Efficient distributed sensing using adaptive censoring-based inference

Beipeng Mu\textsuperscript{a,1}, Girish Chowdhary\textsuperscript{b}, Jonathan P. How\textsuperscript{c}

\textsuperscript{a} Laboratory for Information and Decision Systems, MIT, 77 Massachusetts Ave., Cambridge, MA 02139, USA
\textsuperscript{b} Oklahoma State University, USA
\textsuperscript{c} Aeronautics and Astronautics, MIT, 77 Massachusetts Ave., Cambridge, MA 02139, USA

\textbf{A R T I C L E I N F O}

Article history:
Received 21 February 2013
Received in revised form
18 October 2013
Accepted 17 March 2014
Available online xxxx

Keywords:
Consensus
Bayesian parameter estimation
Conjugacy

\textbf{A B S T R A C T}

In many distributed sensing applications with limited resources, it is likely that only a few agents will have valuable information at any given time. Therefore it is important to ensure that the resources are spent 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 for distributions in the exponential family. Furthermore, validation through numerical simulations and real datasets shows 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.

Published by Elsevier Ltd.

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 (Mockenhaupt, 2009; Office of the Secretary of Defense, 2007). Examples of such missions include aerobiological sampling, persistent surveillance, formation control, distributed resource delivery, and target positioning (Michael, Stump, & Mohta, 2011; Mockenhaupt, 2009; Techy, Woolsey, & Schmale, 2008; Wang, Yadav, & Balakrishnan, 2007; Zhang, Kim, & Manocha, 2007). The tasks in these missions often require the agents to distributedly sense, estimate, or reach agreement on global variables, such as the states of the environment or shared variables related to task settings and assignments (Bourguault, Furukawa, & Durrant-Whyte, 2004; Cao et al., 2007; Makarenko, Brooks, Kaupp, Durrant-Whyte, & Dellaert, 2009; Murray, 2007; Olfati-Saber, Fax, & Murray, 2007). However, while distributed sensing agents have the potential to yield benefits such as scalability, high energy efficiency, and resiliency, these agents typically have limited onboard computation and communication resources. Hence, efficient algorithms are needed to ensure that agents can optimally utilize the limited onboard resources while distributedly estimating global variables. This paper will develop an adaptive Value of Information (VoI)-based distributed estimation framework that addresses the problem of distributed estimation in the presence of uncertainties using limited communication resources.

Many distributed estimation algorithms use the notion of consensus to estimate the variables of interest (e.g., Boyd, Ghosh, Prabhakar, & Shah, 2006, Cao et al., 2007, Carli, Chiuso, Schenato, & Zampieri, 2008, Egerstedt & Mesbahi, 2010, Fraser, Bertuccelli, Choi, & How, 2012, Olfati-Saber et al., 2007, Ren, Beard, & Atkins, 2007). However, these algorithms require repeated and continuous communication to reach consensus. Such communication can be resource-intensive and it is often the case that even when all
agents have been assigned equal priority, not all agents have valuable information to contribute at all times. For example, in some cases the updated variables after new measurements are obtained may not be sufficiently different from the preceding variables, 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. For example, Fig. 1 depicts a situation in which several networked agents are estimating a set of variables \( \theta \). If only two agents (dark colored ones) have valuable information, then requiring all the other agents to keep communicating will result in wasted resources. This possibly inefficient use of communication resources (and thus energy) could make the implementation of the standard consensus-based algorithms difficult in real-world resource-constrained applications.

Another set of algorithms for distributed sensing relies on distributed Bayesian inference using graphical models (e.g., Bourgault et al., 2004, Grime & Durrant-Whyte, 1994, Makarenko et al., 2009, Trivedi & Balakrishnan, 2009). In graphical model-based algorithms, agents build local probability models on the variables 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 acyclic networks, 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., Ahmed, Schoenberg, & Campbell, 2012, Bailey, Julier, & Agamennoni, 2012, Chang, Chong, & Mori, 2010, Farrell & Ganesh, 2009, Julien, 2006), or implement additional algorithms to restructure the network into an acyclic network (Paskin, 2004), which introduces extra computational complexity not easily afforded by resource-constrained networks.

1.1. Related work on distributed sensing using censoring

Many authors have explored the notion of censoring agents or measurements based on some information metric to reduce the communication cost (Cetin et al., 2006, Guestrin, Bodik, Thibaux, Paskin, & Madden, 2004, Kreucher, Hero, & Kastella, 2007, Msechu & Giannakis, 2011, Tay, Tsitsiklis, & Win, 2007, Uney, 2009). Censoring has been mainly explored for centralized estimation frameworks (Msechu & Giannakis, 2011, Tay et al., 2007). Cetin et al. have explored censoring in decentralized graphical model-based inference frameworks in the context of a data association problem (Cetin et al., 2006). In that work, messages are communicated only when the content exceeds a preset information threshold. The authors numerically show a significant reduction in the communication cost by trading off some estimation accuracy but the paper does not provide theoretical insights on how to choose the 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 Cetin et al. (2006), 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 (Oliva, Panzieri, Priolo, & Ulini, 2012). Saligram et al. introduced a random censoring algorithm aimed at reducing the communication cost in consensus-based algorithms. In their algorithm, each agent randomly selects a neighbor and passes to it a “transmission permit (token)” (Saligram & Alanyali, 2011). 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 Vol of the measurements as determined by an appropriate information metric. The idea is similar to Cetin et al. (2006) in that agents identify themselves as informative and communicate their information only when the Vol exceeds a threshold. We go beyond Cetin et al. (2006) by theoretically showing that the choice of the Vol 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 Vol-based Distributed Sensing (A-VoIDS) algorithm is introduced that adjusts the Vol 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 (Fraser et al., 2012). Furthermore, theoretical results are established to guarantee almost sure convergence of the communication cost and the estimation error 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 connected.

This work contributes to the goal of developing the next-generation intelligent distributed sensing algorithms with energy constraints. It provides a more efficient framework for performing distributed estimation than existing consensus or graphical model-based approaches (e.g., Berg & Durrant-Whyte, 1992, Egerstedt & Mesbahi, 2010, Gupta, 2006, Hall & Llinas, 1997, Makarenko et al., 2009, Olfati-Saber, Fracco, Frazzoli, & Shamma, 2005, Varshney, 1997). It extends the notion of censoring marginally useful information in a centralized estimation framework (e.g., Msechu &
Giannakis, Tay et al., 2007, Uney, 2009) to Vol-based self-censoring in a distributed inference framework. A preliminary version of this work appeared in a conference (Mu, Chowdhary, & How, 2013). 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 Vol 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

Bayesian framework is used to estimate the variables of interest, because it can model the uncertainty of variables with probability distributions and sequentially update the distributions with measurements.

Let \( \theta \in \mathbb{R}^d \) denote the variables of interest, \( p(\theta) \) denote the prior distribution, and \( z = \{z_1, z_2, \ldots, z_k\}, z_i \in \mathbb{R}^m, i = 1, \ldots, k \), denote a set of measurements with the likelihood \( p(z|\theta) \). Bayes’ theorem states that the posterior distribution \( p(\theta|z) \) is (e.g., Gelman, Carlin, Stern, & Rubin, 1995):

\[
p(\theta|z) = \frac{p(z|\theta)p(\theta)}{\int p(z|\theta)p(\theta)d\theta}.
\]

In general, it is hard or nearly impossible to compute the posterior because the integral \( \int p(z|\theta)p(\theta)d\theta \) 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. For non-exponential family distributions, the typical approach is to use particles to approximate distributions and use sampling-based methods to approximate the posterior. In this case, theoretical analysis would have to explicitly account for the approximations involved.

Let \( p(x|\theta) \) denote the probability distribution of random variables \( x \in \mathbb{R}^m \) under some appropriate measure \( h(dx) \), given parameters \( \theta \in \mathbb{R}^d \). The exponential family is a set of probability distributions that follow the form (Wainwright & Jordan, 2008):

\[
p(x|\theta) = \exp \left\{ \theta^T T(x) - A(\theta) \right\} ,
\]

where \( T(x) : \mathbb{R}^m \mapsto \mathbb{R}^d \) is the Sufficient Statistic or Potential Function and \( A(\theta) = \ln \int \exp \left\{ \theta^T T(x) \right\} h(dx) \) is the Log Partition or Cumulant Function. It is proven in Wainwright and Jordan (2008) that \( A(\theta) \) is positive, convex and in class \( C^\infty \) within its domain that is well-defined.

The exponential family distributions always have conjugate priors that give closed-form posterior solutions (Gelman et al., 1995). The conjugate priors are also within the exponential family, with hyperparameters of dimension \( d + 1 \) (Wainwright & Jordan, 2008). Let \( \omega \in \mathbb{R}^d, \nu \in \mathbb{R} \) denote the hyperparameters and \( A(\omega, \nu) \) denote the conjugate prior’s Log Partition; then the conjugate prior \( p(\omega|\nu) \) has the following form under appropriate measure \( f(d\omega) \):

\[
p(\omega|\nu) = \exp \left\{ \theta^T \omega - A(\theta|\nu) - A(\omega, \nu) \right\} .
\]

For above exponential family likelihood and conjugate prior, the posterior \( p(\theta|z, \omega, \nu) \) after \( n \) measurements \( z = \{z_1, z_2, \ldots, z_n\} \) are observed always has a closed-form solution (Brown, 1986):

\[
p(\theta|z, \omega, \nu) = \exp \left\{ \theta^T \left[ \omega + \sum T(z_i) \right] - A(\theta|\nu + n) \right\}.
\]

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

\[
p(\theta|\tilde{\omega}) = \exp \left\{ \tilde{\theta}^T \tilde{\omega} - A(\tilde{\omega}) \right\} \\
p(\theta|z, \tilde{\omega}) = \exp \left\{ \tilde{\theta}^T (\tilde{\omega} + \tilde{T}(z)) - A(\tilde{\omega} + \tilde{T}(z)) \right\}.
\]

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

\[
\tilde{\omega} \leftrightarrow \tilde{\omega} + \tilde{T}(z).
\]

2.2. Distributed inference

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

Let graph \( G(v, E) \) represent a network of collaborating agents. Set \( v = \{1, \ldots, N\} \) denotes vertices or agents of the network. Set \( E \) denotes edges, \( E \subset v \times v \). Vertices pair \((i, j) \in E \) if and only if the agents \( i \) can communicate with \( j \) (Egerstedt & Mesbahi, 2010). When \( (i, j) \in E \), agent \( j \) is called a neighbor of agent \( i \). The set of all \( i \)'s neighbors is defined as agent \( i \)'s neighborhood, denoted by \( \mathcal{N}_i \).

Assumption 1. Graph \( G \) is connected. That is, for every vertex pair \((i, j) \), there exists a path connecting \( i \) and \( j \), 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 and processing it, so that time to relay a message across the network is shorter than the interval of a single time step.

It is also assumed that all agents are friendly and functioning properly. This is an assumption generally used by consensus and graphical model-based algorithms. To handle the case of malicious or faulty agents, an additional outlier detection procedure can be added before any message is sent into the network.

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 \( z_i(t) = [z_i^1(t), z_i^2(t), \ldots, z_i^{m_i(t)}(t)] \) 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 easily be 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 each 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 \( N^2 \).
The sum of the sufficient statistic can be relaxed by using topology identification algorithms (e.g., known. This assumption can be restrictive in some scenarios, but this algorithm further assumes that the network topology is on parameter distributions in the exponential family and performs Saber et al., 2007).

As all agents have a chance to send out their messages, the estimation error is continuously decreasing. By choosing smaller $\varepsilon$, the communication cost would be reduced. However, the convergence rate could also be reduced as agents communicate less frequently (Saligrama & Alanyali, 2011).

### 3. Value of information metric

The algorithms discussed so far communicate measurements across agents without differentiating the Vol of the measurements to the estimation task at hand. Even when the noise model is homogeneous across the network, the randomness of noise makes it very unlikely that all agents will have the same Vol at all time steps. 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 (Vol) and their implementation in estimation problems. The Vol-based Distributed Sensing (VoIDS) algorithm will be developed in the next section.

#### 3.1. Value of information metric

The idea of quantifying information dates back to Shannon's information theory (Shannon & Weaver, 1948). 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 (Kullback, 1959; Kullback & Leibler, 1951). Renyi generalized KL divergence by introducing an indexed family of similar divergence measures (Renyi, 1960). Chernoff independently introduced another family of information metric, known as the Chernoff distance, which is different from Renyi divergence only by a multiplicative constant (Chernoff, 1952). Further generalization beyond Renyi includes $f$-divergences (or Ali–Silvey divergences, Ali & Silvey, 1966). These as well as some other metrics are listed in Table 1.

The metrics in Table 1 do not have closed-form solutions for general probability distributions. A Vol metric with a closed form as the number of buffered measurements of agent $i$ and $\bar{S}_t[i] = [S_t^i[t], n_t^i[t]]^T$. Agent $i$ sends out a message containing $\bar{S}_t[i]$ only when a locally generated random number between $[0, 1]$ is under a predefined threshold $\varepsilon$. The algorithm is described in Algorithm 3.

Cost: noting that at each step the probability for an agent to send a message is $\varepsilon$, on average there will be $\varepsilon N$ agents broadcasting messages. Each message will be relayed by all the other agents; therefore on average the step communication cost would be $\varepsilon N^2$. As all agents have a chance to send out their messages, the estimation error is continuously decreasing. By choosing smaller $\varepsilon$, the communication cost would be reduced. However, the convergence rate could also be reduced as agents communicate less frequently (Saligrama & Alanyali, 2011).

#### 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., Boyd et al., 2006, Egerstedt & Mesbahi, 2010, Gupta, 2006, Olfati-Saber et al., 2007). One example of the consensus-based algorithms is Fraser et al.’s Hyperparameter Consensus (HPC) (Fraser et al., 2012). 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., Chowdhary, Egerstedt, & Johnson, 2011, Muhammad & Jadbabaie, 2007, Nabi-Abdolyousefi & Mesbahi, 2012).

Notations $t$ and $z_t[i]$ are defined the same as in Section 2.2.1. Let $\bar{\omega}_t[i]$ denote the augmented local hyperparameters of agent $i$ at $t$. Further let $\beta = [\beta_1^N]^T$ denote the eigenvector of eigenvalue 1 of the corresponding adjacency matrix of the network graph; the algorithm is provided in Algorithm 2. Ref. Fraser et al. (2012) 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 its local update of its 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 Saligrama and Alanyali (2011) in which each agent randomly selects a neighbor to pass a communication token to.

After recording a measurement $z_t[i]$, instead of broadcasting it immediately, agent $i$ stores it in a local buffer. Define $S_t[i]$ as the sum of the **Sufficient Statistic** of buffered measurements, $n_t[i]$ as the number of buffered measurements of agent $i$ and $\bar{S}_t[i] = [S_t^i[t], n_t^i[t]]^T$. Agent $i$ sends out a message containing $\bar{S}_t[i]$ only when a locally generated random number between $[0, 1]$ is under a predefined threshold $\varepsilon$. The algorithm is described in Algorithm 3.

Cost: noting that at each step the probability for an agent to send a message is $\varepsilon$, on average there will be $\varepsilon N$ agents broadcasting messages. Each message will be relayed by all the other agents; therefore on average the step communication cost would be $\varepsilon N^2$. As all agents have a chance to send out their messages, the estimation error is continuously decreasing. By choosing smaller $\varepsilon$, the communication cost would be reduced. However, the convergence rate could also be reduced as agents communicate less frequently (Saligrama & Alanyali, 2011).
solution is desirable, as it would allow Vol 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 Vol can help reduce the computational cost. Note that KL divergence is Renyi divergence when $\alpha \rightarrow 1$. Here we pick KL divergence to be the metric on Vol in our problem. However, other Vol metrics can also be used with the algorithms developed later.

3.2. KL divergence and Bayesian inference

Recall that $p(z|\theta)$, $p(\theta|\tilde{\omega})$ and $p(\theta|z, \tilde{\omega})$ 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 (Nielsen & Nock, 2010):

$$D_{KL}(p(\theta|\tilde{\omega}) \parallel p(\theta|z, \tilde{\omega})) = \mathcal{L} \left( \tilde{\omega} + \tilde{T}(z) \right) - \mathcal{L} \left( \tilde{\omega} \right) - \tilde{T}(z)^T \nabla \mathcal{L} \left( \tilde{\omega} \right),$$

where $\mathcal{L}$ represents the gradient.

3.3. Single agent case

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

Assume the agent takes $m(t)$ measurements $z[t]$ at time $t$, so $|z[t]| = m(t)$. The hyperparameter of the conjugate prior at $t$ is:

$$\tilde{\omega}[t] = \tilde{\omega}[0] + \tilde{T}(1 : t − 1).$$

The following theorem formalizes the intuitive notion that as an agent takes more measurements, its estimate of the variables improves, while the Vol in the new measurements $z[t]$ decreases to zero.

Theorem 1. 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 $\theta_0$. At time $t$, define $V[t]$ as Vol of new measurement $z[t]$.

$$V[t] = D_{KL}(p(\theta|\tilde{\omega}[t]) \parallel p(\theta|z[t], \tilde{\omega}[t])),$$

then $\lim_{t \to \infty} V[t] \to 0$ a.s.

To be concise, the proofs of this and following theorems are left in the Appendix. The proof of Theorem 1 can be found at Appendix A.2.

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 Vol first. The posterior is updated only when the Vol of the buffered measurements exceeds a threshold $V^*$. This process is described in Algorithm 4.

<table>
<thead>
<tr>
<th>Table 1 Information metrics.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metric</td>
</tr>
<tr>
<td>Kullback–Leibler</td>
</tr>
<tr>
<td>Renyi</td>
</tr>
<tr>
<td>Chernoff</td>
</tr>
<tr>
<td>f-divergence</td>
</tr>
<tr>
<td>Variational</td>
</tr>
<tr>
<td>Matusita</td>
</tr>
</tbody>
</table>

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

The following result guarantees that if an agent uses Algorithm 4, then the frequency of the posterior updates will decrease with time, because the Vol of new measurements will decrease with time due to Theorem 1.

Theorem 2. Consider the case where a single agent takes one measurement $z[t]$ at time step $t$ and does inference according to Algorithm 4. Assume that all the measurements are i.i.d. drawn from a static distribution with parameters $\theta_0$, $z[t] \sim p(z|\theta_0)$. At time $t$, let $t_k$ be the next time step the agent updates the hyperparameters, and $n[t_k]$ be the number of measurements buffered at $t_k$; then $\lim_{t \to \infty} n[t_k] \to \infty$ a.s.

The proof of Theorem 2 can be found at Appendix A.3.

4. Vol-based Distributed Sensing (VoIDS)

In this section we develop the Vol-based Distributed Sensing (VolDS) algorithm for a network of multiple agents.

In VolDS, 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 $n_i(t)$ as the number and sum of Sufficient Statistic of buffered measurements for agent $i$ at time $t$, and let $S_i[t] = \sum_{t = 0}^{t} n_i[t]$. Denote $V_i[t]$ as the Vol of agent $i$’s buffered measurements at $t$.

The algorithm proceeds as follows. If $V_i[t] > \tilde{V}$, where $\tilde{V}$ is a predefined threshold $V^*$, agent $i$ labels itself as informative; otherwise it labels itself as uninformative. All informative agents broadcast a message containing $S_i[t]$ to their neighbors, and then clear their local buffers and reset $S_i[t]$ 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 Assumptions 1–3, 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 3. Consider a network of $N$ agents performing distributed inference with Algorithm 5. Assume that the measurements are i.i.d. drawn from a distribution with static parameters $\theta_0$. At time $t$, denote $t_k$ as the next time step agent $i$ sends out a message to update the global
Algorithm 5 Vol based Distributed Sensing (VoIDS)

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

hyperparameters; then for any agent $i$, the number of measurements needed to exceed a predefined Vol threshold $V^*$ will go to infinity, that is $\lim_{t \to \infty} n_i[t^*_i] \to \infty$.

The proof of Theorem 3 can be found at Appendix A.4.

In many distributed estimation scenarios, it is difficult to know the true value of the parameters and the best estimate that can be achieved is the centralized posterior estimate. Here distributed posterior estimates are compared with centralized posterior estimates which are formed using all of the measurements taken so far. 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:

$$e[t] = D_{KL}(p(\theta|\omega[t]) \parallel p(\theta|\omega[t] + \sum_{j \in I} \bar{S}_j[t]))$$

$$= \left( p(\theta|\omega[t] + \sum_{j \in I} \bar{S}_j[t]) \right) - p(\theta|\omega[t])$$

$$= \sum_{i \in I} (p(\theta|\omega[t] + \sum_{j \in I} \bar{S}_j[t]) - p(\theta|\omega[t]))$$

(10)

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

Theorem 4. Consider a network of $N$ agents that performs inference with Algorithm 5. At time instance $t$, let the error $e[t]$ be defined by (10); then $e[t] \leq N^2V^*$ a.s.

The proof of Theorem 4 can be found at Appendix A.5.

5. Adaptive Vol-based Distributed Sensing (A-VoIDS)

With a static Vol 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 variables 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 variables, 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 $N^2V^*$ instead of decreasing. 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 Vol 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 Vol-based Distributed Sensing (A-VoIDS) algorithm discussed in this section provides a way to adaptively adjust the Vol 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 Vol-based Distributed Sensing algorithm

Let indicator function $I_{V_i[t] > V^*[t]}$ 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 $l$:

$$C[t] = \frac{1}{l} \sum_{t-l+1}^{t} \sum_{i=1}^{N} I_{V_i[j] > V^*[j]}.$$

(11)

Variable $C[t]$ reflects the average step communication cost in a fixed length window. It should be noted that because the Vol 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 Vol 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-optimized; thus the algorithm reduces $V^*$ to encourage communication by setting $V^*[t + 1] = \gamma_0 V^*[t]$, $0 < \gamma_0 < 1$ (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 $V^*[t + 1] = \gamma_1 V^*[t]$, $\gamma_1 > 1$ (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 5. 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 $\theta_0$. Then the estimation error $e[t]$ as defined in (10) asymptotically reduces to zero a.s., that is $\lim_{t \to \infty} e[t] \to 0$ a.s.

The proof of Theorem 5 can be found at Appendix A.6.

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...
Algorithm 6 Adaptive VoI Based Distributed Sensing (A-VoIDS)

1: initiate hyperparameters $\bar{o}[0]$
2: for $t$ do
3: for each agent $i$ do
4:   $\dot{o}_i[t] = \dot{o}_i[t-1], C_i[t] = 0$
5:   take measurement and update local buffer
6:   $\bar{S}_i[t] = \bar{S}_i[t-1] + T(z_i[t])$
7:   calculate VoI of current buffer
8:   $V_i[t] = D_{sz}(p(\varphi|\dot{o}_i[t])|p(\varphi|\dot{o}_i[t] + \bar{S}_i[t]))$
9:   if $V_i[t] > V^*[t]$ then
10:      broadcasts $\bar{S}_i[t]$
11:      reset local buffer: $\bar{S}_i[t] = 0$
12:   end if
13: end for
14: for each broadcast message $\bar{S}_i[t]$ do
15:   update the global posterior
16:   $\bar{o}_i[t] = \dot{o}_i[t] + \bar{S}_i[t]$
17:   step communication cost increased by 1
18:   $C_i[t] = C_i[t] + 1$
19: end for
20: adaptively change $V^*[t]$
21: if $C_i[t] < c$ 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

Table 2
Performance summary.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Cost in a step</th>
<th>Error to centralized posterior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full relay</td>
<td>$N^2$</td>
<td>0</td>
</tr>
<tr>
<td>HPC</td>
<td>$N$</td>
<td>Converges to 0</td>
</tr>
<tr>
<td>Random broadcast</td>
<td>$\varepsilon N^2$, $\varepsilon$ tunable</td>
<td>Random; slowly converges to 0</td>
</tr>
<tr>
<td>VoIDS</td>
<td>Converge to 0</td>
<td>Bounded</td>
</tr>
<tr>
<td>A-VoIDS</td>
<td>$cN$, $c$ tunable</td>
<td>Quickly converges to 0</td>
</tr>
</tbody>
</table>

6.1. Evaluation using simulated dataset

The presented simulation considers a group of collaborative agents estimating the Poisson arrival rate $\lambda$ of an entity. The prior distribution of $\lambda$ is chosen to be a Gamma distribution $f(\alpha, \beta)$, which is conjugate to Poisson. The Poisson and Gamma distributions are given in (12) and (13) respectively:

$$p(z|\lambda) = \frac{\lambda^z e^{-\lambda}}{z!}$$  \hspace{1cm}  (12)

$$p(\lambda|\alpha, \beta) = \frac{\beta^\alpha \lambda^{\alpha-1} e^{-\beta \lambda}}{\Gamma(\alpha)}.$$  \hspace{1cm}  (13)

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

$$\alpha = \alpha + z$$

$$\beta = \beta + 1.$$  \hspace{1cm}  (14)

The presented simulation considers a group of collaborative agents estimating the Poisson arrival rate $\lambda$ of an entity. The prior distribution of $\lambda$ is chosen to be a Gamma distribution $f(\alpha, \beta)$, which is conjugate to Poisson. The Poisson and Gamma distributions are given in (12) and (13) respectively:

$$p(z|\lambda) = \frac{\lambda^z e^{-\lambda}}{z!}$$  \hspace{1cm}  (12)

$$p(\lambda|\alpha, \beta) = \frac{\beta^\alpha \lambda^{\alpha-1} e^{-\beta \lambda}}{\Gamma(\alpha)}.$$  \hspace{1cm}  (13)

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

$$\alpha = \alpha + z$$

$$\beta = \beta + 1.$$  \hspace{1cm}  (14)

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

Figs. 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 Fig. 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. The comparison shows that HPC and Random Broadcast continuously reduce their estimation error, implying asymptotic convergence.

As proven by Theorem 3, 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 (11) 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 and 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 the communication cost and the estimation error.

A-VoIDS (Algorithm 6) strikes a better balance between the communication cost and the estimation error. The cumulative cost of A-VoIDS also increases linearly, but 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 Fig. 3 and it can be seen that \( V^* \) drops to zero as shown by Theorem 5. This indicates that unlike VoIDS, A-VoIDS tends to asymptotically reduce the error, since the estimation error is bounded above by \( V^* \) (see Fig. 4).

The performance of the algorithms discussed is compared in Fig. 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 the lower error but higher communication cost. The Random Broadcast algorithm does a trade-off between the cost and the 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 the 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.

6.2. Evaluation using the real dataset

The VoIDS and A-VoIDS algorithms are further evaluated using a real dataset that has been used to analyze distributed sensing algorithms (Bodik et al., 2004; Guestrin, Krause, & Singh, 2005). In this dataset, 54 sensors distributed in the Intel Berkeley Research lab collect timestamped information such as humidity, temperature, light, and voltage values every half minute (Bodik et al., 2004; Guestrin et al., 2005). The sensor layout is shown in Fig. 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 VoIDS and A-VoIDS algorithms are further evaluated using a real dataset that has been used to analyze distributed sensing algorithms (Bodik et al., 2004; Guestrin, Krause, & Singh, 2005). In this dataset, 54 sensors distributed in the Intel Berkeley Research lab collect timestamped information such as humidity, temperature, light, and voltage values every half minute (Bodik et al., 2004; Guestrin et al., 2005). The sensor layout is shown in Fig. 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 average room temperature denoted by \( \theta \). We model the noisy sensor measurements by each sensor using a Gaussian noise model with constant variance, \( z \sim N(\theta, 1) \). 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 \( \bar{\theta} \sim N(\nu, \omega) \), where \( \omega, \nu \) are the hyperparameters of the Gaussian conjugate prior. When a new measurement \( z \) is taken by a sensor, it updates its hyperparameters as in Nielsen and Nock (2010):

\[
\omega = \omega + 2, \\
\nu = \nu + z.
\]
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: 1 Hz, $\frac{1}{2}$ Hz, $\frac{1}{4}$ Hz, and $\frac{1}{8}$ Hz. Note that the communication rate does not affect the number of messages sent out when sensors are running VoIDS or AVoIDS.

The performance of different algorithms is compared in Fig. 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 the 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 the 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-right corner of the error-cost figure, indicating that A-VoIDS can give an estimate with significantly less communication cost than other algorithms that have the similar error.

Fig. 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 (1 Hz) and the centralized estimate drops within 0.1 °C within first 500 s and keeps decreasing. After 1500 s, the error is within 0.05 °C. VoIDS (with $V^* = 0.1$) estimation error also drops within 0.1 °C in 500 s, but does not further decrease; even after 2000 s, 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 (Vol)-Based Distributed Sensing (VoIDS) algorithms are discussed in the framework of Bayesian inference. VoIDS algorithms are designed to overcome known shortcomings such as the excessive communication cost and slow convergence speed of traditional consensus-based algo-

### Appendix

#### A.1. Lemma 1

Before we prove the theorems, we first state a useful lemma.

**Lemma 1.** Denote $\Lambda(\hat{\omega})$ as the log partition of $p(\theta|\hat{\omega})$. If $n$ measurements $\mathbf{z} = \{z_1, \ldots, z_n\}$ are used to update $p(\theta|\hat{\omega})$ by Eq. (5), and $\lim_{n \to \infty} \sum_{i=1}^n \tilde{f}(z_i) = \tilde{T} < \infty$, then $\lim_{n \to \infty} V^2 \Lambda(\hat{\omega} + \tilde{T}(\mathbf{z})) \to 0$ a.s. where $V^2$ represents the Hessian matrix.

**Proof.** It is proven in Wainwright and Jordan (2008) that $V^2 \Lambda(\hat{\omega}) = \text{cov}(\hat{\theta}|\hat{\omega})$; therefore,

$$
\lim_{n \to \infty} V^2 \Lambda(\hat{\omega} + \tilde{T}(\mathbf{z})) = \lim_{n \to \infty} \text{cov}(\hat{\theta}|\hat{\omega} + \tilde{T}(\mathbf{z})) = \lim_{n \to \infty} \int \left( (\hat{\theta} - \tilde{\theta})^T (\hat{\theta} - \tilde{\theta}) \right) p(\theta) \bigg|_{\hat{\omega} + \tilde{T}(\mathbf{z})} \ d\theta. \tag{16}
$$
where $|\cdot|_{\omega + \tilde{T}(z)}$ denote that the expectation and probability is conditioned on $\omega + \tilde{T}(z)$.

We first prove that $p(\theta|\omega + \tilde{T}(z))$ converges to a delta function. From (5),

$$
p(\theta|\omega + \tilde{T}(z)) \propto \exp \left\{ \bar{y}^T (\omega + \tilde{T}(z)) \right\}.
$$

Denote

$$
\theta_n = \arg \max_{\theta} \frac{1}{n} \bar{y}^T (\omega + \tilde{T}(z)) = \arg \max_{\theta} \frac{1}{n} \bar{y}^T (z) + \omega - A(\theta).
$$

For any $\theta \neq \theta_0$, 

$$
p(\theta|\omega + \tilde{T}(z)) \propto \exp \left\{ -(n + \nu) (\theta - \theta_0)^T (\omega + \tilde{T}(z)) \right\}.
$$

Notice $\lim_{n \to \infty} \frac{\hat{\omega} + \tilde{T}(z)}{\nu + n} = \lim_{n \to \infty} \frac{\tilde{T}(z)}{\nu + n} = \tilde{T}(z)$ and 

$$
(\theta - \theta_0) \frac{\hat{\omega} + \tilde{T}(z)}{\nu + n} \leq 0, \text{ so } p(\theta|\omega + \tilde{T}(z)) \text{ decays exponentially fast with number of measurements } n, \text{ and we have}
$$

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

(17)

where $\theta_0 = \arg \max_{\theta} \theta^T \tilde{T}(z)$.

Then for any small $\varepsilon$, we can find large enough $n$, such that for $|\theta - \theta_0| > \varepsilon$, $p(\theta|\omega + \tilde{T}(z)) \leq \varepsilon p(\theta_0|\omega)$, and $|E[\theta|\omega + \tilde{T}(z)] - \theta_0| < \varepsilon$ a.s. Then

$$
\begin{align*}
\frac{1}{|\theta - \theta_0|} & \left( \frac{1}{\nu + n} - \frac{1}{\nu} \right) > 0 \\
\frac{1}{|\theta - \theta_0|} & \left( \frac{1}{\nu + n} - \frac{1}{\nu} \right) \geq 0,
\end{align*}
$$

(16)

Because $\varepsilon$ is arbitrary, (16) becomes:

$$
\lim_{n \to \infty} \frac{1}{\nu + n} + \tilde{T}(z) = \lim_{n \to \infty} \frac{1}{|\theta - \theta_0|} \left( \frac{1}{\nu + n} - \frac{1}{\nu} \right) \geq 0.
$$

(18)

A.2. Theorem 1

**Proof.** Expand $\Lambda (\tilde{T}(z))$ in (7) in a Taylor series around $\Lambda (\tilde{T}(z))$, in the form of both integral remainder and Lagrange remainder (Apostol, 2014):

$$
D(\Lambda (\tilde{T}(z))) = A(\tilde{T}(z)) - \tilde{T}(z)^T \Lambda (\tilde{T}(z)) \tilde{T}(z) \Lambda (\tilde{T}(z)) = \frac{1}{2} \tilde{T}(z)^T \tilde{T}(z) - \tilde{T}(z)^T \Lambda (\tilde{T}(z)) - \tilde{T}(z)^T \Lambda (\tilde{T}(z)) \tilde{T}(z)
$$

Lagrange remainder, 

$$
(19)
$$

where $\delta \tilde{T}(z) \in [0, \tilde{T}(z)]$.

Then the Vol of this agent at $t$ is:

$$
V[t] = D_{KL}(\rho(\tilde{T}(z)) || \rho(\tilde{T}(t)))
$$

$$
= \frac{1}{2} \tilde{T}(z)^T \tilde{T}(z) - \tilde{T}(z)^T \Lambda (\tilde{T}(z)) - \tilde{T}(z)^T \Lambda (\tilde{T}(z)) \tilde{T}(z)
$$

(20)

Given finite measurements $z(t)$, Sufficient Statistic $T(z(t))$ is finite, so vector $\tilde{T}(z(t)) = [T(z(t))^T, 1]^T$ is also finite. Furthermore, from Lemma 1, $lim_{t \to \infty} \infty \Lambda (\tilde{T}(z(t)) + \delta \tilde{T}(z(t))) \to 0$ a.s. Hence

$$
\lim_{t \to \infty} \frac{1}{2} \tilde{T}(z(t))^T \tilde{T}(z(t)) \to 0 \quad \text{a.s.}
$$

that is $lim_{t \to \infty} V[t] \to 0$ a.s. ■

A.3. Theorem 2

**Proof.** Time index $t$ is in between two update events: $t \in (t_k, t_{k+1})$, where $t_k$ represents the last time the agent updated the posterior, and $t_k$ 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, $lim_{n \to \infty} (n(t_k) + t_{k+1}) = lim_{n \to \infty} t_k = \infty$. We have either $lim_{n \to \infty} n(t_k) \to \infty$ and/or $lim_{n \to \infty} t_k \to \infty$.

(i) In the first case, $lim_{n \to \infty} n(t_k) \to \infty$, the theorem holds.

(ii) Consider for the sake of contradiction that $n(t_k)$ is bounded, that is $lim_{n \to \infty} n(t_k) \leq C < \infty$. In this case, it follows that $lim_{n \to \infty} t_k \to \infty$. 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 1, $lim_{k \to \infty} V[t_k] \to 0$ a.s., which means $P(V[t_k] > V) \to 0$. 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 $lim_{n \to \infty} n(t_k) \to \infty$ a.s. ■

A.4. Theorem 3

**Proof.** First assume the case where agent $i$ does not receive any messages from other agents after $t^i_{k-1}$. Define the time it sends next message as $t^i_k = t^i_{k-1} + n(t_k)$. From Theorem 2, $lim_{n \to \infty} n(t_k) \to \infty$, so Theorem 3 holds.

On the other hand, if agent $i$ receives one or more messages from other agents between $t^i_{k-1}$ and $t^i_k$, the global hyperparameters are updated between $t^i_{k-1}$ and $t^i_k$; thus agent $i$ would have used more measurements to update the hyperparameters by time $t^i_{k-1}$. This would only make $\tilde{T}(\theta(t))$ converge to 0 faster than in the first case due to Lemma 1. Hence in order to reach the same Vol threshold $V^*$, agent $i$ needs to take more measurements. Denote $t^i_k$ to be the time agent $i$ sends out the next message; in this case, $n(t^i_k) \geq n(t^i_k)$. Since $lim_{n \to \infty} n(t^i_k) \to \infty$, $lim_{n \to \infty} n(t^i_k) \to \infty$.

Hence, in both cases, $lim_{n \to \infty} n(t^i_k) \to \infty$. ■

A.5. Theorem 4

**Proof.** From (19) and (9):

$$
V_i[t] = \int_0^1 \left( \frac{1}{\tilde{s}[t]} - x \right)^T \tilde{T}(z(t)) - \tilde{T}(z(t)) \tilde{T}(z(t))
$$

$$
(21)
$$

Let $x = y \tilde{s}_t[r]$, then
\begin{equation}
= \int_0^1 (1 - y) \tilde{s}_t[r]^2 \nabla^2 A(\tilde{\omega} + y \tilde{s}_t[r]) \tilde{s}_t[r] dy.
\end{equation}
(21)
For brevity in the following, $[r]$ will be left out of $\tilde{s}_t[r]$. Similarly from (19) and (10):
\begin{equation}
e[r] = \int_0^1 (1 - y) \sum_{k=1}^{\infty} \tilde{s}_t^2 \nabla^2 A \left( \tilde{\omega} + \sum_{j=1}^n \tilde{s}_j + y \sum_{i=1}^m \tilde{s}_i \right) \sum_{k=1}^{\infty} \tilde{s}_k dy.
\end{equation}
(22)
Define $m$ as the index that maximizes $\tilde{s}_m^2 \nabla^2 A(\cdot) \tilde{s}_m$, that is
\begin{equation}
m = \arg \max \tilde{s}_m^2 \nabla^2 A \left( \tilde{\omega} + \sum_{j=1}^n \tilde{s}_j + y \sum_{i=1}^m \tilde{s}_i \right) \tilde{s}_m.
\end{equation}
which leads to
\begin{equation}
e[r] \leq \int_0^1 (1 - y) N^2 \tilde{s}_m^2 \nabla^2 A \left( \tilde{\omega} + \sum_{j=1}^n \tilde{s}_j + y \sum_{i=1}^m \tilde{s}_i \right) \tilde{s}_m^2 dy
\leq N^2 \int_0^1 (1 - y) \tilde{s}_m^2 \nabla^2 A \left( \tilde{\omega} + y \tilde{s}_m \right) \tilde{s}_m^2 dy.
\end{equation}
(24)
In the following, we will show:
\begin{equation}
\tilde{s}_m^T \nabla^2 A \left( \tilde{\omega} + \sum_{j=1}^n \tilde{s}_j + y \sum_{i=1}^m \tilde{s}_i \right) \tilde{s}_m
\leq \tilde{s}_m^T \nabla^2 A \left( \tilde{\omega} + y \tilde{s}_m \right) \tilde{s}_m.
\end{equation}
(25)
$V^4(\cdot)$ is 4th central moment of $\tilde{\theta}$, so $\tilde{s}_m[t]^T V^4(\cdot) \tilde{s}_m[t] \geq 0$. Notice that $\tilde{s}_m[t]^T V^4(\cdot) \tilde{s}_m[t]$ is the second derivative of $\tilde{s}_m[t]^T V^4 A(\cdot) \tilde{s}_m[t]$ with respect to $\tilde{s}_m[t]$ for $i \neq m$, we have that $\tilde{s}_m[t]^T V^4 A(\cdot) \tilde{s}_m[t]$ is convex over $\tilde{s}_m[t]$. By definition, a function $f(x)$ is convex means:
\begin{equation}
\lambda f(x_1) + (1 - \lambda) f(x_2) \geq f(\lambda x_1 + (1 - \lambda) x_2)
\end{equation}
for any $0 \leq \lambda \leq 1$.
Let $f(x) = \tilde{s}_m[t]^T V^4 A(\omega + x) \tilde{s}_m[t]$, $\lambda = \frac{a}{a}$ where $a \geq 1$, $x_2 = y \tilde{s}_m$ and $x_1 = y \tilde{s}_m + a \sum_{j=1}^n \tilde{s}_j + a y \sum_{i=1}^m \tilde{s}_i$; then we have
\begin{equation}
\frac{a - 1}{a} \tilde{s}_m^T V^2 A \left( \tilde{\omega} + y \tilde{s}_m \right) \tilde{s}_m
+ \sum_{j=1}^n \tilde{s}_j^T V^2 A \left( \tilde{\omega} + y \tilde{s}_m + a \sum_{j=1}^n \tilde{s}_j + a y \sum_{i=1}^m \tilde{s}_i \right) \tilde{s}_m
\geq \frac{a}{a} \tilde{s}_m^T V^2 A \left( \tilde{\omega} + \sum_{j=1}^n \tilde{s}_j + y \sum_{i=1}^m \tilde{s}_i \right) \tilde{s}_m.
\end{equation}
(26)
Since (26) holds for all $a \geq 1$, let $a \to \infty$; the first term on the left side of (26) becomes $\tilde{s}_m^T V^2 A \left( \tilde{\omega} + y \tilde{s}_m \right) \tilde{s}_m$. For the second term on the left side of (26) becomes $\tilde{s}_m^T V^2 A \left( \tilde{\omega} + y \tilde{s}_m \right) \tilde{s}_m$. For the second term on the left side of (26) becomes $\tilde{s}_m^T V^2 A \left( \tilde{\omega} + y \tilde{s}_m \right) \tilde{s}_m$. For the second term on the left side of (26) becomes $\tilde{s}_m^T V^2 A \left( \tilde{\omega} + y \tilde{s}_m \right) \tilde{s}_m$.

A6. Theorem 5

Proof. Denote the probability distribution of $V^*[t]$ at time $t$ as $p^*(v)$. At time $t$, define the probability of being in mode 1 as $P_1[t|v] = \mathbb{P}(C[t] < c|v)$ and being in mode 2 as $P_2[t|v] = \mathbb{P}(C[t] \geq c|v)$. From Theorem 3, for any given Vol 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. $\lim_{t \to \infty} I_{V[t] > v} \to 0$ a.s. Therefore, for a fixed window length $\ell$, the average cost $C[t]$ satisfies:
\begin{equation}
\forall v > 0, \quad \lim_{t \to \infty} C[t] = \lim_{t \to \infty} \frac{1}{\ell} \sum_{j=t+1}^{t+\ell} \sum_{i=1}^{N} I_{V[i] > v} \to 0 \quad a.s.
\end{equation}
(27)
Hence, over time the probability of being in mode 1 will approach 1, while being in mode 2 will approach 0, that is
\begin{equation}
\forall v > 0, \quad \lim_{t \to \infty} P_1[t|v] = 1, \quad \text{and} \quad \lim_{t \to \infty} P_2[t|v] = 0.
\end{equation}
(28)
For any given $\xi > 0$, if $V^*[t + 1] \geq \xi$, there are two possibilities, $V^*[t] \geq \frac{\xi}{1}$ and the algorithm falls into mode 1 at $t$, or $V^*[t] \geq \frac{\xi}{1}$ and the algorithm falls into mode 2 at $t$, where $\gamma_1, \gamma_2$ are as defined in Algorithm 6. Therefore, we have
\begin{equation}
\mathbb{P}(V^*[t + 1] \geq \xi) = \int_{\xi}^{\infty} p^*(v) dv
- \int_{\xi}^{\infty} P_1(t|v)p^*(v) dv + \int_{\xi}^{\infty} P_2(t|v)p^*(v) dv.
\end{equation}
(29)
When $t \to \infty$, $P_1 \to 1$, $P_2 \to 0$; therefore taking limit w.r.t. time we have
\begin{equation}
\lim_{t \to \infty} \mathbb{P}(V^*[t + 1] \geq \xi) = \lim_{t \to \infty} \int_{\xi}^{\infty} p^*(v) dv
= \lim_{t \to \infty} \mathbb{P}(V^*[t] \geq \frac{\xi}{1}).
\end{equation}
(30)
Moving two limits to the same side of the equation, we have:
\begin{equation}
\lim_{t \to \infty} \mathbb{P}(V^*[t + 1] \geq \xi) - \mathbb{P}(V^*[t] \geq \frac{\xi}{1}) = 0.
\end{equation}
(31)
Noting that $[t + \tau, t] = \Gamma$ and (31) can be rewritten by adding and subtracting intermediate terms
\begin{equation}
\lim_{t \to \infty} \mathbb{P}(V^*[t + 1] \geq \xi) - \mathbb{P}(V^*[t] \geq \frac{\xi}{1})
= \lim_{t \to \infty} \left\{ \mathbb{P}(V^*[t + \tau] \geq \xi) - \mathbb{P}(V^*[t + \tau - 1] \geq \frac{\xi}{1}) \right\}
+ \cdots
\end{equation}
\begin{equation}
+ \lim_{t \to \infty} \left\{ \mathbb{P}(V^*[t + 2] \geq \frac{\xi}{1}) - \mathbb{P}(V^*[t + 1] \geq \frac{\xi}{1}) \right\}
+ \lim_{t \to \infty} \left\{ \mathbb{P}(V^*[t + 1] \geq \frac{\xi}{1}) - \mathbb{P}(V^*[t] \geq \frac{\xi}{1}) \right\}
\end{equation}
apply (31) to each of the limits
\begin{equation}
= 0 + 0 + \cdots + 0 = 0.
\end{equation}
(32)
Now letting $\tau \to \infty$ we obtain:
\[
\lim_{\tau \to \infty} \mathbb{P}(V^t[r + \tau] \geq c) = \lim_{\tau \to \infty} \mathbb{P}(V^t[r] \geq c / \gamma^t) = 0.
\]

Therefore,
\[
\lim_{\tau \to \infty} \mathbb{P}(V^t[r] \geq c) = 0.
\]

Because by definition of probability measures $\mathbb{P}(V^t[r] \geq \infty) = 0$, we have:
\[
\lim_{\tau \to \infty} \mathbb{P}(V^t[r] \geq \infty) = 0.
\]

Therefore, $\lim_{\tau \to \infty} V^t[r] \to 0$ a.s. From Theorem 4, $\lim_{\tau \to \infty} e(t) \to 0$ a.s.

References


Calculus


Mackenzie, J. (2009). We’ve seen the future, and it’s unmanned. Esquire.


Beipeng Mu is a Ph.D. candidate at Massachusetts Institute of Technology. She received her Master of Science from the Department of Aeronautics and Astronautics at MIT in 2013, and Bachelor of Engineering from the Department of Automation at Tsinghua University in 2011. Her research interests include distributed estimation and decision-making algorithms as applied to unmanned vehicles.

Girish Chowdhary is an Assistant Professor of Mechanical and Aerospace Engineering at Oklahoma State University. He has postdoctoral experience at Massachusetts Institute of Technology’s Laboratory for Information and Decision Systems, and the School of Aerospace Engineering at Georgia Institute of Technology. He received a Ph.D. from Georgia Institute of Technology in 2010. Prior to joining Georgia Tech., he worked as a research engineer with the German Aerospace Center’s (DLR’s) Institute for Flight Systems Technology in Braunschweig, Germany. He received an M.S. degree in Aerospace Engineering from Georgia Tech in 2008, and a B.E. with honors from RMIT university in Melbourne, Australia, in 2003. He is a member of AIAA, IEEE Control Systems Society, and AUVSI.

Jonathan P. How is the Richard C. Maclaurin Professor of Aeronautics and Astronautics at the Massachusetts Institute of Technology (MIT). He received a B.A.Sc. from the University of Toronto in 1987 and his S.M. and Ph.D. in Aeronautics and Astronautics from MIT in 1990 and 1993, respectively. He then studied for two years at MIT as a postdoctoral associate for the Middeck Active Control Experiment (MACE), which flew on-board the Space Shuttle Endeavour in March 1995. Prior to joining MIT in 2000, he was an Assistant Professor in the Department of Aeronautics and Astronautics at Stanford University. Current research interests include robust coordination and control of autonomous vehicles in dynamic uncertain environments. Professor How was the recipient of the 2002 Institute of Navigation Burka Award, a Boeing Special Invention award in 2008, the 2011 IFAC Automatica award for best applications paper. Recipient of the AIAA Best Paper Award from the 2011 and 2012 Guidance Navigation and Control Conferences, and he is an Associate Fellow of AIAA and a senior member of IEEE.