Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

A Hierarchical Bayes Approach for Distributed Binary Classification in Cyber-Physical and Social Networks

Angelo Coluccia * **Giuseppe Notarstefano** *

* *Department of Engineering, Università del Salento via Monteroni, 73100 Lecce, Italy,*
{angelo.coluccia,giuseppe.notarstefano}@unisalento.it.

Abstract: In this paper we consider a network of agents that can evaluate each other according to an *interaction graph* modeling some physical interconnection or social relationship. Each agent provides a score for its (out-)neighboring agents in the interaction graph. The goal is to design a distributed protocol, run by the agents themselves, to group the network nodes into two classes (binary classification) on the basis of the evaluation outcomes. We propose a hierarchical Bayesian framework in which the agents' belonging to one of the two classes is assumed to be a probabilistic event with unknown parameter. Exploiting such a hierarchical framework, we are able to design a distributed classification scheme in which nodes cooperatively classify their own state. We characterize the solution for a fault-diagnosis context in cyber-physical systems, and for an opinion-classification/community-discovery setup in social networks.

1. INTRODUCTION

Cyber-physical systems as power-networks, smart-grids or distributed industrial plants are widely studied contexts in the distributed control area. A fundamental issue arising in such systems is fault diagnosis. Centralized schemes may be not well-suited in such large-scale plants, so that distributed schemes are sought. On a different context, the spreading of social networks in everyday life raises up a series of trust problems as well as opinion classification problems that are becoming more and more important for social users. These two apparently different scenarios share some common characteristics. That is, in both cases, nodes in the network may interact with neighboring ones, evaluate them, and give a score. The evaluation can be a physical diagnosis for cyber-physical systems or a rating for social networks. On the basis of these outcomes the nodes would like to decide on their binary state (functioning vs malfunctioning or pro vs against a community/opinion).

The problem of fault diagnosis has been deeply studied in wired computer networks and parallel systems. However, with the widespread of wireless communication, namely ad-hoc and sensor networks, more fitting and scalable approaches are needed. The distributed paradigm is a natural choice in this context, and several distributed protocols have been proposed (e.g., Chessa and Santi [2002], Lee and Choi [2008]). However, most fault detection algorithms are specific to the protocols implemented in the network and often require a heavy communication overhead to report state information. *Self-diagnosis* is a more appealing and scalable approach, since sensor nodes can infer their state by themselves on the basis of local information, e.g., Liu et al. [2011] and Ma et al. [2012]. The extension of methodologies for technological networks to study social networks has recently become subject of great interest. Typical problems of interest are the quantitative modeling of uncertainty and trust in peer-to-peer networks and recommendation systems Gyarmati and Trinh [2010], Theodorakopoulos and Baras [2006]. A recent survey on the connection between technological and social networks is Chen et al. [2013]. Binary classification in a distributed estimation framework has been proposed in Fagnani et al. [2012] and Chiuso et al. [2011], where consensus-based algorithms have been developed. A hypothesis testing approach for multi-agent decision systems is considered in Dandach et al. [2012], where a fusion center collects independent local decisions from a network of individuals.

The contribution of the present paper is twofold. First, we formulate a binary classification problem in a network context as a distributed estimation problem. Differently from the standard approach based on hypothesis tests, we propose a hierarchical Bayesian framework. This allows us to both take a binary decision on the node state and to obtain a soft, "gray-scale" classification revealing the confidence level of the taken decision. Such a confidence level can be seen as a trust of the node classification. Following the approach introduced in Coluccia and Notarstefano [2013], see also Coluccia and Notarstefano [2014], the main novel idea of the proposed framework is to capture global features of the network (common features of a system/society) by means of a common prior distribution whose parameters, called *hyperparameters*, are unknown and need to be estimated by the network agents. The resulting estimator has a hierarchical structure in which each node can self-classify once the hyperparameters have been estimated by mens of a suitable distributed algorithm. As a second contribution, we exploit the structure of the proposed distributed classifier when the evaluation outcome is itself a binary variable. In a first case study, we consider a fault-diagnosis scenario in which a node can test its neighbor and indicate if it is faulty or not. Since the reliability of the test outcome depends on the state of the testing node, a joint decision must be taken. In a second case study we analyze a social context in which individuals can rate each other by expressing an "I like" or not. For these two frameworks, we show that, in order to estimate the hyperparameter, an aggregated information is sufficient, namely the fraction of positive outcomes over the total number of evaluations. Then, we show that this aggregated information can be distributedly computed in a completely asynchronous and directed network by using a suitable version of a push-sum consensus algorithm. Given the aggregated information, each node can self-classify by computing locally the roots of a polynomial.

The paper is organized as follows. In Section 2 we set-up the binary classification problem in asynchronous networks. In Section 3 we introduce the proposed hierarchical Bayes framework and develop the distributed classifier for a general scenario. Then we analyze the two case-studies of fault-diagnosis in cyber-physical systems and community discovery in social networks. Finally, in Section 4 we provide a performance analysis showing the appealing features of the proposed distributed classifier.

2. THE BINARY CLASSIFICATION PROBLEM IN ASYNCHRONOUS NETWORKS

We consider a *network of agents* with the ability of performing an evaluation on some neighboring agents. The outcome of each evaluation is a score given by the evaluating agent on the evaluated one. The interaction is described by an *interaction graph*. Formally, we let $\{1,\dots,N\}$ be the set of agent identifiers and $G_I = (\{1,\dots,N\}, E_I)$ be a digraph such that $(i,j) \in E_I$ if agent i evaluates agent j. The set of in-neighbors of j in G_I, i.e. the set of nodes that evaluate j, is denoted by N_j^I. We assume that each node has at least one incoming edge in the interaction graph, that is there is at least one agent evaluating it.

Accordingly, let $\mathcal{E}$ be the set of evaluation outcomes ($\mathcal{E}$ can be, e.g., the set of reals, a real or natural interval, or the boolean space $\{0,1\}$). Then, for each agent in the network we associate the following variables:

- $s_i \in \{0,1\}$ is the unobservable *binary state* of agent i
- $t_{ij} \in \mathcal{E}$ is the *score* (evaluation outcome) of the evaluation performed by agent i on agent j

Besides the evaluation capability, the agents have also *communication* and *computation* functionalities. That is, agents can communicate (possibly) asynchronously according to a time-varying directed *communication graph*. Formally, we assume that the network evolution is triggered by a *universal slotted time*, $t \in \mathbb{Z}_{\geq 0}$, not necessarily known by the agents. The agents communicate according to a time-dependent directed communication graph $t \mapsto \mathcal{G}_C(t) = (\{1,\dots,N\}, E_C(t))$, where the edge set $E_C(t)$ describes the communication among agents: $(i,j) \in E_C(t)$ if agent i communicates to j at time $t \in \mathbb{Z}_{\geq 0}$. For each node i, the nodes sending information to i at time t, i.e., the set of $j \in \{1,\dots,N\}$ such that $(i,j) \in E_C(t)$ are called in-neighbors of i at time t. The set of in-neighbors of i at t is denoted by $N_i^C(t)$.

We assume that for each $(i,j) \in E_I$ there exists at least one time instant t such that $(i,j) \in E_C(t)$. Then, we make the following minimal assumption on the communication graph connectivity:

Assumption 2.1. (Uniform joint strong connectivity). There exists an integer $Q \geq 1$ such that the graph $\left(\{1,\dots,N\}, \bigcup_{\tau=tQ}^{(t+1)Q-1} E_C(\tau)\right)$ is strongly connected $\forall\, t \geq 0$.

It is worth remarking that this network setup is very general, since it naturally embeds asynchronous scenarios in which the communication is not necessarily symmetric.

3. DISTRIBUTED BINARY CLASSIFICATION VIA EMPIRICAL BAYES

3.1 Derivation of the Empirical Bayes classifier

We consider a probabilistic model for our classification scenario. That is, we assume that each evaluation outcome t_{ij}, $(i,j) \in E_I$, is determined according to a probability distribution, $p(t_{ij}|s_i,s_j)$, depending on the states of the evaluating and/or evaluated agents, respectively s_i and s_j.

According to the Bayesian framework, we consider the state s_i of a node i as a stochastic variable with a prior distribution $p(s_i)$. We let $p(s_i)$ be a Bernoulli's distribution $\mathcal{B}(1,\rho)$, i.e.,

$$p(s_i|\rho) = \rho^{s_i}(1-\rho)^{1-s_i},$$

where $\rho \in [0,1]$ is the Bernoulli's distribution parameter.

Since the assumption that the prior is known to all nodes would be too strong in a network scenario, we follow the Empirical Bayes approach and assume that the parameter ρ is unknown and needs to be estimated based on the evaluation outcomes. Then, the idea is to use an estimate of ρ to compute the posterior distribution.

Using the Bayes' Theorem the joint distribution of evaluations and states is obtained as

$$p(t_{ij},s_i,s_j|\rho) = p(t_{ij}|s_i,s_j)p(s_i|\rho)p(s_j|\rho). \tag{1}$$

Remembering that the states are binary variables, the probability of the evaluation is obtained by marginalizing (1) with respect to s_i and s_j,

$$p(t_{ij}|\rho) = \sum_{s_i\in\{0,1\}} \sum_{s_j\in\{0,1\}} p(t_{ij},s_i,s_j|\rho).$$

Using the independence of the evaluation outcomes, the Maximum Likelihood (ML) estimate of ρ can be obtained from the posterior distribution $p(t_{ij}|\rho)$ as

$$\hat{\rho} = \arg\max_{\rho} \prod_{(i,j)\in E_I} p(t_{ij}|\rho). \tag{2}$$

The estimate $\hat{\rho}$ can be used to approximate the posterior distribution $p(s_j|\boldsymbol{t}_{N_j^I},\rho)$, where $\boldsymbol{t}_{N_j^I}$ is the vector of evaluations performed on j, i.e. the vector of t_{ij} with $(i,j) \in E_I$.

Remembering that the t_{ij} are independent, we compute

$$p(\boldsymbol{t}_{N_j^I}, s_j|\hat{\rho}) = \prod_{i\in N_j^I} p(t_{ij},s_j|\hat{\rho}),$$

with $p(t_{ij},s_j|\hat{\rho})$ obtained by marginalizing (1) with respect to s_i, and

$$p(\boldsymbol{t}_{N_j^I}|\hat{\rho}) = \prod_{i\in N_j^I} p(t_{ij}|\hat{\rho}).$$

Then, using the Bayes' Theorem, the posterior distribution turns to be

$$p(s_j|\boldsymbol{t}_{N_j^I},\hat{\rho}) = \frac{p(\boldsymbol{t}_{N_j^I}, s_j|\hat{\rho})}{p(\boldsymbol{t}_{N_j^I}|\hat{\rho})}. \quad (3)$$

The state s_j with *maximum a-posteriori probability* (MAP), i.e., the one minimizing the classification error, is given by:

$$\hat{s}_j = \arg\max_{s_j\in\{0,1\}} p(s_j|\boldsymbol{t}_{N_j^I},\hat{\rho})$$

which means that node j will be classified as faulty if:

$$\frac{p(s_j = 1|\boldsymbol{t}_{N_j^I},\hat{\rho})}{p(s_j = 0|\boldsymbol{t}_{N_j^I},\hat{\rho})} > 1 \quad (4)$$

The MAP classifier obtained through equation (4) gives a binary decision. That is, it is a *hard classifier*.

The proposed approach allows us to also have a *soft classifier*. That is, we can classify the state with some "confidence" (a gray-scale classifier vs a black-white one). Indeed, to give a shade to this binary classification, the values $p(s_j = 1|\boldsymbol{t}_{N_j^I},\hat{\rho})$ and $p(s_j = 0|\boldsymbol{t}_{N_j^I},\hat{\rho})$ can be used to define a suitable score. A possible choice can be

$$S_j = \frac{p(s_j = 1|\boldsymbol{t}_{N_j^I},\hat{\rho})}{p(s_j = 1|\boldsymbol{t}_{N_j^I},\hat{\rho}) + p(s_j = 0|\boldsymbol{t}_{N_j^I},\hat{\rho})}, \quad (5)$$

where a score close to one means that the state $s_j = 1$ is very likely, and vice-versa.

We are now ready to design a distributed state classifier. The optimization problem (2) has a very special structure. Indeed, the cost function, after applying the logarithm, is separable. Thus, the problem can be solved in a distributed way by applying distributed optimization algorithms available in the literature. See, e.g., Zanella et al. [2012], Bürger et al. [2014], Nedic and Olshevsky [2013] for distributed optimization algorithms working on asynchronous networks. Once a copy of the ML estimate $\hat{\rho}$ is available at each node, it can self-classify by using the MAP hard classifier (4) or the soft one (5). In both cases the computation can be performed locally given the copy of $\hat{\rho}$.

3.2 Distributed binary classifier for fault-diagnosis

We consider a scenario in which each node can test neighboring nodes in the interaction graph with a binary outcome indicating if the tested node is considered faulty or not. Clearly, since each node performing the evaluation can be itself faulty, its outcome is not always reliable. Also, a node does not know if it itself is faulty or not. We assume that the evaluation outcome is determined according to the following rule. If node i is functioning, then it will return the correct status of the evaluated node j (i.e., $t_{ij} = 1$ if node j is faulty and $t_{ij} = 0$ if it is functioning correctly). If node i is faulty, we assume that the outcome is random with known probability a. In particular, if it is zero or one with the same probability then $a = 1/2$, otherwise a will be biased towards one of the two possible outcomes according to prior information about the node behaviors. This model can be regarded as a possible probabilistic extension of the well known Preparata et al. [1967] model.

Formally, the conditional probability of t_{ij} given the states s_i and s_j is

$$p(t_{ij}|s_i, s_j) = as_i + (1-s_i)\left[s_j t_{ij} + (1-s_j)(1-t_{ij})\right]. \quad (6)$$

The probability of t_{ij} can be obtained by marginalizing (1) with respect to s_i and s_j. Marginalizing with respect to s_i,

$$p(t_{ij}, s_j|\rho) = a\rho^{1+s_j}(1-\rho)^{1-s_j} + \\ + \left[s_j t_{ij} + (1-s_j)(1-t_{ij})\right]\rho^{s_j}(1-\rho)^{2-s_j}, \quad (7)$$

and then with respect to s_j,

$$p(t_{ij}|\rho) = a\rho + t_{ij}\rho(1-\rho) + (1-t_{ij})(1-\rho)^2. \quad (8)$$

With this expression in hand, we can exploit the structure of the MAP classifier and show how it can be computed by each node in a distributed way.

Let $k = \sum_{(i,j)\in E_I} t_{ij} = \sum_{j=1}^N k_j$ be the total number of positive outcomes, where $k_j = \sum_{i\in N_j^I} t_{ij}$ is the number of positive outcomes on agent j. Similarly, let $n = \sum_{j=1}^N |N_j^I| = \sum_{j=1}^N n_j$ be the total number of edges of the interaction graph G_I, where $n_j = |N_j^I|$ is the number of incoming edges of node j.

Substituting (8) in the ML optimization problem (2), we get

$$\hat{\rho} = \arg\max_{\rho\in[0,1]} \left[a\rho + \rho(1-\rho)\right]^k \left[a\rho + (1-\rho)^2\right]^{n-k}. \quad (9)$$

Defining $\phi = \frac{k}{n}$ and denoting $f(\rho; a, \phi)$ the cost function in problem (9), we can differentiate $f(\rho; a, \phi)$ with respect to ρ so that,

$$f'(\rho; a, \phi) = \\ n\left[a\rho + \rho(1-\rho)\right]^{k-1}\left[a\rho + (1-\rho)^2\right]^{n-k-1} g(\rho; a, \phi),$$

where

$$g(\rho; a, \phi) = \left[\phi(a + 1 - 2\rho)(a\rho + (1-\rho)^2) + \\ (1-\phi)(a\rho + \rho(1-\rho))(a - 2 + 2\rho)\right].$$

Clearly, the first two factors have roots that do not depend on ϕ and give a zero cost function. Thus maximizers are obtained by studying the sign of the function $g(\rho; a, \phi)$.

It is worth noting that each node can compute locally the maximizer $\hat{\rho}$, provided that the global quantity $\phi = \frac{k}{n}$ can be computed in a distributed way.

A closed form solution for $\hat{\rho}$ can be computed when $a = \frac{1}{2}$. The roots of $g(\rho; \frac{1}{2}, \phi)$ are $\rho_0 = \frac{3}{4}$, which is independent of ϕ, and $\rho_{1,2} = \frac{1}{4}(3 \pm \sqrt{9 - 16\phi})$. Straightforward computations show that ρ_1 and ρ_2, when real ($\phi \leq \frac{9}{16}$), are local maximizers. However, for $\phi \leq \frac{9}{16}$, $\rho_2 = \frac{1}{4}(3 + \sqrt{9 - 16\phi}) \geq 1$. Thus, we have

$$\hat{\rho} = \begin{cases} \dfrac{3}{4} & \text{if } \phi \geq \dfrac{9}{16} \\ \dfrac{1}{4}(3 - \sqrt{9 - 16\phi}) & \text{otherwise.} \end{cases}$$

With the expression of $\hat{\rho}$ in hand, we are able to compute the binary classifier. Plugging the expressions of $p(t_{ij}, s_j|\hat{\rho})$ and $p(t_{ij}|\hat{\rho})$ into (3) and evaluating it for $s_j = 1$, the fault probability turns to be

$$p(s_j = 1|\boldsymbol{t}_{N_j^I},\hat{\rho}) = \frac{\left[a\hat{\rho}^2 + \hat{\rho}(1-\hat{\rho})\right]^{k_j}\left[a\hat{\rho}^2\right]^{n_j-k_j}}{\left[a\hat{\rho} + \hat{\rho}(1-\hat{\rho})\right]^{k_j}\left[a\hat{\rho} + (1-\hat{\rho})^2\right]^{n_j-k_j}}.$$

An analogous expression can be obtained for $p(s_j = 0|\boldsymbol{t}_{N_j^I},\hat{\rho})$, so that, according to (4), node j self-classifies as faulty if

$$\frac{\left[a\hat{\rho}^2+\hat{\rho}(1-\hat{\rho})\right]^{k_j}\left[a\hat{\rho}^2\right]^{n_j-k_j}}{[a\hat{\rho}(1-\hat{\rho})]^{k_j}\left[a\hat{\rho}(1-\hat{\rho})+(1-\hat{\rho})^2\right]^{n_j-k_j}}>1.$$

We want to stress one more time that, provided that ϕ can be computed in a distributed way, agent j can obtain locally both $\hat{\rho}$ and the above classifier.

3.3 Distributed binary classifier for community discovery

We consider a social network scenario in which each agent evaluates its neighbors in the interaction graph with a binary outcome ("I like" or not). The binary state of each node indicates its belonging or not to a community, or its compliance with a given opinion. We imagine that the evaluation be influenced by the fact that the two nodes have or not the same state. In particular, we assume that, if the two nodes have the same state, the evaluation outcome will be $t_{ij}=1$ with a high probability a. Otherwise the outcome will be $t_{ij}=1$ with a low probability $b \ll a$. Formally, the conditional probability of t_{ij} given the states s_i and s_j is

$$p(t_{ij}|s_i,s_j)=[a^{t_{ij}}(1-a)^{1-t_{ij}}]|s_i-s_j|+ \\ +[b^{t_{ij}}(1-b)^{1-t_{ij}}](1-|s_i-s_j|). \quad (10)$$

We proceed along the same line as in the fault detection scenario. Plugging (10) into (1) and marginalizing with respect to s_i, after some straightforward calculations, one gets

$$p(t_{ij}|s_j,\rho)=\rho^{s_j}(1-\rho)^{1-s_j} \\ \left[a^{t_{ij}}(1-a)^{1-t_{ij}}(s_j(1-\rho)+(1-s_j)\rho)\right. \quad (11) \\ \left.+b^{t_{ij}}(1-b)^{1-t_{ij}}((1-s_j)(1-\rho)+s_j\rho)\right].$$

Marginalizing with respect to s_j,

$$p(t_{ij}|\rho)=2\rho(1-\rho)[a^{t_{ij}}(1-a)^{1-t_{ij}}]+ \\ +(1-2\rho+2\rho^2)[b^{t_{ij}}(1-b)^{1-t_{ij}}]. \quad (12)$$

In the next theorem we provide a closed form solution for $\hat{\rho}$, that each node can independently compute provided it is able to calculate (in a distributed way) the quantity $\phi=\frac{k}{n}$.

Theorem 3.1. Consider a network of agents performing a self-evaluation according to the network model in Section 2, with $t_{ij}\in\mathcal{E}=\{0,1\}$. Let the evaluation outcome be determined according to the conditional probability (6). Then the ML estimate of the hyperparameter ρ is given by:

$$\hat{\rho}=\begin{cases}\frac{1}{2} & \text{if } (1-\phi)\geq\dfrac{a+b}{2}\\ \frac{1}{2}\pm\frac{1}{2}\sqrt{\dfrac{a+b-2(1-\phi)}{a-b}} & \text{otherwise}\end{cases}$$

where the choice of the sign only depends on a common convention used to identify the state zero (or one). □

Proof. Using (12), the ML optimization problem (2) becomes

$$\hat{\rho}=\arg\max_{\rho\in[0,1]}\,[2\rho(1-\rho)(1-a)+(1-2\rho+2\rho^2)(1-b)]^k \\ [2\rho(1-\rho)a+(1-2\rho+2\rho^2)b]^{n-k}.$$

The derivative of the cost function is

$$f'(\rho;a,b,\phi)= \\ n\left[2\rho(1-\rho)(1-a)+(1-2\rho+2\rho^2)(1-b)\right]^{k-1} \\ \left[2\rho(1-\rho)a+(1-2\rho+2\rho^2)b\right]^{n-k-1}g(\rho;a,b,\phi), \quad (13)$$

where, after straightforward calculations,

$$g(\rho;a,b,\phi)= \\ n(a-b)(2-4\rho)[2(a-b)\rho^2-2(a-b)\rho+(1-\phi-b)].$$

The roots of $g(\rho;a,b,\phi)$ are $\rho_0=1/2$ and

$$\rho_{1,2}=\tfrac{1}{2}\pm\tfrac{1}{2}\sqrt{\frac{a+b-2(1-\phi)}{a-b}},$$

which are complex conjugate for $(1-\phi)\geq(a+b)/2$ and real otherwise.

From (13), the other roots of $f'(\rho;a,b,\phi)$ are

$$\rho_{3,4}=\tfrac{1}{2}\pm\tfrac{1}{2}\sqrt{\frac{a+b-2}{a-b}},$$

which are complex conjugate, and

$$\rho_{5,6}=\tfrac{1}{2}\pm\tfrac{1}{2}\sqrt{\frac{a+b}{a-b}},$$

which are real and have a multiplicity of $n-k-1$. Since the cost associated to these two roots is zero, they are clearly (global) minimizers. Also, since they do not depend on ϕ they do not affect the maximizers. Evaluating the sign of the function $g(\rho;a,b,\phi)$, it can be easily shown that both the two roots $\rho_{1,2}$ are local maximizers. Plugging these two roots in the cost function, we get the same cost $[2(1-a)\rho_1\rho_2+(1-b)(\rho_1^2+\rho_2^2)]^k[2a\rho_1\rho_2+b(\rho_1^2+\rho_2^2)]^{n-k}$, so that they are both global maximizers and the proof follows. □

3.4 Push-sum consensus for distributed computation of ϕ

We have shown that if the evaluation outcome is itself a binary value, the global information that the agents need to know to compute locally the classification is the rate $\phi=k/n$ of positive tests over the total. Next we show how, using a generalization of the push-sum consensus algorithm proposed in Bénézit et al. [2010], the nodes can compute such a quantity in a distributed way.

For each $t\in\mathbb{Z}_{\geq 0}$, each node $i\in\{1,\ldots,N\}$ stores in memory two local states $\delta_i(t)$ and $\eta_i(t)$, and an estimate $\phi_i(t)$ of ϕ. Let $w_{ij}(t)\in\mathbb{R}_+$ be a set of weights, satisfying $w_{ij}(t)\geq\gamma>0$ if $(i,j)\in E_C(t)$, $w_{ij}(t)=0$ if $(i,j)\notin E_C(t)$, and $\sum_{i=1}^N w_{ij}(t)=1$.

Initialization: $\delta_i(0)=k_i$, $\eta_i(0)=n_i$, $\phi_i(0)=k_i/n_i$.
Iterate:

$$\delta_i(t+1)=\sum_{j\in N_i^C(t)\cup\{i\}}w_{ij}(t)\delta_j(t) \\ \eta_i(t+1)=\sum_{j\in N_i^C(t)\cup\{i\}}w_{ij}(t)\eta_j(t) \\ \phi_i(t+1)=\frac{\delta_i(t+1)}{\eta_i(t+1)} \quad (14)$$

We want to point out some appealing features of the above push-sum algorithm and thus of the proposed distributed classifier.

(i) The weight matrix used in the protocol is *column stochastic* (thus it is not the usual consensus matrix which is row stochastic). For general directed graphs the weight matrix is not required to be doubly stochastic, since the agents do not need to reach consensus separately on k and n, but only on their ratio.
(ii) To run the above protocol, and thus compute $\phi = k/n$, the agents do not need to know any global network or tuning parameter (as, e.g., the number of nodes N or some step-size).
(iii) The agents do not need to know the (universal) time t. Thus, the algorithm is well-suited for a completely asynchronous implementation. Simply, when node i is not communicating, it is enough to assume that there are no edges (in the communication graph) going out nor coming in agent i (i.e., $N_i^C(t) = \emptyset$).

The next proposition shows the convergence properties of the algorithm.

Proposition 3.2. Assume that Assumption 2.1 holds. Then the distributed algorithm (14) reaches consensus on ϕ, i.e.,

$$\lim_{t\to\infty} \phi_i(t) = \phi \qquad \text{for all } i \in \{1, \dots, N\}.$$

Proof. The proof can be obtained by following the same steps as in Bénézit et al. [2010], hence it is omitted for the sake of conciseness. Notice that, weak ergodicity of the forward product of the (column stochastic) weight-matrices follows by Assumption 2.1. □

Remark 3.3. Assumption 2.1 could be relaxed in Proposition 3.2 if a convergence with probability one is allowed. In particular, the assumption in Bénézit et al. [2010] could be used instead. □

4. PERFORMANCE ANALYSIS

In this section we provide numerical results for the two case studies developed in the previous section. Simulations have been run according to the Monte Carlo method, with 1000 trials for each point.

4.1 Fault diagnosis

A network of $N = 20$ nodes has been considered, with $\rho = 0.2$ and $a = 1/2$. Two instances of the classification algorithm are shown in Fig. 1 for two different interaction graphs: a directed cycle with 20 additional edges randomly added (left), and a directed cycle with 40 additional edges randomly added (right). Numbers inside the nodes are the true states s_i. The decisions '1' of the hard classifier are depicted as a black circle on the node contour, while the score S_j of the soft classifier is depicted as a shade of gray from white ('0') to black ('1'). In the figure some classification errors are visible, either false positive (FP) (i.e., a decision $\hat{s}_i = 1$ when the true state is $s_i = 0$) or false negative (FN) (i.e., a decision $\hat{s}_i = 0$ when $s_i = 1$). Notice that as the number of interactions grows (graph on the right) more insightful information can be drawn from the coloring thanks to the soft classifier.

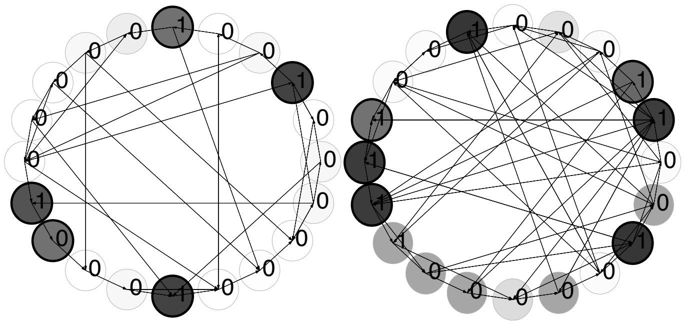

Fig. 1. Examples of hard (black circles for 1) and soft (gray-scale filling, from white 0 to black 1) classification for $\rho = 0.2$ (left: $n = N + 20$ edges; right: $n = N + 40$ edges). Numbers are the true state.

A closer look to the performance of the classifier is reported in Fig. 2, which shows the FP and FN rates for the same network of $N = 20$ nodes as function of the number of edges n. The interaction graph starts from a directed cyclic configuration, then additional edges are added until the graph is complete. The curves, obtained for $\rho = 0.2$, reveal very good performance with both FP and FN below 1% for moderate n. Notice the typical trade-off between FP and FN rates, with the latter increasing as the former decreases, before both converge to zero definitely.

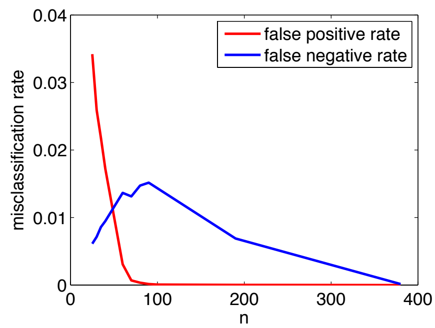

Fig. 2. False positive and false negative rates for $N = 20$, as function of the number of edges n increasing from N (cyclic graph) to $N^2 - N$ (complete graph).

Finally, we have assessed the power of the classifier for increasing values of the fault probability ρ. Simulation results, reported in Fig. 3, show that the classifier exhibits good performance even for ρ close to 0.5, i.e., with half of the nodes providing unreliable information. This is obtained with just one fourth of the possible tests, i.e., $n = \frac{1}{4}(N^2 - N) = 95$. For comparison, the case $n = N$ in cyclic topology is depicted with dashed lines: interestingly, even with this minimal connectivity in the interaction graph, the classifier performs satisfactorily until about 30% of faulty nodes.

4.2 Community discovery

Analogous Monte Carlo simulations have been run for the community discovery classification model. Due to the symmetry in the problem, we report the total misclassification error, i.e., the number of nodes that have associated themselves to the wrong community. The misclassification error for a network of $N = 100$ nodes as function of the

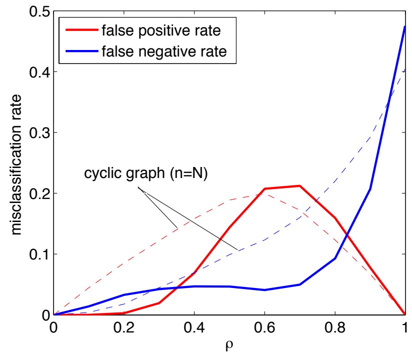

Fig. 3. False positive and false negative rates as function of the fault probability ρ, with number of edges n equal to 1/4 of the complete graph. For reference, the case of cyclic graph ($n = N$) is also shown in dashed lines.

number of edges n is reported in Fig. 4 for $\rho = 0.7$. We have set the probability of agreement between interacting nodes to 99% ($a = 0.99$) for nodes in the same community and to 1% ($b = 0.01$) for nodes in different communities. The picture shows that for a number of edges, n, greater than one fifth of the maximum number (complete graph), the misclassification rate is already below 5%.

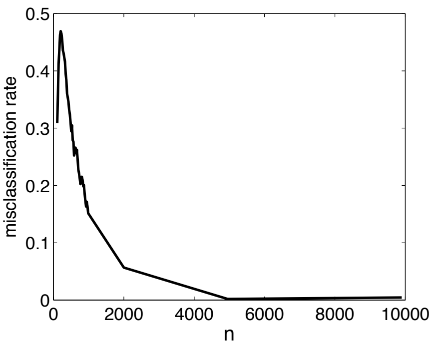

Fig. 4. Misclassification errors for $N = 100$, as function of the number of edges n increasing from N (cyclic graph) to $N^2 - N$ (complete graph).

5. CONCLUSION

In this paper we have proposed a novel distributed scheme, based on a hierarchical Bayes approach, for binary self-classification in cyber-physical and social networks. In the proposed framework agents gain information from the network by distributedly estimating a hyperparameter for the prior distribution of the node states. Based on the estimated hyperparameter each node can take a decision on its state. The estimation perspective allows us to develop, together with the hard decision maker, a soft classifier giving a level of "confidence" of the decision. We have presented two application scenarios, fault-diagnosis in cyber-physical systems and community discovery in social networks, for which the global information of the node binary classifier can be computed through a suitable consensus algorithm. Future research directions include the investigation of a dynamic scenario in which nodes repeatedly evaluate their neighbors and use soft decisions during the convergence process.

REFERENCES

Florence Bénézit, Vincent Blondel, Patrick Thiran, John Tsitsiklis, and Martin Vetterli. Weighted gossip: Distributed averaging using non-doubly stochastic matrices. In *IEEE International Symposium on Information Theory Proceedings*, pages 1753–1757. IEEE, 2010.

M. Bürger, G. Notarstefano, and F. Allgöwer. A polyhedral approximation framework for convex and robust distributed optimization. *"IEEE Transactions on Automatic Control"*, 59(2):384–395, Feb 2014.

Kwang-Cheng Chen, M. Chiang, and H.V. Poor. From technological networks to social networks. *Selected Areas in Communications, IEEE Journal on*, 31(9):548–572, 2013.

Stefano Chessa and Paolo Santi. Crash faults identification in wireless sensor networks. *Computer Communications*, 25, 2002.

Alessandro Chiuso, Fabio Fagnani, Luca Schenato, and Sandro Zampieri. Gossip algorithms for simultaneous distributed estimation and classification in sensor networks. *IEEE Journal of Selected Topics in Signal Processing*, 5(4):691–706, 2011.

A. Coluccia and G. Notarstefano. Distributed estimation of binary event probabilities via hierarchical bayes and dual decomposition. In *52nd IEEE Conference on Decision and Control*, 2013.

A. Coluccia and G. Notarstefano. Distributed bayesian estimation of arrival rates in asynchronous monitoring networks. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2014.

S.H. Dandach, R. Carli, and F. Bullo. Accuracy and decision time for sequential decision aggregation. *Proceedings of the IEEE*, 100(3):687–712, 2012.

Fabio Fagnani, Sophie Fosson, and Chiara Ravazzi. A large scale analysis of a classification algorithm over sensor networks. In *IEEE Conference on Decision and Control*, pages 4835–4839, 2012.

László Gyarmati and Tuan Anh Trinh. Measuring user behavior in online social networks. *IEEE Network*, 24 (5):26–31, 2010.

M.H. Lee and Y.H. Choi. Fault detection of wireless sensor networks. *Computer Communications*, 31, 2008.

K. Liu, q. Ma, X. Zhao, and Y. Liu. Self-diagnosis for large scale wireless sensor networks. In *Proc. of IEEE INFOCOM*, 2011.

Qiang Ma, Kebin Liu, Xin Miao, and Yunhao Liu. Sherlock is around: Detecting network failures with local evidence fusion. In *Proc. of IEEE INFOCOM*, 2012.

Angelia Nedic and Alex Olshevsky. Distributed optimization over time-varying directed graphs. In *IEEE Conference on Decision and Control*, 2013.

F.P. Preparata, G. Metze, and R.T. Chien. On the connection assignment problem of diagnosable systems. *IEEE Transactions on Computers*, EC-16:848–854, 1967.

George Theodorakopoulos and John S Baras. On trust models and trust evaluation metrics for ad hoc networks. *Selected Areas in Communications, IEEE Journal on*, 24 (2):318–328, 2006.

F. Zanella, D. Varagnolo, A. Cenedese, G. Pillonetto, and L. Schenato. Asynchronous newton-raphson consensus for distributed convex optimization. In *3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys'12)*, 2012.