Research Article

Distributed Event-Region Detection in Wireless Sensor Networks

Jun Fang and Hongbin Li

Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA

Correspondence should be addressed to Hongbin Li, hongbin.li@stevens.edu

We propose a graph-based method for distributed event-region detection in a wireless sensor network (WSN). The proposed method is developed by exploiting the fact that the true events at geographically neighboring sensors have a statistical dependency in an event-region detection scenario. This spatial dependence amongst the sensors is modeled using graphical models (GMs) and serves as a regularization term to enhance the detection accuracy. The method involves solving a linear system of equations, which can be readily implemented in a distributed fashion. Numerical results are presented to illustrate the performance of our proposed approach.

1. INTRODUCTION

With the emergence of low-cost and low-power sensors capable of limited computation and communication, the potential applications of WSNs for physical environment monitoring have become well appreciated and received much attention over the past few years [1–5]. In this paper, we focus on one particular class of environment surveillance problems: determining the event regions in an environment from the sensors' noisy observations. Such a problem arises in many scenarios. For example, as part of a building safety system, a WSN may be used to monitor hot spots and smoke. Also, using a WSN to sense the concentration of some chemical, we need to identify which regions have a chemical concentration greater than some threshold.

Consider a WSN composed of N geographically distributed sensor nodes, each sensor makes K noisy observations of its local signal values:

$$x_n(k) = \mu_n(\beta_n) + w_n(k), \quad k = 1, \ldots, K, \quad (1)$$

where x_n denotes the nth sensor's measurements, w_n denotes the zero mean independent and identically distributed (i.i.d.) Gaussian noise, β_n has a binary value with $\beta_n = 1$ indicating

event (signal) presence, and $\beta_n = 0$ indicating event (signal) absence at sensor n, and we have

$$\begin{aligned} \mu_n(0) &= 0, \quad \text{signal absence,} \\ \mu_n(1) &= \theta_n, \quad \theta_n \text{ is the unknown nonzero signal.} \end{aligned} \quad (2)$$

The above model allows for space-varying signal values, that is, θ_n can be dissimilar at different sensors. This corresponds to practical scenarios where the signal levels, such as the chemical concentration, vary across the event-region. The above formulation of event-region detection differs from the traditional distributed detection problem [3, 4] in two aspects. Firstly, the probability distributions of the sensor observations are usually assumed known a priori in [3, 4], while this is not the case for the event-region detection problem because the signal $\{\theta_n\}$ is generally unknown. Secondly, the objective of event-region detection is to identify the locations where event occurs in a sensor network environment. This is different from previous detection techniques [3, 4] that are developed for hypothesis testing of global phenomena.

A simple approach for event-region detection is to let the sensors make their decisions based only on their own measurements. This can be solved by the generalized likelihood

ratio test (GLRT). The local one-sided (suppose $\theta_n > 0$) GLRT at each sensor is given as [6, 7]

$$\hat{\beta}_n = 1 \quad \text{if } \frac{\sigma_{0,n}^2}{\sigma_{1,n}^2} I_{[0,\infty)}(\overline{x}_n) \geq \tau_{\text{GLRT}},$$

$$\hat{\beta}_n = 0 \quad \text{otherwise},$$

(3)

where $\overline{x}_n \triangleq (1/K)\sum_{k=1}^{K} x_n(k)$, $I_A(\overline{x}_n)$ is the indicator function whose value is equal to 1 if $\overline{x}_n \in A$ and 0 otherwise, and we have

$$\sigma_{0,n}^2 \triangleq \frac{1}{K}\sum_{k=1}^{K} x_n^2(k),$$

$$\sigma_{1,n}^2 \triangleq \sigma_{0,n}^2 - \overline{x}_n^2.$$

(4)

The threshold τ_{GLRT} can be determined based on a specified probability of false alarm P_{FA}, and such a choice of τ_{GLRT} is independent of $\{\theta_n\}$ [6, 7]. This approach, albeit simple, ignores the dependence among neighboring sensors. In practice, for a densely distributed sensor network environment, an event-region usually spans across an area which includes a certain number of sensors, and so does a nonevent region. Hence the true event indicator values, $\{\beta_n\}$, of neighboring sensors are statistically dependent. By utilizing this spatial dependence, it is expected that we can remove most of the sporadic decision errors (false alarms and misses) caused by the noise and faulty measurements of unreliable sensors.

Previous works on distributed event-region detection include [6, 8]. The work [6] models the distributed observations as a random field with a Markovian dependence structure and proposed an iterative method. Another work [8] introduced a Bayesian decision algorithm based on local decisions from neighboring sensors to identify the faulty measurements. It requires a precise knowledge of the sensor fault probability, which may not be available in practice.

In this paper, we use graphical models (GMs) to model the spatial dependence amongst the sensors. GM, like Markov random fields (MRFs), provides a natural framework to represent the statistical dependency amongst a set of variables by means of a graph [9]. It has been widely employed in WSN applications, for example, [10–14]. Since the true event indicators $\{\beta_n\}$, as mentioned previously in event-region detection scenarios, are locally dependent, they can be modeled by a locally connected GM, in which only spatially neighboring sensors are connected by *nonzero* weighted edges. This encoded spatial dependence by GM serves as a regularization term to smooth the local GLRT decisions such that the final decisions, to some extent, match the expectation that geographically adjacent sensors generally should have similar decisions. We formulate the event-region detection as an optimization problem which involves solving a linear system of equations. Because of the locally connected structure of the GM, solving the linear equations admits a simple distributed implementation by using iterative matrix inverse techniques such as the Richardson iteration. The resulting implementation scheme only requires that each sensor exchanges data within its neighbors and thus is energy and bandwidth efficient.

2. PROPOSED DISTRIBUTED EVENT-REGION DETECTION APPROACH

We model the WSN as an undirected graph $G = (V, E)$ whose vertices $V = \{1, 2, \ldots, N\}$ are the sensors and whose edges $E = \{e_{i,j}\}$ represent the connections between any two sensors. Each edge of the graph, joining vertices i and j, is assigned a weight $g_{i,j} = g_{j,i} \geq 0$ to measure the statistical dependency between these two sensors. To capture the statistical dependency amongst geographically adjacent sensors, we only set nonzero weights to the edges connecting neighboring vertices (sensors); otherwise they are set to zero. We choose $g_{i,j}$ as (see [15] for a detailed discussion on the construction of a weighted graph)

$$g_{i,j} = e^{-d_{i,j}^2/\phi} \quad \text{if } j \text{ is among } m\text{NN of } i \text{ or}$$
$$\qquad\qquad\qquad \text{if } i \text{ is among } m\text{NN of } j,$$
$$g_{i,j} = 0 \qquad\qquad \text{otherwise},$$

(5)

where $d_{i,j}$ denotes the Euclidean distance between vertices (sensors) i and j, mNN represents the m nearest neighbors in terms of Euclidean distance, ϕ and m are parameters of user choice that will be discussed later. We collect all the weights, $\{g_{i,j}\}$, and form an $N \times N$ symmetric weight matrix $\mathbf{G}$.

2.1. Graph-based decision-dependent regularization term

The statistical dependency amongst the neighboring sensors is measured by the weight matrix. It can serve as a regularization to update the initial estimates. We now discuss the construction of this regularization term. Consider a scalar function $\mathbf{f} \triangleq [f_1 \quad f_2 \quad \cdots \quad f_N]^T$ defined on the set of vertices $V = \{1, 2, \ldots, N\}$, where f_i corresponds to vertex i. A natural way to measure how much the vector $\mathbf{f}$ varies from our expected dependency amongst the neighboring vertices (sensors) is by the following quantity:

$$\sum_{i=1}^{N}\sum_{j=1}^{N} g_{i,j}(f_i - f_j)^2 \stackrel{(a)}{=} 2\mathbf{f}^T(\mathbf{D} - \mathbf{G})\mathbf{f}$$
$$= 2\mathbf{f}^T\mathbf{L}\mathbf{f},$$

(6)

where

$$\mathbf{D} \triangleq \text{diag}\left\{\sum_j g_{1,j}, \ldots, \sum_j g_{N,j}\right\};$$

(7)

(*a*) follows from the fact that

$$\sum_{i=1}^{N}\sum_{j=1}^{N} g_{i,j} f_j^2 = \mathbf{f}^T\mathbf{D}\mathbf{f},$$
$$\sum_{i=1}^{N}\sum_{j=1}^{N} g_{i,j} f_i f_j = \mathbf{f}^T\mathbf{G}\mathbf{f},$$

(8)

where $\mathbf{L} \triangleq \mathbf{D} - \mathbf{G}$ is the so-called graph Laplacian matrix [16]. It can be readily observed that $\mathbf{L}$ is symmetric positive semidefinite and it has one null eigenvalue associated

with the eigenvector $\mathbf{1}$, where $\mathbf{1}$ is the column vector with all unity elements. Clearly, the smaller value in (6), the better the vector $\mathbf{f}$ matches the statistical dependency amongst the neighboring sensors, and vice versa.

2.2. Hard decision regularization

We can use the regularization term defined in (6) to smooth the local GLRT decisions. Let $\hat{\boldsymbol{\beta}} \triangleq [\hat{\beta}_1, \ldots, \hat{\beta}_N]^T$ denote the local GLRT decisions via (3), $\hat{\boldsymbol{\beta}}_{\mathbf{r}} \triangleq [\hat{\beta}_{r,1}, \ldots, \hat{\beta}_{r,N}]^T$ denote the regularized decisions, then we can formulate the estimation of $\hat{\boldsymbol{\beta}}_{\mathbf{r}}$ as the following constrained optimization problem:

$$\min_{\hat{\boldsymbol{\beta}}_{\mathbf{r}}} \lambda \hat{\boldsymbol{\beta}}_{\mathbf{r}}^T \mathbf{L} \hat{\boldsymbol{\beta}}_{\mathbf{r}} + (\hat{\boldsymbol{\beta}}_{\mathbf{r}} - \hat{\boldsymbol{\beta}})^T (\hat{\boldsymbol{\beta}}_{\mathbf{r}} - \hat{\boldsymbol{\beta}})$$
$$\text{s.t. } \hat{\beta}_{r,n} \in \{0, 1\}, \quad \forall n \in \{1, \ldots, N\}, \tag{9}$$

where, as indicated before, the first term serves as the regularization term to account for the spatial dependence; λ is a positive coefficient controlling the participation degree whose choice will be discussed later; the second term represents the distance between the two vectors $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\beta}}_{\mathbf{r}}$, which should be minimized along with the regularization term. Clearly, this optimization is essentially a tradeoff between smoothing the decisions (to match our defined statistical dependency) and fitting the data. Note that the spatial smoothing effect can be easily observed from the fact that the regularization term has a minimal value, that is, zero, when the decisions at all sensors are identical, whatever they are ones or zeros. The optimization, therefore, penalizes isolated decisions that are different from their neighbors. Since decision errors (false alarms and misses) caused by noise and unreliable sensors usually occur in an independent and sporadic way, the optimization helps suppress false alarms and enhance event-region detection.

Note that the above constrained optimization problem is NP-hard. To make it tractable, we relax $\hat{\boldsymbol{\beta}}_{\mathbf{r}}$ to take on real values. The real-valued solution $\hat{\boldsymbol{\beta}}_{\mathbf{r}}$ can be obtained by solving the following equation:

$$(\lambda \mathbf{L} + \mathbf{I})\hat{\boldsymbol{\beta}}_{\mathbf{r}} = \hat{\boldsymbol{\beta}}, \tag{10}$$

where $\mathbf{I}$ denotes the identity matrix and

$$\hat{\boldsymbol{\beta}}_{\mathbf{r}} = (\lambda \mathbf{L} + \mathbf{I})^{-1} \hat{\boldsymbol{\beta}}. \tag{11}$$

This real-valued solution, obviously, will not satisfy the constraint $\hat{\beta}_{r,n} \in \{0, 1\}$. Nevertheless, a splitting point (also called threshold), τ_R, can be employed to transform this real-valued solution into a discrete form, that is,

$$\hat{\beta}_{r,n} = \begin{cases} 1 & \text{if } \hat{\beta}_{r,n} \geq \tau_R, \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

We will discuss the determination of τ_R in the later part of this paper.

2.3. Soft decision regularization

For the case where the noise is i.i.d. across all the network and the signal values inside the event-regions are constant, that is, $\theta_n = \theta, \forall \{n \mid \beta_n = 1\}$, the sample mean value, $\bar{x}_n \triangleq (1/K) \sum_{k=1}^{K} x_n(k)$, can be regarded as local soft decision at sensor n since a larger value of $\bar{x}_n$ indicates a higher possibility of event presence. Hence we can further extend our approach by replacing $\hat{\boldsymbol{\beta}}$ in (9) with the sample mean vector $\bar{\mathbf{x}} \triangleq [\bar{x}_1 \ \bar{x}_2 \ \cdots \ \bar{x}_N]^T$. To simplify the exposition, here we allow an abuse of notation $\hat{\boldsymbol{\beta}}_{\mathbf{r}}$ to denote the regularized soft decision vector which allows for real values. It can be solved as

$$\min_{\hat{\boldsymbol{\beta}}_{\mathbf{r}}} \lambda \hat{\boldsymbol{\beta}}_{\mathbf{r}}^T \mathbf{L} \hat{\boldsymbol{\beta}}_{\mathbf{r}} + (\hat{\boldsymbol{\beta}}_{\mathbf{r}} - \bar{\mathbf{x}})^T (\hat{\boldsymbol{\beta}}_{\mathbf{r}} - \bar{\mathbf{x}}), \tag{13}$$

and consequently,

$$(\lambda \mathbf{L} + \mathbf{I})\hat{\boldsymbol{\beta}}_{\mathbf{r}} = \bar{\mathbf{x}}. \tag{14}$$

The soft decision vector can also be transformed into hard decisions by using a threshold. As compared with the hard decision-based regularization, the soft decision-based method is able to provide a better performance since, for the hard decision case, some information about the observed data is lost after the 1-bit quantization, that is, computing the local GLRT decisions.

2.4. Choice of parameters

In this section, we discuss the choice of the parameters related to our proposed method. The parameters m and ϕ are used to quantify the statistical dependency among geographically adjacent sensors, in which m defines the degree to which the statistical dependency extends, and ϕ controls the values of the weights. Generally speaking, we can choose m from 1 to 4 according to the network topology. If the sensors are densely deployed in a 2D plane, a larger value such as $m = 3$ or $m = 4$ may capture the local statistical dependency better; if the sensors are placed along a line, then a smaller value such as $m = 1$ or $m = 2$ could be more appropriate. Such a choice of m indicates that one sensor is statistically correlated with its closest sensors, which is generally true for most event-region detection scenarios where the sensors are densely distributed. As for ϕ, it can be chosen to guarantee $a < g_{i,j} \leq 1$ for any nonzero $g_{i,j}$, where a can be set 0.5 or $1/e$. Simulation results show that our proposed method is not sensitive to m and ϕ as long as they are set in the above ranges.

The parameter λ controls the participation degree of the spatial smoothing term in the optimization. Clearly, a too small λ may not provide a sufficient involvement to suppress the false alarms. On the other hand, since $\mathbf{1}$ is the eigenvector of $\mathbf{L}$ associated with the smallest (zero) eigenvalue, a too large λ admits an excessive spatial smoothing effect that has the tendency to make the decisions homogeneous. Therefore, an appropriate λ is desirable to our method. Generally speaking, the choice of λ is dependent on the signal values, and the

noise variance, and so forth. In practice, we may obtain some coarse information about the signal, noise, and the event-region from a small subset of sensor observations, which can help us to determine an appropriate λ by a calibration procedure. The calibration procedure is also used in [6] for parameter choices. Generally speaking, a set of "training data" are randomly generated by simulating noisy realizations of a "calibration field." Note that the calibration field is not the true field to be detected but a field constructed from the coarse information about the signal, noise, and the event-region obtained from a small subset of sensor observations of the true field (see [6, Section VI.D] for details). Using the training data generated on the calibration field, we can determine an appropriate λ by minimizing the detection errors.

We discuss the determination of the threshold τ_R used to discretize the real-valued vector $\hat{\boldsymbol{\beta}}_{\mathbf{r}}$. Unlike the GLRT approach, since the real-valued vector $\hat{\boldsymbol{\beta}}_{\mathbf{r}}$ is obtained from $\hat{\boldsymbol{\beta}}_{\mathbf{r}} = (\lambda \mathbf{L} + \mathbf{I})^{-1}\hat{\boldsymbol{\beta}}$ or $\hat{\boldsymbol{\beta}}_{\mathbf{r}} = (\lambda \mathbf{L} + \mathbf{I})^{-1}\overline{\mathbf{x}}$, the resulting entries $\{\hat{\beta}_{r,n}\}$ are correlated and the joint distribution of $\{\hat{\beta}_{r,n}\}$ is dependent on the events. In this case, even if we have knowledge of the event-region and its related signal values, deriving an analytical expression for τ to satisfy a specified false alarm probability P_{FA} is difficult. In practice, training data can be used to help determine τ_R. Assume we have training data $\{\hat{\beta}_i, \beta_i\}_{i=1}^{N_{tr}}$, where $\hat{\beta}_i$ denotes the local GLRT decision and β_i corresponds to the true event indicator value of sensor i. We compute the corresponding regularized decision vector $\hat{\boldsymbol{\beta}}_{\mathbf{r}}$. With the knowledge of the true event indicator values, we can easily find the threshold τ_R to satisfy a specified false alarm probability on the training data. The above discussion applies to soft decision case by simply replacing the local GLRT decisions $\{\hat{\beta}_i\}$ with the sample mean $\{\overline{x}_i\}$. We note that using training data has its disadvantages, for example, the accuracy of τ_R is affected by the capability of the training data in capturing the true field. In practice, the training data should have a good representation of the true field, for example, the number of event-regions and their corresponding sizes.

3. DISTRIBUTED IMPLEMENTATION

We now discuss the distributed implementation of our proposed method. Considering WSNs with a fusion center (FC), we assume that the FC has knowledge of all sensors geographical locations by GPS or some other localization techniques. Therefore, it can compute the weight matrix $\mathbf{G}$ and consequently, the matrix $(\lambda \mathbf{L} + \mathbf{I})^{-1}$ in advance and store the computation results. We can have each sensor report its local GLRT decision $\{\hat{\beta}_n\}$, along with its sensor index n, directly to the FC. The FC then computes $\hat{\boldsymbol{\beta}}_{\mathbf{r}} = (\lambda \mathbf{L} + \mathbf{I})^{-1}\hat{\boldsymbol{\beta}}$, which can be turned into hard decisions by using the estimated threshold τ. However, this implementation scheme may be impractical for the soft decision-based approach (13) since it requires each sensor to send its real-valued data to the FC, which can be quite bandwidth- and power-consuming. Another feasible implementation, like [6, 8], is to let the sensors in the environment organize themselves and make decisions. This scheme is described as follows.

Note that both (10) and (14) involve the inverse of the sparse, positive definite matrix $\mathbf{A} \triangleq (\lambda \mathbf{L} + \mathbf{I})$. Hence iterative matrix techniques that are readily implemented in a distributed fashion can be used to compute the exact closed-form solution. Here we employ the modified Richardson iteration [18, 19] to solve the linear equations (10) and (14). The Richardson iteration for (10) and (14) is given by

$$\hat{\boldsymbol{\beta}}_{\mathbf{r}}^{(k+1)} = \hat{\boldsymbol{\beta}}_{\mathbf{r}}^{(k)} + \omega\left(\hat{\boldsymbol{\beta}} - \mathbf{A}\hat{\boldsymbol{\beta}}_{\mathbf{r}}^{(k)}\right),$$
$$\hat{\boldsymbol{\beta}}_{\mathbf{r}}^{(k+1)} = \hat{\boldsymbol{\beta}}_{\mathbf{r}}^{(k)} + \omega\left(\overline{\mathbf{x}} - \mathbf{A}\hat{\boldsymbol{\beta}}_{\mathbf{r}}^{(k)}\right),$$

(15)

respectively, where $\omega > 0$ is a parameter that has to be chosen such that $\omega < 2/\rho(\mathbf{A})$, and $\rho(\mathbf{A})$ denotes the spectral radius of $\mathbf{A}$. This iteration results in a sequence $\{\hat{\boldsymbol{\beta}}_{\mathbf{r}}^{(k)}\}$ that finally converges to the correct solution. The proof of the convergence of the Richardson iteration can be found in [18, 19] or on the website.[1] Note that for each row of $\mathbf{A}$, its nonzero entries only occur for those $j \in \mathcal{N}_i$, where $\mathcal{N}_i \triangleq \{j \mid j \text{ is among } m\text{NN of } i \text{ or } i \text{ is among } m\text{NN of } j\}$ denotes the neighborhood of sensor i. Thus the update equations at each sensor can be written as

$$\hat{\beta}_{r,i}^{(k+1)} = \hat{\beta}_{r,i}^{(k)} + \omega\left(\hat{\beta}_i - \sum_{j\in\mathcal{N}_i} a_{i,j}\hat{\beta}_{r,j}^{(k)}\right),$$
$$\hat{\beta}_{r,i}^{(k+1)} = \hat{\beta}_{r,i}^{(k)} + \omega\left(\overline{x}_i - \sum_{j\in\mathcal{N}_i} a_{i,j}\hat{\beta}_{r,j}^{(k)}\right),$$

(16)

respectively, where $a_{i,j}$ denotes the (i, j)th element of $\mathbf{A}$. From (16), we can see that at every iteration, each sensor only requires the data from its neighborhood for the update.

We summarize the implementation steps of this scheme as follows.

(1) For each sensor $i \in V$, we randomly generate an initial estimate $\hat{\beta}_{r,i}^{(0)}$.

(2) At iteration $k + 1$ ($k = 0, 1, \dots$), each sensor broadcasts its estimate $\hat{\beta}_{r,i}^{(k)}$, along with its sensor index i, to its neighborhood $\mathcal{N}_i$, and in the mean time, it collects the data from its neighborhood sensors; based on the received data, each sensor updates its estimate according to (16).

(3) Stop if some preset convergence condition is satisfied; otherwise go to Step 2.

(4) Each sensor makes its decision based on the final estimate $\hat{\beta}_{r,i}$ and the specified threshold τ_R.

As we can see, this implementation scheme is parallel, involves communication only among neighboring sensors, and therefore consumes minimal communication energy. This makes it applicable to WSN applications where power and communication are of concern. Moreover, the

---

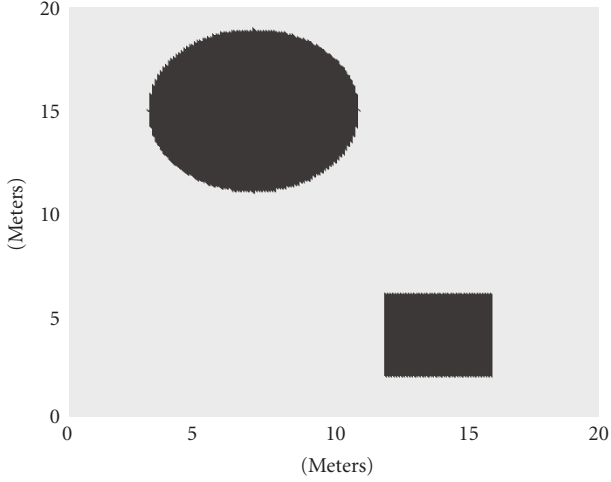[1] http://en.wikipedia.org/wiki/Modified_Richardson_iteration.
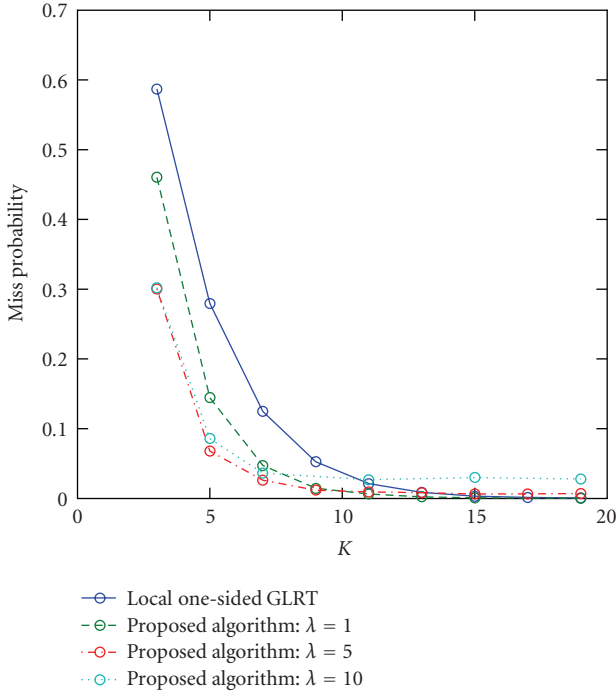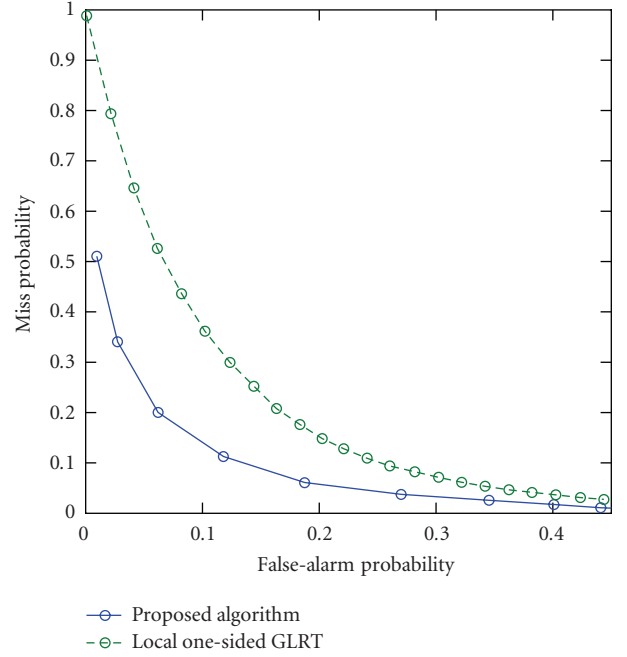
FIGURE 1: Noiseless field for hard decision case.

FIGURE 3: False alarm probability versus miss probability.

FIGURE 2: Miss probabilities versus K.
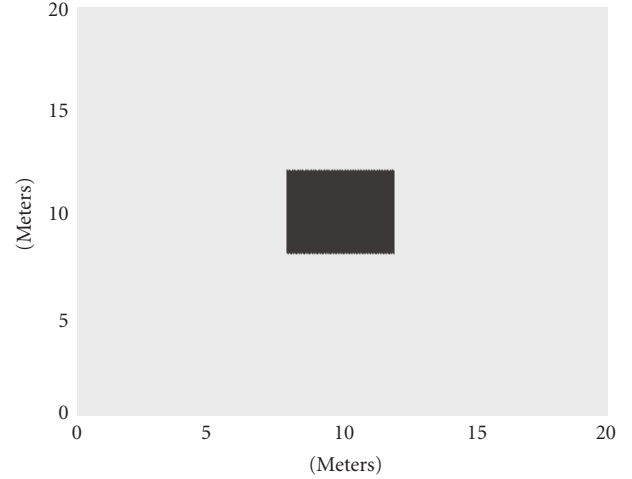
FIGURE 4: Noiseless field for soft decision case.

4. NUMERICAL RESULTS

convergence rate of the Richardson iteration can be controlled by the parameter ω. Specifically, it is determined by $\max_i(|1 - \omega\mu_i|)$, where $\{\mu_i\}$ are eigenvalues of $\mathbf{A}$. We can maximize the convergence rate by choosing $\omega \in (0, 2/\rho(\mathbf{A}))$ to minimize $\max_i(|1 - \omega\mu_i|)$. Besides the modified Richardson technique discussed here, there are some other iterative algorithms, for example, [12, 13], to solve the linear equation (14). These algorithms, as the modified Richardson iteration, admit distributed implementation and, furthermore, they may provide faster convergence rate and are more robust to the transmission errors amongst sensors.

We present simulation results to illustrate the performance of our proposed algorithm. Similarly as [6], we consider a WSN consisting of $N = 300$ sensors randomly distributed on a 20 m × 20 m grid with 1 m uniform spacing. For each sensor, it makes K local noisy observations: $\{x_n(k)\}_{k=1}^{K}$, the noise w_n is i.i.d. Gaussian distributed with zero mean and variance $\sigma_w^2 = 0.5$. The following results are obtained by simulating the distributed implementation scheme discussed in Section 3, in which we assume ideal data transmission amongst sensors. Experiments show that convergence can be achieved within a few tens of iterations. We note that although, in practice, the data transmission amongst sensors
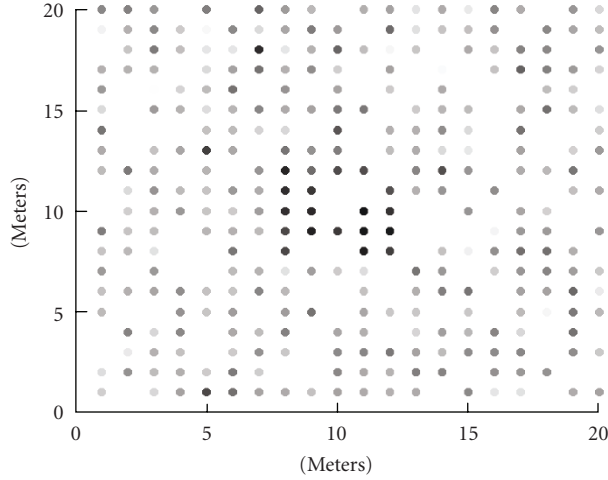
(a) Averaged observations, $\{\bar{x}_n\}$, as functions of the node locations
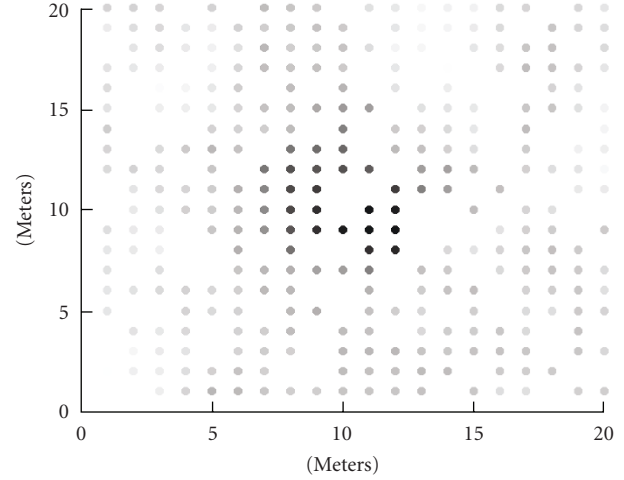
(b) Real-valued soft decisions, $\{\hat{\beta}_{r,n}\}$, as functions of the node locations

FIGURE 5: One realization of the averaged noisy observations, $\{\bar{x}_n\}$, and its associated real-valued soft decisions, $\{\hat{\beta}_{r,n}\}$.

suffers from errors because of data quantization and channel noise, more sophisticated distributed matrix techniques [12, 13], as mentioned in last section, can be employed to enhance the robustness to the transmission errors amongst sensors.

4.1. Results of hard decision regularization

We consider a field containing two event-regions as shown in Figure 1. We have $\mu_n(1) = 1$ for those sensors $\{n\}$ in the rectangle event-region, and $\mu_n(1) = 0.8$ for those sensors $\{n\}$ in the circular event-region. In our simulations, the parameters ϕ and m in (5) are set 2 and 4, respectively. The coefficient λ controlling the spatial smoothing effect is chosen to be 1, 5, and 10, respectively. The local GLRT decisions $\{\hat{\beta}_n\}$ are determined via (3), where the threshold τ_{GLRT} is chosen to satisfy that the false alarm probability is 0.05, that is, $P_{\text{FA}}^{\text{GLRT}} = 0.05$. From $\hat{\beta}$ we can compute a regularized decision vector $\hat{\beta}_{\mathbf{r}}$. The threshold τ_{R} used for binary decisions of $\hat{\beta}_{\mathbf{r}}$ is chosen such that $P_{\text{FA}}^{\text{R}} = P_{\text{FA}}^{\text{GLRT}} = 0.05$, where P_{FA}^{R} denotes the false alarm probability of the proposed regularization method. To overcome the difficulty we mentioned previously (see Section 2.4) in obtaining τ_{R}, we use the knowledge of the true event indicator $\{\beta_n\}$ to help determine τ_{R} to achieve the specified P_{FA}^{R}. Figure 2 shows the miss probabilities as functions of K for the local GLRT approach and for our proposed method under different choices of λ. The results are averaged over 500 independent runs. We observe that, as compared with the local GLRT approach, our proposed method is effective in reducing the miss probabilities under different choices of λ, especially when the number of observations K is small. We also see that an appropriate choice of λ should be related to the signal-to-noise ratio (SNR), it is favorable to choose a large λ for a low SNR while a small λ for a high SNR (note that a large K has the effect of improving SNR and vice versa).
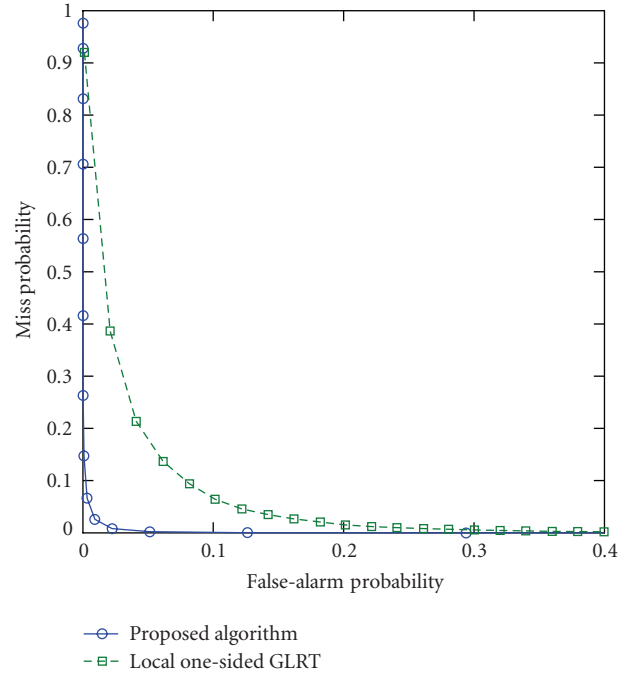

FIGURE 6: False alarm probability versus miss probability.

Let $\lambda = 1$ and $K = 3$, we plot the miss probability versus the false alarm probability in Figure 3, where each point on the curves corresponds to a value of $(P_{\text{FA}}, P_{\text{M}})$ for a given threshold. Note that, when plotting the figure, since our method does not provide an explicit expression in determining a threshold to obtain a specified P_{FA}, we just choose a set of thresholds and compute the $(P_{\text{FA}}, P_{\text{M}})$ associated with each threshold. From the figure, we see that our proposed algorithm presents a clear performance advantage over the GLRT.

4.2. *Results of soft decision regularization*

We now consider the soft decision version of our proposed algorithm associated with the optimization (13). The field we test here contains a rectangle event-region with $\mu_n(1) = 1$; see Figure 4. We set $\lambda = 5$ and $K = 5$. Figure 5 shows one realization of the averaged noisy observations $\{\bar{x}_n\}$ and its corresponding real-valued soft decisions $\{\hat{\beta}_{rn}\}$ as functions of the sensor locations, where $\{\bar{x}_n\}$ and $\{\hat{\beta}_{rn}\}$ are proportionally scaled to $[0, 1]$, respectively, and we use the grey levels to linearly represent the magnitudes of the scaled values (the larger the value, the darker the point). It can be clearly seen that the potential sporadic false alarms have successfully been suppressed, whereas the event-region is intensified. To further investigate the performance, we plot the miss probability versus false alarm probability in Figure 6. For our method, as we did for the hard decision case, we choose a set of thresholds and compute the (P_{FA}, P_M) associated with every threshold. We see that our method presents a clear performance advantage over the GLRT.

5. CONCLUSION

We have proposed a new method for distributed event-region detection in WSNs, where the spatial dependence amongst neighboring sensors is modeled using the GMs and serves as a regularization. The method admits an energy and bandwidth efficient distributed implementation. Numerical simulation results show that our proposed method presents a clear performance advantage over the local GLRT and is effective in improving detection accuracy.

ACKNOWLEDGMENT

REFERENCES

[1] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed, "Detection, classification, and tracking of targets," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 17–29, 2002.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[3] R. Viswanathan and P. K. Varshney, "Distributed detection with multiple sensors—part I: fundamentals," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 54–63, 1997.

[4] R. S. Blum, S. A. Kassam, and H. V. Poor, "Distributed detection with multiple sensors—part II: advanced topics," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 64–79, 1997.

[5] A. Ribeiro and G. B. Giannakis, "Bandwidth-constrained distributed estimation for wireless sensor networks—part I: Gaussian case," *IEEE Transactions on Signal Processing*, vol. 54, no. 3, pp. 1131–1143, 2006.

[6] A. Dogandžić and B. Zhang, "Distributed estimation and detection for sensor networks using hidden Markov random field models," *IEEE Transactions on Signal Processing*, vol. 54, no. 8, pp. 3200–3215, 2006.

[7] S. M. Kay, *Fundamentals of Statistical Signal Processing: Detection Theory*, Prentice-Hall, Upper Saddle River, NJ, USA, 1998.

[8] B. Krishnamachari and S. Iyengar, "Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks," *IEEE Transactions on Computers*, vol. 53, no. 3, pp. 241–250, 2004.

[9] S. L. Lauritzen, *Graphical Models*, Oxford University Press, Oxford, UK, 1996.

[10] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508–2530, 2006.

[11] S. Kar and J. M. F. Moura, "Ramanujan topologies for decision making in sensor networks," in *Proceedings of the 44th Annual Allerton Conference on Communication, Control and Computing*, Monticello, Ill, USA, September 2006.

[12] E. B. Sudderth, M. J. Wainwright, and A. S. Willsky, "Embedded trees: estimation of Gaussian processes on graphs with cycles," *IEEE Transactions on Signal Processing*, vol. 52, no. 11, pp. 3136–3150, 2004.

[13] V. Delouille, R. Neelamani, and R. G. Baraniuk, "Robust distributed estimation using the embedded subgraphs algorithm," *IEEE Transactions on Signal Processing*, vol. 54, no. 8, pp. 2998–3010, 2006.

[14] D. S. Scherber and H. C. Papadopoulos, "Distributed computation of averages over ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 776–787, 2005.

[15] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[16] F. R. K. Chung, *Spectral Graph Theory*, vol. 92 of *CBMS Regional Conference Series in Mathematics*, American Mathematical Society, Providence, RI, USA, 1997.

[17] M. Belkin and P. Niyogi, "Using manifold structure for partially labeled classification," in *Proceedings of the 15th Annual Conference on Neural Information Processing Systems (NIPS '03)*, pp. 929–936, MIT Press, Vancouver, Canada, 2003.

[18] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, Pa, USA, 1997.

[19] D. M. Young, *Iterative Solution of Large Linear Systems*, Academic Press, New York, NY, USA, 1971.