

Reaching Consensus with Imprecise Probabilities Over a Network

Cameron S. R. Fraser*, Luca F. Bertuccelli[†] and Jonathan P. How[‡]

Aerospace Controls Laboratory
Massachusetts Institute of Technology

Information consensus in sensor networks has received much attention due to its numerous applications in distributed decision making. This paper discusses the problem of a distributed group of agents coming to agreement on a probability vector over a network, such as would be required in a decentralized estimation of state transition probabilities or agreement on a probabilistic search map. Unique from other recent consensus literature, however, the agents in this problem must reach agreement while accounting for the uncertainties in their respective probabilities, which are formulated according to generally non-Gaussian distributions. The first part of this paper considers the problem in which the agents seek agreement to the centralized Bayesian estimate of the probabilities, which is accomplished using consensus on *hyperparameters*. The second part shows that the new hyperparameter consensus methodology can ensure convergence to the centralized estimate even while measurements of a static process are occurring concurrently with the consensus algorithm. A machine repair example is used to illustrate the advantages of hyperparameter consensus over conventional consensus approaches.

I. Introduction

Many large-scale systems, such as Unmanned Aerial Vehicle (UAV) teams or geographically separated medical infrastructures, can be modeled as a group of heterogeneous agents with local information that must communicate to accomplish a common objective. Each of the agents can be considered as a node of a more complex network with localized information that may be unavailable to other members of the network. The agents must nonetheless come to an agreement on the course of action and it is often beneficial to share the information across the network and be able to reach an agreement with respect to the nature of the observed event in order to expedite learning or facilitate coordination of the team^{1,2}.

This paper considers a typical scenario in which the agents have observed (or are observing) a discrete-event stochastic process, and must come to agreement with the rest of the network on their *beliefs* of the model of the process. While probabilities dictating the likelihood of each event account for an intrinsic uncertainty in the representation of the problem, the probabilities are often the outcome of a separate estimation process, and are likely to be uncertain themselves³⁻⁶. Uncertainty in these probabilities can have dramatic performance impact. For example, it has been previously shown by other authors that, in the context of a Markov Decision Process, the poor

*Graduate Student, Dept of Aeronautics and Astronautics fraserc@mit.edu

[†]Postdoctoral Associate, Department of Aeronautics and Astronautics: lucab@mit.edu

[‡]Professor, Dept of Aeronautics and Astronautics, Associate Fellow AIAA jhow@mit.edu

knowledge of the transition probabilities of the Markov Chain can degrade optimal performance of decision systems^{7–10}.

The approach in this paper considers the probabilities as *imprecise* and notes that the confidence in these probabilities should also be used in coming to agreement. It is shown here that ignoring this uncertainty can have profound impact on the agreed belief, while incorporating the uncertainty in a non-representative manner can have equally poor results. There are many instances in which probabilistic beliefs may require agreement. For example, cooperative search with a mixed team of humans and UAVs may require the operators to aggregate the information obtained from multiple UAVs (through noisy camera sensors) and ground operators (beliefs which may be affected by stress or fatigue) over a known network of agents before committing a certain number of UAV assets to assess the newly found target. If this aggregation does not properly reflect the true confidences in each information source, the wrong conclusions could be made and an erroneous quantity of resources could be committed. In the medical realm, multiple doctors observing a patient’s health may need to agree on the diagnosis in a distributed manner, especially when the doctors cannot all meet up at one central time and place, and yet come up with a consistent diagnosis. Yet another instance is in the multi-agent learning in Markov Decision Processes, where agents need to calculate new policies while coming to an agreement on the underlying transition probabilities of the dynamic system.

Reaching consensus over a network has been investigated extensively in numerous fields^{11–20}. With regards to agreeing on probabilities, the seminal work of Winkler^{19,20} and DeGroot¹³ provides some of the earlier work on agreement on subjective probabilities. However, the distributed consensus problem was not addressed, in that there was no specific mention of a network structure, nor the limitations of a fully connected implementation. The work by Castañón^{11,12} also discusses agreement on probabilities, but, again, no specific mention of a network is made. Teneketzis and Varaiya¹⁸ and Washburn and Teneketzis²¹ also discuss the issue of agreement on the subjective beliefs of distributed decision-makers, but do not discuss the issue of graph connectivity. Belief propagation²² also is related to this problem, in the context of graphical models, but does not seem to include uncertainty (or confidences) in the probability. Also, the inference problem differs from this work in that the agents considered here are attempting to agree on the common belief, as opposed to simply aggregating evidence.

In a more general consensus framework, agreement on a parameter of interest is the key goal of a large set of linear consensus algorithms^{1,2,23}. These algorithms simply agree to some median value of the agents’ initial conditions, but do not consider any higher order uncertainties of the local beliefs such as the variance in the estimate. The work of Olfati-Saber *et al.*²⁴ and Spanos, Olfati-Saber, and Murray²⁵ discusses the issue of distributed hypothesis testing and distributed Kalman filtering, which did investigate agreement of beliefs, but did not address the issue of these beliefs being uncertain or imprecise.

When information is provided in a Kalman filtering setting, Ren *et al.*^{1,26} addressed the problem of reaching consensus on information by accounting for the error covariance, giving rise to Kalman Consensus algorithms. This work was extended to account for bias compared to the centralized solution by Alighanbari *et al.*²⁷. However, Kalman Consensus is derived assuming Gaussian beliefs, which do not address the issue of consensus on probabilities in that it does not encode that the information is constrained to be a probability. In particular, the information vector of Kalman consensus is not constrained to lie in the probability simplex. Additionally, most probability estimation schemes are formulated using the non-Gaussian Dirichlet distribution^{5,28} for reasons that are discussed in Section III, which further invalidates the use of Kalman Consensus for the imprecise probability agreement problem.

The key contribution of this paper is the *hyperparameter consensus* protocol as applied to the Dirichlet distribution, which converges to the centralized Bayesian estimate given all agents’ local

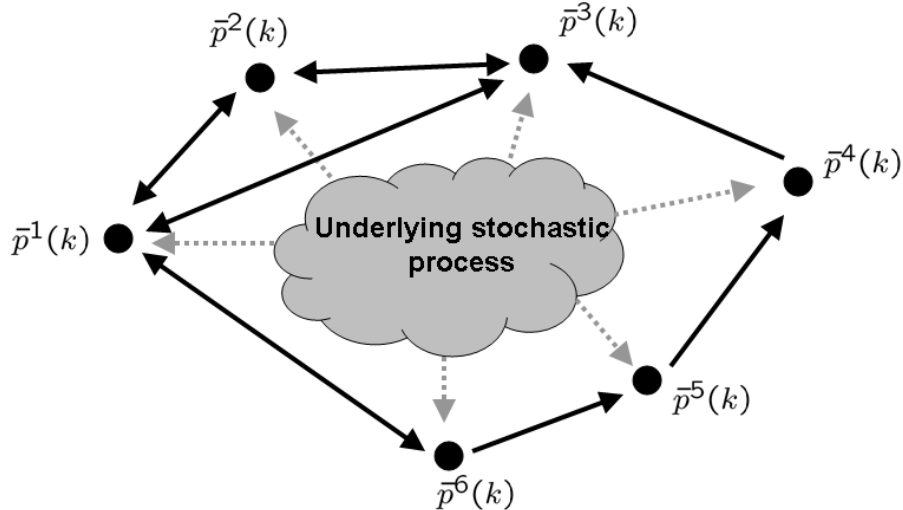


Figure 1. Example of a multi-agent system estimating a probability

uncertain information. This protocol extends two key principles from the Kalman Consensus algorithm: The first is the use of *conjugate* priors for the estimation problem, where the distribution on the value of a parameter is of the same form before and after a measurement is incorporated, and allows for simple analytical measurement updates to the parameters of the prior, or *hyperparameters*. The second concept is to treat other agents' independent information as pseudo-measurements, and apply the same local update to incorporate this information as would be applied to a true measurement. It will be shown that, in order to maintain an uncertain estimate of a parameter, each agent can, instead, maintain precise knowledge of the corresponding hyperparameters, and agreement on these hyperparameters will then allow the agents to agree to the (still uncertain) centralized Bayesian parameter estimate itself.

This paper is outlined as follows: The problem is defined in Section II and an existing linear consensus method is described as it applies to agreement on well-known, precise probabilities. Section III introduces the notion of the Dirichlet distribution to represent uncertainties in the probabilities and derives the hyperparameter consensus protocol to achieve consensus on the beliefs while accounting for their confidence. Section IV provides a formal algorithm and results for the static consensus case, while Section V discusses the issue that arises when the agents are contemporaneously attempting to reach consensus while making new observations regarding their probability estimates. We conclude with an implementation of the proposed hyperparameter consensus to a machine repair problem, and some insights for interesting venues for future work.

II. Problem Statement

The objective of this paper is to achieve a consensus on a probability vector $p(k) \in \mathfrak{R}^M$ among a group of N agents (see Figure 1). In other words, with each agent i having a belief, $p^i(k)$, the objective is to asymptotically reach an agreement such that $p^1(k) = p^2(k) = \dots = p^M(k)$. The agent connectivity is described by a general graph $G(E, V)$, where E denotes the edges of the graph and V denotes the vertices, and the neighbors (or neighboring nodes) of agent i are denoted by N_i . The results in this paper require a strongly connected network, but no additional assumptions on graph topology are required.¹

II.A. Consensus on Precise Probabilities

One method to achieve consensus on a precise probability is to use ideas from linear consensus extended directly to the belief¹ or a linear agreement on the log likelihood²⁴. If the beliefs are uncertain, one approach is to neglect the uncertainties and agree upon only on the *mean belief*, \bar{p} . In this case, each agent i runs a consensus algorithm on their mean belief $\bar{p}^i[k]$ of the form²⁴

$$\bar{p}^i[k+1] = \bar{p}^i[k] + \epsilon \sum_{j \in N_i} (\bar{p}^j[k] - \bar{p}^i[k]) \quad (1)$$

Here N_i is the collection of neighboring nodes of agent i , and ϵ is a constant (chosen by the designer) that depends on network connectivity. The ϵ is constrained in $\epsilon \in (0, 1/\Delta(G))$, where $\Delta(G)$ represents the maximal node degree of the network²⁴. A particularly appealing motivation for linear consensus is that can be shown that the consensus converges to

$$\lim_{k \rightarrow \infty} \bar{p}^i[k+1] = [\bar{p}^1[0] \ \bar{p}^2[0] \ \dots \ \bar{p}^N[0]] \nu = \frac{1}{N} \sum_{j=1}^N \bar{p}^j[0]$$

where ν is a convex weighting vector and can be determined from the network topology¹. If the network is balanced (incoming neighbors equals outgoing neighbors at each node), then $\nu = \frac{1}{N}$, and the steady state value is the arithmetic average,

$$\lim_{k \rightarrow \infty} \bar{p}^i[k+1] = \frac{1}{N} \sum_{j=1}^N \bar{p}^j[0]$$

If the network is not balanced, then the arithmetic average can still be achieved by scaling the initial consensus conditions by:

$$\bar{p}^i[0] \leftarrow \frac{\bar{p}^i[0]}{N\nu^i}$$

It is important to note that this method is immune to redundancies imposed by the topology of the network that generally plagues message passing schemes. By calculating the value of ν , each agent is able to determine exactly how influential their information is in the network and scale their initial conditions accordingly to eliminate network-induced biases.

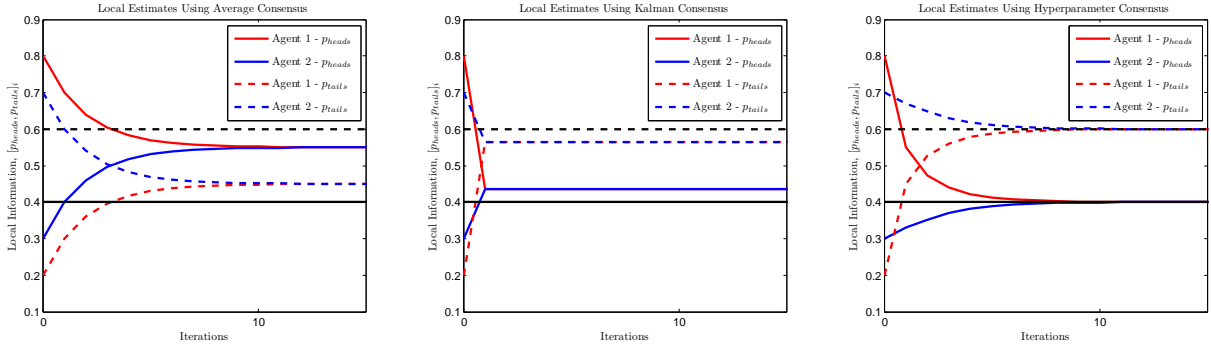
Finally, and perhaps most importantly, this approach preserves some key properties of the belief. Namely,

$$\begin{aligned} \lim_{k \rightarrow \infty} \mathbf{1}^T \bar{p}^i[k] &= 1, & \forall i \\ \mathbf{0} \leq \lim_{k \rightarrow \infty} \bar{p}^i[k] &\leq \mathbf{1}, & \forall i \end{aligned}$$

where the conditions hold for all k if the network is balanced.

However, such a linear consensus protocol does not incorporate sufficient information on the relative uncertainty of the probabilities. As a simple example, consider Figure 2(a) where two agents (blue and green) seek to come to some agreement on the bias of an unknown coin, where agent 1 expects a head with $p_{heads}^1 = 0.8$ and agent 2 with $p_{heads} = 0.3$. However, the first agent based its estimate after observing 4 heads in 5 tosses, while agent 2 saw only 6 heads in 20 tosses. The linear consensus solution is $p_{heads}^{LC} = 0.5(p_{heads}^1 + p_{heads}^2) = 0.55$, which is inconsistent with the underlying observations, which suggest that the result should be biased toward agent 2 since it has taken more measurements. Note that the linear consensus solution differs from the centralized solution (shown in black), found as the overall maximum likelihood estimate: $(4+6)/(5+20) = 0.4$.

While encoding system transitions or beliefs through probabilities represents a fundamental uncertainty in a system, the probabilities themselves are usually the outcome of a separate estimation



(a) Inconsistent agreement on a probability in a ring network using linear consensus
 (b) Inconsistent agreement on a probability in a ring network using Dirichlet mean and variance in the Kalman Consensus algorithm
 (c) Consistent agreement on a probability in a ring network using hyperparameter approach

Figure 2. Probability estimates can deviate from the centralized solution (left, middle) if the uncertainty in the probability is not properly accounted for.

process, and are in turn likely to be uncertain. Thus, a formal method needs to be accounted for to embed a more general notion of the uncertainty in the probabilities themselves. Such an approach is by the use of the Dirichlet distribution, described next.

III. Describing Imprecise Probabilities using the Dirichlet

A method that is frequently used in the artificial intelligence and statistics literature to describe uncertainty, or imprecision, in probabilities is the Dirichlet distribution²⁹. The Dirichlet distribution, f_D describing a probability vector $\mathbf{p} = [p_1, p_2, \dots, p_M]^T$ is defined as follows

$$\begin{aligned}
 f_D(\mathbf{p}|\alpha) &= K \prod_{i=1}^M p_i^{\alpha_i-1}, \quad \sum_i p_i = 1 \\
 &= K p_1^{\alpha_1-1} p_2^{\alpha_2-1} \dots (1 - \sum_{i=1}^{M-1} p_i)^{\alpha_M-1}
 \end{aligned}$$

where K is a normalizing factor that ensures the probability distribution integrates to unity and $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_M]^T$ (with $\alpha_i > 1$) is a vector of hyperparameters that defines the shape of the distribution.

The mean and the variance of the Dirichlet distribution can be calculated directly as follows

$$\bar{p}_i = \alpha_i / \alpha_0 \quad (2)$$

$$\Sigma_{ii} = \frac{\alpha_i(\alpha_0 - \alpha_i)}{\alpha_0^2(\alpha_0 + 1)} \quad (3)$$

The primary reasons for using the Dirichlet distribution is that the mean \bar{p}_i satisfies the requirements of a probability vector $0 \leq \bar{p}_i \leq 1$ and satisfies the unit sum constraint $\sum_i \bar{p}_i = 1$. Furthermore, the hyperparameters α_i can be interpreted as “counts”, or the number of times that the particular event was observed.

An additional benefit of using the Dirichlet is that it can be updated analytically with measurements sampled from a Multinomial distribution. This is of particular importance in cases when the stochastic process is observable over time, since the prior distribution can then be updated with

this new information. The reason for the existence of analytical updates is because the Dirichlet distribution is conjugate to the Multinomial distribution²⁹, and performing a measurement update step on the Dirichlet at time t amounts to a simple addition of currently observed outcomes to the previously observed counts $\alpha(t)$. The posterior distribution $f_D(\mathbf{p}_{t+1}|\alpha(t+1))$ is given in terms of the prior $f_D(\mathbf{p}_t|\alpha(t))$ as

$$\begin{aligned} f_D(\mathbf{p}_{t+1}|\alpha(t+1)) &\propto f_D(\mathbf{p}_t|\alpha(t))f_M(\beta(t)|\mathbf{p}_t) \\ &= \prod_{i=1}^M p_i^{\alpha_i-1} p_i^{\beta_i} = \prod_{i=1}^M p_i^{\alpha_i+\beta_i-1} \end{aligned} \quad (4)$$

where $f_M(\beta(t)|\mathbf{p}_t)$ is a multinomial distribution with hyperparameters $\beta(t) = [\beta_1, \dots, \beta_M]$. Each β_i is the total number of times event i was observed. Upon receipt of the measurements $\beta(t)$, the parameters $\alpha(t)$ are updated in the following manner:

$$\alpha_i(t+1) = \alpha_i(t) + \beta_i(t) \quad \forall i$$

The updated mean and variance can then be calculated by using Eqs. 2 and 3.

III.A. Consensus on Imprecise Probabilities

The following section discusses consensus while accounting for the uncertainty in the probability. In typical consensus algorithms, a convergence criterion has to be met before consensus is achieved, and for the purposes of these sections, the *convergence criterion* is that the difference between the mean beliefs satisfy $\|\bar{p}^i[k] - \bar{p}^j[k]\| \leq \delta$, where δ is some desired threshold.

One interpretation of an agreement problem is to update each agent's information through a fictitious information update from the other agents, the result of which can give a Bayesian combination of each agent's local beliefs, though it is important to avoid double-counting information. For our assumption of a Dirichlet prior on the probabilities, if the agents can exchange their information as Multinomial pseudo-measurements to their neighbors then the desired consensus to an updated Dirichlet is achieved, akin to the local update in Eq. 4. For a consensus initiated at time t , the desired aggregation of agent i 's prior $f_D(\mathbf{p}_t|\alpha^i(t))$ at time t with the information of each other agent j can be modeled as a Bayesian update and will give a desired converged value at time $t+1$ as

$$f_D(\mathbf{p}_{t+1}|\alpha^i(t+1)) \propto \underbrace{f_D(\mathbf{p}_t|\alpha^i(t))}_{\text{Local Dirichlet}} \times \underbrace{\prod_{j \neq i} f_M(\beta^j(t)|\mathbf{p}_t)}_{\text{Neighborhood Pseudo-Measurement}}$$

where it can be found that, letting $f_D(\mathbf{p}_{t-}|\alpha^i(t^-))$ represent any shared information across the network prior to this consensus stage, the Multinomial pseudo-measurement is found by³⁰

$$f_M(\beta^i(t)|\mathbf{p}_t) \propto \frac{f_D(\mathbf{p}_t|\alpha^i(t))}{f_D(\mathbf{p}_{t-}|\alpha^i(t^-))}$$

Thus, similar to the estimation update, the consensus update can be simplified to an addition of

the counts since

$$\begin{aligned}
f_D(\mathbf{p}_{t+1}|\alpha^i(t+1)) &\propto f_D(\mathbf{p}_t|\alpha^i(t)) \times \prod_{j \neq i} \frac{f_D(\mathbf{p}_t|\alpha^j(t))}{f_D(\mathbf{p}_t|\alpha^j(t^-))} \\
&\propto \left(\prod_{m=1}^M p_m^{\alpha_m^i(t)-1} \right) \prod_{j \neq i} \frac{\prod_{m=1}^M p_m^{\alpha_m^j(t)-1}}{\prod_{m=1}^M p_m^{\alpha_m^j(t^-)-1}} \\
&= \prod_{m=1}^M p_m^{\alpha_m^i(t) + \sum_{j \neq i} (\alpha_m^j(t) - \alpha_m^j(t^-)) - 1} \\
&= f_D \left(\mathbf{p}_t \left| \alpha^i(t) + \sum_{j \neq i} (\alpha^j(t) - \alpha^j(t^-)) \right. \right) \\
&= f_D \left(\mathbf{p}_t \left| \alpha(t^-) + \sum_{j=1}^N \beta^j(t) \right. \right) \tag{5}
\end{aligned}$$

where we have assigned $\beta^i(t) = \alpha^i(t) - \alpha(t^-)$.

Linear consensus methods, such as Equation 1, have been shown to be able to converge to an average of the initial conditions of each agent¹. Therefore, if the initial local information is scaled by the number of agents in the network, N , the result will be a *sum* consensus. Here, it is convenient to distinguish the time-scale denoted using (t) to represent subsequent executions of the consensus problem, as opposed to the iteration counter, $[k]$, that define particular steps within the consensus procedure. Thus, if the agents were to run a linear consensus protocol on the scaled Multinomial pseudo-measurement counts, β , at time step t , they would obtain

$$\beta^i(t+1) = \lim_{k \rightarrow \infty} \beta^i[k+1] = \frac{1}{N} \sum_{j=1}^N \beta^j[0] = \frac{1}{N} \sum_{j=1}^N N \beta^j(t) = \sum_{j=1}^N \beta^j(t)$$

where we have defined the initial consensus information as a scaling of the pseudo-measurement counts: $\beta^j[k]|_{k=0} = N \beta^j(t)$. This result can then be used to update the local Dirichlet counts according to

$$\alpha^i(t+1) = \alpha(t^-) + \sum_{j=1}^N \beta^j(t) = \alpha(t^-) + \beta^i(t+1) \tag{6}$$

and derive the desired Bayesian consensus estimate as

$$\bar{p}_j^i(t+1) = \frac{\alpha_j^i(t+1)}{\alpha_0^i(t+1)}$$

This is the premise behind the hyperparameter consensus method, whereby linear consensus results are applied to the hyperparameters (the counts) of the parameters of interest (the probabilities), which effectively weights the more confident agents more and leads to convergence to the proper centralized Bayesian estimate. Note that this approach directly accounts for the uncertainty in the belief of each agent since the local uncertain distributions (the Dirichlets) are explicitly determined by the counts (α 's), and, conversely, the counts contain all information required to reconstruct the uncertainties. Therefore, by summing the counts through the consensus method, the solution will be (correctly) biased towards the agent that has the highest counts, and this will be reflected in the calculation of the mean belief.

Algorithm 1 Dirichlet Consensus on Initial Conditions

- 1: Choose a convergence criterion δ
- 2: Initialize counts: $\beta^i[0] \leftarrow \frac{\alpha^i(t) - \alpha(t^-)}{\nu^i}$
- 3: Initialize iteration counter: $k \leftarrow 0$
- 4: **while** Convergence criterion not satisfied **do**
- 5: Update counts:

$$\begin{aligned}\beta^i[k+1] &= \beta^i[k] + \epsilon \sum_{j \in N_i} (\beta^j[k] - \beta^i[k]) \\ \alpha^i[k+1] &= \beta^i[k+1] + \alpha(t^-)\end{aligned}$$

- 6: Find the mean belief:

$$\begin{aligned}\bar{p}^i[k+1] &= \int_0^1 \mathbf{p}_{k+1} f_D(\mathbf{p}_{k+1} | \alpha^i[k+1]) d\mathbf{p}_{k+1} \\ &= \frac{\alpha^i[k+1]}{\sum_j \alpha_j^i[k+1]}\end{aligned}\tag{7}$$

- 7: Update counter: $k \leftarrow k + 1$
 - 8: Evaluate convergence criterion
 - 9: **end while**
 - 10: Update local probabilities: $\bar{p}^i \leftarrow \bar{p}^i[k]$
-

IV. Results on the agreement problem without information updates

In this section, it is assumed that each agent’s knowledge is static and the problem is to agree upon the initial uncertain beliefs. The problem follows from the previous discussion by letting $t = 0$ and assuming all the agents may have completely independent information and must agree, effectively, on their local Dirichlet *priors*. If this is the case, we let $\alpha(0^-) \rightarrow 0$, implying that there is no shared information among the agents, at which point the desired consensus result is simply a sum of all the local counts, $\beta^i(0) = \alpha^i(0)$. This is the formulation for the “pure consensus” problem where agents with disparate beliefs need to agree. An algorithm to solve this problem is presented in Algorithm 1. Note that in this algorithm the belief is (by construction) constrained to lie in the unit simplex since the normalization (Eq. 7) occurs at each consensus iteration. Figure 2(c) shows the result of applying the algorithm to the unfair coin problem presented in Section II, and that each agent is able to correctly converge to the proper combined estimate. This result is compared to a Kalman Consensus-based approach^a in Figure 2(b), where it is apparent that the result is biased in favor of agent 2 since it has more measurements and therefore a smaller variance, but the result is still biased since the uncertainties are not properly described by a Normal distribution.

^aSpecial consideration is required when using the Kalman Consensus algorithm with the moments of the Dirichlet since the covariance matrix has zero row- and column-sums³¹ and is therefore not invertible. To compensate, the KC algorithm is only applied to a $\{m-1 \times m-1\}$ submatrix of Σ and the $\{m-1 \times 1\}$ subvector of $\bar{\mathbf{p}}$, which are then re-formed after each iteration to find the full covariance matrix and probability vector.

V. Results on the agreement problem with information updates

In the previous section, the consensus problem focused on agreeing over the *prior* uncertain probabilities, but in this section we consider the problem of agreeing over the uncertain probabilities while each agent receives independent observations of the underlying stochastic process. In this sense, this problem is closely related to the problem of Distributed Kalman filtering with consensus investigated by Olfati-Saber²⁴.

Two important factors must be considered for consensus with measurements: first, if a consensus update has been concluded in the past and a new one is to begin soon, the agents must account for the shared information from the previous consensus to avoid double-counting; and second, if a measurement is made while the agents are running the consensus protocol it must be handled properly. The first problem is common in the data fusion community and is generally handled by using *channel filters*³², which maintain a measure of the shared information between each agent pair. These can become very complex in even simple networks, and prevent the scalability of many data fusion systems. However, in the formulation at hand, the problem of double-counting can be addressed by setting the shared global information, $\alpha(t^-)$, equal to the counts resulting from the most recent consensus. This is a common value among all agents, so there is no need to calculate and store the shared information between every *pair* of agents. By keeping track of the global information, agents can effectively remove these counts from the new consensus so that they will not be treated as new, independent information by other agents.

The second problem arises due to the fact that consensus does not occur instantaneously, and, therefore, it is possible for measurements to be received *during* consensus as well as before and after it. If a measurement occurs before a consensus update has begun, then the consensus can be run simply with initial conditions that reflect the new measurement. If, however, a local measurement, corresponding to a count update of β , is received by agent i at iteration k of the consensus, the desired steady state value changes and is found by applying the measurement update in Eq. 4 to the originally desired Dirichlet, $f_D(\mathbf{p}|\alpha(t^-) + \sum_{i=1}^N \beta_i(t))$, which gives the updated counts:

$$\alpha(t^-) + \sum_{i=1}^N \beta_i(t) + \beta$$

To converge to this new value, the measurement must be incorporated as³⁰

$$\beta^i[k] \leftarrow \beta^i[k] + \frac{\beta}{\nu^i} \quad (8)$$

by agent i . This is done online so that the agents are constantly responding to their latest observations, which are assumed to be independent identically distributed (IID) with respect to the other agents.

Note that if no measurements are made during consensus, then this approach collapses to the no-measurement case in the preceding section. Further, the weighted information update suggested in Eq. 8 is not accurate *locally* due to the weighting, and should only be used if a measurement occurs during the consensus phase such that the steady-state value is accurate. If a measurement occurs while the agents are not attempting to agree, then it should be incorporated according to the normal measurement update in Eq. 4 such that the local estimate remains representative of the observed information.

Finally, an alternate formulation will be introduced here which does not require the α ‘reconstruction’ step in Equation 6, and is achieved by running consensus on $\alpha[\cdot]$ itself, where $\alpha^i[0]$ is given by

$$\alpha^i[0] = \alpha(t^-) + \frac{\beta^i[0]}{\nu^i}$$

Algorithm 2 Dirichlet Aggregation at time t with possible information update

- 1: Choose a convergence criterion δ
- 2: Initialize counts: $\alpha^i[0] \leftarrow \alpha(t^-) + \frac{\beta^i(t)}{\nu^i}$
- 3: Initialize iteration counter: $k \leftarrow 0$
- 4: **while** Convergence criterion not satisfied **do**
- 5: **if** Measurement received by agent i **then**
- 6: Update local counts: $\alpha^i[k] \leftarrow \alpha^i[k] + \frac{\beta}{\nu^i}$
- 7: **end if**
- 8: Update counts:

$$\alpha^i[k+1] = \alpha^i[k] + \epsilon \sum_{j \in N_i} (\alpha^j[k] - \alpha^i[k])$$

- 9: Find the mean belief:

$$\begin{aligned} \bar{p}^i[k+1] &= \int_0^1 \mathbf{p}_{k+1} f_D(\mathbf{p}_{k+1} | \alpha^i[k+1]) d\mathbf{p}_{k+1} \\ &= \frac{\alpha^i[k+1]}{\sum_j \alpha_j^i[k+1]} \end{aligned}$$

- 10: Update counter: $k \leftarrow k + 1$
 - 11: Evaluate convergence criterion
 - 12: **end while**
 - 13: Update local counts: $\alpha^i(t+1) \leftarrow \alpha^i[k]$
 - 14: New local distribution is $f_D(\mathbf{p}_{t+1} | \alpha^i(t+1))$
 - 15: Update the time step $t \leftarrow t + 1$
-

It is easy to show that running the consensus protocol in Equation 1 on $\alpha[0]$ as defined above will also result in the outcome required in Equation 5.

The main steps for this alternate approach are shown in Algorithm 2. Just like in the previous section, consensus on the hyperparameters properly accounts for the relative confidence of each agent. Figure 3 shows the result of running Algorithm 2 on an *unbalanced* network, where the jump from ‘initial value’ to ‘weighted value’ refers to the count initialization step (2). A measurement is made by the cyan agent on the 20th consensus iteration which augments the centralized estimate to which the agents converge by iteration 40.

VI. Machine Repair Problems

This section will use a simple machine repair problem to highlight the applicability of hyperparameter consensus to large-scale, cooperative learning problems. Before introducing the multi-machine problem, the single-machine problem is described using a model-based learning approach. The multi-machine repair problem is then introduced as a cooperative, distributed estimation problem where multiple operators are attempting to learn the dynamics of a group of identical machines. Though the problem presented is fairly simple in order to obtain comprehensible results, the prescribed approach can be adapted to many multi-agent learning and estimation problems that fall within the MDP framework.

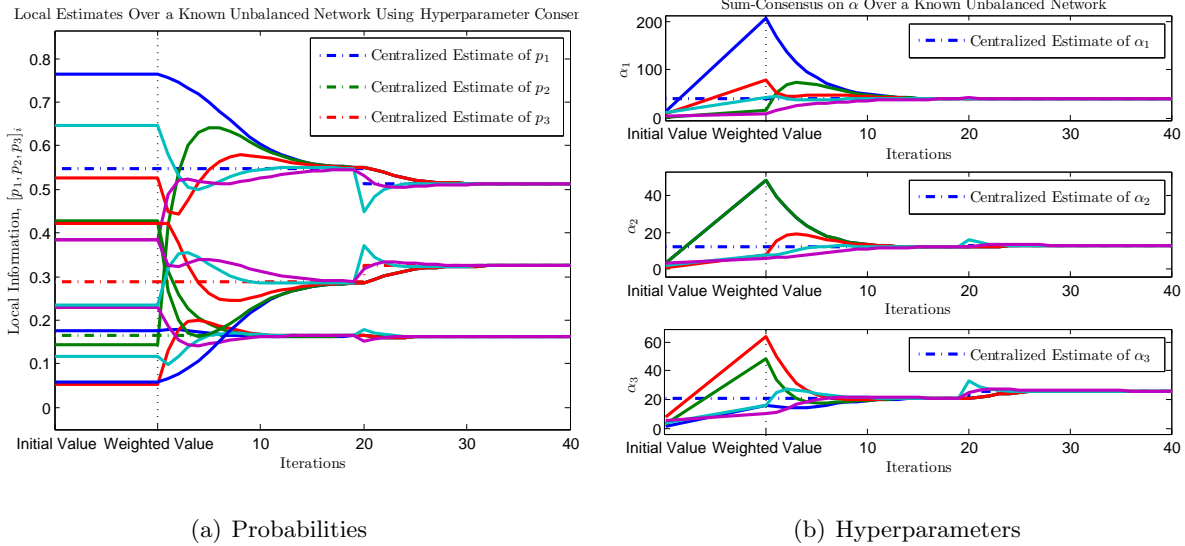


Figure 3. Left: Probability estimates can track the centralized solution even with a measurement during execution of the hyperparameter consensus algorithm. Right: the hyperparameter counts track the sum of the initial counts plus the measurement.

VI.A. The Single-Machine Repair Problem

The single-machine repair problem is a simple, two-state MDP in which, at time t , the machine is either broken ($s_t = 0$) or working ($s_t = 1$). If the machine is working, the operator receives \$100 and can choose to perform maintenance on the machine (action m) which will improve the likelihood of the machine working at the next time step, but at a cost of C_m for parts and labor. The operator could, alternately, opt to do nothing at no additional cost (action n), but with an increased risk of the machine breaking. If the machine is broken, the operator can either fix it (action f) for a fee of C_f or replace it (action r) for C_r , where it is constrained that, for any reasonable problem, the cost to replace is more than to fix. If the machine is replaced, then the new machine is guaranteed to work for the next stage. The model is outlined below:

$$\mathbb{S} = \begin{cases} \text{broken} : s = 0 \\ \text{working} : s = 1 \end{cases}$$

If the machine is working, $s_t = 1$:

$$\begin{aligned} \mathbb{A}(s_t = 1) &= \begin{cases} \text{maintenance} : a_t = m \\ \text{nothing} : a_t = n \end{cases} \\ R(s_t = 1, a_t) &= \begin{cases} 100 - C_m & \text{if } a_t = m \\ 100 & \text{if } a_t = n \end{cases} \\ p(s_{t+1} | s_t = 1, a_t = m) &= \begin{cases} 1 - p_m & \text{for } s_{t+1} = 0 \\ p_m & \text{for } s_{t+1} = 1 \end{cases} \\ p(s_{t+1} | s_t = 1, a_t = n) &= \begin{cases} 1 - p_n & \text{for } s_{t+1} = 0 \\ p_n & \text{for } s_{t+1} = 1 \end{cases} \end{aligned}$$

If the machine is broken, $s_t = 0$:

$$\begin{aligned} \mathbb{A}(s_t = 0) &= \begin{cases} \text{fix} : a_t = f \\ \text{replace} : a_t = r \end{cases} \\ R(s_t = 0, a_t) &= \begin{cases} -C_f & \text{if } a_t = f \\ -C_r & \text{if } a_t = r \end{cases} \\ p(s_{t+1}|s_t = 0, a_t = f) &= \begin{cases} 1 - \mathbf{p}_f & \text{for } s_{t+1} = 0 \\ \mathbf{p}_f & \text{for } s_{t+1} = 1 \end{cases} \\ p(s_{t+1}|s_t = 0, a_t = r) &= \begin{cases} 0 & \text{for } s_{t+1} = 0 \\ 1 & \text{for } s_{t+1} = 1 \end{cases} \end{aligned}$$

The problem considered here assumes that the operator knows the costs associated with each state action pair (ie. R is fully known), but the transitions $T(s_{t+1}|s_t, a_t)$ are unknown and must be learned. When the machine is working, the transition matrix is parameterized by the probabilities \mathbf{p}_m and \mathbf{p}_n , which designate the probability of the machine working at the next time step after, respectively, performing maintenance, m , or taking no action, n :

$$T(\cdot|s_t = 1, a_t) = \begin{bmatrix} 1 - \mathbf{p}_m & \mathbf{p}_m \\ 1 - \mathbf{p}_n & \mathbf{p}_n \end{bmatrix} = \begin{bmatrix} \mathbf{p}_m \\ \mathbf{p}_n \end{bmatrix}$$

When the machine is broken, it is assumed that replacing the machine is known to guarantee a working state at the next stage, so the transition matrix is parameterized only by the probability of the machine working given having fixed it, \mathbf{p}_f :

$$T(\cdot|s_t = 0, a_t) = \begin{bmatrix} 1 - \mathbf{p}_f & \mathbf{p}_f \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{p}_f \\ \mathbf{p}_r \end{bmatrix}$$

The objective is to determine an optimal policy for each machine state, $\pi^*(s)$, that maximizes the time-discounted future reward

$$\pi^*(s) = \arg \max_{\pi} V^{\pi}(s) = \arg \max_{\pi} E \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t)) \middle| s_0 = s \right] \quad (9)$$

where the expectation is taken over the state transitions, $T(s_{t+1}|s_t, \pi(s_t))$. This problem can be solved using value iteration, where the recursion becomes

$$Q(s, a) = R(s, a) + \gamma \sum_{s' \in \mathbb{S}} T(s'|s, a) V_i(s') \quad (10)$$

$$V_{i+1}(s) = \max_{a \in \mathbb{A}(s)} Q(s, a) \quad (11)$$

where $V_0(s)$ is initialized arbitrarily and the recursion is run until convergence. The simulations presented here use a time discount factor $\gamma = 0.8$.

Uncertainty in the probability vectors $\mathbf{p}_a = [1 - \mathbf{p}_a \ \mathbf{p}_a]$ can be described by a Beta distribution, the univariate form of the Dirichlet. The Beta distribution for action a is parameterized by counts α_a and β_a , which represent the number of times action a has resulted in a broken or working state, respectively. As an operator explores the state-action space, the counts are updated to reflect the observed transitions and are used to determine new probabilities for the solver. An undirected Boltzmann method will be used to capture the exploration vs exploitation trade-off.

This approach utilizes the currently expected Q-values to determine the probability of selecting each action according to

$$Pr(a|s) = \frac{e^{Q(s,a)/T}}{\sum_{a'} e^{Q(s,a')/T}}$$

where T is a “temperature” parameter that starts large (so all actions are equally likely) and is set to decay over time to promote greedy selection in the long-run.

VI.B. The Multi-Agent Machine Repair Problem

In the multi-agent setting, N machines are run independently by N agents, but the agents are allowed to share information among themselves to try and agree upon the model of the system. Since each agent observes an independent machine, the measurements it makes are independent from the state or measurements of any of the other machines. For comparison, the hyperparameter consensus algorithm and the average consensus algorithm were used, where the average consensus was run on the local probability estimates themselves. The results shown here are for a range of network sizes where agents take a certain number of measurements then run a each of the consensus algorithms to convergence and evaluate the resulting agreed-upon model. Thus, this comparison shows the relative value of each consensus algorithm at varying stages of the learning process. Unfortunately, the comparison cannot be easily extended to concurrent estimation and consensus problems since the estimation problem utilizes the Dirichlet counts, which are preserved by the hyperparameter consensus method but are not uniquely defined after executing average consensus on the probabilities themselves^b.

The following comparisons not only show the benefit of using the hyperparameter consensus from a performance standpoint, but through maintaining representative counts, hyperparameter consensus is able to work concurrently with the Bayesian estimation scheme where other approaches (Kalman Consensus and Average Consensus) cannot. Finally, it is necessary to note that the results shown are obtained only after the agents have converged on the model parameters. This assumption allows the results to be generalized to an arbitrary, known, strongly connected network such that any inconsistencies in the probabilities that occur during the transient of the average consensus algorithm are ignored and only the resulting steady-state values (which will lie in the unit simplex) are used.

VI.C. Formulation and Results

Each local learning problem was simulated with the true model as given below. If the machine is working at a given time, the cost of maintenance is $C_m = \$10$ and raises the likelihood of the continued functionality of the machine from 0.5 to 0.9:

$$R(s_t = 1, a_t) = \begin{cases} 90 & \text{if } a_t = m \\ 100 & \text{if } a_t = n \end{cases}$$

$$p(s_{t+1}|s_t = 1, a_t = m) = \begin{cases} 0.1 & \text{for } s_{t+1} = 0 \\ 0.9 & \text{for } s_{t+1} = 1 \end{cases}$$

$$p(s_{t+1}|s_t = 1, a_t = n) = \begin{cases} 0.5 & \text{for } s_{t+1} = 0 \\ 0.5 & \text{for } s_{t+1} = 1 \end{cases}$$

^bSince probabilities alone are not enough to uniquely define the counts, additional information, like the variance, is required but unavailable when using average consensus. Further, though the mean and variance will uniquely define the counts, applying a Kalman Consensus Filter using the mean and variance was shown to provide biased results by Fraser³⁰, and is not considered here for that reason

If the machine is broken, however, the operator can either spend \$70 to fix the machine or \$110 to replace it. Since the machine had been previously broken, extra stress may have fatigued other parts, such that fixing it leads to a working machine at the next step only 80% of the time.

$$R(s_t = 0, a_t) = \begin{cases} -70 & \text{if } a_t = f \\ -110 & \text{if } a_t = r \end{cases}$$

$$p(s_{t+1}|s_t = 0, a_t = f) = \begin{cases} 0.2 & \text{for } s_{t+1} = 0 \\ 0.8 & \text{for } s_{t+1} = 1 \end{cases}$$

$$p(s_{t+1}|s_t = 0, a_t = r) = \begin{cases} 0 & \text{for } s_{t+1} = 0 \\ 1 & \text{for } s_{t+1} = 1 \end{cases}$$

The results that follow are the outcome of 300 Monte-Carlo simulations to determine the average response given the true model and some set initial conditions. Three primary metrics will be used to judge the quality of the learning:

- Probability of finding the optimal policy: This metric determines the likelihood of obtaining the optimal policy through value iteration using the best estimate of the transition probabilities. As more measurements are made, the transition probabilities will converge to the truth and will result in convergence to the optimal policy.
- Expected error in the discounted future reward: This metric describes the difference between the average future reward expected using the current model estimate and the true value of the state under the optimal policy:

$$\left| \hat{V}^{\hat{\pi}^*}(s) - V^{\pi^*}(s) \right|$$

where $\hat{\pi}^*$ is the estimated optimal policy found by solving Eq. 9 using the current best estimate of the transitions T , and $\hat{V}^{\hat{\pi}^*}(s)$ represents the converged result of the recursion in Eqs. 10 and 11 with actions dictated by the estimated optimal policy. Like the probability of optimal policy metric, the estimated value function is expected to converge to the truth with infinite measurements, but is a smoother, continuous metric that is an overall measure of how well the MDP model has been learned.

- Error in model parameters: In addition to MDP performance metrics, estimation performance will be evaluated using the expected values of the parameters themselves.

Using a “flat” initial prior on the counts, such that $\alpha_a = \beta_a = 1, \forall a \neq r$ (since replacing is a deterministic transition), and assuming that this prior information is common among the agents, the multi-agent machine repair problem was simulated for a range of agents from $N = 2$ to $N = 200$. To account for the shared initial prior information, two methods for storing, updating, and sharing hyperparameters have been suggested earlier:

- Algorithm 1: Each agent maintains a local estimate of the probabilities given by its current hyperparameters as well as knowledge of any shared, global information. Each agent can then run the hyperparameter consensus algorithm on the difference between the current local hyperparameters and the shared hyperparameters, and then add the result back to the shared hyperparameter value after convergence.
- Algorithm 2: Each agent can update their prior with weighted measurements as in Equation 8, leaving the prior un-weighted (ie. do not multiply the prior by $1/\nu_i$). During consensus, this has the effect of running average consensus on the prior and sum consensus on the measurements, such that the prior remains counted only once but the measurements are given the desired weight.

In situations where there are a large number of measurements that separate complete consensus executions (such as when agents take measurements without communication, stop taking measurements and run consensus to convergence, then continue taking measurements), the first approach is likely desirable since the estimation and agreement problems are effectively separated. Further, the first method allows each agent to maintain a local estimate that is not impacted by the consensus weights ($1/\nu_i$) that are applied for hyperparameter consensus. If the consensus epochs are well defined in the sense that every agent knows when to start communicating and when convergence is reached, then the problem of maintaining knowledge of the shared information is not difficult since it is simply the result of the most recent consensus.

If it is desired that consensus and estimation occur simultaneously, the second approach is beneficial as it requires no distinction between an estimation epoch and a consensus epoch as either a measurement can be made or messages sent or received at any time without any additional processing (such as subtracting and adding shared information in the previous method). Since the comparison problem to be addressed here fits into the first framework, this method will be used for the following discussion.

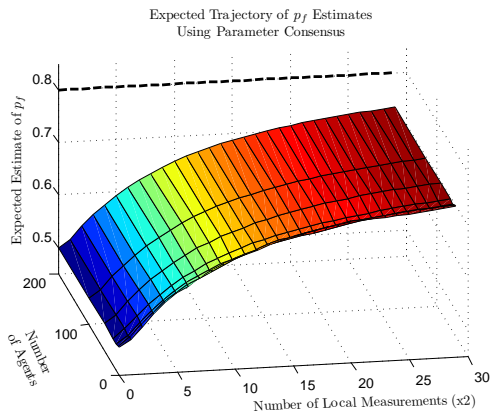
VI.C.1. Simulation Results

Using the hyperparameter consensus algorithm, the agents are able to utilize every measurement made by each agent and are able to learn the model much faster than using average consensus on the probabilities, as shown in Figure 4. Since all agents have an identical prior, both methods start with the same initial estimate. However, using the hyperparameters allows the agents to effectively combine their observations, such that 100 agents with 2 measurements each have the estimation power of a single agent with 200 measurements. Using average consensus on the probabilities themselves, it is apparent that the estimates follow the same profile regardless of the number of agents, suggesting that there is little or no benefit to the estimation problem from agreement through averaging of the probabilities. Further, recall that the use of the linear consensus method for combining the hyperparameters implicitly accounts for the network topology, so there is no need for channel filtering on complex networks as would be required to obtain the same result using traditional distributed data fusion techniques.

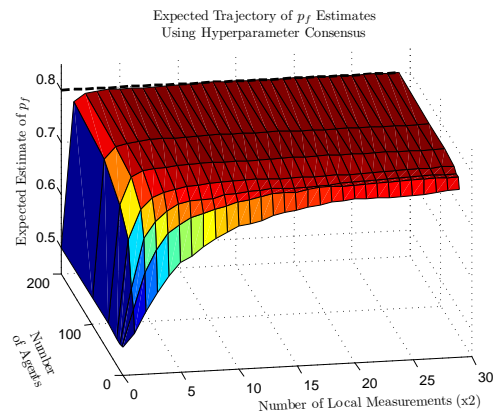
This expedited learning ability allows the agents running the hyperparameter consensus to obtain the optimal policy more often and after fewer local measurements than parameter consensus, as shown in Figure 5. The top plot shows the main transient regime where hyperparameter consensus is able to find the true optimal policy nearly all the time, while parameter consensus methods almost always achieve a sub-optimal policy after consensus. This transient period is where it is important to use hyperparameter consensus since it can take advantage of the combined measurements of all the agents. However, initially all agents have equally limited information and, further, after sufficient local measurements have been made, each agent will individually converge to the true model so that, initially and in the limit, the two methods will give equivalent performance results.

Figures 6 and 7 show the expected evolution of the discounted future reward given the best estimate of the model. As with the other results, initially and in the limit of infinite measurements the two approaches give the same results. In the transient period, however, the hyperparameter consensus method converges much faster to the true future reward. The top plot in both figures shows the difference between the errors in two estimates obtained, and highlights the fact that the error obtained from using the hyperparameter consensus is less than that obtained using parameter consensus by an amount up to over 50% of the true value.

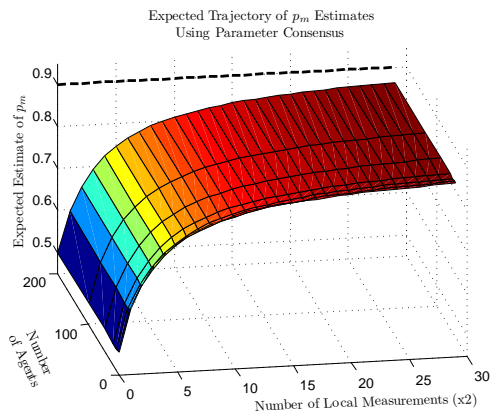
Similar results can be obtained using more informative priors or unique local priors that aren't shared across all agents. In the first case, the results are the same as those shown here but effectively



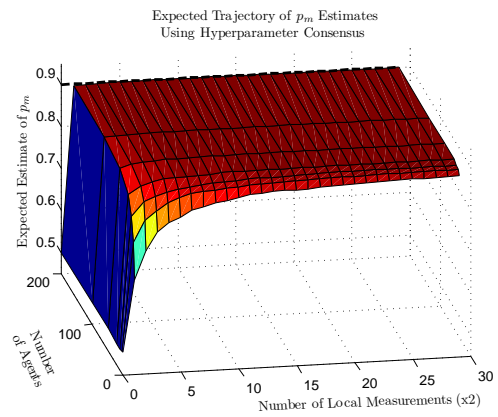
(a) Average Consensus on p_f



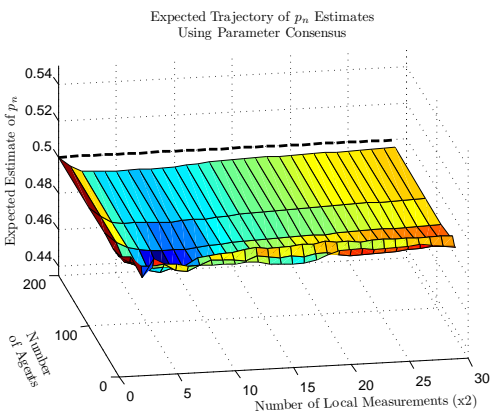
(b) Hyperparameter Consensus on p_f



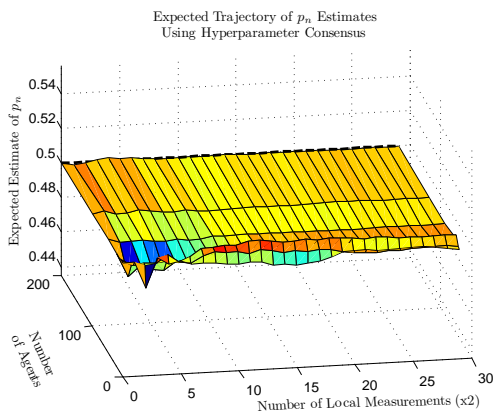
(c) Average Consensus on p_m



(d) Hyperparameter Consensus on p_m

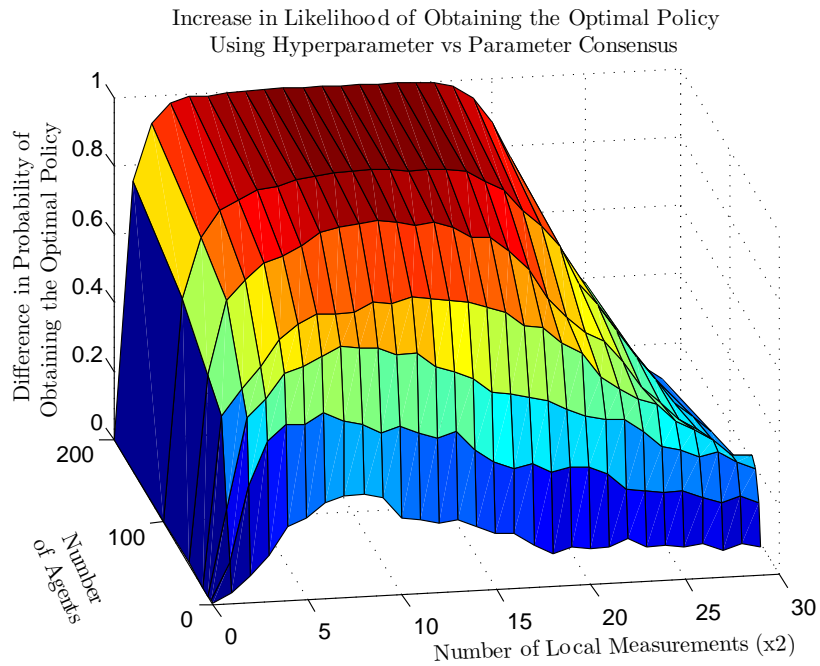


(e) Average Consensus on p_n

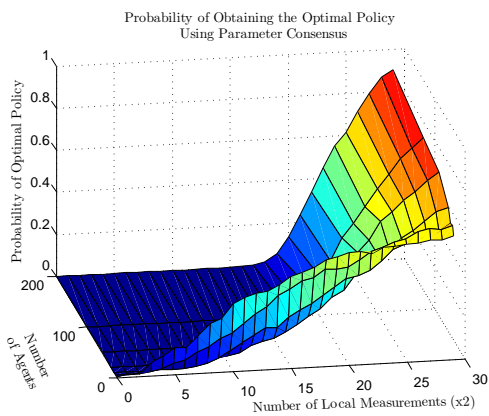


(f) Hyperparameter Consensus on p_n

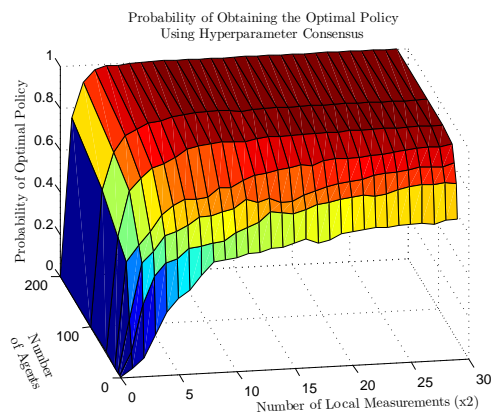
Figure 4. Expected Probability Estimates Using Each Consensus Method for Varying Number of Local Measurements



(a) Increased in Probability of Obtaining the Optimal Policy using Hyperparameter Consensus over Average Consensus on the Probabilities

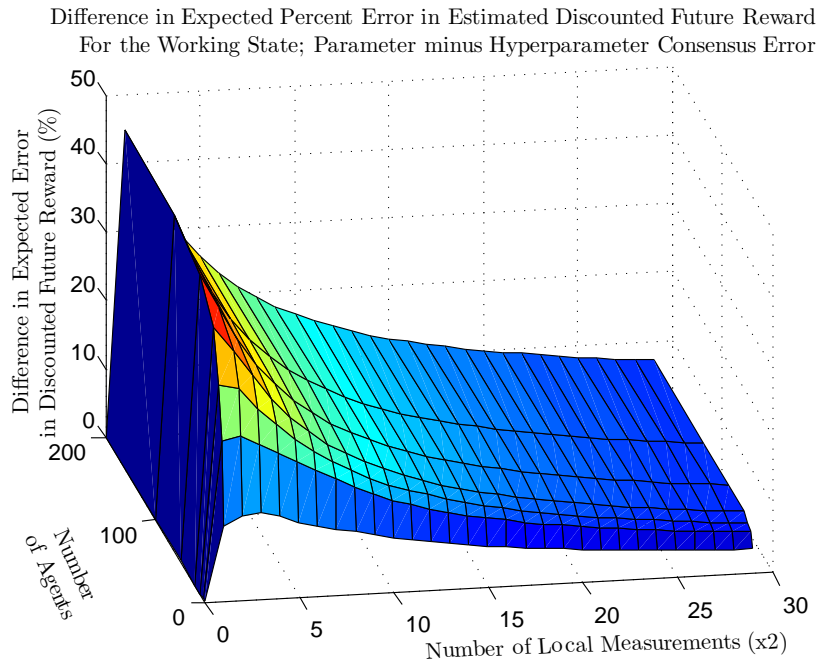


(b) Average Consensus

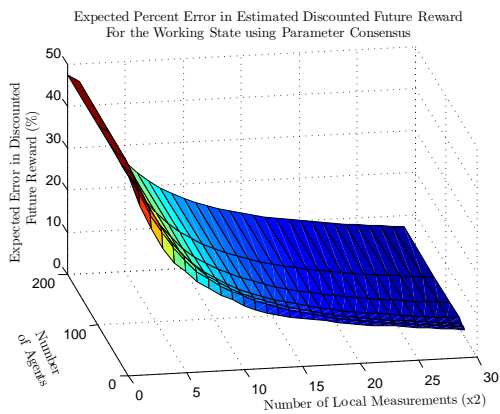


(c) Hyperparameter Consensus

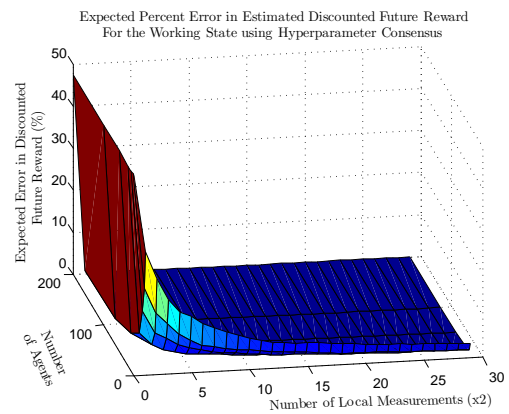
Figure 5. Probability of Obtaining the Optimal Policy Using Each Consensus Method for Varying Number of Local Measurements



(a) Difference in Expected Error in the Discounted Future Reward for the Working State using Hyperparameter Consensus over Average Consensus on the Probabilities

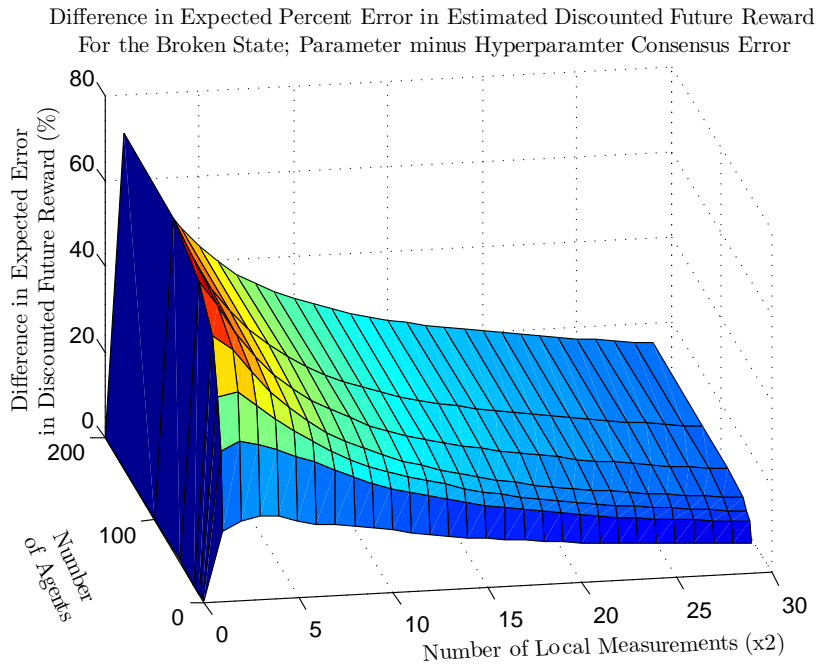


(b) Average Consensus

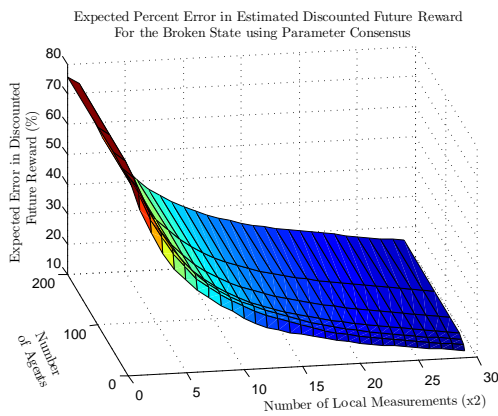


(c) Hyperparameter Consensus

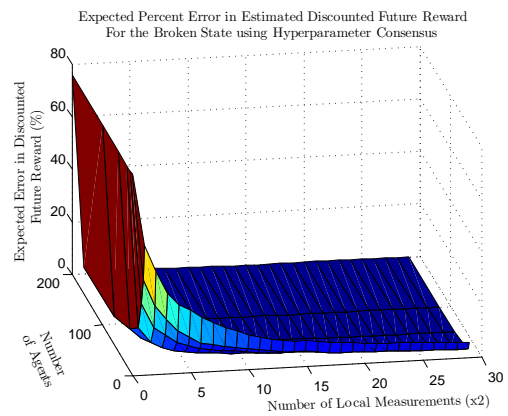
Figure 6. Expected Error in the Discounted Future Reward for the Working State using Each Consensus Method for Varying Number of Local Measurements



(a) Difference in Expected Error in the Discounted Future Reward for the Broken State using Hyperparameter Consensus over Average Consensus on the Probabilities



(b) Average Consensus



(c) Hyperparameter Consensus

Figure 7. Expected Error in the Discounted Future Reward for the Broken State using Each Consensus Method for Varying Number of Local Measurements

started after taking some prior measurements, which would look like the tail end of the figures and would provide the same benefits as already described. The second extension has the same traits, but may not directly mimic the results shown here since each agent is allowed to have a different initial condition. It is important, however, for the initial conditions to still be representative, or else they will bias the results when using hyperparameter consensus, which will not be able to tell the difference between uninformed and informed hyperparameters. If the initial conditions are uncertain and could be biased, then it may be beneficial to add a fading factor to the measurements (akin to process noise in the Kalman Filter setting) such that old information is given less weight than newer information.

VII. Conclusions

This paper has discussed some notions of agreeing on an uncertain or imprecise probabilities in a network of agents. We have shown that ignoring or improperly accounting for the uncertainty in the probabilities can result in inconsistent estimates that are not representative of overall belief of all agents in the network. The hyperparameter consensus method was derived for use with the Dirichlet prior, and was shown to converge to a Bayesian combination of local beliefs with and without prior globally shared information. A method to incorporate measurements from a static stochastic process during execution of the hyperparameter consensus protocol was provided and shown to converge to the combined centralized estimate with measurements included. Further, the hyperparameter consensus method was able to achieve the same desired Bayesian updates as some data fusion problems but without the need for complex channel filters, which allows the hyperparameter method to scale to much larger networks.

Finally, the results were extended to the distributed estimation problem using the multi-agent machine repair scenario. It was shown that the hyperparameter consensus method was able to converge faster to the model parameters due to its ability to aggregate the observations made by all agents through their counts rather than just agreeing on the resulting probabilities. This led to greatly increased performance during the traditional learning transient in both the likelihood of obtaining the true optimal policy and estimation of the expected future reward.

Our ongoing work in the context of robust and adaptive MDPs is investigating the role of hyperparameter consensus on transition probability models for the purposes of robust planning and addressing the exploration vs. exploitation trade-off. Ongoing work is also investigating the impact of heterogeneity in the team and the question of fault detection in some of the sensors, namely the ability to identify if there is a sensor that is broadcasting erroneous information, thereby biasing the team to the wrong probability.

Acknowledgments

This research funded under AFOSR Grant FA9550-08-1-0086.

References

- ¹ Ren, W., Beard, R. W., and Atkins, E. M., “A survey of consensus problems in multi-agent coordination,” *American Controls Conference*, 2005, pp. 1859–1864 vol. 3.
- ² Olfati-Saber, R., Fax, J., and Murray, R., “Consensus and Cooperation in Networked Multi-Agent Systems,” *Proceedings of the IEEE*, Vol. 95, No. 1, Jan. 2007, pp. 215–233.
- ³ Borkar, P. and Varaiya, P., “Adaptive Control of Markov Chains, I: Finite Parameter Set.” *IEEE Trans. on Automatic Control*, Vol. AC-24, No. 6, 1979.
- ⁴ Buckley, J. and Eslami, E., “Fuzzy Markov Chains: Uncertain Probabilities.” *Mathware and*

- Soft Computing*, Vol. 9, 2002, pp. 33–41.
- ⁵ Dearden, R., Friedman, N., and Andre, D., “Model Based Bayesian Exploration,” *Conf on Uncertainty in AI*, 1999.
 - ⁶ Ford, J. and Moore, J., “Adaptive Estimation of HMM Transition Probabilities.” *IEEE Transactions on Signal Processing*, Vol. 46, No. 5, 1998.
 - ⁷ Bagnell, A., Ng, A., and Schneider, J., “Solving Uncertain Markov Decision Processes.” *NIPS*, 2001.
 - ⁸ Iyengar, G., “Robust Dynamic Programming,” *Math. Oper. Res.*, Vol. 30, No. 2, 2005, pp. 257–280.
 - ⁹ Nilim, A. and Ghaoui, L. E., “Robust Solutions to Markov Decision Problems with Uncertain Transition Matrices,” *Operations Research*, Vol. 53, No. 5, 2005.
 - ¹⁰ Bertuccelli, L. F., *Robust Decision-Making with Model Uncertainty in Aerospace Systems.*, Ph.D. thesis, MIT, 2008.
 - ¹¹ Castañón, D. and Teneketzis, D., “Further results on the consensus problem,” *IEEE Conference on Decision and Control*, 1987.
 - ¹² Castañón, D. and Teneketzis, D., “Further results on the asymptotic agreement problem,” *IEEE Transactions on Automatic Control*, Vol. 33, 1988, pp. 515–523.
 - ¹³ DeGroot, M., “Reaching a Consensus,” *Journal of the American Statistical Association*, Vol. 69, 1974, pp. 118–121.
 - ¹⁴ Fortunato, S., “The Krause-Hegselmann Consensus Model with Discrete Opinions,” *International Journal of Modern Physics: C*, Vol. 15, 2004.
 - ¹⁵ Genest, C. and Zidek, J., “Combining Probability Distributions: A Critique and an Annotated Bibliography,” *Statistical Science*, Vol. 1, 1986, pp. 114–148.
 - ¹⁶ Josang, A., “The Consensus Operator for Combining Beliefs,” *Artificial Intelligence*, Vol. 141, 2002, pp. 157–170.
 - ¹⁷ Pennock, D. and Welmman, M., “Graphical Representations of Consensus Belief,” *Proc of Uncertainty in Artificial Intelligence*, 1999.
 - ¹⁸ Teneketzis, D. and Varaiya, P., “Consensus in Distributed Estimation with Inconsistent Beliefs,” *Systems and Controls Letters*, Vol. 4, 1984, pp. 217–221.
 - ¹⁹ Winkler, R., “The Consensus of Subjective Probability Distributions,” *Management Science*, Vol. 15, 1968, pp. 61–75.
 - ²⁰ Winkler, R., “Combining Probability Distributions from Dependent Information Sources,” *Management Science*, Vol. 27, 1981, pp. 479–488.
 - ²¹ Washburn, R. and Teneketzis, D., “Asymptotic Agreement among Communicating Decision-makers,” *Stochastics*, Vol. 13, 1984, pp. 103–129.
 - ²² Pearl, J., “Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference,” *Morgan-Kaufman*, 1988.
 - ²³ Beard, R. and Stepanyan, V., “Information consensus in distributed multiple vehicle coordinated control,” *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, Vol. 2, Dec. 2003, pp. 2029–2034 Vol.2.
 - ²⁴ Olfati-Saber, R., Franco, E., Frazzoli, E., and Shamma, J., “Belief Consensus and Distributed Hypothesis Testing in Sensor Networks,” *Network Embedded Sensing and Control*, edited by P. Antsaklis and P. Tabuada, Vol. 331 of *Lecture Notes in Computer Science*, Springer-Verlag, 2006, pp. 169–182.
 - ²⁵ Spanos, D. P., Olfati-Saber, R., and Murray, R. M., “Approximate distributed Kalman filtering in sensor networks with quantifiable performance,” *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, IEEE Press, Piscataway, NJ,

USA, 2005, p. 18.

- ²⁶ Ren, W., Beard, R. W., and Kingston, D. B., “Multi-agent Kalman consensus with relative uncertainty,” *American Controls Conference*, 2005, pp. 1865–1870 vol. 3.
- ²⁷ Alighanbari, M. and How, J. P., “Unbiased Kalman Consensus Algorithm,” *Journal of Aerospace Computing Information and Control*, Vol. 5, No. 9, 2008, pp. 209–311.
- ²⁸ Strens, M., “A Bayesian Framework for Reinforcement Learning,” *ICML*, 2000.
- ²⁹ Gelman, A., Carlin, J., Stern, H., and Rubin, D., *Bayesian Data Analysis.*, Chapman and Hall, 1995.
- ³⁰ Fraser, C. S. R., *Reaching Consensus with Uncertainty on a Network*, Master’s thesis, Massachusetts Institute of Technology, 2009.
- ³¹ Bertuccelli, L. F., *Robust Decision Making with Model Uncertainty in Aerospace Systems*, Ph.D. thesis, Massachusetts Institute of Technology, 2008.
- ³² Grime, S., Durrant-Whyte, H., and Ho, P., “Communication in decentralized data-fusion systems,” *American Control Conference 1992, Proceedings of the 1992*, Vol. 4, June 1992.