SIAM J. SCI. COMPUT.
Vol. 36, No. 3, pp. B427–B449

NONLINEAR GOAL-ORIENTED BAYESIAN INFERENCE: APPLICATION TO CARBON CAPTURE AND STORAGE*

CHAD LIEBERMAN† AND KAREN WILLCOX†

Abstract. In many engineering problems, unknown parameters of a model are inferred in order to make predictions, to design controllers, or to optimize the model. When parameters are distributed (continuous) or very high-dimensional (discrete) and quantities of interest are low-dimensional, parameters need not be fully resolved to make accurate estimates of quantities of interest. In this work, we extend goal-oriented inference—the process of estimating predictions from observed data without resolving the parameter, previously justified theoretically in the linear setting—to Bayesian statistical inference problem formulations with nonlinear experimental and prediction processes. We propose to learn the joint density of data and predictions offline using Gaussian mixture models. When data are observed online, we condition the representation to arrive at a probabilistic description of predictions given observed data. Our approach enables real-time estimation of uncertainty in quantities of interest and renders tractable high-dimensional PDE-constrained Bayesian inference when there exist low-dimensional output quantities of interest. We demonstrate the method on a realistic problem in carbon capture and storage for which existing methods of Bayesian parameter estimation are intractable.

Key words. goal-oriented inference, carbon capture and storage, partial differential equations, nonlinear, Bayesian

AMS subject classifications. 62F15, 35F30

DOI. 10.1137/130928315

1. Introduction.

Despite many years of research and development of algorithms for estimation of distributed parameters, Bayesian statistical inference of distributed parameters remains a considerable challenge. The difficulties of high-dimensional parameter spaces are exacerbated when likelihood models depend on partial differential equations (PDEs). Recognizing that estimation of parameters is often a step in pursuit of making predictions, and subsequently decisions, we established goal-oriented inference [18], a novel approach to estimation when the goal is to obtain accurate predictions from data without regard for accuracy in parameter estimation. In short, we refocus resources toward estimating the predictions, the true target of the parameter estimation. By exploiting the low-dimensionality of the map from data to predictions, we expose weaknesses in the experimental design, discover the major sources of uncertainty in the predictions, and circumvent the most expensive online computations, making feasible pseudo real-time deployment. In this work we extend goal-oriented inference to Bayesian formulations for systems governed by nonlinear PDEs.

When predictions depend on a system with an unknown distributed parameter, a typical approach is to perform parameter inference on collected data before passing the estimate to the prediction simulation. Data informative about the parameter are first observed. An inference procedure, deterministic or statistical, is then employed

*Submitted to the journal's Computational Methods in Science and Engineering section July 9, 2013; accepted for publication (in revised form) February 3, 2014; published electronically May 22, 2014. This work was supported by the Air Force Office of Scientific Research MURI on Uncertainty Quantification, grant FA9550-09-0613 (Program Manager Dr. F. Fahroo) and the Department of Energy DiaMonD MMICC (Program Manager A. Landsberg).
http://www.siam.org/journals/sisc/36-3/92831.html
†Department of Aeronautics and Astronautics, MIT, Cambridge, MA 02139 (celieber@mit.edu, kwillcox@mit.edu).

to make an estimate of the parameter based on the observed data and an assumed model. The parameter estimate is then used as an input to a simulation that will make predictions.

The inference step is ill-posed and computationally very expensive. Bayesian statistical formulations treat ill-posedness in the inverse problem by specifying a prior distribution over the parameter space, a representation of one's belief in the relative likelihood of different parameters before data are observed [26, 14]. In the limited data context, however, this prior distribution will not be forgotten through Bayesian updates; its effect remains even after all data are processed [28]. There are many computational challenges of Bayesian inference of distributed parameters for models described by PDEs. Markov chain Monte Carlo (MCMC) methods are a popular technique for exploring the posterior distribution of the Bayesian inference [30, 19, 22, 12, 29, 2, 10]. However, MCMC requires efficient exploration of parameter space and many samples, meaning many PDE solves. Efficient exploration is challenging in high-dimensional parameter spaces and many PDE solves makes the computation intractable for the pseudo real-time setting.

These challenges can be addressed by incorporating the final objectives of the inference into the problem statement. In many engineering applications, typically parameter estimation is not the goal but rather an intermediate step in making predictions of the system. Extending the problem statement to include the prediction step makes the problem easier, not more challenging, since we can exploit the low-dimensionality of the required output. Goal-oriented inference sacrifices accuracy in parameter estimation to achieve efficiency in accurate estimation of these predictions.

There are two central components to the developments in goal-oriented inference: an experimental process by which data are obtained and a prediction process yielding the target of our estimation. An *experimental process* is a physical system, or model thereof, as well as an observation paradigm given by experimental design, that produces data depending on the existing, but unknown, parameter. The data are corrupted by noise, which we will regularly model as additive Gaussian. A *prediction process* is a physical system, or model thereof, that yields an estimate of a quantity of interest given a specified value of model parameter. Like the experimental process, the prediction process will typically also consist of the composition of a PDE operator and an observation operator. The PDE need not be the same as the experimental process; however, the two must be linked by a consistent description of the unknown parameter. A block diagram of the two processes, connected by the parameter, is shown in Figure 1.

A critical component to goal-oriented inference is the offline interrogation of the experimental and prediction processes as they depend on the consistent representation of the unknown parameter. Before experiments are conducted and data observed, we propose to learn the joint density of experimental data and predictions as determined by the underlying processes, noise model, and prior distribution on parameter. In the online stage, when experiments are performed and data collected, we condition the joint density on the observed data to obtain the posterior predictive density—the probabilistic description of our prediction estimate. The high-dimensional parameter, which we cannot resolve with limited data, is never inferred.

This paper is organized as follows. In section 2 we present our approach for goal-oriented inference in the context of nonlinear Bayesian statistical inverse problems. We discuss an important problem in carbon capture and storage in section 3. We use data to directly predict trapped volume of carbon dioxide without inferring subsurface properties. In section 4 we present the numerical results for our experiments and

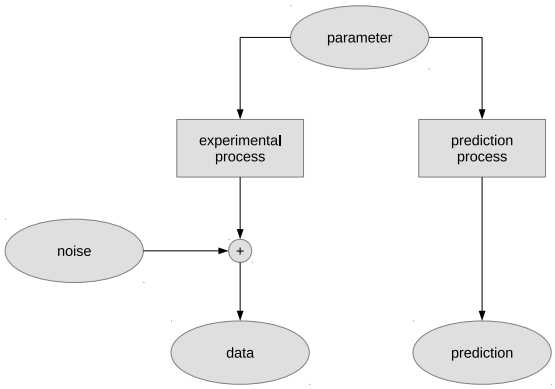

FIG. 1. *The experiment and prediction processes both take as input the shared parameter. The output of the experimental process is corrupted by noise to yield the observed data. The output of the prediction process yields the prediction quantity of interest.*

provide additional discussion of the method and its performance. Finally, we present conclusions in section 5.

2. Nonlinear Bayesian goal-oriented inference. In this section we establish notation, define the problem statement, and present the details of our approach for goal-oriented inference in the context of Bayesian statistical inference problems constrained by nonlinear PDEs with distributed parameters.

2.1. Problem statement. We now present the formal problem statement for nonlinear goal-oriented inference in the statistical setting. We begin by writing the statement for the parameter inference problem and then propagating the posterior through to the prediction. The resulting posterior predictive distribution is the target of our goal-oriented inference.

Let $\mathbf{f}_e(\boldsymbol{\mu}) : \mathbb{R}^q \to \mathbb{R}^r$ be a general nonlinear function representing the experimental process mapping a q-dimensional parameter to expected observational data of dimension $r \ll q$. The function will usually embed a PDE operator and an observation operator. In the carbon capture and storage example presented in the next chapter, $\mathbf{f}_e$ corresponds to the solution of the single-phase flow equations and observation of bottom hole pressure in some wells. We make observations $\mathbf{y}_d = \mathbf{f}_e(\boldsymbol{\mu}) + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is assumed to be a multivariate normal noise such that $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbf{I})$ with noise variance σ_n^2.

Let $p(\boldsymbol{\mu})$ be the prior probability density function of the parameter. In Bayesian statistical approaches to parameter inference, the prior encompasses knowledge of the parameter before data are observed. There are several approaches for choosing the prior density and it has been the focus of significant controversy in the community [1, 3, 9]. Our goal in this work is not to improve the existing parameter inference approach but rather to reformulate it for the goal-oriented inference context. Therefore, we will simply take $p(\boldsymbol{\mu})$ as given, assuming that a routine to generate samples from the prior is available. For our numerical experiments, we will assume that $\boldsymbol{\mu}$ is a lognormal random process with exponential covariance kernel.

With the additive Gaussian noise model assumed above, our likelihood function $L(\boldsymbol{\mu}; \mathbf{y}_d)$ comes directly from the relation $\boldsymbol{\epsilon} = \mathbf{y}_d - \mathbf{f}_e(\boldsymbol{\mu}) \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbf{I})$. That is, the likelihood represents the ratio (as a function of $\boldsymbol{\mu}$) of the posterior to the prior. In the

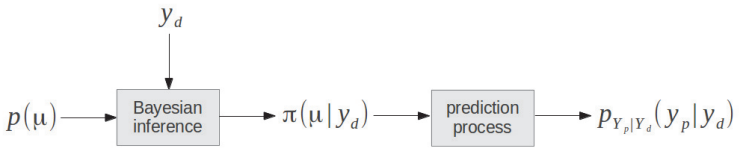

FIG. 2. *Block diagram of Bayesian statistical inference for prediction. The prior $p(\boldsymbol{\mu})$ is specified over the parameters representing the a priori relative likelihood of parameters. Data are observed and incorporated by Bayes's rule to yield the posterior $\pi(\boldsymbol{\mu}|\mathbf{y}_d)$. The posterior is then propagated forward through the prediction process to obtain the posterior predictive $p_{Y_p|Y_d}(y_p|\mathbf{y}_d)$.*

Bayesian update, parameters that are more likely to have generated observed data undergo a more significant change from prior to posterior. Although Gaussian noise models are a typical choice in the PDE-constrained statistical inference literature, our method is amenable to other likelihood models as well.

The posterior probability density function $\pi(\boldsymbol{\mu}|\mathbf{y}_d)$ can be readily expressed by Bayes's rule, i.e.,

$$\pi(\boldsymbol{\mu}|\mathbf{y}_d) \propto L(\boldsymbol{\mu};\mathbf{y}_d)p(\boldsymbol{\mu}). \tag{2.1}$$

The posterior (2.1) is the solution to the Bayesian statistical inference problem. It represents the new probabilistic description of the parameter after the belief is updated based on observed data, accounting for both prior and likelihood models.

We now introduce the prediction process $\mathbf{f}_p(\boldsymbol{\mu}) : \mathbb{R}^q \to \mathbb{R}^s$, a measurable function from the q-dimensional parameter to an s-dimensional prediction output quantity of interest. The function is often a composition of an output functional and a PDE operator. For the carbon capture and storage example, $\mathbf{f}_p$ represents the calculation of trapped carbon dioxide volume under a given injection scenario governed by a vertical equilibrium approximation of the two-phase flow equations.[1]

The ultimate goal of our inference is to obtain the posterior predictive probability density function $p_{Y_p|Y_d}(\mathbf{y}_p|\mathbf{y}_d)$ which represents the push forward of the posterior measure $\pi(\boldsymbol{\mu}|\mathbf{y}_d)$ through the function $\mathbf{f}_p$. It is the representation of our estimate of the prediction given the choice of prior and accounting for the observed data. In the Bayesian paradigm, our estimate of the prediction is itself a random variable and therefore is characterized by a distribution representing the uncertainty in the estimate. If one could solve the parameter inference problem by sampling from the posterior, one could obtain corresponding samples of the posterior predictive by passing each sample through the prediction process. The resulting samples can then be used to calculate moments of the posterior predictive, or since the prediction dimension is very low, one can even fit a density (using kernel density estimation, e.g.) to the provided samples to visualize the complete probability density function. A block diagram of the entire process is shown in Figure 2.

The goal-oriented inference method will obtain the resulting posterior predictive probability density function without involving the posterior itself. Furthermore, the density will be obtained online in real time; that is, when the data are observed, the posterior predictive can be obtained immediately without further sampling or expensive PDE solves. In contrast, in the best-case scenario using MCMC, the data are observed and then samples are generated from the posterior, where each proposal sample requires a full PDE solve (experimental process) and each accepted sample

[1]For this work, we assume that the prediction process is noiseless. An interesting direction of future research would be to consider noisy and/or uncertain prediction processes.

requires another PDE solve (prediction process). This leads to a long delay between data observation and obtaining the posterior predictive. In most cases with high-dimensional parameter space, MCMC will be completely intractable, severing the path to prediction estimates altogether.

2.2. Joint density estimation by Gaussian mixture model. The key to goal-oriented inference for nonlinear problems in the statistical setting is to exploit the low-dimensionality of experimental data and predictions. We propose to essentially integrate out the parameter itself, instead focusing entirely on the conditional relationship between predictions and experimental data.

Consider the concatenation $\mathbf{f}(\boldsymbol{\mu}) = [\mathbf{f}_e^\top(\boldsymbol{\mu}) + \boldsymbol{\epsilon}^\top, \mathbf{f}_p^\top(\boldsymbol{\mu})]^\top : \mathbb{R}^q \to \mathbb{R}^{r+s}$ of the data and prediction models. Let $p_{Y_d,Y_p}(\mathbf{y}_d, y_p)$[2] be the joint density of data and predictions given by the push forward of the prior through $\mathbf{f}(\boldsymbol{\mu})$. Our goal is to use samples to learn the joint density $p_{Y_d,Y_p}(\mathbf{y}_d, y_p)$ in the offline phase. Once the joint density is learned, we move to the online phase where we conduct the experiment. Let $\tilde{\mathbf{y}}_d$ represent the data that are observed after the experiment is conducted, as opposed to $\mathbf{y}_d$, which represents the random variable associated to data that may be observed. When the real data $\tilde{\mathbf{y}}_d$ are observed, we can obtain the conditional density $p_{Y_p|Y_d}(y_p; \tilde{\mathbf{y}}_d)$ analytically from the learned joint density. That conditional density is precisely the posterior predictive that is the objective of our goal-oriented inference.

2.2.1. Sampling the parameter space. Similarly to many other statistical inference methods, we assume that the prior is designed so that the parameter can be efficiently sampled. Since our method is designed for, and our applications typically involve, distributed parameters, we describe the sampling procedure of the random field in this section. In particular, we describe the representation of the prior random field as a truncated Karhunen–Loeve expansion [15, 21]. For the carbon capture application undertaken in the next chapter, the permeability field is given by a lognormal random field. Our discussion here focuses on the representation of $\log \mu$.

We begin with the definition of a positive definite covariance function $C(\vec{x}_1, \vec{x}_2) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, where d is the physical dimension of the domain and $\vec{x}_1$ and $\vec{x}_2$ are two points within the domain. The covariance function describes the correlation between the value of the parameter at $\vec{x}_1$ and the value of the parameter at $\vec{x}_2$. In practice, it is usually selected from a set of well-established choices.[3] The parameters of the covariance function are selected based on the expected amplitude and correlation length in the field, typically as given by experts (in our case, geologists). We aim to generate samples from a Gaussian random field $g(\vec{x}; \xi)$ with zero mean and covariance function given by $C(\vec{x}_1, \vec{x}_2)$. The random field is given by $g(\vec{x}; \xi) = \sum_{i=1}^{\infty} \sqrt{\beta_i} \xi_i v_i(\vec{x})$, where $(\beta_i, v_i(\vec{x}))$ are the eigenpairs of the covariance function and $\xi_i \sim \mathcal{N}(0, 1)$ independently and identically distributed (iid).

In practice, the domain is discretized and our parameter is represented as piecewise constant; we will refer to the discrete approximant as $\mathbf{g}(\xi)$. As is typical practice, we will calculate discrete eigenfunctions of the covariance function by forming the Gram matrix $\mathbf{K} \in \mathbb{R}^{n_{\text{el}} \times n_{\text{el}}}$, where n_{el} is the number of elements in our computational domain. Let $\vec{x}_i$ be the centroid of element i. Then the Gram matrix is given by

[2]For the remainder of the discussion, we will assume $s = 1$; i.e., the prediction output quantity of interest is a scalar. The method is not restricted to such cases, but frequently it will be a scalar, and this makes the exposition and presentation cleaner.

[3]Multipoint statistics can also be used in our goal-oriented Bayesian inference framework.

$$\mathbf{K} = \begin{bmatrix} C(\vec{x}_1,\vec{x}_1) & C(\vec{x}_1,\vec{x}_2) & \cdots & \cdots & C(\vec{x}_1,\vec{x}_{n_{\text{el}}}) \\ C(\vec{x}_2,\vec{x}_1) & C(\vec{x}_2,\vec{x}_2) & \cdots & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \\ C(\vec{x}_{n_{\text{el}}},\vec{x}_1) & \cdots & \cdots & & C(\vec{x}_{n_{\text{el}}},\vec{x}_{n_{\text{el}}}) \end{bmatrix}. \tag{2.2}$$

Once the Gram matrix is formed, we calculate the eigenvalue decomposition $\mathbf{K} = \mathbf{V}\mathbf{H}\mathbf{V}^\top$, where $\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots \mathbf{v}_{n_{\text{el}}} \end{bmatrix}$, $\mathbf{H} = \text{diag}(\eta_1, \eta_2, \ldots, \eta_{n_{\text{el}}})$, and $\mathbf{v}_i$ and η_i are respectively the ith eigenvector and eigenvalue with the ordering $\eta_1 \geq \eta_2 \geq \cdots \geq \eta_{n_{\text{el}}} \geq 0$. The discretized random field is given by

$$\mathbf{g}(\xi) = \sum_{i=1}^{n_{\text{el}}} \xi_i \sqrt{\eta_i} \mathbf{v}_i \tag{2.3}$$

still with $\xi_i \sim \mathcal{N}(0,1)$ iid.

Typically it is not necessary to retain all the modes in the expansion. In practice we truncate the expansion after sufficient decay of the eigenvalues based on the ratio

$$\nu(m) = \frac{\sum_{i=1}^{m} \eta_i}{\sum_{i=1}^{n_{\text{el}}} \eta_i}. \tag{2.4}$$

In the application we will truncate at $m = \arg\min \nu(m)$, where $\nu(m) > 0.98$, thereby retaining 98% of the *energy* in the representation.

A sample of the parameter will therefore be generated by sampling m iid standard normal random variables ξ_i and calculating the expansion

$$\boldsymbol{\mu} = \exp\left\{ \sum_{i=1}^{m} \xi_i \sqrt{\eta_i} \mathbf{v}_i \right\} \tag{2.5}$$

so that $\boldsymbol{\mu}$ is a lognormal random vector with zero mean and covariance $\mathbf{K}$, the discrete approximant of $\mu(\vec{x})$, the infinite-dimensional lognormal random field with zero mean function and covariance function $C(\vec{x}_1, \vec{x}_2)$.

2.2.2. Gaussian mixture models. For each of the parameter samples drawn from the prior, we simulate corresponding experimental data and prediction output. Let N_s be the total number of prior samples. For each sample we evaluate the experimental process and add simulated noise to generate synthetic measurements. Analogously, for each sample, we evaluate the prediction. As a result we obtain a set of ordered pairs $(\mathbf{y}_d^i, y_p^i)$ for $i = 1, \ldots, N_s$.[4] These are samples from the joint density $p_{Y_d,Y_p}(\mathbf{y}_d, y_p)$ of experimental data and predictions.[5] From these data we propose to learn the joint density as a Gaussian mixture model (GMM). Then when

[4] As with any machine learning algorithm, one benefits greatly by perusing the data in advance, e.g., by plotting two-way marginals. Using the results, one should attempt to construct monotonic and differentiable transformations to make the data conform to the model choice (in this case, mixture of Gaussians) [11, 23]. The algorithm can then be applied to the transformed data, and the results can be transformed back appropriately. We will apply such transformations in section 4.1; here, we assume the data are already in a form amenable to our methods.

[5] It should be noted that this process is embarrassingly parallel; that is, the evaluation of experimental and prediction processes for each parameter sample can be performed completely independently. Therefore, optimal parallel scaling is possible.

data are observed, we simply condition the GMM on the given data to obtain the posterior predictive density as desired. In this section, we describe the construction of the GMM. Other density estimation techniques and other mixture models can also be used in this context. We select the GMM because of its convenient conditioning properties, allowing us to obtain real-time predictions.

A GMM [25] is a generative representation of a probability density function that generalizes the k-means clustering algorithm [27] to probabilistic clusters. Let X be a random variable distributed according to the density $p_X(x)$. Let $N(x;\mu,\Sigma)$ be the probability density function of a normal random variable with mean μ and covariance Σ. A GMM approximates

$$p_X(x) \approx \hat{p}_X(x) = \sum_{i=1}^{n_c} \alpha_i N(x;\mu_i,\Sigma_i), \qquad \sum_i \alpha_i = 1, \quad \alpha_i \geq 0, \forall i, \tag{2.6}$$

where n_c is the number of components in the mixture model. The coefficients α_i are considered prior probabilities over the n_c clusters. One can therefore think of the mixture model in a generative manner. To draw a sample from X first select the cluster by sampling from the probability mass function corresponding to α_i. Then, given the mean μ_i and covariance Σ_i of that cluster, draw a sample from the corresponding multivariate normal.

For the moment consider the number of clusters to be fixed. The estimation problem then becomes one of determining the means and covariances of each of the components in the mixture. Typically this is achieved by choosing the parameters that maximize the likelihood of the data. Let x_i for $i = 1,\ldots,N_s$ be samples of X. Then, we have

$$\alpha_i,\mu_i,\Sigma_i = \arg\max \prod_{j=1}^{N_s} \hat{p}_X(x_j). \tag{2.7}$$

The component weights and component parameters are obtained by the well-known expectation-maximization algorithm [5]. We give a brief description of the algorithm here. Let $\theta_i = \{\mu_i,\Sigma_i\}$ be the unknown parameters of component i in the mixture model. Begin with an initial setting of the unknown parameters θ_i and weights α_i. Then calculate the membership weights

$$w_{ji} = \frac{\alpha_i N(x_j;\theta_i)}{\sum_{k=1}^{n_c} \alpha_k N(x_j;\theta_k)} \qquad \forall i,\ \forall j \tag{2.8}$$

corresponding to the data point at x_j and component i. This corresponds to the E step. For the M step, we calculate the new component weights and component parameters

$$\alpha_i = \frac{1}{N_s} \sum_{j=1}^{N_s} w_{ji} \qquad \forall i, \tag{2.9}$$

$$\mu_i = \frac{1}{n_c} \sum_{j=1}^{N_s} w_{ji} x_j \qquad \forall i, \tag{2.10}$$

$$\Sigma_i = \frac{1}{n_c} \sum_{j=1}^{N_s} w_{ji} (x_j - \mu_i)(x_j - \mu_i)^\top \qquad \forall i, \tag{2.11}$$

in that order. The E and M steps are iterated until the likelihood is no longer changing from iteration to iteration, within a given tolerance. In order to select the number of components in our mixture model, we use an approach based on Bayesian information criterion [8] since it accounts explicitly for the limited number of available data in our applications.

Our choice to use a GMM representation for the joint density is motivated by the requirement for rapid conditioning of the density upon data observation and the desire to obtain a parsimonious model in the face of incomplete sampling and imperfect models for experimental process, prediction process, and sensor noise. For theoretical justification of the goal-oriented inference procedure, one could substitute a kernel density estimate (KDE) for the GMM in the representation of the joint density. In that case, as the number of samples tends to infinity, the KDE undergoes strong uniform convergence to the underlying joint density [6].

2.2.3. Evaluation of the posterior predictive online. Once the GMM is learned in the offline stage, i.e., before data are observed, we proceed to the online stage of the process. The experiments are performed and data are collected. It remains only to condition the model on the observed data to obtain the posterior predictive density. Using the GMM, this process is straightforward, as we now describe.

Let $\hat{p}_{Y_p,Y_d}(y_p, \mathbf{y}_d) = \sum_{k=1}^{n_c} \alpha_k N(y_p, \mathbf{y}_d; \mu_k, \Sigma_k)$ be the GMM we built in the offline stage. When we condition on observed data $\tilde{\mathbf{y}}_d$, we will obtain yet another GMM, this time over the prediction variables y_p only. Let $\hat{p}_{Y_p|Y_d}(y_p, y_d) = \sum_{k=1}^{n_c} \beta_k N(y_p; \tilde{\mathbf{y}}_d, \mu_{k|Y_d}, \Sigma_{k|Y_d})$ be the resulting GMM with positive component weights β_k that sum to unity, means $\mu_{k|Y_d}$, and covariances $\Sigma_{k|Y_d}$.

The new parameters are obtained as follows. Let

$$\mu_k = \begin{bmatrix} \bar{y}_{k_p} \\ \bar{\mathbf{y}}_{k_d} \end{bmatrix}, \qquad \Sigma_k = \begin{bmatrix} \Sigma_{k_{p,p}} & \Sigma_{k_{p,d}} \\ \Sigma_{k_{d,p}} & \Sigma_{k_{d,d}} \end{bmatrix} \tag{2.12}$$

be the decomposition of the component means and covariances into the parts corresponding to the prediction variables (subscript p) and data variables (subscript d). The parameters of the conditional GMM are then given by

$$\beta_k = \frac{\alpha_k (2\pi)^{-n_e/2} |\Sigma_{k_{d,d}}|^{-1/2} \exp\left\{-\frac{1}{2}(\tilde{\mathbf{y}}_d - \bar{\mathbf{y}}_{k_d})^\top \Sigma_{k_{d,d}}^{-1} (\tilde{\mathbf{y}}_d - \bar{\mathbf{y}}_{k_d})\right\}}{\sum_{m=1}^{n_c} \alpha_m (2\pi)^{-n_e/2} |\Sigma_{m_{d,d}}|^{-1/2} \exp\left\{-\frac{1}{2}(\tilde{\mathbf{y}}_d - \bar{\mathbf{y}}_{m_d})^\top \Sigma_{m_{d,d}}^{-1} (\tilde{\mathbf{y}}_d - \bar{\mathbf{y}}_{m_d})\right\}},$$

$$\mu_{k|Y_d} = \bar{y}_{k_p} + \Sigma_{k_{p,d}} \Sigma_{k_{d,d}}^{-1} (\tilde{\mathbf{y}}_d - \bar{\mathbf{y}}_{k_d}),$$

$$\Sigma_{k|Y_d} = \Sigma_{k_{p,p}} - \Sigma_{k_{p,d}} \Sigma_{k_{d,d}}^{-1} \Sigma_{k_{d,p}}.$$

The conditional density can then be visualized, samples can be drawn, or moments can be computed. This method has been validated on simple examples where MCMC is tractable [17].

3. Application to carbon capture and storage. In this section we apply the nonlinear goal-oriented inference approach described in the previous section to a realistic application in carbon capture and storage. Sequestering carbon emissions in the subsurface is one method for curbing anthropogenic effects. A model for accurately predicting plume migration requires knowledge of the subsurface parameters (e.g., permeability and porosity). The subsurface parameters are field quantities and can be measured only indirectly by making sparse readings of pressures at boreholes for some experimental conditions. Since our focus will be on predicting trapped volume

(and not resolving plume migration), we utilize the goal-oriented inference approach to establish prediction estimates directly from observed data from the experiment.

3.1. Application description and numerical implementation. Supercritical carbon dioxide is typically injected in saline aquifers in the subsurface. The fluid is buoyant with respect to the resident brine; therefore, it floats and migrates along the caprock of the aquifer. Where the aquifer geometry is suitable, the fluid can be captured in pockets underneath the caprock. Remaining carbon dioxide continues to migrate. Of primary importance in such scenarios is the percentage of carbon dioxide effectively trapped in the injection and migration process over a given period of time. The dynamics of the plume depend heavily on the permeability in the aquifer, the target of parameter inference in the application. Determining the feasibility for injection of a candidate aquifer would typically involve performing experiments in the form of hydraulic interference tests to infer the permeability field. The estimate can then be used as input to an injection simulation to predict trapped volume percentage to evaluate different injection scenarios and ultimately to make decisions.

The computational tasks involving the geometry, numerical solution, and visualization of the experiment and prediction processes for the carbon capture and storage application are performed using SINTEF's MATLAB reservoir simulation toolbox (MRST) [16]. The governing equations for the experiment and prediction processes are discretized using the mimetic finite difference method. For the experiment process we solve the single-phase flow equations in three dimensions. On the other hand, for the two-phase flow governing the migration of the carbon dioxide plume, we use the vertical equilibrium module that ships with MRST. MRST can be found at `www.sintef.no/Projectweb/MRST/` and is freely available under the GNU General Public License, is well maintained, and is updated with each new distribution of MATLAB.

3.2. Computational domain. The computational domain is a hexahedral discretization of a synthetic, but realistic, saline aquifer. The domain is pictured in Figure 3. The aquifer occupies a 1 kilometer by 1 kilometer ground area and varies in thickness from approximately 30 m to 80 m as a function of x and y location. The aquifer's top surface contains both high- and low-frequency variations, and the aquifer itself has a 20 m fault. We have made an effort to include the most challenging aspects of realistic candidate aquifers. The domain has 30,000 cells.

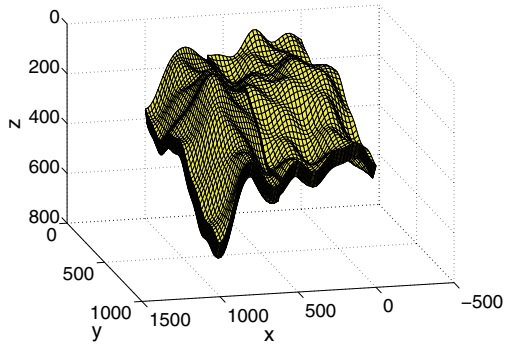

FIG. 3. *The computational domain representing the candidate aquifer.*

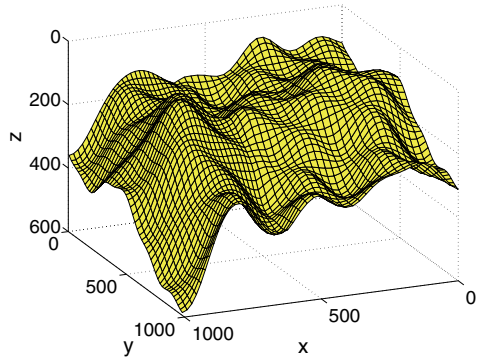

FIG. 4. *The computational domain for the prediction process; the top grid of the three-dimensional computational domain.*

We use the full three-dimensional domain for the experiment process where we solve the single-phase flow equations under given injection and production scenario. For the prediction process, however, we will enlist the vertical equilibrium approximation and use only the top grid, which contains 2500 quadrilateral cells. The top surface grid is pictured in Figure 4.

3.3. Permeability field. The parameter in this goal-oriented inference problem is the permeability field in the aquifer. Let $\underline{\mu}(x, y, z) : \mathbb{R}^3 \to \mathbb{R}^{3\times 3}$ be the permeability tensor field. We will assume that the tensor is anisotropic and has the form

$$\underline{\mu}(x, y, z) = \mu(x, y, z) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}, \tag{3.1}$$

where $\mu(x, y, z)$ is the parameter field.

We model the permeability as a lognormal random field with $\mu(x, y, z; \xi) = \exp g(x, y, z; \xi)$, where $g(x, y, z; \xi)$ is a Gaussian random field. We specify zero mean function and covariance function

$$C(\vec{x}_1, \vec{x}_2) = b \exp\left\{\frac{1}{L}\|\vec{x}_1 - \vec{x}_2\|\right\}, \tag{3.2}$$

where $\vec{x} = (x, y, z)$, b is the amplitude, and L is the correlation length scale. In this application we use $L = 400$ and $b = 5$, which results in samples of permeability that vary by four to five orders of magnitude. This exponential kernel has algebraically diminishing eigenvalues (compared to the exponentially decreasing eigenvalues of the squared exponential covariance kernel, e.g.), which makes the moral dimension of the parameter space still very large.

As mentioned in the previous section, we discretize the random field and represent it as piecewise constant with the permeability varying from cell to cell. To sample approximately from the random field $g(x, y, z; \xi)$ we first construct the Gram matrix $\mathbf{K}$ by evaluating the covariance function at all pairs of centroids of the cells. We then perform the eigenvalue decomposition of the resulting matrix and retain modes corresponding to the highest 98% of the total energy as determined by the eigenvalues. The eigenvalue decay is shown in Figure 5. The first eight eigenvectors are pictured in Figure 6. Eight random samples of the log permeability are shown in Figure 7.

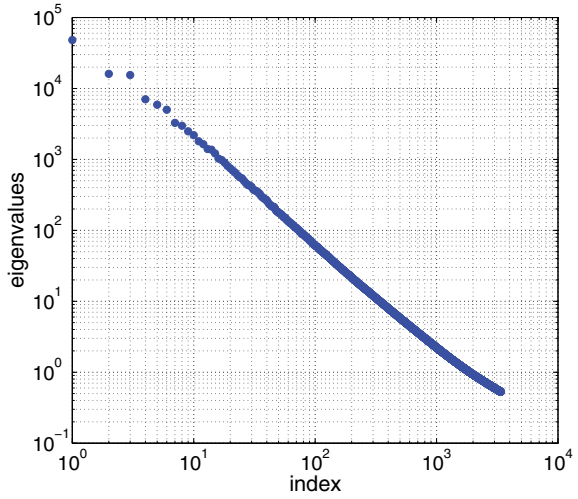

FIG. 5. *Eigenvalues of the Gram matrix with exponential covariance kernel. (Only the retained* 3383 *eigenvalues are pictured.)*

3.4. Experiment process. We now define the experiment process for the carbon capture and storage application. The experiment process is the steady state single-phase flow in the aquifer under given injection and production rates at five injection wells controlled by bottomhole pressure and three production wells controlled by rate. The outputs are the bottomhole pressures at each of the production wells.

Let Ω be the computational domain with boundary faces $\delta\Omega$. We assume no flow boundary conditions (i.e., homogeneous Neumann at each boundary face). The pressure is fixed to 300 bar at the bottom of each of the injection wells. The production wells extract fluid at a rate of 3 m^3/day. The governing equation in the domain outside of the wells (which is solved using a Peaceman well model) is given by conservation of mass and Darcy flow, i.e.,

$$-\nabla \cdot (\mu \nabla u) = q, \qquad \vec{x} \in \Omega, \tag{3.3}$$

where u is the global pressure and q corresponds to the sources/sinks at the injection and production wells.

For each sample of the permeability field, we solve (3.3) using MRST and extract the bottomhole pressure at the production wells; we have $\mathbf{f}_e = [u(\vec{x}_1; \boldsymbol{\mu}), u(\vec{x}_2; \boldsymbol{\mu}), u(\vec{x}_3; \boldsymbol{\mu})]^\top$, where the production wells extend down to the cell whose centroids are at $\vec{x}_1$, $\vec{x}_2$, and $\vec{x}_3$, respectively. An example solution showing the injection and production wells is pictured in Figure 8. The locations of the injection and production wells are given in Table 1.

We simulate noise in the data by sampling from the additive Gaussian noise model with zero mean and standard deviation $\sigma_n = 2$ bar. Histograms of the marginal likelihood of the data for each experimental output are shown in Figure 9.

3.5. Prediction process. The prediction process is given by the two-phase flow of supercritical carbon dioxide and the resident brine in the aquifer. We make the common assumption that the flow is incompressible and that the two phases are immiscible. Furthermore, we will neglect capillary pressure in the model. For the development of the governing equations and the vertical equilibrium approximation,

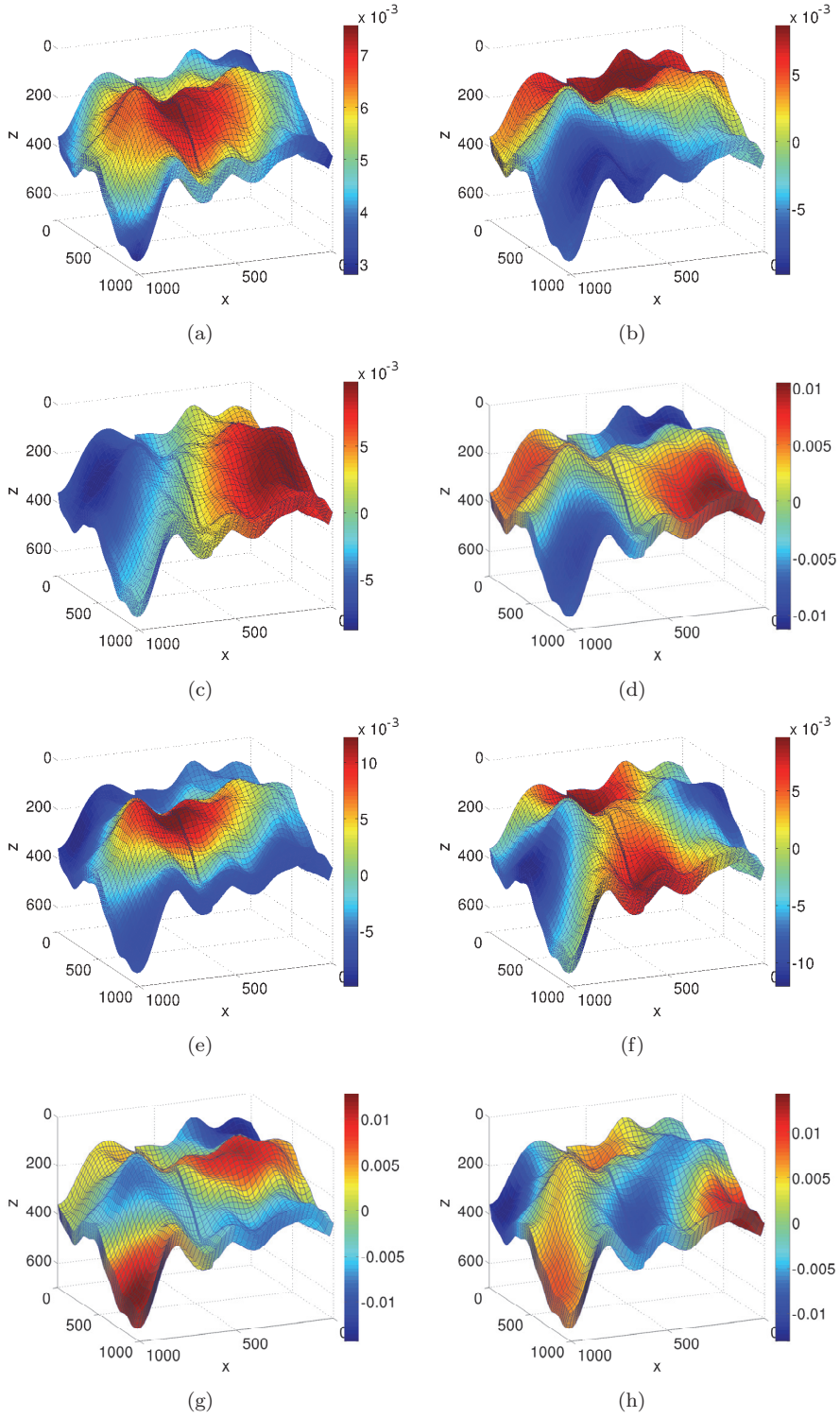

FIG. 6. *The first eight eigenmodes of the Gram matrix. These are the first eight components (without weighting) of the Karhunen–Loeve expansion used to represent the log permeability. The Gram matrix is obtained by evaluating the covariance for the centroids of all cells in the domain.*

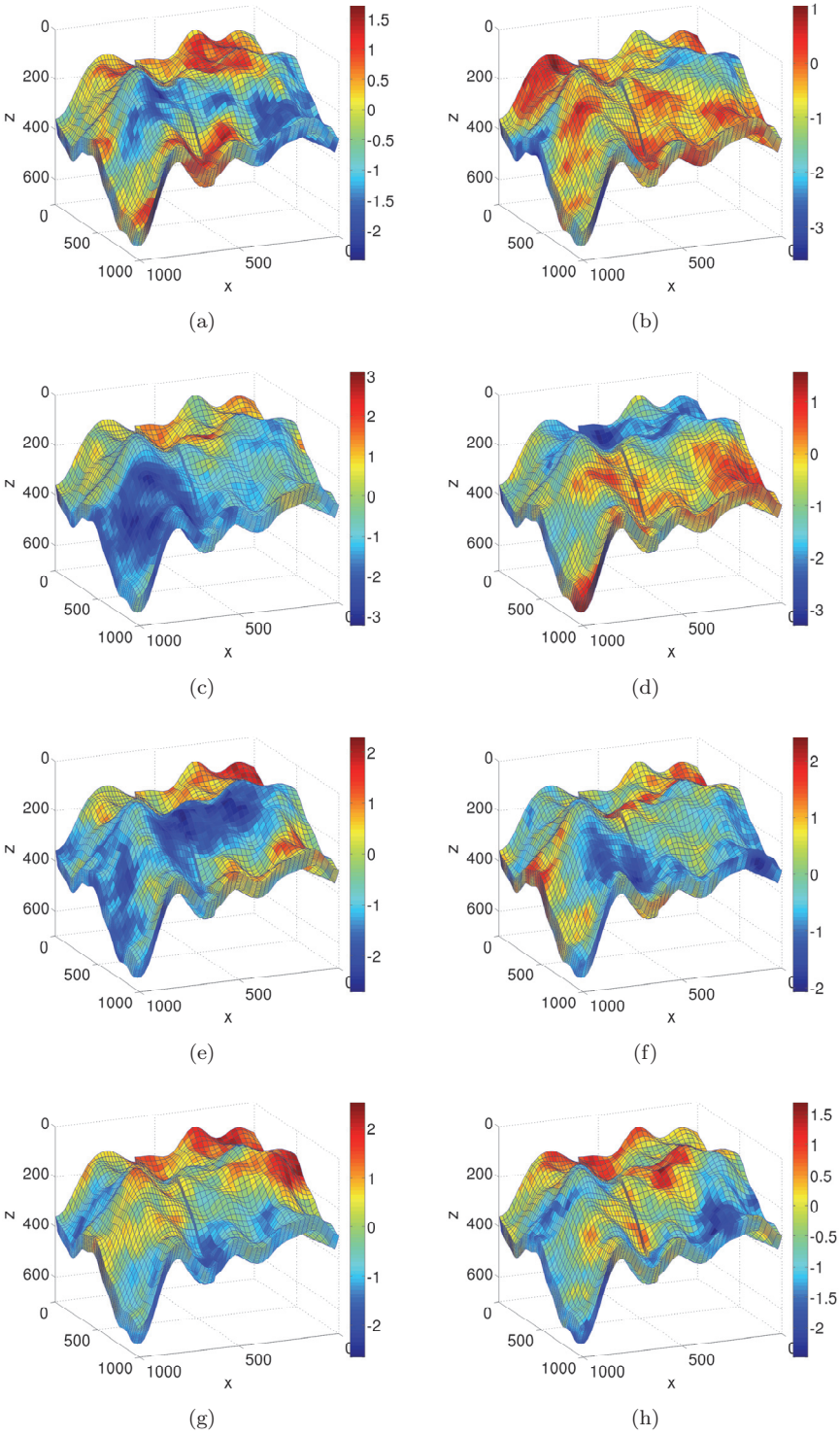

FIG. 7. *Eight samples of the log permeability. The lognormal random process used as the prior on the parameter results in large deviations in permeability from point to point throughout the domain, increasing the difficulty of the inference problem.*

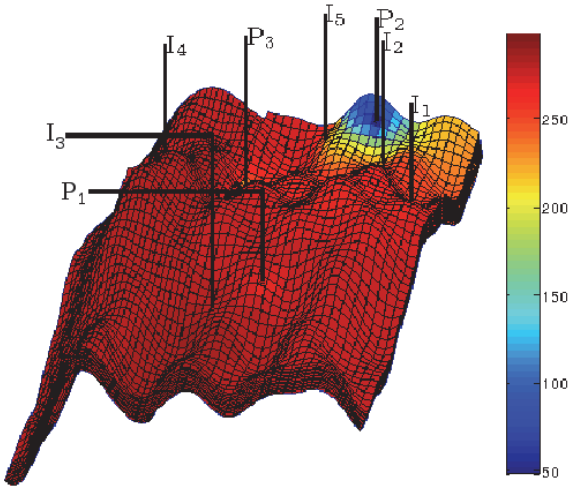

FIG. 8. *An example solution of the pressure (bar) in the experiment process. Five injection and three production wells are pictured.*

TABLE 1
The positions and completions of the injection and production wells for the experimental process.

Label	Completion top (x, y, z)	Completion bottom (x, y, z)
I1	(62.352, 410.000, 135.592)	(63.178, 410.000, 150.430)
I2	(224.8412, 150.000, 194.957)	(223.716, 150.000, 209.392)
I3	(525.123, 729.999, 306.420)	(528.343, 729.998, 322.413)
I4	(784.941, 330.000, 164.954)	(785.166, 330.000, 180.140)
I5	(415.082, 70.001, 205.000)	(413.352, 70.002, 218.782)
P1	(396.613, 689.999, 201.606)	(399.512, 689.999, 215.890)
P2	(275.624, 50.001, 108.887)	(273.756, 50.001, 124.071)
P3	(587.946, 230.000, 225.926)	(587.421, 230.000, 241.260)

we follow [20]. For a more in-depth reference on the vertical equilibrium approximation and other modeling approaches, see [24].

Let φ be the porosity (assumed constant) in the aquifer, p_n and p_w be the pressures of the carbon dioxide (nonwetting) and brine (wetting), $\mathbf{v}_n$ and $\mathbf{v}_w$ be the corresponding velocities, and S_n and S_w be the corresponding saturations. Mass conservation is then expressed by the PDEs

$$\varphi \frac{\partial S_i}{\partial t} + \nabla \cdot \mathbf{v}_i = q_i, \qquad \mathbf{v}_i = -\lambda_i(S)\mu(\nabla p_i - \rho_i \mathbf{g}), \quad i = n, w, \tag{3.4}$$

where ρ_i is the phase density, q_i is the phase source volume rate, $\lambda_i(S)$ is the phase mobility as a function of the saturation, and $\mathbf{g}$ is the gravitational vector. Define now a global pressure p and total velocity $\mathbf{v}$ to obtain

$$\varphi \frac{\partial S}{\partial t} + \nabla \cdot f(S)(\mathbf{v} + \lambda_w(S)\mu\Delta_\rho \mathbf{g}) = q_n, \tag{3.5}$$

$$\mathbf{v} = -\mu\lambda_t(S)(\nabla p - (f(S)\rho_n + (1 - f(S))\rho_w)\mathbf{g}), \tag{3.6}$$

$$\nabla \cdot \mathbf{v} = q_t, \tag{3.7}$$

where $\lambda_t = \lambda_n + \lambda_w$ is the total mobility, $f = \lambda_n/\lambda_t$ is the fractional mobility, $\Delta_\rho = \rho_n - \rho_w$ is the density difference of the two phases, and $q_t = q_n + q_w$ is the total volume rate of source.

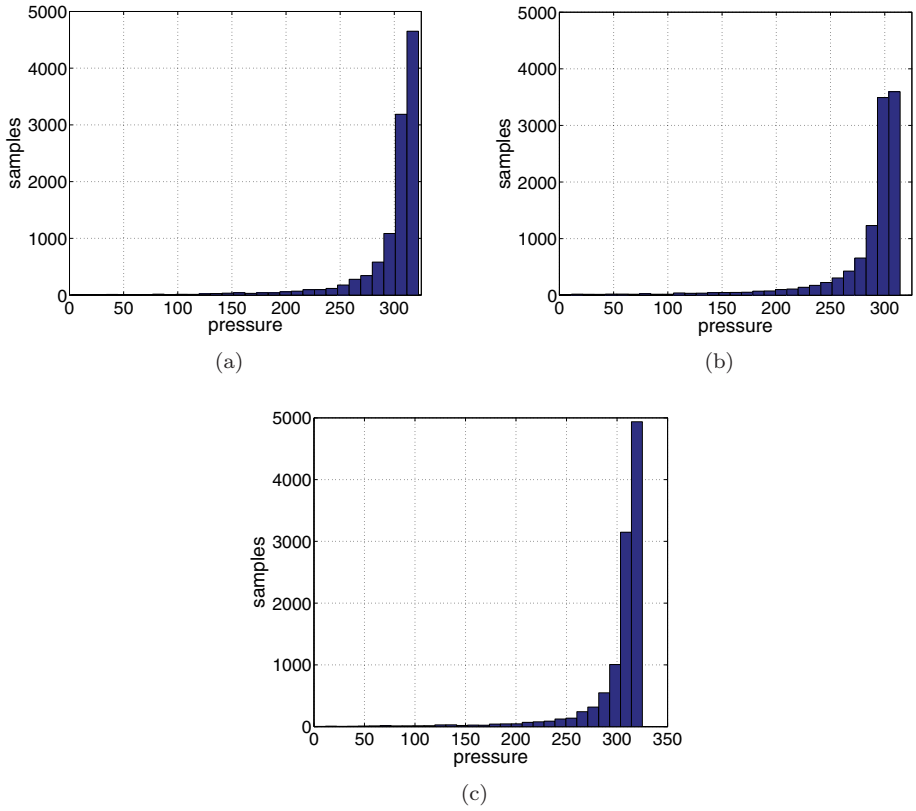

FIG. 9. *For* 11,075 *samples of the parameter, marginal likelihood of the data for the three experimental outputs, bottomhole pressure at production wells* (a) $P1$, (b) $P2$, *and* (c) $P3$.

Let H be the total height of the aquifer and h be the height of the carbon dioxide plume. Define $s = h/H$ to be the relative height of the carbon dioxide plume as a function of position $\vec{x} = (x, y)$ and time t. If we vertically average (3.5), we obtain

$$\begin{aligned}
&\varphi H(\vec{x})\frac{\partial s}{\partial t} + \nabla_{||} \cdot \left(\tilde{f}(s,\vec{x})\mathbf{v}_{ve} + \tilde{f}_g(s,\vec{x})(\mathbf{g}_{||}(\vec{x}) + \nabla p_c(s,\vec{x})\right) = q_n(x,y),\\
&\nabla_{||} \cdot \mathbf{v}_{ve} = q_t(\vec{x}),\\
&\mathbf{v}_{ve} = -\tilde{\lambda}_t(s,\vec{x})\left(\nabla_{||}p_t - (\tilde{f}(s,\vec{x})\rho_n + (1-\tilde{f}(s,\vec{x}))\rho_w)\mathbf{g}_{||}(\vec{x}) + \frac{\tilde{\lambda}_w}{\tilde{\lambda}_t}\nabla_{||}p_c(s,\vec{x})\right),
\end{aligned}$$

where the notation $\mathbf{a}_{||}$ indicates the component of the vector $\mathbf{a}$ parallel to the top surface of the aquifer, and $p_t(\vec{x})$ is the global pressure at the top surface. Since we disregard capillary forces, we have $p_c(s,\vec{x}) = H(\vec{x})g_\perp \Delta_\rho s$, where $g_\perp$ is the component of the gravity vector perpendicular to the top surface.

Heterogeneities in the medium are preserved in the vertical equilibrium approximation by defining modified mobilities and fractional flow functions

$$\tilde{\lambda}_n = \int_0^{sH(\vec{x})} \frac{k_n}{\nu_n}\mu dz, \quad \tilde{\lambda}_w = \int_{sH(\vec{x})}^{H(\vec{x})} \frac{k_w}{\nu_w}\mu dz, \quad \tilde{f} = \frac{\tilde{\lambda}_n}{\tilde{\lambda}_n + \tilde{\lambda}_w}, \quad \tilde{f}_g = \tilde{\lambda}_w \tilde{f}, \tag{3.8}$$

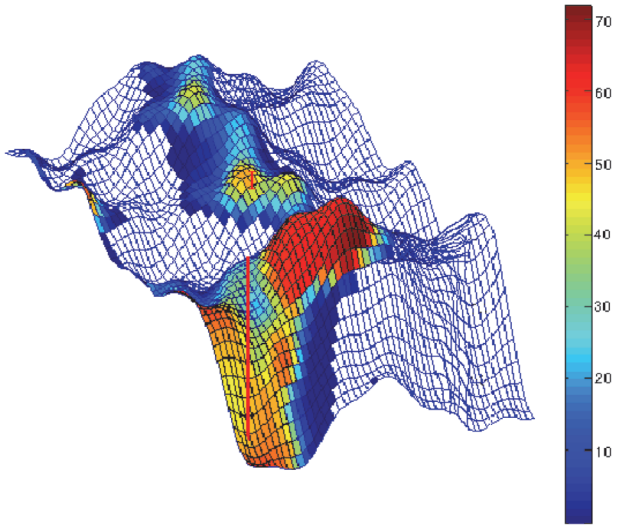

FIG. 10. *The height (m) of the carbon dioxide plume after* 50 *years of injection at* 500 m^3*/day followed by* 450 *years of migration.*

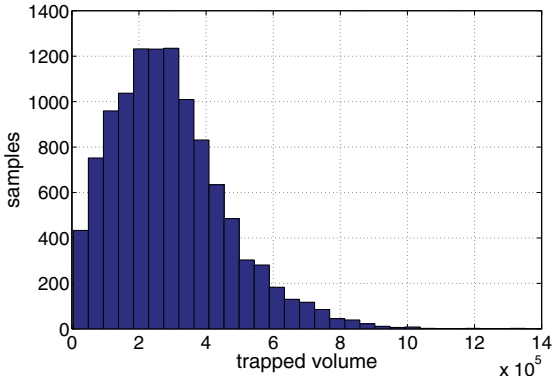

FIG. 11. *Marginal likelihood of the prediction of trapped volume for* 11,075 *samples of the parameter.*

where k_i and ν_i are the relative permeability and viscosity, respectively, of phase i. The evaluation of the relative permeabilities in (3.8) depends on whether the reservoir is locally undergoing drainage or imbibition. Let s_i^{res} be the residual saturation of phase i. When the aquifer is undergoing imbibition, i.e., $s_{max} > s$, where s_{max} is the maximum historical saturation, then the relative permeabilities are evaluated at $1 - s_w^{res}$ for k_n and $1 - s_n^{res}$ for k_w; otherwise, they are evaluated at unit saturation.

We simulate the injection of supercritical carbon dioxide at one well at a rate of 500 m^3/day for 50 years followed by the migration that takes place over the following 450 years. The resulting plume is pictured in Figure 10. The prediction quantity of interest is the total volume of trapped carbon dioxide. This corresponds to the portion of the fluid that has been successfully sequestered under the caprock and is no longer mobile, thereby no longer presenting a risk of leakage through faults or improperly sealed wells in other parts of the subsurface. A histogram of the marginal likelihood of the predictions is shown in Figure 11.

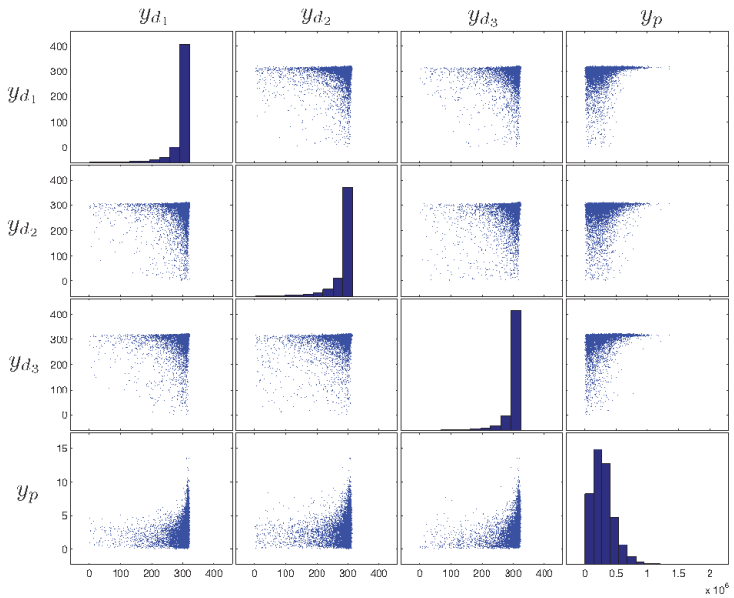

FIG. 12. *Pairwise marginal data and histograms. The first three rows (columns) correspond to the experimental data, and the last row (column) corresponds to the prediction output. It is clear that all components would benefit from certain logarithmic transformations.*

4. Numerical results and discussion. We now undertake the task of learning the joint density of experimental data and prediction. We then perform numerical experiments with synthetic data to demonstrate our approach.

4.1. Transforming the data. We begin by inspecting the data further, a recommended first step before employing any machine learning algorithm. Any additional insight one can gain from perusing the data can help to inform model building or algorithmic choices. The raw data are shown in Figure 12 in the form of pairwise marginals.

From the figure, it is clear that each component of the experimental data and the prediction would benefit from logarithmic transformation. Therefore, we perform the transformations

$$\mathbf{y}_d \leftarrow \ln(330 - \mathbf{y}_d), \qquad y_p \leftarrow \ln y_p, \tag{4.1}$$

which are both differentiable and monotonic. The barrier value at 330 bar was determined based upon inspection of the raw data. The resulting histograms are shown in Figure 13.

4.2. Numerical results. Using the transformed data in Figure 13, we learn a GMM for the joint density [7, 8]. We select automatically the number of components to balance the maximization of likelihood of data and the Bayesian information criterion [4]. Given the 11,075 samples of the joint density we use, the algorithm settles on a GMM for the joint density that has 15 components. As a verification tool, we use a KDE to compare our results. The KDE is obtained using the kernel density estimation toolbox for MATLAB [13].

For the numerical experiments, we select a permeability field at random from the prior distribution. We assume this random sample to be truth, i.e., the true

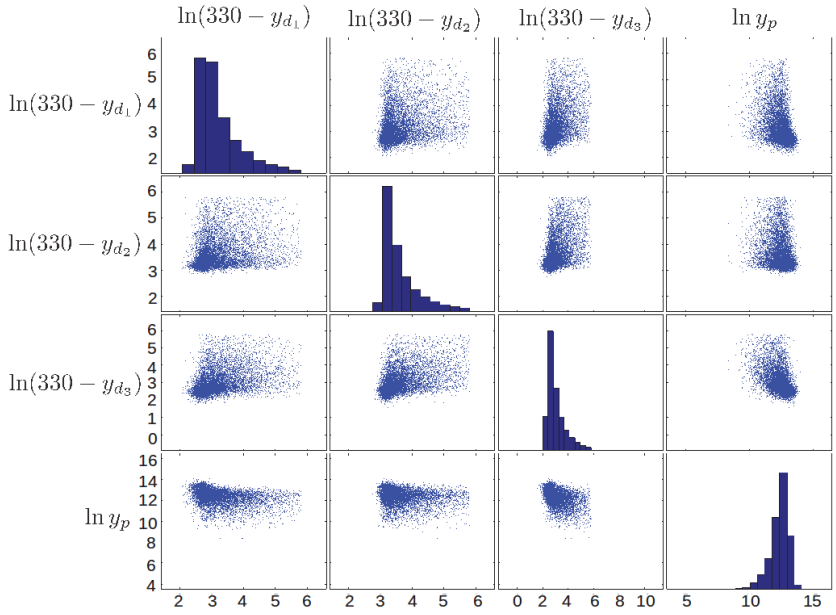

FIG. 13. *Pairwise marginal data and histograms after the transformations of* (4.1) *have been applied.*

TABLE 2

For the six assumed truth samples of the permeability field, we give the figure where the posterior predictive is plotted, the raw observed data, and the true prediction.

Figure	y_{d_1} (bar)	y_{d_2} (bar)	y_{d_3} (bar)	y_p (m^3)
14(a)	282.124	294.804	293.153	1.294×10^5
14(b)	294.865	298.440	295.787	3.416×10^5
14(c)	298.654	294.842	279.562	2.795×10^5
14(d)	291.663	288.973	293.773	1.994×10^5
14(e)	278.557	296.358	298.237	3.764×10^5
14(f)	298.061	273.317	293.704	2.948×10^5

permeability field in the subsurface. Experimental data are simulated by solving the single-phase pressure equation given the injection and production settings specified in section 3.4. Given the simulated noisy data, we carry out the online process of goal-oriented inference: the previously learned joint density is conditioned at the observed data to obtain the posterior predictive density. The raw observed data and predictions based on the truth sample of permeability are given in Table 2.

Figure 14 presents the posterior predictive by both GMM and KDE for some truth samples of the permeability field. For reference, we also show the prior predictive density as obtained by a KDE over all samples of prediction from the offline phase. For each case, we also plot the prediction quantity of interest obtained by simulating the prediction process for the given truth sample. Note that this is a deterministic quantity; however, we expect it to appear within the significant support of the posterior predictive with high probability.

The KDE generally produces results with greater total variation since it is more sensitive to the locality of samples of the joint density near where data are observed. On the other hand, the GMM tends to smooth the result since it involves only

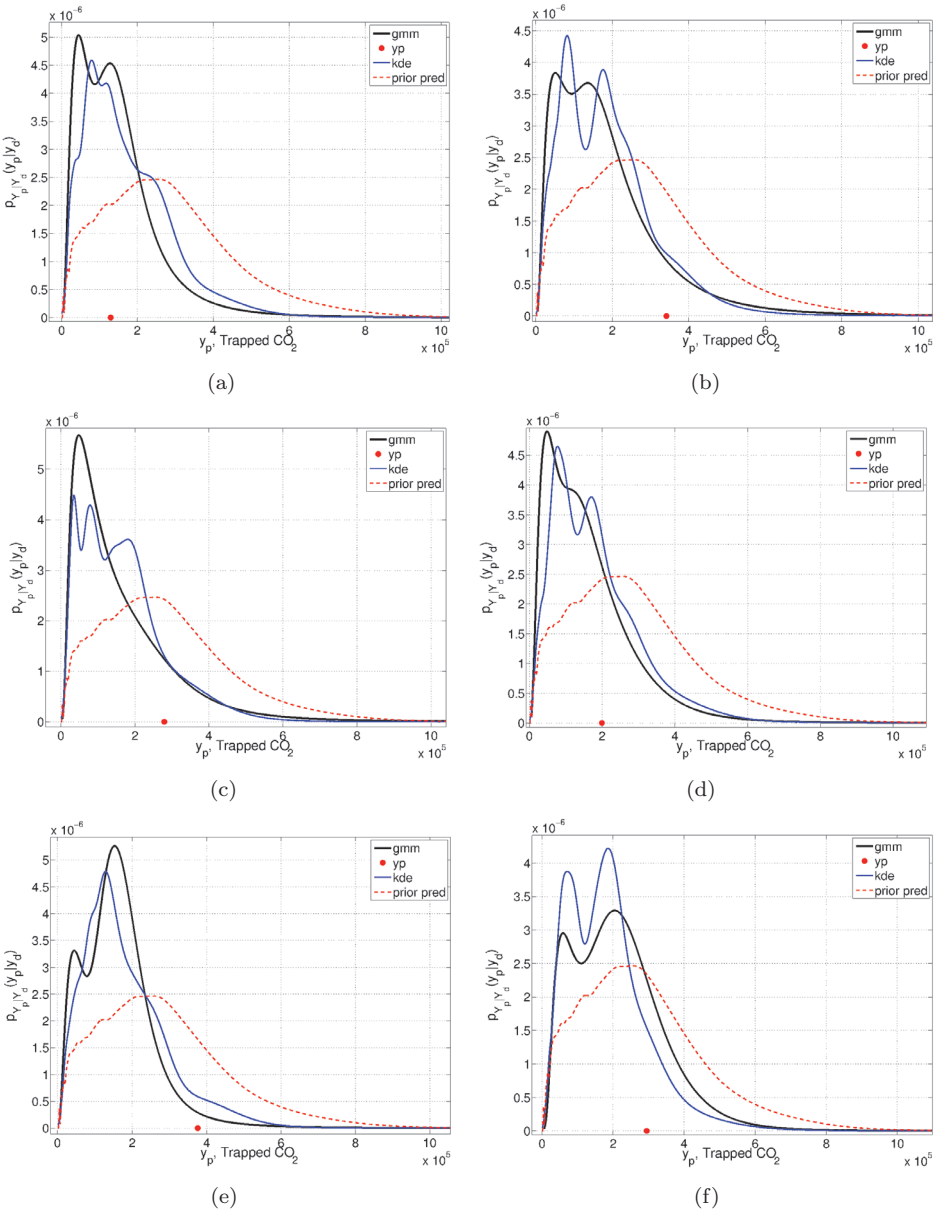

FIG. 14. *Posterior predictive densities for samples of the permeability field. Results from the GMM and KDE are both presented. The prior predictive density and true prediction are given as reference. Observed data values are given in Table* 2.

15 Gaussian components, each with greater kernel width than the kernel from KDE, which has as many components as original samples. In all cases, the GMM and KDE are in general agreement, particularly in the manner with which the posterior predictive density differs from the prior predictive density. It is this update in information based on the observed data which is critical to our goal-oriented inference.

The posterior predictive densities represent our updated belief in the relative likelihood of trapped volume of carbon dioxide under the pumping scenario described

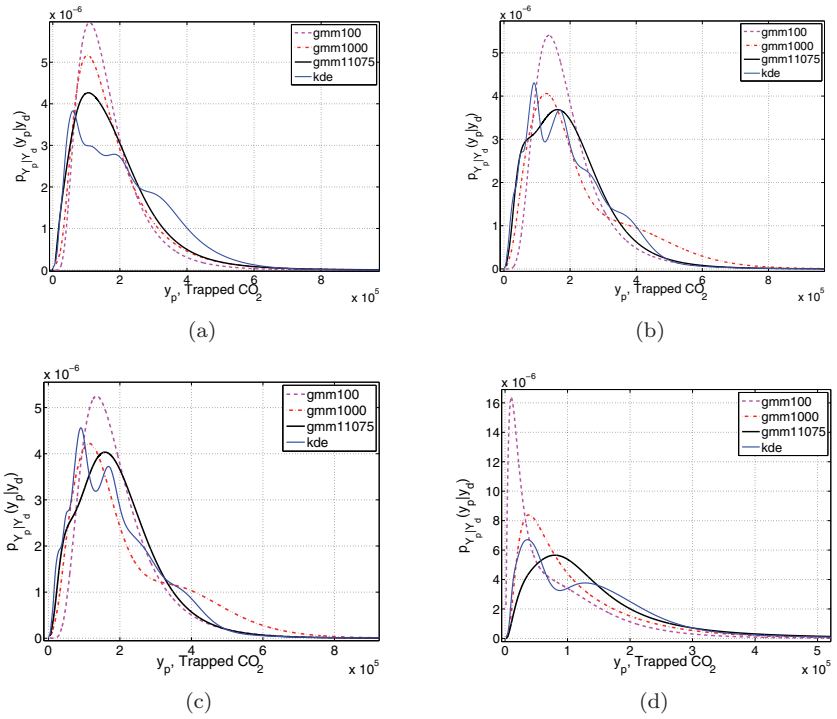

FIG. 15. *Posterior predictive densities for four samples of the permeability field representing the true permeability synthetically. Results from GMMs learned using* 100 *samples,* 1000 *samples, and* 11,075 *samples offline are compared to solutions obtained using KDE.*

in section 3.5 given our prior specification on the parameter and the observed data from experiments. We have a full probabilistic description of the trapped volume enabled by focusing on the relationship between experimental data and the prediction quantity of interest, all in an application where best practices in statistical inference of the parameter would have been insufficient. In practice these results would feed forward to a decision-making process where it would be determined if sufficient volume of the carbon dioxide would be trapped in the aquifer to proceed with the injection.

4.3. Effect of number of offline samples. In this section we investigate numerically the effect of the number of offline samples of the parameter on the resulting prediction accuracy. We learn separate GMMs using 100 samples, 1000 samples, and all 11,075 samples and compare the posterior predictive results against those obtained by KDE for several realizations of the data. The results are shown in Figure 15.

The trend is expected. The approximation of the posterior predictive density by GMM appears to improve with more offline samples. Since we assume that the prior is efficient to sample and we build the GMM before experiments are performed, we recommend that as much data are acquired as possible to build the GMM. If data are difficult or expensive to obtain for some reason, it should be noted that the accuracy of the GMM (as well as many other estimation procedures) will be affected significantly.

In order to study the dependency of the accuracy of the approach on proximity of the observed data point to the offline samples, we explore three cases where we artificially prescribe data. In the first case, we set the data to be in a high-density region of the samples for all the mixture models we trained with different numbers of

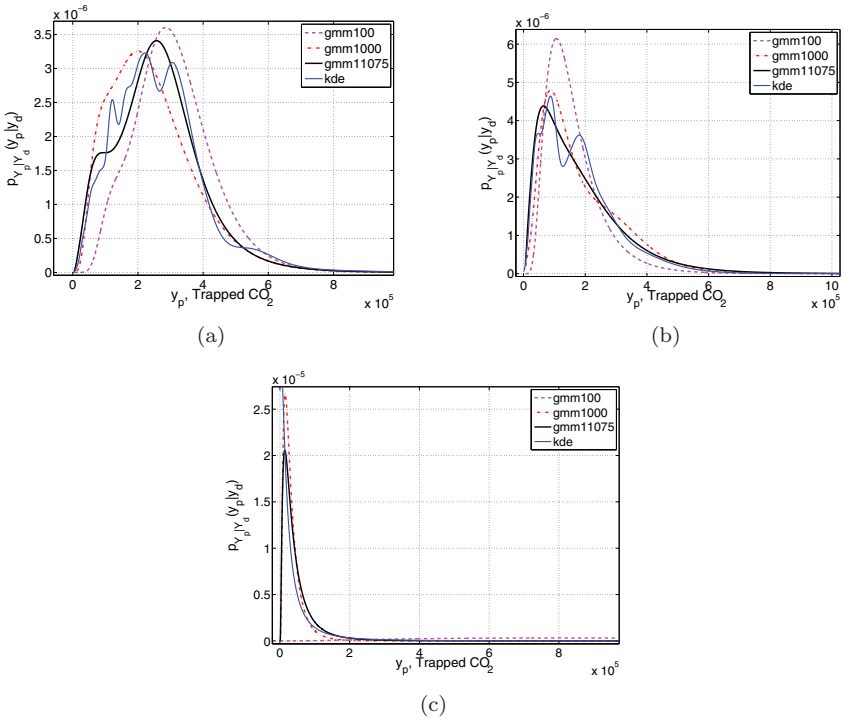

FIG. 16. *Posterior predictive densities for three artificial realizations of data:* (a) *proximal to points used to train all GMMs,* (b) *proximal to points for many-sample GMMs but in a low-density region of the* 100*-sample GMM, and* (c) *in a low-density region of all GMMs. Results from GMMs learned using* 100 *samples,* 1000 *samples, and* 11,075 *samples offline are compared to solutions obtained using KDE.*

offline samples. For the second case, we select the data artificially to be proximal to many samples from the 1000-sample and 11,075-sample GMMs but in a low-density region of the 100-sample GMM. Finally, in the last case, we select the data artificially to be in a low-density region of all the GMMs. The results are shown in Figure 16.

For the first case (see Figure 16(a)), there is general agreement of all the GMMs with the KDE posterior predictive density. This is expected behavior since the model should be accurate in high sample density regions. In the second case (see Figure 16(b)), the GMM based on 100 samples is inaccurate since the observed data occur in a low sample density region. Finally, in the last case (see Figure 16(c)), all the posterior predictive density approximations fail. The observed data are well outside the offline samples. This reflects that these density estimations are susceptible to large errors when extrapolation is required. For this reason, it is critical that one attempt to detect such situations before trusting the resulting posterior predictive density.

5. Conclusion. We have developed a practical method for goal-oriented inference involving a priori parameter sampling and learning the joint density of predictions and experimental data. The method exploits the low-dimensionality of the product space of experimental data and predictions. The accuracy of the method depends on the ability to sample the parameter space and simulate experimental data and predictions; however, this can be completed entirely offline before the real data are

observed, and the process is embarrassingly parallel, meaning that one could obtain optimal parallel scaling on a distributed computing architecture. Once the joint density is learned, the experiment is performed, and the density is conditioned on the observed data to obtain the posterior predictive density in real time. Since we focus on prediction quantities of interest, we circumvent the issues associated with inferring the parameter itself; instead, we focus on the relationship between experimental data and prediction quantities of interest. The method enables real-time predictions for carbon capture and storage, a problem for which state-of-the-art MCMC approaches are intractable.

REFERENCES

[1] J. Berger, *The case for objective Bayesian analysis*, Bayesian Anal., 1 (2006), pp. 385–402.

[2] J. Besag, P. Green, D. Higdon, and K. Mengerson, *Bayesian computation and stochastic systems*, Statist. Sci., 10 (1995), pp. 3–41.

[3] G. Casella, *Objective Bayesian variable selection*, J. Amer. Statist. Assoc., 101 (2012), pp. 157–167.

[4] S. S. Chen and P. S. Gopalakrishnan, *Clustering via the Bayesian information criterion with applications in speech recognition*, in Proceedings of the 1998 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 2, 1998, pp. 645–648.

[5] A. Dempster, N. Laird, and D. Rubin, *Maximum likelihood from incomplete data via the EM algorithm*, J. Roy. Statist. Soc., 39 (1977), pp. 1–38.

[6] L. Devroye and C. S. Penrod, *The strong uniform convergence of multivariate variable kernel estimates*, Canad. J. Statist., 14 (1986), pp. 211–219.

[7] M. Figueiredo and A. Jain, *Gaussian Mixture Models for MATLAB*, http://www.lx.it.pt/∼mtf/ (2002).

[8] M. Figueiredo and A. Jain, *Unsupervised learning of finite mixture models*, IEEE Trans. Pattern Anal. Machine Intell., 24 (2002), pp. 381–396.

[9] A. Gelman, *Objections to Bayesian statistics*, Bayesian Anal., 3 (2008), pp. 445–449.

[10] W. Gilks, S. Richardson, and D. Spiegelhalter, *Markov Chain Monte Carlo in Practice*, Chapman & Hall, London, 1996.

[11] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer-Verlag, Berlin, 2008.

[12] C. Huang, T. Hsing, N. Cressie, A. Ganguly, A. Protopopescu, and N. Rao, *Bayesian source detection and parameter estimation of a plume model based on sensor network measurements*, Appl. Stoch. Models Bus. Ind., 26 (2006), pp. 331–348.

[13] A. Ihler and M. Mandel, *Kernel Density Estimation Toolbox for MATLAB*, http://www.ics.uci.edu/∼ihler/code/kde.html (2003).

[14] J. Kaipio and E. Somersalo, *Statistical and Computational Inverse Problems*, Springer, Berlin, 2005.

[15] K. Karhunen, *Uber lineare methoden in der wahrscheinlichkeitsrechnung*, Ann. Acad. Sci. Fennicae. Seri. A. I. Math.-Phys., 37 (1947), pp. 1–79.

[16] K.-A. Lie, S. Krogstad, I. S. Ligaarden, J. R. Natvig, H. M. Nilsen, and B. Skaflestad, *Open source MATLAB implementation of consistent discretisations on complex grids*, Comput. Geosci., 16 (2012), pp. 297–322.

[17] C. E. Lieberman, *Goal-Oriented Inference: Theoretical Foundations and Application to Carbon Capture and Storage*, Ph.D. thesis, Massachusetts Institute of Technology, 2013.

[18] C. Lieberman and K. Willcox, *Goal-oriented inference: Approach, linear theory, and application to advection-diffusion*, SIAM J. Sci. Comput., 34 (2012), pp. A1880–A1904.

[19] C. Lieberman, K. Willcox, and O. Ghattas, *Parameter and state model reduction for large-scale statistical inverse problems*, SIAM J. Sci. Comput., 32 (2010), pp. 2523–2542.

[20] I. Ligaarden and H. M. Nilsen, *Numerical aspects of using vertical equilibrium models for simulating CO_2 sequestration*, in Proceedings of ECMOR XII—12th European Conference on the Mathematics of Oil Recovery, Oxford, UK, 1995.

[21] M. Loeve, *Probability Theory*, Springer-Verlag, Berlin, 1978.

[22] Y. Marzouk, H. Najm, and L. Rahn, *Stochastic spectral methods for efficient Bayesian solution of inverse problems*, J. Comput. Phys., 224 (2007), pp. 560–586.

[23] T. A. El Moselhy and Y. M. Marzouk, *Bayesian inference with optimal maps*, J. Comput. Phys., 231 (2012), pp. 7815–7850.

[24] J. M. Nordbotten and M. A. Celia, *Geological Storage of CO2: Modeling Approaches for Large-Scale Simulation*, John Wiley & Sons, New York, 2011.
[25] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, Cambridge, UK, 2007.
[26] C. R. Smith and W. T. Grandy, Jr., *Maximum Entropy and Bayesian Methods in Inverse Problems*, Springer, Berlin, 1985.
[27] H. Steinhaus, *Sur la division des corps materiels en parties*, Bull. Acad. Polon. Sci., 4 (1957), pp. 801–804.
[28] A. M. Stuart, *Inverse problems: A Bayesian perspective*, Acta Numer., 19 (2010), pp. 451–559.
[29] L. Tierney, *Markov chains for exploring posterior distributions*, Ann. Statist., 22 (1994), pp. 1701–1728.
[30] G. Wang and S. Chen, *Evaluation of a soil greenhouse gas emission model based on Bayesian inference and MCMC: Parameter identifiability and equifinality*, Ecological Modelling, 253 (2013), pp. 107–116.