2013

Non-Stationary Phenomenon in Large Team Information Sharing

Adam Eck
University of Nebraska-Lincoln, aeck@cse.unl.edu

Leen-Kiat Soh
University of Nebraska, lsoh2@unl.edu

Non-Stationary Phenomenon in Large Team Information Sharing

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

Abstract

Information sharing is important in agent-based sensing, especially for large teams where only a small subset of the agents can directly observe the environment. We consider the impact of non-stationarity in the observed phenomenon on the collective beliefs of such large teams. Non-stationarity is challenging because not only must the team converge to consistent, accurate beliefs (as studied previously), but, most importantly, the team must also frequently revise its beliefs over time as the phenomenon changes values. We analytically and empirically demonstrate the difficulty in revising beliefs over time with the standard model and propose two novel solutions for improving belief convergence when observing non-stationary phenomenon: (1) a change detection and response algorithm for cooperative environments, and (2) a forgetting-based solution for non-cooperative environments.

Introduction

Agent-based sensing (ABS) has grown as a popular application class for intelligent agents and multiagent systems (MAS). In ABS, agents enhance sensing systems beyond merely observing the environment to also provide intelligent capabilities, such as building accurate, up-to-date models of the environment as information is collected or intelligently responding to observations in real-time. ABS has found real-world applications in intelligent user support (e.g., Boutilier, 2002), dialog management systems (e.g., Williams and Young, 2007), robotic exploration and navigation (e.g., Spaan et al., 2010), and wireless sensor networks (e.g., An et al., 2011; Padhy et al., 2006).

One interesting subproblem within ABS research is **Large Team Information Sharing** (LTIS) (e.g., Glinton et al., 2009, 2010, 2011; Pryymak et al., 2012). As in human societies where many people work together striving for a common goal, agents might be tasked to work together in large teams for ABS in order to observe and respond to phenomenon that occur in the environment. Within these teams, only a *limited subset* of the agents might be equipped with sensors capable of directly observing the phenomenon of interest, so agents must share information with one another in order to converge to consistent, accurate beliefs throughout the team. Potential real-world application of LTIS include: (1) a diverse team of search and rescue agents responding to a natural disaster, or (2) a military intelligence network collecting information.

Within prior literature studying LTIS (e.g., Glinton et al., 2009, 2010, 2011; Pryymak et al., 2012), one initial assumption has been that the phenomenon observed by the agents is *stationary*. That is, the phenomenon does not change as the agents perform observations and share information. This assumption simplified the problem for its initial study and enabled researchers to produce a foundation for (1) modeling the dynamics of information sharing within LTIS and (2) developing distributed solutions achieving desired emergent behavior throughout the team. Moreover, such an assumption is perfectly valid in scenarios where the phenomenon being observed either does not change or changes slowly enough that the team will never notice an environment change while performing LTIS.

However, in many potential real-world applications of LTIS, the stationarity assumption is violated and instead, the phenomenon of interest observed by agents changes as the agents observe and form beliefs. For instance, the presence of victims after an earthquake might change as new buildings collapse over time. This non-stationarity property makes ABS more challenging, and failing to account for non-stationarity could cause the team to: (1) fail to adapt outdated beliefs as the phenomenon changes over time, or (2) remain consistently uncertain and unable to successfully perform ABS. Handling non-stationarity is also more challenging in LTIS than in other ABS settings (e.g., information foraging) since only some of the agents directly observe the changing phenomenon, making it harder to detect changes and adapt beliefs over time. Indeed, we will demonstrate the challenge posed to traditional approaches to LTIS due to non-stationarity.

To handle non-stationarity within LTIS, we propose two solutions relying on different levels of interaction between agents. First, we develop a distributed, localized approach for detecting and appropriately responding to changes in the non-stationary phenomenon based on information shared between cooperative agents. However, this algorithm could be vulnerable to malicious agents in a non-

Department of Computer Science & Engineering, University of Nebraska-Lincoln,
Technical Report, TR-UNL-CSE-2013-0001

cooperative setting, so we also develop a forgetting-based solution where each agent works more independently to adapt its beliefs over time as new information is received. Using a standard LTIS setting in an empirical study, we demonstrate that both solutions enable the agents to properly revise their beliefs over time as the non-stationary phenomenon changes values. We also describe the relative strengths and weaknesses of each solution.

Large Team Information Sharing Model

We begin by presenting the formalized LTIS problem model (Glinton et al., 2009, 2010, 2011; Pryymak et al., 2012). In LTIS, a large set of agents A (e.g., $|A| \geq 1000$) work together as a team to collect information about some phenomenon in the environment. However, only a small subset $S \subset A$ (with $|S| \ll |A|$) of the agents have sensors with which to *directly* observe the phenomenon. For simplicity, agents represent the phenomenon with a binary fact F that only takes values from $\{True, False\}$, although the model can be easily extended to a greater number of values (Pryymak et al., 2012). Each sensor returns binary observations o describing the current value of the phenomenon. The sensors are imperfect and only return correct observations 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, the agents must share information to revise the non-sensor agent's beliefs. Because the team is so large, agents can only communicate with nearby neighbors. Each neighborhood is relatively very small (compared to the total number of agents), with average size $\bar{d}$.

Within LTIS, a common set of solution techniques has been adopted (Glinton et al., 2009, 2010, 2011; Pryymak et al., 2012). First, agents communicate only *summarized* information representing their current belief about F, instead of forwarding each individual observation from the sensors. These summarized beliefs are called *opinions* (also denoted by o, described below). This practice (1) simplifies the model, making it easier to study, (2) reduces the amount of potentially *costly* communication, (3) reduces the impact of *over-counting* information, since each agent could repeatedly receive the same sensor observations from multiple neighbors, and (4) hides raw observations which could be sensitive or include private information (e.g., military applications, Glinton et al., 2010).

Beliefs have been commonly represented as a probability distribution describing the likelihood that F is either $True$ or $False$. Agents start with an initial uncertain belief that any value for F is equally likely, then Bayesian updating is used to incorporate newly received information o:

$$b' = \frac{cp(o)\cdot b}{cp(o)\cdot b + (1 - cp(o))\cdot(1-b)} \tag{1}$$

where b is the probability that the fact is $True$ (so $(1-b)$ is the probability that the fact is $False$), b' is the updated belief, and $cp(o)$ is a function describing the conditional probability that the fact is true given the observation. Here, the value of cp is a *weight* for newly received information o (either a sensor observation or a neighbor's opinion), and its value depends on the value and source of o:

$$cp(o) = \begin{cases} r & \text{if } o = True \wedge o \text{ from a sensor} \\ 1-r & \text{if } o = False \wedge o \text{ from a sensor} \\ m_i & \text{if } o = True \wedge o \text{ from a neighbor} \\ 1-m_i & \text{if } o = False \wedge o \text{ from a neighbor} \end{cases} \tag{2}$$

For sensor observations, the weight depends on sensor accuracy r, whereas for opinions from neighboring agents, the weight depends on a_i's *importance level* m_i, i.e., the likelihood that a_i's neighbors share correct opinions.

Because agent beliefs are uncertain, agents only share information when they become reasonably confident that a fact is either $True$ or not based on their received information. Specifically, an agent uses a confidence threshold $\sigma > 0.5$ to discretize its belief into confident opinions o:

$$o = \begin{cases} True & \text{if } b > \sigma \\ False & \text{if } b < 1 - \sigma \\ Unc & \text{else} \end{cases} \tag{3}$$

where Unc denotes an unconfident opinion, which is never shared through communication but only held by the agent when reflecting upon its belief.

For the rest of the paper (unless otherwise specified), we consider the following standard parameter values studied previously (e.g., Glinton et al., 2010): the number of agents $|A| = 1000$, the number of sensors $|S| = 0.05|A| = 50$, the sensor accuracy $r = 0.55$, the average neighborhood size $\bar{d} = 8$, and the confidence threshold $\sigma = 0.8$. For the weight m_i for opinions shared by neighbors, we use the optimal setting given the other parameters: $m_i = 0.63\ \forall a_i \in A$ (Glinton et al., 2010). For our experiments, we use 100 runs with different randomly generated teams.

Prior Work

LTIS research has primarily focused on two aspects: (1) defining models describing the effects of various parameters on information sharing, and (2) developing distributed algorithms to achieve desired emergent behavior.

Glinton et al. (2009) first proposed the LTIS problem. Using branching process theory (2010), they developed a model predicting that different settings of the cp information weighting parameter (specifically the importance level component m_i) can result in one of three phases of emergent behavior: (1) unstable dynamics, where too much weight for new information causes frequent avalanches of information shared between agents, resulting in oscillating beliefs, (2) stable dynamics, where too little weight in new information results in infrequent belief updates caused by 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 belief oscillation. Later, Glinton et al. (2011) discovered that LTIS was highly susceptible to fault when incorrect information was received (either caused by benign noisy observations or malicious injection by an attacker) and an agent's belief was near the confidence threshold σ.

Prior research has also focused on developing distributed algorithms for controlling information sharing by adapting the weight (i.e., importance levels m_i) placed in shared opinions in order to achieve desirable properties. First, Glinton et al. (2010) exploited their model to produce an algorithm (DACOR) that control avalanches within an agent's local neighborhood and globally achieves scale invariant dynamics. More recently, Pryymak et al. (2012) produced another algorithm (AAT) requiring no additional communication to improve belief convergence.

Non-Stationary Phenomenon

In this paper, we add non-stationarity to the standard LTIS model. First, we describe our changes to the model, then discuss why non-stationarity is difficult to handle within the commonly accepted solution techniques for LTIS. Finally, we empirically demonstrate this problem.

To model non-stationary phenomenon in LTIS, we extend the definition of the fact F describing the phenomenon to a time-dependent series $F(t)$, where $F(t)$ defines the value of the fact at time $t \in \mathbb{Z}^+$. For example, Fig. 1 presents (1) a periodic fact $F_{10}(t)$ that switches values between $True$ and $False$ every $\Delta t = 1000$ and (2) another non-stationary fact $F_1(t)$ that switches values once at time $t = 1001$. To account for the changes in the fact value, observations and opinions are time-stamped with the time t indicating when they were observed or shared. Time is discretized into intervals (called ticks), where each interval is the time required for sensors to produce a new observation and an agent to transmit information to a neighbor.

Converging to consistent, accurate beliefs about *non-stationary* phenomenon is a much more challenging problem than observing *stationary* phenomenon because of the amount of information required to correctly revise agents' beliefs after a phenomenon change. To illustrate (without loss of generality), consider the fact $F_1(t)$ provided in Fig. 1. In Fig. 2, we present an agent's beliefs over time after updates as (a) a continuous probability $0 \le b \le 1$, and (b) discrete opinions $\{False, Unc, True\}$ according to Eq. 3. The agent begins with a belief of pure uncertainty $b(0) = 0.5$ and must update its belief to $b(t) \ge \sigma$ in order to achieve a correct opinion of $True$. This requires a belief change of only $\Delta b = \sigma - 0.5$, denoted by (*) in Fig. 2.

However, after the non-stationary phenomenon changes values, the agent must receive a *large enough* sequence of new information to revise its beliefs from $b(t) \ge \sigma > 0.5$ to a later belief of $b(t') \le 1 - \sigma < 0.5$. This requires a belief change of $\Delta b \ge 2(\sigma - 0.5)$, denoted by (**). Therefore, properly revising beliefs for non-stationary phenomenon requires at least twice as much belief change as handling stationary phenomenon, and subsequently, twice the amount of observed and shared information. Please note that this is true for *any* weight placed in new information, since for any weight, two updates with opposing information simply cancel each other out (Eq. 1). Thus, regardless of weight, agents need just as much information as they gathered before the phenomenon first changed to re-reach pure uncertainty ($b = 0.5$), then even more information to revise to a newly correct belief.

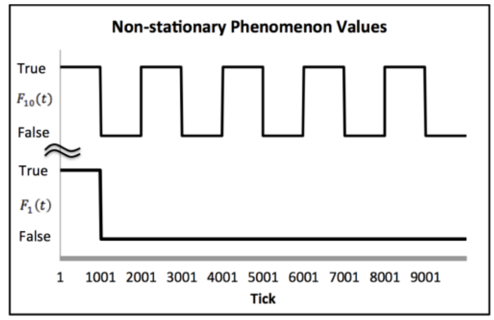

Fig. 1: Non-stationary Phenomenon Values

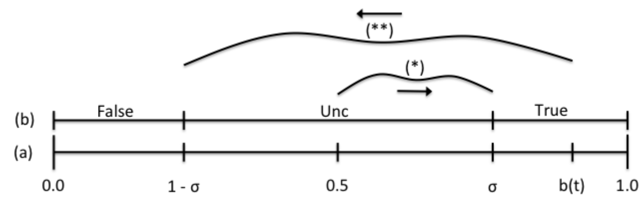

Fig. 2: Agent Belief Updates. Note: (*) distance to reach initial $True$ belief, (**) distance to reach later $False$ belief

Unfortunately, the distances (*) and (**) (in Fig. 2) also result in agents being *less likely to share opinions* from each individual belief update after the phenomenon has changed values than they would be with stationary phenomenon. Here, the team suffers from an **inertia problem**, where *too much information needs to be received by an agent to cause the agent to also share new opinions*. Specifically, recall that agents only share information with neighbors when they cross over a confidence threshold $b' \ge \sigma$ or $b' \le 1 - \sigma$. Since more updates are required to reach a confidence threshold after a phenomenon value change, each individual belief update is less likely to result in the agent sharing a new opinion with its neighbors. Therefore, agents actually share *fewer* opinions with one another. Unfortunately, this is *opposite* of what the agents need in order to adapt to the non-stationary phenomenon 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.

To demonstrate this inertia problem, we conduct an empirical study measuring the proportion of agents achieving accurate beliefs over time as the non-stationary phenomenon changes values, presented in Fig. 3. We vary the importance level placed in new information to confirm that than no ideal weight exists for handling non-stationary phenomenon, as opposed to the existence of an ideal value for stationary phenomenon according to Glinton et al.'s (2010) branching process model. For simplicity, we consider the phenomenon $F_1(t)$ that changes values only once (similar results occur with the more complicated $F_{10}(t)$).

From these results, we observe that although the team converged to consistent, accurate beliefs for the *initial*

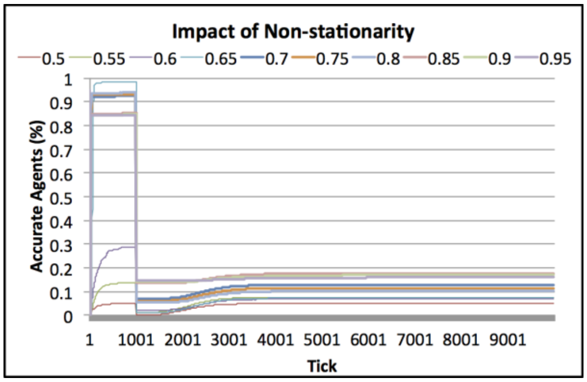

Fig. 3: Impact of Non-stationarity

value of the non-stationary phenomenon (identical to handling stationary phenomenon), a much smaller proportion of agents correctly revised their beliefs *over time*. Indeed, the majority of agents still retained the initial phenomenon value in their beliefs as they are unable to overcome their inertia. Since appropriately choosing a weight for new information is thus not a viable solution for handling non-stationarity (as previously studied for stationary phenomenon), we instead require a new type of solution.

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 desired emergent behavior. Specially, we develop an approach for 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 phenomenon. If the team were able to detect when the phenomenon changes values, then the agents could treat a new value *independent* of the previous value (i.e., as a separate stationary phenomenon). Most importantly, agents would need less information to revise their beliefs, increasing the team's convergence to consistent, accurate beliefs. However, we must avoid incorrectly detecting phenomenon changes, or else the agents' beliefs could oscillate and not converge as desired (similar to unstable team dynamics (Glinton et al. 2010)).

To detect changes to the non-stationary phenomenon, we actually exploit the inertia problem identified in the previous section. Considering how much information is needed to revise an agent's belief (i.e., (**) in Fig. 2), which causes inertia, we note that an agent is very unlikely to share an opinion that conflicts with a previously shared opinion without a phenomenon change. In our example, after the agent has shared a $True$ opinion, sharing a new $False$ opinion indicates to its neighbors that it received much new information reflecting a phenomenon change. The likelihood that this large amount of information would be incorrect is very small. Therefore, an agent sharing an opinion conflicting with its past opinion (having overcome its own inertia) is a likely indicator of phenomenon change to help other agents also overcome their inertia.

```
detectAndRespond(opinion)
if (opinion.value != lastOpinionValue(opinion.sender))
        then // Detected a change?
    rand ~ U(0, 1)
    if (rand ≤ σ) then  // Probabilistically reset beliefs?
        b ← 0.5   // Reset to pure uncertainty
        sendDetectedChangeAlert()  // Broadcast detection
    end if
end if
updateBelief(opinion) // Using Eq. 2
```

Algorithm 1: Detecting and Responding to Change

After detecting a phenomenon change, each agent responds as follows (detailed in Algorithm 1). First, the agent resets its own belief to pure uncertainty ($b = 0.5$), starting a new belief about the observed phenomenon. Next, it broadcasts its detection to its neighbors that are farther away from sensors and thus less likely to have already detected a change as information propagates through the team, encouraging them to also reset their beliefs. Afterwards, it updates its belief using the shared opinion (using Eq. 2). This reaction behavior simultaneously (1) puts agents in a position to quickly revise their beliefs after a detected phenomenon change, and (2) spreads the detection of phenomenon changes within the team to speed up convergence to accurately revised beliefs.

Addressing Concerns. However, one important concern with this solution is that agents trust the uncertain information shared by neighbors too much and might react inappropriately to new opinions. That is, if a neighbor shares an incorrect opinion conflicting with previously shared information, then a false change would be detected and agents would unnecessarily reset their beliefs. Our solution mitigates this concern in three targeted ways. First, agents only reset their beliefs with probability σ, reflecting the *same uncertainty* the sharing neighbor has in its shared belief (Eq. 3). Second, our solution only *locally* reacts within two network hops from the agent that changed opinions (since change detection is only propagated to immediate neighbors of the detecting agent), minimizing the impact of false detection on the entire team. Recall that the team's average connectivity $\bar{d}$ is assumed to be rather small, so these are very local behaviors. Finally, even if an agent resets its beliefs at an incorrect time, it *only* changes its opinion to *uncertain* and does not fully adopt the neighbor's incorrect information. Thus, the agent can reconverge 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.

Evaluation. To evaluate our proposed solution, we conduct another empirical study with the same setup as the previous section, but with the more complicated phenomenon $F_{10}(t)$ that *periodically* changes values. Fig. 4 presents the proportion of accurate agents over time.

First, we observe that unlike our previous results, the team of agents using our algorithm was *indeed* capable of adapting its beliefs over time and converged to consistent, accurate beliefs in a large proportion of the agents. This is evidenced in the high accuracies the curves eventually

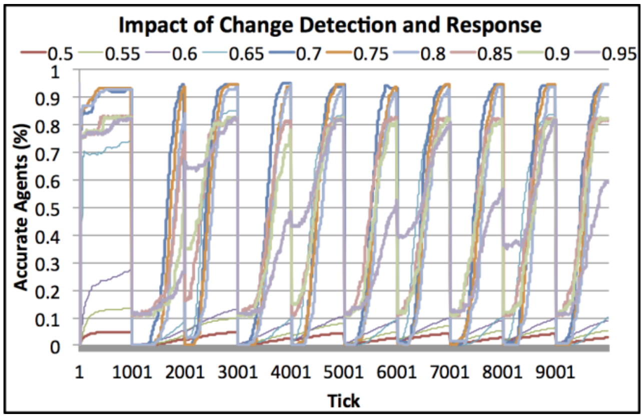

Fig. 4: Impact of Change Detection and Response

reached between phenomenon value changes (e.g., within 1001-2000 ticks, 2001-3000 ticks, etc.).

Second, we note that the team's performance depended greatly on the amount of weight placed in new information, similar to working with stationary phenomenon (Glinton et al. 2010). For both very low (e.g., < 0.65) and very high (e.g., > 0.85) weights, fewer agents converged to accurate beliefs, compared to mid-range weights (between 0.65 and 0.85). This matches our expectations with stationary phenomenon (as observed for the initial phenomenon value between 1 and 1000 ticks). This similarity occurs because we have transformed the problem back to one similar to dealing only with stationary phenomenon. However, the optimal level of weight was now *higher* than that predicted by Glinton et al.'s (2010) model for stationary phenomenon. Specifically, when agents used a weight of 0.70 (>0.63 for stationary phenomenon), they achieved the greatest accuracy over time. We believe that this discrepancy is caused by different local neighborhoods detecting changes at slightly different times within the team (i.e., latencies), decreasing information flow and requiring slightly higher weights to properly incorporate new information. If our approach were more global in detecting and responding to changes, the problem would be even more similar to handling stationary phenomenon as the entire team would reset at the same time. However, this would be riskier since only one false change detection would unnecessarily force the entire team into uncertainty.

Discussion. We also note two important issues with our first solution. First, there was always a significant delay between a phenomenon value change (occurring every 1000 ticks) and when the majority of the team converged to accurate beliefs. This was due to the amount of information required for some agents to first change opinions, which triggered change detection throughout the team. For more frequent phenomenon changes, the team might not react fast enough and could miss some changes. Second, the approach is vulnerable to *malicious* agents (similar to LTIS studied by Glinton et al. (2011)). Specifically, a small number of agents in a non-cooperative setting could intentionally send incorrect conflicting opinions or detected change alerts to their neighbors, affecting local neighborhoods by causing belief resets at inappropriate times and greatly impacting the team's convergence to consistent, accurate beliefs. In the following, we develop another solution that (1) has a free parameter for controlling how quickly agents recognize phenomenon change, and (2) is less vulnerable to malicious agents.

Forgetting Outdated Beliefs

Our second solution is based on the natural assumption that if an agent hasn't received information for a while, its beliefs are less likely to reflect the current value of a non-stationary phenomenon since the phenomenon's value changes over time. Based on this assumption, the agent's beliefs should become *less confident* the longer time has elapsed since the agent last received new information and updated its beliefs. Then, the agent would be more likely (1) to reach a confidence threshold opposing its most recent opinion after a belief update in order to form a new correct belief, and (2) to propagate new opinions throughout the team, enabling other agents to also correctly revise their beliefs. However, care must be taken to ensure that each agent doesn't become uncertain when the phenomenon has *not* changed values, which would cause the team to fail to converge to consistent, accurate beliefs.

Strategy. 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., Murdock 1993), as well as for related problems in artificial intelligence, such as situational awareness (e.g., Hoogendoorn et al. 2011) and information foraging with fewer agents that each directly observe the environment (e.g., Reitter and Lebiere 2012). However, while this approach has been used in other domains, this paper is the first application of belief decay to LTIS. Such an approach is especially strategic for LTIS because each agent (1) adjusts its beliefs *independent* of its neighbors, beneficial in non-cooperative situations (e.g., with malicious agents), and (2) can control the rate of decay, useful for adapting to various frequencies of change in non-stationary phenomenon.

For this solution, we propose adding the following rule to each belief update when an agent receives new information (before incorporating the new information, Eq. 1):

$$b'(t) = 0.5 + (b(t) - 0.5)\lambda^{\delta} \quad (4)$$

where δ represents the amount of time elapsed since the agent's last belief update and $\lambda \in (0, 1)$ is a parameter that controls how quickly the agent's belief decays over time: a smaller λ causes faster decay, whereas a larger λ causes slower changing beliefs. Thus, by choosing an appropriate λ, an agent can adjust how quickly it forgets old information and reacts to changed phenomenon values (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 since its last belief update. Afterwards, performing belief updates with

Eq. 1 incorporates new information into the time-adjusted belief, allowing the agent to potentially cross a confidence threshold so that it can share a new opinion with its neighbors. Of note, another way of looking at belief decay using Eq. 4 is *time-dependent information weighting*. That is, Eq. 4 weights *older* information (already incorporated in the agent's belief) *down* towards uncertainty before incorporating new information using Eq. 1.

Addressing Concerns. To avoid mass uncertainty when the phenomenon has *not* changed values, we propose only decaying beliefs when *new information is received* instead of *every tick*. Recall that most agents infrequently receive new information only when neighbors share new opinions. Decaying every tick would constantly push agents towards uncertainty (even if the phenomenon has *not* changed values). This would make it difficult for agents to reach and maintain confident beliefs, similar to the stable dynamics problem observed by Glinton et al. (2010) where too little weight in new information causes the team to remain uncertain over time. Instead, if agents only decay when new information is received as we prescribe, then agents only forget possibly outdated information if and when they have evidence that the phenomenon might have changed (i.e., when it is most appropriate to forget older beliefs). In other words, an agent cautiously holds on to its belief and only forgets older information when it sees new evidence.

Evaluation. To evaluate our belief decay solution, we conduct a final empirical study with the same parameters used previously (including $F_{10}(t)$). However, rather than varying the weight parameter (fixing $m_i = 0.63$ for all agents, which is optimal for stationary phenomenon (Glinton et al. 2010)), we instead vary λ to test how different decay rates affect agent beliefs. Particularly, we set $\lambda = e^{-1/T_0}$ with $T_0 = 100, 200, \dots, 1000$ (where 1000 is Δt, the period of the phenomenon changes). We present accurate agent proportions for this experiment in Fig. 5.

First, we again observe that using our belief decay solution, the team was able to revise its beliefs over time to accurately reflect changes in the value of the non-stationary phenomenon. Importantly, recall that the agents were able to do so by acting *independently*, focusing only on adapting their own beliefs with no additional communication or coordination. Thus, this solution is less susceptible to undue influence by malicious agents and is more appropriate for non-cooperative settings.

Second, we also observe that the team's convergence to accurate beliefs strongly depended on the value of λ (through varying T_0). Specifically, for smaller T_0 (and thus λ) values, the team adapted its beliefs faster and more agents achieved accurate beliefs before the phenomenon changed values again, as opposed to larger T_0 values. This result matches our earlier intuition about Eq. 4.

Discussion. However, we also observe that *increased convergence rates came at the cost of decreased stability*, since smaller T_0 (and thus λ) values resulted in more *variability* (i.e., less stability) in the proportion of agents with accurate beliefs between dynamic fact value changes (e.g., between 1001-2000 ticks). Therefore, using belief decay

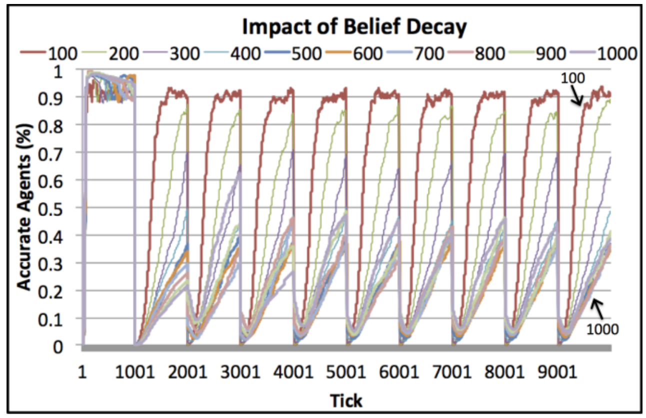

Fig. 5: Impact of Belief Decay

causes an interesting tradeoff between *convergence rates* and *stable beliefs*. If an agent allows its belief to decay too quickly in order to quickly adapt to phenomenon changes, then the system could destabilize; and vice-versa. In the future, we intend to study how λ should be chosen to appropriately balance this tradeoff depending on team and environment characteristics.

Conclusions

In conclusion, we introduced the problem of non-stationarity in observed phenomenon into the existing LTIS model and literature. We demonstrated both analytically and empirically that standard solution techniques for LTIS struggle to adapt agent beliefs after the phenomenon changes values because not enough information is shared to greatly change the agents' belief probabilities (i.e., the inertia problem). Most importantly, this is true *regardless* of the weight placed on new information, which has been the primary focus of prior solutions—making them not readily adoptable or applicable to handle non-stationary phenomenon. Therefore, we proposed two solutions addressing inertia in different ways: (1) a change detection and response algorithm exploiting inertia to benefit neighbors, and (2) a forgetting-based update rule matching time varying uncertainty and changes to the phenomenon over time. Empirically, we demonstrated that both solutions enable the team to converge to consistent, accurate beliefs as the non-stationary phenomenon changes values. Our first solution is best for cooperative environments where all agents can be trusted, whereas our second solution enables agents to act independently (protecting against malicious agents) and has an adjustable parameter for controlling the convergence rate during agent belief revision.

In the future, we intend to enhance this research by producing a better analytical model describing information sharing dynamics under non-stationarity in order to better understand the differences between stationary phenomenon (with strong existing analytical models) and more challenging non-stationary phenomenon. Using this model, we could potentially enhance both of our solutions (e.g., more accurate and timely change detection, as well as tuning or learning an appropriate λ decay rate online).

References

An, B. et al. 2011. Agent-mediated multi-step optimization for resource allocation in distributed sensor networks. Proc. of AAMAS’11. 609-616.

Boutilier, C. 2002. A POMDP formulation of preference elicitation problems. Proc. of AAAI’02, 239-246.

Glinton, R.T., Scerri, P., and Sycara, K. 2009. Towards the understanding of information dynamics in large scale networked systems. Proc. of FUSION’09.

Glinton, R., Scerri, P., and Sycara, K. 2010. Exploiting scale invariant dynamics for efficient information propagation in large teams. Proc. of AAMAS’10, 21-28.

Glinton, R., Scerri, P., and Sycara, K. 2011. An investigation of the vulnerabilities of scale invariant dynamics in large teams. Proc. of AAMAS’11.

Hoogendoorn, M., van Lambalgen, R. M., and Treur, J. 2011. Modeling situational awareness in human-like agents using mental models. Proc. of IJCAI’11. 1697-1704.

Murdock, B.B. 1993. TODAM2: A model for the storage and retrieval of item, associative, and serial-order information. Psychological Review, 100(2), 183-203.

Padhy, P., Dash, R.K., Martinez, K., and Jennings, N.R. 2006. A utility-based sensing and communication model for a glacial sensor network. Proc. of AAMAS’06, 1353-1360.

Pryymak, O., Rogers, A., and Jennings, N.R. 2012. Efficient opinion sharing in large decentralized teams. Proc. of AAMAS’12.

Reitter, D. and Lebiere, C. 2012. Social cognition: memory decay and adaptive information filtering for robust information maintenance. Proc. of AAAI’12. 242-248.

Spaan, M.T.J., Veiga, T.S., and Lima, P.U. 2010. Active cooperative perception in networked robotic systems using POMDPs. In Proceedings of IROS’10, 4800-4805.

Williams, J.D. and Young, S. 2007. Partially observable Markov decision processes for spoken dialog systems, Computer Speech and Language, 21. 393-422.