Energy-efficient Model Inference in Wireless Sensing: Asymmetric Data Processing

Paul G. Flikkema
EECS Department
Northern Arizona University
Flagstaff, AZ USA
Email: paul.flikkema@nau.edu

***Abstract*—The ultimate product of distributed sensing is normally a model that describes the data or a set of processes for which the data is an observation. The sensor network itself affects the collected data, often due to the need to conserve deployment costs, including energy provisioning. These effects include data compression and transmission censoring in addition to the typical noise and distortion of signal transduction. This paper describes a framework for performing data and model inference that incorporates network effects into the sophisticated data/model inference techniques used in environmental and ecological field studies. Both the monitored processes and the effects of network data gathering tend to be strongly non-linear and, with uncertainty arising at different levels, hierarchical. Hence closed-form solutions for estimation are beyond reach. Hierarchical Bayesian modeling can jointly capture models and the effects of compression/censoring, and Markov chain Monte Carol (MCMC) simulation at the data center can form posterior estimates. Integral to the inference are estimates of the uncertainty of data and parameters. As the model evolves at the data center, continuously updated estimates of uncertainty can drive adaptation of source/channel coding and data transmission policies at the sensor nodes. As an example application, this paper considers strongly asymmetric data processing to minimize computational complexity and energy cost at the sensor nodes and exploit abundant data center resources. Using an example of compression via simple requantization of the data, we show that MCMC-based inference can yield good performance even at substantial compression rates.**

I. Introduction

In many potential applications, sensor networks will be used to construct predictive models, whether short-term (e.g., forecasting the spread of a gaseous plume) or long-term (e.g., modeling the effect of climate variations on forest health). Since the model will be used for prediction, point estimates are of limited utility, and perhaps dangerous; this motivates inference that incorporates assessment and management of model uncertainty. At the same time, energy constraints impose the need to compress data and censor transmissions, which can degrade model fidelity. These conflicting needs must be achieved in the context of distortion and noise in transduction of physical processes that can be strongly nonlinear with hierarchical sources of uncertainty.

In many important wireless sensor network applications, wireless communication is responsible for the majority of sensor node energy use, so data compression methods have played a central role in most strategies. These include distributed source coding [1], compressive sensing [2], quantization for estimation and classification [3], and joint coding and transmission control [4]. Deshpande et al. [5] suggested a query-oriented technique that used a one-step Markov multivariate Gaussian data model and Kalman filtering-based estimation. In [6], the sensor node infers an ARMA model that is used to respond to queries. Schemes for transmission censoring that use identical algorithms at the sensing node and the data center include [7], [8]; these also use a time series model. A technique that combines Bayesian inference with randomized transmission control is described in [9]. Statisticians and ecosystems modelers have been developing powerful inferential approaches and applying them to state and parameter estimation in models that capture multiple sources of uncertainty [10], [11]. These approaches combine hierarchical Bayesian models with Markov chain Monte Carlo (MCMC) methods and are appropriate for nonlinear systems with embedded stochasticity. Finally, there has been a growing awareness of the power of dynamic data-driven application systems (DDDAS) [12], [13] approaches that dynamically steer data acquisition and transmission as a function of simulation-based model responses to assimilated data.

Much previous work involving the effect of the network on inference has been on "flat" models, where the data model is a joint density over the measured samples or a spatio-temporal ARMA time series, and the inference results in point estimates. This paper outlines a hierarchical Bayesian framework for state/parameter inference that naturally handles common but difficult problems: (i) the rigorous incorporation of prior knowledge and (ii) hierarchical modeling to manage multiple sources of uncertainty and nesting of effects. For example, readings of multiple transducers at a node may suffer from similar distortion effects, and multiple sensor nodes could measure identical or similar phenomena. Perhaps most importantly, many useful compression and transmission censoring algorithms are inherently non-linear, making classical model inference approaches ineffective.

II. Joint Modeling: Environment and Networked Sensing

A critical challenge in wireless sensing is moving beyond the goal of simply acquiring data; whatever data is acquired should have value in achieving the goal of the network deployment, and this is often construction of predictive models for science, policy, or responsive action.

Networked sensing couples sensed physical processes with the processes of sensing and wireless communication. Physical process modeling involves distortion and noise in measurements, and wireless communication involves coding/compression and transmission control—censoring of entire data transmissions to the data center—to conserve energy.

Here, we describe a framework that encompasses all important components of the network and its embedding environment: a parametric model for the processes of interest in the embedding environment, including spatial, temporal, and spatio-temporal dynamics and covariates; measurement bias and noise; data compression and transmission censoring; communication reliability in multi-hop data transport, including multiple layers of error detection and correction coding; and prior knowledge of both the environment and the network.

The physical processes and network processing embed uncertainty at all levels, lending themselves to probabilistic modeling. Hierarchical Bayesian modeling can jointly capture environmental models and the effects of compression and censoring, and MCMC-based processing at the data center can form posterior estimates. This framework, illustrated in Figure 1, admits joint modeling of the environmental and network data gathering processes and can incorporate hierarchical uncertainty, in contrast to classic models where all uncertainties are shoe-horned into either measurement or process noises.

As shown in Figure 1, the model evolves as data arrives at the data center, where continuously updated estimates of uncertainty can drive adaptation of source coding and transmission control policies at the sensor nodes. These in turn drive delivery of data to the simulation at only the precision needed, minimizing energy consumption.

The focus of this work is on understanding how to optimize source coding and transmission control algorithms in an integrated environment/network model that enables the trade-off of model parameter certainty and network energy conservation. We are investigating algorithms that generalize compression and transmission control in two ways. First, the decoder generates random quantities (not point estimates) that enable quantification and management of uncertainty. Secondly, the act of decoding becomes a natural component of model inference, and benefits from prior knowledge about the network and embedding environment.

III. Model

We consider a dynamic model

$$\begin{aligned} x_{t+1} &= g_t(x_t, u_{t+1}, \eta_{t+1}; \theta) \\ y_t &= \xi(x_t) = q(Dx_t + \omega_t(\theta)) \end{aligned} \quad (1)$$

that is parameterized by θ (see Figure 1). x_t is in general a length-n vector, representing a state-space model of temporal processes or, more generally, discrete-space spatio-temporal processes. u_t is a vector-valued input process, and η_t is a process noise that can absorb uncertainties due to unmodeled effects. Quantization in analog-to-digital conversion is modeled by $q(\cdot)$, and y_t is the observation made by the sensor(s); here D is $m \times n$, $m \leq n$, since there are often unobserved state variables. Observation noise is represented by ω_t, which is also parameterized by θ.

In the sensor network, y_t is subject to compression and censoring, so that $\tilde{y}_t = e_t(y_t)$ may be compressed (here we assume lossy compression) or even missing (due to transmission censoring). We refer to $e_t(\cdot)$ simply as *encoding*. This encoding may be very simple or extremely complex. Here we consider the strongly asymmetric case, where the encoding is extremely simple to minimize computational complexity and energy consumption at the encoding node, but the nearly unlimited power of a data center is used to perform inference that accounts for the simple encoding within a much larger effort to infer both state and model parameters.

We assume that the measured data set consists of T consecutive data vectors $y = (y_0, y_1, \cdots, y_{T-1})$. The collection of sample vectors—requantized or censored—at the data center is $\tilde{y}$. For the sake of simplicity, we do not consider the effects of channel errors, so that $z_t = \tilde{y}_t$ in Figure 1. In many implementations, lower layers of the protocol stack provide a nearly perfect channel. Where this is not the case, another conditional probability can be easily added to the model (see (4)) to account for symbol, word, or message errors. Our goal is an estimate of the state (data) $x = (x_0, x_1, \cdots, x_{T-1})$ and the parameter vector θ.

Let $f(\cdot)$ denote a probability mass or density function (we assume densities below without loss of generality). The likelihood is

$$f(\tilde{y}, y, x|\theta) \propto f(\tilde{y}|y, x, \theta). \quad (2)$$

At this point a conventional analysis could proceed as follows: marginalize $f(\tilde{y}|y, x, \theta)$ over y to obtain $f(\tilde{y}|x, \theta)$ and then obtain the "point" maximum-likelihood (ML) state/parameter estimates

$$\{x_{\mathrm{ML}}, \theta_{\mathrm{ML}}\} = \arg\max_{\{x,\theta\}} f(\tilde{y}|x, \theta).$$

However, a Bayesian approach has the advantages of (i) permitting the incorporation of a prior density $f(\theta)$ on the parameter vector, and, building on this, (ii) the ability to compute posterior densities that quantitatively capture the uncertainty of the posterior estimates. The ML point estimates and densities fall out as a special case when non-informative prior densities are used and where the posterior distribution is maximized.

In the Bayesian approach, the full posterior $f(y, x, \theta|\tilde{y})$ can be written as $f(y, x, \theta|\tilde{y}) = f(\tilde{y}, y, x, \theta)/f(\tilde{y})$ so that

$$f(y, x, \theta|\tilde{y}) \propto f(\tilde{y}, y, x, \theta). \quad (3)$$

Thus the posterior density is proportional to the joint distribution of $(\tilde{y}, y, x, \theta)$, which, noting that the ordered set $\{\theta, x, y, \tilde{y}\}$ is Markov, can be simplified to obtain

$$f(y, x, \theta|\tilde{y}) \propto f(\tilde{y}|y) f(y|x) f(x|\theta) f(\theta). \quad (4)$$

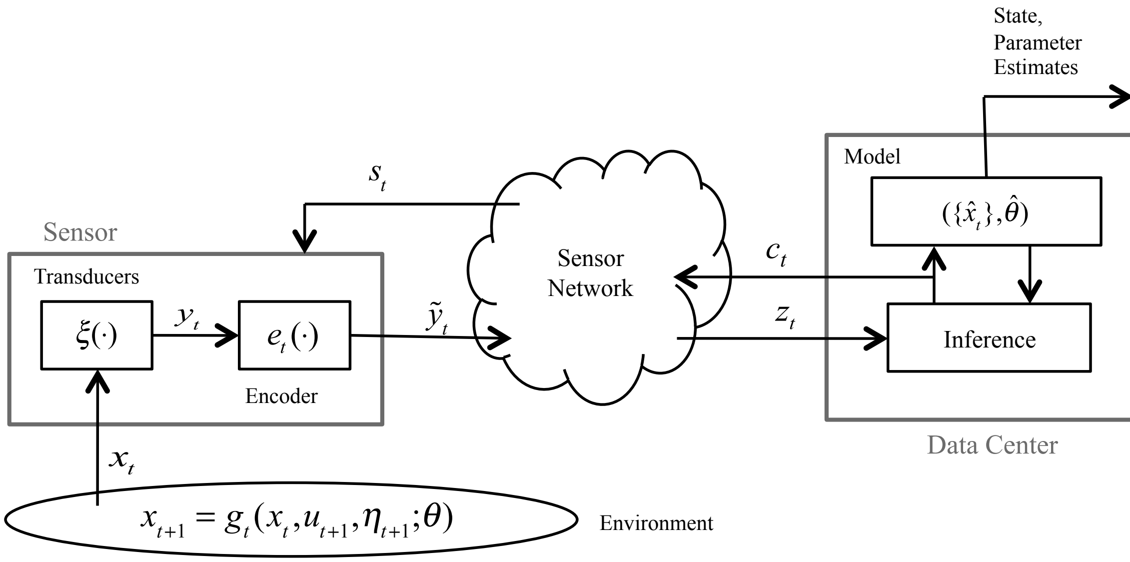

Fig. 1. Block diagram of asymmetric data processing. Processes x_t that capture the state of the environment or its covariates are transduced, encoded using simple algorithms, and sent to a data center, accumulating uncertainty at each step. Model-based inference is performed on the received data that is a corrupted version of the encoded vector data stream $\tilde{y}_t$. Adaptation of compression and transmission censoring is steered via commands c_t.

Here $f(\tilde{y}|y)$ is the encoding, $f(y|x)$ models the measurement process, and $f(\theta)$ is the prior distribution on the parameters, which may be non-informative. Note that $f(x|\theta)$ is the state-vector sequence as a function of θ, viz., x has a distribution that is conditionally dependent on θ. The posterior distributions of x and θ can be computed by marginalizing (4). However, in most cases of interest either the model $g_t(\cdot;\theta)$ is non-linear and time-varying, the encoding $f(\tilde{y}|y)$ is non-linear, or both. Hence $f(y, x, \theta|\tilde{y})$ cannot be computed analytically. Straightforward numerical computation and Monte Carlo simulation approaches are also infeasible since x (and thus y and $\tilde{y}$) typically has high dimensionality.

IV. Inference of State and Parameters

The Markov chain Monte Carlo technique allows side-stepping the computation of the full posterior by instead efficiently simulating (an approximation to) it. The key idea is that the simulation (aka sampling) explores the full posterior by randomly drawing samples not according to the desired—but unfortunately unknown—posterior, but to a simple proposal distribution.

The approach here can be considered "batch" in the sense that the full state-vector sequence is inferred; this is in contrast to sequential Monte Carlo (SMC) (also known as particle filtering) approaches [14], [15] that primarily target on-line state estimation. Here our focus is on exploiting the computational power of a data center to perform inference on encoded data ingested from the network. For this application, the Gibbs sampler is particularly useful, since it is based on tractable computations involving sampling from proposal distributions that are the conditional distributions involving the data, state-vector sequence, and parameters. These distributions are known as the *full* conditional distributions, i.e., the distribution of one component conditioned on the others. The Gibbs sampling algorithm is a special case of the general Metropolis-Hastings MCMC algorithm in that the sampling or proposal distributions are these full conditionals. Gibbs sampling can be inefficient, but performance enhancements are available [16].

The posterior density is not required; instead a density proportional to it is generated, simplifying the analysis and resulting computation. (Normalization can be performed later if needed.) This allows removal of component densities that do not include the variable of interest, since they are constant scaling factors.

In the interest of brevity, we assume no measurement noise or distortion, so that the state sequence x is encoded directly into $\tilde{y}$. The conditional posterior for x becomes

$$\begin{aligned} f(x|\theta,\tilde{y}) &\propto f(\tilde{y}|x)f(x|\theta)f(\theta) \\ &\propto f(\tilde{y}|x)f(x|\theta). \end{aligned} \tag{5}$$

Similarly, the conditional posterior for θ is

$$\begin{aligned} f(\theta|x,\tilde{y}) &\propto f(\tilde{y}|x)f(x|\theta)f(\theta) \\ &\propto f(x|\theta)f(\theta). \end{aligned} \tag{6}$$

V. Example: State and Parameter Inference on Quantized Data

As a specific example, we consider sensing a discrete data stream x_t that is compressed to $\tilde{y}_t$ by requantization at the sensor node to the highest-order k (of n) bits of each value. This requires almost negligible computation and energy.

The data center infers the process state x_t and its model parameters θ from the compressed data. In general, the inference algorithm running at the data center is very complex, since it generates posterior densities for the data and parameters. To accomplish this, the k bits are not decoded into traditional point representatives (quantization centers); instead, decoding is absorbed into the model inference framework, where MCMC-generated random vectors lead to posterior distributions that embed the requantization loss and fully capture the bias and precision of the parameter estimates.

We simulate an affine relationship whose starting value, drift, and process noise are known up to prior distributions. Let $\mathcal{N}_T(\mu, \Sigma)$ denote a T-variate Gaussian distribution with mean vector μ and covariance matrix Σ. The model is

$$x = Mb + \eta, \tag{7}$$

where now $\theta = (b, \sigma_x^2)$ and $\eta \sim \mathcal{N}_T(0, \sigma_x^2 \boldsymbol{I}_T)$. Also $b = (b_0, b_1)^T$ and $\boldsymbol{M}$ is a design matrix with first column of 1's and second column of $(0, 1, ..., T-1)^T$, and $\boldsymbol{I}_T$ is the $T \times T$ identity matrix. Thus $f(x|\theta) = f(x|b, \sigma_x^2) \sim \mathcal{N}_T(\boldsymbol{M}b, \sigma_x^2 \boldsymbol{I}_T)$.

Applying the result for the general case (3), the full posterior is

$$f(x, b, \sigma_x^2|\tilde{y}) \propto f(\tilde{y}|x) f(x|b, \sigma_x^2) f(b, \sigma_x^2), \tag{8}$$

where $f(\tilde{y}|x)$ represents the encoding $e_t(\cdot)$. From (5), the conditional posterior for x is

$$f(x|b, \sigma^2, \tilde{y}) \propto f(\tilde{y}|x) f(x|b, \sigma_x^2). \tag{9}$$

Using (6) the conditional posteriors for b and σ_x^2 are respectively

$$f(b|x, \sigma_x^2, \tilde{y}) \propto f(x|b, \sigma_x^2) f(b) \tag{10}$$

and

$$f(\sigma_x^2|x, b, \tilde{y}) \propto f(x|b, \sigma_x^2) f(\sigma_x^2), \tag{11}$$

where the parameter vector has been split into two components for Gibbs sampling.

To simplify sampling, we specify conjugate prior distributions so the posteriors have standard distributions. The prior on the data parameters is $f(b) \sim \mathcal{N}_2(\mu_b, \Sigma_b)$. Here we take the mean and variance of b as independent, so Σ_b is diagonal. Since the variance σ_x^2 must be non-negative, its prior $f(\sigma_x^2)$ is distributed according to an inverse-gamma (IG) probability law, so the resulting conditional posterior is also IG.

The encoding step of the requantization at the sensor node results in transmitting the most significant k bits to the data center. The parameter k can be included in the packet, or, to save energy, agreed upon by the node and the data center for some period of time. Requantization at the sensing node is deterministic, a simple mapping to the k-bit word $\tilde{y}_t$ representing the interval containing x_t. However, the data center does not attempt to reconstruct x_t using the mid-point of the quantization interval. Instead, this traditional decoding process is replaced in the Gibbs sampler for x (using (9)) by uniform random draws from that interval, denoted by the density $f(\tilde{y}|x)$, so that the true uncertainty resulting from the encoding is incorporated into the inference of x, b, and σ_x^2. Note that $f(\tilde{y}|x)$ models the requantization based on splitting the interval $[0, 1, \ldots, 2^n - 1]$ into 2^k subintervals of length $2^{(n-k)}$. For example, if $n = 8$ and $k = 4$, then $f(\tilde{y}_t|x_t = X_t)$ is uniform on a sequence of length 16 containing X_t. Clearly, for high compression rates $\frac{n}{k}$, the variance of $f(\tilde{y}|x)$ is correspondingly high. However, this uncertainty can be moderated in the inference for x via the prior $f(x|b, \sigma_x^2)$.

A. MCMC Implementation

Datasets were generated according to the model (7). To focus on the effects of encoding via requantization, the observation noise was set to zero.

The compressed dataset drives a Gibbs sampler implementing round-robin draws from (9), (10), and (11). Direct sampling from (10) and (11) is possible due to their standard (resp. Gaussian and IG) distributions. Sampling from (9) proceeds via an embedded Metropolis-Hastings step required by the nonlinear requantization.

B. Results

While this inference framework yields posterior densities, a simple summary measure of performance is useful to evaluate its performance. Denote a parameter's true value by b_i and the mean of its posterior density by $\bar{b}_i$. Then the (absolute) fractional error of the inference is given by

$$\mathcal{E}_f(b_i) = \left| \frac{\bar{b}_i - b_i}{b_i} \right|. \tag{12}$$

Inference was performed on three datasets with varying b vectors but fixed process noise $\sigma_x^2 = 1$. The prior distribution for b was simplified by using a scalar variance parameter σ_b^2 so that its covariance matrix was $\Sigma_b = \sigma_b^2 \boldsymbol{I}_2$. Two values of this parameter were used; $\sigma_b^2 = 10{,}000$ for the case of an uninformative prior, and $\sigma_b^2 = 1$ for a highly informative prior. The prior for σ_x^2 was fixed for all runs. Gibbs samplers of 60,000 iterations were run twice for each dataset and compared to assess convergence, so that for each compression rate 360,000 samples were obtained. The results were confirmed using two additional 30,000-sample runs for each dataset.

The fractional error for a standard least-squares regression using point representatives for reconstruction was also computed and averaged over the three datasets. Figure 2 shows the effect of requantization-based compression on the average fractional error of process parameter b_0; the improvement over the least-squares estimate with quantization centers is substantial. Note that the curves are not always monotonic: the algorithm appears to benefit at certain compression rates, but monotonicity should increase with the number of datasets. As shown using posterior densities in Figure 3, the MCMC-based technique can be used to precisely quantify model parameter uncertainty.

VI. Conclusion

The overall goal of this work is to design and ultimately optimize data gathering protocols for wireless sensor networks driven by models that capture the monitored environment, the network itself, and prior information. The framework presented here is based on two fundamental ideas: (i) joint

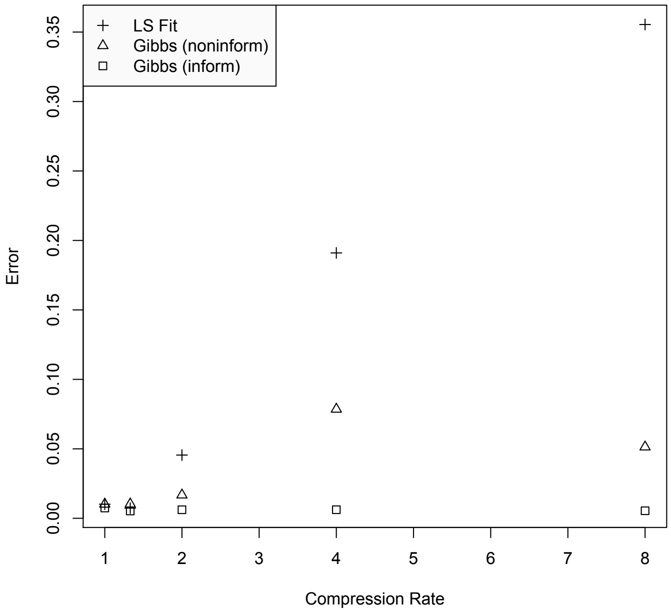

Fig. 2. Fractional error for estimation of model parameter b_0 for least-squares fit and Gibbs samplers with non-informative and informative priors. Higher compression rate means less data sent and correspondingly less energy used.

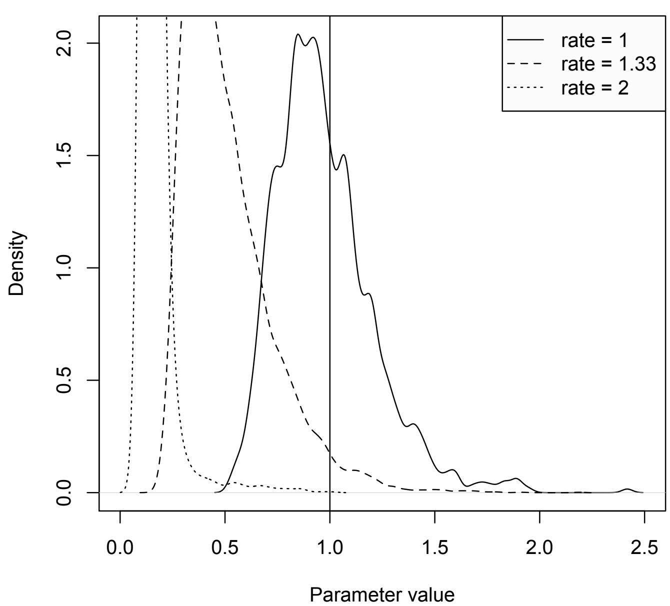

Fig. 3. Posterior density estimates (smoothed with Gaussian kernel) of process noise variance σ_x^2 for different compression rates show convergence as rate decreases. (The true variance value of 1 is shown.)

tracking of the uncertainty in the process modeling and the uncertainty incurred in the processing chain from transduction to inference, and (ii) managing the overall performance of the model/parameter inference via control of energy-intensive network activities. To demonstrate the potential of strongly asymmetric processing, with extremely simple, energy conserving in-network algorithms, the framework was applied to inference from data compressed via requantization. In this case, reconstruction using quantization centers was replaced by inference that captured the true uncertainty resulting from requantization. The performance was compared to traditional least-squares estimation and showed the potential of the inference framework.

Acknowledgment

The author thanks J. S. Clark and A. Gelfand for informative and stimulating discussions. This work was supported by NSF grant CNS-0540414.

References

[1] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," *IEEE Transactions on Information Theory*, vol. 49, no. 3, pp. 626–643, Mar. 2003.

[2] W. U. Bajwa, J. Haupt, A. M. Sayeed, and R. Nowak, "Compressive wireless sensing," in *Proc. 5th Intl. Conf. on Information Processing in Sensor Networks (IPSN '06)*, 2006, pp. 134–142.

[3] R. M. Gray, "Quantization in task-driven sensing and distributed processing," in *Proc. IEEE ICASSP 2006*, 2006, pp. 1049–1052.

[4] S. L. Howard and P. G. Flikkema, "Progressive joint coding, estimation and transmission censoring in energy-centric wireless data gathering networks," in *5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS 2008)*, 2008, pp. 485–490.

[5] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong, "Model-driven data acquisition in sensor networks," in *VLDB '04: Proceedings of the Thirtieth International Conference on Very Large Data Bases*, 2004, pp. 588–599.

[6] D. Tulone and S. Madden, "PAQ: time series forecasting for approximate query answering in sensor networks," in *3rd European Workshop on Wireless Sensor Networks (EWSN'06)*, 2006, pp. 21–37.

[7] C. Olston, J. Jiang, and J. Widom, "Adaptive filters for continuous queries over distributed data streams," in *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, 2003, pp. 563–574.

[8] Y.-A. Le Borgne, S. Santini, and G. Bontempi, "Adaptive model selection for time series prediction in wireless sensor networks," *Signal Processing*, vol. 87, no. 12, pp. 3010–3020, 2007.

[9] G. Hartl and B. Li, "infer: A Bayesian inference approach towards energy efficient data collection in dense sensor networks," in *ICDCS '05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, Washington, DC, USA, 2005, pp. 371–380.

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

[11] B. Carlin, J. Clark, and A. Gelfand, "Elements of hierarchical bayesian inference," in *Hierarchical Modeling for the Environmental Sciences: Statistical Methods and Applications*, J. Clark and A. Gelfand, Eds. Oxford University Press, 2006.

[12] F. Darema, "Dynamic data driven applications systems: A new paradigm for application simulations and measurements," in *ICCS 2004, LNCS 3038*, M. Buback *et al.*, Eds. Springer-Verlag, 2004, pp. 662–669.

[13] P. G. Flikkema, P. K. Agarwal, J. S. Clark, C. Ellis, A. Gelfand, K. Munagala, and J. Yang, "Model-driven dynamic control of embedded wireless sensor networks," in *ICCS 2006, Part III, LNCS 3993*, V. N. Alexandrov *et al.*, Eds. Springer-Verlag, 2006, pp. 409–416.

[14] J. S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *Journal of the American Statistical Association*, vol. 93, pp. 1032–1044, 1998.

[15] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[16] D. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2002.