Decision Fusion in Non-stationary Environments

A Ph.D Dissertation

Submitted to the Faculty

of

Drexel University

by

Ji Wang

ECE Department
Drexel University

Advisor: Dr. Moshe Kam, Dr. Leonid Hrebien, and Dr. Xiaohua Hu

Contents

Glossaries

P_M: miss-detection rate of the binary detection

P_F: false-alarm rate of the binary detection

P_D: detection rate of the binary detection

H_1: hypothesis 1, one of the hypothesis in binary detection

H_0: hypothesis 0, one of the hypothesis in binary detection

N: number of sensors in the fusion system with multiple local sensors

u_i: decision of the i^{th} sensor in the fusion system with multiple local sensors

P_{Mi}: miss-detection rate of the i^{th} sensor

P_{Fi}: false-alarm rate of the i^{th} sensor

τ: the threshold for the likelihood test

U_g: global decision of the fusion system

P_{Fg}: global false-alarm rate of the fusion system

P_{Mg}: global miss-detection rate of the fusion system

S_0: number of samples collected by the sequential sensor

X_s: S_0 samples collected by the sequential sensor

T_{S_0}: statistics (sum) of collected samples

p: the probability of error of a sequential in one sampling stage

q: the probability of detection of a sequential in one sampling stage

r: the probability that no decision was made of a sequential in one sampling stage

e: the overall error rate of a sequential sensor

ASN_1: average sample number of a sequential sensor

ASN_g: global average sample number of fusion system with multiple sequential sensors

e_g: global error rate of fusion system with multiple sequential sensors

$P^{(Z)}$: the probability that, in a fusion system with multiple sequential sensors, at least one sequential sensor stopped sampling and reached a decision at the Z^{th} sampling stage

M: number of local decisions in the genetic code that are formed by multiple binary local-sensor decisions

$HD_{(i,j)}$: Hamming distance between two genetic codes

Maj: the majority decisions in the genetic codes

Min: the minority decisions in the genetic codes

Fit^0: the expectation fitness of a sensor, when the hypothesis changes.

Fit_i^E: the expectation fitness of a sensor, assuming the hypothesis does not change.

Fit_i^T: the true value of sensor's fitness

$Dist_i^{T-E}$: $\left|Fit_i^T - Fit_i^E\right|$ the distance between Fit_i^T and Fit_i^E

$Dist_i^{T-0}$: $\left|Fit_i^T - Fit^0\right|$ the distance between Fit_i^T and Fit^0

u_i^{C-D}: sensor-level decision on whether the hypothesis changes

U_g^{C-D}: global-level decision on whether the hypothesis changes

w_i^{C-D}: The i^{th} sensor level decision's weight

Maj_{Final}^T: the true value of the percentage of the majority decisions in the final generation

Maj_{Final}^E: the expectation of the percentage of the majority decisions in the final generation, assuming the hypothesis does not change.

$Dist_{Final}^{T-E}$: $\left|Maj_{Final}^T - Maj_{Final}^E\right|$ the distance between Maj_{Final}^T and Maj_{Final}^E

$Dist_{Final}^{T-0.5}$: $\left|Maj_{Final}^T - 50\%\right|$ the distance between Maj_{Final}^T and 50%

Abstract

A parallel distributed detection system consists of multiple local sensors/detectors that observe a phenomenon and process the gathered observations using inbuilt processing capabilities. The end product of the local processing is transmitted from each sensor/detector to a centrally located data fusion center for integration and decision making. The data fusion center uses a specific optimization criterion to obtain global decisions about the environment seen by the sensors/detectors. In this study, the overall objective is to make a globally-optimal binary (target/non-target) decision with respect to a Bayesian cost, or to satisfy the Neyman-Pearson criterion. We also note that in some cases a globally-optimal Bayesian decision is either undesirable or impractical, in which case other criteria or localized decisions are used. In this thesis, we investigate development of several fusion algorithms under different constraints including sequential availability of data and dearth of statistical information. The main contribution of this study are: (1) an algorithm that provides a globally optimal solution for local detector design that satisfies a Neyman-Pearson criterion for systems with identical local sensors; (2) an adaptive fusion algorithm that fuses local decisions without a prior knowledge of the local sensor performance; and (3) a fusion rule that applies a genetic In addition, we develop a parallel decision fusion system where each local sensor is a sequential decision maker that implements the modified Wald's sequential probability test (SPRT) as proposed by Lee and Thomas (1984).

1. Introduction

Data fusion is the process of combining information from several different sources pertaining to the same event or phenomenon. The goal is to develop a robust and better understanding of the phenomenon or process of interest than what could be achieved by relying on a single source. The field of data fusion is relevant where a large amount of data must be processed and distilled to develop information of appropriate quality and integrity on which decisions can be made. The volume of literature on data fusion underlines the application areas, which include air traffic control, oil exploration, medical diagnosis, military command and control, electric power networks, weather prediction and target tracking [28].

1.1 Data Fusion Models

In most data fusion problems, there is an environment, process or quantity whose true value or state is unknown. It would be impractical to expect that there be a single source of perfect and complete knowledge about the state of interest, and so information is gathered indirectly from several sources, each providing imperfect knowledge. These sources are then combined in some manner to infer the required state and use it to make subsequent decisions. In a statistical framework, the phenomenon being monitored is typically defined using a set of hypotheses. The binary hypothesis detection problem, the simplest of these scenarios, represents a situation where the observed phenomenon is assumed to be either in “state 0” which represents hypothesis H_0, or in “state 1” which represents hypothesis H_1. The challenge is to infer the true state of the phenomenon, based on collected sensory observations Z, prior knowledge about the hypotheses, and a predefined optimality criterion. The final decision of the sensor can be represented as follows,

$$u = \begin{cases} 1, & H_1 \; is \; accepted \\ -1, & H_0 \; is \; accepted \end{cases}$$

Fig.1 shows the binary detection setup with a single sensor.

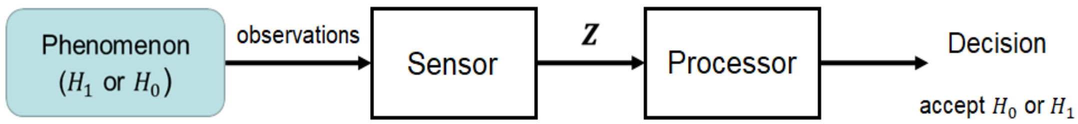


Fig.1. Binary hypothesis testing

Since many multi-hypothesis problems can be represented in terms of a hierarchy of binary hypotheses, this thesis specifically deals with the binary detection problem.

1.2 Optimality Criteria

In a detection setup, the observation data are processed to maximize or minimize a predefined objective function. The most common optimization criteria for binary decisions use a Bayesian framework and minimize the Bayes' risk. Alternatively, we use the Neyman-Pearson criterion (maximization of detection rate (correctly identifying hypothesis H_1) under a constrained false alarm rate (deciding in favor of hypothesis H_1 when H_0 is true)).

The Bayes' risk is a function of the prior probabilities of the hypotheses, the costs of making decisions, and the conditional distributions of the decision conditioned on the hypotheses. It is defined as

$$R = \sum_{i=0}^{1} \sum_{j=0}^{1} C_{ij} P_j P(\text{Decides } H_i | H_j \text{ presents}) \tag{1}$$

where C_{ij} is the cost coefficient when true hypothesis is j while deciding i; P_j is the pror probability of H_j being present.

The Neyman-Pearson criterion fixes the false alarm rate ($P(u = 1|H_0)$) at a pre-specified level $\alpha < 1$ and then attempts to achieve the maximum rate of detection ($P(u = 1|H_1)$)).

A variety of other optimization criteria for hypothesis testing such as Person-by-Person Optimization (PBPO) and use of conditional entropy [11] have been proposed in the literature. The algorithms reported in this thesis are based on the Bayes' risk and the Neyman-Pearson criterion.

1.3 Parallel Data Fusion

The parallel distributed detection problem involves a bank of sensors (each of the form shown in Fig 1) collecting observations about the phenomenon/environment and transmitting a processed version of the observations to a central data fusion center (DFC) which is responsible for data aggregation. The end goal at the data fusion center is to combine the received data in an optimal or near-optimal form to facilitate informed decision making. When no local processing is performed, the data fusion center receives the entire volume of collected information for decision making. This scheme is called centralized data fusion (Fig. 2), and is studied extensively.

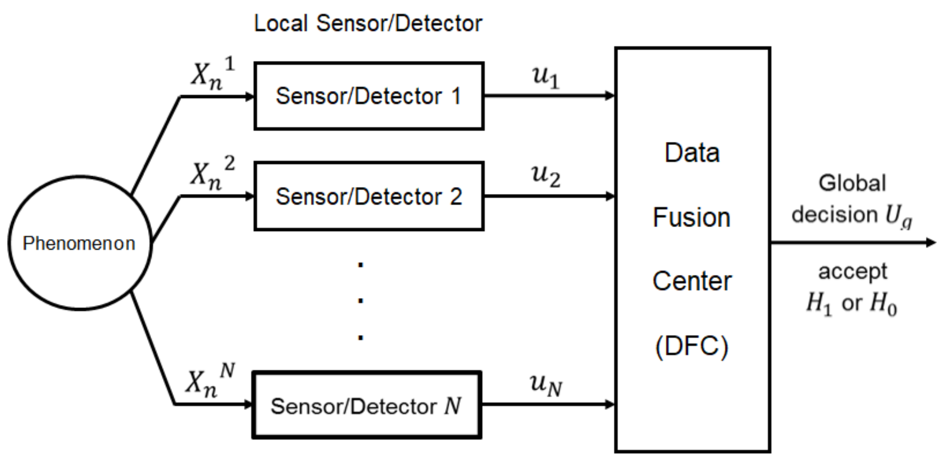


Fig. 2. Centralized Data Fusion: sensors transmit the raw observations to the fusion center.

Since the fusion center in Fig. 2 has complete knowledge of the all information collected, this architecture does not need to reckon with loss of information during the sensing process. On the negative side, the centralized system requires high processing power at the fusion center and high-bandwidth communication channels between the local sensors and the data fusion center. In many architectures, a local processor follows each sensor providing a compressed version of observation of the sensor (e.g., local estimates and decision). Such architectures (see Fig. 3) are motivated by two main considerations: (1) the desire to make the system more modular and flexible (sensors and processors can be exchanged or replaced); and (2) the overhead on communication, data storage, and central computation is usually lower when local compression is employed.

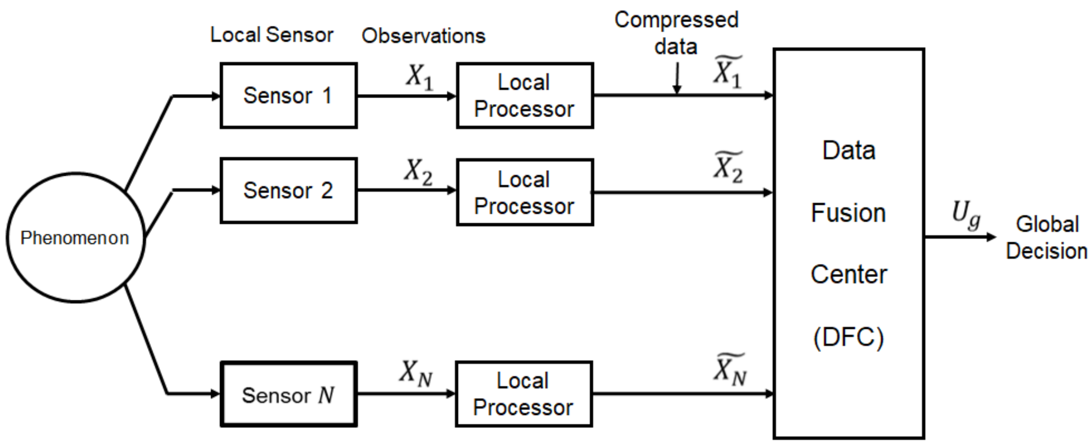


Fig. 3. Distributed Data Fusion: sensors transmit observations to local processor; the local processors transmit processed data to the fusion center

In Fig. 3, which depicts a distributed data fusion, each sensor has an associated local processor which extracts information from the raw sensor observations prior to communication with the data fusion center. Often, a summary or an interpretation of the local observations is sent to the data fusion center, which then makes a global decision based on the compressed data it had received. Various kinds of compressed data can be transmitted to the fusion center from the local detectors such multi-level or binary decisions; changes from a baseline or previous decision; or likelihood ratios of the observed data. In almost all these cases, local processing leads to information loss. Hence even though the distributed scheme is modular, easier to implement and has much less communication bandwidth requirements, it almost always exhibits suboptimal performance as compared to a centralized architecture where the fusion center uses all available information. For any fusion system design, the centralized scheme can therefore be assumed to provide an upper bound on the performance and serve the standard for comparative analysis.

A widely studied variation of the distributed architecture shown in Fig 3 is the parallel binary decision fusion architecture where the local detectors individually generate local decisions about the state of a binary phenomenon (H_0 or H_1) and transmit the local binary decisions to the fusion center for final aggregation. The fusion center uses the local binary decisions $\{\widetilde{X_1} = u_1, \widetilde{X_2} = u_2, \dots, \widetilde{X_N} = u_N\}$ as inputs and generates a global decision U_g.

1.4 Thesis Overview

This thesis delves into the theory and applications of parallel binary decision fusion. We investigate the parallel decision fusion architecture (Fig. 3) using both the Bayes' risk criterion and the Neyman-Pearson criterion.

In Chapter 3, we present an optimal algorithm that provides the decision thresholds of the local detectors such that the global detection rate at the fusion center is maximized*. In this set up, the Neyman-Pearson criterion is used at both local detectors and the fusion center, guaranteeing the every binary decision in the system has a false-alarm rate that does not exceed a certain pre-specified rate α. We compare the performance of our method with the performance of the Person-by-Person Optimization approach and that of a centralized detection scheme.

Chapter 4 focuses on an application of parallel decision fusion. We present a parallel decision fusion system where each local sensor is a sequential decision maker that implements the modified Wald's sequential probability test (SPRT) as proposed by Lee and Thomas (1984). The local decisions are fused at the fusion center by the Chair and Varshney rule (1986). The proposed scheme provides an easy-to-implement decision making architecture using local sequential decision makers. We evaluate the performance of the sensor bank by two criteria: (1) the probability of error; (2) average sample number (ASN) needed to achieve it.

Chair and Varshney (1986) studied parallel decision fusion under the assumption that the local sensor error characteristics (false alarm and mis-detection) are known (fixed) and the local observations are independent conditioned on the hypotheses. They provided an optimal global decision fusion rule that minimizes a Bayes' risk. The Chair-Varshney rule, though widely used in distributed decision making, requires complete knowledge of the prior probabilities of the set of hypothesis (P_0 or P_1) and the error measures (false alarm and mis-detection) of the local detectors. In most applications, these quantities are unknown. Moreover, the local detector performances can be time varying. To address these challenges, we develop, in Chapters 5 and 6, a fusion rule that applies a genetic algorithm to fuse the local-detector binary decisions. The

* *This is joint work with Sayandeep Acharya publish in [39]*

genetic fusion rule does not use the knowledge of prior probabilities, adapts to time varying local sensor error characteristics, and provides near optimal performance (close to the Chair and Varshney rule). It does so at the expense of a larger number of observations and higher computational overhead.

Chapter 7 summarizes the thesis contributions.

2. Background – Parallel Binary Decision Fusion

In binary hypothesis testing, Bayesian detection theory is a widely used method for decision making. It aims at minimizing the Bayesian risk of the binary decision (accept hypothesis H_0 or hypothesis H_1). In system with multiple binary decision-makers observing the same phenomenon, the decisions of the local decision makers are combined to obtain a global decision by a Decision Fusion Center (DFC). The Chair-Varshney fusion rule [1] is a widely used method to combine the decision of fixed binary local detector so as to minimize the Bayesian risk of the DFC.

2.1 Bayesian hypothesis testing

We observe an environment, trying to decide which one of the two hypotheses, H_0 and H_1, it represents. The prior probability of hypothesis H_1 is P_1; the prior probability of hypothesis H_0 is P_0. The decision-maker (sensor) collects observation x from the environment, and accepts either H_0 or H_1 based on the observation. The conditional density function of the observation x under the hypothesis H_i is $P(x|H_i)$, $i = 0, 1$. Four possible situations may occur, as since, when hypothesis H_j $(i = 0, 1)$ is present, the system may accept H_i $(i = 0, 1)$. Two of these four combinations are correct (accept H_i when H_j is true, $i = j$). The others are in error (accept H_i when H_j is true, $i \neq j$). C_{ij} denotes the cost of accepting H_i when hypothesis H_j is true. The Bayesian detection rule minimizes the average cost of the decision, known as the Bayesian risk R.

$$R = \sum_{i=0}^{1} \sum_{j=0}^{1} C_{ij} P_j P\left(\text{Accept } H_i \middle| H_j \text{ is present}\right). \tag{2.1}$$

P_j is the prior probability of hypothesis H_j; C_{ij} is the cost of choosing H_i when H_j is true; $P\left(\text{Accept } H_i \middle| H_j \text{ is present}\right)$ is the probability of the global decision supporting H_i under hypothesis H_j is true. When assuming $C_{00} = C_{11} = 0$ and $C_{01} = C_{10} = 1$, the Bayesian risk becomes the average probability of error.

Assuming $C_{10} > C_{00}$ and $C_{01} > C_{11}$, the minimization results of R in equation (2.1) in the likelihood ratio test [28]

$$\frac{P(x|H_1)}{P(x|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \frac{P_0(C_{10} - C_{00})}{P_1(C_{01} - C_{11})} \tag{2.2}$$

As the natural logarithm is a monotonically increasing function and the two sides of the likelihood ratio test are positive. An equivalent test is

$$\log\left[\frac{P(x|H_1)}{P(x|H_0)}\right] \underset{H_0}{\overset{H_1}{\gtrless}} \log\left[\frac{P_0(C_{10} - C_{00})}{P_1(C_{01} - C_{11})}\right] \tag{2.3}$$

Two types of error may occur. One type of error is miss-detection, defined as accepting H_0, given that H_1 is present. The probability of miss-detection is denoted as P_M. The other type of error is false alarm, defined as accepting H_1, given that H_0 is present. The probability of false alarm is denoted as P_F.

$$P_M = P(\text{Accept } H_0 | H_1 \text{ is present}) \tag{2.4.1}$$

$$P_F = P(\text{Accept } H_1 | H_0 \text{ is present}) \tag{2.4.2}$$

The probability of detection P_D is defined as the probability of accepting H_1, when H_1 is present,

$$P_D = 1 - P_M \tag{2.5}$$

The probability of error of the decision is

$$P(error) = P_1 P_M + P_0 P_F \tag{2.6}$$

2.2 Binary distributed detection

A parallel binary distributed decision fusion system consists of a bank of N local sensors/detector communicating with a fusion center as shown in Fig. 1. The decision fusion center (DFC) combines fixed local detectors. The i^{th} local sensor/detector collects observations from a phenomenon under surveillance and makes binary decisions u_i which are of the following form:

$$u_i = \begin{cases} 1, & H_1 \text{ is true} \\ -1, & H_0 \text{ is true} \end{cases}$$

Here H_1 and H_0 are hypotheses about the state of the phenomenon being observed (H_1 : a target is present; H_0 : a target is not present). There are N local sensors/detectors, so i goes from 1 to N. The local decisions are sent to the decision fusion center (DFC) over error-free channels. The decision fusion center integrates the decisions to a global binary decision $U_g (= 1 \text{ if } H_1 \text{ is accepted}, -1 \text{ otherwise})$.

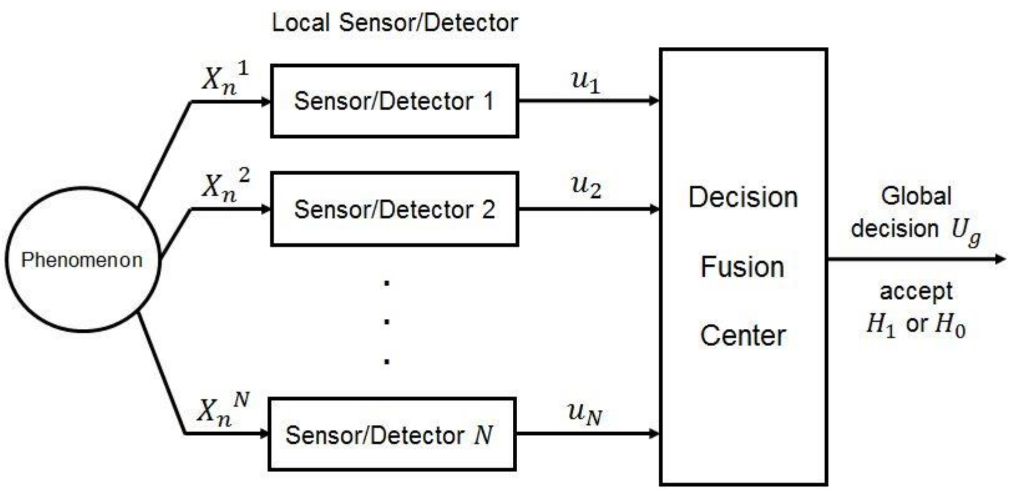


Fig.4. Structure of distributed detection system with parallel sensors

2.3 Chair-Varshney fusion rule

Chair and Varshney [1] developed the optimal fusion algorithm for fixed local sensors/detectors (with known constant local false-alarm and miss-detection rates) and when the local decisions are statistically independent, conditioned on the hypothesis. A generalization of this result under correlated local decisions is available in [2].

The Chair-Varshney fusion rule aims at minimizing the global Bayesian risk, R of global decision U_g (see expression (2.1) and Fig. 4).

The Chair-Varshney fusion rule is given by

$$U_g = \begin{cases} 1, & \sum_{i=1}^{N} w_i u_i > \tau \\ -1, & \sum_{i=1}^{N} w_i u_i < \tau \\ \text{random choose 1 or } -1, & \sum_{i=1}^{N} w_i u_i = \tau \end{cases} \tag{2.7}$$

where $$w_i = \begin{cases} \log \dfrac{1 - P_{Mi}}{P_{Fi}}, & u_i = 1 \\ \log \dfrac{P_{Mi}}{1 - P_{Fi}}, & u_i = -1 \end{cases}$$

$$\tau = \log \left[\frac{P_0 (C_{10} - C_{00})}{P_1 (C_{01} - C_{11})} \right]$$

Here, P_{Fi} and P_{Mi} represent the false alarm rate and missed detection rate of the i^{th} local sensor/detector.

Under the rule in equation (2.7), the fusion system's global false alarm rate P_{Fg} and global missed detection rate P_{Mg} are as [7]:

$$P_{Fg} = \sum_{h_1=0}^{1} \cdots \sum_{h_M=0}^{1} \left| \prod_{i=1}^{N} (h_i - P_{Fi}) \right| U_{-1} \left[-\Delta + \prod_{i=1}^{N} \left(\frac{1 - P_{Mi}}{P_{Fi}} \right)^{1-h_i} \left(\frac{P_{Mi}}{1 - P_{Fi}} \right)^{h_i} \right] \tag{2.8.1}$$

$$P_{Mg} = \sum_{h_1=0}^{1} \cdots \sum_{h_M=0}^{1} \left| \prod_{i=1}^{N} (h_i - P_{Mi}) \right| U_{-1} \left[\Delta - \prod_{i=1}^{N} \left(\frac{1 - P_{Mi}}{P_{Fi}} \right)^{h_i} \left(\frac{P_{Mi}}{1 - P_{Fi}} \right)^{1-h_i} \right] \tag{2.8.2}$$

where, $h_i \in \{0,1\}$. The summation is performed over all possible combinations of local decisions.

When the local sensors are identical, $P_{Fi} = P_F$ and $P_{Mi} = P_M$. The global probability of false alarm becomes [7, 8]:

$$P_{Fg} = \sum_{j=J_f}^{N} \binom{N}{j} {P_F}^j (1 - P_F)^{(N-j)} \tag{2.9.1}$$

The global missed detection rate becomes:

$$P_{Mg} = \sum_{j=J_m}^{N} \binom{N}{j} {P_M}^{j} (1 - P_M)^{(N-j)} \,, \tag{2.9.2}$$

where

$$J_m = int\left\{\frac{N \cdot \log\left(\frac{1-P_M}{P_F}\right) - \log\left(\frac{P_0(C_{10}-C_{00})}{P_1(C_{01}-C_{11})}\right)}{\log\left(\frac{1-P_M}{P_F}\right) + \log\left(\frac{1-P_F}{P_M}\right)}\right\},$$

$$J_f = int\left\{\frac{\log\left(\frac{P_0(C_{10}-C_{00})}{P_1(C_{01}-C_{11})}\right) + N \cdot \log\left(\frac{1-P_F}{P_M}\right)}{\log\left(\frac{1-P_M}{P_F}\right) + \log\left(\frac{1-P_F}{P_M}\right)}\right\},$$

and $int(\cdot)$ is the nearest-integer function, $int(x)$ denotes the largest integer that is smaller than x.

2.4 Decision fusion rule with Neyman-Pearson criterion

Thomopoulos [6] developed a decision fusion rule using the Neyman-Pearson criterion for fixed and independent local sensors/detectors. This decision fusion rule aims at maximizing the global detection rate when an upper bound in the global probability of false alarm is specified. The fusion center implements the Neyman-Pearson criterion to fuse all the local sensor decisions and generate a global decision. The fusion center decision rule formulates a likelihood ratio test:

$$\Lambda(U_g) = \frac{P(u_1, u_2, \dots, u_N|H_1)}{P(u_1, u_2, \dots, u_N|H_0)} = \prod_{i=1}^{N} \frac{P(u_i|H_1)}{P(u_i|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} t \tag{2.10}$$

The threshold t is determined by the global false alarm rate P_{Fg}:

$$\sum_{\Lambda(u)>t^*} P\big(\Lambda(U_g)\big|H_0\big) = P_{Fg} \tag{2.11}$$

In order to implement the Neyman-Pearson criterion, it is needed to compute $P(\Lambda(u)|H_0)$. However, due to the independence assumption of the local sensor, it is easier to obtain the distribution $P\left(\log\wedge\left(U_g\right)\left|H_0\right.\right)$ which can be expressed as the convolution of the individual $P(\log\wedge(u_i)\,|H_0)$:

$$P\left(\log\wedge\left(U_g\right)\left|H_0\right.\right) = P(\log\wedge(u_1)\,|H_0) * \dots * P(\log\wedge(u_N)\,|H_0) \tag{2.12}$$

The likelihood ratio $\wedge(u_i)$ assumes two values. Either $(1 - P_{Di})/(1 - P_{Fi})$ when $u_i = 0$ with probability $(1 - P_{Fi})$ under hypothesis H_0 and probability $(1 - P_{Di})$ under hypothesis H_1; or P_{Di}/P_{Fi}, when $u_i = 1$ with probability P_{Fi} under hypothesis H_0 and probability P_{Di} under hypothesis H_1. Hence, we can write:

$$\begin{aligned} &P\left(\log\wedge\left(U_g\right)\left|H_0\right.\right) \\ &\quad = (1 - P_{Fi})\delta\left\{\log\wedge(u_i) - \log\frac{1 - P_{Di}}{1 - P_{Fi}}\right\} + P_{Fi}\delta\left\{\log\wedge(u_i) - \log\frac{P_{Di}}{P_{Fi}}\right\} \end{aligned} \tag{2.13.1}$$

and

$$\begin{aligned} &P\left(\log\wedge\left(U_g\right)\left|H_1\right.\right) \\ &\quad = (1 - P_{Di})\delta\left\{\log\wedge(u_i) - \log\frac{1 - P_{Di}}{1 - P_{Fi}}\right\} + P_{Di}\delta\left\{\log\wedge(u_i) - \log\frac{P_{Di}}{P_{Fi}}\right\} \end{aligned} \tag{2.13.2}$$

where the (Kronecker) delta function $\delta()$ is defined as

$$\delta(x) = \begin{cases} 0, & x \neq 0 \\ 1, & x = 0 \end{cases}$$

At the fusion center, the global false alarm rate is:

$$P_{Fg} = \sum_{\wedge(u)>t^*} P\left(\wedge\left(U_g\right)\middle|H_0\right) \tag{2.14}$$

where t^* u=is a threshold chosen to satisfy () for a given P_{Fg}. Similarly the global detection rate is:

$$P_{Dg} = \sum_{\wedge(u)>t^*} P\left(\wedge\left(U_g\right)\middle|H_1\right) \tag{2.15}$$

3. Optimal Distributed Decision Fusion using Neyman-Pearson Criterion

This section presents work done in collaboration with Sayandeep Acharya. A paper based on this work was published in the Information Fusion 2013. Ji Wang's contribution is primarily in section 3.1.

3.0 Introduction

Performance of a parallel binary decision fusion architecture is considered where both the local detectors and the Decision Fusion Center use the Neyman-Pearson criterion. The architecture comprises N local detectors in parallel, each sending a binary decision to a fusion center for integration. The algorithm designed here fixes the global false alarm rate and attempts to compute the local detector thresholds and the global fusion rule that achieve the maximum global detection probability. The key computational requirement is to find the roots of a certain Nth order polynomial. We compare the performance of our method with the performance of the Person-by-Person Optimization (PBPO) approach and that of a centralized detection scheme.

3.0.1 The scheme

A group of N local detectors observe a phenomenon. Each local detector decides whether to accept one of two binary hypotheses, and it transmits its decision to a fusion center. Thomopoulos provided a general proof that the optimal decision scheme that maximizes the probability of detection at the fusion center for a fixed false alarm rate consists of a Neyman-Pearson test at the fusion center and likelihood tests at the local detectors.

In [9] and [10], a Person-by-Person-Optimization (PBPO) approach was used for designing the entire system (local and DFC decision rules), which in general is not guaranteed to achieve system wide optimality [11]. The proposed PBPO solutions required simultaneous solution of non-linear coupled equations for decision thresholds, which can become difficult as the number of equations increases.

3.0.2 Objectory

In this study, we attempt to achieve the optimal solution for the entire system when the maximum global probability of false alarm is specified, and when the DFC and the local detectors are designed so that the global probability of detection is maximized. We assume that the local detector observations are independent conditioned on the hypothesis. The principal effort in the design turns out to be to solve for the roots of a certain Nth order polynomial.

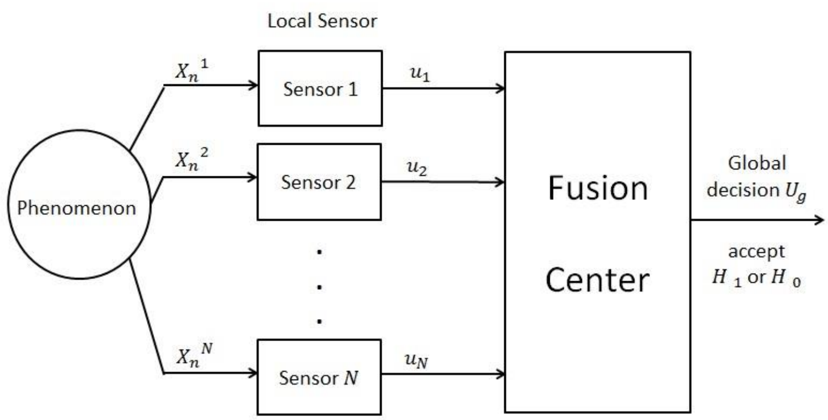


Fig.5. Structure of distributed detection system with parallel sensors/detectors

3.1 Decentralized Neyman-Pearson Decision Fusion

The Neyman-Pearson test fixes the global false alarm rate $P(u_0 = 1|H_0)$ at a pre-specified level $a < 1$ and then attempts to achieve the maximum global probability of detection $P(u_0 = 1|H_1)$. The fusion center decision rule becomes a likelihood ratio test, and takes the form

$$\Lambda(U_g) = \frac{P(u_1, u_2, \dots, u_N|H_1)}{P(u_1, u_2, \dots, u_N|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} t^*$$

The threshold t^* is computed such that the global false alarm does not exceed a. Assuming that the local decisions are independent (conditioned on the hypothesis), we have

$$\Lambda(U_g) = \prod_{i=1}^{N} \frac{P(u_i|H_1)}{P(u_i|H_0)} = \prod_{i=1}^{N} \Lambda(u_i) = \underset{H_0}{\overset{H_1}{\gtrless}} t^*$$

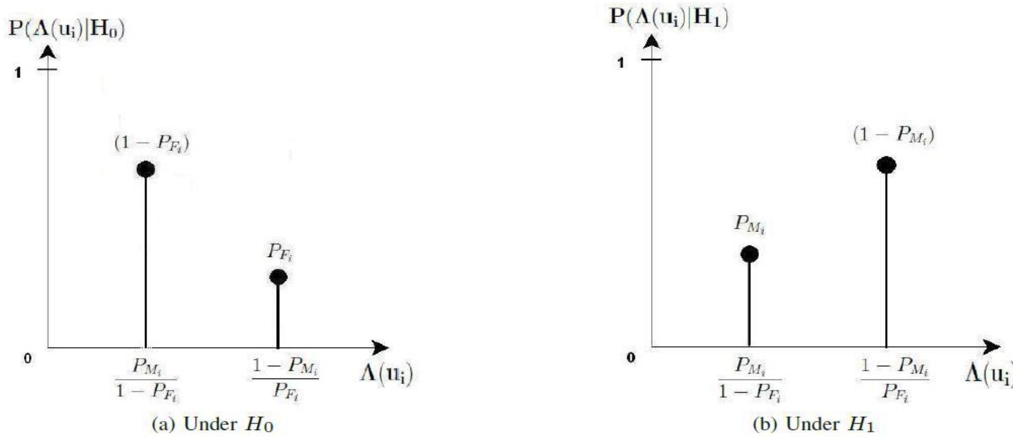


Fig. 6. Probability mass function of local sensor/detector likelihood

As the local decisions u_i are binary, the conditional probability distributions $P(\wedge(u_i)|H_0)$ and $P(\wedge(u_i)|H_1)$ for the i^{th} detector likelihood ratio are discrete as shown in the Fig. 6. Let us consider a decision fusion system consisting of N identical local detectors with local false alarm and missed detection rates given respectively as $P_{Fi} = p$ and $P_{Mi} = q$, with $p, q \in (0,1), i = 1, \dots, N$. The global conditional distributions of the likelihood ratio $\wedge(u)$ can be expressed using the binomial distributions

$$P(\wedge(u_i)|H_1) = \sum_{k=0}^{N} \binom{N}{k} (1-q)^k (q)^{N-k} \left[\delta \left\{\wedge(u) - \left(\frac{q}{1-p}\right)^{N-k} \left(\frac{1-q}{p}\right)^k\right\}\right] \quad (3.1.1)$$

and

$$P(\wedge(u_i)|H_1) = \sum_{k=0}^{N} \binom{N}{k} (p)^k (1-p)^{N-k} \left[\delta \left\{\wedge(u) - \left(\frac{q}{1-p}\right)^{N-k} \left(\frac{1-q}{p}\right)^k\right\}\right] \quad (3.1.2)$$

In the case of N identical detectors, the distributions in (3) and (4) will have $N + 1$ probability masses. Let us index them by $k = 0,1, \dots, N$. An arbitrary global false alarm probability $P_{FA}^{G} = a$ can be realized as a convex combination

$$a = (1-\gamma)\sum_{k=k'}^{N}\binom{N}{k}(p)^k(1-p)^{N-k} + \gamma\sum_{k=k'-1}^{N}\binom{N}{k}(p)^k(1-p)^{N-k} \tag{3.2}$$

Where k' is the smallest value of $k \in [0,1,\dots,N]$ such that

$$a > \sum_{k=k'}^{N}\binom{N}{k}(p)^k(1-p)^{N-k}$$

And the parameter $\gamma \in [0,1]$ is given by

$$\gamma = \frac{a - \sum_{k=k'}^{N}\binom{N}{k}(p)^k(1-p)^{N-k}}{\sum_{k=k'-1}^{N}\binom{N}{k}(p)^k(1-p)^{N-k} - \sum_{k=k'}^{N}\binom{N}{k}(p)^k(1-p)^{N-k}} \tag{3.3}$$

The global probability of detection then becomes

$$P_{Dg} = (1-\gamma)\sum_{k=k'}^{N}\binom{N}{k}(1-q)^k(q)^{N-k} + \gamma\sum_{k=k'-1}^{N}\binom{N}{k}(1-q)^k(q)^{N-k} \tag{3.4}$$

We are looking for the value of the likelihood ratio $\Lambda(u)$ such that the sum of all the probability masses at and to the right of $\Lambda(u)$ is equal to a. For identical detectors, the fusion rule is always k out of N. In this case the desired $\Lambda(u)$ must satisfy the following

$$\Lambda(u) = \left(\frac{q}{1-p}\right)^{N-k}\left(\frac{1-q}{p}\right)^{k} \tag{3.5}$$

for some $k \in [0,1,\dots,N]$. In that scenario, the parameter γ reduces to either 0 or 1. Let k^* denote the value of k that satisfies (5). The global false alarm probability becomes

$$P_{Fg} = a = \sum_{k=k^*}^{N}\binom{N}{k}(p)^k(1-p)^{N-k} \tag{3.6}$$

And the corresponding global

$$P_{Dg} = a = \sum_{k=k^*}^{N} \binom{N}{k} (1-q)^k (q)^{N-k} \qquad (3.7)$$

The system-wide optimal solution is therefore the pair (p, k^*) obtained by solving (6) for p for every k^* and then choosing the pair that maximizes global detection rate. Since $a < 1, k^* \neq 0$. Noting that for a fixed k^*, the summand in (3.6) is a monotonically increasing function of p, the solution for p in (3.6) in the feasible region of $(0,1)$ is unique. Therefore, if the N roots of the equation

$$\sum_{k=k^*}^{N} \binom{N}{k} (p)^k (1-p)^{N-k} - a = 0 \qquad (3.8)$$

are evaluated for every k^*, there would be up to N distinct solutions (one for each k^*). Each one of these solutions would correspond to a value of the global probability of detection (from (3.7)). The optimal local false alarm rate would then be the one that provided the maximum global probability of detection.

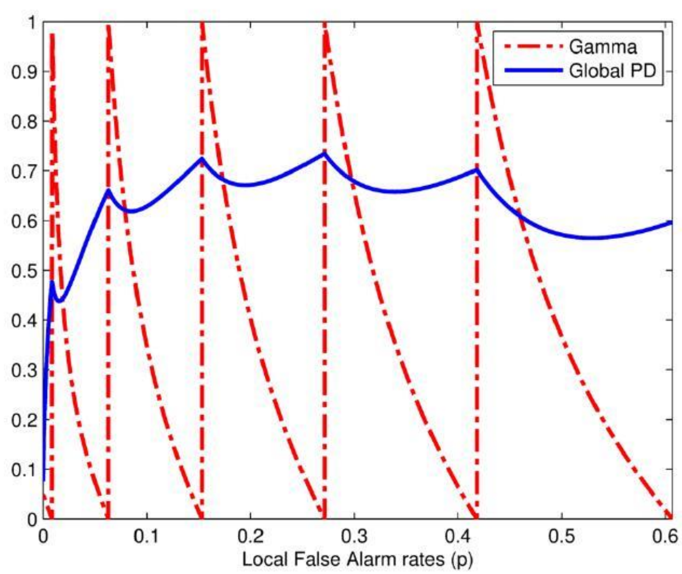


Fig. 7: Variation of global probability of detection P_D^G and γ with local sensor/detector false alarm rate p for identical sensor/detector

In Fig. 7 we show an example of fusion system with 6 local detectors. The global false alarm rate is chosen to be $a = 0.05$. In Fig. 7 we show the variation of the global probability of detection as the local detector false alarm rate (p) is varied. It is notable that the curve representing the global probability of detection has cusps at the locations where switches between its maximum and minimum values (implying a change in k^*); it is not differentiable there. The maximum global detection rate is obtained from the N global detection rates corresponding to the N feasible p values obtained by solving (6) for every k^*.

The computational burden involved with this approach is to compute roots of the N^{th} order univariate polynomial (3.8). We summarize the proposed algorithm as below.

<u>*Optimal Distribution Fusion Algorithm:*</u>

(1) For N detector, consider possible values of k^* in the range $[1, 2, \cdots, N]$

(2) Solve for the roots of (3.8) for each value of k^*. For each k^* there will be up to N distinct roots. Let the root which is in the feasible region of $[0, 1]$ for a particular k^* be denoted by $p_a(k^*)$.

(3) Assuming the local detector observations have a continuous distribution, compute the corresponding local detector threshold t_{loc} using $p_a(k^*)$ as follows

$$\int_{t_{loc}}^{\infty} P(\Lambda(u_i)|H_0) = p_a(k^*)$$

(4) Compute the corresponding local mis-detection rate $q_a(k^*)$ as follows

$$\int_{-\infty}^{t_{loc}} P(\Lambda(u_i)|H_1) = q_a(k^*)$$

(5) For each possible value of k^*, namely 1, 2, …, N, compute the global probability of detection $P_{Dg}(k^*)$ using (3.7) with $q = q_a(k^*)$

(6) Find the value of (k^*) that provide the maximum value of $P_{Dg}(k^*)$

(7) The corresponding $p_a(k^*)$ is the local false alarm for the local detector that would provide the best global detection rate for the maximum global false alarm of a.

In the following section, we briefly discuss the Person-By-Person-Optimization (PBPO) approach frequently used for a distributed decision fusion system design and then proceed to compare the performance of the PBPO approach with the proposed scheme through numerical examples.

3.2 Person-By-Person Optimization

A general method for seeking system-wide design of a decision fusion system is through the Person-By-Person- Optimization (PBPO) method. The distributed detection system is viewed as a team of two members. The group of local detectors forms one member and the DFC is the other member. Performance of each member of the team is optimized separately with the assumption that the other member has already been optimized. This approach requires simultaneous solution of nonlinear coupled equations for local detector thresholds and the global fusion rule. Still, the PBPO optimal solution is not guaranteed to achieve the true team optimum [11]. For an N detector binary decision fusion system with non-identical detectors, the PBPO solution is obtained by simultaneous solution of $N + 2^N$ nonlinear coupled equations [9], [10]. When the local detectors are identical and the observations at the local detectors are independent conditioned on the hypothesis, the number of equations for PBPO approach under Neyman-Pearson criterion drops down to 3. Next we outline the PBPO solution for identical detectors using the Neyman-Pearson criterion.

3.2.1 PBPO-Optimal Local Detector threshold

Under the Neyman-Pearson criterion, the global probability of detection is maximized under the constraint that the global false alarm satisfies $P_{Dg} \leq a$. We therefore form the objective function to be maximized as

$$F = P_{Dg} + \lambda\left(P_{Fg} - a\right) \tag{3.8}$$

where λ is the Lagrange multiplier. Using (3.6) and (3.7), we have

$$F = \sum_{k=k^*}^{N} \binom{N}{k} (1-q)^k (q)^{N-k} + \lambda \left[\sum_{k=k^*}^{N} \binom{N}{k} (p)^k (1-p)^{N-k} - a \right] \tag{3.9}$$

Expanding (3.9) in terms of q and p, the probability of misdetection and false alarm of a local detector respectively, we have

$$F = (1-q)\sum_{k=k^*}^{N}\binom{N}{k}(1-q)^{k-1}(q)^{N-k} + \lambda\left[p\sum_{k=k^*}^{N}\binom{N}{k}(p)^{k-1}(1-p)^{N-k} - a\right] \quad (3.10)$$

Let us define the expressions

$$V_p = \sum_{k=k^*}^{N}\binom{N}{k}(p)^{k-1}(1-p)^{N-k}$$

and

$$V_q = \sum_{k=k^*}^{N}\binom{N}{k}(1-q)^{k-1}(q)^{N-k}$$

The expression in (3.10) becomes

$$F = (1-q)V_q + \lambda\big(pV_p - a\big) \quad (3.11)$$

Since V_q is sum of positive real numbers, $V_q \neq 0$. Hence we have

$$F_q = \frac{F}{V_q}(1-q) + \frac{\lambda V_p}{V_q}\left(p - \frac{a}{V_p}\right) \quad (3.12)$$

Maximizing F_q implies that each local detector maximizes its own probability of detection $(1-q)$ subject to the constraint that its local false alarm is bounded as $p \leq \frac{a}{V_p}$. Each local detector performs a likelihood ratio test as

$$\frac{P(r_i|H_1)}{P(r_i|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} t_{loc} \quad (3.13)$$

where r_i are the i^{th} observations for the detector and the local threshold t_{loc} is computed such that the local false alarm is fixed at $p = \frac{a}{V_p}$. In other words, (3.12) becomes the Lagrangian for a local detector and under the Neyman-Pearson criterion, is given by the Lagrange multiplier therefore implying $t_{loc} = \frac{\lambda V_p}{V_q}$, where λ is the threshold for the global likelihood ratio test. From the local detector optimization, we obtain

$$t_{loc} = \frac{\lambda V_p}{V_q} \tag{3.14}$$

$$p = \frac{a}{V_p} \tag{3.15}$$

3.2.2 PBPO-Optima Global fusion rule

Since the local detectors are identical, the global fusion rule is a k out of N rule [9]. The optimal k (denoted by k^*) can be obtained by noting that the Lagrange multiplier in (3.8) is effectively the threshold of the global likelihood ratio test or the value of $\Lambda(u)$ at which (3.5) is satisfied, and therefore

$$\left(\frac{q}{1-p}\right)^{N-k}\left(\frac{1-q}{p}\right)^{k} = \lambda$$

Taking natural logarithm of both sides , we have

$$k^*\left[\log\left(\frac{1-q}{p}\right) - \log\left(\frac{q}{1-p}\right)\right] = \log(\lambda) - N\log\left(\frac{q}{1-p}\right)$$

Since the constraint is $P_{Fg} \leq a$, we can express the optimal k^*

$$k^* = \left\lceil \frac{\log(\lambda) - N\log\left(\frac{q}{1-p}\right)}{\log\left(\frac{1-q}{p}\right) - \log\left(\frac{q}{1-p}\right)} \right\rceil \tag{3.15}$$

where $\lceil \cdot \rceil$ is the ceiling function defined over the set of integers (Z) as

$$\lceil x \rceil = min\{s \in Z | s \geq x\}$$

The complete PBPO solution for identical detectors under Neyman-Pearson criterion therefore requires the simultaneous solution of the coupled nonlinear equations (3.13), (3.14) and (3.15).

In general the PBPO solution does not converge to the team optimum solution. Bauso and Pesenti in [11] showed that the necessary and sufficient condition for a PBPO solution to converge to the team optimum is satisfied when the team cost function has a unique local minimum. This is not the general case for the problem we study, as shown, for example, in Fig. 7 where the global probability of detection is not unimodal with respect to the local detector false alarm rate, p.

The PBPO ROC is a collection of different ROC curves (each corresponding to a different value of the optimal k^*); the collective PBPO ROC is formed using the upper envelopes of each of those constituent ROC curves. Due to this feature, even though the ROC curves corresponding to any particular k^* is concave, the overall PBPO ROC curve is not concave as the PBPO optimal k^* changes.

3.3 Examples and Discussion

We provide a performance comparison of our method with the PBPO approach using ROC curves for several scenarios. We also include the performance of a centralized fusion scheme, where the data fusion center receives the raw observations and computes the global decision with no involvement of local detectors. Since the centralized architecture performs no local data compression, it provides an upper bound on the performance of a parallel fusion system. In our scenario for the centralized architecture, the fusion center receives $K_c = \mathrm{NK}$ observations and uses a Neyman-Pearson test with specified false alarm probability to arrive at a decision. We consider the following three cases:

(1) Observations are Gaussian distributed with different means under the two hypotheses, namely:

$$P(r_i|H_1) \sim N(m, \sigma^2)$$
$$P(r_i|H_0) \sim N(0, \sigma^2)$$

ROCs are shown in Fig. 8 for three systems, namely: (a) Distributed detection using optimal distributed fusion algorithm; (b) Distributed Neyman-Pearson detection using PBPO; and (c) Centralized detection. Four different values were used for m, namely

3,4,5 and 6. Fig. 8 shows the extent to which the optimum detection scheme improves over PBPO. The centralized system is of course better than both.

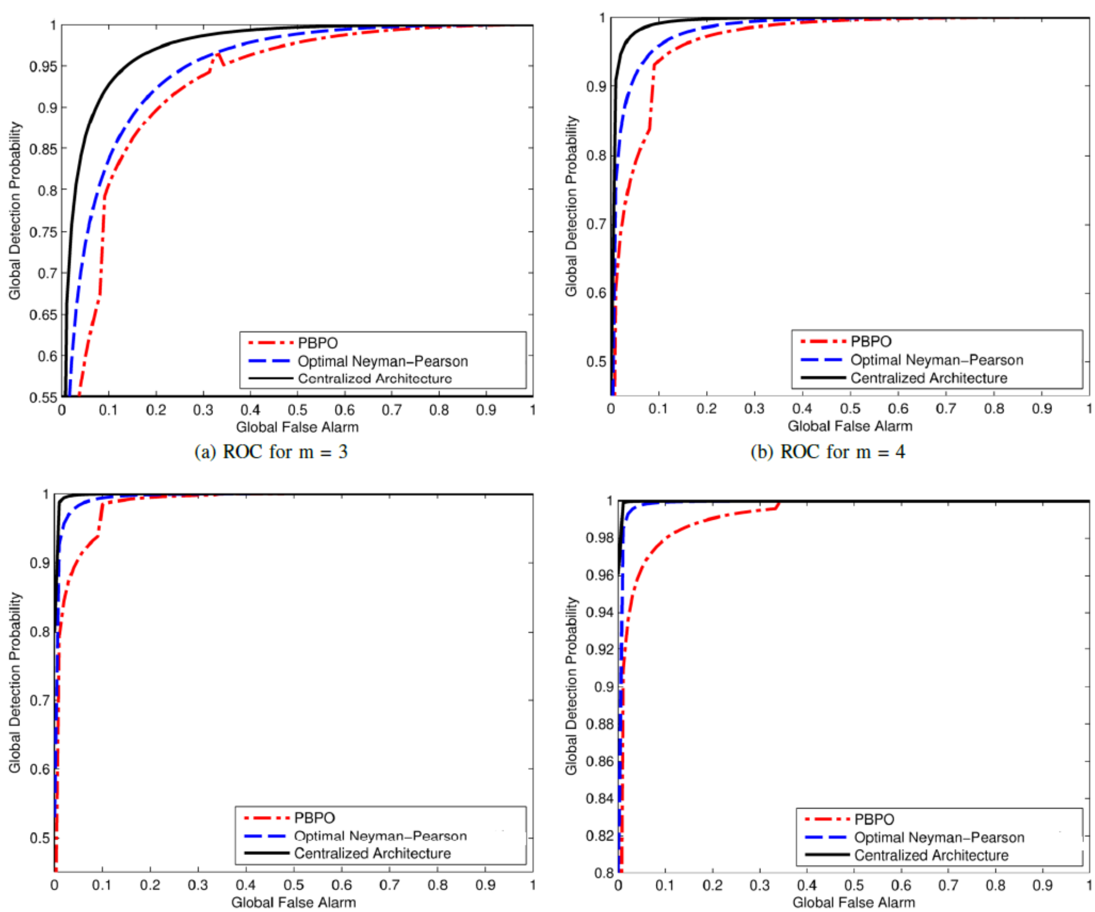
(2) Observations are Exponentially distributed,

$$P(r_i|H_1) \sim Exp(1/2)$$
$$P(r_i|H_0) \sim Exp(1/3)$$

(3) Observations are Gamma distributed,

$$P(r_i|H_1) \sim Gamma(1,2)$$
$$P(r_i|H_0) \sim Gamma(1,1)$$

Fig. 9 shows the system's ROC curves for the Exponential and Gamma distributions, and documents the improvement provided by the optimal algorithm. Depending on the distribution of the local detector observations, some values of global false alarm may not have a corresponding PBPO solution. Several such regions are noticed in Fig. 9.



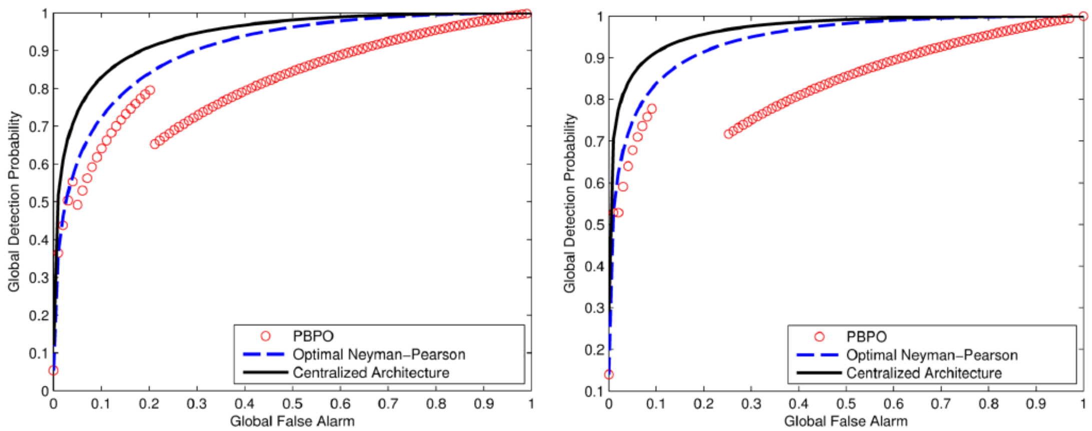
(a) ROC for m = 3

(b) ROC for m = 4

(c) ROC for m = 5

(d) ROC for m = 6

Fig. 8: ROC curves under various SNR for distributed Neyman-Pearson detection using optimal distributed fusion algorithm; distributed Neyman-Pearson detection using PBPO; and centralized Neyman-Pearson detection.



(a) Exponential Distribution (b) Gamma Distribution

Fig. 9: Performance comparison of the three systems when local detector observations are Exponential and Gamma Distributed.

We considered system-wide optimization of a distributed decision fusion system where a group of local sensor/detectors perform binary hypothesis testing on observations from a common volume of surveillance and communicate their decisions to a decision fusion center. The objective is to maximize global probability of detection under a global probability of false alarm constraint. The local detector decision thresholds and the global fusion rule were derived by computing the roots of an N^{th} order polynomial where N is the number of local detectors. The proposed method was compared against the PBPO approach. ROC curves of several scenarios demonstrate the extent to which the optimal solution outperforms PBPO, and the extent to which it is over performed by a centralized detection scheme (where all the raw observations are transmitted to the data fusion center).

4. Decision Fusion for Parallel Sequential Sensors

4.0 Introduction

Lee and Thomas (1984) have introduced a modified version of Wald's sequential probability ratio test. The modified version retains most of the features of Wald's procedure but is easier to analyze and offers efficient truncation procedures. In this study, we use the Lee-Thomas design to analyze the performance of a bank of N parallel sequential sensors whose decisions are fused. We evaluate the performance of the sensor bank by two criteria: (1) the probability of error; (2) average sample number (ASN) needed to achieve it. Three rules are studied: (1) first-to-decide rule (Niu and Varshney, 1984): once at least one sensor has stopped sampling, we adopt the decision of one of the stopped sensors; (2) all-that-decided rule: once at least one sensor has stopped sampling, we integrate all the decisions of stopped sensors through the 1986 Chair-Varshney decision fusion rule; and (3) all-sensors rule: once at least one sensor has stopped sampling, we combine the available decisions of the stopped sensor and the implied decisions of the remaining sensors. Performance of the three rules is calculated and gains with respect to the performance of a single sensor are quantified.

Sequential detection procedures find applications in several areas of research. Lee and Thomas have proposed [5] a modification to Wald's Sequential Probability Ratio Test (SPRT) [3]. Wald's procedure was in turn a significant improvement over previous fixed-sample-size (FSS) detection methods. The Lee-Thomas procedure, titled the *memory-less grouped-data sequential* (MLGDS) procedure, tests a simple hypothesis against a simple location alternative, based on *n* independent and identically distributed samples. Specifically, at each stage, the S_0 consecutive previous samples are taken, and a test statistic based on them is calculated. A two-threshold test is then made. If the test statistic is above the higher threshold or below the lower threshold, a decision is made. Otherwise the S_0 samples are discarded, and the next S_0 samples are collected for calculating the next test statistic. Lee-Thomas MLGDS procedure exhibits simplicity in structure and analysis and retains most of the features of Wald's SPRT.

We present a fusion rule for fusing N isolated and identical sensors that use the optimized MLGDS detection procedure. Once at least one sensor has stopped sampling, we combine the available decisions of the stopped sensor and the implied decisions of the remaining sensors to get a global decision using Chair-Varshney decision fusion rule.

The architecture is shown in Fig. 10. Each local sensor collects information about a phenomenon they observe, and make binary decisions based on this information. The decision is to accept the hypothesis H_0 or accept hypothesis H_1. The decisions follow the MLGDS procedure. They are transmitted to a *Fusion Center*, where they are integrated to generate the system's global binary (H_0 or H_1) decision.

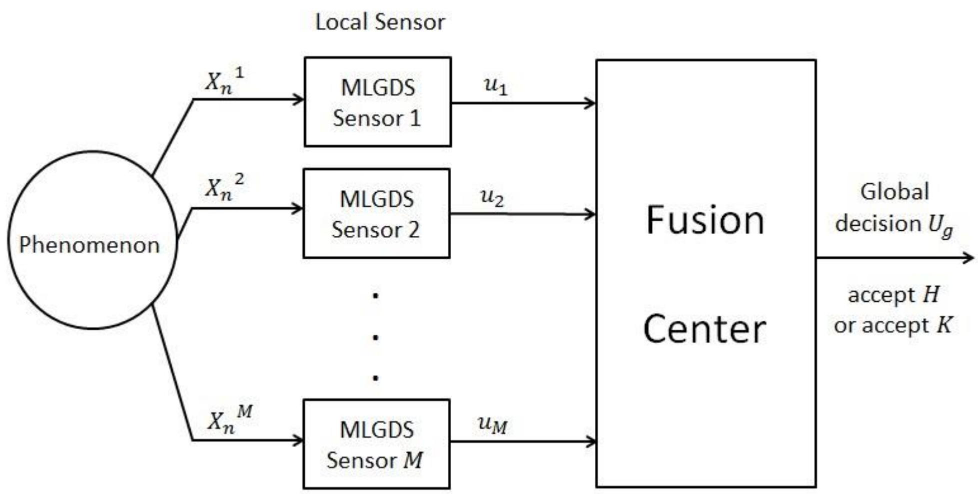


Fig. 10. Structure of distributed detection system with parallel sequential sensors

4.1 Lee and Thomas Modified Sequential Detection Rule

In [1], Lee and Thomas have studied, using a single sensor, a procedure that tests the hypothesis H_0

$$H_0:\ X_i = S_i + \theta_0$$

versus the alternative hypothesis H_1

$$H_1:\ X_i = S_i + \theta_1$$

where $\theta_1 - \theta_0 = \theta > 0$, θ_1 and θ_0 are real numbers; N_i are normally and independently distributed each with mean μ and variance σ.

<u>MLGDS procedure</u> [1]: At the n^{th} stage, using the previous S_0 samples, form a test statistic where

$$X_s = (X_{(s-1)S_0+1}, X_{(s-1)S_0+2}, \cdots, X_{sS_0})$$

$$T_{S_0}(X_s) = \sum_{i=(s-1)S_0-1}^{sS_0} X_i$$

and make the following decisions:

$$T_{S_0}(X_s)\begin{cases} > A \rightarrow H_1 \\ < B \rightarrow H_0 \\ B < T_{S_0}(X_s) < \mathrm{A} \rightarrow R \end{cases}$$

Here *R* = discard all used samples and proceed to the next $(s+1)^{st}$ stage.

A and *B* are testing thresholds with $A > B$. They are predetermined so as to achieve the desired test level and power, and to satisfy other conditions, mostly simplifying the procedure.

One possible choice for the two thresholds (for the particular case we study) is proposed in [1] – the thresholds are placed symmetrically about $S_0(\mu + \theta/2)$. A and B then become

$$A = S_0(\mu + \theta/2) + kS_0\theta$$

$$B = S_0(\mu + \theta/2) - kS_0\theta$$

where k is a parameter, which is a positive real number.

If we use these symmetric thresholds A and$B$, it follows that$P\{T_{S_0} > A \middle| H_0\} = \mathrm{P}\{T_{S_0} < B \middle| H_1\}$ and $P\{T_{S_0} < A \middle| H_0\} = \mathrm{P}\{T_{S_0} > B \middle| H_1\}$.

The probability of error p at the s^{th} stage is

$$p = P\{T_{S_0} > A \middle| H_0\}P_0 + \mathrm{P}\{T_{S_0} < B \middle| H_1\}P_1$$

The probability of detection q at the s^{th} stage is

$$q = P\{T_{S_0} < B \middle| H_0\}P_0 + \mathrm{P}\{T_{S_0} > A \middle| H_1\}P_1$$

Here P_K is the a *priori* probability of hypothesis K; P_H is the a *priori* probability of hypothesis H.

The probability that no decision was made at the n^{th} stage (and hence we proceed to the $(n+1)^{st}$ stage) is

$$r = P\{B < T_{S_0} < \mathrm{A} | H\} P_0 + P\{B < T_{S_0} < \mathrm{A} | H\} P_1 = 1 - q - p$$

The overall error rate e of the sequential sensor is therefore

$$e = p + rp + r^2 p + \cdots = \frac{p}{1-r} = \frac{p}{p+q}$$

The corresponding average sample number ASN_1 is:

$$ASN_1 = S_0 + rS_0 + r^2 S_0 + \cdots = \frac{S_0}{1-r} = \frac{S_0}{p+q}$$

4.2 Fusion Rule for Lee-Thomas Sequential Sensors

For the fusion of Lee-Thomas sequential sensors, we propose three fusion rules: First-to-decided rule, All-that-decided rule, All-sensor rule.

4.2.1 First-to-decided rule

Under this rule, the Fusion Center accepts the first decision that one of these M sensors reported as the global decision, provided of course that at least one of them stopped (namely reached a decision). If exactly one sensor stopped, the Fusion Center accepts its decision as global.

Under this rule, since only one sensor's decision is used as the global decision, the overall error rate will be the error rate of a single sensor.

The probability at least one sensor stopped at the Z^{th} stage is:

$$P^{(Z)} = P(min(N) = Z \times S_0) = r^{(Z-1)N} - r^{ZN},$$

where, Z is a positive integer. N is the number of samples sampled by each of the parallel M sensors before the system stopped.

The global average sample number in this case is

$$ASN_g = E\big(min(N)\big) = \sum_{Z=1}^{\infty} \{(ZS_0) P^{(Z)}\} = \frac{S_0}{1-r^N}$$

Since $r < 1$, it follows that

$$ASN_g = \frac{S_0}{1 - r^N} < ASN_1 = \frac{S_0}{1 - r}$$

and

$$\frac{ASN_g}{ASN_1} = \frac{1 - r}{1 - r^N}$$

$$\text{If } N \to \infty, \lim_{N \to \infty} ASN_g = (1 - r) ASN_1$$

If more than one of the sensors stopped at the end of the stage, the Fusion Center randomly chooses one of the stopping sensors' decision to provide the global decision. For a certain 6-sensor system, Fig. 11 shows the probability that k sensors stopped simultaneously, where $k = 1 \ldots 6$. The probability of k >1 is higher than the probability that k = 1. Furthermore, the larger the number of local sensors used by the system, the higher the probability of having multiple sensors stop simutaneously at the end of the stage. This observation suggests the next rule, which we refer to as the *All-that-decided* fusion rule.

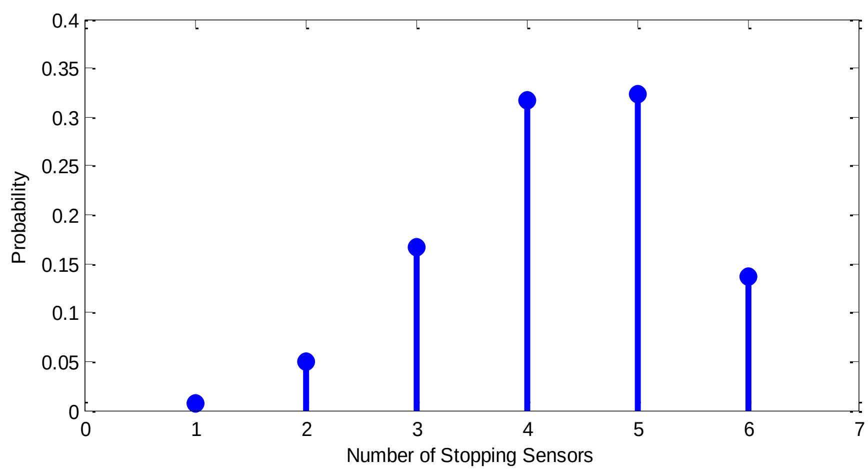


Fig. 11. A 6 sensor system is considered. The probability is shown that when at least one sensor stopped at the end of a stage, the number of simultaneously stopping sensors was 1, 2, 3, 4, 5 or 6.

4.2.2 All-that-decided rule

Under this rule, when at least one sensor has stopped, all the sensors that reached a decision simultaneously are taken into account for calculating the global decision. As these sensors that reached a decision either accept H_0 or accept H_1, we use the Chair-Varshney binary decision fusion rule to integrate these decisions.

The probability (P_n) that exactly n first-stop sensors out of M local sensors reached a decision is

$$P_n = \sum_{Z=1}^{\infty} P_n^{(Z)} = \frac{1}{1-r^N}\binom{N}{n}(1-r)^n \cdot r^{(N-n)}$$

Here $P_n^{(Z)}$ represents the probability that exactly m first-stopped sensors reached a decision at the Z^{th} stage.

As local sensors are assumed to be identical, if exactly m sensors have reached a decision, the system's error rate can be calculated by using equations (12a-12d), namely

$$e_n = P_1 \sum_{j=J_m}^{n} \binom{n}{j} {P_{Mi}}^j (1-P_{Mi})^{(n-j)} + P_0 \sum_{j=J_f}^{n} \binom{n}{j} {P_{Fi}}^j (1-P_{Fi})^{(n-j)}$$

The global error rate is

$$e_{g-all-that-decided} = \sum_{n=1}^{N} P_n e_n$$

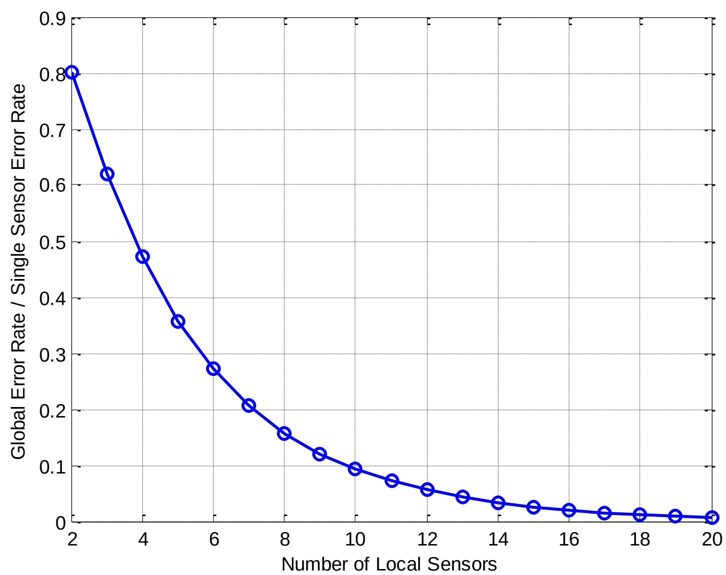


Fig. 12. $e_{g-all-that-decided}$ $/e$ ($e_{g-all-that-decided}$ is the global error rate of All-that-decide fusion system, e is the error rate of a single sensor), for the All-that-decided sequential fusion rule vs. the number of local sensors

While the *All-that-decided* rule makes use of all available decisions of the local sensors, there is still unused information in the system, namely the sampled data of all the sensors that have not yet reached a decision. In the next section, we will introduce an *All-sensors* fusion rule that also takes into account this sampling data by implying a decision for the unstopped sensors.

4.2.3 All-sensor rule

If there are at least one of local sensors stopped and reached a decision, fusion center forces all the sensors that have not stopped yet to provides their implied decisions. Then, the fusion center collects and integrates the decisions of all the sensors that stopped and the implied decisions of all the sensors that haven't stopped. All local sensors are redesigned to have an additional threshold C:

$$C = S_0\left(\mu + \frac{P_0(C_{10} - C_{00})}{P_1(C_{01} - C_{11})} \cdot \frac{\theta}{2}\right)$$

The following rule is used for the sensors that have not stopped:

$$T_{S_0}(X_s)\begin{cases} > C \rightarrow H_1 \\ < C \rightarrow H_0 \end{cases}$$

The threshold C is selected so that the undecided sequential sensors minimize the Bayesian risk [Section II-2]. When we minimize the probability of error, the threshold becomes

$$C = S_0\left(\mu + \frac{P_0}{P_1} \cdot \frac{\theta}{2}\right)$$

With this third threshold C all sensors provide a binary decision to the Fusion Center.

For the sensors that stopped

$$P_{Fi} = P\{decide\ K|H\}^{(A,B)} = P\{T_{S_0} > A \middle| H_0\}$$

$$P_{Mi} = P\{decide\ H|K\}^{(A,B)} = \mathrm{P}\{T_{S_0} < B \middle| H_1\}$$

For the sensors that that did not stop but provided an implied decision

$$P_{Fi} = P\{decide\ K|H\}^{(C)} = P\{T_{S_0} > C \middle| H_0\}$$

$$P_{Mi} = P\{decide\ H|K\}^{(C)} = \mathrm{P}\{T_{S_0} < C \middle| H_1\}$$

We can now use Chair-Varshney fusion rule to integrate all decisions.

The sensors that did not stop and provided an implied decision have a higher error rate than those sensors that use threshold A and B to decide. With the same probability, still, the sensors that use threshold C have better performance than 'pure-chance sensor'. The pure chance sensors simply decide H_0 or H_1 without using information from statistics with an error rate of 0.5.

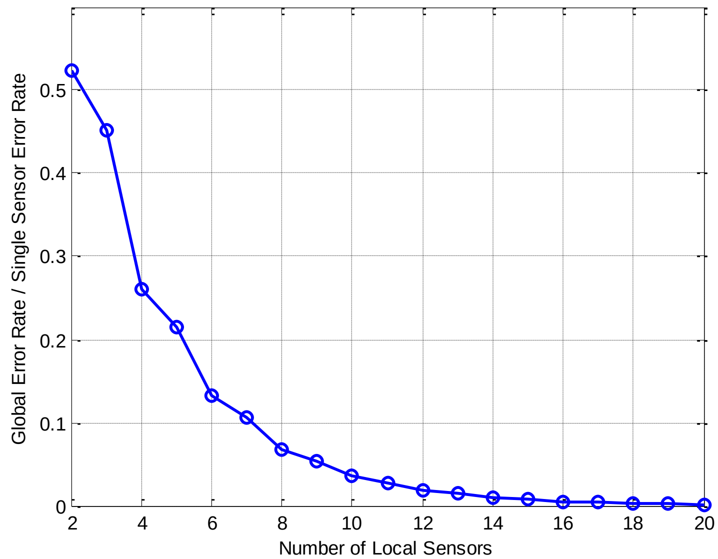


Fig. 13. $e_{g-all-sensors}$ $/e$ ($e_{g-all-sensors}$ is the global error rate of the All-sensors fusion system, e is the error rate of a single sensor) for the All-sensors sequential rule vs. the number of sensors.

Consider a single MLDGS sensor with error rate of 0.0128 and ASN of 986. Table 1, Fig. 13, and Fig. 14 show the performance of a All-sensors sequential fusion rule system as the number of local sensors increases. Table 1, Fig. 13 and Fig. 14 show how the ratio of the global error rate to a single sensor error rate and the ratio of the global ASN to a single sensor ASN are improved by applying the All-sensors sequential fusion rule. They also show the comparison with the performance of a single sensor. As the number of local sensors increases, the ratio of the global error rate to the error rate of a single sensor decreases (and approaches 0). In Fig. 14, as the number of sensors is increased from 2 to 20, the ratio of the global ASN to a single sensor decreases and then settles at $(1-r)ASN_1$.

.

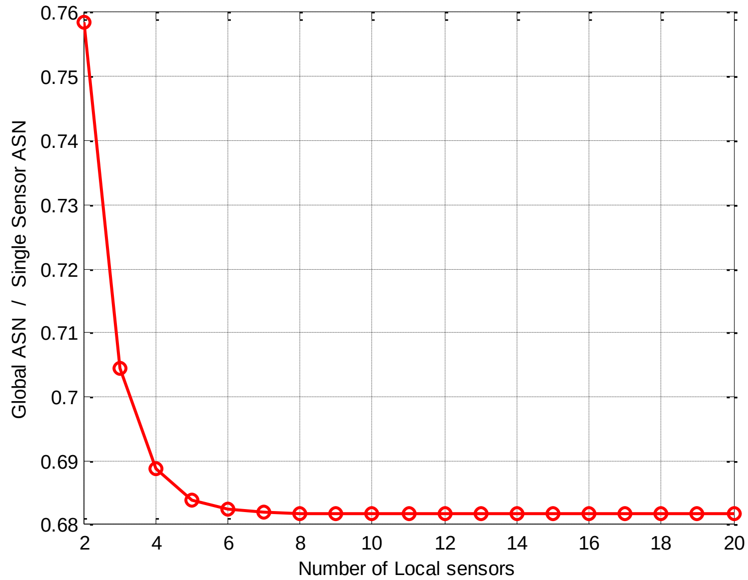


Fig. 14. ASN_g/ASN_1 (ASN_g is the global ASN of fusion system, ASN_1 is the ASN of one sensor) for the All-that-decided sequential fusion rule vs. the number of local sensors.

TABLE I. PERFORMANCE OF MULTIPLE SENSORS FUSION SYSTEM APPLIED ALL-SENSORS FUSION RULE

Number of Sensors	**Global Error Rate e_g**	$\frac{e_g}{e}$	**Global ASN ASN_g**	$\frac{ASN_g}{ASN_1}$
2	0.0067214	0.52347	747.93	0.75855
3	0.0057904	0.45097	694.55	0.70441
4	0.0033502	0.26092	679.12	0.68876
5	0.0027507	0.21423	674.35	0.68393
6	0.0016968	0.13215	672.85	0.68240
7	0.0013532	0.10539	672.37	0.68192
8	0.0008725	0.06795	672.22	0.68176
9	0.0006818	0.05310	672.17	0.68171

10	0.0004536	0.03533	672.16	0.68170
11	0.0003493	0.02720	672.15	0.68169
12	0.0002378	0.01852	672.15	0.68169
13	0.0001811	0.01411	672.15	0.68169
14	0.0001254	0.00977	672.15	0.68169
15	0.0000948	0.00738	672.15	0.68169

We designed a fusion rule for Lee-Thomas sequential sensors. We compared the performance in terms of the global error rate and Average Sampling Number (ASN). This sequential fusion rule can make use of all the sensors in the system, even some sensor haven't reach a decision. We add a new threshold for Lee-Thomas sequential sensors. With this new threshold, the sensors that haven't reach a decision will provide an implied decision. The sequential fusion system show improved ASN and accuracy comparing to the single local sensor.

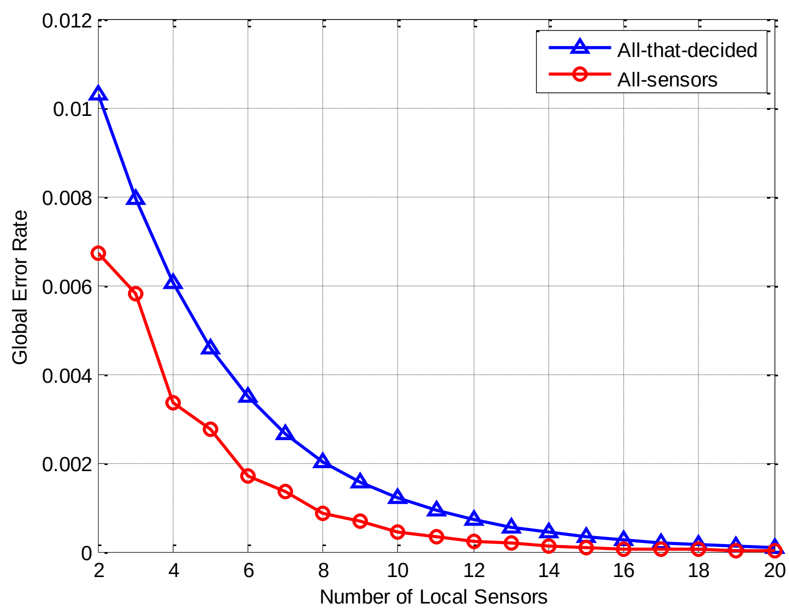


Fig. 15. Global error rate comparison, *All-that-decided* vs. *All-sensors* rule

Fig. 15 compares the global error rate for the *All-sensors* sequential fusion rule to the *All-that-decided* rule for the same sensors considered in the previous paragraphs. The *All-sensors* sequential fusion rule shows lower error rates compared to *All-that-decided* rule.

We examined three fusion rules for sequential sensors in a parallel configuration. We compared the performance in terms of the global error rate and Average Sampling Number (ASN). The first rule is *First-to-decide* rule, viz. the Fusion Center chooses one of the first set of local sensors' decisions it receives as the global decision; the sensors that have not reached a decision are ignored. The second rule is the *All-that-decided* rule, viz. once at least one sensor has stopped sampling, we integrate all the decisions of stopped sensors through the 1986 Chair-Varshney decision fusion rule. The third rule is the *All-sensors rule*, viz. once at least one sensor has stopped sampling, we combine the available decisions of the stopped sensor and the implied decisions of the remaining sensors. Among the three rules examined, the *All-sensors* rule has the lowest global error rate, with the same ANS as the other two rules.

5 Adaptive Decision Fusion in Stationary Environments

5.0 Introduction

In the binary decision fusion system shown in Fig. 5, the fusion center combines all the local decisions to obtain a global decision. For observations that are statistically independent conditioned on the hypothesis, the Chair-Varshney fusion rule minimizes the global Bayesian risk. However, this fusion rule requires knowledge of local sensor performance parameters and the prior probabilities of the hypothesis set. In most applications, these are unavailable. Moreover, local sensor performance may be time varying. Several studies attempted on-line estimation of the unknown local performance metrics and prior probabilities. We develop a fusion rule that applies a genetic algorithm to fuse the local-sensor binary decisions. The rule adapts to time varying local sensor error characteristics and provides near optimal performance at the expense of a larger number of observations and higher computational overhead.

Under the assumption that the local sensor error characteristics are known (fixed) and the local observations are independent conditioned on the hypotheses, Chair and Varshney (1986) proposed an optimal decision fusion rule for distributed detection that minimizes the Bayes' risk of a global decision. The Chair-Varshney rule, though widely used in distributed decision making, requires complete knowledge of the prior probabilities of the set of hypotheses and the error measures (false alarm and mis-detection) of the local detectors. In most applications, these quantities are unknown. Moreover, the local detector performances can be time varying. To address this challenge, we develop a fusion rule that applies a genetic algorithm to fuse the local detectors' binary decisions. The proposed genetic fusion rule adapts to time varying local sensor error characteristics and provides near optimal performance (close to that of the Chair and Varshney rule) at the expense of a larger number of observations and higher computational overhead.

The Chair-Varshney fusion rule requires knowledge of all local sensor performance (P_{Fi} and P_{Mi}) and prior probability of hypothesis P_1 and P_0. In practice, the information may not be available, and the performance variables may also be time varying. Studies such as [15], [16], [17] addressed this challenge by applying various techniques for on-line estimation of the local sensor parameters, and used these estimated parameters for global decision making. Such approaches are however slow to converge and their performance may deteriorate when the parameters are time varying.

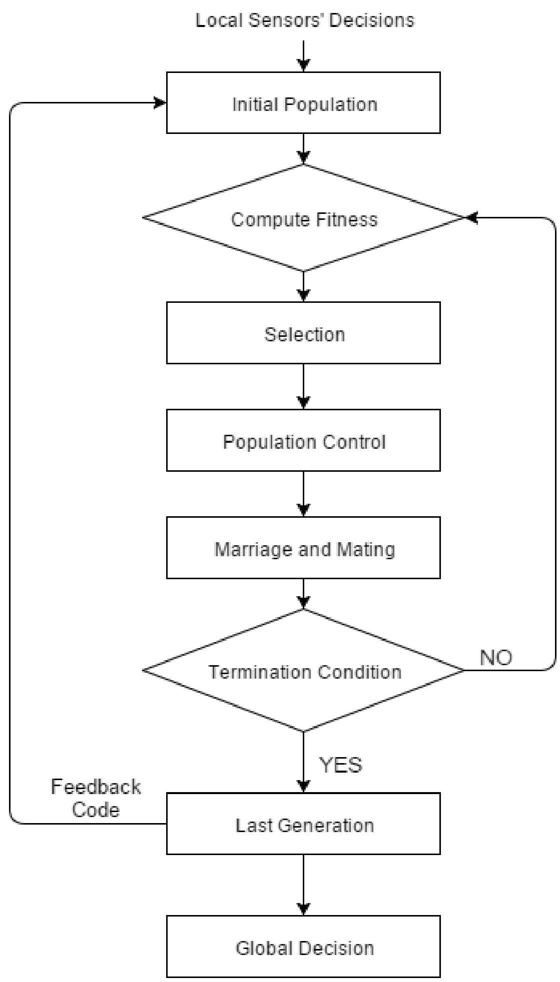


Fig. 16. Flow chart of the genetic fusion algorithm

In this chapter, we aim to design a binary fusion rule that does not that of explicitly on local performance parameters and prior probabilities and yet achieve comparable performance to that of the optimal rule defined in equation (2.7).

For the decision fusion system shown in Fig. 16, we present a genetic fusion algorithm that calculates the weights of local decisions for global decision making such that each local decision is weighed by the decision's current performance. We adopt the standard genetic algorithm procedure. The genetic algorithm tends to produce more "descendants" for decisions with better performance. The number of each local decision's descendants represents the weight of each local decision in the global decision making; more descendants means higher weight for that local decision. The global decision is made using a threshold test where the test statistic is dependent on the number of local decisions supporting each hypothesis.

We apply a genetic algorithm to the distributed detection system (Fig. 5). We assume that one of the two states (target is present or absent) persists for enough time so that the algorithm can converge. Fig. 16 shows the flow chart of the genetic-algorithm based system. Before we provide the details about each step individually, we review some relevant genetic algorithm terminology as used in the context of our distributed decision fusion architecture.

5.1 Adaptive Fusion with Genetic Algorithm

Algorithm Input/Output

At each stage of the algorithm, its input (initial population) is: (1) an $N \times M$ matrix, where M is the number of subsequent local decisions obtained from each of the N local sensors in the fusion system; and (2) an $M \times 1$ feedback code which was calculated during the previous stage.

The output of the algorithm at each stage is a global decision (-1; 1), plus an M-bit code which serves as the feedback code for the next stage. The only stage that deviates from this structure is the first stage, which does not use a feedback code.

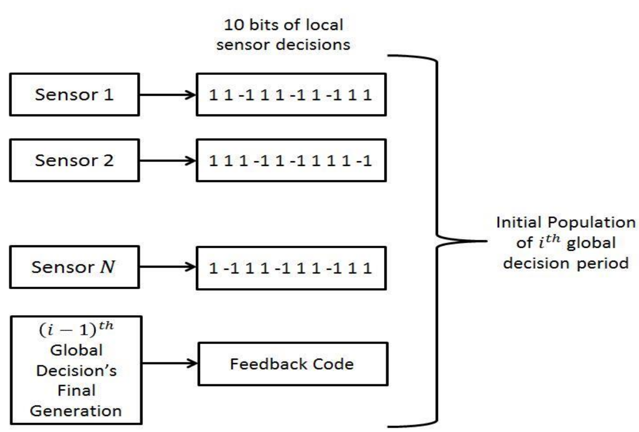


Fig. 17. Initial population

Fitness

The fitness of each genetic code in the population is defined as follows. Let s_1 denote the number of bits in a genetic code that are '1'. The number of majority bits is $Maj = max(s_1, M - s_1)$, and the number of minority bits is $Min = min(s_1, M - s_1)$. The fitness of the genetic code is defined as

$$Fitness = \frac{Maj - Min}{M}$$

Selection

The genetic codes are selected for the procedure of "marriage" and "mating" based on their fitness scores. A genetic code with higher fitness has a higher probability to be selected. For all genetic codes with relatively good consistency (fitness above 0:5), the probability of getting selected is assigned to be 1. For all genetic codes with fitness less than 0:5, the probability of getting selected varies linearly as a function of the fitness scores. The selection probability function we have used is

$$P(Select) = \begin{cases} 2Fitness \times 100\%, & Fitness \leq 0.5 \\ 100\%, & Fitness > 0.5 \end{cases}$$

If too few or too many genetic codes are selected, we add two population control procedures after selection. (Too few genetic codes being selected will cause early convergence; too many genetic codes being selected will cause high computational cost).

Early-Converge Prevention: if the number of genetic codes (N_0) falls bellow N (number of local sensors) in any generation, we mutate the N genetic codes that come from N local sensors, and select the $N - N_0$ of mutated codes with higher fitness.

Population Explosion Prevention: we set an upper limit L ($L = 200$ in our experiment) to the number of genetic codes allowed after selection. If the number of genetic codes exceeds L, we select the L codes with the highest fitness and discard the rest.

Marriage and Mating

After selection, the selected genetic codes are used in marriage and mating process to generate the next generation of population.

Marriage: The probability of marriage between any two genetic codes depends on their similarity; codes with high similarity would have higher tendency to marry. In this study, we define similarity between two genetic codes g_i and g_j using the Hamming distance (HD) as follows

$$Similarity_{(i,j)} = 1 - \frac{HD_{(i,j)}}{M}$$

The probability of marriage between two genetic codes is defined as

$$P\left(Marriage_{(i,j)}\right) = Similarity_{(i,j)}{}^2$$

Mating: Once a pair of genetic codes was selected for marriage, the two codes will mate (swap their bits) depending on a randomly chosen crossover point. The crossover point is determined by selecting a random bit along the length of the genetic code and swapping the bits after that point (an example is shown in Fig. 18). For a 10 bit code, all 9 intersection positions are equally probable of being chosen as the crossover point.

The new generation obtained after the marriage and mating process is tested against the algorithm termination conditions. If the termination conditions are satisfied, this new generation is considered as the last generation for the current decision making stage. If the termination conditions are not satisfied, the new generation is used for the next cycle of genetic algorithm.

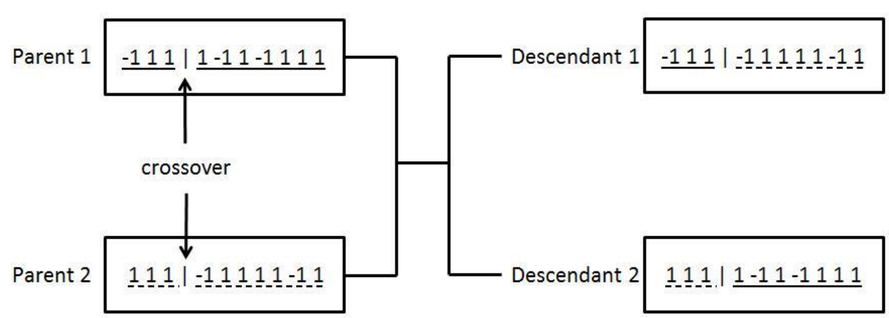


Fig. 18. Mating process of two genetic codes to produce descendant generation

Termination Condition

We employ two termination conditions to control the execution of the genetic algorithm.

Termination Condition 1: All genetic codes in a particular generation are collected and we compute the percentage of decisions supporting $H1$ and $H0$ in this aggregated pool. We denote by $Percent_{Hi}$ as the percentage of decisions in a generation that supports $Hi, i = 0, 1$. For a certain generation, if the percentage of decisions supporting one hypotheses exceeds $T\%$, this is considered the last generation of the current decision stage. The value of T controls the trade-off between accuracy and time of convergence. If higher accuracy is desired, T should be set relatively large; on the other hand, if quick decision making is desired, T should be set relatively smaller. In our simulations, we set the value of T to 99%.

Termination Condition 2: The maximum number of generations is set to be G. The genetic algorithm stops execution when the current set of genetic codes constitute the G^{th} generation. In our experiments we used $G = 7$.

Mutation

We use the following procedure for mutating a bit in a genetic code. Recall our definition for Maj and Min for each genetic code. Bits that belong to the 'minority' are flipped with probability _. Bits that belong to the 'majority' are flipped with probability α. A genetic code will preserve its fitness on average if $\beta = \frac{Min}{Maj}\alpha$, and would have higher fitness on average if $\beta < \frac{Min}{Maj}\alpha$. In our experiment, we use $\alpha = 0.1$ and $\beta = \frac{Min}{Maj}\alpha$.

Feedback

At the k^{th} decision stage, after obtaining the global decision using the last generation of this stage, the system retains the M-bit genetic code with the largest fitness in the last generation of the k^{th} stage. It provides this genetic code as an input into the initial generation of the $(k+1)^{th}$ decision stage.

Global Decision Making

The global decision is made based on the following threshold test

$$\frac{Percent_{H_1}}{Percent_{H_0}} \underset{H_0}{\overset{H_1}{\gtrless}} \frac{P_0(C_{10}-C_{00})}{P_1(C_{01}-C_{11})}$$

Here $Percent_{H_1}$ and $Percent_{H_0}$ are the percentage of the hypothesis in the last generation. The cost were assumed to be $C_{00} = C_{11} = 0$ and $C_{01} = C_{10} = 1$.

For computation of the threshold in (9), the knowledge of the prior probability P_1 is required. In real applications, prior probabilities are seldom known. In this situation, the genetic algorithm can be used to aid in estimating the prior probability of the hypothesis. P_1 can be estimated by the following steps.

(1) Initial the prior probability of hypothesis H_1 as $P_1 = \hat{P}_1{}^0$ (a guess).

(2) Use the recursive expression provided below to estimate P_1 in the k^{th} global decision making,

$$\hat{P}_1{}^k = \frac{k-1}{k}\hat{P}_1{}^{k-1} + \frac{1}{k}\frac{Percent_{H_1}^k}{100}$$

Where $Percent_{H_1}^k$ represents the percentage of the decisions supporting H_1 in last generation of the k^{th} decision making stage.

5.2 Performance of genetic fusion algorithm in simulation

The proposed framework based on genetic algorithm was tested against simulated detection scenarios for performance analysis. We first fix the number of generations to show how the genetic fusion algorithm increases the percentage of correct decisions in each generation. We then compare the performance of the algorithm with and without feedback to show how the feedback improves the performance. We next test the fusion scheme through a simulated example scenario of target detection where error rates of local sensors are time varying.

5.2.1 Decision-making with and without feedback

We fixed the number of generations to 7. We show the performance of the fusion system with and without feedback. We simulated the fusion system with three different local sensors. The local sensor performances were fixed with false alarm rates as 0.3524, 0.3077, 0.3417; the mis-detection rates were 0.3123; 0.3550, 0.3136, respectively.

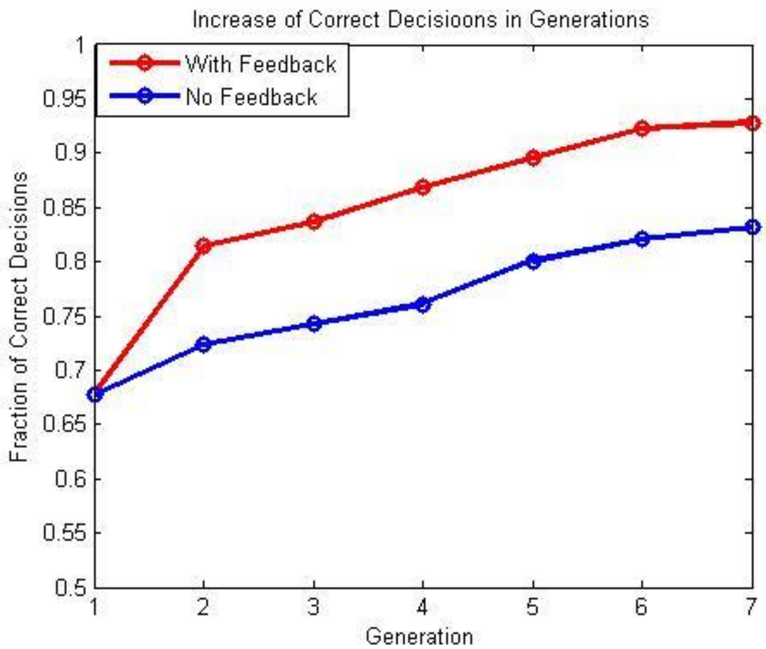


Fig. 19. Compare the performance of genetic fusion algorithm with and without feedback

TABLE II. COMAPRE THE PERFORMANCE OF GENETIC FUSION ALGORITHM WITH AND WITHOUT FEEDBACK

	With Feedback	Without Feedback
Global Probability of Error	7.2×10^{-4}	9.86×10^{-3}
Percentage of Correct Decision in the Initial Generation	66.44%	66.44%
Percentage of Correct Decision in the Last Generation	91.52%	78.75%

We present the performance improvement obtained with the application of the proposed algorithm in Fig. 19 and Table 2. The percentage of correct decisions in the initial population (local sensors' decisions) was 66.44%. Applying the genetic algorithm for 7 generations without feedback, the percentage of correct decisions in the last generation was 78.75%. Incorporating feedback increased the percentage of correct decisions in the last generation to 91.52%. Feedback reduced the global error rate from 9.86×10^{-3} to 7.2×10^{-4}.

6.2.2 Decision-making with time varying local sensor

For the next simulation, the number of generations was decided by the first termination condition. We simulated a distributed detection system with three ($N = 3$) local sensors. We tested the fusion algorithm with feedback under two cases: (1) local sensor performance is fixed; (2) local sensor performance is time-varying. Table 3 summarizes the performance of the proposed fusion algorithm under the two local sensor performance characteristics.

TABLE III. PERFORMANCE OF THE GENETIC FUSION ALGORITHM UNDER FIXED SENSOR CASE AND TIME-VARYING SENSOR CASE

	Fixed Sensor	Time-Varying Sensor
Global Probability of Error	7.2×10^{-4}	7.0×10^{-4}
Percentage of Correct Decision in the Initial Generation	66.44%	66.18%
Percentage of Correct Decision in the Last Generation	91.52%	91.17%
Average number of Generation	6.23	6.22

Fixed Local Sensor Performance: The local sensor performance was fixed with local false alarm rates as 0.3524, 0.3077, 0.3417; and mis-detection rates as 0.3123; 0.3550, 0.3136, respectively. Using feedback, the simulated global error rate was 7.2×10^{-4}. The average number of generations needed to obtain global decision was 6.23.

Time-Varying Local Sensor Performance: In this case, the local sensor performance was time-varying as shown in Fig. 20. The average false alarm rates of the three sensors were: 0.3524, 0.3077, 0.3417; the average miss-detection rates of the three sensors were: 0.3123; 0.3550, 0.3136, respectively.

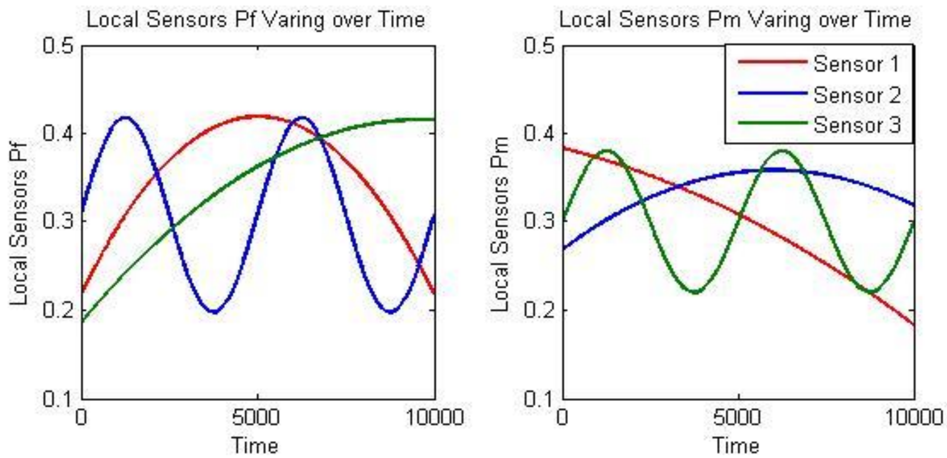


Fig. 20. Local sensor performance is time-varying

The overall percentage of correct decisions in local sensors' decisions was improved from 66.18% in the initial population to 91.17% in the last generation. Using feedback, the simulated global error rate was 7.0×10^{-4}. The average number of generations needed to obtain global decision was 6.22. The genetic algorithm in these simulations was impervious to changes in local error metrics (however, it would be interesting to investigate the performance of the fusion system as a function of the sampling rate vs rate of change of local sensor performance.).

5.2.3 Comparison of the genetic fusion algorithm with Chair-Varshney fusion rule

In this section, we compare the proposed fusion algorithm with the Chair-Varshney fusion rule [1] which assumes precise knowledge of local sensors' performance. The genetic fusion algorithm uses M bits of each local sensor's decisions for a total of $N \times M$ bits to obtain a global decision. In order to have a fair comparison, we use the Chair-Varshney system with $N \times M$ sensors each sending out a 1 bit decision to the fusion center. We vary the number of local sensors (N) from 3 to 12. To simplify the calculation, we assume that the local sensors in both systems are identical with error rates fixed as 0.36 misdetection rate and 0.40 false-alarm rate.

TABLE IV. COMPARE THE PERFORMANCE OF CHAIR-VARSHNEY FUSION RULE AND GENETIC FUSION ALGORITHM

Number of Local Sensors	Global Probability of Error of Chair-Varshney Fusion Rule	Simulation Error Rate of Genetic Fusion Algorithm without Feedback
3	0.1245	0.1259

4	0.1082	0.1091
5	0.0721	0.0727
6	0.0616	0.0623
7	0.0432	0.0449
8	0.0373	0.0384
9	0.0265	0.0273
10	0.0234	0.0237
11	0.0165	0.0170
12	0.0150	0.0151

Table 4 shows that comparison of genetic fusion algorithm (without feedback) and the Chair-Varshney binary fusion rule. We calculate the global error rate for the fusion system applying Chair-Varshney binary fusion rule. For the genetic fusion algorithm without feedback, the error rate is the simulation error rate. From Table 4 we can see that, the genetic algorithm has nearly similar global error rates as compared to those obtained from the Chair-Varshney binary fusion rule. The simulation error rate nearly reaches the optimal bound achieved by the Chair-Varshney rule as the number of sensors is increased. There are two additional burdens in our system: (1) we needed a relatively large number of decisions to develop a global decision; and (2) we needed several cycles (generations) of the algorithm, which mean more computations and more delay.

5.2.4 A simulated 4-Sensor example

We consider a simulated 4-sensor detection problem. The simulation assumed four RF sensors. Fig. 21 shows the positions of the four sensors and the moving track of the target. As the target is moving around, the distances between the target and the sensors are changing. Assuming line of sight propagation, it is well known that the received signal power is inversely proportional to the square of the distance between the sensor and the target [10]. Hence, as the distance changes, the received power varies which in turn influences the detection rates of the sensors. However, the false alarm rate is only affected by noise. In the simulations, we assume that the sensor false alarm rate is fixed at 0.1. Fig. 22 shows the variation in the miss-detection rates of the four sensors. In the simulations, we assume both the local sensor performance parameters and prior probability of hypotheses are unknown.

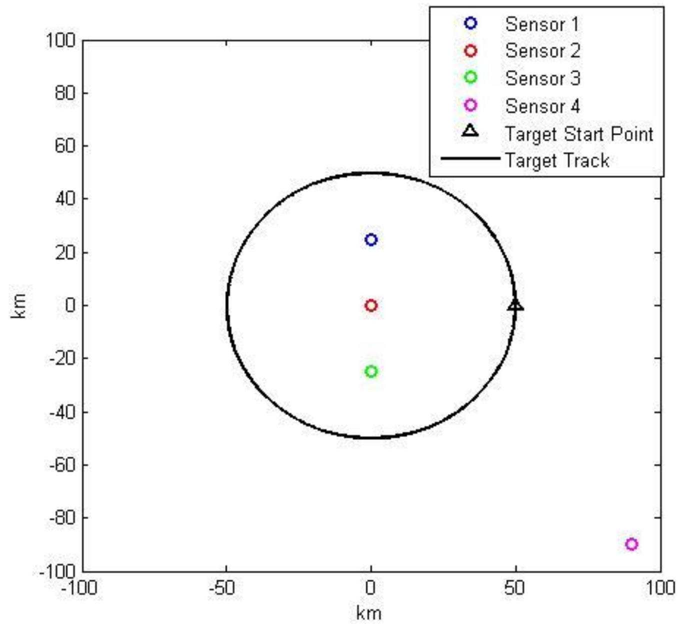


Fig. 21. An 4-sensor detection system and the target track

We test the simulations when the true state changes over time but is still consistent long enough for the genetic fusion algorithm to converge. For each fusion method, we ran 10 simulations and in each simulation 104 global decisions were made.

The average simulated global error rate was 3.16×10^{-4}. In the final generations upon convergence, the percentage of the descendants of each local sensor's decision presents the weights by which this sensor influences the global decision. The overall weights of all four local sensors were: 0.3674 for sensor 1, 0.3098 for sensor 2, 0.2811 for sensor 3, 0.0417 for sensor 4. Sensor 4 has the smallest overall weight, as it had the worst overall detection performance.

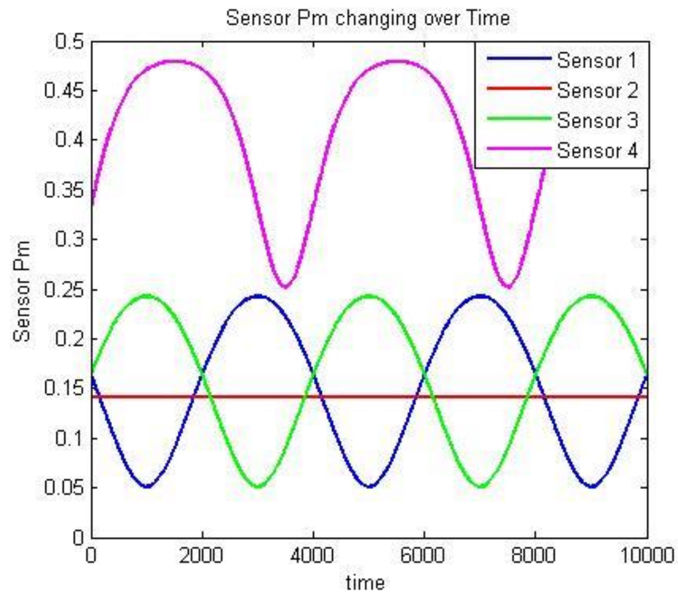


Fig. 22. Local sensors' missed detection rate change over time as the target is moving

We present a distributed decision fusion scheme based on the standard genetic algorithm. The framework aids in combining local decisions to obtain the global decision without using the local sensors' performance parameters. In our simulations, the algorithm was able to adapt to the variations in local sensor performance. The genetic fusion algorithm showed nearly equivalent detection accuracy both under fixed and time-varying local error rates and achieved nearly equivalent performance in detection accuracy as compared to that obtained from the Chair-Varshney fusion rule which assumes precise knowledge of local performance rates. The cost of this performance was that the algorithm required large sets of local decisions to form the initial population. Furthermore, iterating over the steps of the genetic algorithm causes processing delays and requires computational overheads.

6 Adaptive Decision Fusion in Non-stationary Environments

6.0 Introduction

In chapter 6, we proposed an adaptive decision fusion with genetic algorithm under the condition that the environment is stationary (the true hypothesis does not change over time). In this chapter, we develop a hypothesis change detection mechanism for the adaptive decision fusion we proposed in chapter 5, so that the adaptive decision fusion can adapt the non-stationary environments (the true hypothesis changes over time).

The adaptive decision fusion, which we proposed in chapter 5, needs several bits of each local sensor decisions to obtain one global decision. In a fusion system with N local sensors, the fusion center collects M-bits of decisions from each local sensor, and uses these $N \times M$ local decisions to generate a global decision. In chapter 5, we assume the true hypothesis does not change during the collected M-bits (stationary environment). However, in the real applications, this is not the case. If the true hypothesis changes during the collected M-bits (non-stationary environment), the fitness (which we introduce in section 5.1-*Fitness*) of the sensor's genetic code cannot provide the correct information about the performance of the local sensors. Accordingly, the genetic fusion algorithm cannot function properly in this case.

6.1 Problem of true hypothesis changing during the M-bits

Given an error free local sensor, with the application of genetic fusion algorithm ($M = 10$), if the true hypothesis is "1" and does not change during the 10 bits, this sensor's 10-bits decisions will be "1,1,1,1,1,1,1,1,1,1", and the fitness of this sensor will be 1 (section 5.1-*Fitness*). However, if the true hypothesis changes from "1" to "-1", say, at the 6th bit during the 10-bit segment, this sensor's 10-bits decisions would be "1,1,1,1,1,-1,-1,-1,-1,-1", and its fitness would be 0. In such situation, the decision of this error-free sensor has very large probability to be eliminated in the "selection" procedure (section 6.1-*Selection*). Therefore, when applying the adaptive decision fusion in the non-stationary environment (true hypothesis changes during M-bits), the sensor's fitness may not provide the correct information about the local sensor performance, and the genetic fusion algorithm may not function properly.

As an example, we did a simulation to show how the non-stationary environments (true hypothesis changes during the M-bits) decreases the performance of genetic fusion algorithm. We simulate a fusion system with 4 identical local sensors. Local sensor $p_f = p_m = 0.2$. $M = 10$. There are 10 cases in this simulation: the true hypothesis changes at i^{th} $(i = 1,2,\dots,10)$ bit of the collected 10-bits. We did 10^5 tests for each case, and calculated the error rate based these tests. Table 5 shows the results of this simulation.

TABLE V. SIMULATION PERFORMANCE OF THE GENETIC FUSION ALGORITHM WHEN HYPOTHESIS CHANGES DURING THE M-BITS $(M = 10)$

True hypothesis changes at (of 10-bits)	Average Percentage of the true hypothesis in the final generation	Global Error Rate
1st bit	94.27%	2.0×10^{-4}
2nd bit	86.27%	7.3×10^{-4}
3rd bit	78.61%	2.7×10^{-3}
4th bit	66.52%	6.5×10^{-2}
5th bit	57.34%	2.8×10^{-1}
6th bit	50.12%	4.9×10^{-1}
7th bit	57.28%	2.8×10^{-1}
8th bit	66.63%	6.6×10^{-2}
9th bit	78.58%	2.7×10^{-3}
10th bit	86.13%	7.4×10^{-4}

It can be seen in Table 5, if the hypothesis-change happens at the beginning or the end of the M-bits, the genetic fusion algorithm can still function and the change detection is not necessary. However, if the hypothesis-change happens at the middle part of M-bits (5^{th}, 6^{th}, 7^{th} bits), the genetic fusion algorithm cannot function well.

The Fig. 23 shows an example of the simulated global error rate of the genetic fusion algorithm when hypothesis changes at 100^{th}, 200^{th} and 300^{th} bit. When the hypothesis change does not happen, the global error rate stays stable at a small value. When the hypothesis change happens, the global error rate increases, and can rise to nearly 50%.

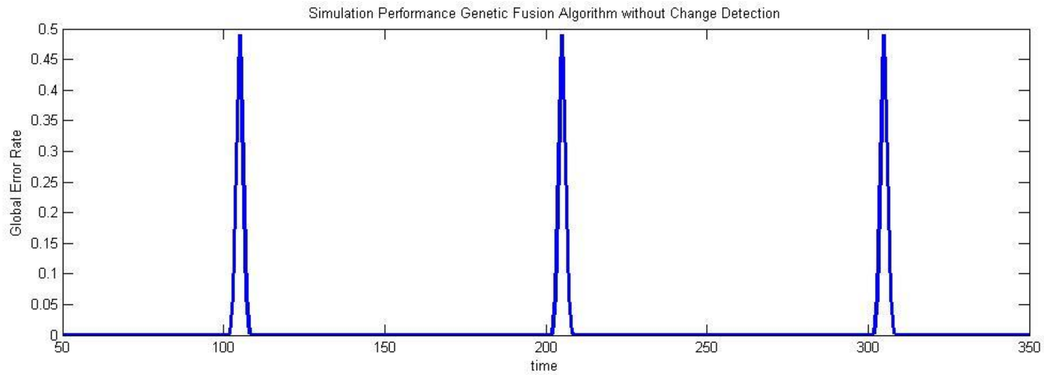


Fig 23: An example of the simulation performance of genetic fusion algorithm when there are 3 hypothesis-change-points in a time period.

6.2 Hypothesis Change Detection in Genetic Algorithm

The non-stationary environment (true hypothesis changes during the M-bits) largely decreases the detection accuracy of the adaptive decision fusion algorithm. To improve the performance of adaptive decision fusion algorithm in non-stationary environments, we need to add a change detection mechanism in the adaptive decision fusion with genetic algorithm. We want the system to accept the decision of hypothesis change detection instead of the decision of genetic fusion algorithm, when there is a larger probability that the hypothesis changes during the M-bits; and accept the decision of genetic fusion algorithm when there is a low probability that the hypothesis changes.

The new mechanism first runs the genetic fusion algorithm (which we proposed in chapter 6). Based on the performance of genetic fusion algorithm, the system decides whether the decision of genetic fusion algorithm is reliable. If the system decides the decision of genetic fusion algorithm is reliable, it will accept the decision of genetic fusion algorithm. Otherwise, the system will decide to apply the hypothesis change detection and accept the decision of hypothesis change detection. (We will discuss how to judge the performance of genetic fusion algorithm and decides whether the decision of genetic fusion algorithm is reliable is section 6.2.1.).

The hypothesis change detection has two procedures: (1) local sensor level detection, and (2) global level detection. In local sensor level detection, N (the number of local sensors) local sensor level decisions are made. The i^{th} sensor level decision is made independently based on the information from the i^{th} sensor. The global level decision is obtained by combining all the N sensor level decisions.

6.2.1 Decide whether to apply the hypothesis change detection in adaptive decision fusion

The genetic fusion algorithm has good performance in stationary environment (true hypothesis change does not happen); but it cannot function well when the true hypothesis changes. Therefore, we want to apply the hypothesis change detection only if there is a larger probability that the hypothesis changes. Otherwise, we want the system to accept the decision of genetic fusion algorithm and there is no need to apply the hypothesis change detection.

The system decides whether to apply the hypothesis change, based on the performance of the genetic fusion algorithm. If the performance of genetic fusion algorithm has a significant decrease in a short time (during several bits of decisions), we suspect there is a high probability that hypothesis change happens.

Assuming the true hypothesis changes in the middle ($\left(\frac{M}{2}+1\right)^{th}$ bit) of the M-bits, the average percentage of the majority decisions in the final generation would be about 50% (from the statistic of the experiment results in Table 4).

On the other hand, assuming the hypothesis does not change, the genetic fusion algorithm should function well and the percentage of the majority decisions in the final generation should not have a significant change. Therefore, we calculate (Maj_{Final}^{E}), the expected percentage of the majority decisions in the final generation using historical information (starting from the last detected-change-point), assuming the hypothesis does not change. Maj_{Final}^{E} is calculated as the mean value of the historical percentage of the majority decisions in the final generation.

The detected-change-point: the point that the global decision of the system changes (from H_1 to H_0 or form H_0 to H_1)).

The genetic fusion algorithm can provide (Maj_{Final}^{T}) the true percentage of the majority decisions in the final generation. We consider there is a larger probability that the hypothesis change happens, if Maj_{Final}^{T} is closer to 50%; and there is a lower probability that the hypothesis changes, if Maj_{Final}^{T} is closer to Maj_{Final}^{E}.

We calculated ($Dist_{Final}^{T-E}$) the distance between Maj_{Final}^{T} and Maj_{Final}^{E}, and ($Dist_{Final}^{T-0.5}$) the distance between Maj_{Final}^{T} and 50%.

$$Dist_{Final}^{T-E} = \left|Maj_{Final}^{T} - Maj_{Final}^{E}\right|$$

$$Dist_{Final}^{T-0.5} = \left|Maj_{Final}^{T} - 50\%\right|$$

If Maj_{Final}^{T} is closer to 50%, the system decides to apply the hypothesis change detection and accept the decision of hypothesis change detection; otherwise the system accepts the decision of genetic fusion algorithm and does not apply hypothesis change detection.

$$\begin{cases} \begin{matrix}\text{Apply Change Detection} \\ \text{Accept Change Detection's Decision'}\end{matrix} & Dist_{Final}^{T-0.5} < Dist_{Final}^{T-E} \\ \\ \begin{matrix}\text{Does not apply Change Detection} \\ \text{Accept Genetic Algorithm's Decision'}\end{matrix} & Dist_{Final}^{T-0.5} \geq Dist_{Final}^{T-E} \end{cases}$$

6.2.2 Hypothesis change detection in sensor level

From table 4 it can be seen that, the genetic fusion algorithm can have acceptable performance when the hypothesis change happens at the beginning or the end of the M-bits; and it has significant bad performance when hypothesis changes in the middle part (around $\left(\frac{M}{2}+1\right)^{th}$ bit) of the M-bits. Therefore, in the hypothesis change detection, we want to detect the change when the hypothesis change happens around the middle part of the M bits. Assuming a sensor with $p_f = p_m$, if M is an even number and the hypothesis changes at the $\left(\frac{M}{2}+1\right)^{th}$ bit of the M bits, the expectation of the fitness of this sensor will be 0 (Fit^0). Therefore, we consider there is a high probability that hypothesis-change happens during the M-bits, if the fitness of sensor is close to 0 (Fit^0).

We consider there is a low probability that hypothesis-change happens during the M-bits, if the fitness of sensor does not have dramatic decrease comparing with its previous value. Assuming the hypothesis does not change, each sensor calculates its expected fitness (Fit_i^E) using its historical fitness value. Fit_i^E is calculated as the mean value of i^{th}sensor's historical fitness. To calculate Fit_i^E, the system needs to record the fitness of sensors starting from the last detected-change-point.

We calculate $Dist_i^{T-E}$, the distance between the true value of sensor fitness (Fit_i^T) and the expectation of sensor fitness (Fit_i^E); and $Dist_i^{T-0}$, the distance between the true value of sensor fitness (Fit_i^T) and 0 (Fit^0) .

$$Dist_i^{T-E} = \left|Fit_i^T - Fit_i^E\right|$$

$$Dist_i^{T-0} = \left|Fit_i^T - Fit^0\right|$$

If the true value of sensor fitness is closer to 0, then this sensor decides the hypothesis changes; otherwise this sensor decides the hypothesis does not change. If the sensor the hypothesis changes, it outputs "1"; if it decides the hypothesis does not change, it outputs "-1".

The i^{th} sensor-level decision u_i^{C-D} of hypothesis change detection are made as

$$u_i^{C-D} = \begin{cases} 1 \text{ (Change)}, & Dist_i^{T-0} > Dist_i^{T-E} \\ -1 \text{ (No change)}, & Dist_i^{T-0} \le Dist_i^{T-E} \end{cases}$$

6.2.3 Hypothesis change detection in global level

Once the sensor-level decisions have been made, fusion center combines all the sensor-level decisions to obtain the global-level decision on whether the hypothesis changes. In the fusion of sensor-level decisions, the sensor-level decisions are weighted by how confidence these sensor-level decisions are on their own decisions.

The i^{th} sensor-level decision's weight w_i^{C-D} are calculated as

$$w_i^{C-D} = \frac{|Dist_i^{T-E} - Dist_i^{T-0}|}{Dist_i^{T-0} + Dist_i^{T-E}}$$

The global-level decision U_g^{C-D} of the hypothesis change detection is made from the weighted sum of sensor-level decisions.

$$U_g^{C-D} = \begin{cases} 1 \text{ (Change)}, & \sum_{i=1}^{N} w_i^{C-D} u_i^{C-D} > 0 \\ -1 \text{ (No Change)}, & \sum_{i=1}^{N} w_i^{C-D} u_i^{C-D} \leq 0 \end{cases}$$

If the system decides the hypothesis changes, the system's output flips the latest global decision of the adaptive decision fusion algorithm (H_0 or H_1); if the system decides the hypothesis does not change, the system's output keeps the latest global decision of the adaptive decision fusion algorithm. For example, when the latest global decision decides H_0 , if the hypothesis change detection decides "hypothesis changes" at current time, the system's output will be H_1; if the hypothesis change detection decides "hypothesis does not change" at current time, the system's output will be H_0

Algorithm of change detection in adaptive fusion using genetic algorithm

Part 1: The adaptive fusion system first applies genetic fusion algorithm (proposed in chapter 6). It records the percentage of the majority decisions in the final generation of the genetic fusion algorithm. The system also records the fitness of all sensors.

Part 2: Decide whether to apply hypothesis change detection

(1) Calculate (Maj_{Final}^{E}), the expected percentage of the majority decisions in the final generation of the genetic fusion algorithm using the records from the last detected-change-point, assuming the hypothesis does not change. Maj_{Final}^{E} is the calculated as the mean value of the records.
(2) Get (Maj_{Final}^{T}) the true value of the percentage of the majority decisions in the final generation of the genetic fusion algorithm.
(3) Calculated ($Dist_{Final}^{T-E} = \left|Maj_{Final}^{T} - Maj_{Final}^{E}\right|$) the distance between Maj_{Final}^{T} (true value) and Maj_{Final}^{E} (the expected value if the hypothesis does not change).
(4) Calculated ($Dist_{Final}^{T-E} = \left|Maj_{Final}^{T} - 50\%\right|$) the distance between Maj_{Final}^{T} (true value) and 50% (the expected value if the hypothesis changes).

(5) If $Dist_{Final}^{T-0.5} < Dist_{Final}^{T-E}$, apply the hypothesis change detection and accept the decision of hypothesis change detection. If $Dist_{Final}^{T-0.5} \geq Dist_{Final}^{T-E}$, accept the decision of genetic fusion algorithm and does not apply the hypothesis change detection.

Part 3: Steps of hypothesis change detection

(1) Calculate Fit_i^E ($i = 1, \dots, N$), expected fitness of the i^{th} sensor, assuming the hypothesis does not change. Fit_i^E is calculated as the mean value of each sensor's historical fitness (starting from the last detected-change-point).

(2) Get Fit_i^T ($i = 1, \dots, N$), the true value sensor fitness from the genetic fusion algorithm.

(3) Calculate $Dist_i^{T-E} = \left|Fit_i^T - Fit_i^E\right|$, the distance between Fit_i^T (true value) and Fit_i^E (the expected value if hypothesis does not change).

(4) Calculate $Dist_i^{T-0} = \left|Fit_i^T - Fit^0\right|$, ($Fit^0 = 0$), the distance between Fit_i^T (true value) and the Fit^0 (the expected value if hypothesis changes).

(5) Get sensor-level decision on whether the hypothesis changes,

$$u_i^{C-D} = \begin{cases} 1 \text{ (Change)}, & Dist_i^{T-0} > Dist_i^{T-E} \\ -1 \text{ (No change)}, & Dist_i^{T-0} \leq Dist_i^{T-E} \end{cases}$$

(6) Get global-level decision on whether the hypothesis changes:

The i^{th} sensor-level decision is weighted by,

$$w_i^{C-D} = \frac{|Dist_i^{T-E} - Dist_i^{T-0}|}{Dist_i^{T-0} + Dist_i^{T-E}}$$

Global level decision-making of hypothesis change detection,

$$U_g^{C-D} = \begin{cases} 1 \text{ (Change)}, & \sum_{i=1}^{N} w_i^{C-D} u_i^{C-D} > 0 \\ -1 \text{ (No Change)}, & \sum_{i=1}^{N} w_i^{C-D} u_i^{C-D} \leq 0 \end{cases}$$

6.3 Simulation Performance

We run several simulations under different circumstances to test the performance of adaptive decision fusion with hypothesis change detection mechanism. We first test the adaptive fusion with fixed sensors. Next, we test it with time-varying sensors. At last, we investigate the influence of the hypothesis change rate on the performance of hypothesis change detection.

7.3.1 Fixed local sensor

We simulate a fusion system with 4 identical local sensors. Local sensor $p_f = p_m = 0.2$. $M = 10$. The true hypothesis changes every 100 bits, (hypothesis change rate = 1 change per 100 bits). We run 10^5 tests for each case (total 10 cases: the true hypothesis changes at the i^{th} $(i = 1, 2, \dots, 10)$ bit of the 10-bits).

TABLE VI. SIMULATION PERFORMANCE OF THE ADAPTIVE DECISION FUSION WITH HYPOTHESIS CHANGE DETECTION, WHEN HYPOTHESIS CHANGES DURING THE M-BITS $(M = 10)$. THE LOCAL SENSOR PERFORMANCE IS FIXED.

True hypothesis changes at (of 10-bits)	Average Percentage of the true hypothesis in the final generation	Percentage of tests accepting the decision of Genetic Algorithm. Percentage of tests accepting the decision of Change Detection.	Global Error Rate
1st bit	94.27%	100% tests accepts Genetic Algorithm 0% tests accepts Change Detection	2.0×10^{-4}
2nd bit	86.27%	99.9% tests accepts Genetic Algorithm 0.1% tests accepts Change Detection	9.3×10^{-4}
3rd bit	78.61%	91.2% tests accepts Genetic Algorithm 8.8% tests accepts Change Detection	3.8×10^{-3}
4th bit	66.52%	30.6% tests accepts Genetic Algorithm 69.4% tests accepts Change Detection	9.5×10^{-3}
5th bit	57.34%	8.5% tests accepts Genetic Algorithm 91.5% tests accepts Change Detection	6.6×10^{-3}
6th bit	50.12%	0.6% tests accepts Genetic Algorithm 99.4% tests accepts Change Detection	4.7×10^{-3}
7th bit	57.28%	8.1% tests accepts Genetic Algorithm 91.9% tests accepts Change Detection	6.5×10^{-3}
8th bit	66.63%	29.7% tests accepts Genetic Algorithm 70.3% tests accepts Change Detection	9.1×10^{-3}
9th bit	78.58%	90.9% tests accepts Genetic Algorithm 9.1% tests accepts Change Detection	3.8×10^{-3}
10th bit	86.13%	99.9% tests accepts Genetic Algorithm 0.1% tests accepts Change Detection	9.3×10^{-4}

Comparing the table.4 and table.5, it can be seen that applying the hypothesis change detection improves the overall performance of the adaptive fusion algorithm when hypothesis changes.

The Fig. 24 shows an example of the simulated global error rate of the adaptive fusion with hypothesis change detection when hypothesis changes every 100 bits. The Fig. 25 shows the comparison of the adaptive fusion with hypothesis change detection and adaptive fusion without hypothesis change detection. Although the global error rate still increases when the hypothesis changes, adding the hypothesis change detection mechanism to the adaptive fusion algorithm does improve the performance of the fusion system a lot.

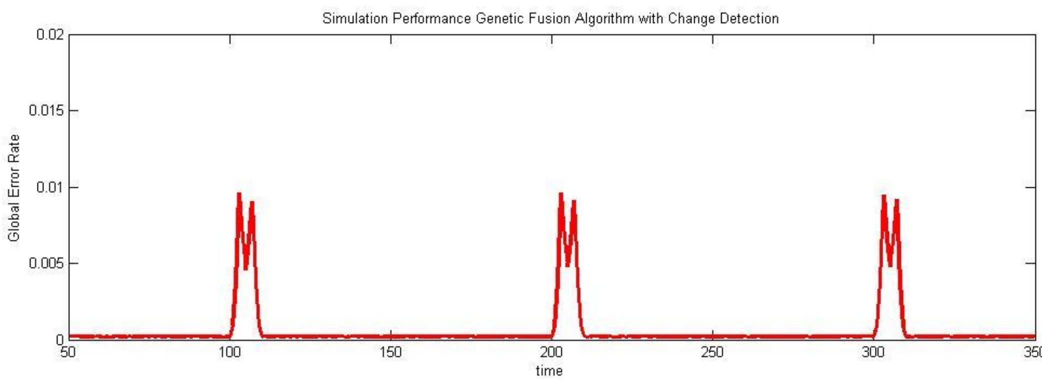


Fig 24: An example of the simulation performance of the adaptive decision fusion with hypothesis change detection, when there are 3 hypothesis-change-points in a time period.

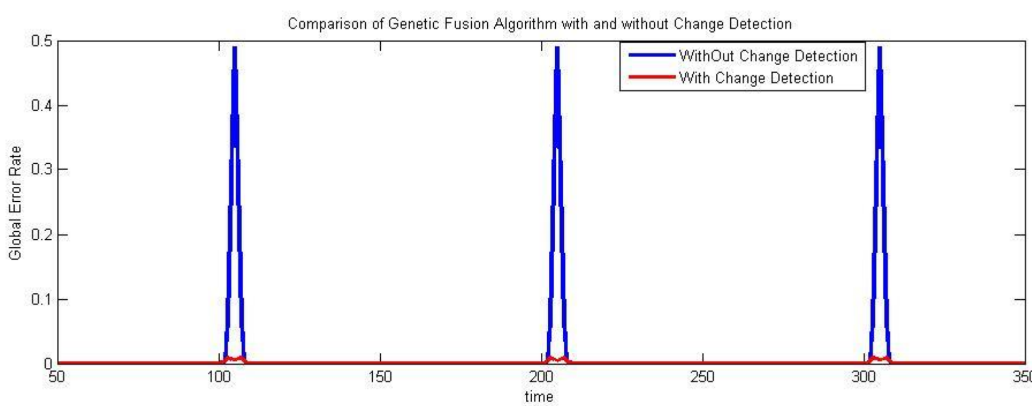


Fig 25: Comparing the simulation performance of the adaptive decision fusion with and without hypothesis change detection.

6.3.2 Time-varying local sensors

We simulate a fusion system with 4 local sensors with different performance. Local sensor p_f and p_m are time-varying (varying smoothly) as Fig. 26 shows. For all the local sensors, their average p_f equals 0.2, and average p_m equals 0.2. $M = 10$. The true hypothesis changes every 100 bits, (hypothesis change rate = 1 change per 100 bits).

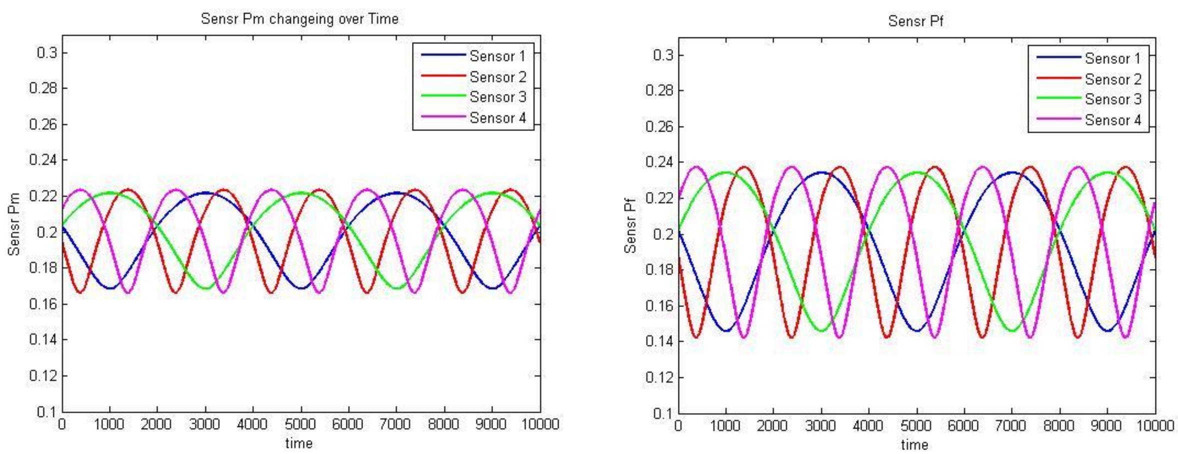


Fig 26: Local sensor performance is time-varying

TABLE VII. SIMULATION PERFORMANCE OF THE ADAPTIVE DECISION FUSION WITH HYPOTHESIS CHANGE DETECTION, WHEN HYPOTHESIS CHANGES DURING THE M-BITS ($M = 10$). THE LOCAL SENSOR PERFORMANCE IS TIME-VARYING.

True hypothesis changes at (of 10-bits)	Average Percentage of the true hypothesis in the final generation	Percentage of tests accepting the decision of Genetic Algorithm. Percentage of tests accepting the decision of Change Detection.	Global Error Rate
1st bit	95.11%	100% tests accept Genetic Algorithm 0% tests accept Change Detection	1.9×10^{-4}
2nd bit	87.02%	99.9% tests accept Genetic Algorithm 0.1% tests accept Change Detection	9.1×10^{-4}
3rd bit	78.34%	91.2% tests accept Genetic Algorithm 8.7% tests accept Change Detection	3.7×10^{-3}
4th bit	66.26%	29.8% tests accept Genetic Algorithm 70.2% tests accept Change Detection	9.7×10^{-3}
5th bit	58.17%	8.3% tests accept Genetic Algorithm 91.7% tests accept Change Detection	6.8×10^{-3}

6^{th} bit	50.25%	0.6% tests accept Genetic Algorithm 99.4% tests accept Change Detection	5.0×10^{-3}
7^{th} bit	57.28%	8.3% tests accept Genetic Algorithm 91.7% tests accept Change Detection	6.6×10^{-3}
8^{th} bit	66.20%	29.8% tests accept Genetic Algorithm 70.2% tests accept Change Detection	9.5×10^{-3}
9^{th} bit	77.96%	91.2% tests accept Genetic Algorithm 8.8% tests accept Change Detection	3.6×10^{-3}
10^{th} bit	86.24%	99.9% tests accept Genetic Algorithm 0.1% tests accept Change Detection	9.0×10^{-4}

Comparing with Table 5, the adaptive decision fusion with hypothesis change detection still has good performance when the local sensors are time-varying. The adaptive decision fusion with hypothesis change detection can adapt the time-varying of local sensor performance.

6.3.3 Tests on different true-hypothesis-change-rate

We simulate a fusion system with 4 identical local sensors. Local sensors have fixed performance $p_f = p_m = 0.2$. $M = 10$. We simulated 12 cases with different true-hypothesis-change-rate. We compare the overall performance (average global error rate) of the 12 cases to investigate how the true-hypothesis-change-rate influences the performance of hypothesis change detection.

TABLE VIII. TEST THE SIMULATION PERFORMANCE OF THE ADAPTIVE DECISION FUSION WITH HYPOTHESIS CHANGE DETECTION ON DIFFERENT HYPOTHESIS CHANGE RATE.

	Hypothesis Change Rate	Average Global Error Rate
Case 1	1 Change per 100 Bits	4.6×10^{-3}
Case 2	1 Change per 90 Bits	4.6×10^{-3}
Case 3	1 Change per 80 Bits	4.6×10^{-3}
Case 4	1 Change per 70 Bits	5.1×10^{-3}
Case 5	1 Change per 60 Bits	5.7×10^{-3}

Case 6	1 Change per 50 Bits	6.4×10^{-3}
Case 7	1 Change per 40 Bits	8.8×10^{-3}
Case 8	1 Change per 30 Bits	1.5×10^{-2}
Case 9	1 Change per 20 Bits	2.4×10^{-2}
Case 10	1 Change per 15 Bits	1.1×10^{-1}
Case 11	1 Change per 10 Bits	3.9×10^{-1}
Case 12	1 Change per 8 Bits	4.2×10^{-1}

The simulation results show that the hypothesis-change-rate can have large influence on the performance of hypothesis change detection. When $M = 10$ and the hypothesis changes frequently (≥1 change per 30 bits), the detection performance of the hypothesis change detection starts to have large decrease.

The adaptive decision fusion with genetic algorithm we proposed in chapter 6 has good performance in stationary environment (true hypothesis does not change). But, if the environment is non-stationary (true hypothesis changes), the genetic fusion algorithm cannot function well. To make our adaptive decision fusion algorithm be able to adapt the non-stationary environments, we developed a hypothesis-change-detection mechanism for the adaptive decision fusion with genetic algorithm. We design the hypothesis change detection to have two procedures: (1) local sensor level detection, and (2) global level detection. In local sensor level detection, N (number of local sensors) local sensor level decisions are made. The global level decision is obtained by combining all N sensor level decisions. Adding the hypothesis change detection mechanism greatly improves the performance of adaptive decision fusion with genetic algorithm in non-stationary environments (averagely decreases the overall global error rate from 1.188×10^{-1} to 4.6×10^{-3} in our experiments).

7 Summary

We investigated elements of the theory and applications of decision fusion in a parallel distributed detection environment. In the domain of probabilistic decision making, the most common optimization criteria used at the fusion center are (1) the Bayes' criterion: minimization of a Bayes' risk function, and (2) the Neyman-Pearson criterion: maximization of detection rate under a constrained false alarm rate. In this thesis, we investigate development of optimal fusion algorithms under both criteria.

We designed an optimal algorithm that computes local detector and the fusion center decision thresholds for a distributed binary hypothesis decision fusion problem. The algorithm assumes that both the local detectors and the fusion center use the Neyman-Pearson criterion. The key computational requirement is to find the roots of a certain N^{th} order polynomial. We compare the performance of our method with the performance of the traditional Person-by-Person Optimization approach and that of a centralized detection scheme.

We also delved into applications of parallel distributed detection architectures using the Chair and Varshney rule. First, we developed a parallel decision fusion system where each local sensor is a sequential decision maker that implements the modified Wald's sequential probability test (SPRT) as proposed by Lee and Thomas (1984). We evaluated the performance of the sensor bank by two criteria: (1) the probability of error; (2) average sample number (ASN) needed to achieve it. We examined three fusion rules for sequential sensors in a parallel configuration. We compared the performance of these rules in terms of the global error rate and average sample number (ASN). The first rule is First-to-decide rule: the fusion center chooses one of the first set of local sensors' decisions it receives as the global decision; the sensors that have not reached a decision are ignored. The second rule is the All-that-decided rule: once at least one sensor has stopped sampling, we integrate all the decisions of stopped sensors through the Chair and Varshney decision fusion rule. The third rule is the All-sensors rule: once at least one sensor has stopped sampling, we combine the available decisions of the stopped sensor and the implied decisions of the remaining sensors. Among the three rules examined, the All-sensors rule had the lowest global error rate, with the same average-sample-number as the other two rules.

Under the schemes where the fusion center minimizes a Bayes' risk function to optimally combine local detector decisions, the Chair and Varshney rule is widely popular. However, the rule assumes complete knowledge of the prior probabilities of the hypotheses and the error characteristics of the local detectors. In real applications these are usually unavailable. A major contribution of this thesis was to present an alternative fusion rule based on the standard genetic algorithm that circumvents the need for this assumption. The framework aids in combining local decisions to obtain the global decision without using the local sensors' performance parameters and prior probabilities. In our simulations, the algorithm was able to adapt to the variations in local sensor performance. The genetic fusion algorithm showed nearly equivalent detection accuracy both under fixed and time-varying local error rates and achieved nearly equivalent performance in detection accuracy as compared to that obtained from the Chair and Varshney fusion rule. The cost of this performance was that the algorithm required large sets of local decisions to form the initial population. Furthermore, iterating over the steps of the genetic algorithm causes processing delays and requires computational overheads.

When applying the adaptive fusion with genetic algorithm, it needs M-bits of each local sensor's decisions to obtain a global decision. If the environment is non-stationary (the true hypothesis changes during collection of the M-bits), the fitness of the sensor's genetic code can misrepresent the correct information which thereby would lead to improper functioning of adaptive fusion with genetic algorithm. To address the problem of time varying hypotheses, we incorporated a hypothesis change detection mechanism in the genetic algorithm setup such that even when the true state of the phenomenon being observed (hypothesis) changed, the fusion architecture could track such changes and continue decision making. The new mechanism first decided whether to apply the hypothesis change detection or to accept the genetic algorithm's decision. If the system decided to apply the hypothesis change detection, the system will accept the hypothesis change detection's decision. On the other hand, if the system decided not to use the hypothesis change detection, the system global decision will accept the genetic fusion algorithm's decision. The change detection had two procedures: (1) local sensor level detection, and (2) global level detection. We carried out several simulations under different circumstances to test the performance of hypothesis change detection. We tested the hypothesis change detection with both fixed and time-varying sensors. We also investigated the influence of the hypothesis change rate on the performance of hypothesis change detection. With the hypothesis change detection mechanism added in the adaptive fusion rule, the performance of the adaptive fusion with genetic algorithm gotoulous improved.

Bibliography

[1] Z. Chair and P. K. Varshney, “Optimal data fusion in multiple sensor detection systems,” *IEEE Transactions on Aerospace and Electronic System*, vol. 22, no. 1. pp. 98-101, 1986.

[2] R. Tenney and N. Sandell, “Detection with distributed sensors,” *IEEE Transactions on Aerospace and Electronic System*, vol. 17, no. 4, pp. 501–510, July 1981.

[3] A. Wald, *Sequential Analysis*. New York: Wiley, 1947

[4] H. Blasbalg, “Experimental results in seguential detection,” *IRE Transactions in Information Theory*, vol. 5, pp. 41–51, June 1959.

[5] C. C. Lee and J. B. Thomas, “A modified sequential detection procedure,” *IEEE Transactions in Information Theory*, vol. 30, No. 1, pp. 16–23, Jan. 1984.

[6] S. Thomopoulos, R. Viswanathan, and D. Bougoulias, “Optimal distributed decision fusion,” *IEEE Transactions on Aerospace and Electronic System*, vol. 25, no. 5, pp. 761–765, Sep 1989.

[7] M., Kam, W. , Chang and Q. , Zhu, “Hardware complexity of binary distributed detection systems with isolated local detectors,” *IEEE Transactions on Systems*, Man and Cybernetics, vol. 21, No. 3, pp. 565-571, 1991

[8] R. Niu and P. K. Varshney, “Sampling schemes for sequential detection with dependent observations,”, *IEEE Transactions in Signal Processing,* vol. 58, No. 3, pp. 1469–1481, March 2010.

[9] I. Y. Hoballah and P. K. Varshney, “Distributed bayesian signal detection,” *IEEE Transactions on information Theory*, vol. 35, no. 5, pp. 905-1000,Sep 1989.

[10] I. Y. Hoballah and P. K. Varshney, “Neyman-pearson detection with distributed sensors,” in *25th IEEE Conference on Decision and Control*, vol. 25, pp. 237–241, Dec 1986.

[11] D.Bauso and R. Pesenti, “Generalized person-by-person optimization in team problems with binary decisions”, in *American Control Conference*, pp. 717–722, 2088.

[12] M. Kam, Q. Zhu, and W. S. Gray, "Optimal data fusion of correlated local decisions in multiple sensor detection system," *IEEE Transactions on Aerospace and Electronic System*, vol. 28, no. 3, pp. 916–920, 1992.

[13] A. Naim and M. Kam, "On-line estimation of probabilities for distributed bayesian detection," *Automatica*, vol. 30, no. 4, pp. 633–642, 1994

[14] N. Ansari, E. S. Hou, B. O. Zhou, J. G. Chen, "Adaptive fusion by reinforcement learning for distributed detection systems," *IEEE Transactions on Aerospace and Electronic System*, vol. 32, no. 2, pp. 524–531, 1996.

[15] G. Mirjalily, Z. Q. Luo, T. N. Davidson, and E. Bosse, "Blind adaptive decision fusion for distributed detection," *IEEE Transactions on Aerospace and Electronic System*, vol. 39, no. 1, pp. 34–52, 2003.

[16] Ansari, N., Hou, E., Zhu, B., and Chen, J., "An adaptive fusion model for distributed detection systems". *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 2, pp. 524-531, 1996.

[17] Chen, J., and Ansari, N., "Adaptive fusion of correlated local decisions". *IEEE Transactions on Systems , Man and Cybernetics–Part C: Applications and Reviews*, vol. 28, no. 2, 276–281, 1998.

[18] Ansari, N., Chen, J-G., and Zhang, Y-Z., "Adaptive decision fusion for unequiprobable sources". IEE Proceedings on Radar, Sonar, Navigation, vol. 144, no. 3, pp. 105–111, 1997.

[19] L. Davis, *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, 1991.

[20] M. Melanie, *An Introduction on Genetic Algorithms*. Cambridge, London, 1998.

[21] M. Skolnik, *Introduction on Radar System*. McGraw-Hill, 1981.

[22] R. R. Salakhutdinov and G. E. Hinton, "Deep boltzmann machines," *Conference on Artifical Intelligence and Statistics*, vol. 12, 2009

[23] H. Lee, C. Ekanadham, and A. Ng, "Sparse deep belief net model for visual area V2", In *NIPS*, 2007.

[24] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Ng, "Multimodal deep learning". In *Proceedings of the 28th international conference on machine learning*, 2011.

[25] N. Srivastava and R. R. Salakhutdinov, "Multimodal learning with deep boltzmann machines", In Advances in neural information processing systems, 2012.

[26] W. Ding, J. Zhu, L. Guo, X. Hu, J. Luo and H. Wang, "Jointly image topic and emotion detection using multi-modal hierarchical latent dirichlet allocation", *Journal of Multimedia Information System*, 2014

[27] M. M. Al-Ibrahim and P. K. Varshney, "A simple multi-sensor sequential detection procedure," in Proceedings of the 27th Conference on Decision and Control, Austin, TX, Dec. 1988.

[28] P. K. Varshney, "*Distributed Detection and Data Fusion*". New York: Springer, 1997.

[29] R. Tenney and N. Sandell, "Detection with distributed sensors," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 17, no. 4, pp. 501 –510, July 1981.

[30] A. Reibman and L. Nolte, "Optimal detection and performance of distributed sensor systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-23, no. 1, pp. 24 –30, Jan. 1987.

[31] F. Sadjadi, "Hypotheses testing in a distributed environment," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-22, no. 2, pp. 134 –137, March 1986.

[32] V. Aalo and R. Viswanathan, "On distributed detection with correlated sensors: two examples," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-25, no. 3, pp. 414 –421, May 1989.

[33] E. Drakopoulos and C. Lee, "Optimum fusion of correlated local decisions," in *Proceedings of the 27th IEEE Conference on Decision and Control*, pp. 2489–2494 vol.3, 1988.

[34] S. Thomopoulos, R. Viswanathan, and D. Bougoulias, "Optimal distributed decision fusion," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 25, no. 5, pp. 761 –765, Sep 1989.

[35] J. Tsitsiklis and M. Athans, "On the complexity of decentralized decision making and detection problems," *IEEE Transactions on Automatic Control*, vol. AC-30, no. 5, pp. 440–446, 1985.

[36] H. Van "Trees, Detection, Estimation, and Modulation Theory", *ser. Detection, Estimation, and Modulation Theory*, no. 1 Wiley, 2004.

[37] J. N. Tsitsiklis, "Decentralized detection," in *Advances in Statistical Signal Processing*. JAI Press, pp. 297–344, 1993.

[38] Veeravalli V V, BaSar T, Poor H V. "Decentralized sequential detection with a fusion center performing the sequential test". *IEEE Transactions on Information Theory*, vol. 39, no. 2, pp. 433–442, 1993.

[39] S. Acharya, J. Wang, M. Kam. "Distributed decision fusion using the Neyman-Pearson criterion", *17th International Conference on Information Fusion*, July 2014.

[40] Teneketzis D, Ho Y C. "The decentralized Wald problem". *Information and Computation*, vol. 73, no. 1, pp. 23–44, 1987.

[41] Ruslan Salakhutdinov and Geoffrey E. Hinton. "Replicated softmax: an undirected topic model". In *NIPS*, pp 1607–1614, 2009.

[42] A Bosch, Andrew Zisserman, and X Munoz. "Image classification using random forests and ferns". *IEEE 11th International Conference on Computer Vision*, vol. 23, pp. 1–8, 2007.

[43] C. Cusano, G. Ciocca, and R. Schettini, "Image annotation using SVM," in *Electronic Imaging 2004*, pp. 330-338, 2004.

[44] D. Putthividhy, H. T. Attias, and S. S. Nagarajan, "Topic regression multi-modal latent dirichlet allocation for image annotation," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 3408-3415, 2010.

[45] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993-1022, May 2003.

[46] V. Vonikakis and S. Winkler, "Emotion-based sequence of family photos," in *Proceedings of the 20th ACM international conference on Multimedia*, pp. 1371-1372, 2012.

[47] C. Lin and Y. He, "Joint sentiment/topic model for sentiment analysis," in *Proceedings of the 18th ACM conference on Information and knowledge management*, pp. 375-384 , 2009.

[48] W. X. Zhao, J. Jiang, H. Yan, and X. Li, "Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 56-65, 2010.

[49] X. Xu, S. Tan, Y. Liu, X. Cheng, and Z. Lin, "Towards jointly extracting aspects and aspect-specific sentiment knowledge," in *Proceedings of the 21st ACM international conference on Information and knowledge management*, pp. 1895-1899, 2012.

[50] W. Wei-ning, Y. Ying-lin, and J. Sheng-ming, "Image retrieval by emotional semantics: A study of emotional space and feature extraction," in *Systems, Man and Cybernetics, 2006. SMC'06. IEEE International Conference on*, pp. 3534-3539, 2006.

[51] Y. Guo and H. Gao, "Emotion recognition system in images based on fuzzy neural network and HMM," in *Cognitive Informatics, 2006. ICCI 2006. 5th IEEE International Conference on Cognitive Informatics*, pp. 73-78, 2006.

[52] Hinton G E, Salakhutdinov R R., “Reducing the dimensionality of data with neural networks”. *Science*, vol. 313, pp. 504-507, 2006.

[53] Cortes C, Vapnik V., “Support-vector networks”. *Machine Learning*, vol. 20, pp. 273-297, 1995.

[54] Chang C C, Lin C J., “LIBSVM: a library for support vector machines”. *ACM Transactions on Intelligent Systems and Technology* (TIST), vol. 2, pp. 27, 2011.

[55] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet allocation”, *Journal of Machine Learning Research*, vol. 3, pp. 993-1022, 2003.

[56] W. Ding, J. Zhu, X. Hu, “Jointly Image Topic and Emotion Detection using Multi-Modal Hierarchical Latent Dirichlet Allocation”, *Journal of Multimedia Information System*, vol. 1, pp. 55-67, 2014