymbol{\theta} \in \Theta$ denotes the proposal parameters. We are interested in solving optimization problems of the general form:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\Theta} C_f(\boldsymbol{\theta}), \tag{6}$$

where $C_f(\boldsymbol{\theta}) \triangleq \mathbb{E}_\pi\left[f(\mathbf{x}, \boldsymbol{\theta})\right]$ and $f : \mathbb{R}^{d_x} \times \Theta \to \mathbb{R}$ is a function of both the unknown $\mathbf{x}$ and the proposal parameters $\boldsymbol{\theta}$. Under the assumption that f is differentiable w.r.t. $\boldsymbol{\theta}$, the gradient of $C_f(\boldsymbol{\theta})$ is simply

$$\nabla_{\boldsymbol{\theta}} C_f(\boldsymbol{\theta}) = \mathbb{E}_\pi\left[\nabla_{\boldsymbol{\theta}} f(\mathbf{x}, \boldsymbol{\theta})\right], \tag{7}$$

since the expectation is taken w.r.t. a distribution that does not depend on $\boldsymbol{\theta}$. For the majority of scenarios, (7) is an intractable integral and cannot be obtained in closed form. One can use MC methods to obtain a noisy approximation of (7) and then employ stochastic optimization schemes to solve the minimization problem. This noisy approximation to the gradient, which we denote by $g_f(\boldsymbol{\theta})$, is called a stochastic gradient and can be computed as

$$g_f(\boldsymbol{\theta}) = \frac{1}{K}\sum_{k=1}^{K} \nabla_{\boldsymbol{\theta}} f(\mathbf{x}, \boldsymbol{\theta})\big|_{\mathbf{x}=\mathbf{z}^{(k)}}, \tag{8}$$

where $\{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(K)}\}$ is a mini-batch of K samples drawn from $\pi(\mathbf{x})$. Given these stochastic gradients, we can use SGD [16] to solve (6). The update rule at the ith iteration of the projected SGD algorithm is given by

$$\boldsymbol{\theta}_{i+1} = \Pi_\Theta\left(\boldsymbol{\theta}_i - \eta_i g_f(\boldsymbol{\theta}_i)\right), \tag{9}$$

where $\Pi_\Theta(\cdot)$ is the projection onto the set Θ that guarantees the feasibility of $\boldsymbol{\theta}_{i+1}$ and η_i is the learning rate at the ith iteration. If f is convex in $\boldsymbol{\theta}$ and the sequence of learning rates $\eta_1, \eta_2, \dots$ satisfies the Robbins-Monro stochastic approximation conditions [16], i.e., $\sum_{i=1}^{\infty}\eta_i = \infty$ and $\sum_{i=1}^{\infty}\eta_i^2 < \infty$, then we have the following convergence guarantee [17]:

$$\mathbb{E}_\pi\left[C_f(\boldsymbol{\theta}_{i+1}) - C_f(\boldsymbol{\theta}^*)\right] \to 0 \quad \text{as} \quad i \to \infty. \tag{10}$$

Unfortunately, we cannot easily draw particles from the target $\pi(\mathbf{x})$ to compute stochastic gradients. One possible solution

is to obtain the stochastic gradients using self-normalized IS. Here we instead propose to use resampled particles to estimate the gradients. In the ith iteration of PMC, a set of resampled particles $\{\tilde{\mathbf{x}}_i^{(m)}\}_{m=1}^M$ are used to adapt the mean parameter of each proposal. These particles are sampled from a distribution $\hat{\pi}_i(\mathbf{x})$ which can be written as

$$\hat{\pi}_i(\mathbf{x}) = \sum_{m=1}^{M} w_i^{(m)} \delta(\mathbf{x} - \mathbf{x}_i^{(m)}), \tag{11}$$

where $\delta(\mathbf{x} - \mathbf{x}_i^{(m)})$ denotes a Dirac delta function centered at $\mathbf{x}_i^{(m)}$. Suppose now that $\tilde{\mathbf{z}}^{(1)}, \ldots, \tilde{\mathbf{z}}^{(K)}$ are drawn from an approximation of $\pi(\mathbf{x})$ as they are in PMC sampling. We can replace (8) with an alternative gradient estimator that is constructed using these particles

$$\tilde{g}_f(\boldsymbol{\theta}) = \frac{1}{K} \sum_{k=1}^{K} \nabla_{\boldsymbol{\theta}} f(\mathbf{x}, \boldsymbol{\theta})\big|_{\mathbf{x}=\tilde{\mathbf{z}}^{(k)}} \approx g_f(\boldsymbol{\theta}). \tag{12}$$

The justification for using resampled particles is motivated by [18]. The idea is that the cumulative distribution function (CDF) of the resampled particles, denoted $\hat{F}_i(\mathbf{x})$, asymptotically approaches the CDF of the target distribution $F(\mathbf{x})$ as $M \to \infty$. For instance, when $d_x = 1$, it is easy to see that:

$$\begin{aligned}\hat{F}_i(a) &= \mathbb{E}_{\hat{\pi}_i}\left[\mathbb{1}(\mathbf{x} \leq a)\right] \\ &= \frac{\frac{1}{M}\sum_{m=1}^{M} \tilde{w}_i^{(m)} \mathbb{1}(\mathbf{x}_i^{(m)} \leq a)}{\frac{1}{M}\sum_{m=1}^{M} \tilde{w}_i^{(m)}} \underset{M\to\infty}{\longrightarrow} F(a).\end{aligned}$$

This implies that for large sample sizes M, drawing particles from $\hat{\pi}_i(\mathbf{x})$ is equivalent to drawing particles from $\pi(\mathbf{x})$ and the stochastic gradient approximation in (12) is justified.

Applying this framework to PMC, the idea now becomes to use these alternative gradient estimators to adapt the proposal parameters in order to minimize the function $C_f(\boldsymbol{\theta})$. Therefore, we propose to replace the resampling step at the ith iteration of PMC with a single step of the projected SGD algorithm:

$$\boldsymbol{\theta}_{i+1}^{(m)} = \Pi_{\Theta}\left(\boldsymbol{\theta}_i^{(m)} - \eta_i^{(m)} \tilde{g}_f(\boldsymbol{\theta}_i^{(m)})\right), \quad m = 1, \ldots, M, \tag{13}$$

where $\eta_i^{(m)}$ denotes the learning rate of the mth proposal's parameters. Given the update in (13), we obtain what we call the SG-PMC algorithm, which adapts the parameters of each proposal distribution by explicitly solving the minimization problem in (6). We remark that each proposal is optimizing the same objective, and in the case that f is convex, the adapted parameters should approach the same global optimum. The exact convergence guarantees of the proposal parameters need further investigation, since the gradients are estimated using resampled particles rather than unbiased MC estimators.

B. Minimum Mean Square Error Criteria

Under the assumption that each proposal is parameterized by a mean and covariance (as they are in Algorithm 1), we can use the SG-PMC formulation to derive novel adaptation procedures for each of these parameters. Here we focus only on the mean parameter, however, we emphasize that the procedure can easily be extended to adapt covariance matrices.

Consider a proposal $q(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ denotes the proposal mean and $\boldsymbol{\Sigma}$ the covariance. Also, consider the following minimization problem:

$$\boldsymbol{\mu}^* = \underset{\boldsymbol{\mu}\in\mathbb{R}^{d_x}}{\arg\min}\, C_{\mathrm{MMSE}}(\boldsymbol{\mu}), \tag{14}$$

where $C_{\mathrm{MMSE}}(\boldsymbol{\mu}) \triangleq \mathbb{E}_\pi\left[\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top(\mathbf{x}-\boldsymbol{\mu})\right]$. The solution $\boldsymbol{\mu}^*$ minimizes the mean-square error (MSE) and is known to be the target mean $\boldsymbol{\mu}_\pi \triangleq \int_{\mathbb{R}^{d_x}} \mathbf{x}\pi(\mathbf{x})d\mathbf{x}$.

Following the SG-PMC strategy, we can solve (14) by differentiating the objective $C_{\mathrm{MMSE}}(\boldsymbol{\mu})$ w.r.t. $\boldsymbol{\mu}$ and obtaining stochastic gradients using resampled particles. The gradient of $C_{\mathrm{MMSE}}(\boldsymbol{\mu})$ is simply

$$\nabla_{\boldsymbol{\mu}} C_{\mathrm{MMSE}}(\boldsymbol{\mu}) = \mathbb{E}_\pi\left[(\boldsymbol{\mu} - \mathbf{x})\right]. \tag{15}$$

At each iteration of PMC, we can approximate the gradient in (15) using a single resampled particle. In this case, the corresponding SG-PMC update rule for the mth proposal at the ith iteration is

$$\begin{aligned}\boldsymbol{\mu}_{i+1}^{(m)} &= \boldsymbol{\mu}_i^{(m)} - \eta_i^{(m)}(\boldsymbol{\mu}_i^{(m)} - \tilde{\mathbf{x}}_i^{(m)}) \\ &= (1-\eta_i^{(m)})\boldsymbol{\mu}_i^{(m)} + \eta_i^{(m)}\tilde{\mathbf{x}}_i^{(m)}.\end{aligned} \tag{16}$$

In the case of $\eta_i^{(m)} = 1$ for $i = 1, \ldots, I$ and for $m = 1, \ldots, M$, the update rule corresponds to standard PMC (Algorithm 1). We remark that the update rule in (16) does not include a projection, since $\boldsymbol{\mu}_{i+1}^{(m)}$ is guaranteed to be in $\mathbb{R}^{d_x}$.

C. Minimum Kullback-Leibler Divergence Criteria

Another objective function we can consider minimizing is the Kullback-Leibler divergence (KLD) between the target and the proposal. Then, the SG-PMC scheme aims at solving the following optimization problem for each of the proposals:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}\in\Theta}{\arg\min}\, C_{\mathrm{KLD}}(\boldsymbol{\theta}), \tag{17}$$

where $C_{\mathrm{KLD}}(\boldsymbol{\theta}) \triangleq \mathbb{E}_\pi\left[\log \pi(\mathbf{x})\right] - \mathbb{E}_\pi\left[\log q(\mathbf{x};\boldsymbol{\theta})\right]$. We can express the gradient of $C_{\mathrm{KLD}}(\boldsymbol{\theta})$ as follows:

$$\nabla_{\boldsymbol{\theta}} C_{\mathrm{KLD}}(\boldsymbol{\theta}) = -\mathbb{E}_\pi\left[\nabla_{\boldsymbol{\theta}} \log q(\mathbf{x};\boldsymbol{\theta})\right]. \tag{18}$$

In the case that $q(\mathbf{x};\boldsymbol{\theta})$ is chosen to be a Gaussian distribution, the gradient of $\log q(\mathbf{x};\boldsymbol{\theta})$ w.r.t. the mean $\boldsymbol{\mu}$ is determined by

$$\nabla_{\boldsymbol{\mu}} \log q(\mathbf{x};\boldsymbol{\theta}) = \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}). \tag{19}$$

This leads to the following update rule for an SG-PMC algorithm that minimizes KLD:

$$\boldsymbol{\mu}_{i+1}^{(m)} = (\mathbb{I}_{d_x} - \eta_i^{(m)}\boldsymbol{\Lambda}^{(m)})\boldsymbol{\mu}_i^{(m)} + \eta_i \boldsymbol{\Lambda}^{(m)}\tilde{\mathbf{x}}_i^{(m)}, \tag{20}$$

where $\boldsymbol{\Lambda}^{(m)} = \left(\boldsymbol{\Sigma}^{(m)}\right)^{-1}$ is the precision matrix of the proposal. Note that we drop the iteration index for $\boldsymbol{\Sigma}^{(m)}$ since it is not adapted with this rule. Also, remark that when $\boldsymbol{\Sigma}^{(m)} = \mathbb{I}_{d_x}$, the rule (20) is the same as the rule in (16).

D. Alternative Optimizers in SG-PMC

The problem with using projected SGD is that the stability of the method heavily depends on the learning rates [19]. One way to overcome this challenge is to use implicit updates, which can be obtained by solving the following equation:

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \eta_i \tilde{g}_f(\boldsymbol{\theta}_{i+1}). \tag{21}$$

The idea is to analytically solve for $\boldsymbol{\theta}_{i+1}$ and establish that solution as the update rule. For simple cost functions like $C_{\mathrm{MMSE}}(\boldsymbol{\mu}^{(m)})$, this can easily be done:

$$\begin{aligned}\boldsymbol{\mu}_{i+1}^{(m)} &= \boldsymbol{\mu}_{i}^{(m)} - \eta_i^{(m)}\left(\boldsymbol{\mu}_{i+1}^{(m)} - \tilde{\mathbf{x}}_i^{(m)}\right) \\ &= \boldsymbol{\mu}_{i}^{(m)} - \eta_i^{(m)}\boldsymbol{\mu}_{i+1}^{(m)} + \eta_i^{(m)}\tilde{\mathbf{x}}_i^{(m)}.\end{aligned}$$

We can solve the above equation for $\boldsymbol{\mu}_{i+1}^{(m)}$ and derive a stable adaptation procedure for SG-PMC algorithm given by

$$\boldsymbol{\mu}_{i+1}^{(m)} = (1+\eta_i^{(m)})^{-1}(\boldsymbol{\mu}_{i}^{(m)} + \eta_i^{(m)}\tilde{\mathbf{x}}_i^{(m)}). \tag{22}$$

Regardless the value of $\eta_i^{(m)}$, this update rule will be numerically stable. We remark that using an implicit update for this cost function achieves the same effect as using the update rule in (16) with constrained learning rates $\eta_i^{(m)} \in [0,1)$ for all m.

Using the same procedure, we can also obtain an implicit update rule for the choice of the cost function $C_{\mathrm{KLD}}(\boldsymbol{\mu}^{(m)})$:

$$\boldsymbol{\mu}_{i+1}^{(m)} = \left(\mathbb{I}_{d_x} + \eta_i^{(m)}\boldsymbol{\Lambda}^{(m)}\right)^{-1}\left(\boldsymbol{\mu}_{i}^{(m)} + \eta_i^{(m)}\boldsymbol{\Lambda}^{(m)}\tilde{\mathbf{x}}_i^{(m)}\right). \tag{23}$$

Note that in SG-PMC it is possible use alternative optimizers that incorporate adaptive learning rates, such as RMSprop or ADAM [20], in order to resolve the parameter updates. These adaptive algorithms are more robust when the stochastic gradients have high variance.

V. Example: Target Localization

Consider the problem of localizing a target in a wireless sensor network (WSN) [9]. Let $\mathbf{p}_0 \in \mathbb{R}^2$ be the target position and let there be V sensors with positions $\mathbf{h}_v \in \mathbb{R}^2$ for $v = 1,\ldots,V$. The nth observation of the vth sensor is modeled as:

$$y_{n,v} = 20\log\left(\|\mathbf{p}_0 - \mathbf{h}_v\|_2\right) + \epsilon_{n,v}, \tag{24}$$

where $\epsilon_{n,v} \sim \mathcal{N}(0,\alpha_v^2)$. The unknowns of this model are the position of the target $\mathbf{p}_0$ and the noise standard deviations of the V sensors $\boldsymbol{\alpha} = [\alpha_1,\ldots,\alpha_V]^\top$. For consistent notation, we define $\mathbf{x} \triangleq [\mathbf{p}_0^\top, \boldsymbol{\alpha}^\top]^\top \in \mathbb{R}^{V+2}$. Our goal is to approximate the posterior of $\mathbf{x}$ given the observations from all sensors. We assume a uniform prior for both the target position and the sensor standard deviations, i.e., $\mathbf{p}_0 \sim \mathcal{U}([-30,30]^2)$ and $\alpha_v \sim \mathcal{U}(0.01,20)$ for all $v = 1,\ldots,V$.

We consider that there are $V = 6$ sensors ($d_x = 8$) and generate $N = 20$ observations for each sensor according to the following settings: $\mathbf{p}_0 = [2.5,2.5]^\top$, $\alpha = [1,2,1,0.5,3,0.2]^\top$, $\mathbf{h}_1 = [3,-8]^\top$, $\mathbf{h}_2 = [8,10]^\top$, $\mathbf{h}_3 = [-4,-6]^\top$, $\mathbf{h}_4 = [-8,1]^\top$, $\mathbf{h}_5 = [10,0]^\top$, and $\mathbf{h}_6 = [0,10]^\top$. We run SG-PMC using $M = 50$ Gaussian proposals that are adapted using (16) over $I = 200$ iterations, where we sampled a logarithm transformation of the standard deviations $\boldsymbol{\alpha}$ to guarantee that the considered samples for the standard deviations were positive. We also consider proposals adapted using the RMSprop algorithm, although other stochastic optimization algorithms which utilize adaptive learning rates could have been considered. We initialized the learning rates as $\eta_1^{(m)} = \eta_1$, the proposal means as $\boldsymbol{\mu}_1^{(m)} \sim \mathcal{U}([1,4]^{d_x})$ and the covariance matrices as $\boldsymbol{\Sigma}_1^{(m)} = \mathbb{I}_8$ for all $m = 1,\ldots,M$. We evaluate the performance of a particular run of SG-PMC at the ith iteration by calculating the MSE of the estimate of $\mathbf{x}$. The results are averaged of 2000 MC simulations and summarized in Figs. 1 and 2.

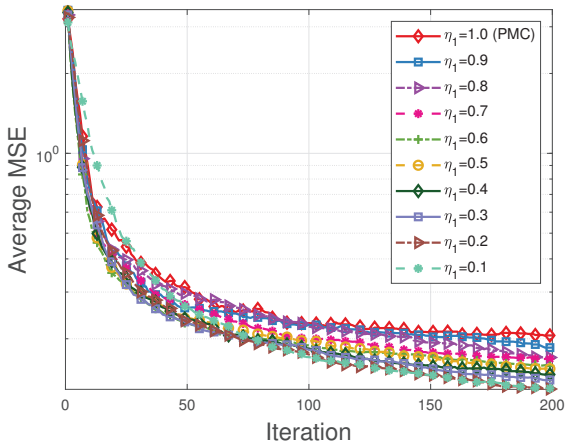

Fig. 1: Average MSE of parameter estimates for SG-PMC using the update rule in (16) with a constant learning rate.

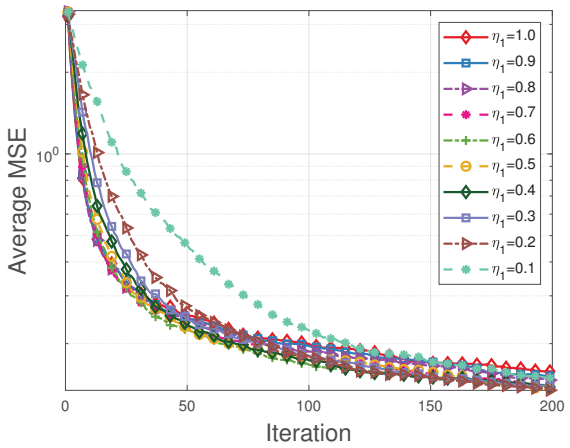

Fig. 2: Average MSE of parameter estimates for SG-PMC using the RMSprop optimizer.

Simulation results indicate that for different learning rates, the proposed SG-PMC algorithm outperforms the standard PMC sampler. The standard PMC sampler corresponds to the blue curve in Fig. 1. We can see that when smaller learning rates are used, the SG-PMC algorithm is able to achieve better performance at a faster rate. Furthermore, Fig. 2 indicates that by using adaptive optimizers like RMSprop, we can obtain good performance across a wider range of initial learning rates.

VI. Concluding Remarks

In this work, we developed a novel adaptation scheme for population Monte Carlo (PMC) based on stochastic optimization. Using resampled particles to approximate stochastic gradients, a general family of PMC algorithms based on minimizing favorable objective functions is derived. The proposed approach encompasses a family of PMC algorithms, where standard PMC is a special case. Simulation results on a localization problem demonstrate the advantages of the proposed stochastic optimization framework over the traditional resampling-based approach.

REFERENCES

[1] C. Robert and G. Casella, *Monte Carlo statistical methods*, Springer Science & Business Media, 2013.

[2] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian data analysis*, Chapman and Hall/CRC, 2013.

[3] J. V. Candy, *Bayesian signal processing: classical, modern, and particle filtering methods*, vol. 54, John Wiley & Sons, 2016.

[4] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The journal of chemical physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[5] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," 1970.

[6] F. Liang, C. Liu, and R. Carroll, *Advanced Markov chain Monte Carlo methods: learning from past samples*, vol. 714, John Wiley & Sons, 2011.

[7] K. L. Mengersen, R. L. Tweedie, et al., "Rates of convergence of the hastings and metropolis algorithms," *The annals of Statistics*, vol. 24, no. 1, pp. 101–121, 1996.

[8] A. Owen and Y. Zhou, "Safe and effective importance sampling," *Journal of the American Statistical Association*, vol. 95, no. 449, pp. 135–143, 2000.

[9] M. F. Bugallo, V. Elvira, L. Martino, D. Luengo, J. Miguez, and P. M. Djuric, "Adaptive importance sampling: the past, the present, and the future," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 60–79, 2017.

[10] O. Cappé, A. Guillin, J. Marin, and C. P. Robert, "Population monte carlo," *Journal of Computational and Graphical Statistics*, vol. 13, no. 4, pp. 907–929, 2004.

[11] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo, "Improving population monte carlo: Alternative weighting and resampling schemes," *Signal Processing*, vol. 131, pp. 77–91, 2017.

[12] O. Cappé, R. Douc, A. Guillin, J. Marin, and C. P. Robert, "Adaptive importance sampling in general mixture classes," *Statistics and Computing*, vol. 18, no. 4, pp. 447–459, 2008.

[13] E. Koblents and J. Míguez, "A population monte carlo scheme with transformed weights and its application to stochastic kinetic models," *Statistics and Computing*, vol. 25, no. 2, pp. 407–425, 2015.

[14] C. Andrieu and C. P. Robert, *Controlled MCMC for optimal sampling*, INSEE, 2001.

[15] P. Toulis and E. M. Airoldi, "Implicit stochastic gradient descent for principled estimation with large datasets," *ArXiv e-prints*, 2014.

[16] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[17] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM Journal on optimization*, vol. 19, no. 4, pp. 1574–1609, 2009.

[18] A. F. M. Smith and A. E. Gelfand, "Bayesian statistics without tears: a sampling–resampling perspective," *The American Statistician*, vol. 46, no. 2, pp. 84–88, 1992.

[19] P. Toulis, D. Tran, and E. Airoldi, "Towards stability and optimality in stochastic gradient descent," in *Artificial Intelligence and Statistics*, 2016, pp. 1290–1298.

[20] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.