MIT Open Access Articles

Efficient distributed sensing using adaptive censoring-based inference

Efficient Distributed Sensing Using Adaptive Censoring-Based Inference

Beipeng Mu   Girish Chowdhary   Jonathan P. How

[a]*77 Massachusetts Ave., Cambridge, MA 02139, USA.*

Abstract

In many distributed sensing applications it is likely that only a few agents will have valuable information at any given time. Since wireless communication between agents is resource-intensive, it is important to ensure that the communication effort is focused on communicating valuable information from informative agents. This paper presents communication-efficient distributed sensing algorithms that avoid network cluttering by having only agents with high *Value of Information* (VoI) broadcast their measurements to the network, while others *censor* themselves. A novel contribution of the presented distributed estimation algorithm is the use of an adaptively adjusted VoI threshold to determine which agents are informative. This adaptation enables the team to better balance between the communication cost incurred and the long-term accuracy of the estimation. Theoretical results are presented establishing the almost sure convergence of the communication cost and estimation error to zero for distributions in the exponential family. Furthermore, validation through numerical simulations and real datasets show that the new VoI-based algorithms can yield improved parameter estimates than those achieved by previously published hyperparameter consensus algorithms while incurring only a fraction of the communication cost.

Key words: Consensus, Bayesian Parameter Estimation, Conjugacy

1 INTRODUCTION

The increasing availability of compact sensing and processing hardware is fueling a trend in which networks of multiple low-cost unmanned autonomous agents collaborate to perform complex missions [1,2]. Examples of such missions include aerobiological sampling, persistent surveillance, formation control, distributed resource delivery, and target positioning [1,3–6]. The tasks in these missions often require the agents to collaboratively sense, estimate, or reach agreement on global parameters/states, such as the states of the environment or shared variables related to task settings and assignments [7–11]. However, while low-cost agents have the potential to yield benefits such as scalability, cost-saving, and resiliency, these agents typically have limited onboard computation and communication resources. Hence, efficient distributed inference algorithms are needed to ensure that agents can optimally utilize the limited onboard resources while collaboratively estimating global parameters/states. This paper develops an adaptive *Value of Information* (VoI) based distributed estimation framework that addresses the problem of estimating global parameters in presence of uncertainties using limited communication resources. Our framework enables significant communication cost savings with comparable estimation error than traditionally used consensus and has no no restriction on network topologies compared with graphical model based estimation frameworks.

Many distributed estimation algorithms use the notion of *consensus* to estimate the parameters/states of interest (e.g., [9–16]). In a typical consensus algorithm, an agent attempts to reach an agreement with its neighbors by performing a sequential update that brings its estimate closer to the states/parameters of (a subset of) all of its neighbors. This process asymptotically converges to the average of all agents' states/parameters under mild assumptions on the connectivity of the communication network formed by these agents. For example, Figure 1 depicts a situation in which several networked agents are estimating the distribution of a set of parameters θ. In a consensus framework, all agents would communicate their local parameters to reach consensus on a global estimate. The advantages of a consensus-based approach is that it is fully decentralized, and often requires little computational effort by each agent. However,

---

[1] Research Assistant, Dept. of Aeronautics and Astronautics, MIT `mubp@mit.edu`

[2] Postdoctoral Associate, Laboratory for Information and Decision Systems, MIT `girishc@mit.edu`

[3] Richard C. Maclaurin Professor of Aeronautics and Astronautics, MIT `jhow@mit.edu`
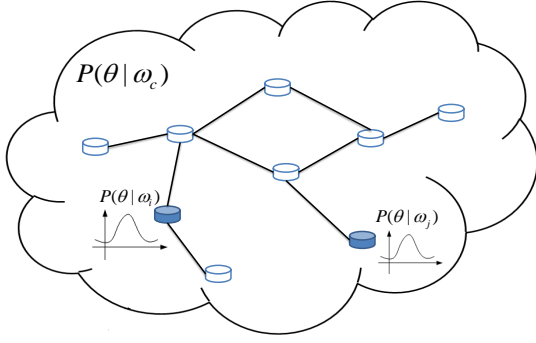
Fig. 1. Agents communicate local estimates $P(\theta|\omega_i)$ to estimate global parameter of interest θ. A consensus based algorithm requires all agents to communicate their estimates at all times to guarantee asymptotic convergence. However, note that if only two agents (e.g. the dark colored ones) have valuable information, then requiring all the other agents to keep communicating will result in wasted resources.

reaching consensus requires repeated and continuous communication, which can be resource-intensive and it is often the case that not all agents have valuable information to contribute at all times (e.g., the updated states/parameters after new measurements are obtained may not be sufficiently different from the preceding states/parameters, or not all agents are in a good position to take useful measurements). Thus, requiring all agents to communicate at all times can result in unnecessary communication that clutters the network with marginally useful information.

Revisiting Figure 1, we note that if only two agents (dark colored ones) have valuable information, then requiring all the other agents to keep communicating will result in wasted resources. One way to prevent network clutter is to censor (stop) uninformative agents from communicating. However, the consensus framework does not easily allow for dynamic censoring of uninformative agents (see Section 1.1 for further discussion on the literature on censoring and consensus). This possibly inefficient use of communication resources (and thus energy) could make the implementation of the standard consensus based algorithms difficult in real-world applications.

Another set of algorithms for distributed sensing relies on distributed Bayesian inference using graphical models (e.g., [7,8,17,18]). In graphical model based algorithms, agents build local probability models on the parameters of interest. When new measurements are observed, agents propagate messages between each other to update their probability models utilizing a priori known information about correlations between each other's probability models. Graphical model based algorithms are only guaranteed to work well on acyclic networks, because in that case there is only one path between any two agents, which guarantees that the messages are not duplicated. For an arbitrary network, one needs to use approximate algorithms (e.g., [19–24]), or implement additional algorithms to restructure the network into an acyclic network [25], which

brings in extra complexity.

1.1 Related Work on Efficient Distributed Sensing using Censoring

Many authors have explored the notion of *censoring* agents/measurements based on some VoI metric to reduce communication cost [26–33]. Censoring has been mainly explored for centralized estimation frameworks [27,28]. Cetin et al. have explored censoring in decentralized graphical model-based inference frameworks in the context of a data association problem [26]. In that work, messages are communicated only when the content exceeds a preset VoI threshold. The authors numerically show a significant reduction in communication cost by trading off some estimation accuracy but the paper does not provide theoretical insights on how to choose the VoI threshold.

In contrast, there appears to have been limited work on improving communication efficiency using censoring in the consensus literature. One possible reason for this is that it is not easy to directly apply censoring, such as in [26], to consensus formulations. Censoring agents would result in a dynamic network topology, which could adversely affect the convergence of baseline consensus-based algorithms. In particular, Oliva et al. have stated that adding an agent to a network engaged in consensus would still guarantee convergence to the unbiased global estimate, which is desirable, however, removing an agent from the network introduces a bias [34]. Saligrama et al. introduced a random censoring algorithm aimed at reducing communication cost in consensus based algorithms. In their algorithm, each agent randomly selects a neighbor and passes to it a "transmission permit (token)" [35]. In this way, the communication cost is reduced because not all agents are selected to communicate at all times. However, that work shows that consensus with only a subset of neighbors communicating takes longer to converge.

1.2 Motivation and Contribution

This research is motivated by the need to develop more communication-cost efficient algorithms for performing distributed estimation than those currently available. We present a Value of Information based Distributed Sensing (VoIDS) algorithm that achieves a significant overall reduction in network communication cost without sacrificing much accuracy. In VoIDS, agents take into account the VoI of the measurements as determined by an appropriate information theoretic VoI metric. The idea is similar to [26] in that agents identify themselves as *informative* and communicate their information only when the VoI exceeds a threshold. We go beyond [26] by theoretically showing that the choice of the VoI threshold results in a upper bound on estimation accuracy. This upper bound drives a dynamic trade-off between the cost of transmitting information, and the accuracy of the final estimate. To accommodate this trade-off, an Adaptive

VoI based Distributed Sensing (A-VoIDS) algorithm is introduced that adjusts the VoI threshold adaptively to ensure that the available communication bandwidth is optimally utilized to guarantee asymptotic reduction of estimation error.

Both VoIDS and A-VoIDS are theoretically and experimentally compared with a Full-Relay algorithm, a censoring-based Random Broadcast algorithm, and a Hyperparameter Consensus (HPC) algorithm [15]. Simulation shows that A-VoIDS incurs only a fraction of the communication cost of HPC, while arriving at an even better estimate of the hyperparameters. The algorithm is also tested on a real dataset (the Intel temperature dataset [36]), where similar results are obtained. Furthermore, strong theoretical results are established to guarantee almost sure convergence of the communication cost and the estimation error to zero for probability distributions in the exponential family. A notable advantage of both VoIDS and A-VoIDS is that they can work on any dynamic network topology, as long as the network remains strongly connected.

This work contributes to the goal of developing the next generation intelligent distributed sensing and distributed inference algorithms. It provides a more efficient framework for performing distributed parameter estimation than existing consensus or graphical model-based approaches (e.g., [8,13,37–42]). Furthermore, it is significant to the distributed inference literature because it extends the notion of *censoring* marginally useful information in a centralized estimation framework (e.g., [27,29–31]) to VoI-based self-censoring in a distributed inference framework. The algorithms discussed here, and their possible variants, could lead to significant resource savings in real-world distributed sensing applications by preventing irrelevant and marginally useful information from cluttering the network. A preliminary version of this work appeared in a conference [43]. The main contributions over that work here are significantly more in-depth mathematical and experimental analysis.

This paper is organized as follows. Section 2 introduces related probability, graph theory, and distributed inference concepts. Section 3 sets up the VoI metric and concepts of exponential families. Section 4 develops the VoIDS algorithm. Section 5 presents the A-VoIDS algorithm. Numerical simulation results are provided in Section 6, and the paper is concluded in Section 7.

2 BACKGROUND

2.1 Bayesian Inference

We use Bayesian framework to estimate the parameters of interest, because it can model the uncertainty of parameters with probability distributions and sequentially update the distributions with measurements of the parameters.

Let $\theta \in \mathbb{R}^d$ denote the parameters of interest, $p(\theta)$ denote the prior distribution, and $\mathbf{z} = \{z_1, z_2, \cdots, z_k\}$ denote a set

of measurements with the likelihood $p(\mathbf{z}|\theta)$. Bayes' theorem states that the posterior distribution $p(\theta|\mathbf{z})$ is (e.g., [44]):

$$p(\theta|\mathbf{z}) = \frac{p(\mathbf{z}|\theta)p(\theta)}{\int p(\mathbf{z}|\theta)p(\theta)\, d\theta}. \tag{1}$$

Consistency is one of the basic metrics on performance of estimation problems. It describes when unlimited measurements are used to update the posterior, whether the estimate will converge, and what it will converge to [45].

Definition 1 Assume the measurements are independent identically distributed (i.i.d.) drawn from the likelihood function $p(z|\theta_0)$ with parameter θ_o. The posterior distribution $p(\theta|\mathbf{z})$ is said to be consistent at parameter θ_0 if $p(\theta|\mathbf{z})$ converges to Dirac delta function $\delta(\theta_0)$ almost surely (a.s.) when the number of measurements that are used to update the posterior goes to infinity [45].

The Schwartz's consistency theorem outlines a sufficient condition for consistency of Bayesian inference:

Theorem 1 (Schwartz's consistency theorem, [45]). Let $p(x|\theta)$ be a class of probability distributions, $p(\theta)$ be a prior on θ, and $\{z_1, z_2, \cdots\}$ be i.i.d. measurements with likelihood function $p(x|\theta_0)$. Suppose for every neighborhood U of θ_0, every $\theta \in U$ satisfies

$$P\left(\theta : \int p(\theta_0) \log \frac{p(\theta_0)}{p(\theta)} d\theta_0 < \epsilon \right) > 0, \quad \forall \epsilon > 0$$

then the posterior over $\{z_1, z_2, \cdots\}$ is consistent at θ_0.

Theorem 1 will be used in this paper to test the error of our Bayesian inference framework.

In general, it is hard or nearly impossible to compute posterior because the integral $\int p(\mathbf{z}|\theta)p(\theta)\, d\theta$ has no closed-form solutions. However, in the case of exponential family distributions, an easily computable closed-form posterior exists, which gives us an easy way of updating posterior without computing the integral.

Let $p(\mathbf{x}|\theta)$ denote the probability distribution of random variables $\mathbf{x} \in \mathbb{R}^m$ under measure $h(d\mathbf{x})$, given parameters $\theta \in \mathbb{R}^d$. The exponential family is a set of probability distributions that follow the form [46]:

$$p(\mathbf{x}|\theta) = \exp\left\{\theta^T T(\mathbf{x}) - A(\theta)\right\}, \tag{2}$$

where $T(\mathbf{x}) : \mathbb{R}^m \to \mathbb{R}^d$ is the *Sufficient Statistic* or *Potential Function* and $A(\theta) = \ln \int \exp\left\{\theta^T T(\mathbf{x})\right\} h(d\mathbf{x})$ is the *Log Partition* or *Cumulant Function*. It is proven in [46] that $A(\theta)$ is positive, convex and in class C^∞ within its domain that is well-defined.

The exponential family distributions always have conjugate priors that give closed-form posterior solutions [44]. The conjugate priors are also within the exponential family, with hyperparameters of dimension $d+1$ [46]. Let $\omega \in \mathbb{R}^d, \nu \in \mathbb{R}$ denote the hyperparameters and $\Lambda(\omega, \nu)$ denote the conjugate prior's *Log Partition*, then the conjugate prior $p(\theta|\omega, \nu)$ has the following form under appropriate measure $f(d\theta)$:

$$p(\theta|\omega, \nu) = \exp\left\{\theta^T \omega - A(\theta)\nu - \Lambda(\omega, \nu)\right\}. \quad (3)$$

For above exponential family likelihood and conjugate prior, the posterior $p(\theta|\mathbf{z}, \omega, \nu)$ after n measurements $\mathbf{z} = \{z_i\}_1^n$ are observed always has a closed-form solution [47]:

$$p(\theta|\mathbf{z}, \omega, \nu) = \exp\left\{\theta^T(\omega + \sum T(z_i)) - A(\theta)(\nu + n),\right.$$
$$\left. -\Lambda(\omega + \sum T(z_i)), \nu + n)\right\}. \quad (4)$$

To simplify notations, define augmented vectors $\breve{\omega}_i = \left[\omega^T, \nu\right]^T, \breve{\theta} = \left[\theta^T, -A(\theta)\right]^T$ and $\breve{T}(\mathbf{z}) = \left[\left(\sum T(z_i)\right)^T, n\right]^T$. Then the prior and posterior can be rewritten as:

$$p(\theta|\breve{\omega}) = \exp\left\{\breve{\theta}^T \breve{\omega} - \Lambda(\breve{\omega})\right\}$$
$$p(\theta|\mathbf{z}, \breve{\omega}) = \exp\left\{\breve{\theta}^T(\breve{\omega} + \breve{T}(\mathbf{z})) - \Lambda(\breve{\omega} + \breve{T}(\mathbf{z}))\right\} \quad (5)$$

It can be seen that the posterior has the same form as the conjugate prior, only with an additive update in the hyperparameters:

$$\breve{\omega} = \breve{\omega} + \breve{T}(\mathbf{z}). \quad (6)$$

The following result can be proven from Theorem 1.

Corollary 1 If the likelihood is within the exponential family and the prior is conjugate to the likelihood, then the Bayesian inference is consistent.

Corollary 2 indicates that when the distribution is within the exponential family, the Bayesian posteriors can get closer estimates of the true parameters by taking more measurements. It will be used to develop theoretical guarantees of our algorithms.

2.2 Distributed Inference

First we define graphs that represents connections between agents and state some assumptions that will be used in different distributed inference algorithms later.

Algorithm 1 Full Relay

1: initiate global priors $p(\theta)$
2: **for** t **do**
3:     **for** each agent i **do**
4:         take measurement $\mathbf{z}_i[t]$
5:         broadcasts $\mathbf{z}_i[t]$ to neighbors
6:         relay each received new message $\mathbf{z}_j[t]$ to neighbors
7:     **end for**
8:     **for** each broadcast message $\mathbf{z}_i[t]$ **do**
9:         update the global posterior $p(\theta|\mathbf{z}_i[t])$
10:     **end for**
11: **end for**

Let graph $G\langle v, E\rangle$ represent a network of collaborating agents. Set $v = \{1, ..., N\}$ denotes vertices or agents of the network. Set E denotes edges, $E \subset v \times v$. Vertice pair $(i, j) \in E$ if and only if the agents i can communicate with, or otherwise sense, the state of agent j [13]. When $(i, j) \in E$, agent j is called a neighbor of agent i. The set of all of i's neighbors is defined as agent i's neighborhood, denoted by $\mathcal{N}_i$.

Assumption 1 Graph G is strongly connected. That is, for every vertics pair (i, j), there exists a path from i to j, which can be formed using elements in E.

Assumption 2 Every agent has a unique identifying label that it can transmit to differentiate its message from others.

Assumption 3 Relaying a message is much faster than obtaining a local measurement, processing it, and then broadcasting it.

2.2.1 Full Relay

Based on Assumptions 1–3, a naive method for distributed inference is that every time an agent gets a new measurement, it broadcasts the measurement to all of its neighbors. Furthermore, each agent relays messages for other agents. In this way, all agents have access to all the information from others, essentially allowing every agent to act as the *center* of the network.

Assume that the network is synchronized and the time is indexed by an integer $t \in \mathbb{N}$. Let $m_i[t]$ denote the number of measurements agent i gets at t, and $\mathbf{z}_i[t] = \{z_i^1[t], z_i^2[t], \cdots, z_i^{m_i[t]}[t]\}$ denote the measurements agent i takes at t, the Full Relay algorithm is given by Algorithm 1. It should be noted that this algorithm can be easily extended to asynchronous scenarios.

Cost: The Full Relay algorithm makes a copy of all measurements over each agent. This could lead to big waste in communication resources. Assume that the cost for an agent to broadcast one message to its neighbors is 1 unit. At every time step, each agent needs to broadcast its own message and relay messages for all other agents. The total number of messages every agent sends out at t is N. The step communication cost at each time t (total number of messages sent out by all agents at t) is therefore N^2.

4

Algorithm 2 Hyperparameter Consensus

1: initiate hyperparameters $\breve{\omega}[0]$
2: **for** t **do**
3:      **for** each agent i **do**
4:          take measurement $\mathbf{z}_i[t]$
5:          compute $\breve{T}(\mathbf{z}_i[t])$
6:          compute local hyperparameters
7:          $\breve{\omega}_i[t] = \breve{\omega}_i[t] + \frac{\breve{T}(\mathbf{z}_i[t])}{\beta_i}$
8:          run consensus algorithm
9:          $\breve{\omega}_i[t+1] = \breve{\omega}_i[t] + \epsilon \sum_{j \in \mathcal{N}_i} (\breve{\omega}_j[t] - \breve{\omega}_i[t])$
10:      **end for**
11: **end for**

2.2.2 Hyperparameter Consensus

In consensus-based methods, each agent computes an average value between its own estimation and estimations from its neighbors. At each time step, an agent only sends out its local estimate instead of relaying all messages for others. Consensus algorithms are proven to asymptotically converge to global averages (e.g., [11–13,41]).

One example of the consensus-based algorithms is Fraser et al.'s Hyperparameter Consensus (HPC) [15]. HPC works on parameter distributions in the exponential family, and performs consensus on hyperparameters. In addition to Assumptions 1 and 2, this algorithm further assumes that the network topology is known. This assumption can be restrictive in some scenarios, but can be relaxed by using topology identification algorithms (e.g., [48–50]).

Notations t and $\mathbf{z}_i[t]$ are defined the same as in Section 2.2.1. Let $\breve{\omega}_i[t]$ denote the augmented local hyperparameters of agent i at t. Further let $\beta = \{\beta\}_1^N$ denote the eigenvector of eigenvalue 1 of the corresponding adjacency matrix of the network graph; the algorithm is provided in Algorithm 2. Ref. [15] proves that the HPC posterior will asymptotically converge to the centralized Bayesian posterior.

Cost: Noting that at each time step, each agent sends out only one message containing an update of its local hyperparameters, the step communication cost of all agents at time t is N.

2.2.3 Random Broadcast

In order to avoid network-wide communication at all times, a random censoring procedure can be used. At every time step, each agent randomly becomes active and sends messages to others. The idea is similar to that in [35] in which each agent randomly select a neighbor to pass a communication token to.

After recording a measurement $\mathbf{z}_i[t]$, instead of broadcasting it immediately, agent i stores it in a local buffer. Define $S_i[t]$ as the sum of the *Sufficient Statistic* of buffered measurements, $n_i[t]$ as the number of buffered measurements

---

Algorithm 3 Random Broadcast

1: initiate hyperparameters $\breve{\omega}[0]$
2: **for** t **do**
3:      $\breve{\omega}[t] = \breve{\omega}[t-1]$
4:      **for** each agent i **do**
5:          take measurement and update local buffer
6:          $\breve{S}_i[t] = \breve{S}_i[t-1] + \breve{T}(\mathbf{z}_i[t])$
7:          **if** random number bigger than threshold ε **then**
8:             broadcasts $\breve{S}_i[t]$
9:             reset local buffer: $\breve{S}_i[t] = 0$
10:          **end if**
11:          relay each received new message $\mathbf{z}_j[t]$ to neighbors
12:      **end for**
13:      **for** each broadcast message $\breve{S}_j[t]$ **do**
14:          update the global posterior
15:          $\breve{\omega}[t] = \breve{\omega}[t] + \breve{S}_j[t]$
16:      **end for**
17: **end for**

---

of agent i and $\breve{S}_i[t] = \left[S_i^T[t], n_i[t] \right]^T$. Agent i sends out a message containing $\breve{S}_i[t]$ only when a locally generated random number between $[0, 1]$ exceeds a predefined threshold ε. The algorithm is described in Algorithm 3.

Cost: Noting that at each step the probability for an agent to send a message is ε, on average there will be εN agents broadcasting messages. Each message will be relayed by all the other agents, therefore on average the step communication cost would be εN^2. As all agents have a chance to send out their messages, the estimation error is continuously decreasing. By choosing smaller ε, the communication cost would be reduced. However, the convergence rate could also be reduced as agents communicate less frequently [35].

3 VALUE OF INFORMATION METRIC

The algorithms discusses so far communicate measurements across agents without differentiating the VoI of the measurements to the estimation task at hand. From the discussion in Section 1.1, a censoring strategy in which only high-value information is transmitted may lead to significant communication resource savings. This section first discusses the metrics of Value of Information (VoI) and their implementation in estimation problems. The VoI based Distributed Sensing (VoIDS) algorithm will be developed in next section.

3.1 Value of Information Metric

The idea of quantifying information dates back to Shannon's information theory [51]. Motivated by Shannon's entropy, Kullback and Leibler introduced the information measure on discrimination between two distributions, now known as the Kullback-Leibler (KL) divergence [52,53]. Renyi generalized KL divergence by introducing an indexed family of similar divergence measures [54]. Chernoff independently

introduced another family of information metric, known as Chernoff distance, which is different from Renyi divergence only by a multiplicative constant [55]. Further generalization beyond Renyi includes f-divergences (or Ali-Silvey divergences, [56]). These as well as some other metrics are listed in Table 1.

Table 1
Information Metrics

Metric	Formula
Kullback-Leibler	$D_{\text{KL}}(p\|q) = \int p \log(\frac{p}{q})dx$
Renyi	$D_\alpha(p\|q) = \frac{1}{\alpha-1}\log\int p^\alpha q^{1-\alpha}dx,\ \alpha > 1$
Chernoff	$D_c(p\|q) = \log\int p^\alpha q^{1-\alpha}dx$
f-divergence	$D_f(p\|q) = \int f(\frac{p}{q})dq(x)$
Varational	$V(p\|q) = \int |p - q|dx$
Matusita	$D_M(p\|q) = \left[\int |p^{\frac{1}{r}} - q^{\frac{1}{r}}|^r dx\right]^{\frac{1}{r}},\ r > 0$

$p(x)$ and $q(x)$ are two probability distributions

The metrics in Table 1 do not have closed-form solutions for general probability distributions. A VoI metric with a closed form solution is desirable, as it would allow VoI to be computed without requiring a costly sampling procedure. If the probability distribution is within the exponential family, Renyi divergence and related metrics have a closed-form solution, thus using Renyi divergence for VoI can help reduce computational cost. Note that KL divergence is Renyi divergence when $\alpha \to 1$. Here we pick KL divergence to be the metric on VoI in our problem. However, other VoI metrics can also be used with the algorithms developed later.

3.2 KL Divergence and Bayesian Inference

Recall that $p(\mathbf{z}|\theta)$, $p(\theta|\breve\omega)$ and $p(\theta|\mathbf{z}, \breve\omega)$ denote the likelihood, the prior distribution and the posterior distribution respectively. If the prior is conjugate to the likelihood as defined in (3), Nielsen and Nock show that the KL divergence between the prior and the posterior is [57]:

$$D_{\text{KL}}(p(\theta|\breve\omega)\|p(\theta|\mathbf{z}, \breve\omega)) \tag{7}$$
$$= \Lambda\left(\breve\omega + \breve{T}(\mathbf{z})\right) - \Lambda(\breve\omega) - \breve{T}(\mathbf{z})^T \nabla \Lambda(\breve\omega)$$

where ∇ represents gradient. Because $\Lambda(\breve\omega)$ is in the class C^∞ [46], $\Lambda\left(\breve\omega + \breve{T}(\mathbf{z})\right)$ can be expanded in a Taylor series around $\Lambda(\breve\omega)$:

$$D_{\text{KL}}(p(\theta|\breve\omega)\|p(\theta|\mathbf{z}, \breve\omega))$$
$$= \Lambda\left(\breve\omega + \breve{T}(\mathbf{z})\right) - \Lambda(\breve\omega) - \breve{T}(\mathbf{z})^T \nabla \Lambda(\breve\omega)$$
$$= \left\{\Lambda(\breve\omega) + \breve{T}(\mathbf{z})^T \nabla\Lambda(\breve\omega) + \int_0^{\breve{T}(\mathbf{z})} (\breve{T}(\mathbf{z}) - x)^T \nabla^2\Lambda(\breve\omega + x)dx\right\}$$
$$- \Lambda(\breve\omega) - \breve{T}(\mathbf{z})^T \nabla\Lambda(\breve\omega)$$
$$= \int_0^{\breve{T}(\mathbf{z})} (\breve{T}(\mathbf{z}) - x)^T \nabla^2\Lambda(\breve\omega + x)dx$$

$$= \frac{1}{2}\breve{T}(\mathbf{z})^T \nabla^2\Lambda(\breve\omega + \delta\breve\omega)\breve{T}(\mathbf{z}) \tag{8}$$

where $\delta\breve\omega \in \left[0, \breve{T}(\mathbf{z})\right]$. It is further proven in [46] that $\nabla^2\Lambda(\breve\omega + \delta\breve\omega) = \text{cov}(\breve\theta|\breve\omega + \delta\breve\omega)$, therefore,

$$D_{\text{KL}}\left(p(\breve\theta|\breve\omega)\|p(\breve\theta|\mathbf{z}, \breve\omega)\right) = \frac{1}{2}\breve{T}(\mathbf{z})^T \text{cov}(\breve\theta|\breve\omega + \delta\breve\omega)\breve{T}(\mathbf{z}) \tag{9}$$

Lemma 1 Assume the likelihood $p(x|\theta)$ is within the exponential family. Denote the conjugate prior as $p(\theta|\breve\omega)$, and posterior after taking n measurements $\mathbf{z} = \{z_i\}_1^n$ as $p(\theta|\breve\omega + \breve{T}(\mathbf{z}))$. If all measurements are i.i.d. drawn from a distribution with static parameter θ_0, $z_i \sim p(x|\theta_0)$, then $\lim_{n\to\infty} \text{cov}(\breve\theta|\breve\omega + \breve{T}(\mathbf{z})) \to 0$ a.s.

Proof 1 From corollary 1,

$$\lim_{n\to\infty} p(\theta|\breve\omega + \breve{T}(\mathbf{z})) \to \delta_{\theta_0} \quad \text{a.s.}$$

Then,

$$\lim_{n\to\infty} \text{cov}(\breve\theta|\breve\omega + \breve{T}(\mathbf{z}))$$
$$= \lim_{n\to\infty} \int \left[\breve\theta^2 - (\mathbb{E}\breve\theta)^2\right] p(\theta|\breve\omega + \breve{T}(\mathbf{z}))df(\theta).$$

Since $\breve\theta$ is a function of θ, we have

$$\lim_{n\to\infty} \text{cov}(\breve\theta|\breve\omega + \breve{T}(\mathbf{z}))$$
$$\to \int \left[\breve\theta^2 - (\mathbb{E}\breve\theta)^2\right] \delta_{\theta_0} df(\theta) \quad \text{a.s.}$$
$$= \breve\theta_0^2 - (\mathbb{E}\breve\theta_0)^2 \quad \text{a.s.}$$
$$= 0 \quad \text{a.s.} \tag{10}$$

since θ_0 is a static parameter.

3.3 Single Agent Case

Here we consider a network with a single agent to show how VoI can be used to improve the efficiency of distributed sensing. The VoI based Decentralized Sensing (VoIDS) algorithm for multiple agent network will be given in the next section.

At time t, the hyperparameter of the conjugate prior is:

$$\breve\omega[t-1] = \breve\omega[0] + \breve{T}(\mathbf{z}[1:t-1]) \tag{11}$$

Assume the agent takes $n[t]$ measurements $\mathbf{z}[t]$, $|\mathbf{z}[t]| = n[t]$. The following theorem formalizes the intuitive notion that as an agent takes more measurements, its estimate of the parameters improves, while the VoI in the new measurements $\mathbf{z}[t]$ decreases to zero.

Algorithm 4 VoI based Sensing for a Single Agent

1: initiate hyperparameters $\breve{\omega}[0]$
2: **for** t **do**
3:     $\breve{\omega}[t] = \breve{\omega}[t-1]$
4:     take measurement and update buffer:
5:         $\check{S}[t] = \check{S}[t-1] + \check{T}(z[t])$
6:     calculate VoI:
7:
$$V[t] = D_{\text{KL}}\left(p(\theta|\breve{\omega}[t])||p(\theta)|\breve{\omega}[t] + \check{S}[t])\right)$$
8:     **if** $V[t] > V^*$ **then**
9:         threshold reached, update posterior
10:            $\breve{\omega}[t] = \breve{\omega}[t-1] + \check{S}[t]$
11:        reset buffer
12:            $\check{S}[t] = 0$
13:    **end if**
14: **end for**

Theorem 2 Consider a single agent that takes a finite measurement at every time instance t and all the measurements are i.i.d. drawn from an exponential distribution with static parameters θ_0. At time t, define $V[t]$ as VoI of new measurement $\mathbf{z}[t]$, i.e, the KL divergence between the conjugate prior and posterior at t, then $\lim\limits_{|\mathbf{z}[1:t-1]|\to\infty} V[t] \to 0$ a.s.

Proof 2 From (9),

$$V[t] = \check{T}(\mathbf{z}[t])^T \text{cov}(\breve{\theta}|\breve{\omega}[t] + \delta\breve{\omega})\check{T}(\mathbf{z}[t])$$
$$= \check{T}(z[t])^T \text{cov}(\breve{\theta}|\breve{\omega}[0] + \delta\breve{\omega} + \check{T}(\mathbf{z}[1:t-1]))\check{T}(z[t])$$
$$\delta\breve{\omega} \in \left[0, \check{T}(\mathbf{z}[t])\right] \tag{12}$$

Given finite measurements $\mathbf{z}[t]$, *Sufficient Statistic* $T(\mathbf{z}[t])$ is finite, so vector $\check{T}(\mathbf{z}[t]) = \left[T(\mathbf{z}[t])^T, 1\right]^T$ is also finite. Furthermore, from Lemma 1, $\lim\limits_{|\mathbf{z}[1:t-1]|\to\infty} \text{cov}(\breve{\theta}|\breve{\omega}[0]+\delta\breve{\omega}+ \check{T}(\mathbf{z}[1:t-1])) \to 0$ a.s. Hence

$$\lim\limits_{|\mathbf{z}[1:t-1]|\to\infty} \check{T}(\mathbf{z}[t])^T \text{cov}(\breve{\theta}|\breve{\omega}[t] + \delta\breve{\omega})\check{T}(\mathbf{z}[t]) \to 0 \text{ a.s.}$$

that is $\lim\limits_{|\mathbf{z}[1:t-1]|\to\infty} V[t] \to 0$ a.s.

Now consider the case in which the agent does not update hyperparameters immediately after taking a new measurement, but instead stores the measurement in a local buffer and calculates the VoI first. The posterior is updated only when the VoI of the buffered measurements exceeds a threshold V^*. Denote $n[t]$ as the number of buffered measurements at t, $S[t]$ as the sum of the *Sufficient Statistic* of buffered measurements, and $\check{S}[t] = \left[S[t]^T, n[t]\right]^T$. Define t_k as the kth time the agent updates the posterior. This process is described in Algorithm 4.

The following result guarantees that if an agent uses Algorithm 4 for inference, then the frequency of the posterior

updates will decrease with time, because the VoI of new measurements will decrease with time due to Theorem 2.

Theorem 3 Consider the case where a single agent takes one measurement $z[t]$ at every time instance t and does inference according to Algorithm 4. Assume all the measurements are i.i.d. drawn from a static distribution with parameters θ_0, $z[t] \sim p(z|\theta_0)$. Let t_k be the kth time the agent updates the hyperparameters, and $n[t]$ be the number of measurements buffered at t, then $\lim\limits_{t\to\infty} n[t_k] \to \infty$ a.s.

Proof 3 From the definition of t_k, at any time instant t, t_{k-1}, $(t_{k-1} < t)$ represents the last time the agent updated the posterior, and t_k, $(t_k \geq t)$ represents the next time the agent will update the posterior.

Furthermore, $n[t_k]$ represents the number of measurements in the buffer at time t_k, i.e. the measurements taken between t_k and t_{k-1}. Since the agent only takes one measurement at every time step, $t_k = n[t_k] + t_{k-1}$. Therefore, $\lim\limits_{t\to\infty}(n[t_k] + t_{k-1}) = \lim\limits_{t\to\infty} t_k \geq \lim\limits_{t\to\infty} t \to \infty$. We have either $\lim\limits_{t\to\infty} n[t_k] \to \infty$ and/or $\lim\limits_{t\to\infty} t_{k-1} \to \infty$.
(i) In the first case, $\lim\limits_{t\to\infty} n[t_k] \to \infty$, the theorem holds.
(ii) Consider for the sake of contradiction that $n[t_k]$ is bounded, that is $\lim\limits_{t\to\infty} n[t_k] \leq C < \infty$. In this case, it follows that $\lim\limits_{t\to\infty} t_{k-1} \to \infty$. In other words, this means that at time t_k, the number of buffered measurements $(n[t_k])$ is bounded, but the number of measurements the agent has used to update the parameters at the previous step (t_{k-1}) goes to infinity. Since t_{k-1} is unbounded it follows from Theorem 2, $\lim\limits_{|\mathbf{z}[1:t_{k-1}]|\to\infty} V[t_k] \to 0$ a.s., which means $\mathbb{P}(V[t_k] > V^*) \to 0$. Therefore, there does not exist a finite time t_k such that $V > V^*$, hence $n[t_k]$ cannot be bounded, this is a contradiction.

Hence, it must follow that $\lim\limits_{t\to\infty} n[t_k] \to \infty$ a.s.

4 VoI based Distributed Sensing (VoIDS)

In this section we develop the VoI based Distributed Sensing (VoIDS) algorithm for a network of multiple sensing agents.

In VoIDS, agents start with the same global prior. This can be accomplished by either externally setting a prior to all agents or through communication between the agents to agree on a global prior, as is done in most distributed sensing algorithms without censoring (see Section 1.1). Similar to the single agent case, upon obtaining a new measurement, agent i records it into its local buffer instead of immediately broadcasting it to others. Denote $n_i[t]$ and $S_i[t]$ as number and sum of *Sufficient Statistic* of buffered measurements for agent i at time t, and let $\check{S}_i[t] = \left[S_i[t]^T n_i[t]\right]^T$. Denote $V_i[t]$ as the VoI of agent i's buffered measurements at t.

Algorithm 5 VoI based Distributed Sensing (VoIDS)

1: initiate hyperparameters $\breve{\omega}[0]$
2: **for** t **do**
3:    $\breve{\omega}[t] = \breve{\omega}[t-1]$
4:    **for** each agent i **do**
5:       take measurement and update local buffer
6:       $\breve{S}_i[t] = \breve{S}_i[t-1] + \breve{T}(z_i[t])$
7:       calculate VoI of current buffer
8:
$$V_i[t] = D_{\mathrm{KL}}\left(p(\theta|\breve{\omega}[t]) || p(\theta|\breve{\omega}[t] + \breve{S}_i[t])\right)$$
9:       **if** $V_i[t] > V^*$ **then**
10:          broadcasts $\breve{S}_i[t]$
11:          reset local buffer: $\breve{S}_i[t] = 0$
12:       **end if**
13:       relay each received new message $\mathbf{z}_j[t]$ to neighbors
14:       **for** each broadcast message $\breve{S}_j[t]$ **do**
15:          update the global posterior
16:          $\breve{\omega}[t] = \breve{\omega}[t] + \breve{S}_j[t]$
17:       **end for**
18:    **end for**
19: **end for**

The algorithm proceeds as follows. If $V_i[t]$ exceeds a predefined threshold V^*, agent i labels itself as informative, otherwise it labels itself as uninformative. All informative agents broadcast a message containing $\breve{S}_i[t]$ to their neighbors, then clear their local buffers and reset $\breve{S}_i[t]$ to zero. Uninformative agents censor themselves from broadcasting their own measurements. All agents relay every message they receive from an informative agent or a relaying agent. Since each agent has a unique identifying label, it can be ensured that messages are not duplicated during relay. By Assumption 1, 2, and 3, all agents are guaranteed to get updates of all informative agents. Then they update their estimates of the global posterior by adding relayed updates to their hyperparameters. The process is described in an algorithmic form in Algorithm 5.

The next theorem shows that the interval between two updates for any agents will go to infinity a.s., which means the average communication cost of each step will approach zero a.s. when using Algorithm 5.

Theorem 4 Consider a network of N agents performing distributed inference with Algorithm 5. Assume the measurements are i.i.d. drawn from a distribution with static parameters θ_0. Denote t_k^i as the kth time agent i sends out a message to update the global hyperparameters, then for any agent i, the number of measurements needed to exceed a predefined VoI threshold V^* will go to infinity, that is $\lim_{t\to\infty} n_i[t_k^i] \to \infty$.

Proof 4 First assume the case where agent i does not receive any messages from other agents after t_k^i. Define the time it sends out next message as $\tilde{t}_{k+1}^i = t_k^i + n_i[\tilde{t}_{k+1}^i]$. From Theorem 3, $\lim_{t\to\infty} n_i[\tilde{t}_k^i] \to \infty$, so Theorem 4 holds.

On the other hand, if agent i receives one or more messages from other agents between t_k^i and $\tilde{t}_{k+1}^i$, the global hyperparameters are updated between t_k^i and $\tilde{t}_{k+1}^i$, thus agent i would have used more measurements to update the hyperparameters by time $\tilde{t}_k^i$. This would only make $\mathrm{cov}(\breve{\theta}|\breve{\omega}[t])$ converge to 0 faster than in the first case due to Lemma 1. Hence in order to reach the same VoI threshold V^*, agent i needs to take more measurements. Denote t_{k+1}^i to be the time agent i sends out the next message, in this case, $n_i[t_{k+1}^i] \geq n_i[\tilde{t}_{k+1}^i]$. Since $\lim_{t\to\infty} n_i[t_k^i] \to \infty$, $\lim_{t\to\infty} n_i[t_k^i] \to \infty$.

Hence, in both cases, $\lim_{t\to\infty} n_i[t_k^i] \to \infty$.

At time t, let I denote the set of informative agents and $\bar{I}$ the uninformative agents. Define the estimation error $e[t]$ as KL divergence between global posterior and the centralized Bayesian posterior:

$$e[t] = D_{\mathrm{KL}}\left(p\left(\theta|\breve{\omega}[t] + \sum_{j\in I}\breve{S}_j[t]\right) || p\left(\theta|\breve{\omega}[t] + \sum_{i=1}^{N}\breve{S}_i[t]\right)\right)$$
(13)

The following theorem shows that the expectation of this error is bounded when using VoIDS (Algorithm 5).

Theorem 5 Consider a network of N agents that performs inference with Algorithm 5. At time instance t, if the error $e[t]$ is defined by (13), then $\mathbb{E}(e[t]) \leq N^2 V^*$.

Proof 5 For a single measurement z, denote $\mathbb{E}T = \mathbb{E}T(z)$ as the expected *Sufficient Statistic*. Then $\mathbb{E}\breve{S}_i[t] = n_i[t]\mathbb{E}T$. From (7), (9) and (13), take expectation of $\mathbb{E}(V_i[t])$ and $e[t]$ in terms of $\breve{S}_i[t]$:

$$\mathbb{E}(V_i[t]) = \int_0^{n_i[t]\mathbb{E}T} (n_i[t]\mathbb{E}T - x)^T \mathrm{cov}\left(\breve{\theta}|\breve{\omega}[t] + x\right) dx \quad (14)$$

Let $y = \mathbb{E}T - \frac{x}{n_i[t]}$

$$= (n_i[t])^2 \int_0^{\mathbb{E}T} y^T \mathrm{cov}\left(\breve{\theta}|\breve{\omega}[t] + n_i[t]\mathbb{E}T - n_i[t]y\right) dy$$

similarly for $e[t]$

$$\mathbb{E}(e[t]) \tag{15}$$
$$= \left(\sum_{i\in\bar{I}} n_i[t]\right)^2 \int_0^{\mathbb{E}T} y^T \mathrm{cov}(\breve{\theta}|\breve{\omega}[t] + \sum_{i=1}^{N} n_i[t]\mathbb{E}T - \sum_{i\in\bar{I}} n_i[t]y) dy$$

From Lemma 1, $\mathrm{cov}(\breve{\theta}|\breve{\omega}[t] + nx) = \nabla^2\Lambda(\breve{\omega}[t] + nx)$ converges to 0 when $n \to \infty$. It can be further proven from [46] that $\mathrm{cov}(\breve{\theta}|\breve{\omega}[t] + nx)$ is convex in n. Hence, $\mathrm{cov}(\breve{\theta}|\breve{\omega}[t] + nx)$ monotonically converges to 0. Therefore,

$$\int_0^{\mathbb{E}T} y^T \mathrm{cov}(\breve{\theta}|\breve{\omega}[t] + \sum_{i=1}^{N} n_i[t]\mathbb{E}T - \sum_{i\in\bar{I}} n_i[t]y) dy$$

$$\leq \int_0^{\mathbb{E}T} y^T \mathsf{cov}\left(\breve{\theta}|\breve{\omega}[t] + n_i[t]\mathbb{E}T - n_i[t]y\right) dy. \quad (16)$$

Furthermore $(\sum_{i \in \bar{I}} n_i[t])^2 \leq N^2 \max_{i \in \bar{I}}(n_i[t])^2$, we have:

$$\left(\sum_{i \in \bar{I}} n_i[t]\right)^2 \int_0^{\mathbb{E}T} x^T \mathsf{cov}\left(\breve{\theta}|\breve{\omega}[t] + \sum n_i[t]x\right) dx$$

$$\leq N^2 \max_{i \in \bar{I}} \left\{ (n_i[t])^2 \int_0^{\mathbb{E}T(z)} x^T \mathsf{cov}\left(\breve{\theta}|\breve{\omega}[t] + n_i[t]x\right) dx \right\}$$
$$(17)$$

Or, $\mathbb{E}(e[t]) \leq N^2 \max_{i \in \bar{I}} \mathbb{E}(V_i[t])$. Because $\mathbb{E}(V_i[t]) < V^*$ for agent $i \in \bar{I}$, we have $\mathbb{E}(e[t]) \leq N^2 V^*$.

5 Adaptive VoI-realized Distributed Sensing (A-VoIDS)

With a static VoI threshold V^*, the communication frequency of VoIDS was shown to decrease over time and the expected error was shown to be bounded by a constant. In particular, at the beginning of the estimation process, agents know little about the parameters of interest, hence new measurements tend to contain more information, so the set of informative agents is larger and there is more communication in the network. In contrast, at later stages of the estimation process, when agents have developed better estimates of the parameters, new measurements are less informative, so agents declare themselves as informative less frequently. While this means that the growth of the communication cost slows down, the error still remains bounded by V^* instead of continuing to decrease. Note that the number of agents declaring themselves as informative depends on V^*. Hence, for a real network that has a fixed communication bandwidth, the VoI threshold needs to be larger in the early stages of estimation to guarantee that the network is not overwhelmed, while in the later stages of the estimation, V^* must be dynamically reduced in order to guarantee continuous reduction of the estimation error. This implies that there is a dynamic tradeoff between the growth of cost and estimation error, and in a network with fixed communication bandwidth, the tradeoff can be handled by dynamically adjusting the value of V^*.

The Adaptive VoI based Distributed Sensing (A-VoIDS) algorithm discussed in this section provides a way to adaptively adjust the VoI threshold V^* to make most of the available communication bandwidth (defined by preset communication limits in a single time step). Because all agents will get messages from informative agents, all agents know the communication cost in the network at any given time step. Therefore it is possible for agents to update V^* in the same manner without introducing extra communication between them.

5.1 Adaptive VoI-realized Distributed Sensing Algorithm

Let indicator function $\mathcal{I}_{V_i[t]>V^*[t]}$ denote whether agent i is informative and sends out a message at time t. Let $C[t]$ denote the number of messages sent out at a single time step averaged among a past window of length l:

$$C[t] = \frac{1}{l} \sum_{j=t-l+1}^{t} \sum_{i=1}^{N} \mathcal{I}_{V_i[j]>V^*[j]}. \quad (18)$$

Variable $C[t]$ reflects the average step communication cost in a fixed length window. It should be noted that because the VoI of measurements taken by agents is not known a priori, the step cost $C[t]$ is a random variable. If $C[t]$ is too high, the communication cost will grow very rapidly, on the other hand if $C[t]$ is too low, then the error reduces very slowly. Therefore, it is desirable to regulate $C[t]$ around a reference value determined by the available communication bandwidth. A-VoIDS achieves this objective by dynamically adjusting the VoI threshold. In A-VoIDS, each agent compares the incurred $C[t]$ with a desired step-cost c, and adjusts $V^*[t]$ accordingly. If $C[t] < c$, the communication cost is lower than desired, which means that the available communication bandwidth is ill-utilized, hence the algorithm decreases V^* to encourage communication by setting $V^*[t+1] = \gamma_1 V^*[t]$, $0 < \gamma_1 < 1$ (mode 1 of the algorithm). If $C[t] \geq c$, the communication cost is higher than desired, so the algorithm increases V^* to limit communication by setting $V^*[t+1] = \gamma_2 V^*[t]$, $\gamma_2 > 1$ (mode 2). The above procedure used by the A-VoIDS algorithm is depicted in an algorithmic form in Algorithm 6. In the following theorem it is shown that the A-VoIDS algorithm guarantees that the estimation error asymptotically approaches zero almost surely.

Theorem 6 Consider a network of N distributed sensing agents. Assume that the measurements of all agents are i.i.d. drawn from a distribution with static parameters θ_0. Then the estimation error $e[t]$ as defined in (13) asymptotically reduces to zero a.s., that is $\lim_{t \to \infty} e[t] \to 0$ a.s.

Proof 6 Denote the probability distribution of $V^*[t]$ at time t as $p^t(v)$. At time t, define the probability of being in mode 1 as $\mathbb{P}_1[t|v] = \mathbb{P}(C[t] < c|v)$ and being in mode 2 as $\mathbb{P}_2[t|v] = \mathbb{P}(C[t] \geq c|v)$. From Theorem 4, for any given VoI threshold $V^* = v$, the interval between two consecutive updates of any agent i will increase to infinity, hence the probability of sending out a message at a particular time t will approach zero, i.e. $\lim_{t \to \infty} \mathcal{I}_{V_i[t]>v} \to 0$ a.s. Therefore, for a fixed window length l, the average cost $C[t]$ satisfies:

$$\forall v > 0, \quad \lim_{t \to \infty} C[t] = \lim_{t \to \infty} \frac{1}{l} \sum_{j=t-l+1}^{t} \sum_{i=1}^{N} \mathcal{I}_{V_i[j]>v} \to 0 \text{ a.s.}$$
$$(19)$$

Hence, over time the probability of being in mode 1 will approach 1, while being in mode 2 will approach 0, that is

$$\forall v > 0, \quad \lim_{t \to \infty} \mathbb{P}_1[t|v] = 1, \quad \text{and} \quad \lim_{t \to \infty} \mathbb{P}_2[t|v] = 0. \quad (20)$$

9

Algorithm 6 Adaptive VoI-realized Distributed Sensing (A-VoIDS)

1: initiate hyperparameters $\breve{\omega}[0]$
2: **for** t **do**
3:    **for** each agent i **do**
4:       $\breve{\omega}[t] = \breve{\omega}[t-1]$, $C[t] = 0$
5:       take measurement and update local buffer
6:       $\breve{S}_i[t] = \breve{S}_i[t-1] + \breve{T}(z_i[t])$
7:       calculate VoI of current buffer
8:       $V_i[t] = D_{\text{KL}}\left(p(\theta|\breve{\omega}[t])||p(\theta|\breve{\omega}[t] + \breve{S}_i[t])\right)$
9:       **if** $V_i[t] > V^*[t]$ **then**
10:          broadcasts $\breve{S}_i[t]$
11:          reset local buffer: $\breve{S}_i[t] = 0$
12:       **end if**
13:       relay each received new message $\mathbf{z}_j[t]$ to neighbors
14:       **for** each broadcast message $\breve{S}_j[t]$ **do**
15:          update the global posterior
16:          $\breve{\omega}[t] = \breve{\omega}[t-1] + \breve{S}_j[t]$
17:          step communication cost increased by 1
18:          $C[t] = C[t] + 1$
19:       **end for**
20:       adaptively change $V^*[t]$
21:       **if** $C[t] < c$ **then**
22:          smaller than bound, too little comm
23:          $V^*[t+1] = \gamma_1 V^*[t]$ $(0 < \gamma_1 < 1)$
24:       **else**
25:          bigger than bound, too much comm
26:          $V^*[t+1] = \gamma_2 V^*[t]$ $(\gamma_2 > 1)$
27:       **end if**
28:    **end for**
29: **end for**

For any given $\zeta > 0$, if $V^*[t+1] \geq \zeta$, there are two possibilities, $V^*[t] \geq \frac{\zeta}{\gamma_1}$ and the algorithm falls into mode 1 at t; or $V^*[t] \geq \frac{\zeta}{\gamma_2}$ and the algorithm falls into mode 2 at t, where γ_1, γ_2 are as defined in Algorithm 6. Therefore, we have

$$\mathbb{P}(V^*[t+1] \geq \zeta)$$
$$= \int_\zeta^\infty p^{t+1}(v)dv$$
$$= \int_{\frac{\zeta}{\gamma_1}}^\infty \mathbb{P}_1(t|v)p^t(v)dv + \int_{\frac{\zeta}{\gamma_2}}^\infty \mathbb{P}_2(t|v)p^t(v)dv \quad (21)$$

When $t \to \infty$, $\mathbb{P}_1 \to 1$, $\mathbb{P}_2 \to 0$, therefore taking limit w.r.t. time we have

$$\lim_{t\to\infty} \mathbb{P}(V^*[t+1] \geq \zeta)$$
$$= \lim_{t\to\infty} \int_{\frac{\zeta}{\gamma_1}}^\infty p^t(v)dv$$
$$= \lim_{t\to\infty} \mathbb{P}\left(V^*[t] \geq \frac{\zeta}{\gamma_1}\right). \quad (22)$$

From (22) it follows that

$$\lim_{t\to\infty} \mathbb{P}(V^*[t+1] \geq \zeta) - \mathbb{P}\left(V^*[t] \geq \frac{\zeta}{\gamma_1}\right) = 0. \quad (23)$$

Noting that $[t+\tau, t] = [t+\tau, t+\tau-1, \cdots, t+1, t]$, (23) can be rewritten by adding and subtracting intermediate terms

$$\lim_{t\to\infty} \mathbb{P}(V^*[t+\tau] \geq \zeta) - \mathbb{P}\left(V^*[t] \geq \frac{\zeta}{\gamma_1^\tau}\right)$$
$$= \lim_{t\to\infty}\left\{\mathbb{P}(V^*[t+\tau] \geq \zeta) - \mathbb{P}\left(V^*[t+\tau-1] \geq \frac{\zeta}{\gamma_1}\right)\right\}$$
$$+ \cdots$$
$$+ \lim_{t\to\infty}\left\{\mathbb{P}\left(V^*[t+2] \geq \frac{\zeta}{\gamma_1^{\tau-2}}\right) - \mathbb{P}\left(V^*[t+1] \geq \frac{\zeta}{\gamma_1^{\tau-1}}\right)\right\}$$
$$+ \lim_{t\to\infty}\left\{\mathbb{P}\left(V^*[t+1] \geq \frac{\zeta}{\gamma_1^{\tau-1}}\right) - \mathbb{P}\left(V^*[t] \geq \frac{\zeta}{\gamma_1^\tau}\right)\right\}$$

apply (23) to each of the limits

$$= 0 + 0 + \cdots + 0 = 0 \quad (24)$$

Now letting $\tau \to \infty$ we obtain:

$$\lim_{\tau\to\infty}\lim_{t\to\infty} \mathbb{P}(V^*[t+\tau] \geq \zeta) - \mathbb{P}\left(V^*[t] \geq \frac{\zeta}{\gamma_1^\tau}\right) = \lim_{\tau\to\infty} 0 \quad (25)$$

therefore,

$$\lim_{t\to\infty} \mathbb{P}(V^*[t] \geq \zeta) - \mathbb{P}(V^*[t] \geq \infty) = 0 \quad (26)$$

Because by definition of probability measures $\mathbb{P}(V^*[t] \geq \infty) = 0$, hence we have:

$$\forall \zeta, \quad \lim_{t\to\infty} \mathbb{P}(V^*[t] \geq \zeta) = 0 \quad (27)$$

Therefore, $\lim_{t\to\infty} V^*[t] \to 0$ a.s. From Theorem 5, $\lim_{t\to\infty} \mathbb{E}(e[t]) \to 0$ a.s. Since $e[t] \geq 0$, it follows that $\lim_{t\to\infty} e[t] \to 0$ a.s.

5.2 Comparison of Performance

Table 2 compares the communication cost in one time step and error of the algorithms discussed in this paper.

6 Experimental Evaluation

In this section, simulated data and a real dataset are used to compare the performance of VoIDS and A-VoIDS with existing distributed sensing algorithms (Full Relay, Random Broadcast, and HPC, see Section 2 for details) in terms of the communication cost incurred and the error to the centralized Bayesian estimate (which is assumed to be the truth).

Table 2
performance Summary

Algorithm	Cost in a step	Error
Full Relay	N^2	0
HPC	N	converges to 0
Random Broadcast	ϵN^2, ϵ tunable	random; slowly converges to 0
VoIDS	converge to 0	bounded
A-VoIDS	cN, c tunable	quickly converges to 0

6.1 Evaluation Using Simulated Dataset

The presented simulation considers a group of collaborative agents estimating the Poisson arrival rate λ of an entity. The prior distribution of λ is chosen to be a Gamma distribution $\Gamma(\alpha, \beta)$, which is conjugate to Poisson. The Poisson and Gamma distributions are given in (28) and (29) respectively:

$$p(z|\lambda) = \frac{(\lambda)^z e^{-\lambda}}{z!} \tag{28}$$

$$p(\lambda|\alpha, \beta) = \frac{\beta^\alpha \lambda^{\alpha-1} e^{-\beta\lambda}}{\Gamma(\alpha)} \tag{29}$$

This conjugacy results a closed-form update of hyperparameters when a measurement z is taken:

$$\begin{aligned} \alpha &= \alpha + z \\ \beta &= \beta + 1 \end{aligned} \tag{30}$$

The total number of agents in the network is set to be 100. At each time step t, every agent i takes one measurement $z_i[t] \sim Poi(\lambda_i)$. The local arrival rate parameters λ_i are biased from the true global value $\lambda = 5$ with uniform noise: $\lambda_i \sim U(4,6)$. For VoIDS, the VoI threshold are chosen to be $V^* = 0.02, 0.1, 0.5$ respectively. For A-VoIDS, the parameters are set to $\gamma_1 = 0.97$, $\gamma_2 = 1.01$ $l = 30$, $V^*[0] = 0.5$, and two communication bandwidths are tested $c = 0.10, 0.05$.

Figure 2 and 3 show the cumulative cost (the sum of all step costs up to current time) and the error to centralized Bayesian estimate of Full Relay, HPC, Random Broadcast, VoIDS and A-VoIDS algorithms (error shown in Figure 3 is smoothed over a window with $l = 30$).

Since the step communication costs of consensus-based approach (HPC), Full Relay, and Random Broadcast are constant (see Section 2), the cumulative costs of these algorithms increase linearly. The cost of HPC is significantly less than Full Relay, the cost of Random Broadcast is less than HPC for the chosen probability of communicating/censoring (see Section 2.2.3). Full Relay converges to the centralized Bayesian estimation immediately and has zero error, however, its communication cost is the highest of all the algorithms. Both HPC and Random Broadcast continuously reduce their estimation error, implying asymptotic convergence.
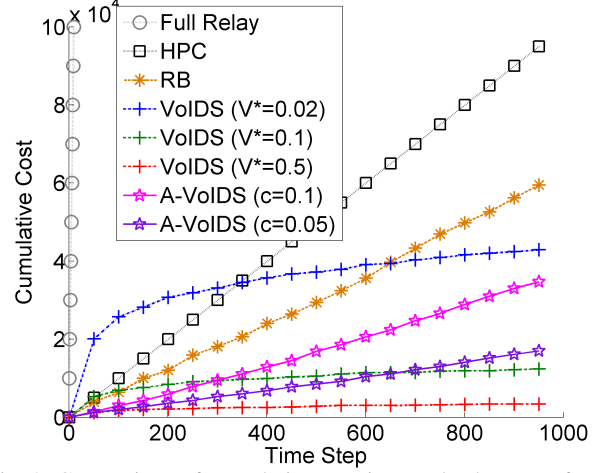

Fig. 2. Comparison of cumulative cost incurred. The cost of HPC and Random Broadcast is less than Full Relay. The cost of VoIDS can be more than HPC at the beginning, but levels off quickly. Cost of A-VoIDS grows continuously, but its rate of growth is controlled.
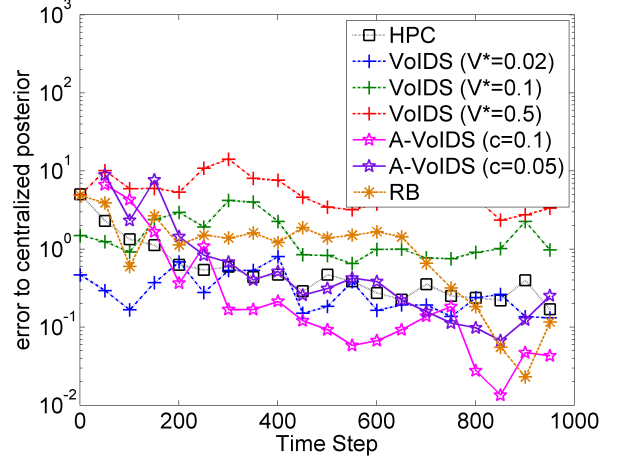

Fig. 3. Comparison of error to centralized posterior. Full Relay converge to the correct answer immediately and has zero error (not shown). VoIDS is not able to continuously reduce the error, while HPC, Random Broadcast and A-VoIDS are able to.

As proven by Theorem 4, the cumulative cost of VoIDS tends to grow quickly at the beginning (cost of VoIDS with lower broadcast threshold V^* can be higher than the cost of HPC and Random Broadcast at the beginning), however it levels off gradually as the step cost (18) approaches zero. On the other hand, VoIDS has a steady state error because the fixed V^* threshold prevents communication after some time into the simulation, thus cannot further reduce the error. In particular, with a high V^* threshold, agents have less communication cost but higher estimation error, and with a lower V^* threshold, agents have lower estimation error but higher communication cost. This indicates a dynamic tradeoff between communication cost and estimation error.

On the other hand, A-VoIDS (Algorithm 6) strikes a better balance between communication cost and estimation error. The cumulative cost of A-VoIDS also increases linearly, but
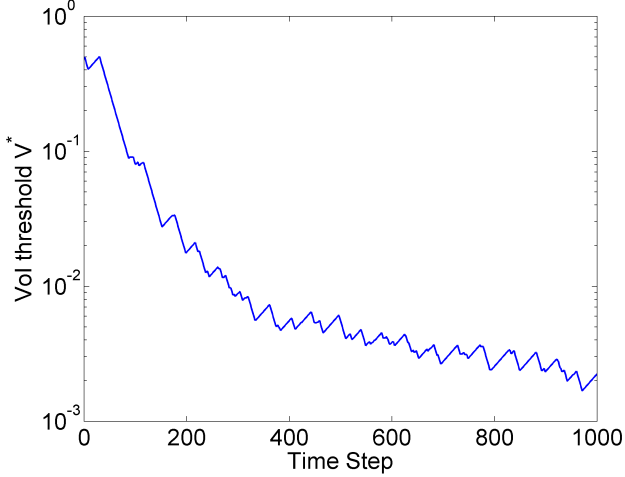
Fig. 4. VoI threshold V^* in A-VoIDS ($c = 0.10$) asymptotically drops to 0, as postulated by Theorem 6. Since the error is bounded by V^* (Theorem 5), the estimation error also approaches 0.
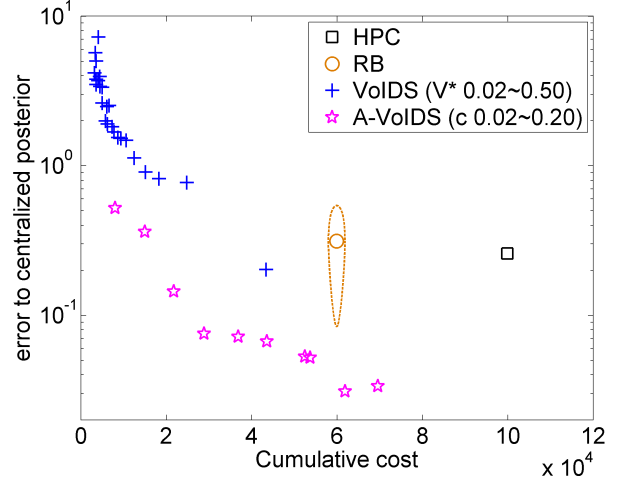

Fig. 5. Error to Centralized Posterior vs Cumulative Cost. A-VoIDS is closest to the left bottom corner, indicating it has less communication cost and less error.

the rate of growth can be tuned via c to reflect the available communication bandwidth. The cost of A-VoIDS is observed to be less than HPC for the chosen parameters. The evolution of V^* for A-VoIDS ($c = 0.10$) is shown in Figure 3, and it can be seen that V^* drops to zero as shown by Theorem 6. This indicates that unlike VoIDS, A-VoIDS tends to asymptotically reduce the error, since the estimation error is bounded above by V^*.

The performance of the algorithms discussed is compared in Figure 5 in cost-error coordinates. The horizontal axis represents the final cost at the end of the simulation and the vertical axis represents the average error to the centralized estimate of the hyperparameters (the centralized estimate converges to the correct hyperparameters) in the last 300 time steps. An ideal algorithm would be situated in the bottom-left corner of that graph, with low error and low communication cost. HPC is situated in the bottom-right corner, with low error but high communication cost. VoIDS with bigger V^* thresholds (e.g., $V^* = 0.5$) is in the top-left corner, with low communication cost (because the agents do not declare themselves as informative easily) but high error. When V^* is set to lower values, the algorithm results in lower error but higher communication cost. The Random Broadcast algorithm does a trade off between cost and error, however the performance is not always consistent due to the randomness in which agents broadcast the measurement. The bronze circle represents the average performance of 100 runs, and the dashed circle around it indicates one standard deviation of the cost and error. Several instances of A-VoIDS are plotted for different values of c, and increasing c will result in increased communication cost but lower estimation error. The general trend observed is that A-VoIDS dominates the bottom left half of the figure when compared to other algorithms. This indicates that A-VoIDS is capable of achieving excellent estimates without incurring high communication cost.
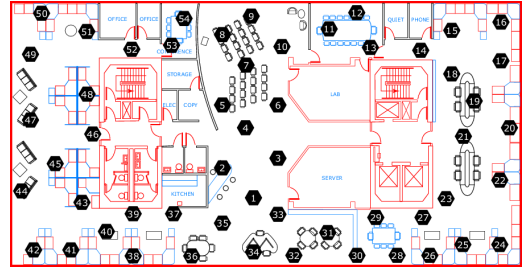
Fig. 6. Sensor Layout in the Intel Berkeley lab, 54 distributed sensors collect time stamped information such as temperature, humidity, light etc.

6.2 Evaluation Using Real Dataset

The VoIDS and A-VoIDS algorithms are further evaluated using a real dataset that has been used to analyze distributed sensing algorithms [32,58]. In this dataset, 54 sensors distributed in the Intel Berkeley Research lab collect times-tamped information such as humidity, temperature, light, and voltage values every half minute [32,58]. The sensor layout is shown in Figure 6. This dataset reflects real effects such as sensor noise, sensor bias, and time varying albeit slowly drifting parameters. The temperature data was selected for evaluating the algorithm. Over shorter intervals (e.g. an hour), the drift in temperature in a climate-controlled room is typically small (within $0.2°C$) and can be assumed to be approximately constant. Therefore a record of about an hour's temperature measurements is selected to evaluate the algorithms. The positive results reported in this section indicate that the algorithms tend to work well even when the parameters to be estimated are slowly varying instead of being constant as is assumed in the theoretical development.

The goal is to collaboratively estimate the the average room temperature denoted by θ. We model the noisy sensor measurements by each sensor using a Gaussian noise model with constant variance, $z \sim \mathcal{N}(\theta, 1)$. Since the conjugate prior of

a Gaussian distribution is also Gaussian, the Gaussian sensor noise model allows for a closed form update of the hyperparameters. Hence, the average room temperature can be modeled by $\theta \sim \mathcal{N}(\frac{\omega}{\nu}, \frac{1}{\nu})$, where ω, ν are the hyperparameters of the Gaussian conjugate prior. When a new measurement z is taken by a sensor, it updates its hyperparameters as in [57]:

$$
\begin{aligned}
\omega &= \omega + z, \\
\nu &= \nu + 1.
\end{aligned}
\tag{31}
$$

The rate at which the sensors check for or relay messages is a tunable parameter in this scenario, and it can affect the performance of HPC. In particular, increasing this rate tends to increase the speed of HPC error reduction but also increases HPC communication cost by increasing the number of messages sent out. In the presented results, we compare HPC's performance over a range of message communication rates: 1Hz, $\frac{1}{2}$Hz, $\frac{1}{3}$Hz, and $\frac{1}{4}$Hz. Note that the communication rate does not affect the number of messages sent out when sensors are running VoIDS or A-VOIDS.

The performance of different algorithms is compared in Figure 7 for different values of communication rate for HPC, V^* for VoIDS, and c for A-VoIDS. At the end of the evaluation run, HPC (1 Hz) results in an estimate with the least error but highest communication cost of all HPC runs. Decreasing the communication rate decreases HPC communication cost, but the error increases. VoIDS significantly reduces the communication cost compared to HPC, but also has a larger error. The Random Broadcast reduces cost by randomly censoring agents, but its performance has high variance and is no better than VoIDS on average. As before, A-VoIDS with higher values of c are situated closest to the bottom-left corner of the error-cost figure, indicating that A-VoIDS can give an estimate with significantly less communication cost than other algorithms that have similar error.

Figure 8 shows the estimated room temperature over time. The horizontal axis represents the time in seconds, and the vertical axis represents the temperature in Celsius. The blue solid line shows the centralized Bayesian estimate. The performance of the Random Broadcast algorithm has a lot of variance. The error between HPC (1Hz) and the centralized estimate drops within 0.1°C within first 500 seconds and keeps decreasing. After 1500 sec, the error is within 0.05°C. VoIDS (with $V^* = 0.1$) estimation error also drops within 0.1°C in 500 sec, but does not further decrease, even after 2000 sec, the error is still as much as 0.1°C. On the other hand, A-VoIDS ($c = 0.1$) starts with larger error than VoIDS ($V^* = 0.1$), but the error quickly decreases, and over time the A-VoIDS error is less than that of HPC.

7 CONCLUSION

In this paper, Value of Information (VoI) realized Distributed Sensing (VoIDS) algorithms are discussed in the
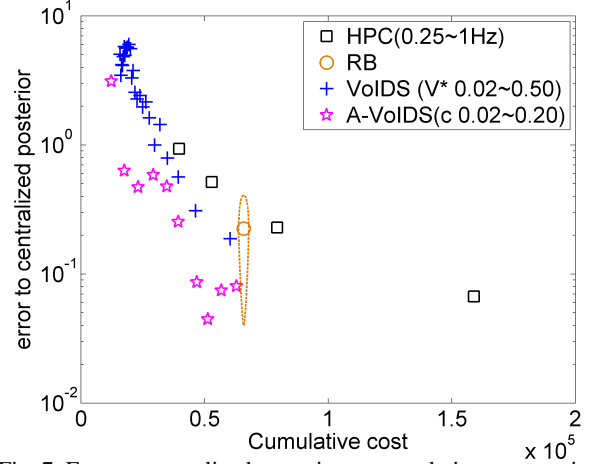

Fig. 7. Error to centralized posterior vs cumulative communication cost on the Intel dataset [32,58]. A-VoIDS tends to cluster near bottom left half of the figure, indicating that it can strike an excellent balance between accuracy and communication cost. It can be seen that A-VoIDS outperforms HPC in cost-error coordinates for various values of communication rates.
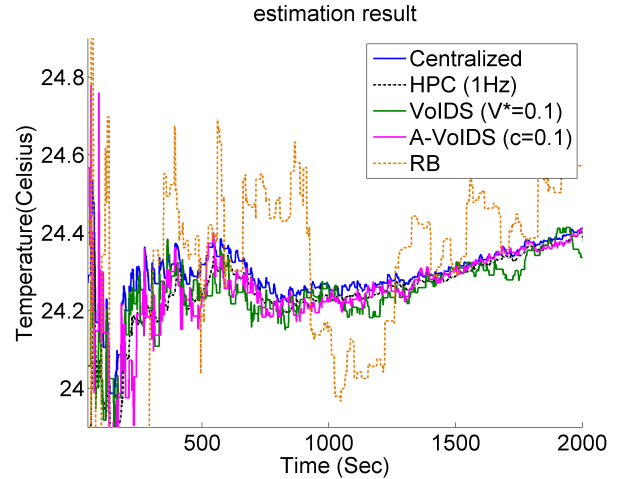

Fig. 8. Estimated room temperature by different algorithms when new measurements are taken continuously. The centralized estimate is believed to be the truth. The performance of the RB algorithm has a lot of variance. HPC (1Hz) error drops within 0.1°C within first 500 sec and keeps decreasing to less than 0.05°C. VoIDS ($V^* = 0.1$) estimate also drops within 0.1°C in the first 500 sec, but is unable to decrease further. A-VoIDS ($c = 0.1$) starts with a larger error than VoIDS ($V^* = 0.1$), but the error drops quickly to within 0.05°C, which is better than that of both HPC and VoIDS.

framework of Bayesian inference. VoIDS algorithms are designed to overcome known shortcomings such as excessive communication cost and slow convergence speed of traditional consensus-based algorithms. Furthermore, VoIDS algorithms do not require the knowledge of network topology, and are not limited to acyclic networks. Therefore, they also alleviate known limitations of inference algorithms based on graphical models. However, it is shown that the fixed VoI threshold leads to a dynamic tradeoff between estimation accuracy and communication cost. An Adaptive-

VoIDS (A-VoIDS) algorithm is presented and shown to guarantee almost sure convergence to zero estimation error while attempting to exploit all of the available communication bandwidth. Numerical simulations indicate that the A-VoIDS algorithm incurred only fraction of the communication cost of HPC, a consensus-based inference algorithm, and Random Broadcast, an algorithm with random censoring procedure, but arriving at an even better estimate of the hyperparameters than both of them. The algorithms are also tested on experimental data by using the Intel temperature dataset [36], where similar results are achieved. The algorithms discussed here, and their possible variants (including Kalman filter like variants for estimating dynamic states), would translate to significant resource savings in real-world distributed sensing applications by preventing irrelevant and marginally useful information from cluttering the network. This work therefore contributes towards developing the next generation efficient and accurate information extraction systems.

ACKNOWLEDGMENTS

References

[1] B. Mockenhaupt, "We've seen the future, and it's unmanned," *Esquire*, September 2009.

[2] "Unmanned aircraft systems roadmap: 2007–2032," Office of the Secretary of Defense, Tech. Rep., 2007.

[3] F. Zhang, D. Fratantoni, D. Paley, J. Lund, and N. Leonard, "Control of coordinated patterns for ocean sampling," *International Journal of Control*, vol. 80, no. 7, pp. 1186–1199, 2007.

[4] L. Techy, C. Woolsey, and D. Schmale, "Path planning for efficient uav coordination in aerobiological sampling missions," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*. IEEE, 2008, pp. 2814–2819.

[5] N. Michael, E. Stump, and K. Mohta, "Persistent surveillance with a team of MAVs," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 2708–2714.

[6] X. Wang, V. Yadav, and S. N. Balakrishnan, "Cooperative UAV formation flying with obstacle/collision avoidance," *Control Systems Technology, IEEE Transactions on*, vol. 15, no. 4, pp. 672–679, 2007.

[7] F. Bourgault, T. Furukawa, and H. F. Durrant-Whyte., "Decentralized Bayesian negotiation for cooperative search." *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.

[8] A. Makarenko, A. Brooks, T. Kaupp, H. Durrant-Whyte, and F. Dellaert, "Decentralised data fusion: A graphical model approach," in *Information Fusion, 2009. FUSION '09. 12th International Conference on*, july 2009, pp. 545 –554.

[9] R. Murray, "Recent research in cooperative control of multi-vehicle systems," *ASME Journal of Dynamic Systems, Measurement, and Control*, 2007.

[10] Y. Cao, W. Ren, N. Sorensen, L. Ballard, A. Reiter, and J. Kennedy, "Experiments in consensus-based distributed cooperative control of multiple mobile robots," in *International Conference on Mechatronics and Automation*, Aug. 2007, pp. 2819–2824.

[11] R. Olfati-Saber, A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 95, no. 1, pp. 215–233, January 2007.

[12] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, pp. 2508 – 2530, June 2006.

[13] M. Egerstedt and M. Mesbahi, *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[14] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, "Distributed Kalman filtering using consensus strategies," in *IEEE Conference on Decision and Control (CDC)*, Dec. 2007, pp. 5486–5491.

[15] C. S. Fraser, L. F. Bertuccelli, H.-L. Choi, and J. P. How, "A hyperparameter consensus method for agreement under uncertainty," *Automatica*, vol. 48, no. 2, pp. 374–380, February 2012.

[16] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Systems Magazine*, vol. 27, no. 2, pp. 71–82, April 2007.

[17] J. Yedidia, W. Freeman, and Y. Weiss, *Exploring Artificial Intelligence in the New Millennium*. Science and Technology Books, 2003.

[18] S. Grime and H. Durrant-Whyte, "Data fusion in decentralized sensor networks," *Control Engineering Practice*, vol. 2, no. 5, pp. 849 – 863, 1994.

[19] A. Makarenko, A. Brooks, S. Williams, H. Durrant-Whyte, and B. Grocholsky, "A decentralized architecture for active sensor networks," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, 2004, pp. 1097–1102.

[20] T. Bailey, S. Julier, and G. Agamennoni, "On conservative fusion of information with unknown non-gaussian dependence," in *Information Fusion (FUSION), 2012 15th International Conference on*, july 2012, pp. 1876 –1883.

[21] K. Chang, C.-Y. Chong, and S. Mori, "Analytical and computational evaluation of scalable distributed fusion algorithms," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 46, no. 4, pp. 2022 –2034, oct. 2010.

[22] W. Farrell and C. Ganesh, "Generalized chernoff fusion approximation for practical distributed data fusion," in *Information Fusion, 2009. FUSION '09. 12th International Conference on*, july 2009, pp. 555 –562.

[23] S. Julier, "An empirical study into the use of chernoff information for robust, distributed fusion of gaussian mixture models," in *Information Fusion, 2006 9th International Conference on*, july 2006, pp. 1 –8.

[24] N. Ahmed, J. Schoenberg, and M. Campbell, "Fast weighted exponential product rules for robust general multi-robot data fusion," in *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2012.

[25] M. A. Paskin, "Exploiting locality in probabilistic inference," Ph.D. dissertation, Berkeley, CA, USA, 2004, aAI3165519.

[26] M. Cetin, L. Chen, J. Fisher III, A. Ihler, R. Moses, M. Wainwright, and A. Willsky, "Distributed fusion in sensor networks," *Signal Processing Magazine, IEEE*, vol. 23, no. 4, pp. 42–55, 2006.

[27] W. P. Tay, J. Tsitsiklis, and M. Win, "Asymptotic performance of a censoring sensor network," *Information Theory, IEEE Transactions on*, vol. 53, no. 11, pp. 4191 –4209, nov. 2007.

[28] E. Msechu and G. Giannakis, "Distributed measurement censoring for estimation with wireless sensor networks," in *Signal Processing Advances in Wireless Communications (SPAWC), 2011 IEEE 12th International Workshop on*, june 2011, pp. 176 –180.

[29] ——, "Distributed measurement censoring for estimation with wireless sensor networks," in *IEEE 12th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, June 2011, pp. 176 –180.

[30] W. P. Tay, "Decentralized detection in resource-limited sensor network architectures," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 2008.

[31] M. UNEY, "Decentralized eestimation under ccommunication constraints," Ph.D. dissertation, MIDDLE EAST TECHNICAL UNIVERSITY, Turkey, 2009.

[32] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, and S. Madden, "Distributed regression: an efficient framework for modeling sensor network data," in *Information Processing in Sensor Networks, 2004. IPSN 2004. Third International Symposium on*, april 2004, pp. 1 – 10.

[33] C. Kreucher, A. Hero, K. Kastella, and M. Morelande, "An information-based approach to sensor management in large dynamic networks," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 978 –999, may 2007.

[34] G. Oliva, S. Panzieri, A. Priolo, and G. Ulivi, "Adding and removing nodes in consensus," in *Control Automation (MED), 2012 20th Mediterranean Conference on*, july 2012, pp. 1031 –1036.

[35] V. Saligrama and M. Alanyali, "A token-based approach for distributed computation in sensor networks," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 4, pp. 817 –832, aug. 2011.

[36] C. Guestrin, A. Krause, and A. P. Singh, "Near-optimal sensor placements in gaussian processes," in *International Conference on Machine Learning (ICML)*, 2005.

[37] D. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proceeedings of the IEEE*, vol. 85, no. 1, pp. 6–23, 1997.

[38] T. Berg and H. Durrant-Whyte, "Distributed and decentralized estimation," in *Intelligent Control and Instrumentation, 1992. SICICI '92. Proceedings., Singapore International Conference on*, vol. 2, feb 1992, pp. 1118 –1123.

[39] J. Ferber, *Multi-Agent Systems : An Introduction to Distributed Artificial Intelligence*. New York: Addison-Wesley, 1999.

[40] P. K. Varshney, *Distributed Detection and Data Fusion*. Springer-Verlag New York Inc., 1997.

[41] V. Gupta, "Distributed estimation and control in networked systems," Ph.D. dissertation, California Institute of Technology, 2006.

[42] R. Olfati-Saber, E. Franco, E. Frazzoli, and J. S. Shamma, "Belief consensus and distributed hypothesis testing in sensor networks," *Workshop of Network Embedded Sensing and Control*, vol. October, 2005.

[43] B. Mu, G. Chowdhary, and J. P. How, "Efficient distributed information fusion using adaptive decentralized censoring algorithms," in *American Control Conference (ACC)*. Washington DC: IEEE, July 2013 (accepted).

[44] A. Gelman, J. Carlin, H. Stern, and D. Rubin, *Bayesian Data Analysis*. Chapman and Hall, 1995.

[45] L. Schwartz, "On bayes procedures," *Probability Theory and Related Fields*, vol. 4, pp. 10–26, 1965, 10.1007/BF00535479. [Online]. Available: http://dx.doi.org/10.1007/BF00535479

[46] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Found. Trends Mach. Learn.*, vol. 1, no. 1-2, pp. 1–305, January 2008. [Online]. Available: http://dx.doi.org/10.1561/2200000001

[47] L. D. Brown, *Fundamentals of statistical exponential families: with applications in statistical decision theory*. Hayworth, CA, USA: Institute of Mathematical Statistics, 1986.

[48] A. Muhammad and A. Jadbabaie, "Decentralized computation of homology groups in networks by gossip," in *American Control Conference*, 2007.

[49] M. Nabi Abdolyousefi and M. Mesbahi, "Network identification via node knockout," in *IEEE Conference on Decision and Control*, 2010.

[50] G. Chowdhary, M. Egerstedt, and E. N. Johnson, "Network discovery: An estimation based approach," in *American Control Conference*, San Francisco, CA, June 2011.

[51] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*. Urbana, IL: The University of Illinois Press, 1948.

[52] S. Kullback, *Information Theory and Statistics*. New York: John Wiley and Sons, 1959.

[53] S. Kullback and R. A. Leibler, "On information and sufficiency," *Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[54] A. Renyi, "On measures of information and entropy," in *Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability*, 1960, pp. 547–561. [Online]. Available: http://digitalassets.lib.berkeley.edu/math/ucb/text/math_s4_v1_article-27.pdf

[55] H. Chernoff, "A measure of asymptotic efficiency for tests of a hypothesisbased on the sum of observations," *Annals of Mathatical Statistics*, vol. 23, no. 4, pp. 493–507, 1952.

[56] S. M. Ali and S. D. Silvey, "A general class of coefficients of divergence of one distribution from another," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 28, no. 1, pp. 131–142, 1966.

[57] F. Nielsen and R. Nock, "Entropies and cross-entropies of exponential families," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, sept. 2010, pp. 3621 –3624.

[58] P. Bodik, W. Hong, C. Guestrin, S. Madden, M. Paskin, and R. Thibaux, "Intel lab data," Tech. Rep., Feb. 2004.