Data Fusion without Knowledge of the Ground Truth Using Tseltin-like Automata

Anis Yazidi, Frode Eika Sandnes
Dept. of Computer Science
Faculty of Technology, Art and Design
Oslo and Akershus University College of Applied Sciences
Oslo, Norway
Email: anis.yazidi@hioa.no

Abstract—**The fusioning of data from unreliable sensors has received much research attention. The main stream of research assesses the reliability of a sensor by comparing its readings to the ground truth in an online or offline manner. For instance, the Weighted Majority Algorithm is a representative example of a large class of similar legacy algorithms. Recently, some advances have been achieved in identifying unreliable sensors *without any knowledge of the ground truth* which seems a paradox in itself. In this paper, we present a simple mechanism for solving the problem using Tsetlin-like Learning Automata (LA). Our approach leverages a Random Walk (RW) inspired by Tsetlin LA so that to gradually learn the identity of the reliable and unreliable sensors. In this perspective, we resort to a team of RWs, where a distinct RW is associated with each sensor. By virtue of the limited memory requirement of our devised LA, we achieve adaptive behavior at the cost of negligible loss in the accuracy.**

Keywords: *Sensor Fusion, Random Walk, Learning Automata*

I. Introduction

Data fusion from uncertain sources of information is an important research topic that has gained an increasing research attention during recent years [1], [2]. Furthermore, it is known that fusioning information from a set of unreliable sensors can give a more robust information about the process being monitored [3], [4]. An important body of research has focused on assessing the reliability of a sensor or more generally an "information source" by comparing the readings with the ground truth. The Weighted Majority Voting algorithm [5], a well-known Machine Learning algorithms, is a typical example of a class of approaches that assess the reliability of a sensor by comparing its readings to the ground truth in a online manner. Once the reliability of the sensors is inferred, this information can be used as input to a fusioning process so that to mitigate the undermining effect of the unreliable sensors on the quality of the fusioned information. However, in many real life applications, access to the ground truth is simply impossible. This is particularly true in the field of "Softsensing" where the harsh nature of the environment prohibits accessing the ground truth. In such settings where the ground truth is inaccessible, the question of assessing the reliability of the sensors is *apparently* impossible to solve. A recent work by Yazidi et al. [6], [7] has pioneered a solution by which it is feasible to solve the problem of identifying which sensors are unreliable *without any knowledge of the ground truth*, a claim that *is counter-intuitive*. The essence of the approach presented in [6], [7] stems from the simple intuition that the "agreement" between the sensors themselves can give invaluable knowledge about their respective reliabilities. In a stochastic environment where errors can take place according to some unknown underlying stochastic process, those sensors that tend to deviate from the decision of the majority are more likely to be unreliable than those that adhere to the decision of the majority. Such simple and intuitive remark works under the premise that the decision of the majority has some high likelihood of revealing the truth [6], [7]. Moreover, Yazidi et al. [6], [7] provided a formalism to model the latter description with a rigorous mathematical framework invoking results from Boland [8] which can be regarded as a generalization of the well-known Concordet Jury Theorem for majority voting. Boland [8] treats the generalized case where the voters are divided into two groups, where each group has a different "true" interest from the opposite group. Thus, Boland's work [8] can be considered as an extension of theory of Concordet on majority voting under which the voters are homogeneous, forming a single group. Inspired by the work of Boland [8], we model the pool of sensors as two groups, namely reliable group and unreliable group, where the "true interest" of the reliable group is to report the ground truth whereas the "true interest" of the unreliable group is to merely misreport the ground truth. This subtle remark led to the solution reported in [6], [7]. To the best of our knowledge, this is the first solution in the literature that treats the problem of information fusion without knowledge of the ground truth and thus represents the state-of-the-art. The main tool used to solve that problem was the theory of Variable Structure Learning Automata (VSLA) [9]. Nevertheless, a major disadvantage with the previous solution was the rather complex manner by which reward and penalty were defined as well as the lack of adaptivity in dynamic settings where sensors might "drift" from reliable state to unreliable state or vice-versa. This paper describes an alternative solution to the problem by devising a LA that falls under the category of Fixed Structure LA (FSLA) [9]. Our LA is adaptive by virtue of its finite memory, and thus is able to track the reliability of the sensors under dynamic environment. In addition, the solution is simpler than the original solution presented in [6], [7]. In fact, we propose a more intuitive manner by which reward and penalty are defined. A reward reflects more confidence in the current hypothesis about the identity of the sensor, and thus, triggers a transition towards the most internal state of the current action of the Tsetlin-like LA [9] while a penalty weaknesses the current hypothesis by moving towards the corresponding boundary state of the current arm.

In short, we propose to solve the Sensor-Type Partitioning Problem ($STTP$) using a RW inspired of Tsetlin-Automata [10], [11]. Our solution can adaptively and in an on-line manner distinguish between reliable sensors and unreliable senors using finite memory. We present theoretical results that illustrate the convergence of the scheme as well as some representative empirical experiments.

A. Paper Organization

Section I presented the research problem and reviewed some of the related work. The rest of the paper is organized as follows. Section II gives a formal statement of the problem. Thereafter, in Section III we present our solution, which is the Tsetlin-like LA scheme for identifying unreliable sensors in a stochastic environment in the absence of knowledge of the ground truth. Some experimental results that validates the theoretical results are presented in Section IV. Section V concludes the paper.

II. Modeling the Problem

We consider a population of N sensors, $\mathcal{S} = \{s_1, s_2, \ldots, s_N\}$. Let the unknown ground truth at the time instant t be modeled by a binary variable $T(t)$, which can take one of two possible values, 0 and 1. The value of T is unknown and can only be inferred through measurements from sensors. The output from the sensor s_i is referred to as x_i. Let π be the probability of the state of the ground truth, i.e., $T = 0$ with probability π.

We suppose that the probability of the sensor reporting a value erroneously is symmetric. Formally, this reduces to:

$$Prob(x_i = 0|T = 1) = Prob(x_i = 1|T = 0). \tag{1}$$

Further, let p_i denote the Correctness Probability (CP) of sensor s_i, where:

$p_i = Prob(x_i = 0|T = 0) = Prob(x_i = 1|T = 1)$.

It is easy to proof $Prob(x_i = T)$ is, indeed, p_i.

We can define a reliable sensor to be one that has a CP $p_i > 0.5$ and an unreliable sensor as one that has a CP $p_i < 0.5$.

In addition, we assume that every p_i can have one of two possible values from the set $\{p_R, p_U\}$, where $p_R > 0.5$ and $p_U < 0.5$. Then, a sensor s_i is said to be reliable if $p_i = p_R$, and is said be unreliable if $p_i = p_U$. We assume that p_R and p_U are unknown to the algorithm.

Based on the above, the set of reliable sensors is $\mathcal{S}_R = \{s_i|p_i = p_R\}$, and the set of unreliable sensors is $\mathcal{S}_U = \{s_i|p_i = p_U\}$.

Throughout this paper, we will resort to the following assumption: $(N_R - 1)p_R + N_U p_U > (N_R + N_U)/2$.

The above mild condition that we formulate in this paper rests on the philosophical fundament found in the society where the truth is a virtue among the individuals, and that the truth prevails over lies.

III. The Solution

A. Overview

In this section, we provide a novel solution to the $STTP$, based on the field of RW and more particularly Tsetlin LA. In his pioneering work, Tsetlin [10] attempted to use LA to model biological learning. In the first LA designs, the transition and the output functions were time invariant, and for this reason these LA were considered "Fixed Structure Stochastic Automata" (FSSA) [11], [12]. It is worth mentioning that FSLA are regaining renewed attention in the literature due to their plausible properties [12].

Our solution involves a team of LA where each LA is uniquely attached to a specific sensor. Each automaton attached to sensor s_i, has two actions. The aim of LA is to infer the identity of the sensor in question by exploiting the agreement/disagreement of the sensor in question with the majority voting of the rest of the sensors as a form for feedback. First, for the sake of completeness, we will present two main theorems [7] that we will use in the design of our LA in Section III-B.

The respective proofs of Theorem 1 and Theorem 2 are found in [7].

Theorem 1: Consider the scenario when $(N_R - 1)p_R + N_U p_U > (N_R + N_U)/2$ and when $N_R + N_U - 1 \geq 3$. Let $s_i \in \mathcal{S}_R$. Consider now the agreement between the opinion of a reliable sensor s_i and the opinion of the majority formed by all the rest of the sensors $S\backslash\{s_i\} = (\mathcal{S}_R\backslash\{s_i\}) \cup \mathcal{S}_U$. Let $y_{(N_R-1,N_U)}$ be the decision of a majority voting scheme $S\backslash\{s_i\}$, based on the responses of $N_R - 1$ reliable and N_U unreliable sensors.

Then, if x_i is the output of s_i: $Prob(x_i = y_{(N_R-1,N_U)}) > 0.5$.

We shall now consider the converse case of the agreement of an unreliable sensor with the decision of the majority voting scheme formed of by the rest of the sensors.

Theorem 2: Consider the scenario when $(N_R - 1)p_R + N_U p_U > (N_R + N_U)/2$ and when $N_R + N_U - 1 \geq 3$. Let $s_i \in \mathcal{S}_U$. Consider now the agreement between the opinion of an unreliable sensor s_i and the opinion of the majority formed by all the rest of the sensors $S\backslash\{s_i\} = \mathcal{S}_R \cup \mathcal{S}_U\backslash\{s_i\}$. Let $y_{(N_R,N_U-1)}$ be the decision of a majority voting scheme formed of $S\backslash\{s_i\}$, based on the responses of N_R reliable and $N_U - 1$ unreliable sensors.

Then, if x_i is the output of s_i: $Prob(x_i = y_{(N_R,N_U-1)}) > 0.5$.

B. Construction of the LA

This section describes the design of the Tsetlin-like LA that we have devised in order to infer the identity of a sensor. In brief, the task of an LA is to decide whether a specific sensor is reliable or unreliable. By observing agreement of the sensor with the reset, the correctness of an hypothesis is inferred.

The current LA is inspired by the family of fixed structured LA [11]. Accordingly, a LA can be defined in terms of a quintuple [11]:

$$\{\underline{\Phi}, \underline{\alpha}, \underline{\beta}, \mathcal{F}(\cdot,\cdot), \mathcal{G}(\cdot,\cdot)\}.$$

Here, $\underline{\Phi} = \{\phi_1, \phi_2, \ldots, \phi_s\}$ is the set of internal automaton states. $\underline{\alpha} = \{\alpha_1,\ \alpha_2, \ldots,\ \alpha_r\}$ is the set of automaton actions. Further, $\underline{\beta} = \{\beta_1, \beta_2, \ldots, \beta_m\}$ is the set of inputs that can be given to the automaton. An output function $\alpha_t = \mathcal{G}[\phi_t]$ determines the action performed (or chosen) by the automaton given the current automaton state. Finally, a transition function $\phi_{t+1} = \mathcal{F}[\phi_t, \beta_t]$ determines the new state of the automaton from: (1) The current state of the automaton and (2) The response of the environment.

Based on the above generic framework, the crucial issue is to design automata that can learn the optimal action when interacting with the environment. Several designs have been proposed in the literature, and the reader is referred to [11] for an extensive treatment.

Briefly stated, we construct an automaton with 2δ states:

- States: $\underline{\Phi} = \{1, \ldots, 2\delta\}$.
- Actions: $\underline{\alpha} = \{\alpha_0, \alpha_1\}$.
- Inputs: $\underline{\beta} = \{\mathit{Reward}, \mathit{Penalty}\}$.

We suppose that δ is a strictly positive integer.

Formally, G is defined as follows:

$$G(\phi_i) = \begin{cases} \alpha_0, & \text{if } \phi_i \in \{1, \ldots, \delta\} \\ \alpha_1, & \text{if } \phi_i \in \{\delta+1, \ldots, 2\delta\} \end{cases}$$

We refer to the states $\{1, \ldots, \delta\}$ as the *Unreliability States*, and to the states $\{\delta+1, \ldots, 2\delta\}$ as the *Reliability Tracking States*,

The $\mathcal{G}$ matrix can be summarized as follows. If the automaton state lies in the set $\{1, \ldots, \delta\}$(i.e, *Unreliability States*) the LA will choose the action α_0 which hypothesises that the sensor is unreliable. If, on the other hand, the state is one of the states in the set $\{\delta+1, \ldots, 2\delta\}$ (i.e, *Reliability States*), it will choose the action α_1 which assumes that the sensor is reliable. Thus, we have two actions, or arms, where each arm is composed of δ states. Note that since initially the identity of the sensor is unknown, we set the initial state of our automaton to δ which is simply the border state of the arm attached to action α_0.

The state transition matrix $\mathcal{F}$ determines how the learning proceeds. We follow the LA nomenclature used in [11] and denote $\mathcal{F}(0)$ the transition matrix in case of Reward and $\mathcal{F}(1)$ the transition matrix in case Penalty.[1]

[1]In LA theory, "0" usually refers to Reward while "1" is used for Penalty.

The matrix $\mathcal{F}(0)$ is defined as:

$$\mathcal{F}(0) = \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 & 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 & 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & 1 \\ 0 & 0 & \ldots & 0 & 0 & 0 & 0 & \ldots & 1 \end{pmatrix}$$

The matrix $\mathcal{F}(1)$ is given by:

$$\mathcal{F}(1) = \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 0 & 1 & 0 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 & 1 & 0 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 & 0 & 1 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \ldots & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Now, let us describe the transition of the reward and penalty mechanism described by $\mathcal{F}(0)$ and $\mathcal{F}(1)$:

1) Case when sensor s_i is in the states $\{1, \ldots, \delta\}$ referred to as *Unreliability States*: A reward will take place whenever s_i disagrees with the opinion of the majority voting scheme associated with $S \backslash \{s_i\}$. Thus, the automaton will move one step towards the most internal state 1. Loosely, speaking, a reward takes place since the disagreement increases the certitude in the hypothesis that s_i is unreliable. Similarly, a penalty will take place whenever s_i agrees with the the opinion of the majority voting which makes the automaton move one step forward towards the boundary state δ. In the latter case, whenever the automaton is in the boundary state δ and is penalized, it will advance to state $\delta+1$ and consequently will switch action to α_1 according to G.
2) Case when sensor s_i is in the states $\{\delta+1, \ldots, 2\delta\}$ referred to as *Reliability States*: A reward will take place whenever s_i agrees with the opinion of the majority voting scheme associated with $S \backslash \{s_i\}$. Thus, the automaton will move one step towards the most internal state 2δ. A reward takes place since the agreement increases the certitude in the hypothesis that s_i is reliable. Similarly, a penalty will take place whenever s_i disagrees with the the opinion of the majority voting which makes the automaton move one step backward towards the boundary state $\delta+1$. In the latter case, whenever the automaton is in state $\delta+1$ and is penalized, it will move to state δ and consequently the action is switched to α_0 according to G.

The philosophy behind the above definition of reward and penalty is inspired by Tsetlin Automata. A transition towards the internal states (i.e, state 1 and state 2δ) is considered as reward, a transition towards the boundary states is considered as penalty.

In simple terms, any backward transition from one of the states $\{1, \ldots, \delta\}$ is considered as reward as it confirms the hypothesis that the sensor is unreliable. While any forward transition from the latter states is a penalty.

Similarly, any forward transition from any of the states $\{\delta+1, \ldots, 2\delta\}$ is considered a reward as it confirms the hypothesis stating that the sensor reliable, while any backward transition is considered as penalty.

Let $P_0^i(\delta)$ the probability of choosing action α_0, which depends on the number of states 2δ. Let $P_1^i(\delta)$ the probability of choosing α_1.

Theorem 3: According to the definition of the automaton above, the probability of choosing the action α_0 by sensor s_i is given by the following expression:

$$P_0^i(\delta) = \frac{1 - [\lambda_i/(1-\lambda_i)]^{\delta}}{1 - [\lambda_i/(1-\lambda_i)]^{2\delta}} \tag{2}$$

where

$$\lambda_i = \begin{cases} Prob(x_i = y_{(N_R-1,N_U)}), & \text{if } s_i \in \mathcal{S}_R, \\ Prob(x_i = y_{(N_R,N_U-1)}), & \text{if } s_i \in \mathcal{S}_U. \end{cases} \tag{3}$$

Proof:

In order to compute $P_0^i(\delta)$, we need to examine the Markov Chain associated to the transitions of the automaton. The Markov chain has transition matrix $H^i = [h_{kl}^i]$ for $1 \le k, l \le 2\delta$. where

$$\begin{aligned} h_{k,k+1}^i &= \lambda_i, 1 \le k \le 2\delta - 1 \\ h_{k,k-1}^i &= 1 - \lambda_i, 2 \le k \le 2\delta \end{aligned}$$

The transitions for the most internal states are self loops and are given as:

$$\begin{aligned} h_{1,1}^i &= 1 - \lambda_i \\ h_{2\delta,2\delta}^i &= \lambda_i \end{aligned}$$

We shall now compute π_k^i, the stationary (or equilibrium) probability of the chain being in state k for the RW associated to sensor s_i. Clearly H^i represents a single closed communicating class whose periodicity is unity. The chain is ergodic, and the limiting probability vector is given by the eigenvector of ${H^i}^T$ corresponding to the eigenvalue unity.

The vector of steady state probabilities $\Pi^i = [\pi_1^i, \ldots, \pi_{2\delta}^i]^T$ can be computed using ${H^i}^T \Pi^i = \Pi^i$.

The stationary state probabilities π_s^i of the Markov chain must obey the following recurrence relation for the non-extremal states:

$$\pi_s^i = \lambda_i \pi_{s-1}^i + (1-\lambda_i)\pi_{s+1}^i. \tag{4}$$

We will prove Equation (5) defined by:

$$\pi_s^i = \left(\frac{\lambda_i}{1-\lambda_i}\right)^{s-1} \pi_1^i \qquad 1 \le s \le 2\delta, \tag{5}$$

where π_1 is a normalizing constant such that $\sum_{s=1}^{r} \pi_s^i = 1$, and is given by:

$$\pi_1^i = \frac{1 - \lambda_i/(1-\lambda_i)}{1 - [\lambda_i/(1-\lambda_i)]^{2\delta}}. \tag{6}$$

In order to simplify the notations, we define $r = 2\delta$.

Consider first the basic case for π_1 by setting $s = 1$. If we substitute the solution into the recurrence relation (4), we get:

$$\begin{aligned} \pi_s &= \lambda_i \pi_{s-1}^i + (1-\lambda_i)\pi_{s+1}^i \\ &= \lambda_i \left(\frac{\lambda_i}{1-\lambda_i}\right)^{s-2} \pi_1^i + (1-\lambda_i)\left(\frac{\lambda_i}{1-\lambda_i}\right)^{s} \pi_1^i \\ &= \frac{\lambda_i^{s-1}}{(1-\lambda_i)^{s-2}}\pi_1^i + \frac{\lambda_i^{s}}{(1-\lambda_i)^{s}}\pi_1^i - \frac{\lambda_i^{s+1}}{(1-\lambda_i)^{s}}\pi_1^i \\ &= \frac{\pi_1^i}{(1-\lambda_i)^s}\left[\lambda_i^{s-1}(1-\lambda_i)^2 + \lambda_i^s - \lambda_i^{s+1}\right] \\ &= \frac{\pi_1^i}{(1-\lambda_i)^s}\left[\lambda_i^{s-1}\left(1 - 2\lambda_i + \lambda_i^2\right) + \lambda_i^s - \lambda_i^{s+1}\right] \\ &= \frac{\pi_1^i}{(1-\lambda_i)^s}\left[\lambda_i^{s-1} - 2\lambda_i^s + \lambda_i^{s+1} + \lambda_i^s - \lambda_i^{s+1}\right] \\ &= \frac{\lambda_i^{s-1}(1-\lambda_i)}{(1-\lambda_i)^s}\pi_1^i = \frac{\lambda_i^{s-1}}{(1-\lambda_i)^{s-1}}\pi_1^i \\ &= \left(\frac{\lambda_i}{1-\lambda_i}\right)^{s-1} \pi_1^i, \end{aligned}$$

which shows that the recurrence is valid for the internal states. The stationary probability of the extreme state must be computed directly from the recurrence relation for the final state as this is different from the relation for the internal states. To do this, we observe that:

$$\pi_r^i = \lambda_i \pi_{r-1}^i + \lambda_i \pi_r^i.$$

Now sorting the state probabilities, we get:

$$\pi_r^i - \lambda_i \pi_r^i = \lambda_i \pi_{r-1}^i, \text{ whence } (1-\lambda_i)\pi_r^i = \lambda_i \pi_{r-1}^i.$$

Simplifying the above (after re-arranging), we see that:

$$\begin{aligned} \pi_r^i &= \left(\frac{\lambda_i}{1-\lambda_i}\right)\pi_{r-1}^i = \left(\frac{\lambda_i}{1-\lambda_i}\right)\left[\left(\frac{\lambda_i}{1-\lambda_i}\right)^{r-2}\pi_1^i\right] \\ &= \left(\frac{\lambda_i}{1-\lambda_i}\right)^{r-1}\pi_1^i. \end{aligned}$$

To obtain the normalizing constant, let $a = \lambda_i/(1-\lambda_i)$. Then the sum of π^i's over all states is

$$\sum_{s=1}^{r} \pi_s^i = \sum_{s=1}^{r} a^{s-1}\pi_1^i = \pi_1^i \sum_{s=1}^{r} a^{s-1}$$

Thus, from the formula for the sum of a finite power series,

$$= \pi_1^i \left[\frac{1 - a^{2\delta}}{1 - a}\right]$$

Note that the the normal requirement that $|a| < 1$, which is necessary for the convergence of an infinite power series sums, is not needed for *finite* sums. Since this sum must be unity for a probability distribution, the normalising constant can be found as:

$$\pi_1^i = \frac{1}{\frac{1-a^{2\delta}}{1-a}} = \frac{1-a}{1-a^{2\delta}} = \frac{1-\lambda_i/(1-\lambda_i)}{1-[\lambda_i/(1-\lambda_i)]^{2\delta}}.$$

Thus, we obtain:

$$\pi_s^i = \left(\frac{\lambda_i}{1-\lambda_i}\right)^{s-1} \pi_1^i \qquad 1 \le s \le r, \tag{7}$$

where π_1 is a normalizing constant such that $\sum_{s=1}^{r} \pi_s^i = 1$, and is given by:

$$\pi_1^i = \frac{1-\lambda_i/(1-\lambda_i)}{1-[\lambda_i/(1-\lambda_i)]^{2\delta}}. \tag{8}$$

Now, we are ready to compute $P_0^i(\delta)$

$$P_0^i(r) = \sum_{s=1}^{\delta} \pi_s^i \qquad = \frac{1-[\lambda_i/(1-\lambda_i)]^{\delta}}{1-[\lambda_i/(1-\lambda_i)]^{2\delta}} \tag{9}$$

■

Theorem 4: Consider the scenario when $(N_R - 1)p_R + N_U p_U > (N_R + N_U)/2$ and when $N_R + N_U - 1 \ge 3$. Whenever the number of states tend to infinity, the following is true

If $s_i \in \mathcal{S}_R$, then $\lim_{r\to\infty} P_1^i(\delta) \to 1$;
If $s_i \in \mathcal{S}_U$, then $\lim_{r\to\infty} P_0^i(\delta) \to 1$.

Proof: The proof relies on examining the two cases $\lambda_i > \frac{1}{2}$ and $\lambda_i < \frac{1}{2}$ separately. In addition, it is easy to observe that $P_1^i(\delta) + P_0^i(\delta) = 1$. The details of the proof is omitted here for the sake of space limitations.

■

IV. Experimental results

In order to assess the accuracy of our approach we compute $P_1^i(\delta)$ for different sizes of the state space 2δ, different pool sizes of the sensors, and different values of (p_R, p_U). Please note that as stated in our theoretical results, $P_1^i(\delta)$ denotes the accuracy of the scheme. All the experiments were run for 1000 iterations and average out over an ensemble of 1000 experiments. In Tables I, II and III, we increase the number of states 2δ from 10 to 1000. First and most importantly, we observe that for a value of 2δ as small as 10 (see Table I), the accuracy approaches 1 which is quite impressive given the simplicity of the automaton as well as its limited memory footprint. As we increase 2δ to 1000, the accuracy increases slightly. For example for $(p_R, p_U) = (0.8, 0.1)$ and for $2\delta = 10$, the accuracy for the reliable sensors and unreliable sensors are 0.996844, 0.999593 respectively, while for $2\delta = 100$, we obtain slightly larger values, namely, 0.998555 and 0.999676. Note that for $2\delta = 1000$, the corresponding values are 1, 1. Thus, we can conclude that the gain resulting from increasing the memory in terms of accuracy is "diminishing".

Another key remark concerns the effect of (p_R, p_U) on the accuracy. In the experiments, the settings were chosen so that the condition $N_R p_R + (N_U - 1)p_U > (N_R + N_U)/2$ was met, reflecting the phenomenon where "the truth prevails over lying". The higher the gap between p_R and p_U, the "easier" is the environment and the more accurate is the scheme. For example, for $2\delta = 10$, and for $(p_R, p_U) = (0.8, 0.2)$, the accuracy is 0.997637 and 0.998794. By increasing p_R to 0.95 and decreasing p_U to 0.1, the accuracy is improved to 0.99986 and 0.999833.

(p_R, p_U)	Accuracy for $s_i \in \mathcal{S}_R$	Accuracy for $s_i \in \mathcal{S}_U$
(0.8, 0.1)	0.996844	0.999593
(0.8, 0.2)	0.997637	0.998794
(0.85, 0.1)	0.999041	0.999825
(0.85, 0.2)	0.999146	0.999035
(0.9, 0.1)	0.999554	0.999811
(0.9, 0.2)	0.999598	0.999238
(0.95, 0.1)	0.99986	0.999833
(0.95, 0.2)	0.999834	0.999221

TABLE I: Accuracy for the case when $(N_R, N_U) = (40, 20)$, $2\delta = 10$.

(p_R, p_U)	Accuracy for $s_i \in \mathcal{S}_R$	Accuracy for $s_i \in \mathcal{S}_U$
(0.8, 0.1)	0.998555	0.999676
(0.8, 0.2)	0.998531	0.999222
(0.85, 0.1)	0.999183	0.999817
(0.85, 0.2)	0.999255	0.999452
(0.9, 0.1)	0.999598	0.999822
(0.9, 0.2)	0.999612	0.999449
(0.95, 0.1)	0.999832	0.999875
(0.95, 0.2)	0.999809	0.999462

TABLE II: Accuracy for the case when $(N_R, N_U) = (40, 20)$, $2\delta = 100$.

(p_R, p_U)	Accuracy for $s_i \in \mathcal{S}_R$	Accuracy for $s_i \in \mathcal{S}_U$
(0.8, 0.1)	1.0	1.0
(0.8, 0.2)	1.0	1.0
(0.85, 0.1)	1.0	1.0
(0.85, 0.2)	1.0	1.0
(0.9, 0.1)	1.0	1.0
(0.9, 0.2)	1.0	1.0
(0.95, 0.1)	1.0	1.0
(0.95, 0.2)	1.0	1.0

TABLE III: Accuracy for the case when $(N_R, N_U) = (40, 20)$, $2\delta = 1000$.

Furthermore, we perform some experiments that illustrate the ability of our scheme to adapt to an arbitrarily changing environment, where a sensor might drift from being reliable to unreliable and vice-versa. We also test the effect of the number of states on the accuracy of the scheme. We chose $(p_R, p_U) = (0.8, 0.1)$ and $(N_R, N_U) = (40, 20)$. Initially, the sensors with indices from 0 to $N_R - 1 = 39$ are reliable, while sensors with indices in the range N_R to $N_R + N_U - 1 = 59$ are unreliable.

In order to simulate a drift in the environment, at time instant 500, we invert the identity of sensor s_0 to being unreliable, while also inverting the identity of sensor s_{59} and rendering it reliable. In simple terms, before time instant 500, $(p_0, p_{59}) = (0.8, 0.1)$ and then after time instant 500, we will swap the identity of the sensors by merely choosing $(p_0, p_{59}) = (0.1, 0.8)$.

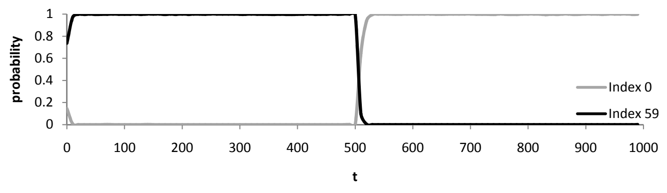

Fig. 1: $2\delta = 10$

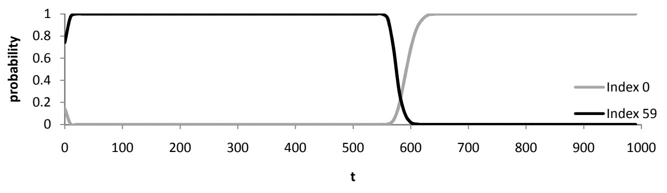

Fig. 2: $2\delta = 100$

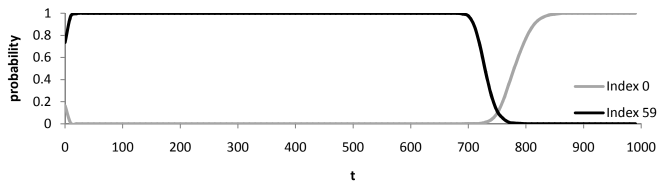

Fig. 3: $2\delta = 300$

Fig. 4: Dynamics of the probability of choosing action α_0 of sensor 0 and sensor 59 with environment's change at $t = 500$.

In Fig. 1, we report the evolution of the probability of choosing action α_0 for sensor 0 and sensor 59 for $2\delta = 10$ by averaging out an ensemble of 1000 experiments. As expected, before time instant 500, $P_0^0(\delta)$ converges to the neighborhood of 0, while $P_{59}^0(\delta)$ stabilizes around 1 which reflects that s_0 is identified as reliable while s_{59} is identified as unreliable. The opposite behavior takes place few time instants after the environment's change which occurs at time instant 500. Such behavior demonstrates the ability of our Tsetlin-like LA to adapt to a dynamic environment. In Fig. 2 and Fig. 3, we increase the number of states to the following respective values: 100 and 300. Interestingly, we observe a delay "in the adaptability". For instance, as we can see from Fig. 2, it takes around 100 time instants for the respective LA attached to s_0 and s_{59} to adapt to the optimal values of $P_0^0(\delta)$ and $P_{59}^0(\delta)$ after the swap that takes place at instant 500. Such delay was not noticed for the case of $2\delta = 10$ in Fig. 1. Loosely speaking, the higher the number of states per action, the longer time the LA takes to "forget" old decisions (actions) and converge to the new optimal decision.

We conclude that the advantage of using limited memory is higher adaptivity capability at the cost of negligible accuracy loss. From the experiments, it seems that a memory as small as 10 is sufficient for achieving high accuracy and in same time allows high adaptivity.

V. Conclusion

The question of identifying unreliable sensors without knowledge of the ground truth is an important and paradoxal research question [6], [7]. In this paper, we devise a novel LA solution to the latter problem. Our solution enjoys plausible properties compared to the-state-of-the-art initial solution reported in [6], [7] as it achieves high accuracy with low memory footprint while yielding excellent adaptivity in dynamic environments.

References

[1] W. Elmenreich, "Fusion of continuous-valued sensor measurements using confidence-weighted averaging," *Journal of Vibration and Control*, vol. 13, no. 9-10, pp. 1303–1312, 2007.

[2] M. A. Hossain, P. K. Atrey, and A. El Saddik, "Learning multisensor confidence using a reward-and-punishment mechanism," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 5, pp. 1525–1534, 2009.

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

[4] J. Frolik, M. Abdelrahman, and P. Kandasamy, "A confidence-based approach to the self-validation, fusion and reconstruction of quasi-redundant sensor data," *IEEE Transactions on Instrumentation and Measurement*, vol. 50, no. 6, pp. 1761–1769, 2001.

[5] R. Polikar, "Ensemble learning," in *Ensemble machine learning*. Springer, 2012, pp. 1–34.

[6] A. Yazidi, B. J. Oommen, and M. Goodwin, "On distinguishing between reliable and unreliable sensors without a knowledge of the ground truth," in *2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, vol. 2. IEEE, 2015, pp. 104–111.

[7] ——, "On solving the problem of identifying unreliable sensors without a knowledge of the ground truth: The case of stochastic environments," *Accepted in IEEE transactions on Cybernetics, 2016*.

[8] P. J. Boland, "Majority systems and the condorcet jury theorem," *The Statistician*, pp. 181–189, 1989.

[9] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*. New Jersey: Prentice-Hall, 1989.

[10] M. L. Tsetlin, *Automaton Theory and the Modeling of Biological Systems*. New York: Academic Press, 1973.

[11] K. S. Narendra and M. A. L. Thathachar, *Learning Automata: An Introduction*. New Jersey: Prentice-Hall, 1989.

[12] A. Yazidi, O.-C. Granmo, and B. J. Oommen, "Learning-automaton-based online discovery and tracking of spatiotemporal event patterns," *IEEE Transactions on Cybernetics*, vol. 43, no. 3, pp. 1118–1130, 2013.