SIAM J. SCI. COMPUT.
Vol. 40, No. 3, pp. A1301–A1321

AN ADAPTIVE INDEPENDENCE SAMPLER MCMC ALGORITHM FOR BAYESIAN INFERENCES OF FUNCTIONS*

ZHE FENG† AND JINGLAI LI‡

Abstract. Many scientific and engineering problems require one to perform Bayesian inferences in function spaces, in which the unknowns are of infinite dimension. In such problems, many standard Markov chain Monte Carlo (MCMC) algorithms become arbitrarily slow under the mesh refinement, which is referred to as being dimension dependent. In this work we develop an independence sampler based MCMC method for the Bayesian inferences of functions. We represent the proposal distribution as a mixture of a finite number of specially parametrized Gaussian measures. We also design an efficient adaptive algorithm to adjust the parameter values of the mixtures from the previous samples. Finally we provide numerical examples to demonstrate the efficiency and robustness of the proposed method, even for problems with multimodal posterior distributions.

Key words. adaptive Markov chain Monte Carlo, Bayesian inference, Gaussian mixture, independence sampler, inverse problem

AMS subject classifications. 65C05, 62F15

DOI. 10.1137/15M1021751

1. Introduction.

Nonparametric Bayesian inferences have applications in many scientific problems, ranging from regression [15] to inverse problems [17, 34]. In those problems the unknowns that we want to infer are often functions of space or time. In many practical problems, the posterior distributions do not admit a closed form and need to be computed numerically. Specifically one first represents the unknown function with a finite dimensional parametrization, for example, by discretizing the function on a predetermined mesh grid, and then solves the resulting finite dimensional inference problem with the Markov chain Monte Carlo (MCMC) simulations. It has been known that standard MCMC algorithms, such as the random walk Metropolis–Hastings (RWMH), can become arbitrarily slow as the discretization mesh of the unknown is refined [31, 33, 6, 26]. That is, the mixing time of an algorithm can increase to infinity as the dimension of the discretized parameter approaches to infinity, and in this case the algorithm is said to be *dimension-dependent*. To this end, a very interesting line of research is to develop MCMC algorithms whose acceptance probabilities are independent of discretization dimensionality. One way to develop such algorithms is to formulate them directly in the function spaces. For example, a family of function-space MCMC algorithms were presented in [8] by constructing a preconditioned Crank–Nicolson (pCN) discretization of a stochastic partial differential equation that preserves the reference measure.

*Submitted to the journal's Methods and Algorithms for Scientific Computing section May 18, 2015; accepted for publication (in revised form) November 9, 2017; published electronically May 8, 2018.

http://www.siam.org/journals/sisc/40-3/M102175.html

Funding: This work was supported by the NSFC, under grant 1771289.

†Department of Mathematics, Zhiyuan College, Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai 200240, China (sjtufz@sjtu.edu.cn).

‡Corresponding author. Institute of Natural Sciences, Department of Mathematics, and the MOE Key Laboratory of Scientific and Engineering Computing, Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai 200240, China. Present address: Department of Mathematical Sciences, University of Liverpool, Liverpool, L69 7ZL, UK (jinglaili@sjtu.edu.cn, lijinglai@gmail.com).

Just like its finite dimensional counterparts, the sampling efficiency of the function-space MCMC can be improved by incorporating the data information in the proposal design. One way of doing so is to guide the proposal with the local derivative information of the likelihood function. Methods in this category include the stochastic Newton MCMC [25, 28], the operator-weighted proposal method [20], the infinite dimensional Metropolis-adjusted Langevin algorithm (MALA) [7, 5], and the dimension-independent likelihood-informed MCMC [9], just to name a few. An alternative type of method to improve the efficiency with the data information is the adaptive MCMC (cf. [1, 2, 32] and the references therein), which automatically adjusts the proposal as the algorithm proceeds. While the first type of approach utilizes the gradient or the Hessian of the likelihood function to accelerate the computation, the adaptive methods do not require such information, which makes the particularly convenient for problems with black-box models.

In this paper we propose an adaptive MCMC algorithm with independence sampler (IS) [35] for such function space inference problems. IS, also known as the independent Metropolis–Hastings (MH) [16], or the Metropolized independent sampling [23], is an alternative to the popular RWMH algorithm, which proposes from a stationary distribution, i.e., one that is independent of the present position. The design principle for the independence sampler method is rather straightforward: loosely speaking, one should choose the proposal distribution to be as close to the target distribution as possible. The basic idea here is to represent the proposal distribution with a mixture of a finite number of parametrized Gaussian measures and optimize the parameters as the algorithm proceeds. Our specific parametrization ensures the algorithm is well-defined in function spaces. As is mentioned earlier, a major advantage of the proposed method is that it can propose efficiently without using the derivative information of the likelihood function. Moreover as is demonstrated by our numerical examples in section 5, our method performs well for *multimodal* posterior distributions which can be challenging for many existing algorithms.

The rest of the paper is organized as the following. In section 2 we introduce the basic setup of the function space Bayesian inference problem. In section 3 we present the Gaussian mixture based independence sampler for Bayesian inference in function spaces and show that the acceptance probability associated to the proposal is independent of discretization dimensionality. The proposal distribution that we use is parametrized by a finite number of parameters and in section 4 we describe the adaptive algorithm to adjust the proposal parameters to improve the sampling efficiency. Section 5 provides several numerical examples of the proposed method.

2. Problem setup. We consider a separable Hilbert space X with inner product $\langle\cdot,\cdot\rangle_X$. Our goal is to estimate the unknown $u \in X$ from data $y \in Y$, where Y is the data space and y is related to u via the likelihood function

$$L(u, y) = \frac{1}{Z}\exp(-\Phi^y(u)),$$

where Z is a normalization constant. In what follows, without causing any ambiguity, we shall drop the superscript y in Φ^y for simplicity. In this work we require that the functional Φ satisfies [8, Assumptions (6.1)], i.e.,

(a) there exists $q > 0$, $Q > 0$ such that, for all $u \in X$,

$$0 \le \Phi(u) \le Q(1 + \|u\|_X^q);$$

(b) for every $r > 0$ there is $Q_r > 0$ such that, for all $u, v \in X$ with $\max\{\|u\|_X, \|v\|_X\} < r$,

$$|\Phi(u) - \Phi(v)| \le Q_r \|u - v\|_X.$$

We do not have any restrictions on the space Y.

In the Bayesian inference we assume that the prior μ_0 of u is a (without loss of generality) zero-mean Gaussian measure defined on X with covariance operator C_0, i.e., $\mu_0 = N(0, C_0)$. Note that C_0 is symmetric positive and of trace class. The range of $C_0^{\frac{1}{2}}$,

$$E = \left\{ u = C_0^{\frac{1}{2}} x \,|\, x \in X \right\} \subset X,$$

which is a Hilbert space equipped with inner product [10],

$$\langle \cdot, \cdot \rangle_E = \left\langle C_0^{-\frac{1}{2}} \cdot, C_0^{-\frac{1}{2}} \cdot \right\rangle_X,$$

is called the Cameron–Martin space of measure μ_0. In this setting, the posterior measure μ^y of u conditional on data y is provided by the Radon–Nikodym derivative,

$$\frac{d\mu^y}{d\mu_0}(u) = \frac{1}{Z} \exp(-\Phi(u)), \tag{2.1}$$

which can be interpreted as the Bayes' rule in the infinite dimensional setting. Our goal is to draw samples from the posterior μ^y with MCMC algorithms.

Note that the definition of the maximum a posteriori (MAP) estimator in finite dimensional spaces does not apply here, as the measures μ^y and μ_0 are not absolutely continuously with respect to the Lebesgue measure; instead, the MAP estimator in X is defined as the minimizer of the Onsager–Machlup functional (OMF) [11, 21],

$$I(u) := \Phi(u) + \frac{1}{2}\|u\|_E^2, \tag{2.2}$$

over the Cameron–Martin space E. In section 5, we shall use OMF as an indicating quantity to compare the performance of various MCMC algorithms. Finally we quote the following lemma [10, Chapter 1], which will be useful in next section.

LEMMA 2.1. *There exists a complete orthonormal basis $\{e_k\}_{k\in\mathbb{N}}$ on X and a sequence of nonnegative numbers $\{\alpha_k\}_{k\in\mathbb{N}}$ such that $C_0 e_k = \alpha_k e_k$ and $\sum_{k=1}^{\infty} \alpha_k < \infty$, i.e., $\{e_k\}_{k\in\mathbb{N}}$ and $\{\alpha_k\}_{k\in\mathbb{N}}$ being the eigenfunctions and eigenvalues of C_0, respectively.*

Without loss of generality, we assume that the eigenvalues $\{\alpha_k\}_{k=1}^{\infty}$ are in a descending order.

3. Gaussian mixture based independence sampler. In this section, we present our Gaussian mixture based independence sampler and show that it is well-defined in the function space.

3.1. Independence sampler MCMC. We start by briefly reviewing the independence sampler MCMC algorithm. Given a proposal distribution μ, we define measures

$$\nu(du, du') = \mu(du')\mu^y(du),$$
$$\nu^\dagger(du, du') = \mu(du)\mu^y(du')$$

on the product space $X \times X$. When $\nu^\dagger$ is absolute continuous with respect to ν, we can define the acceptance probability [36]

$$A(u, u') = \min\left\{1, \frac{d\nu^\dagger}{d\nu}(u, u')\right\}, \tag{3.1}$$

where

$$\frac{d\nu^\dagger}{d\nu}(u, u') = \frac{d\mu^y}{d\mu}(u')\frac{d\mu}{d\mu^y}(u). \tag{3.2}$$

The IS MCMC in a function space proceeds as follows in each iteration:

1. Draw a sample u_{proposed} from the proposal μ.
2. Let $u_{\text{next}} = u_{\text{proposed}}$ with probability $A(u_{\text{current}}, u_{\text{proposed}})$ and $u_{\text{next}} = u_{\text{current}}$ with probability $1 - A(u_{\text{current}}, u_{\text{proposed}})$.

It is obvious that the acceptance probability (3.1) of the algorithm is well-defined if and only if $\nu^\dagger$ is absolutely continuous with respect to ν, which requires that μ and μ^y are equivalent to each other. Since μ^y and μ_0 are equivalent, it suffices to require μ and μ_0 to be equivalent. Interestingly, the pCN scheme with a specific choice of parameter values yields a dimension-independent IS whose proposal distribution is simply the prior. Despite its dimension-independence property, simply proposing according to the prior is inefficient when the data is highly informative, i.e., the posterior being far apart from the prior. Next we shall introduce a more efficient proposal measure than the prior that is to be used in IS MCMC algorithms.

3.2. Gaussian mixture proposals. In finite dimensional Bayesian inference problems, Gaussian mixture (GM) distributions [27] are often used as the IS proposal distributions for their flexibility and convenience to draw samples from. We now extend the use of GM to the infinite dimensional setting. Let $\{\mu_j\}_{j=1}^J$ be a set of Gaussian measures on X with $\mu_j = \mathcal{N}(m_j, C_j)$ for $j = 1, \ldots, J$, and we define the Gaussian mixture proposal as

$$\mu(dx) = \sum_{j=1}^J w_j \mu_j(dx), \tag{3.3}$$

where $\{w_j\}_{j=1}^J$ are the mixing weights with $\sum_{j=1}^J w_j = 1$. It is clear that μ is equivalent to μ_0 as long as each μ_j is equivalent to μ_0, and moreover the Radon–Nikodym derivative of μ to μ_0 is

$$\frac{d\mu}{d\mu_0}(u) = \sum_{j=1}^J w_j \frac{d\mu_j}{d\mu_0}(u). \tag{3.4}$$

Next we discuss our parametrization of each μ_i. First recall that, according to Lemma 2.1, $\{e_k\}_{k\in\mathbb{N}}$ form a complete basis set of X. Our parametrization of μ_i is in the form of

$$m_j = \sum_{k=1}^\infty x_{j,k}\alpha_k e_k, \tag{3.5a}$$

$$C_j^{-1} = C_0^{-1} + H_j, \tag{3.5b}$$

where each H_j is defined as

$$H_j \cdot = \sum_{k=1}^{\infty} h_{j,k} \langle e_k, \cdot \rangle e_k \tag{3.5c}$$

and $x_{j,k}$ and $h_{j,k}$ are coefficients. The following theorem provides a sufficient condition for $\mu_j = \mathcal{N}(m_j, C_j)$ to be a well defined Gaussian measure on X and equivalent to μ_0.

THEOREM 3.1. *If $x_j, h_j \in l_2$, and $h_{j,k} > -\frac{1}{\alpha_k}$ for all $k \in \mathbb{N}$, $\mu_j = \mathcal{N}(m_j, C_j)$ is a Gaussian measure on X that is equivalent to μ_0.*

Proof. We let $\{\beta_{j,k}\}_{k\in\mathbb{N}}$ be the eigenvalues of C_j, i.e, $C_j e_k = \beta_{j,k} e_k$ for all $k \in \mathbb{N}$. And it is easy to see that

$$\beta_{j,k} = \left(\alpha_k^{-1} + h_{j,k}\right)^{-1} = \frac{\alpha_k}{1 + \alpha_k h_{j,k}}. \tag{3.6}$$

As $x_j, h_j \in l_2$, $\frac{1}{1+\alpha_k h_{j,k}}$ is bounded and thus $\sum_{k=1}^{\infty} \beta_{j,k} < \infty$. It follows that $C_j \in L_1^+(X)$ and $\mu_j = \mathcal{N}(m_j, C_j)$ defines a Gaussian measure on X.

We now show that μ_j is equivalent to μ_0. First we introduce $\mu_j' = \mathcal{N}(0, C_j)$. Using (3.6) and $h_{j,k} > -\frac{1}{\alpha_k}$ for all $k \in \mathbb{N}$, we can get

$$\sum_{k=1}^{\infty} \frac{(\beta_{j,k} - \alpha_k)^2}{(\beta_{j,k} + \alpha_k)^2} = \sum_{k=1}^{\infty} \frac{\alpha_k^2 h_{j,k}^2}{(2 + \alpha_k h_{j,k})^2} \leq \sum_{k=1}^{\infty} \alpha_k^2 h_{j,k}^2 < \infty,$$

as $\lim_{k\to\infty} \alpha_k = 0$ and $h_j \in l_2$. By the Feldman–Hajek theorem [10], we have that μ_j' is equivalent to μ_0. Now recall that $m_j \in E = C_0^{\frac{1}{2}}(X) = C_j^{\frac{1}{2}}(X)$, and so we have $\mu_j' = \mathcal{N}(0, C_j)$ and $\mu_j = \mathcal{N}(m_j, C_j)$ are equivalent, which completes the proof. ∎

Let us assume for now that the conditions in Theorem 3.1 are satisfied and we shall verify this assumption later. It is easy to show that

$$\begin{aligned} \frac{d\mu_j}{d\mu_0}(u) &= \frac{|C_0|^{1/2}}{|C_j|^{1/2}} \exp\left(-\frac{1}{2} \left\| C_j^{-1/2} m_j \right\|_X^2 + \langle u, C_j^{-1} m_j \rangle_X - \frac{1}{2} \langle u, H_j u \rangle_X \right) \\ &= \prod_{k=1}^{\infty} \sqrt{\frac{\alpha_k}{\beta_{j,k}}} \exp\left[-\frac{1}{2} \sum_{k=1}^{\infty} \left(\frac{\alpha_k^2}{\beta_{j,k}} x_{j,k}^2 + h_{j,k} u_k^2 - \frac{2\alpha_k}{\beta_{j,k}} x_{j,k} u_k \right) \right], \end{aligned} \tag{3.7}$$

where $u_k = \langle u, e_k \rangle$ is the projection of u onto e_k. Note that the density $d\mu_j/d\mu_0$ actually depends on m_j and h_j, and thus for convenience's sake, we define a function $f(\cdot, \cdot, \cdot)$ such that

$$f(u, x_j, h_j) = \frac{d\mu_j}{d\mu_0}(u),$$

and we then can derive from (3.4) that

$$\frac{d\mu^y}{d\mu}(u) = \frac{1}{Z} \exp(-\Phi(u)) / \left(\sum_{j=1}^{J} w_j f(u, x_j, h_j) \right),$$

and the density $d\mu/d\mu_y$ can be computed accordingly.

3.3. Minimizing the Kullback–Leibler divergence. Now recall that for the algorithm to be efficient we need the proposal μ to be close to μ^y and a natural choice is to determine μ by minimizing the Kullback–Leibler divergence (KLD) between μ^y and μ:

$$D_{KL}(\mu^y||\mu) = \int \log \frac{d\mu^y}{d\mu}(u)\mu^y(du), \tag{3.8}$$

where μ is parametrized with (3.5). Note that x_j and h_j are set to be of infinite dimensions in the formulation above. In numerical simulations, however, x_j and h_j must be truncated at some finite number K. Such a truncation is also reasonable from a practical point of view. In fact, one often can realistically assume that the data is only informative on a finite number of directions [9, 8] in X, and under this assumption, we only need to keep a finite number of components of each x_j and h_j. We emphasize that K, which represents the number of dimensions that are informed by the data (i.e., the so-called intrinsic dimensionality), should not be confused with the discretization dimensionality of the problem, i.e., the number of mesh points used to represent the unknown. Determining the value of K is an important task for our algorithm and here we choose K with a heuristic approach:

$$K = \min\left\{ k \in \mathbb{N} \;\middle|\; \frac{\alpha_k}{\alpha_1} < \epsilon \right\},$$

where ϵ is a prescribed threshold. In what follows, we shall adopt this finite, K-dimensional formulation, and thus we have the following optimization problem:

$$\min_{\{x_j,\, h_j \in \mathbb{R}^K,\, w_j \in [0,1]\}_{j=1}^J} D_{KL}(\mu^y||\mu), \tag{3.9}$$

subject to $\sum_{j=1}^J w_j = 1$. By some elementary calculations, we reduce formula (3.9) to

$$\min_{\{x_j,\, h_j \in \mathbb{R}^K,\, w_j \in [0,1]\}_{j=1}^J} -\int \log \left[\sum_{j=1}^J w_j f(u, x_j, h_j) \right] \mu^y(du), \tag{3.10}$$

subject to $\sum_{j=1}^J w_j = 1$. We now show that the proposal μ constructed this way is well-defined in function space, and to this end we have the following corollary.

COROLLARY 3.2. *If* $\{x_j, h_j, w_j\}_{j=1}^J$ *is a solution of* (3.10)*, the resulting* μ *is equivalent to* μ_0*.*

Proof. It is obvious that if $\{x_j, h_j, w_j\}_{j=1}^J$ is a solution of (3.10), $x_j, h_j \in l_2$. Taking the partial derivative of the objective function in (3.10) with respect to $h_{j,k}$ and setting it to be zero yields the following equation:

$$\int \frac{w_j f(u, x_j, h_j)}{\sum_{l=1}^J w_l f(u, x_l, h_l)} d\mu^y \frac{\alpha_k}{1 + \alpha_k h_{j,k}} = \int \frac{w_j f(u, x_j, h_j)(\alpha_k x_{j,k} - u_k)^2}{\sum_{l=1}^J w_l f(u, x_l, h_l)} \Big) d\mu^y.$$

As the following two integrals are obviously positive,

$$\int \frac{w_j f(u, x_j, h_j)}{\sum_{l=1}^J w_l f(u, x_l, h_l)} d\mu^y > 0, \quad \text{and} \quad \int \frac{w_j f(u, x_j, h_j)(\alpha_k x_{j,k} - u_k)^2}{\sum_{l=1}^J w_l f(u, x_l, h_l)} \Big) d\mu^y > 0,$$

we have $1 + \alpha_k h_{j,k} > 0$. Thus all the conditions of Theorem 3.1 are satisfied and the corollary follows immediately. ∎

Finally we note that, in the special case where $J = 1$, namely, the proposal being simply a Gaussian distribution, our parametrization is similar to the finite rank representation used in [29, 30]. In fact, the aforementioned works also proposed to approximate the posterior with a Gaussian distribution by minimizing the KLD between the two distributions. The major difference is the KLD (recall that it is asymmetric) formulation: the authors of [29, 30] compute the divergence from the Gaussian approximation to the true posterior, while here we compute the divergence the other way around. An advantage of the present formulation is that the solution to (3.10) can be explicitly obtained:

$$x_k = \frac{1}{\alpha_k} \int u_k d\mu^y, \tag{3.11a}$$

$$h_k = \frac{1}{\int (\alpha_k x_k - u_k)^2 d\mu^y} - \frac{1}{\alpha_k} \tag{3.11b}$$

for $k = 1, \ldots, K$, while in the formulation of [30] the resulting optimization problem has to be solved with a stochastic optimization algorithm. The explicit solutions (3.11) are of essential importance in our adaptive algorithm.

4. The adaptive algorithm. In this section we discuss the algorithm to implement the IS method proposed in section 3, starting with an introduction to the adaptive MCMC.

4.1. Adaptive MCMC. The basic idea of the adaptive MCMC is to repeatedly adjust the proposal parameters using the information in the previous samples. Here we are focused on the adaptive algorithms with IS [16, 13, 19, 12], while noting that other types of adaptive algorithms include the adaptive MH [14], the adaptive MALA [3, 24], and the adaptive Metropolis-within-Gibbs [32]. Specifically our adaptive algorithm has the following three key ingredients. First, to enforce the asymptotic ergodicity, we terminate the adaptation in a finite number of steps. Second, we use a tempered prerun to obtain the initial parameter values for the iteration. Simply speaking the technique of tempering is to construct a sequence of intermediate distributions that converge to the true posterior μ^y and use these intermediate distributions to guide the MCMC samples to the true posterior. This strategy is particularly useful for multimodal posterior distributions. Without loss of generality, we assume that the tempering distributions are augmented by a tempering parameter λ,

$$\frac{d\mu^{y,\lambda}}{d\mu_0} \propto \exp(-\lambda \Phi(u)),$$

and clearly $\mu^{y,\lambda} = \mu^y$ when $\lambda = 1$ and the tempering distribution is "wider" than the true posterior for $0 \le \lambda < 1$. In practice we can choose a finite number of tempering parameters $\{\lambda_i\}_{i=1}^{I_{\text{temp}}}$, where $0 \le \lambda_1 < \lambda_2 < \cdots < \lambda_{I_{\text{temp}}} = 1$. We also note that for problems where the posterior is not too far apart from the prior, tempering may not be necessary. Finally we estimate and update the proposal parameters after every fixed number of iterations. The adaptive scheme is summarized as the following:

- Initialization: the total number of iterations I_{tol}, the number of adapted iterations I_{adp}, the number of prerun (tempering) iterations I_{temp}, a set of tempering parameters $\{\lambda_i\}_{i=1}^{I_{\text{temp}}}$, and the number of samples used in each tempered iteration N_{temp}, and the number of samples in each iteration N_S.

- Prerun (optional): let $\mu'_{(0)} = \mu_0$; for $i = 1 : I_{\text{temp}}$ perform the following:
  1. Run MCMC with proposal $\mu'_{(i-1)}$ to draw a set of N_{temp} samples from μ^{y,λ_i}, denoted by S'_i.
  2. Update the parameter values with samples S'_i obtaining proposal $\mu'_{(i)}$.
- Iteration: let $S = \emptyset$ and $\mu_{(0)} = \mu'_{(I_{\text{temp}})}$; for i= 1 to I_{tol} perform the following:
  1. Run MCMC with proposal $\mu_{(i-1)}$ to draw a set of N_S samples from μ^y, denoted by S_i. Let $S = S \cup S_i$.
  2. If $i < I_{\text{adp}}$, update the parameter values with samples S obtaining proposal $\mu_{(i)}$; otherwise, let $\mu_{(i)} = \mu_{(i-1)}$.

The adaptive algorithm presented above is rather simple; we note, however, that our method is rather flexible and one can pair it with any desired adaptive IS algorithm. A key step in the adaptive algorithm is to estimate the parameters from the samples, which is done by solving the sample average estimator of the optimization problem (3.10):

$$\max_{\{x_j,h_j,w_j\}_{j=1}^J} \sum_{n=1}^N \log\left[\sum_{j=1}^J w_j f(u^n, x_j, h_j)\right], \tag{4.1}$$

subject to $\sum_{j=1}^J w_j = 1$. Next we discuss two methods to solve (4.1).

4.2. Expectation maximization algorithm. The expectation maximization (EM) is one of the most popular methods to determine the parameters in mixture models [27]. Simply put, the EM algorithm iteratively updates the parameter values in a way that the function value is always increased until convergence is achieved. Each iteration consists of an expectation-step and a maximization-step. It should be noted that the EM algorithm is not guaranteed to converge to the optimal solutions in general [37]. The theory and implementation details of the EM algorithm and its application to mixture models can be found in the aforementioned references, and we shall not repeat them here. When applied to our problem, the update formula in each iteration can be explicitly obtained. In the expectation-step, the membership probability q_j^n, namely, the probability that a sample u^n is in the mixture j, is computed,

$$q_j^n = \frac{w_j f(u^n, h_j, m_j)}{\sum_{j=1}^J w_j f(u^n, h_j, m_j)} \tag{4.2}$$

for each $j = 1, \ldots, J$ and $n = 1, \ldots, N$; in the maximization-step, the parameter values are updated using the following equations:

$$w_j = \frac{1}{N}\sum_{i=1}^N q_j^i, \tag{4.3a}$$

$$x_{j,k} = \frac{1}{N\alpha_k w_j}\sum_{n=1}^N q_j^n u_k^n, \tag{4.3b}$$

$$h_{j,k} = \sum_{n=1}^N q_j^n \left(\sum_{n=1}^N q_j^n(\alpha_k x_{j,k} - u_k^n)^2\right)^{-1} - \frac{1}{\alpha_k}, \tag{4.3c}$$

where $u_k^n = \langle u^n, e_k\rangle$. The EM algorithm is arguably the most common method to estimate the parameters of mixtures. However, our numerical tests indicate that in some

practical problems the EM algorithm is not sufficiently reliable especially when the sample set only contains a small number of accepted draws. Moreover, our algorithm frequently updates the proposal parameters, which makes the computationally intensive EM algorithms less attractive from an efficiency perspective. For these reasons, we propose an alternative method to EM, which estimates the mixture parameters using clustering.

4.3. Estimating parameters with clustering. Our estimation method with clustering is largely based on the finite dimensional method developed in [13]. The idea is rather simple: one first partitions the samples into several clusters and then fits each cluster with a Gaussian distribution. A difficulty here is that our MCMC samples are of infinite dimension, which makes clustering challenging. To solve the problem, we first project the samples onto the K eigenfunctions of the covariance operator and then cluster the resulting K dimensional data $\{(u_1^n, \dots, u_K^n)\}_{n=1}^N$ and $u_k^n = \langle u^n, e_k \rangle$. Specifically we use the k-means algorithm to cluster the data, and the number of clusters J is determined with the Bayesian information criteria method [27]. In fact we have found in our numerical tests that the algorithm is rather robust against the number of clusters. We then use the Gaussian distribution parametrized in the form of (3.5) to fit each cluster, and thanks to (3.11), the parameters values can be estimated explicitly as

$$x_{j,k} = \frac{1}{N_j \alpha_k} \sum_{u^n \in \Theta_j} u_k^n, \tag{4.4a}$$

$$h_{j,k} = \frac{1}{\frac{1}{N_j} \sum_{u^n \in \Theta_j} (u_k^n)^2 - m_{j,k}^2} - \frac{1}{\alpha_k}, \tag{4.4b}$$

where Θ_j is the jth cluster of samples, N_j is the sample size of Θ_j for $j = 1, \dots, J$, and $k = 1, \dots, K$. The mixture weights are simply determined by the fraction of samples in each cluster. We note that the clustering based method does not generally yield a solution to (4.1) and thus we regard it as an approximate method to estimate the parameters. We conclude the section with a pseudo code (Algorithm 4.1) of our algorithm, and interested readers can use it as a basis for their own implementation.

5. Numerical examples.

5.1. An ordinary differential equation example. Our first example is a simple inverse problem where the forward model is governed by an ordinary differential equation

$$\frac{dx(t)}{dt} = -u(t)x(t) \tag{5.1}$$

with a prescribed initial condition. We assume that the solution $x(t)$ is observed at several times in the interval $[0, T]$ and we want to infer the unknown coefficient $u(t)$ for $t \in [0, T]$.

In our numerical experiments, we let the initial condition be $\eta(0) = 1$ and $T = 1$. Now suppose that the solution is measured every $T/20$ time unit from 0 to T and the error in each measurement is assumed to be an independent zero-mean Gaussian random variable with variance 0.05^2. In the computation, 100 equally spaced grid points are used to represent the unknown. Moreover, we assume that the state space for u is $X = L_2([0, T])$ and the prior is a zero-mean Gaussian measure in X with an

Algorithm 4.1 The complete algorithm for the adaptive IS with GM. I_{temp} is the number of tempered iterations. $\{\lambda_i\}_{i=1}^{I_{\text{temp}}}$ are the tempering parameters. N_{temp} is the number of samples used in each tempered iteration. N_{tol} is the total number of samples drawn by the algorithm. N_{adp} is the number of samples drawn between two consecutive parameter updates. $N_{\max}$ is the maximum length of chain before the adaptation is terminated.

input : I_{temp}, $\{\lambda_i\}_{i=1}^{I_{\text{temp}}}$, N_{temp}, N_{tol}, $N_{\max}$, N_{adp}.
output: N_{tol} samples drawn from μ^y: $\{u^n\}_{n=1}^{N_{\text{tol}}}$.
$\mu \leftarrow \mu_0$;
for $i \leftarrow 1$ **to** I_{temp} **do**
  draw $u^0 \sim \mu$;
  for $n \leftarrow 1$ **to** N_{temp} **do**
    draw $u' \sim \mu$;
    draw $a \sim U[0,1]$ and compute

$$A \leftarrow \min\left\{1, \frac{d\mu^{y,\lambda_i}}{d\mu}(u')\frac{d\mu}{d\mu^{y,\lambda_i}}\left(u^{n-1}\right)\right\};$$

    if $A > a$ **then** $u^n \leftarrow u'$; **else** $u^n \leftarrow u^{n-1}$;
  end
  cluster $\{u^0, \ldots, u^{N_{\text{temp}}}\}$ into J subsets: $\Theta_1, \ldots, \Theta_J$;
  for $j \leftarrow 1$ **to** J **do**
    $N_j \leftarrow$ sample size of Θ_j, $w_j \leftarrow N_j/N_{\text{temp}}$;
    compute parameters x_j and h_j using (4.4);
    compute μ_j using (3.7);
  end
  $\mu \leftarrow \sum_{j=1}^{J} w_j \mu_j$;
end
draw $u^0 \sim \mu$;
for $n \leftarrow 1$ **to** N **do**
  draw $u' \sim \mu$;
  draw $a \sim U[0,1]$ and compute

$$A \leftarrow \min\left\{1, \frac{d\mu^{y}}{d\mu}(u')\frac{d\mu}{d\mu^{y}}\left(u^{n-1}\right)\right\};$$

  if $A > a$ **then** $u^n \leftarrow u'$; **else** $u^n \leftarrow u^{n-1}$;
  if $(n < N_{\max})\&(n \mod N_{\text{adp}} = 0)$ **then**
    cluster $\{u^0, \ldots, u^n\}$ into J subsets: $\Theta_1, \ldots, \Theta_J$;
    for $j \leftarrow 1$ **to** J **do**
      $N_j \leftarrow$ sample size of Θ_j, $w_j \leftarrow N_j/n$;
      compute parameters x_j and h_j using (4.4);
      compute μ_j using (3.7);
    end
    $\mu \leftarrow \sum_{j=1}^{J} w_j \mu_j$;
  end
end

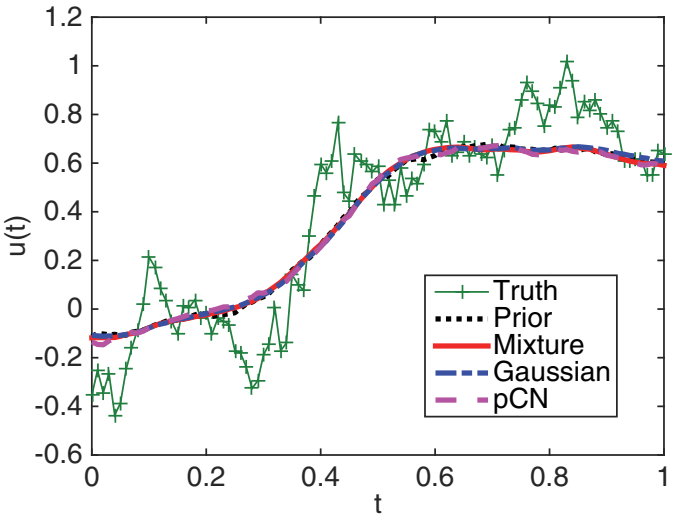

FIG. 1. *(For example 1) The posterior mean computed with the four different MCMC schemes. The truth is also plotted for comparison.*

exponential covariance function:

$$C(t,t') = \exp\left(-\frac{|t-t'|}{2}\right). \tag{5.2}$$

The true coefficient $u(t)$ is a realization from the prior (shown in Figure 1) and the data is simulated accordingly.

We now draw samples from the posterior of $u(t)$ with four different MCMC schemes: prior based IS, adaptive IS with Gaussian approximation, adaptive IS with Gaussian mixtures, and the random walk pCN (RW-pCN). In each MCMC scheme, 3×10^5 draws are generated. In the prior based IS, one simply proposes according to the prior distribution, and no adaptation is used. In the adaptive IS with Gaussian approximation, the proposal is restricted to be a single Gaussian (i.e., $J = 1$), and in this case clustering is not needed. In both of the adaptive IS methods, the parameters are updated after every 1000 draws, and the parameter adaptation is terminated in the last 10^5 iterations. We do not use tempering in this example. The RW-pCN algorithm used in this work iterates as follows:

1. Propose $u_{\text{proposed}} = \sqrt{1-\beta^2}u_{\text{current}} + \beta w$, where $w \sim \mu_0$.
2. Let $u_{\text{next}} = u_{\text{proposed}}$ with probability

$$a = \min\{1, \exp(\Phi(u_{\text{proposed}}) - \Phi(u_{\text{current}}))\},$$

and let $u_{\text{next}} = u_{\text{current}}$ with probability $1 - a$.

In this example we use $\beta = 0.1$. Note that, in all the numerical examples, we choose the stepsize β so that the resulting acceptance probability is in the range $20\% - 30\%$, as is recommended in [33].

In Figure 1, we show the posterior mean computed by the four MCMC schemes, while the truth is also shown for comparison purpose. One can see that the results of the four algorithms are nearly identical, suggesting that all the algorithms can estimate the posterior mean to a similar level of accuracy. We then use the OMF as

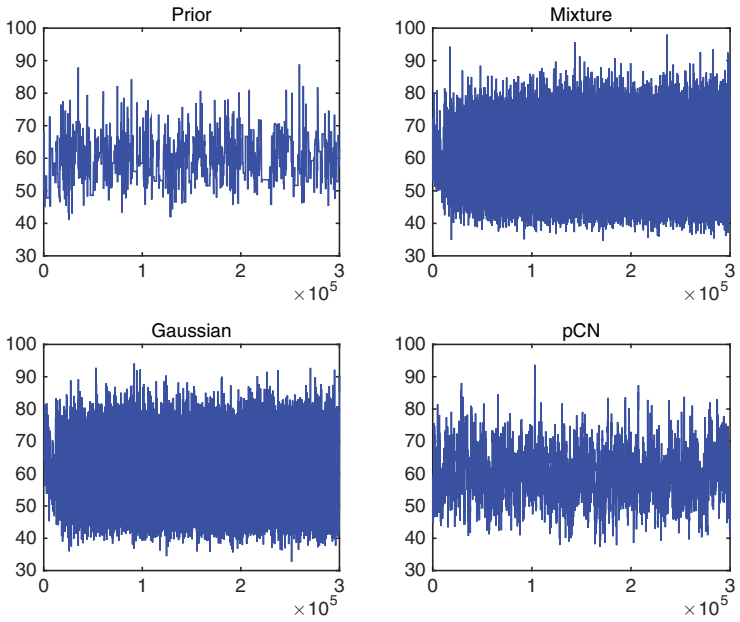

FIG. 2. *(For example 1) The trace plots of the OMF for the four different MCMC schemes.*

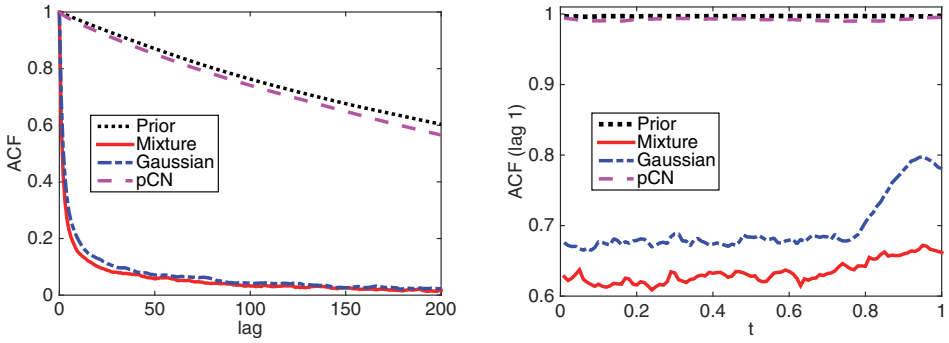

FIG. 3. *(For example 1) ACF for the four different MCMC methods. Left: ACF of the OMF plotted as a function of lags. Right: the lag 1 ACF for u at each grid point.*

an indicative parameter and show the trace plots of it in Figure 2. We see from the plots that the two adaptive IS algorithms achieve a much faster mixing rate than the other two methods. To further compare the efficiency of the methods, we compute the autocorrelation functions (ACF) of various quantities with the samples drawn by the four methods, and plot the ACF results in Figure 3. In particular, we plot the ACF of the OMF as a function of lag in Figure 3 (left) and show the lag 1 ACF for the unknown u at each grid point in Figure 3 (right). It can be seen from the figure that our adaptive algorithms with single Gaussian proposal and with mixtures both result in much lower ACF values than the other two methods. When comparing the two adaptive algorithms, the mixture proposal outperforms the single Gaussian. For the

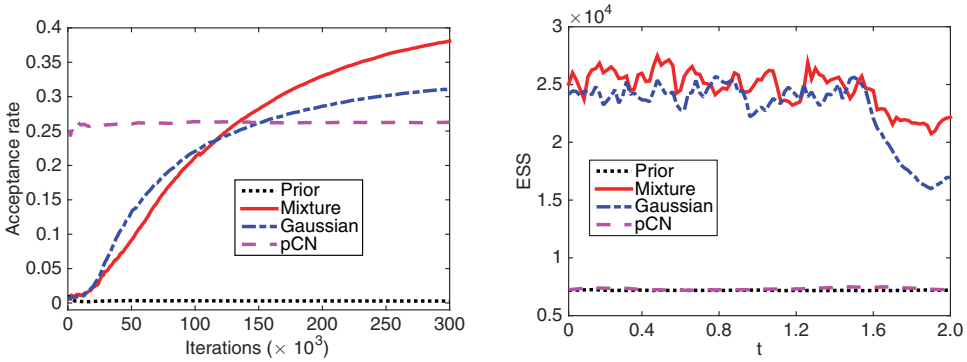

FIG. 4. *(For example 1) Left: the acceptance rate of the four MCMC schemes. Right: the ESS at each grid point.*

IS algorithms, the acceptance probability is also a useful performance indicator, where higher acceptance rates are usually preferred, while it is not the case for random walk algorithms [33]. In Figure 4 (left) we plot the acceptance probability as a function of iterations for all the methods. For the three IS algorithms, one can see that the two adaptive algorithms have significantly higher acceptance probability than the prior based method. Meanwhile, the acceptance probability of IS with mixtures is higher than that of the one with the single Gaussian. The effective sample size (ESS) is another common measure of the sampling efficiency of MCMC [18]. ESS is computed by

$$\mathrm{ESS} = \frac{N}{1+2\tau},$$

where τ is the integrated autocorrelation time and N is the total sample size, and it gives an estimate of the number of effectively independent draws in the chain. We computed the ESS of the unknown u at each grid point and show the results in Figure 4 (right). Once again, the plots indicate that the adaptive algorithms produce much more effectively independent samples than the prior based IS and the RW-pCN, while the mixture proposal outperforms the single Gaussian one in most of the dimensions. In summary, in this simple nonlinear inverse problem, we show that our adaptive algorithms are significantly more efficient than the prior based IS and the RW-pCN. Meanwhile, the mixture proposal outperforms the single Gaussian one, indicating that the more flexible mixture representation does improve the efficiency.

5.2. A bimodal likelihood function example. Our second example is an artificially constructed bimodal problem. Once again we assume the unknown $u \in X = L^2([0,1])$ and the prior is a zero mean Gaussian measure with the same covariance function equation (5.2) as the first example. We consider a bimodal likelihood function, given by

$$\exp(-\Phi(u)) \propto \exp\left(-\frac{1}{2}\|u-\sin(2\pi t)\|_2^2\right) + \exp\left(-\frac{1}{2}\|u+\sin(2\pi t)\|_2^2\right),$$

and it can be verified that the $\Phi(\cdot)$ chosen this way satisfies [8, Assumptions (6.1)]. It is easy to see that the posterior distribution should have two modes: one is close to $\sin(2\pi t)$ and the other is close to $-\sin(2\pi t)$.

We draw samples from the posterior of $u(t)$ with the same four MCMC schemes used in the first example, and in each MCMC scheme, 5×10^5 draws are generated. In

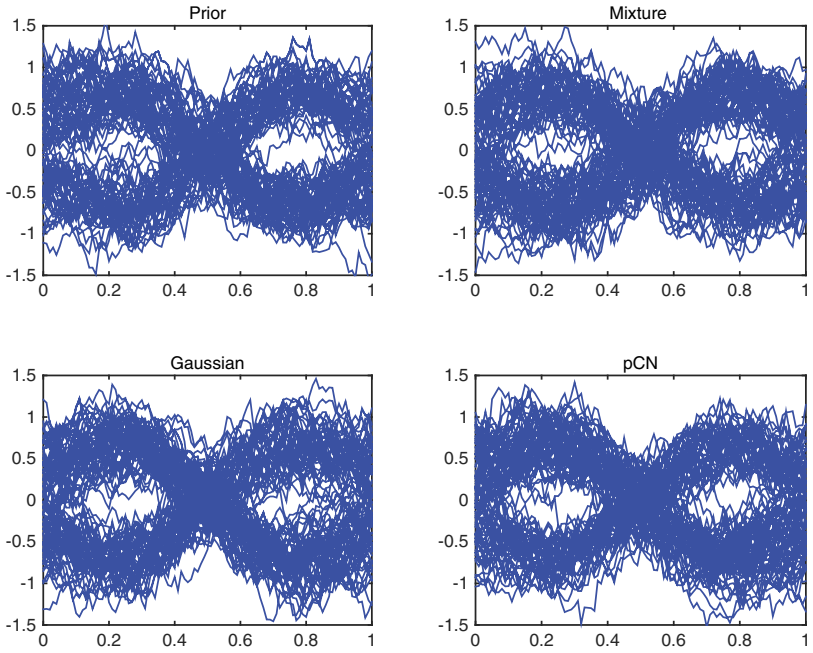

FIG. 5. *(For example 2)* 100 *samples randomly selected from the chain drawn by each method.*

both of the adaptive IS methods, the parameters are updated after every 1000 draws, and the adaptation is terminated in the last 10^5 iterations, with no tempering used. In the RW-pCN, we choose $\beta = 0.5$. In all the computations, 100 grid points are used to represent the unknown function u.

As has been mentioned, the posterior distribution has two modes and we shall examine whether the algorithms can capture both of them. In this respect, we randomly select 100 samples from the chain generated by each algorithm and plot them in Figure 5. We can see that the results of each algorithm can capture the two models of the posterior. Next we shall compare the efficiency of the four algorithms. As before, we first show the trace plots of the OMF for the four algorithms in Figure 6 and one can see that the results of the two adaptive methods and pCN all obtain fairly good mixing results, while the prior based IS seems to have a much slower mixing rate than the other three. Figure 7 (left) plots the ACF of the OMF as a function of lag and Figure 7 (right) shows the lag 1 ACF for the unknown at each grid point. Both figures indicate that the adaptive IS with mixtures has the best performance in terms of ACF values. Figure 8 (left) plots the acceptance rate against the number of iterations, which shows that the three IS algorithms perform very differently: the prior based IS results in an acceptance rate less than 1%, the adaptive IS with one Gaussian results in a rate up to 17%, and that of the adaptive IS with mixtures rises to around 80% as the iteration proceeds. We compute the ESS of each dimension and show the results in Figure 8 (right), and we see that the ESS of the adaptive IS with mixtures is significantly higher than that of the other three methods, indicating that the adaptive IS with mixtures has a substantial advantage in this multimodal problem.

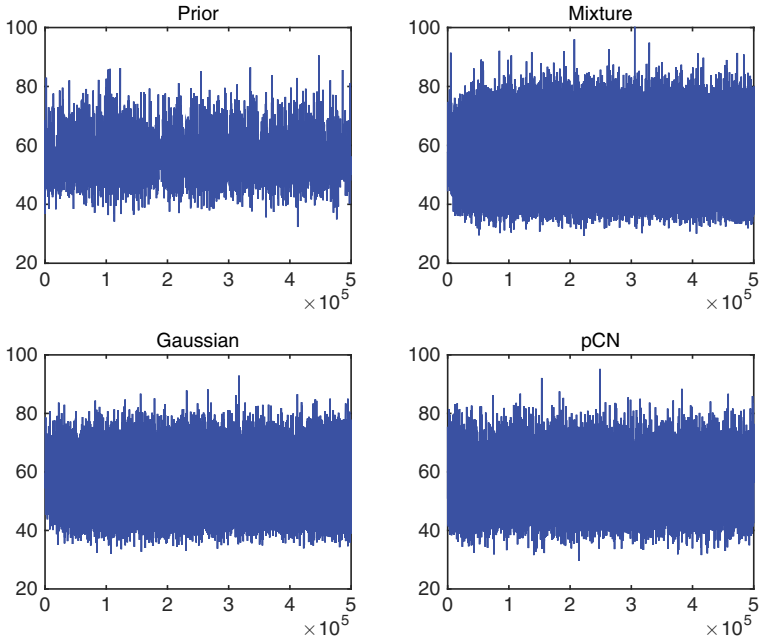

FIG. 6. *(For example 2) The trace plots of the OMF for the four different MCMC schemes.*

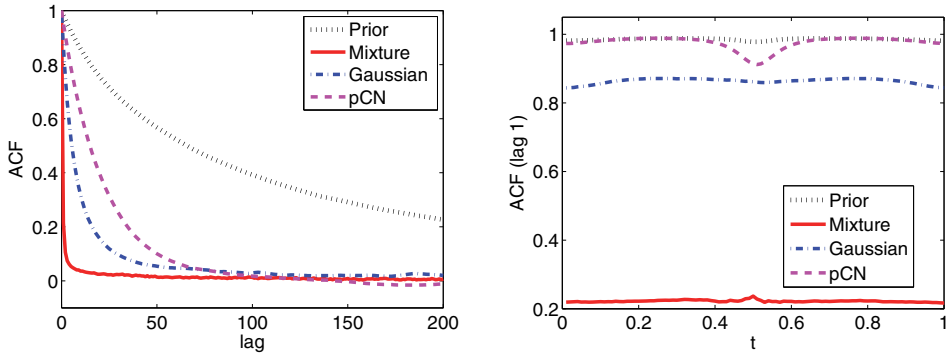

FIG. 7. *(For example 2) ACF for the four different MCMC methods. Left: ACF of the OMF plotted as a function of lags. Right: the lag 1 ACF for u at each grid point.*

Finally to understand the limitation of the proposed method, we test it on another bimodal likelihood function:

$$\exp(-\Phi(u)) \propto \exp\left(-\frac{1}{2}\|u - 2\sin(2\pi t)\|_2^2\right) + \exp\left(-\frac{1}{2}\|u + 2\sin(2\pi t)\|_2^2\right).$$

We drew 5×10^5 samples with the mixture based IS algorithm and with the pCN. We plot the mean of the samples drawn by both methods in Figure 9 (left), and in 9 (right), we plot 100 samples drawn by each algorithms. It can be seen from the figures that both methods can only capture one mode of the posterior distribution,

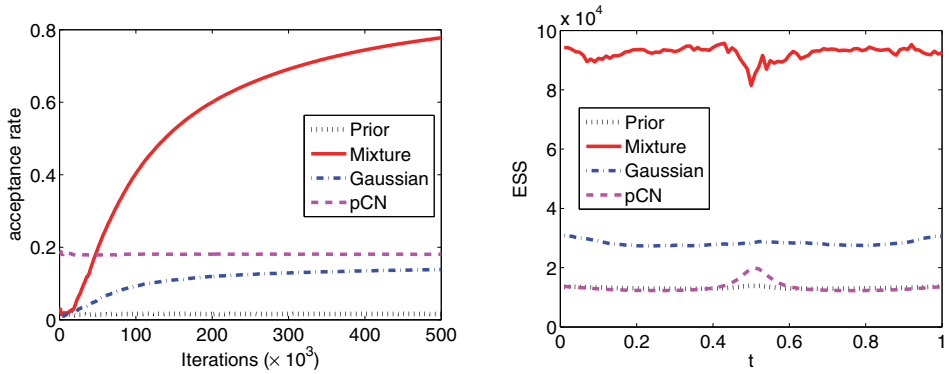

FIG. 8. *(For example 2) Left: the acceptance rate of the four MCMC schemes. Right: the ESS at each grid point.*

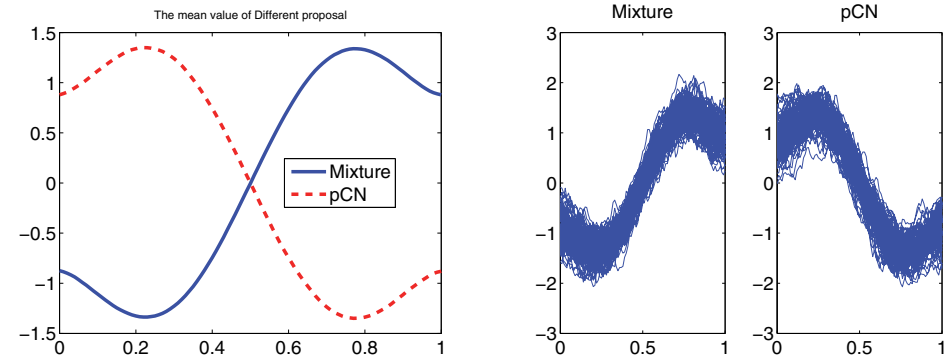

FIG. 9. *(For example 2) Left: the sample mean of the mixture based IS method (solid) and that of the pCN method (dashed). Right: samples drawn by the mixture IS method and by the pCN method.*

indicating that the problem becomes challenging for our method and the pCN when the modes of the target distribution are far apart.

5.3. Inverse heat conduction under model uncertainty. Our last example is the inverse heat conduction (IHC) problems, which consist of estimating temperature or heat flux density on an inaccessible boundary from a measured temperature history inside a solid. These problems have been studied over several decades due to their importance in a variety of scientific and engineering applications [4]. The IHC problems become nonlinear if the thermal properties are temperature dependent, where the inversion is significantly more difficult than the linear ones. In this example we consider a one dimensional heat conduction equation

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left[c(u)\frac{\partial u}{\partial x}\right] \tag{5.3}$$

with initial $u(x,0) = u_o(x)$. Here x and t are the spatial and temporal variable, $u(x,t)$ is the temperature, and $c(u)$ is the temperature dependent thermal conductivity, and the length of the medium is L, all in dimensionless units. We now assume that a heat flux is injected through the left boundary ($x = 0$), yielding a Neumann boundary

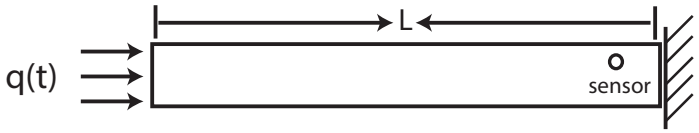

FIG. 10. *Schematic diagram of the IHC problem.*

condition:

$$\frac{\partial}{\partial x}u(0,t) = q(t).$$

The boundary condition (BC) at $x = L$ is subject to uncertainty: with probability 0.8 it is

$$\frac{\partial}{\partial x}u(L,t) = 0, \tag{5.4a}$$

and with probability 0.2 it is

$$\frac{\partial}{\partial x}u(L,t) = -u. \tag{5.4b}$$

The interpretation is that the system has two possible states: one with a perfectly insulted boundary at $x = L$, and the other with heat diffusion at $x = L$.

Suppose that we place a temperature sensor in the medium ($x = x_s$) and the goal is to infer the heat flux $q(t)$ for $t \in [0, T]$ from the temperature history measured by the sensor in the time interval. The schematic of this problem is shown in Figure 10. A similar problem without model uncertainty has been studied in [22].

In the simulation, we let $L = 1$, $T = 2$, $c(u) = u^2 + 1$, $x_s = 0.9$, and the initial condition be $u_o(x) = 0$. The temperature is measured 50 times (equally spaced) and the error in each measurement is assumed to be an independent zero-mean Gaussian random variable with variance 0.1^2. We assume the prior on $q(t)$ is a stationary zero-mean Gaussian process with a squared exponential covariance function:

$$C(t,t') = \exp\left(-\frac{|t-t'|^2}{2d^2}\right), \tag{5.5}$$

where $d = 0.3$. The "truth flux" $q(t)$ is a realization of the prior (shown in Figure 12) and the data is simulated with the generated flux $q(t)$ and the boundary condition (5.4b). In this problem the likelihood function becomes

$$\frac{d\mu^y}{d\mu_0} = 0.8\exp(-\Phi_1(u)) + 0.2\exp(-\Phi_2(u)),$$

where $\Phi_1(u)$ corresponds to (5.3) with BC (5.4a) and $\Phi_2(u)$ corresponds to (5.3) with BC (5.4b).

We draw samples from the posterior of $u(t)$ with the four MCMC schemes used in the previous examples. In each MCMC scheme, 1.5×10^5 draws are generated. In both of the adaptive IS methods, the parameters are updated after every 500 draws, and the adaptation is terminated after 10^5 draws. To accelerate the convergence, we use tempering in the first 11 iterations (5,500 draws) with tempering parameter $\lambda = (i-1)/10$ for $i = 1, \ldots, 11$. In the RW-pCN, we choose $\beta = 0.1$.

We first show the trace plot of the OMF in Figure 11, and it is quite clear that the results of the two adaptive methods are better than those of the prior based IS

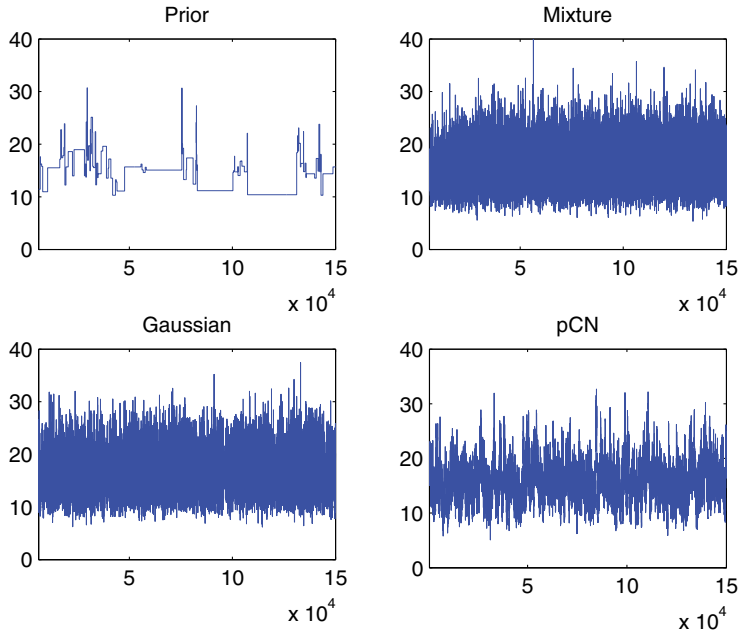

FIG. 11. *(For example 3) The trace plots of the OMF for the four different MCMC schemes.*

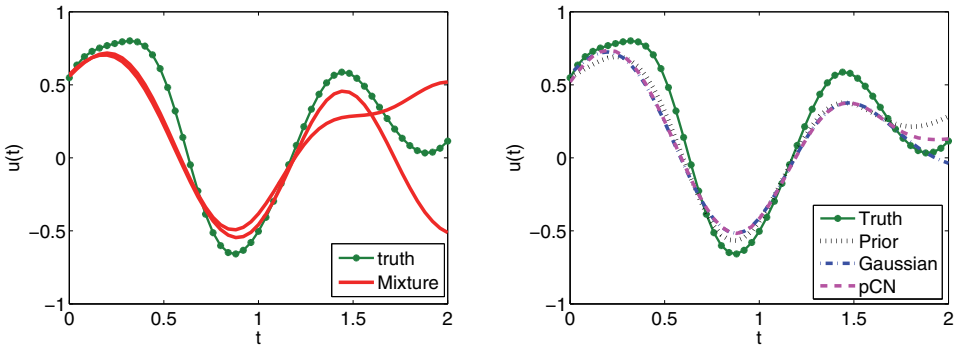

FIG. 12. *(For example 3) Left: the means of the samples in each cluster of the chain drawn by the IS with mixtures and the true flux. Right: the means of the samples drawn by the adaptive IS with a single Gaussian, the prior based IS, and the RW-pCN.*

and the pCN. Because of the multimodality of the likelihood function, the posterior may have multiple modes, and to verify this, we apply the K-means method described in section 4 to cluster the samples drawn by the four methods. The samples of the adaptive IS with mixtures can be successfully classified into two groups and we plotted the mean of each group in Figure 12 (left), compared against the true heat flux. The K-means method, however, fails to separate the samples drawn by the other three methods, likely because the chains have not reached the target posterior distribution yet. We plot the means of the samples of the three methods in Figure 12 (right). Like the previous examples we show the ACF results of the four methods in Figures 13, and

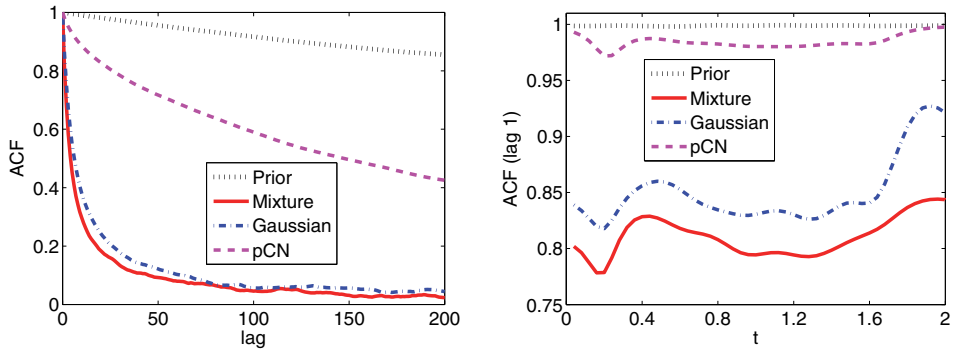

FIG. 13. *(For example 3) ACF for the four different MCMC methods. Left: ACF of the OMF plotted as a function of lags. Right: the lag 1 ACF for u at each grid point.*

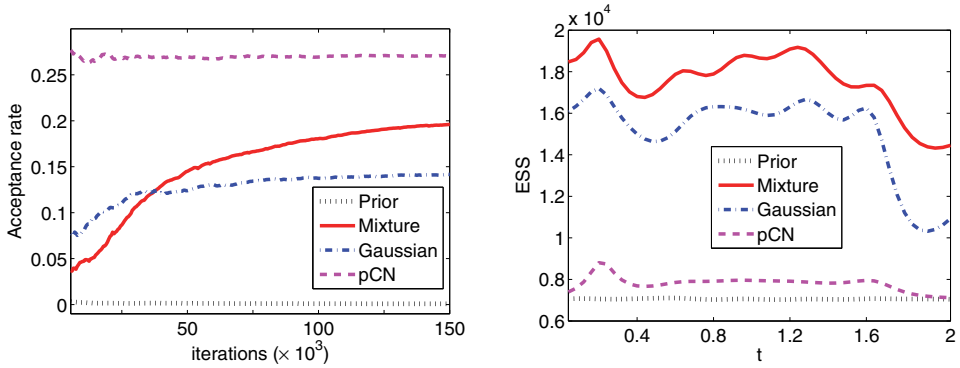

FIG. 14. *(For example 3) Left: the acceptance rate of the four MCMC schemes. Right: the ESS at each grid point.*

the acceptance rates and the ESS in Figures 14. In all the plots, the adaptive IS with mixtures exhibits the best performance, followed by the IS with a single Gaussian.

6. Conclusions. In conclusion, we have presented an adaptive IS algorithm to implement Bayesian inference for functions. Namely, we choose a Gaussian mixture with a particular parametrization as our proposal and adaptively adjust the parameter values using sample history. We also develop an efficient algorithm based on clustering to compute the parameter values in each iteration. We demonstrate the efficiency of the proposed method with numerical examples and in particular we show that it performs well for multimodal posteriors. We emphasize that the proposed method is easy to implement, treating the problem as a black box model, and requiring no information on the mathematical structure of the forward model.

As has been demonstrated by the numerical examples, the mixture proposals can generally provide faster mixing rates than the single Gaussian, thanks to their higher flexibility. On the other hand, given that the Gaussian approximation is less complex computationally (without the clustering step), we recommend to use the single Gaussian approximation in problems where the posterior distributions do not deviate too much from a Gaussian measure, and to use mixtures for strongly nonGaussian posteriors.

There are number of possible extensions of the work. First in this work we approx-

imate the solution to the KLD minimization problem with clustering. It is possible that if we can modify the standard EM algorithm and use it to solve the optimization problem directly, we may obtain a better mixture proposal in each iteration and improve the sampling efficiency. Second, the intrinsic dimensionality K is of essential importance for our method, and in the present work, K is determined rather heuristically. Thus developments of more effective and theoretically justified methods certainly deserve further studies. Finally, the algorithm developed here is based on an independence sampler, and we are also interested in extending the ideas to the development of adaptive random walk algorithms for functions. We plan to investigate these problems in the future.

REFERENCES

[1] C. Andrieu and J. Thoms, *A tutorial on adaptive MCMC*, Statist. Comput., 18 (2008), pp. 343–373.

[2] Y. Atchade, G. Fort, E. Moulines, and P. Priouret, *Adaptive Markov chain Monte Carlo: Theory and methods*, in Bayesian Time Series Models, Cambridge University Press, Cambridge, 2011, pp. 32–51.

[3] Y. F. Atchade, *An adaptive version for the Metropolis adjusted Langevin algorithm with a truncated drift*, Methodol. Comput. Appl. Probab., 8 (2006), pp. 235–254.

[4] J. Beck, B. Blackwell, and C. St Clair, *Inverse Heat Conduction: Ill-Posed Problems*, John Wiley and Sons, New York, 1985.

[5] A. Beskos, *A stable manifold MCMC method for high dimensions*, Statist. Probab. Lett., 90 (2014), pp. 46–52.

[6] A. Beskos, G. Roberts, A. Stuart, et al., *Optimal scalings for local Metropolis–Hastings chains on nonproduct targets in high dimensions*, Ann. Appl. Probab., 19 (2009), pp. 863–898.

[7] A. Beskos, G. Roberts, A. Stuart, and J. Voss, *MCMC methods for diffusion bridges*, Stoch. Dyn., 8 (2008), pp. 319–350.

[8] S. L. Cotter, G. O. Roberts, A. Stuart, D. White, et al., *MCMC methods for functions: Modifying old algorithms to make them faster*, Statist. Sci., 28 (2013), pp. 424–446.

[9] T. Cui, K. J. Law, and Y. M. Marzouk, *Dimension-Independent Likelihood-Informed MCMC*, J. Comput. Phys., 304 (2016), pp. 109–137.

[10] G. Da Prato, *An Introduction to Infinite-Dimensional Analysis*, Springer, New York, 2006.

[11] M. Dashti, K. J. H. Law, A. M. Stuart, and J. Voss, *MAP estimators and their consistency in Bayesian nonparametric inverse problems*, Inverse Problems, 29 (2013), 095017.

[12] J. Gåsemyr, *On an adaptive version of the Metropolis–Hastings algorithm with independent proposal distribution*, Scand. J. Stat., 30 (2003), pp. 159–173.

[13] P. Giordani and R. Kohn, *Adaptive independent Metropolis–Hastings by fast estimation of mixtures of normals*, J. Comput. Graph. Statist., 19 (2010), pp. 243–259.

[14] H. Haario, E. Saksman, and J. Tamminen, *An adaptive Metropolis algorithm*, Bernoulli, 7 (2001), pp. 223–242.

[15] N. L. Hjort, C. Holmes, P. Müller, and S. G. Walker, *Bayesian Nonparametrics*, Vol. 28, Cambridge University Press, Cambridge, 2010.

[16] L. Holden, R. Hauge, and M. Holden, *Adaptive independent Metropolis–Hastings*, Ann. Appl. Probab., 19 (2009), pp. 395–413.

[17] J. Kaipio and E. Somersalo, *Statistical and Computational Inverse Problems*, Vol. 160, Springer, New York, 2005.

[18] R. E. Kass, B. P. Carlin, A. Gelman, and R. M. Neal, *Markov chain Monte Carlo in practice: A roundtable discussion*, Amer. Statist., 52 (1998), pp. 93–100, https://doi.org/10.2307/2685466.

[19] J. M. Keith, D. P. Kroese, and G. Y. Sofronov, *Adaptive independence samplers*, Statist. Comput., 18 (2008), pp. 409–420.

[20] K. J. Law, *Proposals which speed up function-space mcmc*, J. Comput. Appl. Math., 262 (2014), pp. 127–138.

[21] J. Li, *A note on the Karhunene–Loeve expansions for infinite-dimensional Bayesian inverse problems*, Statist. Probab. Lett., 106 (2015), pp. 1–4.

[22] J. Li and Y. M. Marzouk, *Adaptive construction of surrogates for the Bayesian solution of inverse problems*, SIAM J. Sci. Comput., 36 (2014), pp. A1163–A1186.

[23] J. S. Liu, *Metropolized independent sampling with comparisons to rejection sampling and importance sampling*, Statist. Comput., 6 (1996), pp. 113–119.
[24] T. Marshall and G. Roberts, *An adaptive approach to Langevin MCMC*, Statist. Comput., 22 (2012), pp. 1041–1057.
[25] J. Martin, L. C. Wilcox, C. Burstedde, and O. Ghattas, *A stochastic Newton MCMC method for large-scale statistical inverse problems with application to seismic inversion*, SIAM J. Sci. Comput., 34 (2012), pp. A1460–A1487.
[26] J. C. Mattingly, N. S. Pillai, A. M. Stuart, et al., *Diffusion limits of the random walk Metropolis algorithm in high dimensions*, Ann. Appl. Probab., 22 (2012), pp. 881–930.
[27] G. McLachlan and D. Peel, *Finite Mixture Models*, John Wiley and Sons, New York, 2004.
[28] N. Petra, J. Martin, G. Stadler, and O. Ghattas, *A computational framework for infinite-dimensional Bayesian inverse problems, Part* II: *Stochastic Newton MCMC with application to ice sheet flow inverse problems*, SIAM J. Sci. Comput., 36 (2014), pp. A1525–A1555.
[29] F. Pinski, G. Simpson, A. Stuart, and H. Weber, *Kullback–Leibler approximation for probability measures on infinite dimensional spaces*, SIAM J. Math. Anal., 46 (2015), pp. 4091–4122.
[30] F. J. Pinski, G. Simpson, A. M. Stuart, and H. Weber, *Algorithms for Kullback–Leibler approximation of probability measures in infinite dimensions*, SIAM J. Sci. Comput., 37 (2015), pp. A2733–A2757.
[31] G. O. Roberts, A. Gelman, W. R. Gilks, et al., *Weak convergence and optimal scaling of random walk Metropolis algorithms*, Ann. Appl. Probab., 7 (1997), pp. 110–120.
[32] G. O. Roberts and J. S. Rosenthal, *Examples of adaptive MCMC*, J. Comput. Graph. Statist., 18 (2009), pp. 349–367.
[33] G. O. Roberts, J. S. Rosenthal, et al., *Optimal scaling for various Metropolis–Hastings algorithms*, Statist. Sci., 16 (2001), pp. 351–367.
[34] A. M. Stuart, *Inverse problems: A Bayesian perspective*, Acta Numer., 19 (2010), pp. 451–559.
[35] L. Tierney, *Markov chains for exploring posterior distributions*, Ann. Statist., 22 (1994), pp. 1701–1728.
[36] L. Tierney, *A note on Metropolis–Hastings kernels for general state spaces*, Ann. Appl. Probab., 8 (1998), pp. 1–9.
[37] C. J. Wu, *On the convergence properties of the EM algorithm*, Ann. Statist., 11 (1983), pp. 95–103.