Intelligent Information Sharing for Localized, Non-Stationary Phenomena

Adam Eck and Leen-Kiat Soh
University of Nebraska-Lincoln,
Department of Computer Science & Engineering,
256 Avery Hall Lincoln, NE, USA 68588
{aeck,lksoh}@cse.unl.edu

Abstract. Information sharing is important in agent-based sensing, especially when only a small subset of a large team of agents can directly observe local environment phenomena. Moreover, sharing is further complicated in non-stationary environments, where changes in the phenomena over time require the team to collectively revise their beliefs as the phenomena change. In this paper, we first analytically and empirically demonstrate the difficulty inherent in sharing information and revising beliefs over time about localized, non-stationary phenomena, uncovering the inertia-based Institutional Memory Problem. Subsequently, we propose two solutions for addressing this problem: 1) a change detection and response algorithm, and 2) a forgetting-based solution. We test our solutions under several network structures to verify the efficacy of our approaches and evaluate their robustness in the presence of faulty and/or malicious agents injecting incorrect information into the team.

Keywords: Information sharing; Localized phenomena; Non-stationarity; Change detection; Forgetting

1 Introduction

Real-world environments contain complex phenomena that are increasingly observed by computational devices and systems, often to enhance human knowledge and/or provide real-time support for some task. For example, sensors networks and robot teams provide area surveillance (e.g., [8, 9, 12]), autonomous robots discover victims of disasters in search and rescue applications (e.g., [2]), and human relationships and preferences are tracked in social networking (e.g., [16]).

In many of these environments, the observed phenomena are very localized, such as detected events (e.g., fires) in a specific area, victims trapped in particular buildings, or individual user’s preferences. Although there might be many sensing units within the system, only a few sensing units are capable of directly observing such local phenomena, limiting the ability of the system to gather information en mass. Furthermore, the phenomena are also often non-stationary and change dynamically over time. Thus, information gathered by sensing units becomes outdated and must be revised frequently to adapt with the changing phenomena.
Intelligent Information Sharing for Localized, Non-Stationary Phenomena

To address these challenging phenomena properties and improve the quality and use of gathered information, intelligent software and hardware agents can be employed to control sensing units. Such intelligent agents are capable of exhibiting social behavior by sharing information with one another, helping overcome the localization problem in real-world applications. Agents can also provide both goal-directed behavior to accomplish system goals, as well as reactive behavior to adapt system performance in unexpected situations [15]. In this manner, intelligent agents can reason about the sensing performed by the system in order to optimize or improve the information gathered (e.g., [8, 12]) and can improve the robustness, scalability, effectiveness, and efficiency of observational systems.

Prior research has studied both (1) information sharing between cooperative agents (e.g., [4, 5, 10]), and (2) detecting and adapting to non-stationary information gathered by individual agents (e.g., [14]). However, little work has considered these two components of agent-based sensing in combination. Both are vital to sensing localized, non-stationary phenomena in real-world environments; however, localization and non-stationarity together make both sharing and change detection more difficult. Thus, it is important to study both components together to understand their relationship to the two phenomena properties.

In this paper, we begin to fill this gap in the literature by considering the impact of both localization and non-stationarity in observed phenomena on information sharing and change detection within teams of sensing agents. We start with a known model for information sharing: large team information sharing (LTIS) [4, 5, 10], a formalized model where many agents work together but only a small subset of the agents can directly observe any particular phenomena. This model was chosen as a starting point due its ability to handle the localization property and its growing popularity in the agent literature. To this model, we then add non-stationarity and study the effects of these challenging properties together to develop new solutions for handling both properties simultaneously.

We contribute (1) a formalization of non-stationarity within the LTIS model, alongside localization; (2) an analysis of the difficulty of non-stationarity during belief updates using information shared by the few local agents capable of observing the phenomena; (3) two distinct solutions for overcoming the challenges of non-stationarity and localization without requiring explicit knowledge of environment dynamics: (i) cooperative change detection and response in local neighborhoods, and (ii) individually forgetting outdated information; (4) empirical studies investigating the impact of localized, non-stationary phenomena on large teams of agents controlling sensing units, as well as the effect of our solutions on adapting to such phenomena; and (5) a discussion of the strengths and weaknesses of our solutions and their appropriateness in different environments.

2 LTIS

2.1 LTIS Model

We first present the formalized LTIS model [4, 5, 10] serving as the foundation for our problem formulation. In LTIS, a large set of agents $A$ work together as a
team to collect information about some environment phenomena. However, only a small subset \( S \subset A \) (with \( |S| << |A| \)) of the agents have sensors that can directly observe a phenomenon. For simplicity, agents represent a phenomenon as a binary fact \( F \in \{ True, False \} \), although the model can be easily extended to a greater number of values [10]. Each sensor returns binary observations \( ob \) describing the current value of the phenomenon. The sensors are imperfect and only return correct \( ob \) with accuracy probability \( r \). For agents with sensors, these observations are used to revise the agent’s belief about the correct value of \( F \). However, since the team has limited sensors that can observe the particular phenomenon, the agents must share information to revise the other agents’ beliefs. Because the team is so large, agents can only communicate with nearby neighbors. Each neighborhood is relatively very small, with average size \( d << |A| \).

A common set of solution techniques have been adopted for LTIS \([4, 5, 10]\). First, agents only communicate summarized information representing their current belief about \( F \), instead of forwarding each individual observation from the sensors. These summarized beliefs are called opinions (denoted by \( op \), described below). This practice (1) reduces the amount of potentially costly communication, (2) minimizes the impact of over-counting information, since each agent could repeatedly receive the same forwarded observation from multiple neighbors, and (3) hides raw observations which could be sensitive or include private information (e.g., enemies in the surveilled area, user purchasing habits) \([4]\).

Given uncertain facts, beliefs are represented by a probability distribution describing the likelihood that \( F \) is either \( True \) or \( False \). Agents start with an initial uncertain belief that any value is equally likely, then Bayesian updating incorporates new information \( o \) (an observation \( ob \) from a sensor, or an opinion \( op \) from a neighbor):

\[
b' = \frac{cp(o) \cdot b}{cp(o) \cdot b + (1 - cp(o)) \cdot (1 - b)}
\]

where \( b \) is the probability that \( F \) is \( True \) (so \( (1 - b) \) is the probability it is \( False \)), \( b' \) is the updated belief, and \( cp(o) \) is the conditional probability that \( F \) is \( True \) given the new information. Here, \( cp \) weights newly received information \( o \), and its value depends on the value and source of \( o \):

\[
cp(o) = \begin{cases} 
  r & \text{if } o = True \land o \text{ an observation } ob \\
  1 - r & \text{if } o = False \land o \text{ an observation } ob \\
  mj & \text{if } o = True \land o \text{ an opinion } op \\
  1 - mj & \text{if } o = False \land o \text{ an opinion } op
\end{cases}
\]

For observations \( ob \), \( cp \) depends on sensor accuracy \( r \), whereas for opinions \( op \), \( cp \) depends on \( mj \), the likelihood that \( a_j \)'s neighbors share correct opinions.

Because beliefs are uncertain, agents only share information when they become confident that \( F \) is either \( True \) or not from received information. In particular, a confidence threshold \( \sigma > 0.5 \) discretizes beliefs into confident opinions:
Intelligent Information Sharing for Localized, Non-Stationary Phenomena

\[
op = \begin{cases} 
  \text{True} & \text{if } b > \sigma \\
  \text{False} & \text{if } b < 1 - \sigma \\
  \text{Unc} & \text{else}
\end{cases}
\]  

(3)

where Unc denotes an unconfident opinion that is never communicated but internally noted by the agent. Fig. 1 in Section 3 illustrates this discretization.

2.2 Prior LTIS Research

Prior LTIS research has primarily focused on two aspects: (1) identifying important emergent behaviors during information sharing within large teams, and (2) developing distributed algorithms to achieve desired emergent behavior.

Using branching process theory, Glinton et al. [4] developed an analytical model predicting that different settings of the \(cp\) information weighting parameter (specifically \(m_j\) for weighting opinions from neighbors) can result in three phases of emergent behavior: (1) unstable dynamics, where too much weight causes frequent avalanches of sharing between agents, resulting in oscillating beliefs, (2) stable dynamics, where too little weight results in infrequent belief updates and few confident beliefs, and (3) scale invariant dynamics, where the optimal amount of weight permits enough sharing to propagate beliefs throughout the team without causing oscillation. Later, they [5] discovered that LTIS was vulnerable when incorrect information was received (either from benign error or malicious injection by an attacker) and an agent’s belief was near the confidence threshold \(\sigma\).

Prior research has also focused on developing distributed algorithms for controlling information sharing by adapting the weight (i.e., \(m_j\)) placed in shared opinions in order to achieve desirable properties. Glinton et al. [4] exploited their model to produce an algorithm (DACOR) that controls avalanches within an agent’s local neighborhood to globally achieve scale invariant dynamics. Later, Pryymak et al. [10] developed an algorithm (AAT) requiring no additional communication to improve belief convergence.

In this paper, we contribute to both avenues of research on LTIS. First, we study the emergent behavior caused by including non-stationarity in the LTIS model, through which we describe analytically the impact of this property on agent information sharing. Second, we develop distributed solutions for adapting information sharing and belief updates to handle non-stationarity. We also evaluate these solutions empirically using different settings of teams likely to occur in real-world applications (e.g., different network structures connecting agents, and the presence of malicious or faulty agents as previously studied [5]) to demonstrate the advantages and disadvantages of each approach.

3 Non-stationary Phenomena

As described previously, the LTIS model is useful for addressing the challenging localization property in observed environmental phenomena because it explicitly
Intelligent Information Sharing for Localized, Non-Stationary Phenomena

considers the reality that only a small subset of the agents can make direct observations. In this section, we extend the LTIS model to also include a second important property of many observed phenomena: non-stationarity.

Recall that non-stationarity is caused by dynamic environments that result in changes to the phenomena of interest as agents perform observations (e.g., events occurring in areas of interest, additional buildings collapsing after a disaster trapping new victims, changing human user preferences). In such environments, agents must not only be capable of determining the initial value of a phenomenon (equivalent to forming beliefs about stationary phenomena in static environments as previously studied with LTIS), but agents must also be capable of properly adapting their beliefs over time as a phenomenon changes values.

3.1 Modeling Non-stationarity in LTIS

To model non-stationary phenomena in LTIS, we extend the existing model by adding a time component to the relevant factors in order to reflect changes to the phenomena over time. This approach produces the following changes.

First, we discretize time into different intervals, represented by \( t \in \mathbb{Z}^+ \). One time interval represents the amount of time required for a sensor to produce an observation and for an agent to transmit an opinion to one of its neighbors. Second, we redefine a fact from a constant \( F \) to a time-dependent sequence \( F(t) \) expressing the phenomenon’s changing value at each elapsed time interval. For example, a fact might be (1) periodic and switch values every \( \Delta t \) ticks, (2) random and switch values with differing durations, or (3) simply switch values once. Third, observations \( ob \) and opinions \( op \) are time-stamped with the time \( t \) when they were observed or shared. Finally, to reflect changing fact values over time in the agents’ beliefs, probabilistic beliefs are also extended to time-dependent sequences \( b(t) \). Of note: since an agent can receive one or more opinions from its neighbors and also an observation from a sensor in the same time interval \( t \), a chain of several belief updates \( b' \) can occur for \( b(t) \). Thus, the agent might need to incorporate multiple updates from different sources in the same time interval.

3.2 Analyzing the Effect of Non-stationarity

Forming consistent, accurate beliefs about non-stationary phenomena is a much more challenging problem than observing stationary phenomena because of the amount of information required to correctly revise agents’ beliefs after a phenomenon change. To illustrate (without loss of generality), consider a simple phenomenon \( F_1(t) \) that is initially \( True \), then changes to \( False \) at \( t = 1001 \). Observing this phenomenon results in updates to an agent’s beliefs over time illustrated in Fig. 1 as (a) a continuous probability \( 0 \leq b \leq 1 \), and (b) a discrete opinion \( op \in \{False, Unc, True\} \) (Eq. 3).

Here, the agent begins with pure uncertainty \( b(0) = 0.5 \) and must update its belief to \( b(t) \geq \sigma \) (recall \( \sigma > 0.5 \)) to achieve a correct opinion of \( True \). This requires a belief change of only \( \Delta b_1 = \sigma - 0.5 \), denoted by (*) in Fig. 1.
After the non-stationary phenomenon changes values, the agent must receive a sequence of new information to revise its beliefs from \( b(t) \geq \sigma > 0.5 \) to a later \( b(t') \leq 1 - \sigma < 0.5 \). This requires a belief change of \( \Delta b_2 \geq 2(\sigma - 0.5) \), denoted by (**). Since \( 2\Delta b_1 = 2(\sigma - 0.5) \leq \Delta b_2 \), we find that properly revising beliefs for non-stationary phenomena requires at least twice as much belief change as observing stationary phenomena, and subsequently, twice as much observed and shared information. This requirement holds for any change in a phenomenon value, not just in the example used here.

Unfortunately, choosing a weight placed in shared information cannot overcome this problem, as used previously to control the flow of information through the team to achieve consistent, accurate beliefs [4, 10]. Instead, the above problem arises regardless of the weight selected. That is, given the belief update rule (Eq. 1) and any chosen value for \( cp(o) \), two updates with opposing information simply cancel each other out. This explains why an agent needs twice as much information to revise its belief (than it takes to arrive at an initial confident belief). This result implies that controlling information sharing by selecting a weight for new information (namely \( m_j \) for shared opinions \( op \)) as studied previously for LTIS does not address the challenges posed by non-stationarity. Instead, a different type of solution for guiding agent information sharing and belief updates is necessary. We propose two such solutions in Sections 4 and 5 that exploit different ways of closing the gap between (*) and (**) (from Fig. 1) in order to speed up belief convergence after a change in the non-stationary phenomenon.

Furthermore, we note that the distances (*) and (**) (in Fig. 1) also result in agents being less likely to share opinions from each belief update after the phenomenon has changed values than they would with stationary phenomenon. Here, the team suffers from an inertia problem, which we call the:

**Institutional Memory Problem**: too much information needs to be received by an agent to cause the agent to also share new opinions, resulting in the team becoming stuck with outdated beliefs that do not change even when new information is observed.

Specifically, recall that agents only share information with neighbors when they cross a confidence threshold \( b' \geq \sigma \) or \( b' \leq 1 - \sigma \). Since more updates are required to reach a threshold after a phenomenon value change, each individual
Intelligent Information Sharing for Localized, Non-Stationary Phenomena

belief update is less likely to result in sharing a new opinion. Therefore, agents actually share fewer opinions with one another. Unfortunately, this is opposite of what the agents need in order to adapt to non-stationarity since they actually need more updates to reach a new accurate belief, causing agents to fail to adapt and either become stuck with (1) outdated beliefs or (2) uncertainty.

The Institutional Memory Problem should not to be confused with the stable dynamics emergent behavior discovered by Glinton et al. [4]. In their work studying stationary environments, insufficient information is exchanged due to too little weight placed on new information, resulting in uncertain beliefs. In our work, an inability to overcome previous confident beliefs limits information exchange. To demonstrate this distinction, Fig. 2 presents the results of an empirical study using a team of agents observing the aforementioned simple phenomenon $F_1(t)$ (using the Random Network parameters given in Section 6), where we varied the weight for new information from neighbors. Although the team could converge to consistent, accurate beliefs for the initial value of the non-stationary phenomenon (identical to stationary phenomena), a much smaller number of agents correctly revised their beliefs over time. Indeed, the majority of agents were unable to overcome inertia and simply retained the initial phenomenon value in their beliefs. Since appropriately choosing a weight for new information is thus not a viable solution for handling non-stationarity (as previously studied for stationarity), we instead require a new type of solution.

Overall, we observe the following about the relationship between the two phenomena properties. First, localization magnifies the impacts of non-stationarity by limiting the flow of information into the team by restricting observations about the changing phenomena necessary to update beliefs over time. Second, non-stationarity magnifies the impacts of localization by limiting the flow of information within the team by restricting shared opinions also necessary to update beliefs. Thus, these two challenging properties work together adversely.

4 Change Detection and Response

Similar to prior algorithms for LTIS, our first solution relies on cooperative agents making simple yet effective local decisions within neighborhoods to achieve
Intelligent Information Sharing for Localized, Non-Stationary Phenomena

desired emergent behavior (i.e., properly adapting agent beliefs over time). Here, we develop an approach for explicitly detecting and responding to non-stationarity.

**Strategy.** Our strategy is to convert the problem of handling non-stationarity to one closer to forming beliefs about (simpler) stationary phenomena. We start with the insight that if the team were able to detect when a phenomenon changes values, then the agents could treat a new value independent of the previous value – that is, as a separate stationary phenomenon and a separate instance of the original stationary LTIS problem. Then, each agent would need less information to revise its beliefs after a phenomenon change, having instead only to change beliefs from pure uncertainty to a new confident belief \((\Delta b_1)\), as opposed to moving from one confident belief to its opposite \((\Delta b_2 \geq 2\Delta b_1)\). In turn, this behavior would mitigate the Institutional Memory Problem by reducing inertia and subsequently increase the team’s convergence to consistent, accurate beliefs.

To detect changes to a non-stationary phenomenon, we actually exploit the cause of the inertia property of the Institutional Memory Problem identified in the previous section. Specifically, considering how much information is needed to revise an agent’s belief (i.e., \(\Delta b_2 \geq 2\Delta b_1\), illustrated by (***) in Fig. 1) causing the inertia, we note that any particular neighbor is very unlikely to share a new opinion that conflicts with its most recently shared opinion without an actual change in the phenomenon. For instance, in our prior example (Fig. 1), sharing a new False opinion (after previously sharing True) indicates to an agent’s neighbors that it received much new information reflecting a phenomenon change. In which case, the new opinion is highly likely to be accurate since the likelihood of receiving such a large chain of information that is instead incorrect would be small. Therefore, changed opinions by neighbors provide more information than just new opinions, but also indicators signaling that the phenomenon indeed likely changed values, which other agents can exploit to overcome their inertia.

After detecting a phenomenon change by receiving a newly conflicting opinion from a neighbor, an agent responds as follows (detailed in Algorithm 1). First, the receiving agent resets its own belief to pure uncertainty \((b = 0.5)\), starting a new, fresh belief about the phenomenon under observation. Thus, this agent is now closer to a new correct opinion than any formerly confident belief about the previous value of the phenomenon, without having had to receive as much information as its neighbor. Next, the receiving agent broadcasts its detection (i.e., `sendDetectedChangeAlert()`) to its other neighbors that are farther away from sensors and thus less likely to have already detected a change as information propagates, encouraging them to also reset their beliefs. Afterwards, it updates its belief using the information in the shared opinion (Eq. 1).

This reaction behavior simultaneously (1) enables agents to quickly revise beliefs after a detected change by moving away from previously confident beliefs before updating, and (2) spreads change detection locally to speed up belief convergence without requiring every agent to receive a large chain of information.

**Addressing Concerns.** However, agents must be careful to avoid incorrectly detecting phenomenon changes, or else their beliefs could oscillate (similar to unstable team dynamics [4]). That is, if a neighbor shares an incorrect new
Intelligent Information Sharing for Localized, Non-Stationary Phenomena

detectAndRespond(op)
if op.value ≠ lastOpinion(op.sender).value then
    rand ∼ U(0, 1)
    if rand ≤ σ then
        b ← 0.5
    end
    sendDetectedChangeAlert()
end
updateBelief(op)

Algorithm 1: Change Detection and Response (CD & R) Algorithm

opinion when the phenomenon hasn’t changed, then a false change would be detected and agents would unnecessarily reset their beliefs away from correctness.

Our solution mitigates this concern in three targeted ways. First, agents only reset their beliefs with likelihood σ, reflecting the same uncertainty the sharing neighbor has in its opinion (Eq. 3). Second, our solution only locally reacts within two network hops from the agent that initially changed opinions, minimizing the impact of false detection on the entire team. Recall that the team’s average connectivity d is assumed to be rather small (relative to the size of the team), so these are very local behaviors. Finally, even if an agent incorrectly resets its beliefs, it only changes its opinion to Unc and does not fully adopt the neighbor’s incorrect information. Thus, the agent’s belief is just as close to the correct belief as it is to the neighbor’s shared incorrect belief, and the agent can re-converge to the correct belief with new information just as easily as it would converge to the incorrect belief that triggered the reset in the first place.

5 Forgetting Outdated Beliefs

Our second solution also relies on agents to exhibit local behaviors to adapt their beliefs over time to non-stationary phenomena. However, unlike our first solution, it is even more localized since each agent adapts independently of its neighbors, lessening the reliance of agents on one another. Specifically, we develop a solution employing belief decay to enable agents to forget outdated beliefs and independently and quickly adapt to changes to non-stationary phenomena.

The goals behind this solution are that it should (1) produce faster adaptation to non-stationarity since agents do not need to wait for neighbor’s conflicting opinions to begin adaptation, and (2) be more robust in the presence of faulty or malicious agents [5] since it doesn’t rely on neighbors for change detection.

Strategy. This solution is based on the natural assumption that if an agent has not received information for a while, its beliefs are less likely to reflect the current value of non-stationary phenomena since each phenomenon’s value changes over time. Thus, the agent’s beliefs should become less confident the

---

1 Detection is only propagated to the neighbors of the detecting agent, which is itself a neighbor of the changed agent.
Intelligent Information Sharing for Localized, Non-Stationary Phenomena

longer time has elapsed since the agent last received new information and updated its beliefs. Then, the agent would be more likely to (1) reach a different confidence threshold in order to form a new correct belief, and (2) propagate new opinions throughout the team, enabling other agents to also correctly revise their beliefs and avoid inertia and the Institutional Memory Problem.

To appropriately adapt agent uncertainty over time, we propose a solution based on belief decay, where each agent forgets older beliefs the longer time passes between belief updates. Belief decay has been previously used to describe the behavior of human knowledge and memory in the cognitive science literature (e.g., [7]), as well as for related problems in artificial intelligence, such as situational awareness (e.g., [6]) and information foraging with fewer agents that each directly observe the environment (e.g., [11]). However, while this approach has been used in other domains, this paper is the first application of belief decay to information gathering problems with localized phenomena such as LTIS, so its benefits are unclear a priori. We expect that such an approach is especially strategic for LTIS because each agent (1) adjusts its beliefs independent of its neighbors, reducing the agent’s reliance on its neighbors to adapt to changes, and (2) can control the rate of decay, useful for adapting to various frequencies of change in non-stationary phenomena.

For this solution, we propose adding the following rule to each belief update, to be executed before incorporating the new information using Eq. 1:

\[
b'(t) = 0.5 + (b(t) - 0.5)\lambda^\delta
\]  

where \(\delta\) measures the time elapsed since the agent’s last belief update and \(\lambda \in (0, 1)\) is a parameter controlling how quickly the agent’s belief decays over time: smaller \(\lambda\) causes faster decay, whereas larger \(\lambda\) causes slower changing beliefs. Thus, by choosing an appropriate \(\lambda\), an agent can adjust how quickly it forgets old information and reacts to phenomenon changes (unlike our first solution).

Using Eq. 4, an agent’s belief always decays towards pure uncertainty \((b = 0.5)\), and the amount of decay is proportional to the amount of time elapsed since its last belief update. Thus, the agent moves towards the best position to form a new belief after a phenomenon value change, and it requires less evidence of change (avoiding inertia) the longer since an update when it is more likely that the phenomenon indeed changed values. Afterwards, performing updates with Eq. 1 incorporates new information into the time-adjusted belief, potentially crossing a confidence threshold so the agent can share a new opinion.

Another way of looking at Eq. 4 is time-dependent information weighting. That is, Eq. 4 weights older information (already in the belief) down towards uncertainty before incorporating new information (Eq. 1), and the down-weighting is proportional to the time elapsed since the older information was received.

**Addressing Concerns.** However, we want to ensure that belief decay does not cause agents to become uncertain if the phenomenon has not actually changed for a while, which would lead the team to fail to maintain accurate beliefs. We propose only decaying beliefs when new information is received instead of every tick. Recall that most agents infrequently receive information: only when new
Intelligent Information Sharing for Localized, Non-Stationary Phenomena

Fig. 3. Example of Performing Belief Decay (a) Only upon Receipt of Information vs. (b) Every Tick. Note: Shaded area above \( \sigma \) line indicates accumulated time with a confident belief, which is much greater for (a) than (b).

information is available. Delaying belief decay until receipt of new information allows the agent to (cautiously) retain its prior beliefs when it has no evidence that the phenomenon has changed. Decaying every tick (even with a smaller decay rate) would instead constantly push agents towards uncertainty, even if the phenomenon has not changed, as illustrated in Fig. 3. Thus, agents would spend more time with uncertain beliefs, making it difficult to maintain confident beliefs, similar to the stable dynamics problem noted by Glinton et al. [4] where too little weight in new information causes the team to be uncertain over time.

6 Experimental Setup

We conducted an empirical study to evaluate the performance of our solutions in order to better understand how our solutions address the challenges posed by localized, non-stationary phenomena. As a baseline, we compared against agents that know a priori the ideal weight for new information, finding which is the goal of prior algorithms for stationary phenomena (e.g., DACOR [4] and AAT [10]). Thus, our baseline represented an upper-bound on prior algorithm performance. The goals of our study were to (1) determine whether our algorithms improve the ability of the team to converge to consistent, accurate beliefs about localized, non-stationary phenomena, and (2) evaluate the robustness of our algorithms in the presence of malicious and/or faulty agents that share incorrect information.

Within each goal, we also considered how the network structure of the team (dependent on the application) impacts performance. Specifically, we tested three different types of networks present in real-world applications of multiagent sensing, including: (1) Random networks (RN), where connections between agents are randomly determined, such as in ad hoc sensor networks, (2) Small world networks (SWN), where agents are clustered in large, important subgroups, such as surveillance applications, and (3) Scale-free networks (SFN), where connectivity is power-law distributed (i.e., a few agents have many neighbors, whereas many agents have small connectivity), common to internet applications.

To create these networks, we used the Erdos-Renyi [3], Watts-Strogatz (rewire \( p = 0.5 \)) [13], and Barabasi-Albert preferential attachment [1] models, respectively. For each network, we used the standard setting from prior studies (e.g., [4, 10]): the number of agents \(|A| = 1000\), the number of sensors \(|S| = 0.05|A| = 50\),
sensor accuracy $r = 0.55$, average neighborhood size $d = 8$, confidence threshold $\sigma = 0.8$, and (unless specified) optimal $m_j = 0.63 \forall a_j \in A$ [4].

Within each of these networks, we also varied the presence of faulty and/or malicious agents in order to evaluate the robustness of our solutions (which has been demonstrated to be a concern even for stationary phenomena [5]). These bad agents share incorrect opinions every time they reach a new confident opinion and were intentionally chosen as the highest connectivity agents, which represents a worst case scenario since these agents have most influence.

For non-stationarity, we used random sequences alternating values 10 times with random lengths (chosen uniformly such that the total length is 10,000 ticks). These sequences represent non-regular phenomena placing varying stress on the agent network, mimicking the difficulty likely present in real-world applications.

To evaluate our solutions, we use two measures of agent performance. First, we consider the average number of phenomena values about which the team collectively forms correct beliefs, represented by $N_{800}$. That is, following tradition (e.g., [4]), we consider a team’s belief correct if $80\%$ of the agents ($0.8|A| = 800$ in our experiments) hold a correct, confident belief at the same time before the phenomenon changes values again. This measures how well the team as a whole accomplishes its goal. Second, we also consider the average number of agents holding each of the three types of discrete beliefs: correct ($C$) and incorrect ($I$) confident beliefs and unconfident beliefs ($U$). This measure further illuminates how the individual beliefs held by agents change over time during adaptation.

### 7 Results

**Performance.** We first evaluate the general performance of our two solutions (without faulty/malicious agents). Table 1 reports the measures of team performance ($N_{800}, C, I, U$) for each of the three network types. Please note that these results represent the best performance of each solution: using the ideal $\lambda$ for our forgetting-based solution (found by varying $\lambda \in [0.9, 1.0]$ in 0.01 increments) and the ideal $m_j$ for the baseline and CD & R solutions (found by varying $m_j \in [0.5, 1.0]$ first in 0.05 increments, then in 0.01 increments around the ideal value). These ideal settings are also provided in Table 1.

We first observe that in all network types, both of our solutions formed significantly more correct beliefs about the different number of phenomena values ($N_{800}$). As expected due to the Institutional Memory Problem, agents using the baseline approach only quickly converged to the first value of the phenomenon ($True$), then maintained that belief regardless of new information received. As a result, the baseline approach only resulted in correct beliefs about half of the phenomenon values (since half were equal to the initial value). In contrast, both of our solutions successfully adapted agent beliefs over time after phenomenon changes, resulting in many more correct collective beliefs (with $N_{800}$ closer to the maximum 10), as well as superior numbers of individually correct ($C$) and incorrect ($I$) agents. Therefore, both of our solutions are improvements over the previously successful LTIS approaches when considered in non-stationary
Intelligent Information Sharing for Localized, Non-Stationary Phenomena

Table 1. Solution Comparison (with 95% Confidence Intervals)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$N_{800}$</th>
<th>$C$</th>
<th>$I$</th>
<th>$U$</th>
<th>$m_j$</th>
<th>$\lambda$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>4.97 ± 0.06</td>
<td>5 ± 0.00</td>
<td>4.99 ± 0.02</td>
<td>0.68</td>
<td>0.97</td>
<td>0.95</td>
</tr>
<tr>
<td>CD &amp; R</td>
<td>7.62 ± 0.28</td>
<td>7.60 ± 0.27</td>
<td>7.40 ± 0.29</td>
<td>0.63</td>
<td>0.67</td>
<td>0.67</td>
</tr>
<tr>
<td>Forget</td>
<td>9.38 ± 0.13</td>
<td>9.24 ± 0.15</td>
<td>9.44 ± 0.13</td>
<td>0.68</td>
<td>0.97</td>
<td>0.95</td>
</tr>
</tbody>
</table>

environments. Moreover, since this result held for each of the network types, we conclude that our solutions behave equally well in a wide range of settings.

Comparing our two solutions, we observe that the forgetting solution significantly outperformed the CD & R algorithm. We suspect this is due to the agents’ ability to adapt to changes independently by time-decaying beliefs without having to wait for a neighbor to signal a change. That is, the ideal forgetting rate allowed the agents to move towards uncertainty faster after a phenomenon change, indicated by more unconfident agents $U$, thereby overcoming inertia faster. The forgetting solution also achieved a much greater number of correct belief agents ($C$), indicating that it resulted in better individual beliefs, too.

However, the performance of the forgetting solution was highly dependent on the $\lambda$ value used. Although we do not have space to report the full results of varying $\lambda$, we observed a sharp decline in performance when $\lambda$ was below its ideal value, quickly falling to $N_{800}$ near 0 (due to almost exclusively unconfident agents) with decreases in $\lambda$ of only 0.04. Thus, although our forgetting solution outperformed our CD & R solution, it requires more fine-tuning. Therefore, the forgetting solution would require more consideration if deployed to real-world applications, whereas the CD & R solution requires less foresight. One observable factor in the ideal setting of $\lambda$ is the connectivity of the network: in the SFN network with super-connected agents, a lower $\lambda$ was ideal, allowing the team to more quickly forget what was previously shared by the most connected agents.

Robustness against Malicious/Faulty Agents. Next, we compare our solutions’ performance in the presence of malicious and/or faulty agents propagating incorrect information, with results presented in Fig. 4. As expected, the CD & R algorithm is indeed more susceptible to bad information from malicious and/or faulty agents. On the other hand, the forgetting solution was even more robust than might be expected. Specifically, correct convergence still occurred
Intelligent Information Sharing for Localized, Non-Stationary Phenomena

Fig. 4. Impact of Malicious/Faulty Agents

for many phenomena values \((N_{500} > 8)\) in the RN and SWN as the number of bad agents approached 50. This is significant because 50 is also the number of agents with sensors inputting new information into the system. Therefore, even as the amount of bad information approached the amount of freshly observed information, the forgetting-based solution maintained high performance.

In contrast to our earlier results (Table 1) with no malicious or faulty agents, network structure did impact team performance once bad agents were included. In particular, in the SFN case, team performance quickly declined as the number of malicious/faulty agents increased. Recall that in our experiments, bad agents were deliberately chosen to be the most connected agents that exhibit the greatest influence on the team. In SFN, these agents have greater connectivity than in the RN and SWN, increasing the influence of such malicious/faulty agents and thus degrading team performance. In the future, we intend to study how to improve robustness in the presence of such super-connected agents.

Interestingly, agent performance did not monotonically decrease as the number of malicious/faulty agents increased. Instead, a few agents sharing incorrect information was actually beneficial to overcoming inertia in the Institutional Memory Problem. That is, occasionally receiving incorrect information seemed to cause agents to fail to reach overly confident opinions, and thus less inertia towards new beliefs. This lower inertia enabled the team to achieve more correct beliefs for the shorter duration phenomenon values that were difficult to observe.

8 Conclusions

In conclusion, we addressed information sharing in multiagent systems observing localized, non-stationary phenomena common in emerging computational systems. We analytically predicted the impact of adding non-stationarity to an existing model for information sharing of localized phenomena called LTIS. We discovered the Institutional Memory Problem caused by inertia in the agents’ beliefs, then developed two distributed solutions not requiring explicit knowledge of environment dynamics: (1) a change detection and response algorithm for improving information sharing in local neighborhoods, and (2) a forgetting-based solution for independent adaptation by individual agents. We empirically
evaluated our solutions while varying the type of network structure (representing different real-world scenarios), as well as the number malicious or faulty agents, and discovered that our CD & R algorithm yielded improved performance over prior algorithms for stationary phenomena, whereas our forgetting-based solution achieved even greater performance and robustness to bad information.

In the future, we intend to advance our research by (1) developing analytical models describing agent beliefs under non-stationarity and localization, extending prior models (e.g., [4]), (2) developing an approach to automatically tune the $\lambda$ parameter for our forgetting-based solution, (3) relaxing some assumptions of the LTIS model (e.g., Bayesian updating, equal neighbor weighting), and (4) evaluating in real-world deployments of multiagent information sharing.

References