Centralized active tracking of a Markov chain with unknown dynamics

Mrigank Raman, Ojal Kumar, Arpan Chattopadhyay

***Abstract*—In this paper, selection of an active sensor subset for tracking a discrete time, finite state Markov chain having an unknown transition probability matrix (TPM) is considered. A total of N sensors are available for making observations of the Markov chain, out of which a subset of sensors are activated each time in order to perform reliable estimation of the process. The trade-off is between activating more sensors to gather more observations for the remote estimation, and restricting sensor usage in order to save energy and bandwidth consumption. The problem is formulated as a constrained minimization problem, where the objective is the long-run averaged mean-squared error (MSE) in estimation, and the constraint is on sensor activation rate. A Lagrangian relaxation of the problem is solved by an artful blending of two tools: Gibbs sampling for MSE minimization and an on-line version of expectation maximization (EM) to estimate the unknown TPM. Finally, the Lagrange multiplier is updated using slower timescale stochastic approximation in order to satisfy the sensor activation rate constraint. The on-line EM algorithm, though adapted from literature, can estimate vector-valued parameters even under time-varying dimension of the sensor observations. Numerical results demonstrate approximately 1 dB better error performance than uniform sensor sampling and comparable error performance (within 2 dB bound) against complete sensor observation. This makes the proposed algorithm amenable to practical implementation.**

***Index terms*—** Active tracking, sensor selection, stochastic approximation, Gibbs sampling, on-line expectation maximization.

I. Introduction

Remote estimation of physical processes via sensor observations is an integral part of cyber-physical systems. These estimates are typically fed to some controller in order to control a physical process or system. Typical applications of remote estimation include object tracking, environment monitoring, industrial process monitoring and control, state estimation in smart grid, system identification and disaster management. One key challenge is such remote estimation problems is that the sensors are required to perform high-quality sensing, control, communication, and tracking, but they are constrained in terms of energy and bandwidth availability. Hence, it is necessary to activate only the most informative sensor subset at each time, so that a good compromise is achieved between the fidelity of the estimates and energy/bandwidth usage by sensors.

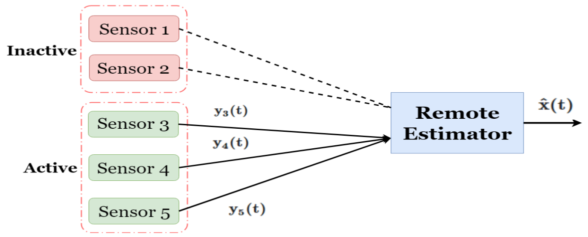
Inactive
Sensor 1
Sensor 2
Active
Sensor 3
Sensor 4
Sensor 5
$\mathbf{y}_3(\mathbf{t})$
$\mathbf{y}_4(\mathbf{t})$
$\mathbf{y}_5(\mathbf{t})$
Remote Estimator
$\hat{\mathbf{x}}(\mathbf{t})$

Figure 1. Remote state estimation with active sensing.

Herein, we consider the problem of designing a *low-complexity* algorithm for dynamically activating an optimal sensor subset, that minimizes the time-averaged MSE under a sensor activation rate constraint reflecting a constraint on the total energy consumed across sensors. The setup is *centralized* in the sense that sensors directly report their observations to a remote estimator; in distributed tracking, there can be multiple nodes, each individually estimating a process, via information exchanged over a multi-hop mesh network. In this paper, we consider *centralized* tracking of a Markov chain with unknown TPM, and solve the problem via a combination of Gibbs sampling, stochastic approximation, and on-line EM. We also work out the problem for the known TPM case. While these algorithms are motivated by solid theoretical consideration, numerical results demonstrate promising MSE performance despite reduced complexity.

A. Literature survey

Active sensor subset selection for process tracking may be either centralized or distributed. In centralized tracking, sensors send observations to a remote estimator which estimates the process. In distributed tracking, a number of nodes, each having a number of sensors, constitute a connected, multihop network; each node estimates the global state using its local sensor observation and the information coming from the adjacent nodes.

There have been considerable recent work on centralized active tracking of a process with several applications; see [1] for application on sensor networks, [2] for mobile crowdsensing, [3] for body-area sensing application and [4] for target tracking application. Even when the process does not have time-variation, calculating the estimation error given sensor observations and finding the best subset of sensors poses some serious technical challenges. To address these challenges, the authors of [1] provided a lower bound on performance, and used a greedy algorithm for subset selection. On the other hand, there have been a number of recent work on centralized active tracking of a time-varying process: [5] for single sensor selection by a centralized controller to track a Markov chain, Markov decision process formulation for sensor subset selection in [6] to track a Markov chain with known TPM, energy aware active sensing [7], existence and structure of active subset selection policy under linear quadratic Gaussian

Mrigank Raman and Ojal Kumar are with the Mathematics department, and Arpan Chattopadhyay is associated with the EE department and the Bharti School of Telecom Technology and Management, IIT Delhi. Email: {mt1170736, mt1170741 }@maths.iitd.ac.in, arpanc@ee.iitd.ac.in. This work was funded by the PDA and PLN6R budget head, IIT Delhi.

(LQG) model [8], active sensing for a linear process with unknown Gaussian noise statistics via Thompson sampling [2], etc. The paper [9] considered the model where an observation is shared across multiple sensors. Recently, there have been a series of work for i.i.d. process tracking: [10] using Gibbs sampling based subset selection for an i.i.d. process with known distribution, [11] for learning an unknown parametric distribution of the process via stochastic approximation (see [12]), and an extended version [13] of these two papers. On the other hand, the authors of [14] have proposed an algorithm for distributed tracking a Markov chain with *known* TPM using tools from stochastic approximation, Gibbs Sampling and Kalman-consensus filter.

B. Our Contribution

Following points are our contribution in the paper:

1) We provide an online learning algorithm for active sensing to track a Markov chain with *unknown TPM*. The unknown TPM is learnt via online EM (see [15]), suitably adapted for variable dimension of observations due to active sensing. Gibbs sampling (see [16]) is used for *low-complexity* sensor activation, and stochastic approximation is used for meeting the sensor activation constraint.
2) An interesting trick to handle variable dimension of observations was to maintain a global library of unknown parameter estimates, and update only the relevant ones in an asynchronous fashion, depending on sensor activation and observation sequence. This idea was absent in the online EM algorithm [15].
3) Another interesting feature of the algorithm which we have proposed is running various iterates in multiple timescales.

C. Organisation

We have organised our paper in the following manner. In Section II, we have described system model. The necessary mathematical background is summarised in Section III. In Section IV we have provided active sensing, state estimation and learning algorithms. Numerical results are provided in Section V and finally we have conclusion in Section VI.

II. System Model

Bold capital and bold small letters will represent matrices and vectors respectively, whereas sets will be represented by calligraphic font throughout this paper.

A. Sensing and observation model

We consider a remote estimation setting as in Figure 1. The set of sensors is $\mathcal{N} = \{1, 2, 3, ..., N\}$. The sensors are used to sense a discrete time process $\{\boldsymbol{x}(t)\}_{t\geq 0}$, which is a time-homogeneous positive recurrent Markov chain with q states. For the sake of mathematical convenience, we denote the j-th state as $\boldsymbol{e_j}$ which is the j-th standard basis (column) vector of length q, with 1 at the j-th coordinate and 0 everywhere else. Hence, the state space of $\{\boldsymbol{x}(t)\}_{t\geq 0}$ becomes $\mathcal{X} := \{\boldsymbol{e_j} : j = 1, 2, \cdots, q\}$. The TPM of $\{\boldsymbol{x}(t)\}_{t\geq 0}$ is denoted by $\boldsymbol{A}'$ (transpose of $\boldsymbol{A}$), which is *unknown* to the remote estimator.

Note that:

$$\boldsymbol{x}(t+1) = \boldsymbol{A}\boldsymbol{x}(t) + \underbrace{\boldsymbol{x}(t+1) - \boldsymbol{A}\boldsymbol{x}(t)}_{\doteq \boldsymbol{w}(t)} \qquad (1)$$

where $\boldsymbol{w}(t)$ is a zero-mean process noise (non-Gaussian).

At time t, let $\boldsymbol{b}(t) \in \{0,1\}^N$ denote the activation status of N sensors; if $b_k(t) = 1$, then the k-th sensor is active, otherwise it is inactive. Let $\boldsymbol{b}_{-k}(t) \in \{0,1\}^{N-1}$ be the same as $\boldsymbol{b}(t)$, except that $b_k(t)$ is removed. Also, let $(\boldsymbol{b}_{-k}(t), 0)$ be the same as $\boldsymbol{b}(t)$ except that $b_k(t) = 0$, and let us assume similar notation for $(\boldsymbol{b}_{-k}(t), 1)$. An active sensor makes an observation and communicates that observation to the remote estimator, whereas an inactive sensor does neither of these. If $b_k(t) = 1$ and if $\boldsymbol{x}(t) = \boldsymbol{e_i}$, then the observation $\boldsymbol{y}_k(t)$ from sensor k follows a Gaussian distribution as $\boldsymbol{y_k}(t) \sim \mathcal{N}(\boldsymbol{\mu}_{k,i}, \boldsymbol{Q}_{k,i}) \in \mathbb{R}^{n_k \times 1}$. Mathematically, we can write the observation coming from sensor i at time t as:

$$\boldsymbol{y_k}(t) = \begin{cases} \boldsymbol{H}_k\boldsymbol{x}(t) + \underbrace{\boldsymbol{v}_k(t)}_{\sim\mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_{k,i})} \ , & b_k(t) = 1, \boldsymbol{x}(t) = \boldsymbol{e_i} \\ \emptyset, & b_k(t) = 0 \end{cases} \qquad (2)$$

where $\boldsymbol{H}_k \doteq [\boldsymbol{\mu}_{k,1} : \boldsymbol{\mu}_{k,2} : \cdots : \boldsymbol{\mu}_{k,q}] \in \mathbb{R}^{n_k \times q}$ is the observation matrix of sensor k, and $\boldsymbol{v}_k(t)$ denotes the Gaussian observation noise at sensor k. We assume that $\boldsymbol{v}_k(t)$ is independent across sensors and time. *We will consider the cases where* $(\boldsymbol{\mu}_{k,i}, \boldsymbol{Q}_{k,i})$ *can be either known or unknown.*

The collection of observations $\{\boldsymbol{y}_k(t) : b_k(t) = 1, k \in \mathcal{N}\}$, arranged as a column vector via vertical concatenation, is called $\boldsymbol{y}(t)$. This $\boldsymbol{y}(t)$ is collected by the remote estimator to estimate $\hat{\boldsymbol{x}}(t)$ at each time t. *The estimate* $\hat{\boldsymbol{x}}(t)$ *can be viewed as a belief vector on* $\mathcal{X}$.

B. The optimization problem

Let $\tilde{\pi}_1$ and $\tilde{\pi}_2$ denote two generic rules (deterministic or randomized) for sensor activation and process estimation, respectively. In this paper, we seek to solve the following problem:

$$\begin{aligned} & \min_{\tilde{\pi}_1, \tilde{\pi}_2} \limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}||\hat{\boldsymbol{x}}(t) - \boldsymbol{x}(t)||^2 \\ \text{such that} \quad & \limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}||\boldsymbol{b}(t)||_1 \leq \bar{N} \end{aligned} \qquad (3)$$

By standard Lagrange multiplier theory, problem (3) can be solved by solving the following relaxed problem:

$$\min_{\tilde{\pi}_1, \tilde{\pi}_2} \limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}||(\hat{\boldsymbol{x}}(t) - \boldsymbol{x}(t)||^2 + \lambda||\boldsymbol{b}(t)||_1 (4)$$

under a suitable Lagrange multiplier λ^* such that the inequality constraint is met with equality.

III. Background

In this section, we provide a basic background that will be useful in solving (4).

Gibbs sampling: Let us assume (for the sake of illustration) that $\boldsymbol{x}(t)$ is i.i.d. with known distribution. Then, there exists an optimal $\boldsymbol{b}^*$ such that, using $\boldsymbol{b}(t) = \boldsymbol{b}^*$ over the entire time horizon along with MMSE estimation is optimal for (4). Thus, the problem reduces to:

$$\min_{\boldsymbol{b} \in \{0,1\}^N} \underbrace{f(\boldsymbol{b}) + \lambda ||\boldsymbol{b}||_1}_{\doteq h(\boldsymbol{b})} \tag{5}$$

where $f(\boldsymbol{b})$ is the MSE under sensor activation vector $\boldsymbol{b}$, and $h(\boldsymbol{b})$ is the cost under this activation vector. In order to avoid searching over 2^N possible activation vectors, the authors of [10] used Gibbs sampling for sensor activation. Gibbs sampling generates a Markov chain $\{\boldsymbol{b}(t)\}_{t \geq 0}$ whose stationary distribution is $\pi_\beta(\boldsymbol{b}) \doteq \frac{e^{-\beta h(\boldsymbol{b})}}{\sum_{\tilde{\boldsymbol{b}} \in \{0,1\}^N} e^{-\beta h(\tilde{\boldsymbol{b}})}}$ with a parameter $\beta > 0$ interpreted as the inverse temperature in statistical physics. Note that, $\lim_{\beta \to \infty} \pi_\beta(\boldsymbol{b}^*) = 1$ if the unique minimizer for $h(\cdot)$ is $\boldsymbol{b}^*$. Hence, for sufficiently large β, Gibbs sampling under steady state selects $\boldsymbol{b}^*$ with high probability, and we obtain a near-optimal solution of (4). At any time t, Gibbs sampling randomly selects sensor $k \in \mathcal{N}$ with uniform distribution, and sets $b_k(t) = 1$ with probability $\frac{e^{-\beta h(\boldsymbol{b}_{-k},1)}}{e^{-\beta h(\boldsymbol{b}_{-k},1)} + e^{-\beta h(\boldsymbol{b}_{-k},0)}}$ and $b_k(t) = 0$ otherwise. Then the k-th sensor is activated accordingly, and the activation status of other sensors remain unchanged. Finally, the constrained problem (3) was solved by using a stochastic approximation (see [12]) iteration $\lambda(t+1) = \lambda(t) + a(t)(||\boldsymbol{b}(t)||_1 - \bar{N})$, to satisfy the activation constraint.

Expectation maximization: The expectation maximization (EM) algorithm (see [17]) is used to estimate an unknown parameter $\boldsymbol{\theta}$ from noisy observation $\boldsymbol{y}$ of a random vector $\boldsymbol{x}$ having a parametric distribution with unknown parameter $\boldsymbol{\theta}$. It maintains an iterate $\boldsymbol{\theta}(t)$ at iteration t. In the E step, $\mathbb{E}(\log p(\boldsymbol{x}|\boldsymbol{\theta})|\boldsymbol{y}, \boldsymbol{\theta}(t))$ is computed, and this is maximized over $\boldsymbol{\theta}$ to obtain $\boldsymbol{\theta}(t+1)$ in the M step. It was shown in [17] that, under certain regularity conditions, $\boldsymbol{\theta}(t)$ converges to the set of stationary points such that $\frac{\partial p(\boldsymbol{y})}{\partial \boldsymbol{\theta}} = \boldsymbol{0}$. Later, the authors of [15] proposed one online EM algorithm for hidden Markov model; their model is similar to our process and observation models, except that they do not consider active sensing and consider *scalar* observations of fixed dimension. However, due to active sensing, our problem allows variable dimension of observations, which requires some nontrivial modification of the algorithm of [15].

IV. The GEM algorithm

In this section, we propose an algorithm called GEM (Gibbs Expectation Maximization) to solve (3). Since the algorithm is technically involved, we will first describe the major components and concepts related to the algorithm and finally provide a summarised version of the complete algorithm.

A. Key components of the algorithm

1) Some useful notation:

Symbols	Meaning
$\boldsymbol{b}(t)$	Activation status of N sensors
$\lambda(t)$	Estimate of the cost of a sensor activation
$\hat{\boldsymbol{A}}_t$	Running Estimate of Transpose of Transition Probability Matrix
$h^{(t)}(\boldsymbol{b}), f^{(t)}(\boldsymbol{b})$	Cost and MSE estimates under activation vector $\boldsymbol{b}$
$\hat{\boldsymbol{\mu}}_{k,i}(t), \hat{\boldsymbol{Q}}_{k,i}(t)$	Running Estimate of mean and covariance of $\boldsymbol{y}_k(t)$ where $\boldsymbol{x(t)} = \boldsymbol{e}_i$
$\hat{\boldsymbol{\Lambda}}_t$	Running estimate of the observation matrix $[\boldsymbol{H}'_1 : \boldsymbol{H}'_2 : \cdots : \boldsymbol{H}'_N]$
$\hat{\boldsymbol{\Psi}}_i(t)$	$blkdiag(\hat{\boldsymbol{Q}}_{1,i}(t), \hat{\boldsymbol{Q}}_{2,i}(t), \cdots, \hat{\boldsymbol{Q}}_{N,i}(t))$
$\hat{\boldsymbol{M}}_i^{\boldsymbol{b(t)}}(t), \hat{\boldsymbol{\Psi}}_i^{\boldsymbol{b(t)}}(t)$	Running estimates of those components of $\hat{\boldsymbol{\Lambda}}_t$ and $\hat{\boldsymbol{\Psi}}_i(t)$, that correspond to the active sensors under activation vector $\boldsymbol{b(t)}$

The proposed algorithm maintains running estimates $\hat{\boldsymbol{\mu}}_{k,i}(t)$ and $\hat{\boldsymbol{Q}}_{k,i}(t)$ for $\boldsymbol{\mu}_{k,i}$ and $\boldsymbol{Q}_{k,i}$, respectively. Equivalently, it maintains an estimate $\hat{\boldsymbol{\Lambda}}_t$ of the matrix $[\boldsymbol{H}'_1 : \boldsymbol{H}'_2 : \cdots : \boldsymbol{H}'_N]$ where $\boldsymbol{H}_k \doteq [\boldsymbol{\mu}_{k,1} : \boldsymbol{\mu}_{k,2} : \cdots : \boldsymbol{\mu}_{k,q}]$. The algorithm also maintains q block-diagonal matrices $\{\hat{\boldsymbol{\Psi}}_i(t) : 1 \leq i \leq q\}$ where $\hat{\boldsymbol{\Psi}}_i(t) \doteq blkdiag(\hat{\boldsymbol{Q}}_{1,i}(t), \hat{\boldsymbol{Q}}_{2,i}(t), \cdots, \hat{\boldsymbol{Q}}_{N,i}(t))$ (block diagonal matrix consisting of these N covariance matrix estimates at time t). For an activation vector $\boldsymbol{b}$, we also define another matrix $\hat{\boldsymbol{\Psi}}_i^{\boldsymbol{b}}(t) \doteq blkdiag\{\hat{\boldsymbol{Q}}_{k,i}(t) : 1 \leq k \leq N, b_k = 1\}$ which can be extracted from $\hat{\boldsymbol{\Psi}}_i(t)$. Similarly, we define $\hat{\boldsymbol{\mu}}_i^{\boldsymbol{b}}(t)$ as an estimate of of the column vector $vertcat(\hat{\boldsymbol{\mu}}_{k,i} : 1 \leq k \leq N, b_k = 1)$ (vertical concatenation of these column vectors). Clearly, given $\boldsymbol{x}(t) = \boldsymbol{e}_i$ and activation vector $\boldsymbol{b}(t)$, our algorithm assumes that $\boldsymbol{y}(t) \sim \mathcal{N}(\hat{\boldsymbol{\mu}}_i^{\boldsymbol{b}(t)}(t), \hat{\boldsymbol{\Psi}}_i^{\boldsymbol{b}(t)}(t))$. For a given activation vector b we also define the matrix $\hat{\boldsymbol{M}}^{\boldsymbol{b}}(t) \doteq [\hat{\boldsymbol{\mu}}_1^{\boldsymbol{b}}(t) : \hat{\boldsymbol{\mu}}_2^{\boldsymbol{b}}(t) : \cdots : \hat{\boldsymbol{\mu}}_q^{\boldsymbol{b}}(t)]$. The algorithm also maintains an estimate $\hat{\boldsymbol{A}}_t$ for $\boldsymbol{A}$.

The algorithm also maintains the iterates $h^{(t)}(\boldsymbol{b}) \forall \boldsymbol{b} \in \{0,1\}^N$, $f^{(t)}(\boldsymbol{b}) \forall \boldsymbol{b} \in \{0,1\}^N$, and $\lambda(t)$ (see Section III), as estimates of $h(\boldsymbol{b}) \forall \boldsymbol{b} \in \{0,1\}^N$, $f(\boldsymbol{b}) \forall \boldsymbol{b} \in \{0,1\}^N$, and λ^*, respectively. Estimate of the MSE under activation vector $\boldsymbol{b}$ can be denoted by $f^{(t)}(\boldsymbol{b}) \forall \boldsymbol{b} \in \{0,1\}^N$ at any time t. The quantities $h^{(t)}(\boldsymbol{b}) \forall \boldsymbol{b} \in \{0,1\}^N$ are used as cost in Gibbs sampling at time t to decide the sensor activation set. The Lagrange multiplier $\lambda(t)$ is updated using stochastic approximation so that the activation constraint in (3) satisfies the equality constraint.

We define $\nu_{\boldsymbol{b}}(t) \doteq \sum_{\tau=1}^{t} \mathbb{I}\{\boldsymbol{b}(\tau) = \boldsymbol{b}\}$ as the number of times the activation vector $\boldsymbol{b}$ is used up to time t.

2) Step sizes: The algorithm maintains two non-increasing, positive step size sequences $\{\alpha(t) : t = 0, 1, 2, \cdots\}$ and $\{\gamma(t) : t = 0, 1, 2, \cdots\}$ for running multi-timescale stochastic approximation updates. The step size sequences satisfy the following properties: (i) $\sum_{t=0}^{\infty} s(t) = \infty, s \in \{\alpha, \gamma\}$, (ii) $\sum_{t=0}^{\infty} s^2(t) < \infty, s \in \{\alpha, \gamma\}$, (iii) $\lim_{t \to \infty} \frac{\gamma(t)}{\alpha(t)} = 0$. The first two requirements are standard for stochastic approximation. The third condition is required for timescale separation; the $f^{(t)}(\cdot)$ update uses step-size $\alpha(t)$, and the $\lambda(t)$ update and online EM updates will use step size $\gamma(t)$.

3) Gibbs sampling: At time t, pick a random sensor $j_t \in \mathcal{N}$ uniformly and independently. For sensor j_t, choose $\boldsymbol{b}_{j_t}(t) = 1$ with probability $p_t = \frac{e^{-\beta h^{(t)}(\boldsymbol{b}_{-j_t}(t-1),1)}}{e^{-\beta h^{(t)}(\boldsymbol{b}_{-j_t}(t-1),1)} + e^{-\beta h^{(t)}(\boldsymbol{b}_{-j_t}(t-1),0)}}$ and choose $b_{j_t}(t) = 0$ with probability $(1-p_t)$. For all $k \neq j_t$, we choose $b_k(t) = b_k(t-1)$. Activate sensors according to $\boldsymbol{b}(t)$ and obtain the observations $\boldsymbol{y}(t)$.

4) $\lambda(t)$ update: The Lagrange multiplier is updated as:

$$\lambda(t+1) = [\lambda(t) + \gamma(t)(||\boldsymbol{b}(t)||_1 - \bar{N})]_0^l \tag{6}$$

The iterates are projected onto an interval which is compact namely $[0, l]$ to ensure boundedness, where $l > 0$ is a sufficiently large number. The intuition behind this approach is that, if $||\boldsymbol{b}(t)||_1 > \bar{N}$, then $\lambda(t)$ (the cost of a sensor activation) is increased, and $\lambda(t)$ is decreased otherwise.

5) State estimation: We use a *Kalman-like state estimator* from [6], designed to track a Markov chain. In this algorithm, $\hat{\boldsymbol{x}}_{t+1|t}$ and $\hat{\boldsymbol{x}}_{t|t}$ denote the estimates of $\boldsymbol{x}(t+1)$ and $\boldsymbol{x}(t)$ respectively, given $\{\boldsymbol{y}(0), \boldsymbol{y}(1), \cdots, \boldsymbol{y}(t)\}$; here $\hat{\boldsymbol{x}}_{t|t}$ is basically the final estimate $\hat{\boldsymbol{x}}(t)$ declared by the estimator at time t, and $\hat{\boldsymbol{x}}_{t|t-1}$ is an intermediate estimate at time t. Additionally, $\boldsymbol{\Sigma}_{t|t}$ and $\boldsymbol{\Sigma}_{t|t-1}$ will denote covariance matrices estimates of the estimation and prediction errors $(\hat{\boldsymbol{x}}_{t|t} - \boldsymbol{x}(t))$ and $(\hat{\boldsymbol{x}}_{t|t-1} - \boldsymbol{x}(t))$ respectively. Unlike standard Kalman filter where the observation dimension is fixed, this Kalman-like estimator has variable observation dimension depending on $\boldsymbol{b}(t)$, and the gain and error covariance matrix updates also take into account $\boldsymbol{b}(t)$ at time t.

State Estimation algorithm

Recursion: For each $t \geq 0$, do:

1) $\hat{\boldsymbol{x}}_{t|t-1} = \boldsymbol{A}\hat{\boldsymbol{x}}_{t-1|t-1}$,
   (Comment: State estimate at time t, keeping observations up to time $(t-1)$ in account.)
2) $\mathbf{y}_{t|t-1} = \hat{\boldsymbol{M}}^{\boldsymbol{b}(t)}(t)\hat{\boldsymbol{x}}_{t|t-1}$,
3) $\boldsymbol{\Sigma}_{t|t-1} = diag(\hat{\boldsymbol{x}}_{t|t-1}) - \hat{\boldsymbol{x}}_{t|t-1}\hat{\boldsymbol{x}}'_{t|t-1}$,
   (Comment: Estimate of error covariance matrix at time t, keeping observations up to time $(t-1)$ in account.)
4) $\tilde{\boldsymbol{\Psi}}_t = \sum_{i=1}^q \hat{\boldsymbol{x}}_{t|t-1}(i)\hat{\Psi}_i^{\boldsymbol{b}(t)}$,
   (Comment: Estimate of the observation noise covariance matrix under activation vector $\boldsymbol{b}(t)$, averaged over the belief $\hat{\boldsymbol{x}}_{t|t-1}(.)$ on the state of the process.)
5) $\mathbf{G}_t = \boldsymbol{\Sigma}_{t|t-1}\hat{\boldsymbol{M}}^{\boldsymbol{b(t)}}(t))' * (\hat{\boldsymbol{M}}^{\boldsymbol{b(t)}}(t))\boldsymbol{\Sigma}_{t|t-1}((\hat{\boldsymbol{M}}^{\boldsymbol{b(t)}}(t))' + \tilde{\boldsymbol{\Psi}}_t)^{-1}$,
   (Comment: Kalman gain update.)
6) Compute $\hat{\boldsymbol{x}}_{t|t} = \hat{\boldsymbol{x}}_{t|t-1} + \mathbf{G}_t(\boldsymbol{y}(t) - \boldsymbol{y}_{t|t-1})$, and project it on the probability simplex.
   (Comment: State estimate at time t, keeping observations up to time $(t-1)$ in account. Projection ensures that the estimate is a valid probability belief vector on the state space.)
7) $\boldsymbol{\Sigma}_{t|t} = diag(\hat{\boldsymbol{x}}_{t|t}) - \hat{\boldsymbol{x}}_{t|t}\hat{\boldsymbol{x}}'_{t|t}$.
   (Comment: Estimate of error covariance matrix at time t, keeping observations up to time t in account.)

6) $f^{(t)}(\cdot)$ update: At time t, MSE estimate $f^{(t)}(\cdot)$ (for sensor subset $\boldsymbol{b}(t)$ only) is updated using the following equation:

$$f^{(t+1)}(\boldsymbol{b}) = [f^{(t)}(\boldsymbol{b}) + \mathbb{I}\{\boldsymbol{b} = \boldsymbol{b}(t)\}\alpha(\nu_{\boldsymbol{b}}(t))(\mathrm{Tr}(\boldsymbol{\Sigma}_{t|t}) - f^{(t)}(\boldsymbol{b}))]_0^l \tag{7}$$

7) Online EM for parameter estimation: Online EM requires an initial distribution π for the Markov chain. However, unlike [15], here we have *vector*-valued observations whose dimension change over time. Also, the unknown parameters $\hat{\boldsymbol{\mu}}_i^{\boldsymbol{b}(t)}$ and $\hat{\boldsymbol{\Psi}}_i^{\boldsymbol{b}(t)}$ (known $\hat{\boldsymbol{\mu}}_i^{\boldsymbol{b}(t)}$ and $\hat{\boldsymbol{\Psi}}_i^{\boldsymbol{b}(t)}$ can be handled in a straightforward way) will also have different dimensions for different values of t. This requires an asynchronous update of various components of the unknown parameters, depending on the currently active sensors and their observations.

Since the online EM algorithm is heavy in notation, we have made some explanatory comments in between the steps of the algorithm. See [15] for a detailed understanding.

The algorithm requires the auxiliary functions $\hat{\phi}, \hat{\rho}^{\boldsymbol{A}}, \hat{\rho}^g, \hat{S}^{\boldsymbol{A}}$ and $\hat{\boldsymbol{S}}^g$, and $g_t(\boldsymbol{e}_i, \boldsymbol{y})$. We also define $\boldsymbol{y}^2(t) = blkdiag\{\boldsymbol{y}_k(t)\boldsymbol{y}'_k(t) : 1 \leq k \leq N, b_k(t) = 1\}$. In general, we need $\boldsymbol{y}^d(t)$ in the algorithm where $d \in \{0, 1, 2\}$; we define $\boldsymbol{y}^0(t) = 1(scalar), and \boldsymbol{y}^1(t) \doteq \boldsymbol{y}(t)$.

Online EM algorithm

Input: Initial distribution $\boldsymbol{\pi}$ of $\{\boldsymbol{x}(t)\}_{t\geq 0}$, $\boldsymbol{y}(t)$.
Initialization: Initialize $\hat{\boldsymbol{A}}_0$ $\hat{\Psi}_i(0)$ for all $1 \leq i \leq q$ and $\hat{\Lambda}_0$ randomly and compute, for all $1 \leq i, j, k \leq q$ and $0 \leq d \leq 2$,

$$\hat{\phi}_0(k) = \frac{\pi(k)g_0(\boldsymbol{e}_k, \boldsymbol{y}_0)}{\sum_{k_1=1}^{q} g_0(\boldsymbol{e}_{k'}, \boldsymbol{y}_0)},$$

$$\hat{\rho}_0^{\boldsymbol{A}}(i, j, k) = 0,$$

$$\hat{\rho}_{0,d}^g(i, k) = \delta_{ik}\boldsymbol{y}^d(0)$$

Recursion: For $t \geq 1$, for all $1 \leq i, j, k \leq q$ and $0 \leq d \leq 2$, DO

Approx. Filter Update

$$\hat{\phi}_t(\boldsymbol{e}_k) = \frac{\sum_{k_1=1}^{q} \hat{\phi}_{t-1}(\boldsymbol{e}_{k_1})\hat{A}'_{t-1}(\boldsymbol{e}_{k_1}, \boldsymbol{e}_k)g_{t-1}(\boldsymbol{e}_k, \boldsymbol{y}(t))}{\sum_{k_1=1}^{q}\sum_{k_2=1}^{q} \hat{\phi}_{t-1}(\boldsymbol{e}_{k_1})\hat{A}'_{t-1}(\boldsymbol{e}_{k_1}, \boldsymbol{e}_{k_2})g_{t-1}(\boldsymbol{e}_{k_2}, \boldsymbol{y}(t))}$$

where $g_t(\boldsymbol{e}_i, \boldsymbol{y}) \doteq exp[-(\boldsymbol{y} - \hat{\boldsymbol{\mu}}_i^{\boldsymbol{b}(t)})[\boldsymbol{\Psi}_i^{\boldsymbol{b}(t)}]^{-1}(\boldsymbol{y} - \hat{\boldsymbol{\mu}}_i^{\boldsymbol{b}(t)})^T/2]$

(Comment: Here $\hat{\phi}_t(\cdot)$ is interpreted as an estimate of the steady state probability distribution of the Markov chain given all observations up to time t.)

E-Step

1) $\hat{\rho}_t^{\boldsymbol{A}}(i, j, k) = \gamma(t)\delta_{jk}\hat{r}_t(i|j) + (1 - \gamma(t)) \sum_{k_1=1}^{q} \hat{\rho}_{t-1}^{\boldsymbol{A}}(i, j, k_1)\hat{r}_t(k_1|k)$
2) $\hat{\rho}_{t,d}^g(i, k) = \gamma(t)\delta_{ik}\boldsymbol{y}^d(t) + (1 - \gamma(t)) \sum_{k_1=1}^{q} \hat{\rho}_{t-1,d}^g(i, k_1)\hat{r}_t(k_1|k)$

where $\hat{r}_t(i|j) = \dfrac{\hat{\phi}_{t-1}(\boldsymbol{e}_i)\hat{A}'_{t-1}(\boldsymbol{e}_i, \boldsymbol{e}_j)}{\sum\limits_{i_1=1}^{q} \hat{\phi}_{t-1}(\boldsymbol{e}_{i_1})\hat{A}'_{t-1}(\boldsymbol{e}_{i_1}, \boldsymbol{e}_j)}$.

(Comment: $\hat{\rho}_t^{\boldsymbol{A}}(\cdot,\cdot,\cdot)$) and $\hat{\rho}^g_{t,d}(\cdot,\cdot)$ together constitute the expectation of a sufficient statistic for the expected log-likelihood involved in the E step. The sufficient statistic is updated over time using Bayes' theorem. Details can be found in [15].)

M-Step

1) $\hat{S}_t^{\boldsymbol{A}}(i,j) = \sum\limits_{k_1=1}^{q} \hat{\rho}_t^{\boldsymbol{A}}(i,j,k_1)\hat{\phi}_t(\boldsymbol{e}_{k_1})$
2) $\hat{\boldsymbol{A}}'_t(\boldsymbol{e}_i, \boldsymbol{e}_j) = \dfrac{\hat{S}_t^{\boldsymbol{A}}(i,j)}{\sum\limits_{j_1=1}^{q} \hat{S}_t^{\boldsymbol{A}}(i,j_1)}$
   (Comment: Update for the estimate of the transition probability matrix.)
3) $\hat{S}^g_{t,d}(i) = \sum\limits_{k_1=1}^{q} \hat{\rho}^g_{t,d}(i,k_1)\hat{\phi}_t(\boldsymbol{e}_{k_1})$
4) $\hat{\mu}_i^{\boldsymbol{b}(t)} = \dfrac{\hat{S}^g_{t,1}(i)}{\hat{S}^g_{t,0}(i)}$
   (Comment: Update for the estimate of the sensor observation mean for subset $\boldsymbol{b}(t)$.)
5) $\hat{\Psi}_i^{\boldsymbol{b}(t)} = \dfrac{\hat{\boldsymbol{S}}^g_{t,2}(i)}{\hat{S}^g_{t,0}(i)} - \hat{\mu}_i^{\boldsymbol{b}(t)}(\hat{\mu}_i^{\boldsymbol{b}(t)})'$
   (Comment: Update for the estimate of the sensor observation covariance for subset $\boldsymbol{b}(t)$.)
6) For every active sensor k such that $b_k(t) = 1$ and for all $1 \leq i \leq q$, modify the values of $\hat{\boldsymbol{\mu}}_{k,i}(t)$ in $\hat{\boldsymbol{\Lambda}}_{\boldsymbol{t-1}}$ and $\hat{\boldsymbol{Q}}_{k,i}(t)$ in $\hat{\boldsymbol{\Psi}}_i(t-1)$ accordingly.

B. The complete algorithm

1) GEM algorithm: The complete GEM algorithm is outlined below.

```
The Complete GEM Algorithm
Input: $\{\alpha(t)\}_{t\geq 0}$, $\{\gamma(t)\}_{t\geq 0}$, $\boldsymbol{\pi}$, $N$, $\beta$
Initialization: Initialise $\boldsymbol{b(0)}$, $\lambda(0) \geq 0$, $\hat{\boldsymbol{A}}_0$, $\hat{\boldsymbol{\mu}}_{k,i}(0)$ and
$\hat{\boldsymbol{Q}}_{k,i}(0)$ for all $1 \leq k \leq N, 1 \leq i \leq q$,
    $\hat{\boldsymbol{x}}_{0|-1} = \hat{\boldsymbol{x}}_{0|0} = \boldsymbol{\pi}$.
Recursion: For all $t \geq 0$, DO:

    1) Use Gibbs Sampling described in Section IV-A3 to attain $\boldsymbol{b}(t)$, activate sensors accordingly, and collect the corresponding observations $\boldsymbol{y}(t)$.
    2) Update $\lambda(t)$ as in Section IV-A4.
    3) Compute $\hat{\boldsymbol{x}}(t) = \hat{\boldsymbol{x}}_{t|t}$ as in Section IV-A5, by running the Kalman-like algorithm.
    4) Calculate $f^{(t)}(\boldsymbol{b}(t))$ as in Section IV-A6.
    5) Calculate $h^{(t+1)}(\boldsymbol{b}) = f^{(t+1)}(\boldsymbol{b}) + \lambda(t+1)||\boldsymbol{b}||_1$
    6) Compute the estimates $\hat{\boldsymbol{A}}_t$, $\hat{\boldsymbol{\mu}}_{k,i}(t)$ and $\hat{\boldsymbol{Q}}_{k,i}(t)$ for all $1 \leq k \leq N, 1 \leq i \leq q$, by using online EM as in Section IV-A7.
```

2) Discussion:

- The GEM algorithm runs in multiple timescales (see [12]). Gibbs sampling corresponds to the fastest timescale, and the $\lambda(t)$ update and online EM run in the slowest timescale. Timescale separation is ensured by $\lim_{t\to\infty} \frac{\gamma(t)}{\alpha(t)} = 0$.
- $f^{(t)}(\cdot)$ update involves asynchronous stochastic approximation for various sensor subsets.
- Projection on $[0, l]$ is done to make sure that all iterates remain bounded.

3) Computational complexity of GEM for every time t: At each time, Gibbs sampling and $\lambda(t)$ update require $\mathcal{O}(1)$ computations. Updating $f^{(t)}(\cdot)$ requires $\mathcal{O}(q)$ computations for trace calculation. The approximate filter update step for all states will require $\mathcal{O}(qN^3 \max_{1\leq k\leq N}\{n_k^3\})$ computations due to the matrix inversion involved in $g_t(\cdot,\cdot)$ calculation. The first step in E step requires $\mathcal{O}(q^4)$ computations. Computational complexity of other steps of online EM as well as state estimation is dominated by these two steps and hence the computational complexity of online EM at each time is $\mathcal{O}(qN^3 \max_{1\leq k\leq N}\{n_k^3\} + q^4)$, where n_k is the dimension of $\boldsymbol{y}_k(t)$ as defined before. The function $h^{(t)}(\cdot)$ needs to be computed only for three vectors as required by Gibbs sampling, and hence this requires $\mathcal{O}(1)$ computations.

V. Numerical results

We consider number of sensors $N = 20$, number of states $q = 10$, activation constraint $\bar{N} = 5$, inverse temperature $\beta = 10$, $\alpha(t) = \frac{1}{t^{0.7}}$, $\gamma(t) = \frac{1}{t^{0.8}}$ and $\lambda(0) = 0.1$. The TPM $\boldsymbol{A}'$ is chosen randomly and then the rows are normalized to obtain a stochastic matrix. The quantities $\{\boldsymbol{\mu}_{k,i}, \boldsymbol{Q}_{k,i}\}_{1\leq k\leq N, 1\leq i\leq q}$ are also chosen randomly. We have not considered larger value of N because it is unlikely (even in current cyber-physical systems) that so many sensors estimate a single physical process, especially when the observation from each sensor is a vector.

Under this setting, we compare the performance of the following six algorithms:

1) GEM-K: Here the observation mean and covariances are known (i.e., K), but TPM is unknown.
2) GEM-UK: Here observation covariances are known, but observation mean and TPM are unknown (i.e., UK).
3) GEM-FO: This is GEM Algorithm with full observation. Here all the sensors are always active. The observation mean and covariances are known, but the TPM is unknown.
4) GEM-U: This is GEM Algorithm with uniform random sampling of sensors: at each time t, a sensor is activated independently with probability $\frac{\bar{N}}{N}$. The observation mean and covariances are known, but the TPM is unknown.
5) GEM-FI: This is GEM with full information of the TPM, observation mean and covariances.
6) GEN: Here GEN stands for genie. At time t, the estimator perfectly knows $\boldsymbol{x}(t-1)$, but no observation is available from sensors. In this case, the MSE will be the limiting variance of $\boldsymbol{x}(t)$ given $\boldsymbol{x}(t-1)$.

A. Convergence of the algorithms

Figure 2 shows the convergence of $\hat{\boldsymbol{A}}_t$ to $\boldsymbol{A}$ for GEM-K. Similarly, we have noticed that $\hat{\boldsymbol{A}}_t$ converges to $\boldsymbol{A}$ for GEM-FO and GEM-U. However, the TPM estimate does not

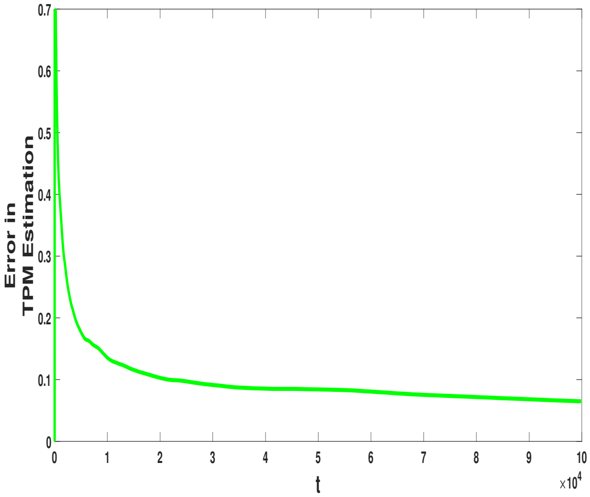

Figure 2. Variation of the TPM estimation error $\frac{1}{t}\sum_{\tau=1}^{t}||\boldsymbol{A}-\hat{\boldsymbol{A}}_{\tau}||_F$ with t for GEM-K.

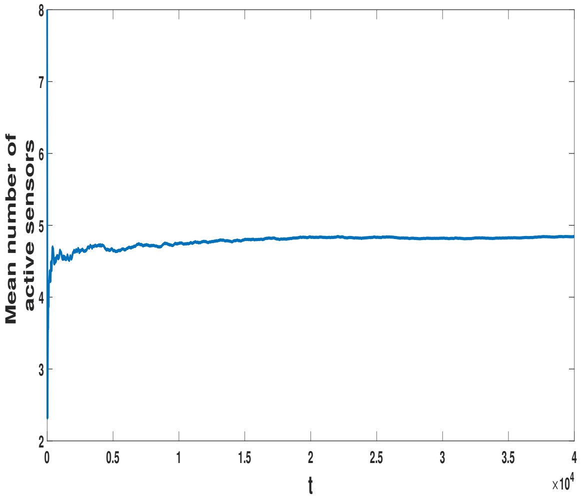

Figure 3. Variation of $\frac{1}{t}\sum_{\tau=1}^{t}||\boldsymbol{b}(\tau)||_1$ with t for GEM-K.

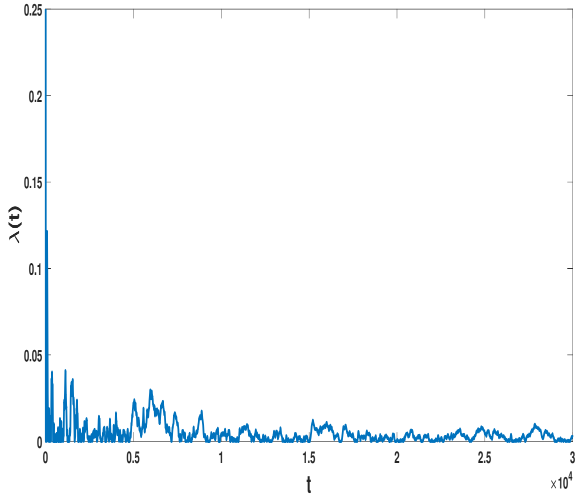

Figure 4. Variation of $\lambda(t)$ with t for GEM-K.

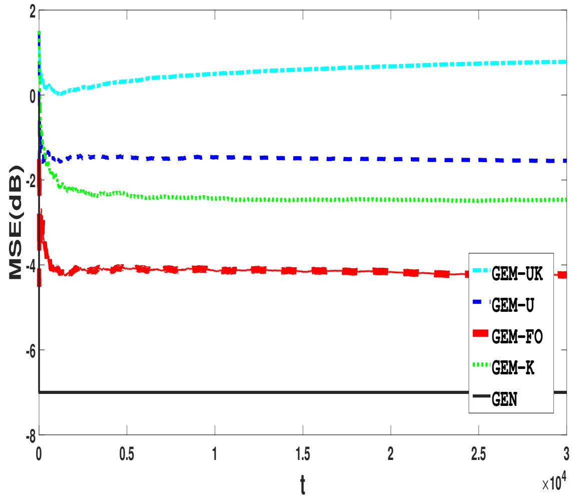

Figure 5. MSE performance comparison among various algorithms.

converge to the true TPM for GEM-UK; instead, it converges to some local optimum as guaranteed by the EM algorithm. For all relevant algorithms, we have noticed that the mean number of active sensors, calculated as $\frac{1}{t}\sum_{\tau=1}^{t}||\boldsymbol{b}(\tau)||_1$, converges to $\bar{N}$; this has been illustrated only for GEM-K algorithm in Figure 3 and the corresponding $\lambda(t)$ variation is shown in Figure 4. We observe that $\lambda(t)$ converges at a slower rate.

B. Performance comparison

In Figure 5, we have compared the MSE performance of various algorithms. We observe that, the TPM estimate in GEM-K converges to the true TPM, and hence the asymptotic MSE performance of GEM-K and GEM-FI are same. Hence, we do not show the MSE performance of GEM-FI separately. Figure 5 shows that, GEN has the best MSE performance due to perfect knowledge of the previous state, and GEM-UK has the worst MSE performance because it converges to a local optimum. GEM-FO performs better than GEM-K because it uses more sensor observation, but it cannot outperform GEN. GEM-U outperforms GEM-UK, since it has knowledge of observation mean and covariances. However, we note in multiple simulations that, on a fair comparison with GEM-K, GEM-U performs worse by approximately 1 dB; this shows the power of Gibbs sampling against uniform sampling.

We have repeated this for 10 different instances, and found that the ordering of MSE performances across algorithms remained unchanged, though the relative performance gaps among the algorithms varied. However, performance gap of GEM-K was observed to be within 2 dB of the MSE of GEM-FO, and within several dB from the MSE of GEN. This shows that GEM is very useful for tracking a Markov chain.

VI. Conclusions

We have provided a low-complexity active sensor selection algorithm for centralized tracking of a Markov chain with unknown transition probability matrix. The algorithm uses Gibbs sampling, multi-timescale stochastic approximation, online EM and Kalman-like state estimation to achieve a good compromise among computational complexity, fidelity of estimate, and energy and bandwidth usage in state estimation. Performance of the algorithm has been validated numerically. We seek to prove convergence of the proposed algorithm, and also extend this work for distributed tracking problems in our future research.

References

[1] D. Wang, J. W. Fisher III, and Q. Liu, "Efficient observation selection in probabilistic graphical models using bayesian lower bounds."

[2] F. Schnitzler, J. Yu, and S. Mannor, "Sensor selection for crowdsensing dynamical systems," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015, pp. 829–837.

[3] W. Hui, C. Hyeok-soo, A. Nazim, D. M. Jamal, and H. J. Won-Ki, "Information-based energy efficient sensor selection in wireless body area networks," in *Communications (ICC), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–6.

[4] F. R. Armaghani, I. Gondal, and J. Kamruzzaman, "Dynamic sensor selection for target tracking in wireless sensor networks," in *Vehicular Technology Conference (VTC Fall), 2011 IEEE*. IEEE, 2011, pp. 1–6.

[5] V. Krishnamurthy and D. Djonin, "Structured threshold policies for dynamic sensor scheduling—a partially observed markov decision process approach," *IEEE Transactions on Signal Processing*, vol. 55, no. 10, pp. 4938–4957, 2007.

[6] D. Zois, M. Levorato, and U. Mitra, "Active classification for pomdps: A kalman-like state estimator," *IEEE Transactions on Signal Processing*, vol. 62, no. 23, pp. 6209–6224, 2014.

[7] ——, "Energy-efficient, heterogeneous sensor selection for physical activity detection in wireless body area networks," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1581–1594, 2013.

[8] W. Wu and A. Arapostathis, "Optimal sensor querying: General markovian and lqg models with controlled observations," *IEEE Transactions on Automatic Control*, vol. 53, no. 6, pp. 1392–1405, 2008.

[9] V. Gupta, T. Chung, B. Hassibi, and R. Murray, "On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage," *Automatica*, vol. 42, pp. 251–260, 2006.

[10] A. Chattopadhyay and U. Mitra, "Optimal sensing and data estimation in a large sensor network," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–7.

[11] ——, "Optimal active sensing for process tracking," in *2018 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2018, pp. 551–555.

[12] V. S. Borkar, *Stochastic approximation: a dynamical systems viewpoint.* Cambridge University Press, 2008.

[13] A. Chattopadhyay and U. Mitra, "Dynamic sensor subset selection for centralized tracking of an iid process," *IEEE Transactions on Signal Processing*, 2020.

[14] ——, "Active sensing for markov chain tracking," in *GlobalSIP 2018*. IEEE, 11 2018, pp. 1050—1054.

[15] O. Cappé, "Online em algorithm for hidden markov models," *Journal of Computational and Graphical Statistics*, vol. 20, no. 3, pp. 728–749, 2011.

[16] P. Brémaud, *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*. Springer Science & Business Media, 2013, vol. 31.

[17] B. Hajek, *An Exploration of Random Processes for Engineers*. Lecture Notes for ECE 534, 2011.