


## Online Updating of Computer Model Output Using Real-Time Sensor Data

Huijing Jiang, Xinwei Deng, Vanessa López & Hendrik F. Hamann


To cite this article: Huijing Jiang, Xinwei Deng, Vanessa López & Hendrik F. Hamann (2016) Online Updating of Computer Model Output Using Real-Time Sensor Data, Technometrics, 58:4, 472-482, DOI: [10.1080/00401706.2015.1080761](https://doi.org/10.1080/00401706.2015.1080761)

To link to this article: <http://dx.doi.org/10.1080/00401706.2015.1080761>

 View supplementary material 

 Accepted author version posted online: 29 Sep 2015.  
Published online: 11 Oct 2016.

 Submit your article to this journal 

 Article views: 51

 View Crossmark data 

# Online Updating of Computer Model Output Using Real-Time Sensor Data

Huijing JIANG

IBM Watson Research Center  
Yorktown Heights, NY 10598  
([huijiang@us.ibm.com](mailto:huijiang@us.ibm.com))

Xinwei DENG

Department of Statistics  
Virginia Tech  
Blacksburg, VA 24061  
([xdeng@vt.edu](mailto:xdeng@vt.edu))

Vanessa LÓPEZ and Hendrik F. HAMANN

IBM Watson Research Center  
Yorktown Heights, NY 10598  
([lopezva@us.ibm.com](mailto:lopezva@us.ibm.com); [hendrikh@us.ibm.com](mailto:hendrikh@us.ibm.com))

Data center thermal management has become increasingly important because of massive computational demand in information technology. To advance the understanding of the thermal environment in a data center, complex computer models are extensively used to simulate temperature distribution maps. However, due to management policies and time constraints, it is not practical to execute such models in a real time fashion. In this article, we propose a novel statistical modeling method to perform real-time simulation by dynamically fusing a *base*, steady-state solution of a computer model, and *real-time* thermal sensor data. The proposed method uses a Kalman filter and stochastic gradient descent method as computational tools to achieve real-time updating of the base temperature map. We evaluate the performance of the proposed method through a simulation study and demonstrate its merits in a data center thermal management application. Supplementary materials for this article are available online.

KEY WORDS: Data fusion; Dynamic Gaussian process; Kalman filter; Online optimization; Stochastic projected gradient.

## 1. INTRODUCTION

Data center energy consumption has increased dramatically in recent years, driven by the massive computing demand in every sector of the economy. There is great interest in developing data center thermal management to assist effective operation while avoiding excessive use of energy (López and Hamann 2011). Figure 1 shows the layout of a typical data center. Servers and other IT equipment are mounted in racks on a raised floor. The data center has alternating “cold aisles” and “hot aisles.” The inlet side of a server faces a cold aisle, while the exhaust side faces a hot aisle. Air conditioning units (ACUs) with large scale fans blow cooled air into the plenum of the data center thereby pressurizing it. The cooled air is then provided through perforated tiles that are placed in the cold aisles to the inlets of the servers. The heated exhaust air from the servers is returned to the ACUs via the intake locations.

To advance the understanding of thermal environment in data centers, it is essential to collect relevant environmental information, such as temperature, which is subject to change over time. A set of real-time sensors is used to provide the dynamic information about the operating conditions in a data center. The sensor readings of temperature are collected at fixed, short time intervals. However, because of the cost, these real-time sensors can only be deployed at a few selected locations. Thus, although the available sensor readings are in real-time, they lack detailed spatial information.

A temperature distribution map of high spatial resolution is typically obtained by the output of physics-based computer

models (López and Hamann 2011). Due to policies and time constraints, a computer model is usually scheduled to run at fixed time periods, for example, once per day. Thus, although such computer model outputs are available at a dense grid in space, they can only provide a relatively “static” temperature distribution map.

In an effort to obtain a real-time temperature distribution map of high resolution, we propose a novel modeling strategy by leveraging “static” but high-resolution computer model outputs and sparsely deployed real-time sensor data. Combining computer model output with observed measurement data, known as “data fusion,” has been well studied in the literature. The existing work in this field can be generally grouped into two directions. The first one is so-called Bayesian melding (Raftery, Givens, and Zeh 1995; Poole and Raftery 2000) where information from observational data and computer model output are combined through a latent data process to yield a Bayesian posterior distribution of the quantities of interest. McMillan et al. (2010) extended the Bayesian melding to analyze spatio-temporal data for air pollution monitoring. Another direction takes regression approaches where observations are regressed on computer model output and regression coefficients are allowed to vary over space (Guillas et al. 2008; Liu, Le, and Zidek 2008)

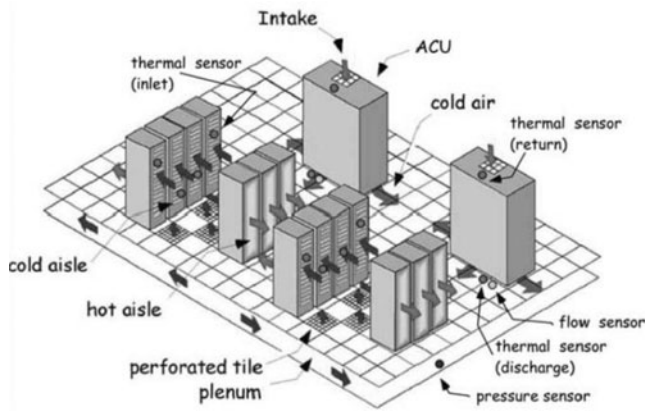


Figure 1. The layout of a typical data center (figure adapted with permission from Hamann et al. 2009).

or space-time (Berrocal, Gelfand, and Holland 2010a, 2010b, 2012).

In this article, we propose a dynamic Gaussian process model to leverage real-time sensor data to (i) update computer model outputs in real-time and (ii) issue forecasts that can be used to detect changes and schedule the execution of the computer model. Key contributions of this work are summarized as follows.

First, much existing “data fusion” work has focused on the situation where computer model output and observed data are available simultaneously. In this study, the observed data are sensor measurements coming in real time while the computer model is scheduled to run only at fixed time periods. Consequently, the computer model output is not as up-to-date as the sensor data. Moreover, to detect changes in a data center thermal environment, it is important to issue temperature forecasts. It is not straightforward to use existing “data fusion” methods for this purpose. In contrast, the proposed method considers the computer model output as a “static” base map and develops a *dynamic* Gaussian process model to update the base map whenever sensor data are brought in. The inference obtained from the estimated model can be used to issue one-step ahead forecasting for change detection.

Second, an effective and fast algorithm is essential for real-time updating of computer model output using sensor data. To achieve this goal, we take a fully frequentist approach for estimation and inference, and derive an online algorithm using a Kalman filter and stochastic gradient methods. We also develop theoretical justification of the stochastic gradient method in terms of optimal step size and associated error for the correlated (autocorrelated and spatially correlated) data. While in the field of “data fusion” existing work mostly relies on Bayesian computation, in general, Bayesian computation may not be designed for online updating, although a sequential Monte Carlo method (Doucet, Godsill, and Andrieu 2000; Storvik 2002) can potentially be applicable.

Finally, the way we model the spatio-temporal bias process aims at issuing temporal forecasting and spatial prediction simultaneously with a fast and stable algorithm. The existing research work on spatio-temporal modeling either considers time as continuous and emphasizes more on the spatial prediction (Gneiting 2002; Gneiting and Schlather 2002; Stein 2005;

Gneiting, Genton, and Guttorp 2007; Rodrigues and Diggle 2010; Fonseca and Steel 2011) or focuses more on forecasting the future by extending multivariate time-series models to spatio-temporal problems (Mardia et al. 1998; Cressie and Wikle 2011; Nobre, Sanso, and Schmidt 2011). The computation of the latter approach mainly relies on Markov chain Monte Carlo algorithms, which may not be computationally affordable for our application.

The remainder of this article is organized as follows. Section 2 details the methodology of the proposed dynamic Gaussian process model. In Section 3, we describe the procedure of parameter estimation using the Kalman filter and stochastic gradient algorithm as computational tools. The setup and results of experiments conducted on simulated data are provided in Section 4. In Section 5, we demonstrate the proposed method through its application in a representative data center example. Concluding remarks and future extensions are given in Section 6.

## 2. THE PROPOSED MODEL

To fuse the “static” computer model output with real-time sensor data, similar to Bayesian melding (McMillan et al. 2010), we consider a latent “true” process  $y_i^R(s)$  in terms of location  $s$  and time  $t$ . Following Kennedy and O’Hagan (2001), we formulate the relationship between the “true” process and the computer model as  $y_i^R(s) = y^M(s) + b_t(s)$ , where  $b_t(s)$  is the corresponding bias process between the computer model output and the latent “true” process. On the other hand, sensor data  $y_i^F(s)$  does not reflect the “true” process perfectly (Kennedy and O’Hagan 2001; Bayarri et al. 2007) and can be modeled as  $y_i^F(s) = y_i^R(s) + \epsilon_{s,t}^F$ , where  $\epsilon_{s,t}^F$  is the measurement error at time  $t$  and location  $s$ . Here, we assume that the measurement error is normally distributed with mean 0 and variance  $\sigma^2$ . Therefore, the relationship between real-time sensor data and “static” computer model output can be characterized as follows:

$$y_i^F(s) = y^M(s) + b_t(s) + \epsilon_{s,t}^F, \text{ for } t = 1, \dots, T. \quad (1)$$

Note that we can only observe deviance of the sensor data from computer model output  $y_i^F(s) - y^M(s)$  at the sensor locations, which are sparsely located in space.

To make predictions and issue forecasts, the key is to model the spatio-temporal structure of the underlying bias process  $b_t(s)$ . We consider that the bias at time  $t$  can be correlated with the bias of the past  $L$  time points. Thus, we propose an autoregressive AR( $L$ ) model for  $b_t(s)$ ,

$$b_t(s) = \sum_{l=1}^L \alpha_l b_{t-l}(s) + w_t(s), \quad (2)$$

where  $\alpha_l$  is the  $l$ th autoregressive coefficient jointly defining the correlation between the model discrepancy at time  $t$  and that at the  $l$ th past time point. Here we consider  $w_t(s)$  as a zero-mean Gaussian process (GaSP) (Kennedy and O’Hagan 2001) with covariance function

$$\mathbb{C}(w_t(s), w_t(s')) = \tau^2 \rho(s - s'; \theta) = \tau^2 \exp\{-\theta \|s - s'\|^\kappa\}, \quad (3)$$

where  $\mathbb{C}(\cdot, \cdot)$  is the covariance function,  $\theta$  and  $\kappa$  are specified values or unknown parameters, and  $\tau^2$  is the variance of the

process. Specifically, given  $n$  locations  $s_1, \dots, s_n$ , the process realization  $\mathbf{w}_t = (w_t(s_1), \dots, w_t(s_n))'$  follows a multivariate normal distribution  $N(\mathbf{0}, \tau^2 \boldsymbol{\Sigma}_\theta)$ . Here  $\boldsymbol{\Sigma}_\theta$  is an  $n \times n$  correlation matrix specified by (3) as  $\boldsymbol{\Sigma}_\theta = (\rho(s_i - s_j; \theta))_{n \times n}$ .

The proposed model in (2) implies that the dynamic change of model discrepancy is not completely random over time but autocorrelated. We consider a spatially and dynamically stationary model for the bias process because the goal is to use computer model output to capture the nonstationary trends in the data and the remaining bias is expected to be stationary. When any significant change occurs in the system setting, the proposed model is expected to detect this change and automatically update the computer model so that the model bias process remains stationary. Moreover, under the correlation function in (3) for the Gaussian process, we have the following results.

*Theorem 1.* Under the assumption that the autoregressive model in (2) is stationary, the bias process  $b_t(s)$  has a separable spatial-temporal covariance function  $\mathbb{C}(b_t(s), b_{t'}(s')) = \tau^2 \rho(s - s'; \theta) \gamma_L(t - t'; \boldsymbol{\alpha})$ , where  $\gamma_L(t - t'; \boldsymbol{\alpha})$  is an AR(L) autocovariance function of unit variance white noise.

The details of the proof are given in the supplementary materials. Here we adopt a separable and stationary spatio-temporal covariance model as our high-quality and relatively simple approximation. Other useful and flexible covariance models in this context can be found in research work of the spatio-temporal covariance model as discussed in Section 1.

### 3. ESTIMATION

Let  $\mathbf{y}_t^F = (y_t^F(s_1), \dots, y_t^F(s_n))'$  and  $\mathbf{y}^M = (y^M(s_1), \dots, y^M(s_n))'$  be the vector of sensor data at time  $t$  and computer model output at the sensor locations, respectively. Denote  $\mathbf{u}_t = \mathbf{y}_t^F - \mathbf{y}^M$  as the vector of observed deviances of real-time sensor measurements from computer model output and  $\mathbf{b}_t = (b_t(s_1), \dots, b_t(s_n))'$  as the unobserved model bias. One conventional way to predict the dynamic bias function is to use the joint distribution of model bias  $\mathbf{b}_{0:t} = (\mathbf{b}_0, \dots, \mathbf{b}_t)$  conditional on the past observed deviances  $\mathbf{u}_{0:t} = (\mathbf{u}_0, \dots, \mathbf{u}_t)$ . However, this joint estimation approach requires inversion of an  $n \times n$  matrix  $\boldsymbol{\Sigma}_\theta$  and a  $t \times t$  matrix  $\boldsymbol{\Sigma}_{\boldsymbol{\alpha}, t} = \{\gamma_L(t_i - t_j; \boldsymbol{\alpha})\}_{t_i, t_j=1, \dots, t}$  using all the data up to time  $t$ . Although the number of sensor locations  $n$  is assumed to be fixed in our model, the number of time points  $t$  will increase quickly as time goes by. Therefore, the estimation procedure will become very inefficient since the computational cost of matrix inversion operation on  $\boldsymbol{\Sigma}_{\boldsymbol{\alpha}, t}$  is  $O(t^3)$ .

To facilitate computational efficiency, we transform the proposed model (2) into a state-space model representation by taking advantage of the autoregressive model structure. We therefore can use a Kalman filter to predict the bias function  $b_t(s)$  as well as to estimate unknown parameters  $\boldsymbol{\psi}$ . The key advantage of a Kalman filter is that it provides a recursive procedure to avoid the inversion operation on a matrix of increasing size by using only the most recent data  $\mathbf{u}_t$  as well as the previous prediction values.

### 3.1 Kalman Filter

Let the unobserved state vector  $\boldsymbol{\beta}_t = (\mathbf{b}_t', \mathbf{b}'_{t-1}, \dots, \mathbf{b}'_{t-L+1})'$  be the model bias of the past  $L$  time points, then the equivalent state-space model representation of (2) takes the form of

$$\begin{aligned} \mathbf{u}_t &= \mathbf{H} \boldsymbol{\beta}_t + \boldsymbol{\epsilon}_t, \boldsymbol{\epsilon}_t \sim N(\mathbf{0}, \sigma^2 \mathbf{I}_n) \\ \boldsymbol{\beta}_t &= \mathbf{A} \boldsymbol{\beta}_{t-1} + \mathbf{R} \mathbf{w}_t, \mathbf{w}_t \sim N(\mathbf{0}, \tau^2 \boldsymbol{\Sigma}_\theta), \end{aligned} \quad (4)$$

where  $\mathbf{H} = (\mathbf{I}_n, \mathbf{0}, \dots, \mathbf{0})$ ,  $\mathbf{R} = (\mathbf{I}_n, \mathbf{0}, \dots, \mathbf{0})'$ , and  $\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 & \mathbf{A}_{2:L} \\ \mathbf{I}_{n \times (L-1)} & \mathbf{0} \end{pmatrix}$  with  $\mathbf{A}_{2:L} = (\mathbf{A}_2, \dots, \mathbf{A}_L)$  and  $\mathbf{A}_l = \alpha_l \mathbf{I}_n$ .

The observation Equation (4) links the observed response  $\mathbf{u}_t$  with the unobserved state vector  $\boldsymbol{\beta}_t$ . The state Equation (5) describes the dynamics of the state vector  $\boldsymbol{\beta}_t$  driven by the stochastic input  $\mathbf{w}_t$ . Matrices  $\mathbf{A}$  and  $\boldsymbol{\Sigma}_\theta$  depend on unknown parameters  $\theta$  and  $\alpha_1, \dots, \alpha_L$ . Under the state-space model representation, we can apply the Kalman filter technique (Kalman 1960; Shumway and Stoffer 2000; Durbin and Koopman 2001) to estimate the unknown state vector  $\boldsymbol{\beta}_t$  or equivalently  $\mathbf{b}_t$  in a recursive fashion. Details of the recursive estimation algorithm are provided in the supplementary materials.

Moreover, the updated estimation from the Kalman filter provides a reformulation to the likelihood function, from which we can estimate the unknown parameters  $\boldsymbol{\psi}$ . Denote  $f(\mathbf{u}_0, \dots, \mathbf{u}_T; \boldsymbol{\psi})$  to be the joint probability density of  $\mathbf{u}_0, \dots, \mathbf{u}_T$ , and  $f(\mathbf{u}_t | \mathbf{u}_{0:t-1}; \boldsymbol{\psi})$  to be the conditional probability density of  $\mathbf{u}_t | \mathbf{u}_{0:t-1}$ . Since  $\mathbb{E}[\mathbf{u}_t | \mathbf{u}_{0:t-1}] = \mathbf{b}_{t|t-1}$  and  $\mathbb{V}[\mathbf{u}_t | \mathbf{u}_{0:t-1}] = \mathbf{H} \mathbf{P}_{t|t-1} \mathbf{H}' + \sigma^2 \mathbf{I}_n = \mathbf{F}_t$  where  $\mathbf{P}_{t_1|t_2} = \mathbb{V}[\boldsymbol{\beta}_{t_1} | \mathbf{u}_{0:t_2}]$ , the likelihood function of our model can be rewritten as

$$\begin{aligned} \log L(\boldsymbol{\psi}) &= \log f(\mathbf{u}_0, \dots, \mathbf{u}_T; \boldsymbol{\psi}) \\ &= -\frac{1}{2} \sum_{t=1}^T [\log |\mathbf{F}_t| + \mathbf{v}_t' \mathbf{F}_t^{-1} \mathbf{v}_t] + \text{constant}, \end{aligned} \quad (6)$$

where  $\mathbf{v}_t = \mathbf{u}_t - \mathbf{b}_{t|t-1}$ . Using the log-likelihood function in (6), the estimate of the unknown parameter vector  $\boldsymbol{\psi}$  can be obtained by

$$\hat{\boldsymbol{\psi}}_T^* = \arg \min \sum_{t=1}^T \log |\mathbf{F}_t| + \mathbf{v}_t' \mathbf{F}_t^{-1} \mathbf{v}_t, \quad (7)$$

under appropriate constraints. Note that all the elements  $\mathbf{F}_t$  and  $\mathbf{v}_t$  in the above objective function are intermediate results from the Kalman filter algorithm. Therefore, once we obtain updates from the Kalman filter, the likelihood function in (6) is readily available to be optimized for the estimation of unknown parameters  $\boldsymbol{\psi}$ .

However, solving such a nonlinear optimization problem can be time-consuming because the objective function in (7) is highly nonconvex and nonsmooth with respect to the parameters  $\boldsymbol{\psi}$ . To estimate the parameters promptly, in the next section, we apply the stochastic projected gradient method (Nemirovski et al. 2009) to update  $\boldsymbol{\psi}$  in an online fashion.



### 3.2 Online Parameter Estimation

The stochastic projected gradient approach is commonly used in online optimization (see Nemirovski et al. 2009; Bottou 2010, and references therein). Here we consider using a batch version of the stochastic projected gradient method to update  $\boldsymbol{\psi}$ . Define an index set  $\mathcal{B} = \{m, \dots, m + Q\}$  of size  $Q$  and we update the parameters  $\boldsymbol{\psi}$  when a new batch of data of size  $Q$  is available. Given the previous parameter estimates  $\hat{\boldsymbol{\psi}}_{B-1}$ , we update the parameter vector by

$$\hat{\boldsymbol{\psi}}_B = P_G \left( \hat{\boldsymbol{\psi}}_{B-1} - \eta_B \sum_{t \in \mathcal{B}} \frac{\partial \log f(\mathbf{u}_t | \mathbf{u}_{0:t-1}; \boldsymbol{\psi})}{\partial \boldsymbol{\psi}} \Big|_{\hat{\boldsymbol{\psi}}_{B-1}} \right), \quad (8)$$

where  $P_G(\cdot)$  is a projection operator onto the feasible set  $G$  and  $\eta_B$  is the step size for batch  $B$ . Note that the objective function  $-\log f(\mathbf{u}_t | \mathbf{u}_{0:t-1}; \boldsymbol{\psi}) \equiv l(\boldsymbol{\psi}, \mathbf{u}_t | \mathbf{u}_{0:t-1})$  is not convex. We also examine the convergence of the proposed online parameter estimation under several regular assumptions:

- A1. The expectation of the stochastic subgradient  $g(\boldsymbol{\psi}) = \mathbb{E}[\mathbf{G}(\boldsymbol{\psi}, \mathbf{u}_t | \mathbf{u}_{0:t-1})] = \partial \mathbb{E}[l(\boldsymbol{\psi}, \mathbf{u}_t | \mathbf{u}_{0:t-1})] / \partial \boldsymbol{\psi}$  exists and is bounded, that is, there exist a positive number  $M$  such that  $\mathbb{E}[\|\mathbf{G}(\boldsymbol{\psi}, \mathbf{u}_t | \mathbf{u}_{0:t-1})\|_2^2] \leq M^2, \forall \boldsymbol{\psi} \in \Psi$ .
- A2. The feasible region of function  $l(\boldsymbol{\psi}, \mathbf{u}_t | \mathbf{u}_{0:t-1})$  can be divided into  $K$  convex sets, that is,  $\Psi = \Psi_1 \cup \Psi_2 \cup \dots \cup \Psi_K$ .
- A3. The initial value  $\boldsymbol{\psi}_0$  is an interior point of the convex region  $\Psi_k$  where the optimum value  $\boldsymbol{\psi}^*$  lies.

*Theorem 2.* Under Assumptions A1–A3, the optimal step size  $\eta_B = \frac{\sqrt{2D_k}}{M\sqrt{|\mathcal{B}|}}$ , where  $D_k = \max_{\boldsymbol{\psi}, \boldsymbol{\psi}' \in \Psi_k} \mathbb{E}[\|\boldsymbol{\psi} - \boldsymbol{\psi}'\|_2^2]$ . The associated expected error of the objective value is  $\frac{1}{T} \mathbb{E}[\sum_{B=1}^T \sum_{t \in \mathcal{B}} \mathbb{E}[l(\hat{\boldsymbol{\psi}}_B, \mathbf{u}_t | \mathbf{u}_{0:t-1}) - \log L(\boldsymbol{\psi}^*)]] \leq \frac{\sqrt{2D_k}M}{\sqrt{|\mathcal{B}|}}$ .

The theorem implies that a large batch size  $|\mathcal{B}|$  can lead to a tight bound in terms of errors. However, it also increases the computational cost for the online updating. On the other hand, a large value of  $D_k$  indicates a large range for the convex regions  $\Psi_k$ , which could result in loose error bound given the same batch size. To achieve the same error bound, one can increase the batch size. Similar arguments apply to  $M$ . When the observed data contain more noise, the log-likelihood function is less smooth. It results in a larger value of  $M$ , which would increase the error bound given the same batch size.

### 3.3 Prediction and Forecasting

In the data center application, sensor data are often measured at limited locations but the essential interest is to get a full temperature distribution map of the entire region. Therefore, it is important to predict the model discrepancy at new locations. Denote  $\mathbf{s}^*$  as a new location to issue the prediction, and given observed deviances up to current time  $t$ , the distribution of predicted model bias at the new location  $\mathbf{s}^*$  is  $b_t(\mathbf{s}^*) | \mathbf{u}_{0:t} \sim N(b_{t|t}(\mathbf{s}^*), C_{t,t|t}(\mathbf{s}^*))$ , where  $b_{t|t}(\mathbf{s}^*) = \boldsymbol{\rho}'_{\theta}(\mathbf{s}^*) \boldsymbol{\Sigma}_{\theta}^{-1} \mathbf{b}_{t|t}$  and  $C_{t,t|t}(\mathbf{s}^*) = \boldsymbol{\rho}'_{\theta}(\mathbf{s}^*) \boldsymbol{\Sigma}_{\theta}^{-1} \mathbf{C}_{t,t|t} \boldsymbol{\Sigma}_{\theta}^{-1} \boldsymbol{\rho}_{\theta}(\mathbf{s}^*) + \tau^2 \gamma_L(0; \boldsymbol{\alpha}) \{1 - \boldsymbol{\rho}'_{\theta}(\mathbf{s}^*) \boldsymbol{\Sigma}_{\theta}^{-1} \boldsymbol{\rho}_{\theta}(\mathbf{s}^*)\}$ . Here  $\boldsymbol{\rho}_{\theta}(\mathbf{s}^*) = \{\rho(\mathbf{s}^* - \mathbf{s}_i; \theta)\}_{i=1, \dots, n}$  is the correlation vector between the observed location  $\mathbf{s}^*$  and the sample locations  $\mathbf{s}_1, \dots, \mathbf{s}_n$  and  $\gamma_L(0; \boldsymbol{\alpha})$  is the autocovariance function of unit variance noise at  $t - t' = 0$ . With a base tem-

perature map based on computer model output  $y^M(\mathbf{s}^*)$ , we can then obtain a bias-corrected prediction of temperature as  $y_{t|t}(\mathbf{s}^*) = y^M(\mathbf{s}^*) + b_{t|t}(\mathbf{s}^*)$ .

Similarly, we can derive the distribution of one-step ahead forecasting is  $b_{t+1}(\mathbf{s}^*) | \mathbf{u}_{0:t} \sim N(b_{t+1|t}(\mathbf{s}^*), C_{t+1,t+1|t}(\mathbf{s}^*))$ . In addition, we can obtain a prediction interval and one-step ahead forecasting interval from the following result.

*Theorem 3.* The unconditional variances of prediction errors and the unconditional variances of forecasting errors are  $\mathbb{V}[b_t(\mathbf{s}^*) - b_{t|t}(\mathbf{s}^*)] = C_{t,t|t}(\mathbf{s}^*)$  and  $\mathbb{V}[b_{t+1}(\mathbf{s}^*) - b_{t+1|t}(\mathbf{s}^*)] = C_{t+1,t+1|t}(\mathbf{s}^*)$ .

## 4. SIMULATION STUDY

We evaluate the performance of the proposed model in (2) through a simulation study. In this study, the sensor locations  $S = \{\mathbf{s}_i \in [0, 20] \times [0, 20] : i = 1, \dots, n\}$  are randomly sampled from a square  $[0, 20] \times [0, 20]$  and the computer model is simulated at grid-points  $G = \{\mathbf{g}_{ij} = (i - 1/2, j - 1/2) : i = 1, \dots, 20; j = 1, \dots, 20\}$ . We generated  $u_t(\mathbf{s}) = y_t^F(\mathbf{s}) - y^M(\mathbf{s})$  at both  $N = 400$  prediction grids and  $n$  sensor locations from the joint distribution  $u_t(\mathbf{s}) = b_t(\mathbf{s}) + \epsilon_{t,\mathbf{s}}$  with  $\mathbf{b} \sim GP(0, \boldsymbol{\Gamma}_{\alpha,T} \otimes \tau^2 \boldsymbol{\Sigma}_{\theta})$  and  $\boldsymbol{\epsilon} \sim N(0, \sigma^2 \mathbf{I}_{(N+n)T})$ . Here  $\mathbf{b} = \{\mathbf{b}_t(\mathbf{s}), t = 1, \dots, T; \mathbf{s} \in S \cup G\}$  is the “true” model bias vector to be recovered and  $\boldsymbol{\epsilon} = \{\epsilon_{t,\mathbf{s}}, t = 1, \dots, T; \mathbf{s} \in S \cup G\}$  is the measurement error. The total number of time points  $T$  is 1000 in this simulation study. The true values of model parameters are shown in Table 1.

The performance of the proposed method is evaluated via parameter estimation accuracy, prediction and forecasting capability, and computational time for online and offline methods. The batch size of the online method  $Q$  is 100 in this study. Here, the offline method is defined as estimating parameters by directly solving the optimization problem in (7). We examine the proposed method by considering the following three aspects, (i) the density of sample locations measured by the ratio of the number of sample locations to the number of prediction grid-points, that is,  $n/N$ ; (ii) the signal-to-noise ratio (SNR) condition defined as the variance of the bias process divided by the measurement error variance, that is,  $\text{SNR} = \tau^2 / \sigma^2$ ; and (iii) the extent of spatial dependence measured by  $\theta$ .

To evaluate the prediction performance of the proposed method, the simulated data at  $n$  sampled locations are taken as observed deviance  $u_t(\mathbf{s})$  and then we predict the model bias  $b_t(\mathbf{g})$  at  $N$  grid-points  $\mathbf{g} \in G$ . For the prediction  $b_{t|t}(\mathbf{g})$  and one-step ahead forecasting  $b_{t+1|t}(\mathbf{g})$ , the prediction accuracy is measured by the mean squared prediction error (MSPE):  $\frac{1}{N} \frac{1}{T} \sum_{\mathbf{g}_{ij} \in G} \sum_{t=1}^T [b_{t|t}(\mathbf{g}_{ij}) - b_t(\mathbf{g}_{ij})]^2$  and mean squared forecasting error (MSFE):  $\frac{1}{N} \frac{1}{T-1} \sum_{\mathbf{g}_{ij} \in G} \sum_{t=1}^{T-1} [b_{t+1|t}(\mathbf{g}_{ij}) - b_{t+1}(\mathbf{g}_{ij})]^2$ , respectively. We also evaluate the accuracy of interval estimation through its coverage defined as the percentage of predicted values falling into the calculated intervals. Tables 1–3 report the average values and standard deviations of parameter estimation, the MSPE and the coverage of prediction intervals (PI), and the MSFE and coverage of one-step ahead forecasting interval (FI) over 100 replications for the offline and online methods. We summarize our findings as follows.

Table 1. The means and standard errors (in parentheses) of the model parameters under the *offline* | *online* methods

$n$	SNR	$\theta$	$\alpha_1 = 0.5$	$\alpha_2 = 0.3$	$\alpha_3 = 0.1$
20	10	0.25	0.502 (0.008) 0.489 (0.012)	0.301 (0.010) 0.303 (0.014)	0.097 (0.009) 0.112 (0.013)
20	10	0.04	0.505 (0.010) 0.473 (0.011)	0.303 (0.012) 0.298 (0.012)	0.096 (0.012) 0.119 (0.011)
20	5	0.25	0.503 (0.008) 0.476 (0.012)	0.302 (0.011) 0.304 (0.013)	0.096 (0.010) 0.126 (0.013)
20	5	0.04	0.504 (0.011) 0.452 (0.010)	0.303 (0.012) 0.295 (0.011)	0.097 (0.012) 0.135 (0.010)
50	10	0.25	0.502 (0.006) 0.486 (0.007)	0.301 (0.006) 0.302 (0.007)	0.098 (0.006) 0.115 (0.007)
50	10	0.04	0.504 (0.008) 0.475 (0.009)	0.301 (0.010) 0.298 (0.010)	0.099 (0.008) 0.122 (0.0106)
50	5	0.25	0.503 (0.006) 0.478 (0.016)	0.301 (0.006) 0.303 (0.016)	0.099 (0.007) 0.125 (0.018)
50	5	0.04	0.504 (0.008) 0.452 (0.009)	0.302 (0.010) 0.296 (0.009)	0.098 (0.009) 0.135 (0.009)

$n$	SNR	$\theta$	$\theta = 0.25(0.04)$	$\tau^2 = 0.8$	$\sigma^2 = 0.08(0.16)$
20	10	0.25	0.252 (0.006) 0.245 (0.011)	0.798 (0.012) 0.794 (0.016)	0.083 (0.005) 0.091 (0.011)
20	10	0.04	0.041 (0.001) 0.039 (0.002)	0.798 (0.013) 0.800 (0.017)	0.082 (0.002) 0.106 (0.010)
20	5	0.25	0.257 (0.007) 0.241 (0.012)	0.799 (0.014) 0.792 (0.019)	0.165 (0.007) 0.187 (0.015)
20	5	0.04	0.042 (0.001) 0.040 (0.002)	0.801 (0.014) 0.803 (0.018)	0.164 (0.003) 0.209 (0.011)
50	10	0.25	0.253 (0.003) 0.247 (0.007)	0.799 (0.007) 0.797 (0.010)	0.082 (0.002) 0.088 (0.007)
50	10	0.04	0.041 (0.000) 0.044 (0.014)	0.795 (0.011) 0.801 (0.015)	0.081 (0.001) 0.092 (0.005)
50	5	0.25	0.255 (0.004) 0.245 (0.008)	0.802 (0.008) 0.798 (0.011)	0.163 (0.003) 0.177 (0.009)
50	5	0.04	0.042 (0.000) 0.039 (0.001)	0.798 (0.011) 0.801 (0.015)	0.162 (0.001) 0.208 (0.007)

Table 2. The means and standard errors (in parentheses) of the MSPE and coverage of 90% and 95% prediction intervals (PI) for prediction  $b_{t|t}$  under the *offline* | *online* method

$n$	SNR	$\theta$	MSPE	90% Coverage	95% Coverage
20	10	0.25	2.215 (0.393) 2.216 (0.393)	0.902 (0.030) 0.900 (0.045)	0.951 (0.021) 0.948 (0.030)
20	10	0.04	0.617 (0.261) 0.619 (0.257)	0.904 (0.057) 0.888 (0.062)	0.952 (0.039) 0.941 (0.045)
20	5	0.25	2.236 (0.395) 2.234 (0.394)	0.903 (0.030) 0.899 (0.045)	0.952 (0.021) 0.948 (0.030)
20	5	0.04	0.677 (0.273) 0.684 (0.272)	0.905 (0.055) 0.878 (0.063)	0.953 (0.038) 0.934 (0.047)
50	10	0.25	1.439 (0.271) 1.440 (0.271)	0.902 (0.029) 0.899 (0.033)	0.952 (0.020) 0.949 (0.023)
50	10	0.04	0.180 (0.071) 0.190 (0.080)	0.884 (0.049) 0.889 (0.064)	0.939 (0.035) 0.940 (0.047)
50	5	0.25	1.513 (0.279) 1.514 (0.280)	0.905 (0.027) 0.901 (0.053)	0.953 (0.019) 0.948 (0.035)
50	5	0.04	0.410 (0.258) 0.418 (0.261)	0.895 (0.069) 0.883 (0.055)	0.942 (0.037) 0.937 (0.041)

Table 3. The means and standard errors (in parentheses) of the MSFE and coverage of 90% and 95% forecasting intervals (FI) for one-step ahead forecasts  $b_{t+1|t}$  under the *offline* | *online* methods

$n$	SNR	$\theta$	MSPE	90% Coverage	95% Coverage
20	10	0.25	2.404 (0.410) 2.406 (0.410)	0.901 (0.030) 0.900 (0.042)	0.951 (0.021) 0.948 (0.029)
20	10	0.04	1.220 (0.440) 1.223 (0.440)	0.900 (0.063) 0.890 (0.066)	0.950 (0.044) 0.943 (0.047)
20	5	0.25	2.415 (0.412) 2.414 (0.411)	0.902 (0.030) 0.899 (0.043)	0.951 (0.021) 0.948 (0.029)
20	5	0.04	1.253 (0.449) 1.261 (0.450)	0.900 (0.062) 0.882 (0.067)	0.950 (0.043) 0.938 (0.049)
50	10	0.25	1.832 (0.309) 1.833 (0.309)	0.901 (0.028) 0.898 (0.031)	0.951 (0.020) 0.949 (0.022)
50	10	0.04	0.909 (0.338) 0.917 (0.340)	0.895 (0.069) 0.896 (0.069)	0.946 (0.049) 0.947 (0.048)
50	5	0.25	1.878 (0.314) 1.883 (0.315)	0.902 (0.028) 0.901 (0.048)	0.952 (0.019) 0.949 (0.031)
50	5	0.04	1.069 (0.426) 1.078 (0.429)	0.895 (0.069) 0.887 (0.071)	0.946 (0.049) 0.941 (0.051)

First, under a more extensive spatial correlation ( $\theta = 0.04$ ), larger signal-to-noise ratio (SNR = 10) and/or more sample locations ( $n = 50$ ), the prediction/forecasting accuracy measured by MSPE is improved significantly. Second, the accuracy of parameter estimation and interval estimation does not change much across various settings. Third, the online method performs reasonably well compared with the offline method. The accuracy of parameter estimation, prediction/forecasting, and interval estimation from the online method is very close to that

from the offline method. But we do notice that the standard deviation of the online method is slightly larger than the offline method. This is reasonable since the online method uses fewer data points each time and can be affected more by the variability in the data.

Table 4 compares the estimation results of different batch size  $Q$ . Based on the numerical results, the estimation accuracy of  $\theta$  and  $\tau$  is greatly improved as batch size  $Q$  increases. Moreover, the prediction accuracy measured by MSPE and MSFE is also

Table 4. The means and standard errors (in parentheses) of the estimation results for varying  $Q$  ( $n = 20$ ,  $\text{SNR} = 10$ ,  $\theta = 5$ )

	$Q = 5$	$Q = 10$	$Q = 50$	$Q = 100$
$\alpha_1 = 0.5$	0.494(0.023)	0.488(0.015)	0.478(0.011)	0.477(0.011)
$\alpha_2 = 0.3$	0.306(0.017)	0.305(0.015)	0.300(0.012)	0.299(0.012)
$\alpha_3 = 0.1$	0.112(0.019)	0.119(0.014)	0.120(0.010)	0.120(0.020)
$\theta = 0.04$	10.505(11.999)	6.952(9.071)	0.119(0.206)	0.041(0.011)
$\tau = 0.8$	0.727(0.049)	0.770(0.026)	0.799(0.206)	0.800(0.017)
$\sigma = 0.08$	0.621(0.953)	0.511(0.583)	0.098(0.028)	0.098(0.008)
MSPE	2.348(1.493)	2.111(1.457)	0.859(0.620)	0.623(0.262)
90% PI coverage	0.918(0.073)	0.920(0.080)	0.906(0.077)	0.887(0.070)
95% PI coverage	0.960(0.055)	0.960(0.065)	0.950(0.059)	0.940(0.051)
MSFE	2.508(1.351)	2.330(1.295)	1.401(0.641)	1.225(0.442)
90% FI coverage	0.915(0.068)	0.918(0.066)	0.904(0.045)	0.893(0.066)
95 % FI coverage	0.958(0.047)	0.960(0.045)	0.952(0.045)	0.945(0.047)

improved with the increase of batch size  $Q$ . It is worth pointing out that the accuracy of other parameter estimates is quite stable with different batch size  $Q$ .

The computational efficiency of the offline and online methods is reported in Table 5. The computation time of the offline method is referred to as the time of solving the constrained optimization problem (7) using data of all  $T = 1000$  time points. The computation time of the online method is the total time of updating the parameters across the whole time period. Since we update the parameters every  $Q = 100$  time points, it is equivalent to the total time needed to solve  $B = T/Q$  stochastic projected gradient descent problems (8). Under all the settings, we clearly observe that the offline method is very slow for real-time updating while the online method greatly speeds up the computation. We thus conclude that the online method can provide reasonably accurate results and is computationally efficient, which is crucial for updating computer model output using sensor data in a real-time fashion.

We also examine the use of model selection (McQuarrie and Tsai 1998), including Akaike information criterion (AIC) and Bayesian information criterion (BIC), to select the optimal AR order  $L$  under the setting  $\theta = 0.25$  and  $\text{SNR} = 5$ . Specifically, they select the order  $L$  by minimizing  $-2 \log L(\psi) + J(L)$ , where  $\log L(\psi)$  is the log-likelihood function in (6), and  $J(L) = 2L$  for AIC and  $J(L) = L \log(n)$  for BIC. Figure S1 in the supplementary materials reports AIC and BIC values for models of AR orders ranging from 2 to 5. The result indicates that both information criteria can identify the correct AR order

of  $L = 3$ . For other settings, AIC and BIC are also expected to work reasonably well.

### 5. CASE STUDY

In this section, we present a case study motivated by an industrial project aiming at better managing data center thermal system and reducing energy cost (Hamann et al. 2009).

*Data Description.* As shown in Figure 2(a), the data center in this case study has 15 racks, labeled R1–R15, and two ACUs. Unlabeled objects represent equipment without a significant amount of power consumption. Since cooled air comes out along inlet sides, it is most critical to monitor the temperature at the server inlet sides. As an illustration, we presented the results along the inlet faces of R1–R5. Referring to Figure 2(b), there are  $n = 35$  sensors marked as solid dots along this inlet face to monitor the real-time temperature. The temperature outputs from the computer model are generated at grid points  $g_j$ ,  $j = 1, \dots, N$ , which are marked as circles in Figure 2(b).

The detailed temperature data are available for 25 cases corresponding to different fan settings of the air conditioning units (López and Hamann 2011). These cases were used to generate simulated sensor readings for 1000 time points. They were designed to represent a network of sparsely located sensors, as would be typically present in a data center, and represent a realistic scenario in data center energy management systems. The simulated sensor readings provided input data for the computer model solver, generating temperature maps along the 1000 time points.

Figure 3 provides a diagram of the proposed real-time updating and detecting strategy. In Step 1, we initialize the model parameters using “historical data” and obtain the output from the computer model given data at  $t = 0$ . Step 2 updates the model parameters whenever a new batch of sensor data are available and issues bias-corrected temperature prediction (Step 2(b)) at grid points and one-step ahead forecasting (Step 2(c)) at the sensor locations. In Step 3, the one-step ahead forecasts are compared with newly arriving sensor data to examine if any significant change occurs in the current data center settings.

*Online Prediction.* Following Step 1(a) in Figure 3, we consider sensor data of the first 300 time points as “historical” data, and use them to obtain initial estimates  $\hat{\psi}_0$ . Specifically,

Table 5. Comparison of the computation time (in seconds) for the offline and online methods: the mean and the standard deviations (in parentheses)

$n$	SNR	$\theta$	Offline	Online
20	10	0.25	1940.689(528.685)	4.083(0.838)
20	10	0.04	1944.242(519.749)	3.590(0.282)
20	5	0.25	2023.049(536.360)	4.065(0.816)
20	5	0.04	1910.227(566.626)	2.817(0.190)
50	10	0.25	7223.010(1622.481)	9.417(1.282)
50	10	0.04	5674.745(1580.333)	8.378(1.579)
50	5	0.25	6879.867(2423.806)	10.573(4.049)
50	5	0.04	8073.445(3105.651)	9.975(3.894)

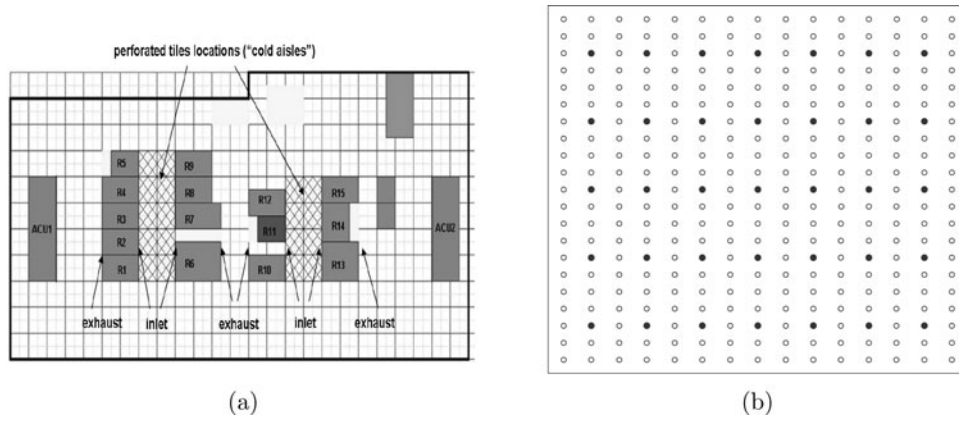


Figure 2. Sample data center: (a) 2D view from the top; (b) server inlet faces for racks 1–5 where sensor location are marked as solid dots and the prediction grid-points are marked as circles.

the computer model output is obtained using temperature sensor data and air flow settings at  $t = 1$  to provide input values for the boundary conditions of PDEs. Given sensor data at  $t = 2, \dots, 300$ , we calculate the deviations of temperature measurements from the computer model output for time  $t = 1$  at the sensor locations  $s_i$ , that is,  $u_t(s_i), t = 2, \dots, 300; i = 1, \dots, n$ . Finally, these calculated deviations are used to obtain the initial parameter  $\hat{\psi}_0$  through the constrained optimization (7).

The remaining 700 sensor data points are then used as temperature measurements in real-time. For notation convenience, we denote the time stamps as  $t \leftarrow (t - 301)$ . Then one can view the “real-time” sensor measurements as collected at  $t = 0, 1, \dots, 699$ . Following Steps 1(b)–(c), we obtain the initial temperature map from the output of the computer model.

We remark that although in this example, “historical” and “real-time” are two consecutive time periods, “historical” data can come from any previous time period since they are simply used for initial estimation.

Figure 4 displays the prediction results at time  $t = 5, 55, 105$ , respectively. Here, we set the batch size  $Q = 50$  so that the parameters  $\psi$  are updated every 50 time points. To examine the accuracy of the prediction results, we request the computer model to be executed at the  $t = 5, 55, 105$  time stamps with sensor measurements as input of boundary information. Since it is not realistic to obtain temperature maps from real measurements, such computer model outputs are considered as the closest scenario to the real temperature maps. From the results in Figure 4, the predicted bias from our method closely

Step 1. Set  $t = 0$  and  $B = 0$ , run

- a. Use “historical” data to get initial parameter estimates  $\hat{\psi}_0$ .
- b. Run computer model to get simulation output at meshes  $\{y_0^M(\mathbf{g}_j)\}_{j=1,\dots,N}$  based on sensor data  $\{y_0^F(\mathbf{s}_i)\}_{i=1,\dots,n}$ .

Step 2. Run

- a. When  $t = Q(B + 1)$ , update parameter estimates  $\hat{\psi}_B = P_G(\hat{\psi}_{B-1} - \eta_B \sum_{m=1}^Q \frac{\partial \log f(\mathbf{u}_{t-m} | \mathbf{u}_{0:t-m-1}; \psi)}{\partial \psi} |_{\hat{\psi}_{B-1}})$  and set  $B = B + 1$ .
- b. Issue temperature prediction at time  $t$  on grid points, i.e.,  $y_{t|t}^M(\mathbf{g}_j) = y_0^M(\mathbf{g}_j) + b_{t|t}(\mathbf{g}_j), j = 1, \dots, N$  and 95% prediction intervals.
- c. Issue one-step ahead temperature forecasts at time  $t + 1$  on  $n$  sensor locations, i.e.,  $y_{t+1|t}^M(\mathbf{s}_i) = y_0^M(\mathbf{s}_i) + b_{t+1|t}(\mathbf{s}_i), i = 1, \dots, n$  and 95% forecasting intervals.

Step 3. Run

- a. Acquire new sensor data  $\{y_{t+1}^F(\mathbf{s}_i)\}_{i=1,\dots,n}$ .
- b. If  $\{y_{t+1}^F(\mathbf{s}_i)\}_{i=1,\dots,n}$  are significantly different from  $y_{t+1|t}^M(\mathbf{s}_i)$ , go to Step 1; otherwise, set  $t = t + 1$  and go to Step 2.

Figure 3. A diagram of real-time updating the computer model output.



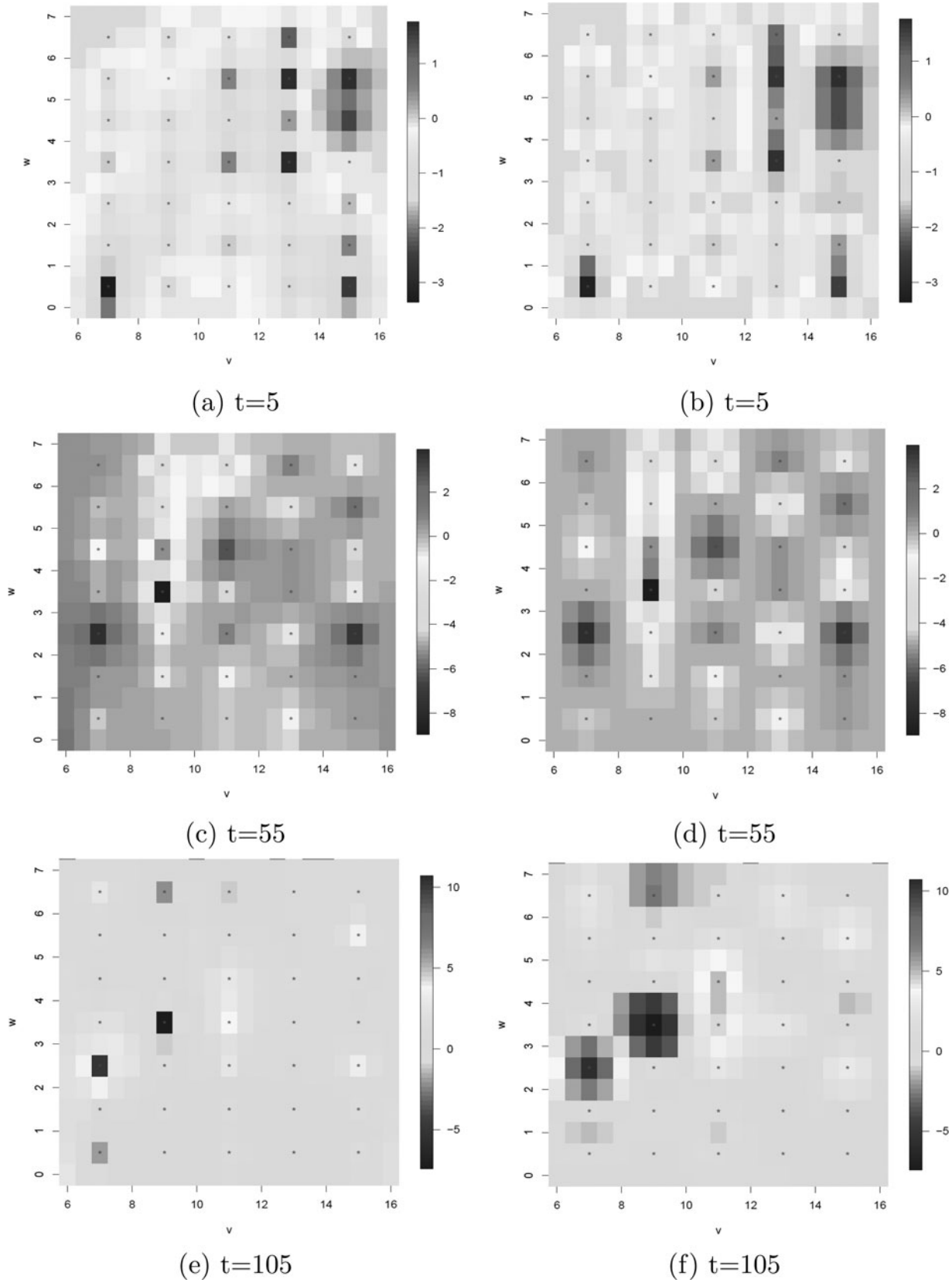


Figure 4. Prediction of temperature measurements: (a), (c), (e) for the deviance of the computer model output executed at  $t = 5, 55, 105$  from the base temperature map; (b), (d), (f) for predicted bias from the proposed model.

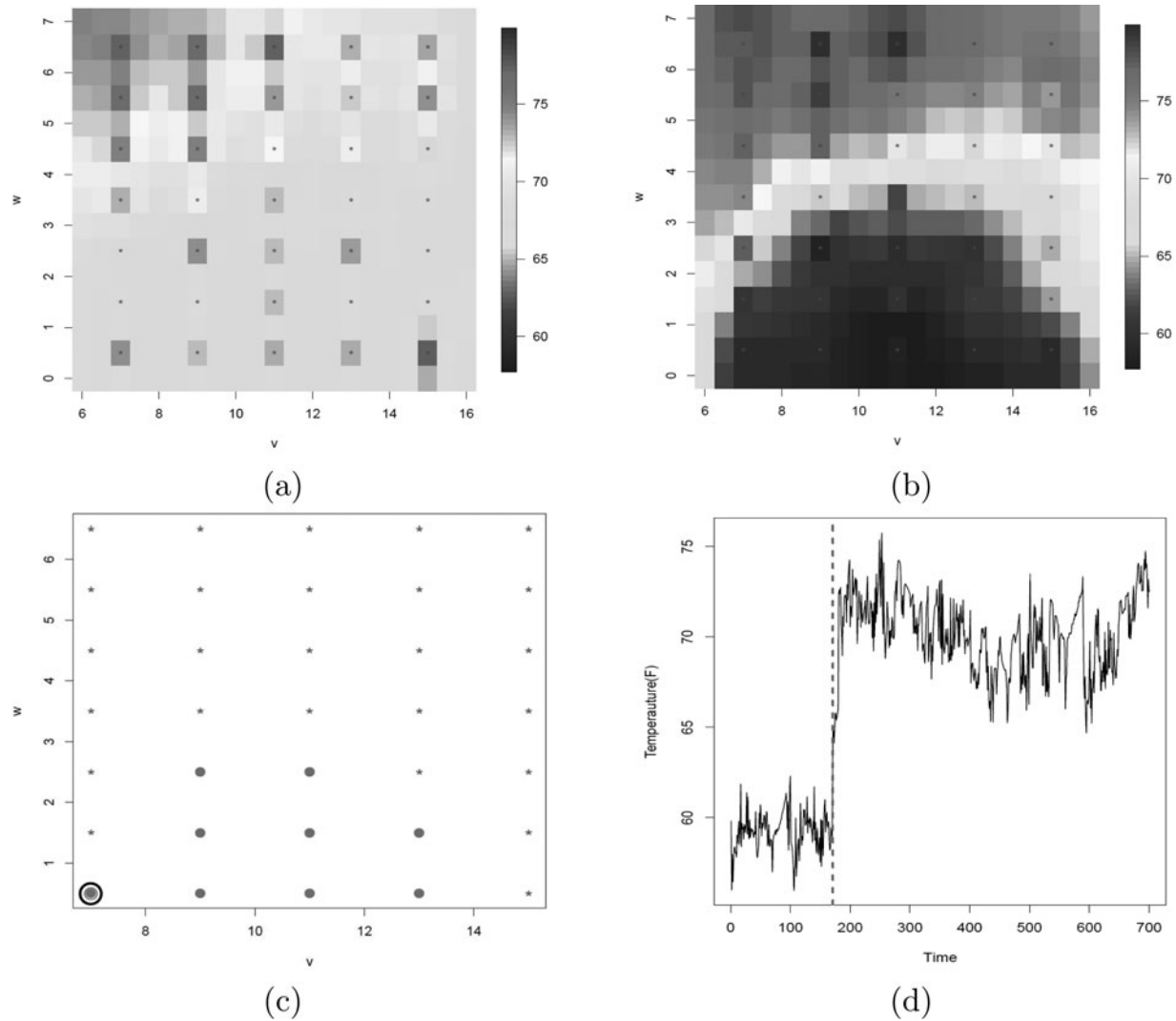


Figure 5. Online detection results: (a) true temperature map; (b) one-step ahead temperature forecast map; (c) detected temperature change at sensor locations highlighted in solid dots; (d) temperature measurements of the sensor circled in (c).

resembles the deviance of the computer model output executed at  $t = 5, 55, 105$  from the base temperature map.

We also applied a multivariate Portmanteau statistic (Hosking 1980) to test the goodness of fit of the stationary AR( $L$ ) model for the bias process (2). Since we expect changes in the system settings, the temperature data would only be stationary at the beginning until certain significant system change occurs. Therefore, we applied the Portmanteau test to the first 10, 50, 100 time points, and the results confirm that a stationary AR( $L$ ) fits well to the bias process.

**Online Detection.** Under the proposed framework, at time stamp  $t$ , we can issue forecasting of the expected temperature for the next time stamp  $t + 1$  and quantify their uncertainties. Once the sensor measurements at time  $t + 1$  are available, we can compare them with our forecasts. If they deviate significantly from the issued forecasts, it is very likely that a significant change occurred in the current operating conditions. This indicates that we need to reexecute the computer model to reflect the changes in the operating conditions.

In this study, statistical hypothesis testing is applied to examine if the forecasted temperature  $y_{t+1|t}(s_i)$  are significantly

different from the sensor measurements  $y_{t+1}(s_i)$ . Note that measurements from multiple sensors need to be compared simultaneously and thus a multiple testing approach is appropriate. In the multiple testing, each test has the null hypothesis  $H_i^0 : y_{t+1}(s_i) - y_{t+1|t}(s_i) = 0$  against the alternative  $H_i^1 : y_{t+1}(s_i) - y_{t+1|t}(s_i) \neq 0$  for  $i = 1, \dots, n$ . As the spatially correlated model bias leads to dependent test statistics, we apply the false discovery ratio (FDR) test (Benjamini and Hochberg 1995) for multiple testing. It can effectively control the false discovery rate for dependent testings (Benjamini and Yekutieli 2001). The key idea of the procedure is as follows. Under the individual  $H_i^0$ , we compute the observed  $p$ -values  $p_i = 2[1 - \Phi(\frac{|y_{t+1}(s_i) - y_{t+1|t}(s_i)|}{\sqrt{C_{t+1,t+1|t}(s_i) + \sigma^2}})]$ , where  $\Phi$  is the cumulative distribution function of the standard normal distribution,  $y_{t+1|t}(s_i) = y^M(s_i) + b_{t+1|t}(s_i)$  is the forecast of temperature, and  $C_{t+1,t+1|t}(s_i)$  is the  $s_i$ th diagonal element of the var-covariance matrix.

Let  $p_{(1)} \leq p_{(2)} \leq p_{(n)}$  be the ordered observed  $p$ -values. Given the false discovery rate  $q$ , we define  $k = \max\{j : p_{(j)} \leq \frac{j}{n}q\}$  and reject  $H_{(1)}^0, \dots, H_{(k)}^0$ . If there exists a  $j$  such that

$p_{(j)} \leq \frac{j}{n}q$ , we reject the null hypothesis and conclude that there is a statistically significant change of temperature in the data center.

Figure 5 reports one significant change detected under the proposed framework with  $q = 0.05$  used in the FDR test. Comparing Figure 5(b) with Figure 5(a), it is clear that the forecasted temperature map appears significantly different from the temperature distribution of the computer model output at  $t = 171$ . The FDR test also confirms that 9 out of 35 testings claim to reject the null hypothesis. In other words, at these nine sensor locations, highlighted as solid dots in Figure 5(c), the temperature forecasts based on the conditions at  $t = 0$  are significantly different from the computer model output based on sensor data at  $t = 171$ . To verify the change in temperature distribution around the time point  $t = 171$ , Figure 5(d) shows the time series of the temperature measurements at the sensor location circled in Figure 5(c). There clearly is a large jump in the time series, which also occurs at the other eight sensors located at the bottom of the inlet side. As a matter of fact, at  $t = 171$ , there was a change of around 16,700 cubic feet per minute in the amount of air supplied by the air conditioning units. Once a significant change is detected, requests of reexecuting the computer models can be automatically processed via the energy management system. The procedure will then follow what is summarized in Figure 3.

## 6. DISCUSSION

In this work, we proposed an online updating strategy to achieve three important goals in data center thermal management: (i) assess the deviance of a dynamic temperature process (monitored by real-time thermal sensors) from a base temperature map (generated from a computer model); (ii) obtain real-time updates of high-resolution temperature maps by adjusting the dynamic deviance; and (iii) issue forecasts to detect changes and automatically schedule the computer model to reexecute. In the proposed framework, a dynamic Gaussian process model is developed to leverage the sensor data that are dense in time but sparse in space to update the high-resolution computer model output in real-time. To achieve online updating, we cast the dynamic Gaussian process model to a state-space representation, which enables us to use the Kalman filter as a computational tool. Moreover, the parameters in the dynamic Gaussian process model are updated in an online fashion through the stochastic projected gradient method.

The proposed method assumes a separable spatio-temporal dependence structure. One future research direction is to consider a nonseparable spatial-temporal dependence by, for example, allowing autoregressive structure to vary over spatial locations. Moreover, a more realistic model assumption is that measurement error variances differ for each sensor, which will improve the detection accuracy and reduce false alerts. These extensions can be accommodated into the proposed method through a Bayesian hierarchical modeling approach and online learning can be achieved via a particle filtering algorithm (Doucet, Godsill, and Andrieu 2000; Storvik 2002).

Another interesting future research topic is to investigate the estimation consistency for model parameters. However, deriving consistency results for parameters in spatial-temporal mod-

els is challenging in general. To our best knowledge, if one only considers spatial Gaussian process, some consistency results (Zhang 2004; Kaufman, Schervish, and Nychka 2008) can be established under the Matern kernel, of which the exponential spatial covariance model used in our work is a special case. On the other hand, if we only consider the temporal autoregressive model, there are also several theoretical works (Brockwell, Davis, and Trindade 2004; Basu and Michailidis 2015) on the consistency of estimated autoregressive coefficients. Under the state-space model representation, the consistency of autoregressive coefficients and covariance matrix has been studied in Anderson et al. (1969). However, they only focused on least-square estimation rather than the likelihood estimation used in the proposed method. Also the covariance matrix in their study is unstructured while the proposed model assumes a parametric spatial covariance model. Additionally, in our proposed method, both autoregressive coefficients and parameters in the covariance structure are estimated simultaneously, which makes the study of consistency even more challenging.

The forecast outputs from the proposed method can be used for online change detection. To illustrate this capability, in this study, we adopt the false discovery rate as a multiple testing approach. It will be interesting to fully investigate other testing methods to possibly improve the detection accuracy in terms of false positive/negative rate.

Finally, Figure 5(a)–5(b) demonstrates a clear edge discontinuity in the temperature map. It will be an interesting research topic to explore various methods in image processing (Qiu 1998, 2005, 2007; Qiu and Mukherjee 2010) to properly account for such edge discontinuities.

Although motivated by online updating and scheduling of computer models for data center thermal management, the proposed method is by no means restricted to the application of real-time thermal management in data centers. It can be applied for updating other high-resolution but computationally expensive models with real-time but sparsely sampled observations.

## SUPPLEMENTARY MATERIALS

The supplementary material for this article contains the following: (1) details of the recursive estimation algorithm for the Kalman filter technique discussed in Section 3.1; (2) plots corresponding to the simulation study presented in Section 4; (3) proofs of Theorem 1 (Section 2), Theorem 2 (Section 3.2), and Theorem 3 (Section 3.3); and R code for implementation of the proposed method.

## ACKNOWLEDGMENTS

This work was prepared as an account of work sponsored by an agency of the United States government (Grant Number DE-EE0002897). Neither the United States government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represented that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or any agency thereof.

[Received September 2013. Revised July 2015.]

## REFERENCES

- Anderson, W., Kleindorfer, G., Kleindorfer, P. R., and Woodroffe, M. (1969), "Consistent Estimates of the Parameters of a Linear System," *The Annals of Mathematical Statistics*, 40, 2064–2075. [481]
- Basu, S., and Michailidis, G. (2015), "Regularized Estimation in Sparse High-Dimensional Time Series Models," *The Annals of Statistics*, 43, 1535–1567. [481]
- Bayarri, M., O Berger, J., Paulo, R., Sacks, J., Cafeo, J., Cavendish, J., Lin, C., and Tu, J. (2007), "A Framework for Validation of Computer Models," *Technometrics*, 49, 138–154. [473]
- Benjamini, Y., and Hochberg, Y. (1995), "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing," *Journal of the Royal Statistical Society, Series B*, 57, 289–300. [480]
- Benjamini, Y., and Yekutieli, D. (2001), "The Control of the False Discovery Rate in Multiple Testing Under Dependency," *Annals of Statistics*, 29, 1165–1188. [480]
- Berrocal, V. J., Gelfand, A. E., and Holland, D. M. (2010a), "A Bivariate Spatio-Temporal Downscaler Under Space and Time Misalignment," *Annals of Applied Statistics*, 4, 1942–1975. [473]
- (2010b), "A Spatio-Temporal Downscaler for Outputs from Numerical Models," *Journal of Agricultural, Biological, and Environmental Statistics*, 15, 176–197. [473]
- (2012), "Space-Time Data Fusion Under Error in Computer Model Output: An Application to Modeling Air Quality," *Biometrics*, 3, 837–848. [473]
- Bottou, L. (2010), "Large-Scale Machine Learning with Stochastic Gradient Descent," in *Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT'2010)*, pp. 177–187. [475]
- Brockwell, P. J., Davis, R. A., and Trindade, A. A. (2004), "Asymptotic Properties of Some Subset Vector Autoregressive Process Estimators," *Journal of Multivariate Analysis*, 90, 327–347. [481]
- Cressie, N., and Wikle, C. K. (2011), *Statistics for Spatio-Temporal Data* (1st ed.), Hoboken, NJ: Wiley. [473]
- Doucet, A., Godsill, S., and Andrieu, C. (2000), "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering," *Statistics and Computing*, 10, 197–208. [473,481]
- Durbin, J., and Koopman, S. J. (2001), *Time Series Analysis by State Space Methods*, New York: Oxford University Press. [474]
- Fonseca, T. C. O., and Steel, M. F. J. (2011), "A General Class of Nonseparable Space-Time Covariance Models," *Environmetrics*, 22, 224–242. [473]
- Gneiting, T. (2002), "Nonseparable, Stationary Covariance Functions for Space-Time Data," *Journal of the American Statistical Association*, 2, 590–600. [473]
- Gneiting, T., Genton, M. G., and Guttorp, P. (2007), "Geostatistical Space-Time Models, Stationarity, Separability and Full Symmetry," in *Statistical Methods for Spatio-Temporal Systems*, eds. B. Finkenstadt, L. Held, and V. Isham, Boca Raton, FL: Chapman & Hall/CRC, pp. 151–175. [473]
- Gneiting, T., and Schlather, M. (2002), "Space-Time Covariance Models," in *Encyclopedia of Environmetrics* (Vol. 4), eds. A. El-Shaarawi and W. Piegorisch, Chichester: Wiley, pp. 2041–2045. [473]
- Guillas, S., Bao, J., Choi, Y., and Wang, Y. (2008), "Statistical Correction and Downscaling of Chemical Transport Model Ozone Forecasts Over Atlanta," *Atmospheric Environment*, 42, 1338–1348. [472]
- Hamann, H., van Kessel, T., Iyengar, M., Chung, J.-Y., Hirt, W., Schappert, M., Claassen, A., Cook, J., Min, W., Amemiya, Y., López, V., Lacey, J., and O'Boyle, M. (2009), "Uncovering Energy Efficiency Opportunities in Data Centers," *IBM Journal of Research and Development*, 53, 1–10:12. [477]
- Hosking, J. (1980), "The Multivariate Portmanteau Statistic," *Journal of the American Statistical Association*, 75, 602–608. [480]
- Kalman, R. E. (1960), "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME Journal of Basic Engineering*, 82, 35–45. [474]
- Kaufman, C. G., Schervish, M. J., and Nychka, D. W. (2008), "Covariance Tapering for Likelihood-Based Estimation in Large Spatial Data Sets," *Journal of the American Statistical Association*, 103, 1545–1555. [481]
- Kennedy, M., and O'Hagan, A. (2001), "Bayesian Calibration of Computer Models" (with discussion), *Journal of the Royal Statistical Society, Series B*, 63, 425–464. [473]
- Liu, Z., Le, N., and Zidek, J. V. (2008), "Combining Measurements and Physical Model Outputs for the Spatial Prediction of Hourly Ozone Space-Time Fields," Technical Report #239, Department of Statistics, The University of British Columbia. [472]
- López, V., and Hamann, H. (2011), "Heat Transfer Modeling in Data Centers," *International Journal of Heat and Mass Transfer*, 54, 5306–5318. [472,477]
- Mardia, K., Goodall, C., Redfern, E., and Alonso, F. (1998), "The Krige Kalman Filter," *Test*, 7, 217–282. [473]
- McMillan, N., Holland, D. M., Morara, M., and Feng, J. (2010), "Combining Numerical Model Output and Particulate Data Using Bayesian Space-Time Modeling," *Environmetrics*, 21, 48–65. [472,473]
- McQuarrie, A. D., and Tsai, C.-L. (1998), *Regression and Time Series Model Selection*, River Edge, NJ: World Scientific. [477]
- Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A. (2009), "Robust Stochastic Approximation Approach to Stochastic Programming," *SIAM Journal of Optimization*, 19, 1574–1609. [474]
- Nobre, A., Sanso, B., and Schmidt, A. (2011), "Spatially Varying Autoregressive Processes," *Technometrics*, 53, 310–321. [473]
- Poole, D., and Raftery, A. (2000), "Inference for Deterministic Simulation Models: the Bayesian Melding Approach," *Journal of the American Statistical Association*, 54, 1244–1255. [472]
- Qiu, P. (1998), "Discontinuous Regression Surfaces Fitting," *The Annals of Statistics*, 26, 2218–2245. [481]
- (2005), *Image Processing and Jump Regression Analysis*, New York: Wiley. [481]
- (2007), "Jump Surface Estimation, Edge Detection, and Image Restoration," *Journal of the American Statistical Association*, 102, 745–756. [481]
- Qiu, P., and Mukherjee, P. (2010), "Edge Structure Preserving Image Denoising," *Signal Processing*, 90, 2851–2862. [481]
- Raftery, A., Givens, G., and Zeh, J. (1995), "Inference from a Deterministic Population Dynamics Model for Bowhead Whales" (with discussion), *Journal of the American Statistical Association*, 90, 402–430. [472]
- Rodrigues, A., and Diggle, P. (2010), "A Class of Convolution-Based Models for Spatio-Temporal Processes with Non-Separable Covariance Structure," *Scandinavian Journal of Statistics*, 37, 553–567. [473]
- Shumway, R., and Stoffer, D. S. (2000), *Time Series Analysis and Its Applications*, New York: Springer. [474]
- Stein, M. L. (2005), "Space-Time Covariance Functions," *Journal of the American Statistical Association*, 100, 310–321. [473]
- Storvik, G. (2002), "Particle Filters for State-space Models with the Presence of Unknown Static Parameters," *IEEE Transactions on Signal Processing*, 50, 281–289. [473,481]
- Zhang, H. (2004), "Inconsistent Estimation and Asymptotically Equal Interpolations in Model-Based Geostatistics," *Journal of the American Statistical Association*, 99, 250–261. [481]