We are IntechOpen, the world’s leading publisher of Open Access books
Built by scientists, for scientists

6,600
Open access books available

177,000
International authors and editors

195M
Downloads

154
Countries delivered to

TOP 1%
Our authors are among the most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™
Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected. For more information visit www.intechopen.com
Chapter
Decision Fusion for Large-Scale Sensor Networks with Nonideal Channels
Yiwei Liao, Xiaojing Shen, Junfeng Wang and Yunmin Zhu

Abstract

Since there has been an increasing interest in the areas of Internet of Things (IoT) and artificial intelligence that often deals with a large number of sensors, this chapter investigates the decision fusion problem for large-scale sensor networks. Due to unavoidable transmission channel interference, we consider sensor networks with nonideal channels that are prone to errors. When the fusion rule is fixed, we present the necessary condition for the optimal sensor rules that minimize the Monte Carlo cost function. For the $K$-out-of-$L$ fusion rule chosen very often in practice, we analytically derive the optimal sensor rules. For general fusion rules, a Monte Carlo Gauss-Seidel optimization algorithm is developed to search for the optimal sensor rules. The complexity of the new algorithm is of the order of $O(LN)$ compared with $O(LN^2)$ of the previous algorithm that was based on Riemann sum approximation, where $L$ is the number of sensors and $N$ is the number of samples. Thus, the proposed method allows us to design the decision fusion rule for large-scale sensor networks. Moreover, the algorithm is generalized to simultaneously search for the optimal sensor rules and the optimal fusion rule. Finally, numerical examples show the effectiveness of the new algorithms for large-scale sensor networks with nonideal channels.

Keywords: decision fusion, multisensor detection, nonideal channels, Monte Carlo method, importance sampling

1. Introduction

Distributed detection has been an active research area in the past decades [1–7]. It involves the design of decision rules for the sensors\(^1\) and the fusion rule [8]. Early work on distributed detection mainly focused on conditionally independent sensor observations, such as [2, 4, 9, 10], and the resulting optimal sensor decision rules, as well as the fusion rule, were likelihood ratio tests (LRTs). Details on distributed detection with conditionally independent sensor observations can be seen in [1, 6, 7] and references therein.

\(^1\) In the rest of the paper, the term “sensor rules” refers to the “decision rules at the sensors.”
In this chapter, we focus on conditionally dependent observations in sensor networks. In [5], the computational difficulty of obtaining the optimal sensor rules was shown by a rigorous mathematical approach. Some early progress was made on the derivation of sensor rules for the dependent observation case such as in [11–15]. More recently, a hierarchical conditional independence model was provided that was applicable to some specific classes of multisensor detection problems with dependent observations [16]. Copula-based distributed decision fusion methods have been proposed to deal with dependent observations in sensor networks, such as [17–19] and references therein. Given a fusion rule, Monte Carlo methods were proposed to reduce the computational complexity of deriving sensor decision rules with ideal channels in [20, 21], and the optimal sensor rules were obtained analytically for the $K$-out-of-$L$ fusion rule in [20].

Some works on the derivation of optimal fusion rules can be seen in [15, 22–24]. For some specific parallel network decision systems, a unified fusion rule was presented in [15]. Some further results on the problem are available in [25, 26]. In [27], the authors provided methods that search for the sensor rules and the fusion rule simultaneously by combining the methods of [2] and [15] in order to attain near-optimal system performance.

The works discussed thus far assumed the availability of ideal channels in sensor networks. However, channel errors between the sensors and the fusion center are omnipresent in practical multisensor detection networks, and, therefore, studies on multisensor detection in the presence of nonideal channels have attracted some recent interest, such as in [8, 28–33]. Under the Neyman-Pearson criterion, the design of sensor rules in the presence of nonideal channels was addressed in [32]. The parallel fusion structure was extended by incorporating the fading channel layer and two alternative fusion schemes were presented based on fixed sensor rules in [28]. It was shown that the optimal sensor decision rule that minimizes the error probability at the fusion center is equivalent to a local LRT for independent sensor observations in [28]. Under Neyman-Pearson and Bayesian criteria, the work was generalized to dependent and noisy channels, respectively, in [8]. In [31], the authors considered the optimal sensor rules with channel errors via Riemann sum approximation under a given fusion rule for general dependent sensor observations. Although the method based on the Riemann sum approximation has been developed for dependent observations with channel errors, it is too computationally expensive to be of practical use in large-scale sensor networks.

In this chapter, a Monte Carlo importance sampling method is provided to reduce the computational complexity of multisensor detection fusion with channel errors. Based on the strong law of large numbers, the Bayesian cost function is approximated by its empirical average through the Monte Carlo importance sampling method. The main contributions of this chapter are listed below:

1. When the fusion rule is fixed, we derive a necessary condition for the optimal sensor rules that minimize the approximated Bayesian cost function. A Monte Carlo Gauss-Seidel optimization algorithm is developed and it is shown to be finitely convergent. The complexity of the new algorithm is shown to be of the order of $O(LN)$ compared with $O(LN^2)$ of the previous algorithm based on the Riemann sum approximation.

2. When the fusion rule is the $K$-out-of-$L$ rule, we prove that there exists an analytical form for the optimal sensor rules in the presence of nonideal channels. Thus, the proposed method allows us to design decision rules for large-scale sensor networks.
3. The Monte Carlo Gauss-Seidel optimization algorithm is extended to simultaneously search for the optimal sensor rules and the optimal fusion rule. Numerical examples show the effectiveness of the new algorithms for large-scale sensor networks with dependent observations and channel errors.

The rest of this chapter is organized as follows: In Section 2, the parallel binary Bayesian detection network with channel errors is formulated and the Monte Carlo cost function is introduced. In Section 3, the necessary condition for the optimal sensor rules is presented. For the \( K \)-out-of-\( L \) fusion rule, the analytical form for the optimal sensor rules is provided. In Section 4, the Monte Carlo Gauss-Seidel iterative algorithm and its convergence analysis are presented. The extension to search for the optimal sensor rules and the optimal fusion rule are simultaneously described in Section 5. Simulation results are provided in Section 6. Conclusions are contained in Section 7.

2. Preliminaries

2.1 Problem formulation

The \( L \)-sensor parallel Bayesian detection network structure with two hypotheses \( H_0 \) and \( H_1 \) in the presence of nonideal channels is considered (see Figure 1). Assume that \( y_1, y_2, \ldots, y_L \) are sensor observations and the \( j \)th sensor compresses the \( n_j \)-dimension vector observation \( y_j \) to one bit: \( I_j(y_j) : \mathbb{R}^{n_j} \rightarrow \{0, 1\} \). For notational convenience, \( n_j = 1 \) in the following description. The \( L \) sensors transmit the compressed data to the fusion center and the fusion center makes the decision between \( H_0 \) and \( H_1 \). Since external interference and internal errors may occur, the channels are not reliable and the fusion center may not correctly receive the symbol \( I_j \) sent by the \( j \)th sensor. Let \( I_j^0 \) denote the received bit by the fusion center for \( j = 1, 2, \ldots, L \). Generally speaking, \( I_j^0 \) may not be equal to \( I_j \). The definition and assumptions on channel errors (see e.g., [29, 31]) are summarized below:

Definition 1: The channel errors between the \( j \)th sensor and the fusion center are described as \( P_{j,1}^{ce} = P(I_j^0 = 0 | I_j = 1) \) and \( P_{j,0}^{ce} = P(I_j^0 = 1 | I_j = 0) \) for \( j = 1, 2, \ldots, L \), where \( P_{j,1}^{ce} \) is the probability of channel error when the \( j \)th sensor sends 1 but the fusion center receives 0, and \( P_{j,0}^{ce} \) is the probability of channel error when the \( j \)th sensor sends 0 but the fusion center receives 1.

![Figure 1](image)

*Figure 1.* The \( L \)-sensor parallel binary Bayesian detection network structure in the presence of nonideal channels.
Assumption 1: The probabilities of channel error are statistically independent of the hypotheses, namely $P(I^0_j|H_j, H_v) = P(I^0_j|H_j)$, $v = 0, 1$.

Remark 1: Assumption 1 is due to the hierarchical structure based on the Markov property (see [29]).

Assumption 2: The channels that connect the sensors to the fusion center are independent, i.e., $P(I^0_1, I^0_2, ..., I^0_L| I_1, I_2, ..., I_L) = \prod_{l=1}^L P(I^0_l|I_l)$.

We consider the parallel binary Bayesian detection network with nonideal channels that is built on the above definition and assumptions. The final decision is made by the fusion center based on the received binary bits $(I^0_1, I^0_2, ..., I^0_L)$ from the $L$ sensors. From the definition of a general Bayesian cost function given in [25], the $L$-sensor binary Bayesian cost function with channel errors at the fusion center can be written as follows:

$$C(I^0_1(y_1), ..., I^0_L(y_L); F^0) = C_{00}P_0P(F^0 = 0|H_0) + C_{01}P_1P(F^0 = 0|H_1) + C_{10}P_0P(F^0 = 1|H_0) + C_{11}P_1P(F^0 = 1|H_1)$$

$$= c + aP(F^0 = 0|H_1) - bP(F^0 = 0|H_0),$$

where $C_{00}, a, b = 0, 1$ are suitable cost coefficients, $P_0$ and $P_1$ are the prior probabilities for the hypotheses $H_0$ and $H_1$, respectively, $F^0$ is the fusion rule, and $P(F^0 = \mu|H_\nu)$, $\mu, \nu = 0, 1$ denotes the conditional probability of the event that the fusion center decides in favor of hypothesis $\mu$ when the real hypothesis is $H_\nu$. The cost function (1) is simplified to (2) by defining $c = C_{10}P_0 + C_{11}P_1$, $a = P_1(C_{01} - C_{11})$, $b = P_0(C_{10} - C_{00})$. $F^0$ is actually a function of the disjoint set of all possible binary messages $(I^0_1, I^0_2, ..., I^0_L)$. The received decisions are divided into two sets denoted as $H^0_0$ and $H^0_1$ which are given by

$$H^0_0 = \left\{ (u^0_1, u^0_2, ..., u^0_L) : F^0((I^0_1, I^0_2, ..., I^0_L)) = 0, I^0_j = u^0_j, u^0_j = 0/1, j = 1, ..., L \right\};$$
$$H^0_1 = \left\{ (u^0_1, u^0_2, ..., u^0_L) : F^0((I^0_1, I^0_2, ..., I^0_L)) = 1, I^0_j = u^0_j, u^0_j = 0/1, j = 1, ..., L \right\}.$$

Obviously, $H^0 = \left\{ (u^0_1, u^0_2, ..., u^0_L) : I^0_j = u^0_j, u^0_j = 0/1, j = 1, ..., L \right\} = H^0_0 \cup H^0_1$.

For any binary decisions $(I^0_1, I^0_2, ..., I^0_L)$ received by the fusion center, the original sensor decision bits before transmission are $(I_1, I_2, ..., I_L)$ and they consist of the set $H = \{ (u_1, u_2, ..., u_L) : I_j = u_j, u_j = 0/1, j = 1, ..., L \}$. Therefore, based on the law of total probability, the conditional probability formula, and Assumption 1:

$$P(F^0 = 0|H_\nu) = \sum_{D^0 \in H_\nu} P(D^0|H_\nu) = \sum_{D^0 \in H_\nu} \sum_{D \in H} P(D^0|D)P(D|H_\nu),$$

where $D^0 = (I^0_1, I^0_2, ..., I^0_L)$, $s^0 = (s^0(1), ..., s^0(L))$, $I^0_j = s^0(j)$, and $s^0(j) = 0/1$ is a specific value of $I^0_j$; in the same way, $D = (I_1, I_2, ..., I_L)$, $s = (s(1), ..., s(L))$, $I_j = s(j)$, and $s(j) = 0/1$ is a specific value of $I_j$. Strictly speaking, we should use $P(D^0 = s^0|H_\nu)$ to represent $P(D^0|H_\nu)$ and we use the latter for notational simplicity. It is similar to $P(D|H_\nu)$. Based on Assumption 2:
\[ P(D^0|D) = \prod_{j=1}^{L} P\left(y_j^0 | I_j\right), \]

where for any \(1 \leq j \leq L\)

\[ P\left(y_j^0 | I_j\right) = (1 - P^{e_0}_j)(1 - f_j^0)(1 - I_j) + P^{e_0}_j f_j^0 (1 - I_j) + \left(1 - P^{e_1}_j\right) f_j^0 I_j + P^{e_1}_j (1 - f_j^0) I_j. \]

Thus, the cost function (2) becomes

\[ C(I_1(y_1), \ldots, I_L(y_L); P^0) = c + \sum_{\rho \in H^0} \sum_{\rho' \in H} P(D^0|D)[aP(D|H_1) - bP(D|H_0)] \]

\[ = C(I_1(y_1), \ldots, I_L(y_L); P^0, P^{e_0}, P^{e_1}), \]

where \(P^{e_0} = (P^{e_0}_1, \ldots, P^{e_0}_L), P^{e_1} = (P^{e_1}_1, \ldots, P^{e_1}_L).\) Hence, the cost function now becomes a function of the sensor rules \((I_1, \ldots, I_L),\) the probabilities of channel errors \(P^{e_0}, P^{e_1},\) and the fusion rule \(P^0.\) The goal of this chapter is to optimize the sensor rules and the fusion rule so as to minimize the cost function with known probabilities of channel errors.

We rewrite \(aP(D|H_1) - bP(D|H_0)\) as follows:

\[ aP(D|H_1) - bP(D|H_0) = \int_{\Omega} a p_{y_1, \ldots, y_L} |H_1 - b p_{y_1, \ldots, y_L} |H_0 dy_1 \cdots dy_L \]

\[ = \int_{\Omega} [a p_{y_1, \ldots, y_L} |H_1 - b p_{y_1, \ldots, y_L} |H_0 dy_1 \cdots dy_L, \]

where \(\Omega = \{(y_1, \ldots, y_L): I_1(y_1) = s(1), \ldots, I_L(y_L) = s(L)\}, I_\Omega\) is an indicator function on \(\Omega,\) and the region of integration in (8) is the full space. Assume that \(p(y_1, y_2, \ldots, y_L | H_v), v = 0, 1 (or p(y | H_v))\) are the known conditional joint probability density functions. If not, we can learn the joint probability density functions from training data using copula functions (see, e.g., [17]). Note that \(I_1(y_1), \ldots, I_L(y_L)\) are indicator functions and \(s(j) = 0/1, j = 1, \ldots, L,\)

\[ I_{\Omega} = I\{y_1, \ldots, y_L\} = \left(\begin{array}{c} I_1(y_1) = s(1) \ldots I_L(y_L) = s(L) \end{array}\right) \]

\[ = \left(\begin{array}{c} I_{\Omega_1}(y_1) \ldots I_{\Omega_L}(y_L) \end{array}\right) \]

\[ = [(1 - I_1)(1 - s(1)) + I s(1)] \cdots [(1 - I_L)(1 - s(L)) + I_{L}s(L)]. \]

For simplicity, denote \(Q_j(I_j) = (1 - I_j)(1 - s(j)) + I s(j).\) Substituting (8) into (6),

\[ C(I_1(y_1), \ldots, I_L(y_L); P^0, P^{e_0}, P^{e_1}) \]

\[ = c + \frac{\sum_{\rho \in H^0} \sum_{\rho' \in H} P(D^0|D)[aP(D|H_1) - bP(D|H_0)]}{\int_{\Omega} a p_{y_1, \ldots, y_L} |H_1 - b p_{y_1, \ldots, y_L} |H_0 dy} = c + \int_{H^0} L(y) dy, \]

where \(P_{H_0} = \sum_{\rho \in H^0} \sum_{\rho' \in H} P(D^0|D)Q_1(I_1) \cdots Q_L(I_L)\) and \(L(y) = a p_{y | H_1} - b p_{y | H_0}.\) Note that from the definition of \(H^0_0, H^0_1, H^0,\) and \(H,\) we have
An essential difficulty of the Bayesian cost function (10) is the required high dimensional integration when dealing with large-scale sensor networks. Monte Carlo importance sampling is an attractive method to deal with this problem. In this subsection, we approximate the Bayesian cost function (10) by the Monte Carlo importance sampling method (see, e.g., [34, 35]). According to (10),

\[ C(I_1(y_1), ..., I_L(y_L); F^0, P^{\text{ref}}, P^{\text{ref1}}) = c + \sum_{I^0 \in H^0} P_{I^0}(I_L(y_L); F^0, P^{\text{ref}}, P^{\text{ref1}}) \tilde{L}(y) g(y) dy = \mathbb{E}_g \frac{P_{I^0}(Y(Y); F^0, P^{\text{ref}}, P^{\text{ref1}})}{g(Y)} + c, \]  

(13)

where \( y = (y_1, y_2, ..., y_L) \), and \( g(y) \) is a given importance sampling density such that (12) is well-defined (i.e., \( g(y) > 0 \)). In (13), the expectation is taken with respect to the importance sampling density \( g \). Consequently, assume that \( N \) samples \( Y_1, ..., Y_N \) are generated from the density \( g \), that is, \( Y \sim g(y) \), where \( Y_i = [Y_{i1}, Y_{i2}, ..., Y_{iL}] \). Then

\[ C(I_1(y_1), ..., I_L(y_L); F^0, P^{\text{ref}}, P^{\text{ref1}}) \approx \frac{1}{N} \sum_{i=1}^{N} P_{I^0}(Y_{i1}, Y_{i2}, ..., Y_{iL}) \tilde{L}(Y_i) + c \]

\( \triangleq C_{MC}(I_1(y_1), ..., I_L(y_L); F^0, P^{\text{ref}}, P^{\text{ref1}}, N). \)  

(14)

(15)

Based on the strong law of large numbers, the expectation (13) can be approximated by the empirical average (14). Denote (14), namely the Monte Carlo cost function, as \( C_{MC}(I_1(y_1), ..., I_L(y_L); F^0, P^{\text{ref}}, P^{\text{ref1}}, N) \). The optimal importance sampling density is \( g(y_1, y_2, ..., y_L; \mathcal{P}_{H^0} \tilde{L}(y_1, y_2, ..., y_L)) \) (see, e.g., [34, 35]).

The initial goal is to minimize the Bayesian cost function (10). Instead, we can minimize the Monte Carlo cost function (15) by selecting a set of optimal sensor rules \( I_1(y_1), I_2(y_2), ..., I_L(y_L) \) and an optimal fusion rule \( F^0 \). In this manner, the high-dimensional integration problem is converted to a problem where we need to deal with the single summation objective function for large-scale sensor networks. Thus, for dependent observations with channel errors, the computational complexity is reduced significantly by the Monte Carlo importance sampling method. In the following sections, we assume that the samples drawn from the importance sampling
density are fixed so that \( C_{MC}(I_1, \ldots, I_L; P^0, P^e_0, P^e_1, N) \) does not have any randomness, since only deterministic decision rules are considered in this chapter.

3. A necessary condition for the optimal sensor rules

In this section, when the fusion rule is fixed, we derive a necessary condition for the optimal sensor rules that minimize the Monte Carlo cost function. First, we need some equivalent transformations for \( P_{H_0} \). Then based on the transformations, the necessary condition can be obtained. At the same time, an analytical result is obtained when the fusion rule is the \( K \)-out-of-\( L \) rule.

3.1 Necessary condition

First, we need some equivalent transformations for \( P_{H_0} \).

Lemma 1 \( P_{H_0} \) can be rewritten as follows:

\[
P_{H_0} = \left[ 1 - I_j(y_j) \right] P_1(1, \ldots, I_{j-1}(y_{j-1}), I_{j+1}(y_{j+1}), \ldots, I_L(y_L); F^0; P^e_0, P^e_1)
+ P_2(1, \ldots, I_{j-1}(y_{j-1}), I_{j+1}(y_{j+1}), \ldots, I_L(y_L); F^0; P^e_0, P^e_1),
\]

(16)

where for \( j = 1, 2, \ldots, L, \)

\[
P_{j1}(\cdot) \triangleq \sum_{k=1}^{2^L} \left[ 1 - F(s_k) \right] \left[ 1 - P^e_j - P^e_1 \right] (1 - 2s_k(j)) P_{m \neq j},
\]

(17)

\[
P_{j2}(\cdot) \triangleq \sum_{k=1}^{2^L} \left[ 1 - F(s_k) \right] \left[ s_k(j) + P^e_1 (1 - 2s_k(j)) \right] P_{m \neq j},
\]

(18)

\[
P_{m \neq j} \triangleq \prod_{m, m \neq j} \left\{ (1 - P^e_j)(1 - s_k(m))(1 - I_m) + P^e_0 s_k(m)(1 - I_m) + (1 - P^e_1) s_k(m) I_m + P^e_1 s_k(m) I_m \right\}.
\]

(19)

**Proof:** If \( s_k(m) = I_m(y_m) \) for all \( m = 1, \ldots, L \), then the continued product term \( \prod_{m=1}^{L} \left\{ (1 - I_m(y_m))(1 - s_k(m)) + I_m(y_m) s_k(m) \right\} = 1 \) in \( P_{H_0} \). Otherwise, it is 0. Thus, \( P_{H_0} \) can be rewritten as \( P_{H_0} = \sum_{k=1}^{2^L} \left[ 1 - F(s_k) \right] P(s_k|(I_1, I_2, \ldots, I_L)) \), where the terms that equal zero are omitted and \( P(s_k|(I_1, I_2, \ldots, I_L)) = \prod_{j=1}^{L} P(s_k(j)|I_j) \). Recalling the conditional probability formula (5), we rewrite \( P(s_k(j)|I_j) \) as \( P(s_k(j)|I_j) = [1 - I_j] \left( 1 - P^e_j - P^e_1 \right) (1 - 2s_k(j)) + s_k(j) + P^e_1 (1 - 2s_k(j)) \). Based on these transformations, \( P_{H_0} \) can be decomposed as (16).

Remark 2: Note that \( P_{j1}(\cdot) \) and \( P_{j2}(\cdot) \) are both independent of \( I_j(y_j) \) for \( j = 1, \ldots, L \). In addition, they can also be applied in the Riemann sum approximation (see, e.g., [31]).
Compared with [36], the sum of $2^L$ terms about $s_k$ is eliminated and it greatly reduces the computational time. In addition, the expression for $P_j(\cdot)$ given in (17) is also a key equation in the following results:

Substituting the transformations (16) into (15), we obtain

$$ C_{MC}(I_1(y_1), \ldots, I_L(y_L); F^0, P^{ce}, P^{cr}, N) $$

$$= c + \frac{1}{N} \sum_{i=1}^{N} \left\{ [1 - f_j(Y_i)] P_j(I_1(Y_i), \ldots, I_{j-1}(Y_{i,j-1}), I_j(Y_{i,j}), \ldots, I_L(Y_{iL}); F^0, P^{ce}, P^{cr}) \right\} \cdot \hat{L}(Y_i), $$

(20)

where $Y_i = (Y_{i1}, Y_{i2}, \ldots, Y_{iL})$. According to (20), the necessary condition for the optimal sensor rules that minimize $C_{MC}(I_1(y_1), \ldots, I_L(y_L); F^0, P^{ce}, P^{cr}, N)$ in the parallel Bayesian detection network, then they must satisfy the following equations:

$$ I_1(Y_{i1}) = f \left[ P_{11}(I_2(Y_{i2}), I_3(Y_{i3}), \ldots, I_L(Y_{iL}); F^0, P^{ce}, P^{cr}) \cdot \hat{L}(Y_i) \right], $$

(21)

$$ I_2(Y_{i2}) = f \left[ P_{21}(I_1(Y_{i1}), I_3(Y_{i3}), \ldots, I_L(Y_{iL}); F^0, P^{ce}, P^{cr}) \cdot \hat{L}(Y_i) \right], $$

(22)

$$ \ldots $$

$$ I_L(Y_{iL}) = f \left[ P_{L1}(I_1(Y_{i1}), I_2(Y_{i2}), \ldots, I_{L-1}(Y_{iL-1}); F^0, P^{ce}, P^{cr}) \cdot \hat{L}(Y_i) \right], $$

(23)

where $P_{j1}(\cdot)$ are defined by (17) and $f(\cdot)$ is an indicator function defined as follows:

$$ f(x) = \begin{cases} 1, & \text{if } x \geq 0; \\ 0, & \text{if } x < 0. \end{cases} $$

(24)

**Proof:** Note that both $P_{j1}(\cdot)$ and $P_{j2}(\cdot)$ are independent of $I_j(y_j)$ for $j = 1, \ldots, L$. If $I_1(Y_{i1})$ minimizes the Monte Carlo cost function under the given $I_2(Y_{i2}), \ldots, I_L(Y_{iL})$, we only need to minimize the first term of the summation in (20), that is,

$$ [1 - f_j(Y_i)] P_{j1}(I_2(Y_{i2}), I_3(Y_{i3}), \ldots, I_L(Y_{iL}); F^0, P^{ce}, P^{cr}) \cdot \hat{L}(Y_i). $$

Note that the value of $I_1(Y_{i1})$ is 0 or 1 and $g(y)$ is well defined, that is, $g(Y_i) > 0$, $I_1(Y_{i1})$ should be equal to 1 when $P_{j1}(I_2(Y_{i2}), I_3(Y_{i3}), \ldots, I_L(Y_{iL}); F^0, P^{ce}, P^{cr}) \hat{L}(Y_i) \geq 0$ for $i = 1, \ldots, N$, otherwise it should be equal to 0. Therefore, we obtain (21) by the definition of $f(x)$ in (24). Similarly, we obtain (22) and (23) by minimizing (20).

### 3.2 An analytical result for the K-out-of-L rule

When the fusion rule is a $K$-out-of-$L$ rule, we would obtain an analytical result in the presence of nonideal channels. It is described as follows:

**Theorem 1:** If the fusion rule is a $K$-out-of-$L$ rule and the probabilities of channel errors are less than 0.5 (i.e., $0 < P^{ce} < 0.5, 0 < P^{cr} < 0.5$) for each channel, the optimal sensor rules are $I_j(Y_{ij}) = f(\hat{L}(Y_i))$ for $i = 1, \ldots, N$ and $j = 1, \ldots, L$. 

8
Proof: From Lemma 1, we know

\[ P_{j1}(\cdot) = \sum_{k=1}^{N_S} (1 - F(s_{k})|P_{m\neq j}) \cdot (1 - P_{j}^{c0} - P_{j}^{c1})[1 - 2s_{k}(j)] \]

\[ = \left(1 - P_{j}^{c0} - P_{j}^{c1}\right) \sum_{k=1}^{N_S} \left[(1 - F(s_{k}|s_{k}(j) = 0)) - (1 - F(s_{k}|s_{k}(j) = 1))\right]P_{m\neq j} \]

\[ = \left(1 - P_{j}^{c0} - P_{j}^{c1}\right) \sum_{k=1}^{N_S} [F(s_{k}|s_{k}(j) = 1) - F(s_{k}|s_{k}(j) = 0)]P_{m\neq j}. \]

(25)

Since \(0 < P_{j}^{c0} < 0.5, 0 < P_{j}^{c1} < 0.5\), we have \(1 - P_{j}^{c0} - P_{j}^{c1} > 0\). Obviously, \(P_{m\neq j} > 0\) holds from its definition. If \(F(s_{k}|s_{k}(j) = 1) - F(s_{k}|s_{k}(j) = 0)\) \(\geq 0\), \(P_{j1}(\cdot) \geq 0\) can be derived. When the fusion rule is a \(K\)-out-of-\(L\) rule, \(F(s_{k}) = I\left[\sum_{j=1}^{L}s_{k}(j) - K\right]\). Thus,

\[ F(s_{k}|s_{k}(j) = 1) = I\left[\sum_{m=1}^{L}s_{k}(m) + 1 - K\right], \]

\[ F(s_{k}|s_{k}(j) = 0) = I\left[\sum_{m=1}^{L}s_{k}(m) + 0 - K\right]. \]

If \(\sum_{m=1}^{L}s_{k}(m) + 0 - K \geq 0\), then \(\sum_{m=1}^{L}s_{k}(m) + 1 - K \geq 0\), and we can get that \(F(s_{k}|s_{k}(j) = 1) - F(s_{k}|s_{k}(j) = 0) = 0\). If \(\sum_{m=1}^{L}s_{k}(m) + 0 - K < 0\), then \(F(s_{k}|s_{k}(j) = 1) - F(s_{k}|s_{k}(j) = 0) \geq 0\). In a word, \(F(s_{k}|s_{k}(j) = 1) - F(s_{k}|s_{k}(j) = 0) \geq 0\) is derived, thus \(P_{j1} \geq 0\). It is easy to find a \(s_{k}(m), m \neq j\) so that \(\sum_{m=1}^{L}s_{k}(m) + 0 = K - 1\) and \(\sum_{m=1}^{L}s_{k}(m) + 1 - K\). Thus, there must exist \(F(s_{k}|s_{k}(j) = 1) - F(s_{k}|s_{k}(j) = 0) > 0\). Therefore, the \(P_{j1} > 0\) is derived. Recalling the necessary condition for the optimal sensor rules, that is, \(I(Y_{ij}) = I\left[\sum_{j=1}^{L}1 - L(Y_{i})\right]\), the proof is completed.

Remark 3: The \(K\)-out-of-\(L\) rule counts the number of sensors that vote in favor of \(H_{1}\) and compares it with a given threshold \(K\) [37]. It is also referred to as the counting rule or voting rule and is widely used in the practical decision fusion area [38, 39]. It encompasses a general class of fusion rules such as AND, OR, and Majority Boolean fusion rules [40]. The reason we assume that the probabilities of channel errors are less than 0.5 is based on practical considerations. If the probabilities of channel errors are greater than or equal to 0.5, the channel is totally unreliable and the performance is not better than a random decision. Obviously, the analytical solution is very efficient to tackle large-scale sensor networks with dependent observations and channel errors.

4. Monte Carlo Gauss-Seidel iterative algorithm and its convergence

For general fusion rules that do not have the form of a \(K\)-out-of-\(L\) rule, an efficient algorithm can be obtained that is inspired by Lemma 2. Next, we present a Monte Carlo Gauss-Seidel iterative algorithm and derive its convergence, when the fusion rule is fixed.
### 4.1 Monte Carlo Gauss-Seidel iterative algorithm

Based on the fixed-point type necessary condition given in Lemma 2, the Monte Carlo Gauss-Seidel iterative algorithm is presented in Algorithm 1.

**Algorithm 1:** Optimization of the sensor rules.

| Step 1: Generate \( N \) samples: \( Y_1, \ldots, Y_N \sim g(y) \), where \( g(y) \) is an importance sampling density and \( Y_i = \{Y_{i1}, Y_{i2}, \ldots, Y_{il}\} \).

| Step 2: Initialize the \( L \) sensor rules, for \( j = 1, 2, \ldots, L \) and \( i = 1, \ldots, N \), \( I_j^{(0)}(Y_{ij}) = 0/1 \). (26)

| Step 3: Iteratively search for the \( L \) sensor rules until a termination criterion in Step 4 is satisfied. The \( n + 1 \)th iteration is given as follows: for \( i = 1, \ldots, N \),

\[
I_1^{(n+1)}(Y_{i1}) = I_L^{(n)}(Y_{i1})
\]

\[
I_2^{(n+1)}(Y_{i2}) = I_L^{(n)}(Y_{i2}) \quad \text{as in } \[31\].
\]

\[
I_L^{(n+1)}(Y_{il}) = I_L^{(n)}(Y_{il})
\]

Remark 4: When we obtain \( I_i(Y_{il}) \) for \( i = 1, \ldots, N \), we can compress \( y_{ij} \) by defining \( I_i(Y_{il}) = I_i(Y_{il}) \) if the distance \( \|y_{i1} - Y_{il}\| \leq \|y_{i2} - Y_{il}\| \) for all \( i' \neq i \). In the same way, we can compress \( y_j \) for \( j = 1, \ldots, L \). In fact, the method is to find one nearest neighbor of \( y_i \) for \( i = 1, \ldots, L \) and use the corresponding compression rule. Moreover, we can utilize the k-nearest neighbor (knn) to compress \( y_j \) (see more in \[41\]).

Remark 5: The main computation burden of Algorithm 1 is included in (27)–(29). If we let the number of discretized points \( N_1 = N_2 = \ldots = N_L = N \) in \[31\], then \( P_{11}(L(Y_{il})), j = 1, \ldots, L, \) and \( i = 1, \ldots, N \) are computed \( LN^L \) times for each iteration, as in \[31\]. But they only need to be computed \( LN \) times in Algorithm 1. Thus, the computational complexity of Algorithm 1, i.e., \( O(LN) \) is much less than that in \[31\], that is, \( O(LN^L) \). It is more efficient to tackle large-scale sensor networks with dependent observations and channel errors.

---

**Functional Calculus - Recent Advances and Development**

---

| Step 4: For \( i = 1, \ldots, N \), the termination criterion of the iteration process is

\[
I_1^{(n+1)}(Y_{i1}) = I_1^{(n)}(Y_{i1}),
\]

\[
I_2^{(n+1)}(Y_{i2}) = I_2^{(n)}(Y_{i2}),
\]

\[
I_L^{(n+1)}(Y_{il}) = I_L^{(n)}(Y_{il}).
\]

---

---
4.2 Convergence of the iterative algorithm

Now, we show that Algorithm 1 must converge to a stationary point and the algorithm cannot oscillate infinitely, that is, it terminates after a finite number of iterations.

Lemma 3: Given the fusion rule $P^0$, for any initial values $(I_{1}^{0}, ..., I_{L}^{0})$ in (26), $C_{MC}(I_{1}^{(n)}, ..., I_{j}^{(n)}, I_{j+1}^{(n)}, ..., I_{L}^{(n)}; P^0, P^{\infty 0}, P^{\infty 1}, N)$ must converge to a stationary point after a finite number of iterations.

**Proof:** For $j = 1, ..., L$, we denote $C_{MC}$ (20) in the $n + 1$th iteration process by

$$C_{MC}(I_{1}^{(n+1)}, ..., I_{j}^{(n+1)}, I_{j+1}^{(n)}, ..., I_{L}^{(n)}; P^0, P^{\infty 0}, P^{\infty 1}, N)$$

$$= \frac{1}{N} \sum_{i=1}^{N} \left\{ 1 - I_{i}^{(n+1)}(Y_{i}) \right\} P_{j}^{1}(I_{i}^{(n+1)}(Y_{i}), ..., I_{j-1}^{(n+1)}(Y_{i-j+1}), I_{j}^{(n+1)}(Y_{i-j+1}), ..., I_{L}^{(n+1)}(Y_{iL}); P^0; P^{\infty 0}, P^{\infty 1}, N)$

$$\frac{\hat{L}(Y_{i})}{g(Y_{i})} + c.$$  

(31)

Similarly, we denote the $(n + 1)$th iteration process of the iterative items $P_{j}(\cdot) \hat{L}(\cdot)$ in (27)–(29) by

$$G_{j}^{i} = P_{j}^{1}(I_{1}^{(n+1)}(Y_{i}), ..., I_{j-1}^{(n+1)}(Y_{i-j+1}), I_{j}^{(n+1)}(Y_{i-j+1}), ..., I_{L}^{(n+1)}(Y_{iL}); P^0; P^{\infty 0}, P^{\infty 1}, N) \hat{L}(Y_{i}),$$

(32)

for $i = 1, ..., N$ and $j = 1, ..., L$. Plugging $G_{j}^{i}$ into (31), we know

$$C_{MC}(I_{1}^{(n+1)}, ..., I_{j}^{(n+1)}, I_{j+1}^{(n)}, ..., I_{L}^{(n)}; P^0, P^{\infty 0}, P^{\infty 1}, N) = \frac{1}{N} \sum_{i=1}^{N} \left( 1 - I_{i}^{(n+1)}(Y_{i}) \right) G_{j}^{i} + C_{j}^{i},$$

(33)

where $C_{j}^{i} = c + \frac{1}{N} \sum_{i=1}^{N} P_{j}^{1}(I_{1}^{(n+1)}(Y_{i}), ..., I_{j-1}^{(n+1)}(Y_{i-j+1}), I_{j}^{(n+1)}(Y_{i-j+1}), ..., I_{L}^{(n+1)}(Y_{iL}); P^0; P^{\infty 0}, P^{\infty 1}, N)$ is independent of $I_{j}^{(n)}$ and $I_{j}^{(n+1)}$. Splitting $1 - I_{i}^{(n+1)}(Y_{i})$ into two terms, we obtain

$$C_{MC}(I_{1}^{(n+1)}, ..., I_{j}^{(n+1)}, I_{j+1}^{(n)}, ..., I_{L}^{(n)}; P^0, P^{\infty 0}, P^{\infty 1}, N)$$

$$= \frac{1}{N} \sum_{i=1}^{N} \left( 1 - I_{i}^{(n)}(Y_{i}) \right) + \left( I_{j}^{(n)}(Y_{i}) - I_{j}^{(n+1)}(Y_{i}) \right) G_{j}^{i} + C_{j}^{i},$$

(34)

$$= \frac{1}{N} \sum_{i=1}^{N} \left( 1 - I_{i}^{(n)}(Y_{i}) \right) G_{j}^{i} + C_{j}^{i} + \frac{1}{N} \sum_{i=1}^{N} \left( I_{j}^{(n)}(Y_{i}) - I_{j}^{(n+1)}(Y_{i}) \right) G_{j}^{i}$$

$$= C_{MC}(I_{1}^{(n+1)}, ..., I_{j-1}^{(n+1)}, I_{j}^{(n)}, ..., I_{L}^{(n)}; P^0, P^{\infty 0}, P^{\infty 1}, N) + D_{j}^{(n+1)}.$$
functional calculus - recent advances and development

where

\[ D_j^{(n+1)} = \frac{1}{N} \sum_{i=1}^{N} \frac{[f_j^{(n)}(Y_i) - f_j^{(n+1)}(Y_i)]}{g(Y_i)} G_j. \]  

(35)

Note that (27)–(29) imply that \( f_j^{(n+1)}(Y_i) = 0 \) if and only if \( G_j < 0 \) and \( f_j^{(n+1)}(Y_i) = 1 \) if and only if \( G_j \geq 0 \) for \( i = 1, \ldots, N, j = 1, \ldots, L \). It means

\[ [f_j^{(n)}(Y_i) - f_j^{(n+1)}(Y_i)] G_j \leq 0. \]  

(36)

Thus, for \( \forall i, j \)

\[ [f_j^{(n)}(Y_i) - f_j^{(n+1)}(Y_i)] G_j g(Y_i) \leq 0, \]  

(37)

where the inequality (35) holds since \( g(\cdot) \) is well-defined (i.e., \( g(\cdot) > 0 \)). Substituting (35) into (33) yields \( D_j^{(n+1)} \leq 0 \). Thus, for \( \forall j \leq L \),

\[ C_{MC}(I_1^{(n+1)}, \ldots, I_j^{(n+1)}, \ldots, I_L^{(n+1)}; P^0, P^c, P^{c^1}, N) \leq C_{MC}(I_1^{(n+1)}, \ldots, I_{j-1}^{(n+1)}, I_j^{(n+1)}, \ldots, I_L^{(n+1)}; P^0, P^c, P^{c^1}, N). \]  

(38)

Furthermore,

\[ C_{MC}(I_1^{(n+1)}, I_2^{(n+1)}, \ldots, I_L^{(n+1)}; P^0, P^c, P^{c^1}, N) \leq C_{MC}(I_1^{(n)}, I_2^{(n)}, \ldots, I_L^{(n)}; P^0, P^c, P^{c^1}, N). \]  

(39)

It means \( C_{MC} \) is nonincreasing. Note that \( C_{MC}(I_1^{(n)}, I_2^{(n)}, \ldots, I_L^{(n)}; P^0, P^c, P^{c^1}, N) \) is a finite value. We conclude that it must converge to a stationary point after a finite number of iterations.

Theorem 1.2: Given the fusion rule \( P^0 \), the sensor rules \( I_1^{(n)}, I_2^{(n)}, \ldots, I_L^{(n)} \) are finitely convergent, i.e., Algorithm 1 converges after a finite number of iterations.

Proof: By Lemma 3, \( C_{MC} \) must attain a stationary point after a finite number of iterations. It means that the value of \( C_{MC} \) cannot change after \( n \)th iteration, that is,

\[ C_{MC}(I_1^{(n+1)}, \ldots, I_j^{(n+1)}, \ldots, I_L^{(n+1)}; P^0, P^c, P^{c^1}, N) = C_{MC}(I_1^{(n+1)}, \ldots, I_{j-1}^{(n+1)}, I_j^{(n+1)}, \ldots, I_L^{(n+1)}; P^0, P^c, P^{c^1}, N). \]  

(40)

Using (32) and (37), we derive that \( D_j^{(n+1)} = 0 \). Combining (33)–(35), we know

\[ [f_j^{(n)}(Y_i) - f_j^{(n+1)}(Y_i)] G_j = 0, \text{ for } i = 1, \ldots, N, \]  

(41)

which implies either \( f_j^{(n)}(Y_i) = f_j^{(n+1)}(Y_i) = 0 \), i.e., \( I_j^{(n+1)}(Y_i) = I_j^{(n+1)}(Y_i) \) or \( G_j = 0 \), i.e., \( I_j^{(n+1)}(Y_i) = 1, I_j^{(n)}(Y_i) = 0 \). It follows that when \( C_{MC} \) converges to a stationary
point, either $I_j^{(n+1)}(Y_g)$ is invariant or $I_j^{(n+1)}(Y_g) = 1$, $I_j^{(n)}(Y_g) = 0$. Namely, $I_j^{(n+1)}(Y_g)$ can only change from 0 to 1 at most a finite number of times. Therefore, the $I_1^{(n)}$, $I_2^{(n)}$, ..., $I_L^{(n)}$ are finitely convergent.

5. Extension for simultaneous search for the optimal sensor rules and fusion rule

In this section, we extend the Monte Carlo method to search for the optimal sensor rules and the optimal fusion rule simultaneously. Firstly, the necessary condition is generalized to search for the optimal sensor rules and the optimal fusion rule simultaneously. Secondly, we describe a generalized Monte Carlo Gauss-Seidel iterative algorithm. We also give the convergence of the iterative algorithm.

5.1 A necessary condition for the optimal sensor rules and the optimal fusion rule

Note that (15) can be rewritten as follows:

$$C_{MC}(I_1(y_1), \ldots, I_L(y_L); F^0, \ldots, F^{p^c}, P_{c1}, N)$$

$$= c + \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{2^L} \sum_{k=1}^{2^L} [1 - F^0(s_{k'})] P(s_{k'}|s_k) \cdot P_n(I(y_i)) \frac{\hat{L}(Y_i)}{g(Y_i)}$$

$$= c + \frac{1}{N} \sum_{k=1}^{2^L} [1 - F^0(s_{k'})] \sum_{i=1}^{N} \sum_{k=1}^{2^L} P(s_{k'}|s_k) \cdot P_n(I(y_i)) \frac{\hat{L}(Y_i)}{g(Y_i)},$$  \hspace{1cm} (42)

where $P_n(I(y_i)) \triangleq \prod_{j=1}^{L} [s_k(j)I_j(y_g) + (1 - s_k(j))(1 - I_j(y_g))]$ and $I(y_i) = (I_1(y_{i1}), I_2(y_{i2}), \ldots, I_L(y_{iL}))$. Since $P_n(I(y_i)) = 1$ if and only if $I_j = s_k(j)$ for all $j = 1, \ldots, L$, (39) can be simplified as follows:

$$C_{MC}(I_1(y_1), \ldots, I_L(y_L); F^0, \ldots, F^{p^c}, P_{c1}, N)$$

$$= c + \frac{1}{N} \sum_{k=1}^{2^L} [1 - F^0(s_{k'})] \cdot \sum_{i=1}^{N} P(s_{k'}|I(y_{i1}), \ldots, I_L(y_{iL})) \frac{\hat{L}(Y_i)}{g(Y_i)},$$  \hspace{1cm} (43)

where the terms $P_n(I(y_i)) = 0$ are eliminated.

Remark 6: According to (20) and (40), the necessary condition for the optimal sensor rules is similar to Lemma 2 and the necessary condition for the optimal fusion rule is given by

$$F^0(s_{k'}) = I \left[ \sum_{i=1}^{N} P(s_{k'}|I(y_{i1}), \ldots, I_L(y_{iL})) \cdot \frac{\hat{L}(Y_i)}{g(Y_i)} \right]$$  \hspace{1cm} (44)

for $k' = 1, \ldots, 2^L$. The proofs are similar to Lemma 2.
5.2 Generalized Monte Carlo Gauss-Seidel iterative algorithm

Based on the fixed-point type necessary condition, the generalized Monte Carlo Gauss-Seidel iterative algorithm is presented in Algorithm 2.

Remark 7: For any initial values \( (I_{1}^{(0)}, \ldots, I_{L}^{(0)}; F^{(0)}) \), the Monte Carlo cost function \( C_{MC} (I_{1}^{(n)}, \ldots, I_{L}^{(n)}; F^{(n)}; P_{ce}, P_{cr}, N) \) must converge to a stationary point and Algorithm 2 terminates after a finite number of iterations. The proofs are similar to those of Lemma 3 and Theorem 1.2.

**Algorithm 2**: Simultaneous optimization of the sensor rules and the fusion rule.

- **Step 1**: Generate \( N \) samples: \( Y_{1}, \ldots, Y_{N} \sim g(y) \), where \( g(y) \) is an importance sampling density and \( Y_{i} = [Y_{i1}, Y_{i2}, \ldots, Y_{il}] \).

- **Step 2**: Initialize the \( L \) sensor rules and the fusion rule, respectively, for \( j = 1, 2, \ldots, L, i = 1, \ldots, N \), and \( k' = 1, \ldots, 2^{L} \),
  
  \[
  I_{j}^{(0)}(y_{j}) = 0/1, F^{(0)}(s_{k'}) = 0/1.
  \]

- **Step 3**: Iteratively search for the \( L \) sensor rules and the fusion rule until a termination criterion in Step 4 is satisfied. The \( n + 1 \)th iteration is given as follows: for \( i = 1, \ldots, N \) and \( k' = 1, \ldots, 2^{L} \),

  \[
  I_{1}^{(n+1)}(Y_{il}) = I \left[ P_{1i} I_{1}^{(n)}(Y_{i2}), I_{2}^{(n)}(Y_{i3}), \ldots, I_{L}^{(n)}(Y_{il}); F^{(n)}; P_{ce}, P_{cr} \right] \cdot \hat{L}(Y_{i})],
  
  I_{2}^{(n+1)}(Y_{i2}) = I \left[ P_{2i} I_{1}^{(n+1)}(Y_{i1}), I_{3}^{(n)}(Y_{i3}), \ldots, I_{L}^{(n)}(Y_{il}); F^{(n)}; P_{ce}, P_{cr} \right] \cdot \hat{L}(Y_{i})],
  
  \vdots
  
  I_{L}^{(n+1)}(Y_{il}) = I \left[ P_{L1} I_{1}^{(n+1)}(Y_{i1}), I_{2}^{(n+1)}(Y_{i2}), \ldots, I_{L-1}^{(n+1)}(Y_{i(l-1)}); F^{(n)}; P_{ce}, P_{cr} \right] \cdot \hat{L}(Y_{i})],
  
  F^{(0)}(s_{k'}) = I \left[ \sum_{i=1}^{N} P(s_{k'} | I_{1}^{(n+1)}(Y_{i1}), I_{2}^{(n+1)}(Y_{i2}), \ldots, I_{L}^{(n+1)}(Y_{il})) \hat{L}(Y_{i})/g(Y_{i}) \right].
  
- **Step 4**: For \( i = 1, \ldots, N \) and \( k' = 1, 2, \ldots, 2^{L} \), the termination criterion of the iteration process is

  \[
  I_{1}^{(n+1)}(Y_{i1}) = I_{1}^{(n)}(Y_{i1}),
  
  I_{2}^{(n+1)}(Y_{i2}) = I_{2}^{(n)}(Y_{i2}),
  
  \vdots
  
  I_{L}^{(n+1)}(Y_{il}) = I_{L}^{(n)}(Y_{il});
  
  F^{(0)}(s_{k'}) = F^{(0)}(s_{k'}).
6. Numerical examples

In this section, in order to evaluate the performance of Algorithms 1 and 2, we present some examples with a Gaussian signal $s$ observed in the presence of Gaussian sensor noises.

The random signal $s$ and observation noises $v_1, v_2, \ldots, v_L$ are as follows:

$$
H_0: \ y_j = v_j; \quad H_1: \ y_j = s + v_j, \quad \text{for} \ j = 1, \ldots, L, \quad (45)
$$

where $v_1, v_2, \ldots, v_L, s$ are all mutually independent and

$$
v_j \sim N(0, 0.6), \quad s \sim N(1, 0.4), \quad \text{for} \ j = 1, \ldots, L.
$$

Thus, given $H_0$ and $H_1$, the two conditional probability density functions are

$$
p(y_1, y_2, \ldots, y_L | H_0) \sim N(\mu_0, \Sigma_0), \quad p(y_1, y_2, \ldots, y_L | H_1) \sim N(\mu_1, \Sigma_1),
$$

where $\mu_0, \mu_1, \Sigma_0, \Sigma_1$ are easily obtained from the relationship of $s, v_1, v_2, \ldots, v_L$.

Assume that each sensor is required to transmit a bit through a channel with probabilities of $P_x = P_x^c = p$, where $p = 0.05, 0.15, 0.3$, for $j = 1, 2, \ldots, L$. In the cost function (2), let the cost coefficients $C_{00} = C_{11} = 0$ and $C_{10} = C_{01} = 1$. The receiver operating characteristics (ROC) curves are used to evaluate the performance of the algorithms. $P_f$ and $P_d$ denote the probability of false alarm and the probability of detection, respectively.

6.1 Two-sensor network

We compare the Monte Carlo Gauss-Seidel iterative algorithm with the centralized algorithm and the iterative algorithm based on the Riemann sum approximation in [31] by using the receiver operating characteristics (ROC) curves.

In this case, we know $\mu_0 = [0, 0]^T, \mu_1 = [1, 1]^T$ and $\Sigma_0 = [0.6, 0; 0, 0.6], \Sigma_1 = [1, 0.4; 0.4, 1]$. Some discrete values of $a$ and $b$ are used to plot ROC curves. We refer to the optimal importance sampling density $g(y) \propto \mathcal{P}(y) L(y)$ in Section 2.2 and $|L(y)| = |ap(y|H_1) - bp(y|H_0)|$. The form is similar to the mixture-Gaussian distribution. Therefore, the importance sampling density $g(y)$ is chosen to be the mixture-Gaussian distribution. The effects of choosing different $g(y)$ in terms of the performance of the Monte Carlo method were shown in [21] via numerical examples. For Algorithm 1, we take $N = 200$ samples from the density $g(y)$. For the Riemann sum approximation iterative algorithm in [31], we take a discretized step-size $\Delta = 0.09$, $y_1 \in [-8, 10]$, i.e., $N_1 = N_2 = N = 200$. The ROC curves for three important fusion rules: AND, OR, and XOR rules with $p = 0.05, 0.15, 0.3$ are plotted in Figure 2. We compare the computational time of the two algorithms with $p = 0.15$ in Figure 3. Note that the analytical solution is used for the AND rule and the OR rule. Since the XOR rule is not a $K$-out-of-$L$ rule, we use Algorithm 1 to search for the sensor rules.

Some observations in Figures 2 and 3 are presented as follows:

- Given the fusion rule, the two points $(0, 0)$ and $(1, 1)$ may not be the beginning or ending points of the ROC curves, which is different from the case in the ideal channel cases. In addition, the larger the probability of channel errors is, the
farther away from (0, 0) or (1, 1) the ROC curves are. A possible reason is that the detection probability is not equal to 0 or 1, even when the false alarm probability is 0 or 1 in the presence of channel errors.

- From Figure 2, when the probability of channel transmission errors increases, the decision fusion performance of different methods using the optimal sensor rules decreases.

- It can be seen in Figure 2 that the ROC curves of the new Monte Carlo approach are very close to those of the previous algorithm based on the Riemann sum approximation. However, from Figure 3, the computational time of the Monte Carlo importance sampling approximation is much less than that of the Riemann sum approximation for the three different fusion rules. It also implies that the new method can be used to deal with large-scale sensor networks.

- Note that the computational time of the AND rule and the OR rule is less than that of the XOR rule for the Monte Carlo importance sampling approximation from Figure 3. The reason is that the AND rule and the OR rule belong to the \( K \)-out-of-\( L \) rules. The analytical form is used for the AND rule and the OR rule, therefore, the corresponding computation time is relatively lower.

Figure 2.
Two-sensor ROC curves with the probability of channel errors \( p = 0.05, 0.15, 0.3 \).

Figure 3.
Two-sensor computational time as \( N \) increases with the probability of channel errors \( p = 0.15 \).
6.2 Ten-sensor network

We consider a larger sensor network with 10 sensors, which cannot be dealt with by the previous decision fusion algorithm based on the Riemann sum approximation due to its heavy computation requirements. For different probabilities of channel errors, the ROC curves of the AND rule, the OR rule, the 4-out-of-10 rule, the 6-out-of-10 rule, and Algorithm 2 are plotted in Figure 4.

Some observations in Figure 4 are presented as follows:

- The ROC curves for the ten-sensor network exhibit similar behavior as those for the two-sensor network.

- Given the fusion rule and the probability of channel errors, the decision fusion performance of the AND rule is better than the other rules for a small false alarm probability and the decision fusion performance of the OR rule is better than the other rules for a large false alarm probability. The reason may be that both of them are extreme cases of the fusion rules. For other cases, the 4-out-of-10 rule and the 6-out-of-10 rule perform better than the AND rule and the OR rule, respectively.

- Regardless of the centralized detection algorithm, Figure 4 shows that the ROC curves generated by Algorithm 2 obtain almost the best performance for different probabilities of channel errors. It implies that a simultaneous search for the sensor rules and the fusion rule would provide better decision fusion performance.

6.3 One-hundred-sensor network

We consider a large-scale network with one hundred sensors. The parameter settings are similar to Section 6.2. The ROC curves of the 20-out-of-100 rule, the 40-out-of-100, the 50-out-of-100, the 60-out-of-100 rule, and the 80-out-of-100 rule are plotted in Figure 5.

From Figure 5, it can be seen that the dramatically lower computational requirement of our method enables us to handle a large sensor network consisting of one hundred sensors. This is due to the fact that we have shown that there exist analytical forms of the optimal sensor rules for the $K$-out-of-$L$ rule. In addition, the decision fusion performance of different methods is improved, as the number of sensors becomes large.
7. Conclusion

By employing the Monte Carlo importance sampling technique, decision fusion algorithms have been provided for large-scale sensor networks with dependent observations and channel errors. The Bayesian cost function is approximated by the Monte Carlo cost function. The necessary conditions for the optimal sensor rules and the optimal fusion rule that minimize the Monte Carlo cost function have been obtained. Computationally efficient Monte Carlo Gauss-Seidel iterative algorithms have been proposed to search for the optimal sensor rules and the optimal fusion rule. These algorithms have been shown to converge after a finite number of iterations. The computational complexity of the new algorithm (i.e., $O(LN)$) is much less than that of the previous algorithm based on Riemann sum approximation (i.e., $O(LN^2)$). For the $K$-out-of-$L$ rule, an analytical solution has been presented for the optimal sensor rules. Simulations have demonstrated the effectiveness of Algorithms 1 and 2. Future work will include the decision fusion algorithms under the Monte Carlo framework for other networks such as tandem networks, tree networks, and sensor networks under Byzantine attack.

Acknowledgements

This work was supported in part by the Sichuan Youth Science and Technology Innovation Team under Grant 2022JDTD0014, and Grant 2021JDJQ0036.
Decision Fusion for Large-Scale Sensor Networks with Nonideal Channels
DOI: http:/dx.doi.org/10.5772/intechopen.106075

Author details

Yiwei Liao\textsuperscript{1,2}, Xiaojing Shen\textsuperscript{3}\*, Junfeng Wang\textsuperscript{4} and Yunmin Zhu\textsuperscript{3}

\textsuperscript{1} School of Data Science, The Chinese University of Hong Kong, Shenzhen, Guangdong, China
\textsuperscript{2} Shcool of Information Science and Technology, University of Science and Technology of China, Hefei, Anhui, China
\textsuperscript{3} School of Mathematics, Sichuan University, Chengdu, Sichuan, China
\textsuperscript{4} School of Computer Science, Sichuan University, Chengdu, Sichuan, China

\*Address all correspondence to: shenxj@scu.edu.cn

© 2022 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.
References


[32] Thomopoulos SCA, Zhang L. Distributed decision fusion in the presence of networking delays and...
channel errors. Information Sciences. 1992;66(1-2):91-118


[37] Vergara L. On the equivalence between likelihood ratio tests and counting rules in distributed detection with correlated sensors. Signal Processing. 2007;87(7):1808-1815


