Sensor Selection for Crowdsensing Dynamical Systems

François Schnitzler
Technion

Jia Yuan Yu
IBM research - Ireland

Shie Mannor
Technion

Abstract

We model crowdsensing as the selection of sensors with unknown variance to monitor a large linear dynamical system. To achieve low estimation error, we propose a Thompson sampling approach combining submodular optimization and a scalable online variational inference algorithm to maintain the posterior distribution over the variance. We also consider three alternative parameter estimation algorithms. We illustrate the behavior of our sensor selection algorithms on real traffic data from the city of Dublin. Our online algorithm achieves significantly lower estimation error than sensor selection using a fixed variance value for all sensors.

1 INTRODUCTION

Today, many cities or countries leverage large numbers of new or existing sensors to obtain a fine, real-time picture of their territory. Monitoring and predicting the behavior of traffic, water or power networks, to name a few, lead to better management and planning. However, sensors typically do not cover the whole city. For example, around 750 junctions are equipped with SCATS[1] vehicle-count sensors in the Greater Dublin Area, which amounts to only 4% of all junctions.

Crowdsensing, that is leveraging information provided by sensors carried or set up by citizens, is an attractive addition or alternative to dedicated sensors: it is versatile and has no deployment cost. Crowdsensing is actively being used for large-scale monitoring, for example Waze[2] or (Venanzi *et al.*, 2013).

[1] Sydney Co-ordinated Adaptive Traffic System
[2] https://www.waze.com/

Appearing in Proceedings of the 18^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2015, San Diego, CA, USA. JMLR: W&CP volume 38.

Here, we jointly address two crowdsensing challenges.

1. Measurement noise affecting privately owned sensors can vary and can hardly be estimated a priori. For example, traces of mobile phones owned by bus users or bikers may be less useful than those of mobiles traveling by car to measure traffic.
2. The number of sensors that can be queried may be limited by the bandwidth or the budget available if querying a sensor has an associated cost.

We consider crowdsensing for monitoring a big linear dynamical system (LDS) x_t (traffic, pollution...), modeled as the **selection of sensors** with a priori **unknown variance** Θ. We propose and evaluate algorithms to select at each time step a subset u_t of sensors to minimize the state estimation error. We now discuss a few related works before stating our contributions.

Sensor Selection and Submodularity Constrained sensor selection problems have been cast or approximated as submodular optimization problems. For example, Krause *et al.* (2008) select the best GPS traces to query to minimize the estimation error in a spatial traffic model. Meliou *et al.* (2007) optimize the path of a robot to maximize submodular functions of a spatio-temporal process. In these works, all model parameters are known, whereas we must estimate Θ.

Optimizing unknown submodular functions was studied by Hazan and Kale (2009); Streeter and Golovin (2008). In these works, the numerical noisy result of each selection u_t is observed and that output must be maximized. In our setting, the objective (the estimation error) is not observed.

Parameter Estimation Online parameter estimation in a LDS has been studied a lot, but, to our knowledge, rarely together with sensor selection.

Venanzi *et al.* (2013) estimate the noise of crowdsourced radiation sensors in Japan. However, they do not select sensors, a key difference with our work. Their method is also not online. However, we assume a model of the spatio-temporal process is available whereas they learn one (but with only 2 parameters).

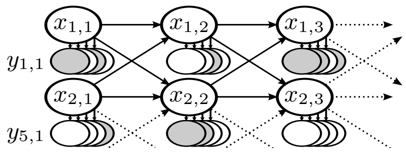


Figure 1: A hidden process x_t is monitored by sensors y_t. Sensors $u_t \in y_t$ are selected for each t. Gray/white variables are observed/hidden. $u_1 = \{1, 3, 8\}$.

Parameter Estimation and Sensor Selection Landmarks selection has been considered in the Simultaneous Localization And Mapping (SLAM) problem: the joint estimation of the position of a robot (x_t) and of landmarks used for localization (equivalent to Θ). Martinez-Cantin *et al.* (2009) optimize the landmarks selected based on a Monte-Carlo approximation of the expectation of the objective function. The main difference between SLAM and our work is the dimension of the state space. While robots are typically characterized by few state variables (2D position), we consider systems larger by at least 2 orders of magnitude, e.g., the number of streets. Sampling based methods used in SLAM algorithms cannot scale up.

Crowdsourcing is the inference of unknown object labels from labels provided by imperfect experts (in our case, the sensors) rather than from a reliable expert (a city owned sensor). The EM algorithm (Raykar *et al.*, 2010) and sequential Bayesian estimation (Donmez *et al.*, 2010) have for example been used to assess the accuracy of each worker. Crowdsourcing is simpler than our crowdsensing problem: we estimate states of a spatio-temporal process, not independent labels.

Krause and Guestrin (2007) select observations to maximize estimation accuracy in a Gaussian process (GP) whose parameters are unknown but belong to a known finite set. They do not consider individual sensor noise and so this setting is simpler than ours.

Hoang *et al.* (2014) consider active sensing for a GP with unknown parameters by solving, over a finite horizon T, Bellman equations incorporating the uncertainty about the parameters. All sensors share the same noise parameter, a discrete set of parameter values is considered and the state distribution is approximated by Monte-Carlo in the Bellman equations. Applying their solution to crowdsensing would be computationally expensive. To differentiate between the best and suboptimal sensors, finely discretized parameter values must be considered for each sensor. To explore this large parameter space, T must be larger. To select each set of sensors and for each step of the planning horizon, a score is computed for each sensor, requiring an integration over the parameter space. Finally, the number of Monte-Carlo samples is exponential in T.

Our three contributions

1. Our key contribution is a scalable, online crowdsensing algorithm handling both sensor selection and unknown sensing noise.
2. Up to our knowledge, the online variational inference algorithm used within our main algorithm has not been applied to linear dynamical systems before. Similar algorithms have however been used on problems where each iteration is performed on independent realizations.
3. We also evaluate three alternative algorithms (based on Expectation-Maximization (EM), online EM or Gibbs sampling).

Outline First, Section 2 formalizes the problem considered. We then present our approaches in Section 3 and empirically evaluate them in Section 4 on a Brownian motion, a more complex artificial model and on Dublin traffic data. Finally, Section 5 provides some additional perspectives before Section 6 concludes.

2 PROBLEM FORMULATION

We consider a LDS, where $x_t \in \mathbb{R}^d$ denotes the set of hidden variables at time step t, as a mathematical model of the spatio-temporal process. LDS are commonly used to model such processes, including traffic (Wang *et al.*, 2008). Hidden variables x_t are monitored by selecting at each time step a subset of K sensors $u_t \subseteq \{1, \dots I\}$ and receiving the selected observations $y_{i,t}, i \in u_t$. More formally,

$$x_{t+1} = A_t x_t + w_t \tag{1}$$

$$y_{i,t} = C_{i,t} x_t + v_{i,t} \qquad \text{for } i \in u_t \tag{2}$$

$$y_{i,t} = \emptyset \qquad \text{for } i \notin u_t \ . \tag{3}$$

Each observation $y_{i,t}$ is perturbed by a Gaussian noise $v_{i,t} \sim \mathcal{N}(0, \sigma_i^2)$ of variance σ_i^2. This models the noise of the sensors, unknown to the operator ($\Theta \equiv \{1/\sigma_i^2\}_{i=1}^I$ is unknown). The initial state is $x_1 \sim \mathcal{N}(\bar{x}_1, \Sigma_1)$, the state transition noise is $w_t \sim \mathcal{N}(\bar{w}_t, \Sigma_{w,t})$. We assume that $A_t, C_{i,t}, \bar{x}_1, \Sigma_1, \Sigma_{w,t}, \bar{w}_t \in \mathbb{R}^{d \times d}, \mathbb{R}^{1 \times d}, \mathbb{R}^d, \mathbb{R}^{d \times d}, \mathbb{R}^{d \times d}, \mathbb{R}^d$ are known $\forall i, t$. For $t' \leq t$, let $\hat{x}_{t|t',u}$ and $\hat{\Sigma}_{t|t',u}$ respectively denote the system state estimate and the covariance of the estimation error at time t, given a selection of sensors $u_{1:t'} = \{u_1, ..., u_{t'}\}$. The estimates $\hat{x}_{t|t',u}$ and $\hat{\Sigma}_{t|t',u}$ can be computed efficiently by the Kalman filter (see Appendix 7.1 for a brief reminder). Moreover, $\hat{\Sigma}_{t|t',u}$ can be computed in advance, since it does not depend on the actual value of the measurements, for a given $u_{1:t'}$. For ease of notation, $y_t \equiv \{y_{i,t}\}_{i \in u_t}$. A subscript i will always denote a sensor and t a time step.

We are interested in settings with a large number of sensors and where querying a sensor has an associated

Algorithm 1 Sensor selection meta-algorithm

1: **for** $t = 1 \to \infty$ **do**
2: Obtain $\hat{\Theta}_t$, $\hat{p}(x_t|y_{1:t})$ from u_t, y_t, $\hat{p}(x_{t-1}|y_{1:t-1})$ {Algorithms 2 or 3}
3: Optimize u_{t+1} based on $\hat{\Theta}_t$, $\hat{p}(x_t|y_{1:t})$ {Greedy}
4: **end for**

cost. To model this cost, only $K < I$ sensors can be queried at each time step.

Our goal is to select, at each time step t, the variables $y_{i,t}$ that are revealed (a subset of sensors $u_t \subseteq \{1, \ldots, I\}$) to minimize the quadratic state estimation error $||x_t - \hat{x}_{t|t',u}||^2_{l_2}$. Since x_t is unknown, we optimize over the expected error or equivalently we maximize the reduction in expected quadratic estimation error: $f_t(u) : 2^{I^t} \to \mathbb{R} = \mathrm{tr}(\hat{\Sigma}_t - \hat{\Sigma}_{t|t,u})$, where tr denotes the trace of a matrix. This is a typical objective (Krause *et al.*, 2008; Chen *et al.*, 2012), see also Appendix 7.2.

In a LDS, $u_{1:t}$ may also impact state uncertainty for $t' > t$, so the selection must be optimized on a horizon T, possibly with a discount factor $\gamma \in [0, 1[$:

$$u^*_{1:T} = \arg\min_{u_{1:T}:|u_t|\leq K\ \forall t} \sum_{t=1}^{T} \gamma^t f_t(u_{1:t}) \ . \qquad (4)$$

We assume that $A_t, C_{i,t}, \bar{x}_1, \Sigma_1, \Sigma_{w,t}$ and $\bar{w}_t$ are known, which could be considered a strong hypothesis. C_t is known if the position of each sensor is known. While we acknowledge that crowdsourced sensors might not be localized, we feel that this assumption is reasonable, given the widespread use of GPS. A_t, $\bar{x}_1$, Σ_1, $\Sigma_{w,t}$ and $\bar{w}_t$ characterize the process monitored. Spatio-temporal processes taking place in cities (such as traffic, flooding etc.) have been monitored, studied and modeled extensively. Postulating the existence of a model provided by an expert or of historical measurements from which a model can be derived seems reasonable to us.

If no model is available, there are several possibilities:

- assuming temporal independence and using a Gaussian Process with known parameters (as Krause *et al.* (2008); Venanzi *et al.* (2013));
- combining our approach with existing works that learn a few model parameters (Krause and Guestrin, 2007; Hoang *et al.*, 2014);
- adapting the algorithms we propose to learn the system parameters as well.

Developing and studying these alternatives is outside the scope of the present work. Data or expertise available in any particular application influences the best way to estimate system parameters. However, estimating the observation noise is relevant whenever using crowdsourced sensors.

3 CROWDSENSING ALGORITHM

The problem described in Section 2 has an inherent trade-off between exploration (select sensors to estimate Θ) and exploitation (select good sensors to estimate x_t). Our first contribution is an algorithm based on Thompson sampling, an implicit method to deal with such trade-off (Thompson, 1933; Chapelle and Li, 2011). Thompson sampling maintains a posterior distribution $P(\Theta|y_{1:t})$ over unknown parameters and, at each time step t, it (1) samples $\hat{\Theta}_t$ from $P(\Theta|y_{1:t})$ and (2) selects an optimal action conditionally on $\hat{\Theta}_t$. In our case, each action corresponds to the selection of a set of sensors. An estimate over the state distribution must also be maintained. This process is summarized by Algorithm 1. The algorithm is initialized by $\hat{P}(x_1) = \mathcal{N}(\bar{x}_1, \Sigma_1)$ and a prior $P(\Theta)$.

Maintaining a posterior $P(\Theta|y_{1:t})$ in a scalable way is not trivial for our setting. While Gibbs sampling can generate $\Theta, x_t \sim P(\Theta, x_t|y_{1:t})$, this batch approach cannot scale up to the continuous monitoring of a large system (the scale of a city). Instead, we use online variational inference with stochastic approximation.

We also consider an alternative step 2 of Algorithm 1: estimating Θ by ML instead of sampling from $P(\Theta|y_{1:t})$. Using ML estimation does not handle the exploration-exploitation trade-off. However, it is often considered in crowdsourcing problems (independent labels inference). We do not expect the resulting sensor selection algorithms to outperform Thompson sampling. We include them in our experiments for comparison and to cover 4 distinct classes of methods: batch or online, maximum likelihood or Bayesian.

We now detail these 4 alternative Θ estimation/sampling algorithms (Section 3.1) and the sensor selection procedure (Section 3.2).

3.1 Parameter Estimation (Algorithm 1, l.2)

This section describes the online variational inference algorithm (second contribution) and three other methods to estimate or sample Θ based on a growing sequence of observations $y_{1:t}$. We consider both maximum likelihood (ML) and Bayesian estimations of Θ. ML estimation (Section 3.1.1) is simpler but cannot be used to address the exploration/exploitation trade-off. A Bayesian approach does not provide an estimate for Θ, but samples a realization from a posterior distribution $P(\Theta|y_{1:t})$ or an approximation. We describe two such approaches: Gibbs sampling (Section 3.1.2) and finally the key online variational inference generating samples online and efficiently (Section 3.1.3).

Algorithm 2 Online EM algorithm for crowdsensing (update for time t)

Require: u_t, $\{y_{i,t}\}_{i \in u_t}$, $\hat{p}(x_{t-1}|y_{1:t-1}, \Theta_{0:t-2})$, $\left[\{k_i\}_{i=1}^I\right]$, $\Theta_{t-1} = \{\sigma^2_{i,t-1}\}_{i=1}^I$

1: $\hat{p}(x_t|y_{1:t}, \Theta_{0:t-1}) = \int \hat{p}(x_{t-1}|y_{1:t-1}, \Theta_{0:t-2})\, p(x_t|x_{t-1}, \Theta_{t-1}) p(y_t|x_t, \Theta_{t-1}) dx_{t-1}$ $\{\mathcal{N}(\hat{x}_{t,u_{1:t}}, \hat{\Sigma}_{t,u_{1:t}})$, using Kalman filter$\}$

2: $\Theta_t = \Theta_{t-1}$ {Maintain parameters for $i \notin u_t$}

3: **for** $i \in u_t$ **do**

4: $s_{i,t} = y_{i,t} y_{i,t}^T - C_{i,t} \hat{x}_{t,u_{1:t}} y_{i,t}^T - C_{i,t}^T \hat{x}_{t,u_{1:t}}^T y_{i,t} + C_{i,t} \left(\hat{\Sigma}_{t,u_{1:t}} + \hat{x}_{t,u_{1:t}} \hat{x}_{t,u_{1:t}}^T\right) C_{i,t}^T$ {New statistics}

5: $\sigma^2_{i,t} = \lambda_{k_i} s_{i,t} + (1 - \lambda_{k_i}) \sigma^2_{i,t-1}$ {Update Θ_t}

6: $k_i = k_i + 1$

7: **end for**

output $\Theta_t = \{\sigma^2_{i,t}\}_{i=1}^I$, $\hat{p}(x_t|y_{1:t}, \Theta_{0:t-1})$, $\left[\{k_i\}_{i=1}^I\right]$

The first three methods are not new. To the best of our knowledge, the online variational inference method is novel, although similar methods have been proposed.

3.1.1 Maximum Likelihood

The ML estimate of Θ is given by

$$\hat{\Theta} = \arg\max_{\Theta} p(y_{1:t}|\Theta) \tag{5}$$

$$= \arg\max_{\Theta} E_{x_{1:t}} L(x_{1:t}, y_{1:t}, \Theta) \ , \tag{6}$$

where $p(.|.)$ denotes a conditional probability density function and $L(.,.,.)$ the complete log-likelihood. Equation (6) cannot usually be solved analytically.

EM Algorithm (Batch ML) The EM algorithm (Dempster *et al.*, 1977) performs ML estimation for models with hidden variables. In a nutshell, EM solves equation (6) iteratively, alternating between two steps: computing $Q_k(\Theta)$, an expectation of the log-likelihood given fixed-length observations $y_{1:T}$ and the current estimate Θ_k of the parameters; and maximizing this expectation: $\Theta_{k+1} = \arg\max_\Theta Q_k(\Theta)$.

In our case, the E and M steps are easily derived from $Q_k(\Theta)$ (details in Appendix 7.3):

$$s_{i,t,k} = E_{\tilde{x}_t^k|\Theta_k y_{1:T}} \left[(y_{i,t} - C_{i,t} \tilde{x}_t^k)^2\right] \tag{7}$$

$$= (y_{i,t} - C_{i,t} \hat{x}^k_{t,u_{1:t}})^2 + C_{i,t} \hat{\Sigma}^k_{t,u_{1:t}} C_{i,t}^T \tag{8}$$

$$\Theta_{k+1}(i) = \frac{1}{\sum_{t'=1}^T \mathbb{1}(i \in u_{t'})} \sum_{t=1}^T \mathbb{1}(i \in u_t) s_{i,t,k} \ . \tag{9}$$

Online EM Algorithm (Online ML) EM operates on batch data and has a computational complexity $\mathcal{O}(T(d^3 + K^3))$ per iteration. Applying EM on a growing sequence of observations will become intractable. Using a finite window is also problematic because some sensors might not be queried during the time window used. Therefore, we consider an online EM algorithm based on stochastic approximation (Krishnamurthy and Moore, 1993).

This algorithm updates state and parameter estimates whenever a new subset of observations is received. Hence, we denote parameters estimates by Θ_t. The update scheme is based on a recursive estimate of $s_{i,t,t}$, shortened to $s_{i,t}$:

$$s_{i,t} = E_{x_t|\Theta_{0:t-1} y_{1:t}} \left[(y_{i,t} - C_{i,t} x_t)^2\right] \tag{10}$$

$$\hat{p}(x_t|\Theta_{0:t-1}, y_{1:t}) = \int \hat{p}(x_{t-1}|\Theta_{0:t-2}, y_{1:t-1})\, p(x_t|\Theta_{t-1}, y_t, x_{t-1}) dx_{t-1} \ . \tag{11}$$

The notation $\Theta_{0:t}$ in the latter equation stresses that the state density estimate is updated recursively, as apposed to EM which recomputes all state density estimates (for $t = 1, \ldots, t$) using the latest parameters estimate Θ_t. The update rule for $\sigma^2_{i,t} : i \in u_t$ is

$$\Theta_t(i) = \lambda_{k_i} s_{i,t} + (1 - \lambda_{k_i}) \sigma^2_{i,t-1} \ , \tag{12}$$

where k_i is the number of updates performed on $\sigma^2_{i,t}$ and where the sequence $\{\lambda_t\}$ satisfies $\lim_{T \to \infty} \sum_{t=1}^T \lambda_t = \infty$ and $\lim_{T \to \infty} \sum_{t=1}^T \lambda_t^2 < \infty$. The parameters of the sensors which are not queried ($\{1, \ldots, I\} \backslash u_t$) are not updated. The full procedure is detailed in Algorithm 2. Note that inputs and outputs between brackets correspond to internal states of the estimation algorithm, not returned to Algorithm 1.

3.1.2 Gibbs Sampling (Batch Bayesian)

Deriving $P(\Theta|y_{1:t})$ analytically is typically not possible. Gibbs sampling can sample a realization from $P(\Theta|y_{1:T})$ without computing it explicitly. Wills *et al.* (2012), for example, applied it to LDS. Used for noise estimation only, Gibbs sampling alternates between sampling $x^k_{1:T} \sim P(x_{1:T}|y_{1:T}, \Theta_k)$, computed by a Kalman filter, and $\Theta_{k+1} \sim P(\Theta|y_{1:T}, x^k_{1:T}) = \prod_{i=1}^I invG(\alpha_{i,k}, \beta_{i,k})$ with

$$\beta_{i,k} = \beta + \sum_{t=1}^T \mathbb{1}(i \in u_t) \left[(y_{i,t} - C_{i,t} x_t^k)^2\right] / 2 \tag{13}$$

$$\alpha_{i,k} = \alpha + \sum_{t=1}^T \mathbb{1}(i \in u_t)/2 \ , \tag{14}$$

where the conjugate prior $P(\Theta)$ is a product of inverse gamma distributions $invG(\alpha, \beta)$. β and α respectively correspond to half the sum of empirical quadratic errors and half the number of samples (Murphy, 2007).

We use this conjugate prior because it leads to convenient updates. If more information is available, a more suitable prior can be used.

We use $\beta_{i,0} = 0.5$ to obtain a prior distribution with a large variance (low informative prior). $\alpha_{i,0}$ should be chosen such that the mean of the distribution is at a sensible value. In our experiments, $\alpha_{i,0}$ is typically such that $E(invG(\alpha_{i,0}, \beta_{i,0})) = \sum_{i=1}^{I} \sigma_i^2 / I$.

3.1.3 Variational Inference (Online Bayesian)

This section details the key algorithm allowing an online and efficient first step of each Thompson sampling iteration, sampling $P(\Theta|y_{1:T})$. Variational inference approximates intractable posterior distributions such as $P(\Theta, x_t|y_{1:T})$. It corresponds to solving an optimisation problem, the minimization of $D_{KL}(\breve{P}(x_t, \Theta|y_{1:t})||P(x_t, \Theta|y_{1:t}))$, where $\breve{p}(x_t, \Theta|y_{1:t})$ is an approximate probability density. In typical batch variational inference, the gradient is computed using all time steps. The online algorithm we propose here relies on stochastic approximation. Noisy estimates of the gradient are computed by considering only a few time steps (one, in our case). Alternatively, the difference between this online variational inference algorithm and batch variational inference is the same difference as between EM and online EM.

For online variance estimation in a LDS, Sarkka and Nummenmaa (2009) use the following assumptions:

$$\breve{p}(x_t, \Theta|y_{1:t}) \equiv \breve{p}(x_t|y_{1:t})\breve{p}(\Theta|y_{1:t}) \tag{15}$$

$$\breve{p}(x_t|y_{1:t}) \equiv \mathcal{N}(\breve{x}_t, \breve{\Sigma}_t) \tag{16}$$

$$\breve{p}(\Theta|y_{1:t}) \equiv \prod_{i=1}^{I} invG(\breve{\alpha}_{i,t}, \breve{\beta}_{i,t}) \tag{17}$$

and describe online updates in order to derive $\{\breve{x}_t, \breve{\Sigma}_t, \{\breve{\alpha}_{i,t}, \breve{\beta}_{i,t}\}_i\}$ from previous values

$$\{\breve{x}_{t-1}, \breve{\Sigma}_{t-1}, \{\breve{\alpha}_{i,t-1}, \breve{\beta}_{i,t-1}\}_i\} \tag{18}$$

for a LDS with time-varying observation noise.

At each time step t, when y_t is received, $\mathcal{N}(\breve{x}_t^0, \breve{\Sigma}_t^0) \equiv \breve{p}^0(x_t|y_{1:t-1})$ is computed by a time transition:

$$\breve{x}_t^0 = A_{t-1}\breve{x}_{t-1} \tag{19}$$

$$\breve{\Sigma}_t^0 = A_{t-1}\breve{\Sigma}_{t-1}A_{t-1}^T + \Sigma_{w,t-1} \tag{20}$$

$$\{\breve{\alpha}_{i,t}^0, \breve{\beta}_{i,t}^0\}_i = \{\rho\breve{\alpha}_{i,t-1}, \rho\breve{\beta}_{i,t-1}\}_i \ . \tag{21}$$

The forgetting factor ρ is used to handle the evolution of the variance. Our Θ is constant, so we use $\rho = 1$. Then, $\{\breve{x}_t, \breve{\Sigma}_t\}$ and $\{\breve{\alpha}_{i,t}, \breve{\beta}_{i,t}\}_i$ are alternatively updated for a few iterations (of index k):

$$\breve{\Theta}_t^k = E[\Theta|\{\breve{\alpha}_{i,t}^k, \breve{\beta}_{i,t}^k\}_i] \tag{22}$$

$$= diag(\{\breve{\beta}_{i,t}^k/\breve{\alpha}_{i,t}^k\}_i) \tag{23}$$

$$\mathcal{N}(\breve{x}_t^k, \breve{\Sigma}_t^k) \equiv \breve{p}^k(x_t|y_{1:t}) \propto \mathcal{N}(\breve{x}_t^0, \breve{\Sigma}_t^0)p(y_t|\breve{\Theta}_t^k, x_t)$$

$$\breve{\alpha}_{i,t}^{k+1} = \breve{\alpha}_{i,t}^0 + 1/2 \qquad \forall i \tag{24}$$

$$s_{i,t} = (y_i - C_{i,t}\breve{x}_t^k)^2 + C_{i,t}\breve{\Sigma}_t^k C_{i,t}^T \qquad \forall i \tag{25}$$

$$\breve{\beta}_{i,t}^{k+1} = \breve{\beta}_{i,t}^0 + s_{i,t}/2 \qquad \forall i \ . \tag{26}$$

We modify this algorithm by considering a subset of observations (which is simple), by removing ρ (observation noise is time invariant), by performing only one iteration at each time step and, more importantly, by using a stochastic approximation scheme to ensure convergence of the hyperparameters (lines 5-6 in Algorithm 3).

Directly applying stochastic approximation to $\{\breve{\alpha}_{i,t}, \breve{\beta}_{i,t}\}_i$ is not possible: they correspond to a sum (of sufficient statistics) and not to an expectation. Instead, we associate a weight w_t (we omit the subscript i) to each expected empirical quadratic error S_t (of expectation $\breve{\beta}_{i,t}/\breve{\alpha}_{i,t}$) and attribute to the latest sufficient statistics s_t a weight of 1. When $\lambda_t = 1/t$, a stochastic approximation update corresponds to a weighted average between S_{t-1} (weighted by $w_{t-1} = N - 1$) and s_t. When $\lambda_t > 1/t$, the update corresponds to a weighted average between S_{t-1} and s_t, where the weight of S_{t-1} has been discounted by d. Comparing the weighted average and the stochastic approximation update provides a value for $2\breve{\alpha}_t = w_t = dw_{t-1} + 1$, the weight after update:

$$S_t = \frac{dw_{t-1}S_{t-1} + s_t}{dw_{t-1} + 1} = (1 - \lambda_t)S_{t-1} + \lambda_t s_t \tag{27}$$

$$\rightarrow d = (1 - \lambda_t)/(\lambda_t w_{t-1}) \tag{28}$$

$$\rightarrow 2\breve{\alpha}_t = dw_{t-1} + 1 = 1/\lambda_t \ . \tag{29}$$

We use $1/\lambda_t$ as the new equivalent number of samples after the t^{th} stochastic approximation update. See line 6 in Algorithm 3 for the exact updates. A stochastic approximation was also introduced in an online variational inference algorithm by Hoffman *et al.* (2013), but only for temporally independent hidden variables.

The resulting procedure is detailed in Algorithm 3. The first difference with the online EM algorithm is the computation of the expected value of Θ according to $\hat{P}(\Theta|y_{1:t})$ (line 1). This value is then used for the Kalman filter update (line 2). The update of $\sigma_{i,t}^2$ is replaced by an update of $\{\breve{\alpha}_{i,t}, \breve{\beta}_{i,t}\}$ (line 6). Finally, $\breve{P}(\Theta_t|y_{1:t})$ is sampled, and the resulting parameter values are returned (line 9). They are used to select the sensors queried at the next time step.

$\{\breve{\alpha}_{i,t}, \breve{\beta}_{i,t}\}$ are initialized to the same values as the ones used for Gibbs sampling.

Algorithm 3 Online, Scalable Variational Inference for crowdsensing (update for time t)

input $u_t,\ \{y_{i,t}\}_{i\in u_t},\ \breve{p}(x_{t-1}|y_{1:t-1}),\ \left[\{\breve{\alpha}_{i,t-1},\breve{\beta}_{i,t-1}\}_{i=1}^{I},\ \{k_i\}_{i=1}^{I}\right]$

$\breve{\Theta}_{t-1} = \{\breve{\beta}_{i,t-1}/(\breve{\alpha}_{i,t-1}-1)\}_{i=1}^{I}$ $\quad\{\sigma_{i,t}^2 = mean(invG(\breve{\alpha}_{i,t-1},\breve{\beta}_{i,t-1}))\}$

$\mathcal{N}(\breve{x}_{t,u_{1:t}},\breve{\Sigma}_{t,u_{1:t}}) = \breve{p}(x_t|y_{1:t},\Theta_{0:t-1}) = \int \breve{p}(x_{t-1}|y_{1:t-1},\Theta_{0:t-2})p(x_t|x_{t-1},\Theta_{t-1})\ p(y_t|x_t,\Theta_{t-1})dx_{t-1}$

$\{\breve{\alpha}_{i,t},\breve{\beta}_{i,t}\}_{i=1}^{I} = \{\breve{\alpha}_{i,t-1},\breve{\beta}_{i,t-1}\}_{i=1}^{I}$ $\quad$ {Maintain parameters of non-queried users}

for $i \in u_t$ **do**

$\quad s_{i,t} = y_{i,t}y_{i,t}^T - C_{i,t}\breve{x}_{t,u_{1:t}}y_{i,t}^T - C_{i,t}^T\breve{x}_{t,u_{1:t}}^T y_{i,t} + C_{i,t}\left(\breve{\Sigma}_{t,u_{1:t}} + \breve{x}_{t,u_{1:t}}\breve{x}_{t,u_{1:t}}^T\right)C_{i,t}^T$ $\quad$ {New statistics}

$\quad \breve{\beta}_{i,t} = 0.5\left[s_{i,t} + \frac{1-\lambda_{k_i}}{\lambda_{k_i}}\frac{\breve{\beta}_{i,t-1}}{\breve{\alpha}_{i,t-1}}\right],\ \breve{\alpha}_{i,t} = 0.5/\lambda_{k_i},\ k_i = k_i+1$ $\quad$ {Hyperparameters}

end for

$\sigma_{i,t}^2 \sim invG(\breve{\alpha}_{i,t},\breve{\beta}_{i,t}) \qquad \forall i \in \{1,...,I\}$ $\quad$ {Draw $\breve{\Theta}_t$ from $\breve{p}(\Theta|y_{1:T},x_{1:T})$}

output $\breve{\Theta}_t = \{\sigma_{i,t}^2\}_{i=1}^{I},\ \breve{P}(x_t|y_{1:t},\breve{\Theta}_{0:t-1}),\ \left[\{\breve{\alpha}_{i,t},\breve{\beta}_{i,t}\}_{i=1}^{I},\ \{k_i\}_{i=1}^{I}\right]$

3.2 Sensor Selection (Algorithm 1, l.3)

The second step of the Thompson sampling iteration is to select the observations based on a given Θ. To do so, we assume the objective is submodular and adapt (Krause *et al.*, 2008) for a LDS:

Definition 3.1. A submodular function $f : 2^V \to \mathbb{R}$ is a function of a subset of a set V such that, for all subsets $A, B : A \subseteq B \subseteq V$ and element $c \in V\backslash B$,

$$f(A\cup c) - f(A) \geq f(B\cup c) - f(B)\ . \tag{30}$$

Intuitively, a submodular function models diminishing returns: the more readings are already available, the less a new sensor reading helps. Our objective function (Equation 4) is not submodular if particular independence relationships between variables do hold, but submodularity is typically assumed (Krause *et al.*, 2008; Meliou *et al.*, 2007). A sufficient condition for submodularity is the conditional suppressor-freeness (Das and Kempe, 2008). Our work can be directly applied on any submodular sensor selection problem. If a problem is not submodular, the combination of Thompson sampling and online variational inference (our main contribution) stays relevant, only the optimisation step (this subsection) must be adapted.

The sensors queried at time step t (u_t) are selected at time t by solving the objective function with a finite horizon ΔT, starting from $\{\hat{x}_{t|t-1,u},\hat{\Sigma}_{t|t-1,u}\}$:

$$u_{t:t+\Delta T} = \underset{u'_{t:t+\Delta t}:|u'_{t'}|\leq K\ \forall t'}{\arg\min} f_{u_t}(u_{t:t+\Delta T}) \tag{31}$$

$$f_{u_t}(u_{t:t+\Delta T}) = \sum_{t'=t}^{t+\Delta T} \gamma^{t'} f_{t'}^{t}(u_{t:t'}) \tag{32}$$

$$f_{t'}^{t}(u_{t:t'}) = \mathrm{tr}(\hat{\Sigma}_{t'|t-1,u_{1:t-1}} - \hat{\Sigma}_{t'|t',u_{1:t'}})\ . \tag{33}$$

In other words, we optimize future measurements based on $\hat{P}(x_{t-1}|y_{1:t-1})$.

We make the assumption that each function $f_{t'}^{t}$ is submodular (cf. Definition 3.1). In that case, the objective function f_{u_t} is submodular too, because any finite sum of submodular functions is also submodular, and finding an optimal solution to Equation 31 is NP-hard. However f_{u_t} is nondecreasing and null for an empty input, so a greedy solution $u_{t:t+\Delta T}^{G}$ enjoys the following guarantee with respect to an optimal solution $u_{t:t+\Delta T}^{*}$ (Nemhauser *et al.*, 1978):

$$f_{u_t}(u_{t:t+\Delta T}^{G}) \geq (1-1/e)f_{u_t}(u_{t:t+\Delta T}^{*})\ . \tag{34}$$

4 EMPIRICAL RESULTS

In this section, we illustrate the respective performance and advantages of the four crowdsensing algorithms introduced in Section 3. We first compare algorithms based on batch EM (*EM*) and Gibbs sampling (*Gibbs*) to their online approximations (respectively *EMo* and *VB*) on a toy problem (Section 4.1), in terms of the number of optimal decisions and of run time. We then study the state estimation error achieved by *EMo* and *VB* on an artificial model (Section 4.2) and on traffic data collected in central Dublin (Section 4.3). Estimation of Θ is discussed in Appendix 7.5.

We compare these methods to three baselines: random selection (*random*); an oracle using the true value of Θ to select sensors and an estimated Θ for state estimation (*oracle*); and submodular optimization using a fixed $\hat{\Theta} : \hat{\sigma}_i^2 = mean(\Theta)\ \forall i$ (*meanNoise*). We tried other constant values for $\hat{\sigma}_i^2$, but this one lead to the most accurate results. The first two baselines estimate Θ with the online EM algorithm.

All 3 settings are briefly described below. The Dublin related one is further detailed in Appendix 7.4. We use $\gamma = 0.7$, $\lambda_k = 1/k^{\frac{2}{3}}$ and $\Delta T = 0$. Increasing ΔT up to 2 had no noticeable influence on the estimation error achieved. Ties are broken arbitrarily.

time steps	*EM*	*EMo*	*Gibbs*	*VB*
10	0.19	0.03	1.3	0.03
50	5	0.12	28	0.14
100	15	0.19	97	0.22

Table 1: Brownian motion model: run-time (seconds)

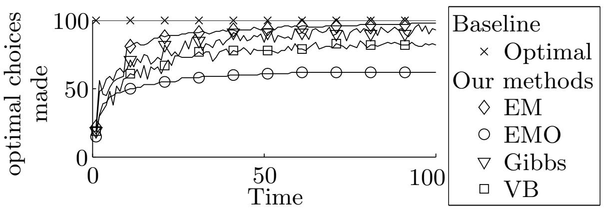


Figure 2: Performance of the online algorithms against the batch ones (Brownian motion model, 100 runs).

Brownian motion There is one hidden variable. Its dynamics is $x_t = x_t + w_t$. $\Theta = \{[1, 0.5, 0.1, 0.05]\}$, $K = 3$, $w_t \sim \mathcal{N}(0, 0.02)$,$\hat{\Theta}_0 = \{2\}_{i=1}^4$ and $(\alpha, \beta) = (1, 0.5)$.

Artificial model The model contains 65 variables, an arbitrary compromise between scale and run time.

$$x_{1,t} = 0.2x_{1,t-1} + w_{1,t} \tag{35}$$

$$x_{i,t} = 0.2x_{i,t-1} + x_{\lfloor i/2 \rfloor,t-1} + w_{i,t} \quad \forall i > 1 \ . \tag{36}$$

Furthermore, $\Sigma_{w,t} = I$. Each hidden variable can be observed by 5 sensors. The variances of the noise of each set of 5 sensors $= \{1, 2, 3, 4, 5\}$ and $K = 98$ ($\approx$ 30% of the sensors). $\sigma_{i,0}^2 = 3 \ \forall i$. $(\alpha, \beta) = (1.5, 0.5)$.

Traffic in Dublin We apply our methods to the estimation of the saturation of 470 road lanes. We construct a LDS modeling the 5am to 12am evolution of the saturations of these road lines in Central Dublin to have a realistic system dynamic for evaluation purpose[3]. These saturation values correspond to x_t. 5 artificial sensors of variance $\{20, 50, 80, 110, 140\}$ can observe each hidden saturation value (2350 sensors in total). $K = 700$. $\sigma_{i,0}^2 = 100 \ \forall i$. $(\alpha, \beta) = (50, 0.5)$.

4.1 Results on Brownian Motion

Figure 2 compares the number of optimal choices made by our methods for all times. The problem is simple, so *EM* and *Gibbs* converge to the optimal, although *Gibbs* overexplores a bit. The online algorithms *EMo* and *VB* improve more slowly. *EMo* does not converge to the best choice every run and fares worse than *VB*.

Our online methods are orders of magnitude faster, as can be seen from the average run times listed in Table 1. The run times of *EM* and *Gibbs* in the other settings would be prohibitive, so they are not considered.

[3]using 4 months of data available at `http://dublinked.ie/datastore/datasets/dataset-305.php`

4.2 Results on Artificial Model

The accuracy $||x_t - \hat{x}_{t|\Theta_{0:t-1}y_{1:t}}||_{l_2}^2$ achieved by each method is illustrated in Figure 3(a). Both our online methods achieve an estimation error close to *oracle* after some time, and significantly better than *random* and *meanNoise*. The behavior of *EMo* and *VB* is similar, due to the relatively good initial estimates of Θ.

If these initial estimates are worse, as in the experiment presented in Figure 3(b), *EMo* can be worse than *random*, due to its greedy nature. *VB* is still better than *random* and *meanNoise* and converges to *oracle*.

4.3 State Estimation for Traffic in Dublin

Typical estimation errors are illustrated in Figure 4. Both *EMo* and *VB* are again significantly more accurate than *random* and *meanNoise*. The initial estimate of Θ was good: *EMo* is a bit more accurate than *VB*.

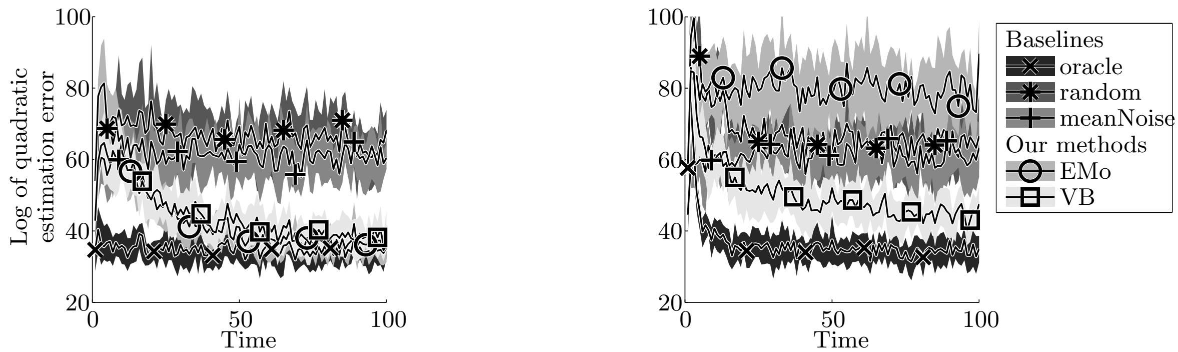
On 2013-01-22 (Figure 5), 4 SCATS recordings (20 selectable sensors) contain constant, aberrant values for several hours. Therefore, corresponding elements of Θ do not reflect the true variances of these 20 sensors. Because *oracle* uses Θ to make its decisions, it selects suboptimal sensors during this anomaly. Since our *VB* can adapt, it produces more accurate estimates than *oracle*. Appendix 7.6 contains results for all days.

5 DISCUSSION

When all parameters are known, submodular sensor selection comes with theoretical guarantees. A natural question is whether such guarantees can be extended to our algorithms. Answering this question is outside the scope of this paper, but here are a few insights.

Parameter estimation of EM and the online EM converge to a (local) maximum of the likelihood function. However, the quality of the parameter estimates still depends on the number of queries answered by each sensor. Therefore, estimates of sensors that are no longer selected cannot improve much. These methods are clearly not optimal. Possible improvements include penalizing the choice of a sensor by a monotonic function of the number of past queries and deriving confidence intervals over the noise parameters. We are only aware of one result related to the latter approach. Joglekar *et al.* (2013) derive confidence intervals in crowdsourcing settings where labels are binary. Developing the new tools for a similar analysis of our algorithms requires significant work.

There have long been substantial empirical evidence that Thompson sampling is competitive with the state of the art in many problems and recent theoretical op-



(a) With a good initial Θ estimate, the errors of *EMo* and *VB* converge to the errors of the oracle.

(b) With worse initial parameter estimates, *EMo* provides worse state estimates than *random* or *meanNoise*.

Figure 3: Quality of state estimation on the artificial model, for $\sigma^2_{i,0} = 3\ \forall i, \alpha = 1.5$ (left) and $\sigma^2_{i,0} = 10\ \forall i, \alpha = 5$ (right). Filled areas correspond to 0.25-0.75 quantiles of 20 simulations, dark lines with markers to the means.

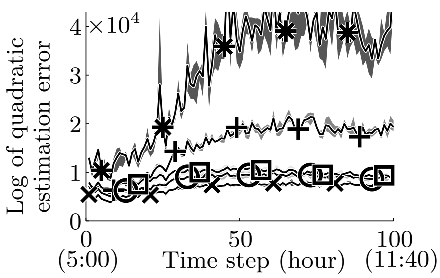


Figure 4: *EMo* and *VB* are more accurate than *random* and *meanNoise* on 2013-01-10.

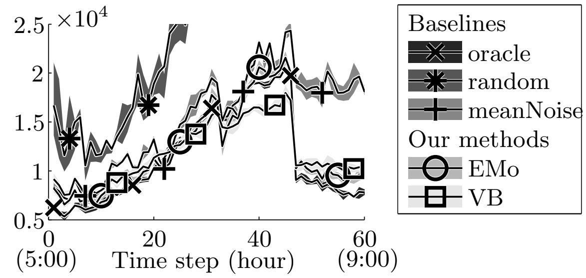


Figure 5: On 2013-01-22, faulty sensors lead *oracle* to take suboptimal decisions. *VB* adapts and is more accurate than *oracle*.

timality results for simple problems. Whether these results can be extended to our setting, with Gibbs sampling or with a VB approximation, is an open question.

Our online algorithms perform each update based on the last time step. They can be modified to operate on several past time steps, increasing estimation accuracy. Increasing the planning horizon Δ_T did not influence accuracy. This suggests that the transition noise is either evenly distributed or much higher than the uncertainty propagated in time.

6 CONCLUSION

Crowdsourced sensors are versatile, mobile, come in huge numbers and cost nothing to deploy but are typically unreliable and uncalibrated. Legal and economical concerns may limit the number of queries.

We develop real-time algorithms to decide which crowdsourced sensors should be queried to reduce the estimation error on a linear dynamical system the scale of a city. We develop Thompson sampling algorithms combining a submodular sensor selection procedure (Krause *et al.*, 2008) with two Bayesian parameter sampling algorithms. As an alternative, we also consider maximum-likelihood estimation algorithms. The algorithm based on online variational Bayes inference is the best candidate for the large-scale systems we target. Up to our knowledge, online variational inference has not been applied to dynamical systems before.

Our methods are not limited to LDS. They are easily applicable to problems where temporal independence is enforced, including the classical crowdsourcing setting. If the parameters of the system are unknown, our methods are also applicable to crowdsensing based on Gaussian processes (Krause *et al.*, 2008; Venanzi *et al.*, 2013), which result in simpler models than the LDS we use. If the objective function is not submodular, the optimisation subroutine must be changed, but the combination of Thompson sampling and online variational inference stays relevant.

Legal issues are currently being tackled before the deployment of our algorithms in a real city with the size of approximately a million people. Another attractive application is smart grids, for example querying smart meters to estimate the power consumed and the renewable electricity produced by each house.

Acknowledgements

This work is funded by the EU FP7 INSIGHT project (project number 318225).

References

Chapelle, O. and Li, L. (2011). An empirical evaluation of Thompson sampling. In *Advances in Neural Information Processing Systems*, pages 2249–2257.

Chen, J., Low, K. H., Tan, C. K.-Y., Oran, A., Jaillet, P., Dolan, J. M., and Sukhatme, G. (2012). Decentralized data fusion and active sensing with mobile sensors for modeling and predicting spatiotemporal traffic phenomena. In *Uncertainty in Artificial Intelligence*, pages 163–173.

Das, A. and Kempe, D. (2008). Algorithms for subset selection in linear regression. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 45–54. ACM.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, **39**, 1–38.

Donmez, P., Carbonell, J. G., and Schneider, J. G. (2010). A probabilistic framework to learn from multiple annotators with time-varying accuracy. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, volume 2, pages 826–837.

Hazan, E. and Kale, S. (2009). Beyond convexity: Online submodular minimization. In *Advances in Neural Information Processing Systems*, pages 700–708.

Hoang, T. N., Low, K. H., Jaillet, P., and Kankanhalli, M. (2014). Nonmyopic ϵ-Bayes-optimal active learning of gaussian processes. In *Proceedings of the 31st International Conference on Machine Learning*, pages 739–747.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research*, **14**, 1303–1347.

Joglekar, M., Garcia-Molina, H., and Parameswaran, A. (2013). Evaluating the crowd with confidence. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 686–694, New York, NY, USA. ACM.

Krause, A. and Guestrin, C. (2007). Nonmyopic active learning of gaussian processes: an exploration-exploitation approach. In *Proceedings of the 24th International Conference on Machine Learning*, pages 449–456. ACM.

Krause, A., Horvitz, E., Kansal, A., and Zhao, F. (2008). Toward community sensing. In *Proceedings of the 7th international conference on Information processing in sensor networks*, pages 481–492. IEEE Computer Society.

Krishnamurthy, V. and Moore, J. B. (1993). On-line estimation of hidden markov model parameters based on the kullback-leibler information measure. *Signal Processing, IEEE Transactions on*, **41**(8), 2557–2573.

Martinez-Cantin, R., de Freitas, N., Brochu, E., Castellanos, J., and Doucet, A. (2009). A bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot. *Autonomous Robots*, **27**(2), 93–103.

Meliou, A., Krause, A., Guestrin, C., and Hellerstein, J. M. (2007). Nonmyopic informative path planning in spatio-temporal models. In *Proceedings of the National Conference on Artificial Intelligence*, volume 22, pages 602–607.

Murphy, K. P. (2007). Conjugate bayesian analysis of the gaussian distribution. Technical Report $2\sigma2$, University of British Columbia.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. (1978). An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, **14**(1), 265–294.

Raykar, V., Yu, S., Zhao, L., Valadez, G., Florin, C., Bogoni, L., and Moy, L. (2010). Learning from crowds. *Journal of Machine Learning Research*, **99**, 1297–1322.

Sarkka, S. and Nummenmaa, A. (2009). Recursive noise adaptive kalman filtering by variational bayesian approximations. *Automatic Control, IEEE Transactions on*, **54**(3), 596–600.

Streeter, M. and Golovin, D. (2008). An online algorithm for maximizing submodular functions. In *Advances in Neural Information Processing Systems*, pages 1577–1584.

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, **25**(3/4), 285–294.

Venanzi, M., Rogers, A., and Jennings, N. R. (2013). Crowdsourcing spatial phenomena using trust-based heteroskedastic gaussian processes. In *First Conference on Human Computation and Crowdsourcing (HCOMP)*, pages 182–189. AAAI Press.

Wang, Y., Papageorgiou, M., and Messmer, A. (2008). Real-time freeway traffic state estimation based on extended kalman filter: Adaptive capabilities and real data testing. *Transportation Research Part A: Policy and Practice*, **42**(10), 1340 – 1358.

Wills, A., Schön, T. B., Lindsten, F., and Ninness, B. (2012). Estimation of linear systems using a Gibbs sampler. In *Proceedings of the 16th IFAC Symposium on System Identification (SYSID)*, pages 203–208.