Forty-Fifth Annual Allerton Conference
Allerton House, UIUC, Illinois, USA
September 26-28, 2007

Design of Distributed Detection Systems with Correlated Heterogeneous Sensors

Kalyan Veeramachaneni *Student Member*, Lisa Osadciw, *Senior Member, IEEE*

Abstract— **The optimization of sensor thresholds and fusion rule for heterogeneous and correlated sensor suite is accomplished through a particle swarm optimization algorithm. Different correlation structures are assumed and the effect of correlation on the choice of final fusion rule and thresholds is analyzed. Optimal decision fusion for correlated sensors includes estimation of 2^n joint probabilities. Bahadur-Lazarfeld expansion is used to reduce the computational burden. Bahadur-lazarfeld expansion reduces the burden to evaluation of N joint probabilities. Numerical integration is proposed to evaluate the joint probability. Examples using Gaussian distributions assumed under both the hypothesis are presented. The results achieved using a particle swarm optimization algorithm are compared to the traditional decision fusion strategies based on the fusion methodology developed by Moshe Kam et al.**

I. INTRODUCTION

In decision level fusion, using binary hypothesis, each sensor applies thresholds and gives decision regarding the presence or absence of phenomena. The decisions from multiple sensors are fused at the fusion processor producing a decision. This type of decision-making minimizes the communication bandwidth of a sensor network by transmitting a single bit representing the decision to the fusion center.

Apart from the obvious bandwidth advantages decision level fusion weighs each sensor's decision based on the sensor's performance characteristics. The weighting of a sensor's observation (decision) is buried in the decision fusion processing. Hence, decision level fusion can be very advantageous in accuracy as well as bandwidth when the sensors are statistically disparate.

The Majority voting rule and Chair-Varshney [13] optimal fusion rule are two examples of decision-level fusion schemes. The Chair- Varshney (CVR) [13] optimal decision fusion rule is achieved using the individual sensor performance indices. The resulting optimal fusion rule from the CVR can be the majority-voting rule but is not limited to it.

The biggest problem preventing optimum performance in decision level fusion however, is the optimal setting of individual decision thresholds. There are $2^{2^{\wedge N}}$possible fusion rules for a binary hypothesis and N sensors system. Most of the sensor fusion work done in the past neglects all possible rules at decision level. Also, the decision threshold for each individual sensor is optimally set to minimize the error [2]. This is done even before fusion is carried out. This typically entails selection of an operating point from the Receiver Operating Characteristic (ROC) curve for the individual sensors, which will minimize the error for the given costs of false alarm. Once the decision thresholds for individual sensors are set, the majority voting rule or the CVR is used as the fusion rule. This method, however, does not guarantee optimum performance after fusion. Error is defined under Bayesian criterion in this paper.

Most fusion work assumes independence and the CVR is applied minimizing errors. Some of our work in the past also optimized the fusion system under the independence assumption [19]. There have been some studies dealing with correlation among sensors. Kam et al. [15] presented optimal fusion rule for a distributed detection system with correlated sensors. The local sensor thresholds are fixed before finding the optimal fusion rule. Willet et al. evaluated different fusion rules for 2 sensors and defined regions of correlation in which different rules can be applied. They also concluded that the optimal fusion rule, in the case of dependence, is not limited to monotonic fusion rules [17]. This makes the problem of searching for optimal thresholds and optimal fusion rule intractable.

Tang et al. presented the gradient descent approach for correlated sensors under multiple hypotheses [24]. The algorithm is designed for sensors that have strictly concave receiver-operating curves, ROCs. Special conditions need to be fulfilled for the ROCs to be concave [23]. The algorithm is a top down approach and relies on gradient information. The variable of interest is determined, while others are fixed. When sensor observations are correlated, this requires further approximation and poor performance.

The wide variety of applications of sensor systems relying on statistical inference and distributed detection motivates this research. The ROCs of the sensors do not satisfy the required conditions, in order to be solved by the traditional optimization approaches. For example, for the detection of an intruder the likelihoods and the ROCs are obtained using training data and are not strictly concave.

In this paper, a particle swarm optimization based algorithm is presented to design the optimal local thresholds and the fusion rule for a distributed detection with correlated heterogeneous sensors. Particle swarms start with multiple solutions and use the performances of these solutions to further guide the search. The algorithm is compared to two

Kalyan Veeramachaneni and Lisa Osadciw are with Department of Electrical Engineering and Computer Science at Syracuse University, Syracuse, NY 13244

traditional decision fusion strategies.

A two-step optimization procedure is adopted for this case. First the optimal thresholds are achieved using the likelihood ratio test (LRT). The second step involves determination of the optimal fusion rule using the formulation presented in this paper. First, ignoring the underlying correlation, CVR is applied, determining the fusion rule. A generalized version of the CVR is adopted from [15] that deals with the correlation between the sensors is applied as a second strategy. The performance benefits due to use of this new rule over CVR are illustrated. The PSO based optimization technique is applied and the resultant is compared with the two strategies detailed in previous paragraph.

The paper presents the results achieved for sensors whose distributions under both the hypotheses are assumed to be multivariate Gaussian with a given mean and covariance matrix. The experimental sensor suite used in this paper is presented in the APPENDIX.

The rest of paper is organized as follows. In section 2, decision level fusion is detailed. The estimation of error probabilities in for correlated and independent sensors at decision level fusion are presented. The traditional decision fusion strategies are detailed. The problem of optimal decision fusion in case of independent sensors and correlated sensors is presented as well as its intractability. PSO techniques are discussed in Section 3. Formulation of the particle for this problem, Bayesian cost function, and PSO settings are detailed in this section. Section 4 presents the results achieved on multivariate Gaussian distributions. Section 5 concludes the paper.

II. Distributed Detection

Consider a binary hypothesis-testing problem with sensors evaluating observations that are conditionally dependent, the two hypotheses are

H_0 : Presence of phenomena

H_1 : Absence of phenomena

The two types of errors commonly known as probability of false alarm and probability of miss are

$$P_{FA} = P(U_0 = 1 / H_0) \quad (1)$$

$$P_M = P(U_0 = 0 / H_1) \quad (2)$$

Where U_0 is the decision of the fusion processor, which takes in the decisions from the local sensors and fuses them, using the fusion rule. Also probability of detection is given by,

$$P_D = 1 - P_M$$

In the following sections, these error probabilities for fusion involving independent and dependent sensors are derived.

A. Bayesian Error in Decision Fusion

In this paper an assumption of equal a prior probabilities is made, and an additional cost of making errors is defined, which is the cost of false alarm and cost of miss. These are used to evaluate the fusion system performances. The Bayesian cost (error), which the paper intends to minimize, is

$$R = C_{FA} \times P_{FA} + C_M \times P_M \quad (3)$$

where

$$C_{FA} = 2 - C_M \quad (4)$$

C_{FA} is cost of false alarm, C_M is cost of a miss and P_{FA} and P_M are the error probabilities defined in (1) and (2) and are derived in following sections.

Equation (3) is a weighted multi-objective function, which is minimized by the optimization routine. It is assumed that the costs of making an error are given as requirements to the system. Note that the costs of detection and true miss are set to zero and are not part of (3).

B. Decision Level Fusion for N Sensor System Under the Assumption of Independence

In this section the calculation of the error probabilities for a N sensor system given the individual thresholds and fusion rule is derived. P_{FA}, P_M of the fused system is calculated from the fusion rule and j[th] sensor's ${P_{FA}}^j$ and ${P_M}^j$.

TABLE 1: FUSION RULE FORMATION FOR TWO SENSORS

u_1	u_2	f
0	0	d_0
0	1	d_1
1	0	d_2
1	1	d_3

For example, with two sensors the fusion rule consists of 4 bits, as presented in Table 1. In Table 1, u_1 is the first sensor decision, and u_2 is the second sensor decision. The fusion rule is of length *l* bits where

$$l = \log_2 p \quad (5)$$

where $p = 2^{2^N}$, and, N is the number of sensors.

The global decision replaces { d_0, d_1, d_2, d_3 } with 0s and 1s in their respective locations within *f*. The global error rates can then be computed directly from

$$P_{FA} = \sum_{i=0}^{p-1} d_i \times \left\{ \prod_{j=1}^{N} er_j \right\} \quad (6)$$

where

$$er_j = \begin{cases} 1 - {P_{FA}}^j, & (u_j = 0) \\ {P_{FA}}^j, & (u_j = 1) \end{cases} \quad (7)$$

and

$$P_M = \sum_{i=0}^{p-1} (1 - d_i) \times \left\{ \prod_{j=1}^{N} er_j \right\} \quad (8)$$

where

$$er_j = \begin{cases} P_M{}^j, & (u_j = 0) \\ 1 - P_M{}^j, & (u_j = 1) \end{cases} \quad (9)$$

$P_{FA}{}^j$ is the probability of false alarm of the j[th] sensor and is given by

$$P_{FA}{}^j = P(u_j = 1 / H_0) \quad (10)$$

$$P_{FA}{}^j = \int_{\lambda_j}^{\infty} P(x_j / H_0) dx_j \quad (11)$$

and

$$P_M{}^j = P(u_j = 0 / H_1) \quad (12)$$

$$P_M{}^j = \int_{-\infty}^{\lambda_j} P(x_j / H_1) dx_j \quad (13)$$

where λ_j is the threshold for the sensor. x_i is the raw output of the sensor conditioned over H_1 or H_0. Hence, P_{FA} is a function of thresholds (local decision rules) for the sensors and the optimal fusion rule or simply can be shown as

$$P_{FA} = g(\lambda_1, \lambda_2 ....\lambda_n, f) \quad (14)$$

where, '*f*' is the fusion rule. Similarly, P_M is also the function of local decision rules and fusion rule.

1) Chair-Varshney Optimal Fusion Rule

Traditional decision level fusion strategies apply the maximum likelihood ratio test as in,

$$\log \frac{P(x_j / H_1)}{P(x_j / H_0)} \underset{H_1}{\overset{H_0}{\lessgtr}} \log \frac{C_{FA}}{C_M} \quad (15)$$

to derive the optimal threshold for each sensors measurement. Then the threshold is applied to arrive at a hard decision.

The hard decisions can be combined using a majority voting rule or CVR as in

$$\sum_{j=1}^{N} \left[u_j \log \left\{ \frac{1 - P_{M_j}}{P_{FA_j}} \right\} + (1 - u_j) \log \left\{ \frac{P_{M_j}}{(1 - P_{FA_j})} \right\} \right] \underset{H_1}{\overset{H_0}{\lessgtr}} \log \left(\frac{C_{FA}}{2 - C_{FA}} \right) \quad (16)$$

where u_j is the decision of the j[th] sensor. In this paper, comparisons are done between the optimized decision level fusion and traditional decision level fusion. It is important to realize that the Chair-Varshney (CVR) rule simplifies the problem by assuming orthogonality of the sensors. The CVR thus minimizes the Bayesian error cost function in (3) while oversimplifying the problem resulting in a non-optimum fusion rule as well as over estimating performance of the rule.

C. Decision Level Fusion for N Sensor System With Correlation

In the previous section, the error probabilities are estimated for the case of orthogonal sensors. The resulting independence decouples all the joint probabilities allowing joint probabilities of differently grouped local sensor decisions to be estimated using individual error probabilities. Hence, reducing performance estimates to only N, i.e. (11) and (13).

In this section, the error probabilities are estimated in the case of correlation. Let $U = [u_1, u_2 .....u_n]$ be the vector of local decisions. It is assumed that sensors are correlated under both the hypotheses. This leads to a generic, more complex formulation of the problem. Note that for 'N' sensors we need to estimate the joint probabilities for 2^n combinations in U. For each combination, a multivariate integral needs to be evaluated, which is an expensive operation. This process can become tedious as number of sensors increase.

An alternative method uses Bahadur-Lazarfeld expansion [21, 15]. Bahadur-Lazarfeld expansion allows the computation of all the joint probabilities by estimating only N multivariate integrals, where N is the number of sensors. Alternative methods of expansions and evaluation of these joint probabilities are given in [21]. The method involves the following steps:

1. The derivation first assumes normalized local decisions giving a random variable z_j with zero mean and unit variance given by

$$z_{j_h} = \frac{u_j - p_{j_h}}{\sqrt{p_{j_h} q_{j_h}}} \quad (17)$$

where, $p_{j_h} = P(u_j = 1 | H_h)$ and $q_{j_h} = 1 - p_{j_h} \ \forall j$.

2. Let

$$\overline{P}_{h=1}(U) = \prod_{j=1}^{n} (p_{j_1})^{u_j} (q_{j_1})^{(1-u_j)} \quad (18)$$

and

$$\overline{P_{h=0}}(U) = \prod_{j=1}^{n} (p_{j_0})^{(1-u_j)} (q_{j_0})^{u_j} \quad (19)$$

Note: (18) and (19) are equivalent to the products as the in (6) and (8). (6) and (8) give joint probability for a given local decision vector in the case orthogonality.

3. In the case of correlation, the joint probability can be estimated using [21, 15]

$$P_h(U) = \overline{P}_h(U) \left[1 + \sum_{i<j} \gamma_{ij_h} z_{i_h} z_{j_h} + \sum_{i<j<k} \gamma_{ijk_h} z_{i_h} z_{j_h} z_{k_h} + ... \right] \quad (20)$$

It is clear from (20) that the joint probability estimates in the case of correlation are the sum of the joint probabilities from the independence case except for an additional correlation factor.

4. The $\prod_{j=1}^{k} z_{j_h}$'s are the Bahadur-Lazarfeld polynomials given in [21, 15]. The variable, γ, is the correlation coefficient and is given by

$$\gamma_{123...n_h} = E\left(\prod_{j=1}^{n} z_{j_h} | H_h \right) \quad (21)$$

where

$$z_{i_h} = \frac{u_i - P(u_i = 1/H_h)}{\sqrt{P(u_i = 1/H_h) \times (1 - P(u_i = 1/H_h))}} \quad (22)$$

There are $2^N - 1$ coefficients for 'N' sensors. In (20) however, only $2^N - 1 - N$ are calculated since the correlation coefficients are zero for 'N' of them with single z's. In many situations higher ordered coefficients can be ignored [21, 15].

Note that γ is independent of the local decision vector U. It is the expected value of product of z_j*'s* conditioned on the hypothesis *h*. In (20) γ's are multiplied by z_j*'s* which are dependent on the local decision vector U as given in (22). An expansion of γ in case of two sensors is given by

$$\gamma_{ij_h} = \frac{E(u_i u_j \mid H_h) - (P(u_i = 1 \mid H_h) \times P(u_j = 1 \mid H_h))}{\prod_{k=i,j} \sqrt{P(u_k = 1 \mid H_h) \times (1 - P(u_k = 1 \mid H_h))}} \quad (23)$$

Based on the joint probability estimates derived for the conditional dependence case the global error probabilities are estimated to be

$$P_{FA} = \sum_U d_i \times \{P_0(U)\} \quad (24)$$

and

$$P_M = \sum_U d_i \times \{P_1(U)\} \quad (25)$$

where, d is from table 1 and similar to as in (6), $P_h(U)$ are given by (20).

1) Moshe Kam (MKR) Optimal Fusion Rule

The individual thresholds for the sensors are found using the LRT given in (15). Given the joint probability estimates under both the hypotheses for different local decision vectors, the optimal fusion rule is again derived using the well-known LRT [15] as in

$$\log \frac{P_1(U)}{P_0(U)} \underset{H_1}{\overset{H_0}{\lessgtr}} \log \frac{C_{FA}}{C_M} \quad (26)$$

which results in,

$$\Omega + \left[\log \left[\frac{1 + \sum_{i<j} \gamma_{ij_1} z_{i_1} z_{j_1} + \sum_{i<j<k} \gamma_{ijk_1} z_{i_1} z_{j_1} z_{k_1} + \ldots\ldots}{1 + \sum_{i<j} \gamma_{ij_0} z_{i_0} z_{j_0} + \sum_{i<j<k} \gamma_{ijk_0} z_{i_0} z_{j_0} z_{k_0} + \ldots\ldots} \right] \right] \underset{H_1}{\overset{H_0}{\lessgtr}} \Lambda(C_{FA}) \quad (27)$$

where, Ω corresponds to the left hand side of (16), $\Lambda(C_{FA})$ corresponds to the right hand side of (16). This is traditional decision fusion strategy applied in case of conditional dependence. This gives close to optimal performance if the sensors are statistically identical. This is not the case in many real world applications. The statistical disparity among sensors is significantly high. The two-step optimization procedure adopted leads to poorly performing distributed detection systems. To achieve the full benefits of the distributed detection one needs to simultaneously optimize the thresholds and the fusion rule. The complexity of optimizing thresholds and the fusion rule is NP Complete [23].

III. Particle Swarm Optimization

The PSO algorithm was originally introduced in terms of social and cognitive behavior by Kennedy and Eberhart in 1995 [3]. The power in the technique is its simple computations and sharing of information as it internally communicates imitating the social behavior of individuals. The individuals, called particles are flown through the multi-dimensional search space with each particle representing a possible solution to the multi-dimensional problem. Each solution's fitness is based on a multi-objective performance function related to the optimization problem being solved.

The movement of the particles is influenced by two factors using information from iteration-to-iteration as well as particle-to-particle. As a result of iteration-to-iteration information, the particle stores in its memory the best solution visited so far, called *pbest,* and experiences an attraction towards this solution as it traverses through the solution search space. As a result of the particle-to-particle information, the particle stores in its memory the best solution visited by any particle, and experiences an attraction towards this solution, called *gbest,* as well. The first and second factors are called cognitive and social components, respectively. After each iteration the *pbest* and *gbest* are updated for each particle if a better or more dominating solution (in terms of fitness) is found. This process continues, iteratively, until either the desired result is converged upon, or it's determined that an acceptable solution cannot be found within computational limits.

The PSO formulae define each particle in the D-dimensional space as $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$, where the subscript 'i' represents the particle number and the second subscript is the dimension. The memory of the previous best position is represented as $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$ and a velocity along each dimension as $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$. After each iteration, the velocity term is updated and the particle is pulled in the direction of its own best position, P_i and the global best position, P_g, found so far. This is apparent in the velocity update equation, [3], as in

$$V_{id}^{(t+1)} = \omega \times V_{id}^{(t)} + U[0,1] \times \psi_1 \times (p_{id}^{(t)} - x_{id}^{(t)}) + U[0,1] \times \psi_2 \times (p_{gd}^{(t)} - x_{id}^{(t)}) . \quad (28)$$

$$X_{id}^{(t+1)} = X_{id}^{(t)} + V_{id}^{(t+1)} \quad (29)$$

where U [0,1] is a sample from a uniform random number generator, *t* represents a relative time index, ψ_1 is a weight determining the impact of the previous best solution, and ψ_2 is the weight on the global best solution's impact on particle velocity. For more details of the particle swarm optimization algorithm the reader is referred to [11].

A. PSO for Decision Level Fusion

Each particle in this problem has 'N' dimensions, where N is the number of sensors in the sensor network. Each of the N dimensions is a threshold at which a particular sensor is set. The fusion rule, which determines how all the decisions from the sensors are fused is calculated using the MKR. Hence the representation of each particle is, $X_i = (\lambda_{i1}, \lambda_{i2}..........\lambda_{in})$. The sensor thresholds are continuous. The fusion rule, however, is a binary number having a length of $\log_2 p$ bits, where $p = 2^{2^N}$ for 'N' sensors, with a decimal value varying from $0 \le dec(f) \le p-1$. The two objectives for this problem are given by (24), (25). The goal is to minimize both the P_{FA} and P_M. At each iteration, the particles representing the solution for the problem are evaluated for these objectives using the weighted cost function (3). The memory of the particle is updated if it finds better minima. The particles are moved in the search space based on equations (28) and (29) and these steps are iteratively repeated till convergence occurs or the requirements are fulfilled.

Start PSO Algorithm

1. Initialize sensor models (mean and covariance matrix), PSO parameters
2. Initialize 'q' particles randomly each being the solution, $X_i = (\lambda_1, \lambda_2..........\lambda_n)$; initialize P_i to be same.
3. Cost Evaluation *
   For each particle;
   a. Evaluate $(P_{FA_1},.......P_{FA_n})$ and $(P_{M_1},.....P_{M_n})$ using (11) and (13)
   b. Obtain the optimal fusion rule using (27)
   c. Obtain the error probabilities using (24) and (25)
   d. Calculate the Bayesian cost for the particle given by (3), and call it E_q
   End
4. Determine the best performing particle, 'g', P_g
5. Main Loop
   For t= 1 to the max. bound of the number on iterations,
   For i=1 to the population size
   For d=1 to the problem dimensionality,
   Apply the velocity update equation:
   $$V_{id}^{(t+1)} = \omega \times V_{id}^{(t)} + U[0,1] \times \psi_1 \times (p_{id}^{(t)} - x_{id}^{(t)}) + U[0,1] \times \psi_2 \times (p_{gd}^{(t)} - x_{id}^{(t)})$$
   where, P_i is the best position visited so far by X_i, P_g is the best position visited so far by any particle
   Update Position:
   $$X_{id}^{(t+1)} = X_{id}^{(t)} + V_{id}^{(t+1)}$$
   End- for-d;
   Compute cost for each particle using Step 3.
   If needed, update historical information regarding Pi and Pg;
   End-for-i;
   Terminate if Pg meets problem requirements;
   End-for-t;

End PSO

Figure 1: Pseudo Code for the Algorithm determining the optimal thresholds using PSO

20 particles are used for simulation. The parameter values used are $\psi_1 = \psi_2 = 1$, $\omega = 0.8$

PSO is a bottoms up approach. The algorithm starts out with a few solutions randomly initialized in the search space. The objective function value for each particle is then calculated and these values are used to move the solutions in the search space.

IV. Results on Gaussian Examples

Sensors are modeled using Gaussian distributions under both the hypothesis. The sensor models used in this paper are given in Table 3. The covariance matrix is constructed using (33). The swarm optimization based algorithm is run on two types of correlation structures. The first one is symmetric correlation. In symmetric correlation, the same correlation is assumed under both the hypothesis. In the second set of experiments, a different correlation is assumed under both hypotheses. The correlation used in this paper under the hypotheses and between different sensors is given in Table 4. Note, that LRT for the sensor models presented in this paper is quadratic, requiring two thresholds. The PSO however, designs the solution with single threshold.

Let us consider a multivariate Gaussian density function under both the hypotheses given by

$$f_{X_1,X_2,....X_n}(x_1,x_2...x_n \mid H_h) = \frac{|[C_X]^{-1}|^{1/2}}{(2\pi)^{N/2}} \exp\left\{-\frac{\left[x-\overline{X}\right]^t [C_X]^{-1} \left[x-\overline{X}\right]}{2}\right\} \quad (30)$$

where, $[C_X]$ is the covariance matrix and $\left[x-\overline{X}\right]$ is the matrix where each variables mean is subtracted from the value at which the density is to be found. These multivariate Gaussian distributions with the means $[\mu_X]$ and covariance matrix $[C_X]$ are assumed to be known under both the hypotheses. The joint probability estimates under both hypotheses can be evaluated using (20) for a given local decision vector defined by U. However, estimation of the joint probability functions needs the calculation of the cumulative density function for the multivariate normal as in

$$E(u_1,u_2....u_n \mid H_h) = \sum_U \left[\prod_{j=1}^{n} u_j \times P(u_j \mid H_h)\right] = P(u_j = 1, \forall j \mid H_h) \quad (31)$$

Equation (31) needs the evaluation of multivariate normal integral defined by

$$F([t]) = \int_{t_1}^{\infty}\int_{t_2}^{\infty}......\int_{t_n}^{\infty} f_{X_1,X_2,....X_n}(x_1x_2....x_n \mid H_h) dx_1 dx_2...dx_n \quad (32)$$

While closed form solutions exist for specific conditions imposed on the mean, covariance structure [17] and the threshold vector [t], the integral can only be calculated numerically. For most of the detection problems, these conditions on mean and covariance matrix do not hold true. If conditions are applied over the threshold vector as in [15], the purpose of optimization is defeated. Hence, we resort to numerical integration of the above integral.

For numerical integration of the above integral, we use the method proposed by Alan Genz [16]. For a positive semi-definite covariance matrix, a series of transformations lead to a simplified algorithm for evaluation of this integral. More details can be found in [16]. Genz developed the method for zero mean, normal variables. After a simple linear transformation of our problem, it can be handled by the same integration procedure.

We solve the problem for the 2 sensors problem in this paper. Similarly the problem can be solved for N sensors applying the above procedure and (20). The means, standard deviation and the correlation coefficients to generate the covariance matrix for the two-sensor case are given in the appendix section of this paper. PSO is used to generate the optimal fusion rule and the optimal thresholds.

A. *Asymmetric Correlation*

Table 2 presents results for the three strategies for asymmetric correlation. The correlations under both hypotheses are given in Table 4. The results in Table 2 are for 2-sensor suite. The correlation under H_0 is 0.76 and under H_1 is 0.0185. The results achieved for a few a priors are presented. Equal costs are assumed for both the errors. For the sake of brevity, the results are presented for only a few samples of the prior probability.

The Bayesian cost values for the three different strategies are given. First, CVR is applied assuming that sensors are not correlated. However, the fused error probabilities consider the underlying correlation. Then MKR for correlation is applied. This incorporates the additional processing in (27). In both strategies the thresholds of sensors are first found using LRT. Finally, PSO is applied to achieve the optimal thresholds.

For Gaussian examples, the optimal fusion rule (MKR) for correlated sensors achieves better performance than the standard CVR. This is expected as CVR assumes independence, erroneously. Additional benefits are achieved by using PSO based optimization, which also does not assume independence, and uses (24) and (25) to evaluate its fitness. The performance benefits are nearly 6% over the traditional MKR rule, which is the traditional decision fusion approach for correlation. Note that the performance benefits will vary based on the underlying correlation. However, in this paper a very high correlation of 0.76 is assumed under H_1. For higher H_0 a priori, the performance benefits significantly decline due to higher correlation between both the sensors under H_0. However, PSO still achieves a performance benefit of 4.7% over the MKR.

TABLE 2 RESULTS ON GAUSSIAN EXAMPLES

P0	CVR	MKR	PSO Based	% Impr. 2 vs. 1	% Impr. 3 vs. 2
0.2	0.01817	0.0040	0.00372	77.72%	8.019%
0.4	0.0308	0.0062	0.00581	79.70%	7.198%
0.5	0.03583	0.007	0.00652	80.46%	6.838%
0.6	0.03974	0.0074	0.00699	81.17%	6.480%
0.8	0.2267	0.0073	0.00697	96.74%	5.53%
0.9	0.1229	0.0063	0.00603	94.84	4.713%

B. *Symmetric Correlation*

Figure 2 gives the results for a 2-sensor case. Equal costs are assumed for both the errors. Equal a priori probabilities are assumed. The Bayesian cost is then called probability of error. The PSO based optimization scheme is compared with two traditional fusion strategies. In both the traditional fusion strategies, the local decision rule is set using LRT. In the first strategy, CVR is used to arrive at the optimal fusion rule. In the second strategy, MKR is used. MKR performs significantly better than CVR as expected. The PSO based design performs significantly better across a wide range of correlation values. However at higher values of correlation, PSO based optimization performs only slightly better than MKR. The error value increases as correlation increases.

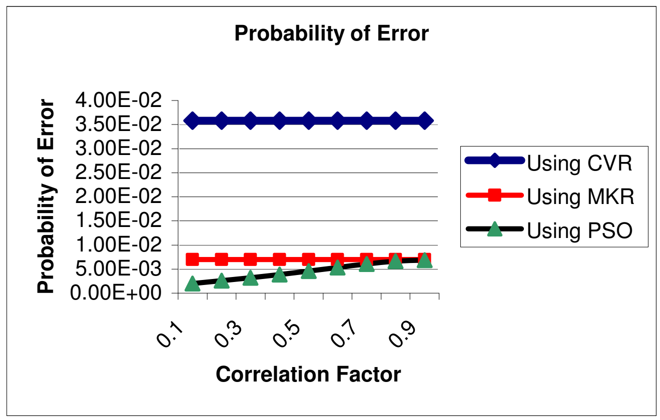

Figure 2: Probability of error for a 2 Sensor Network Using Different Strategies

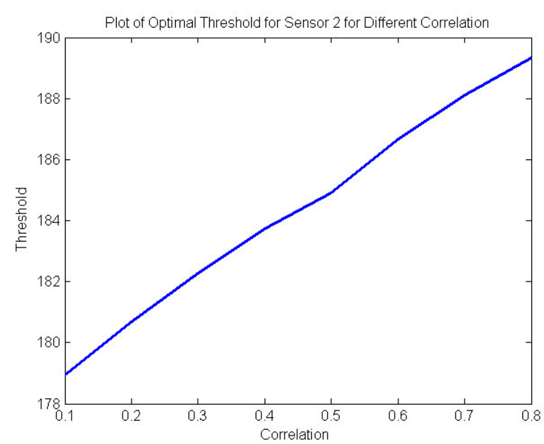

Figure 3: Plot of threshold for Sensor 2 as determined by PSO for varied degrees of correlation

Figure 3 and 4 show the variation of the threshold as set by the PSO algorithm to achieve optimum performance. The optimal fusion rule across different correlation was found to be AND rule.

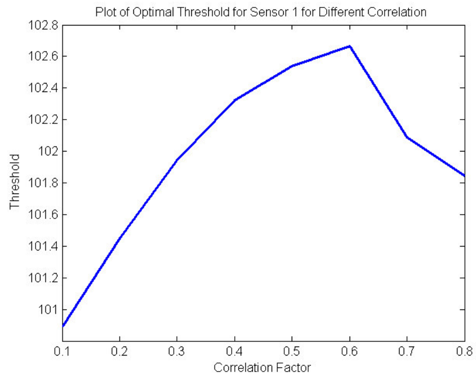

Figure 4: Plot of threshold for Sensor 1 as determined by PSO for varied degrees of correlation

V. CONCLUSIONS

In this paper we presented a particle swarm optimization based algorithm to design distributed detection systems with correlated heterogeneous sensors. The estimation of the error probabilities using Bahadur-Lazarfeld expansion is adopted. The evaluation of multivariate integral is done numerically.

We compared the algorithm with traditional decision fusion strategy, i.e., MKR. We also compared the two strategies with the CVR, which assumes independence in identifying the optimal fusion rule. The MKR based optimal fusion performs better than CVR. PSO based optimization achieved significantly better performance as a function of correlation between the sensors and a priori of hypotheses. The testing of performance of PSO against other traditional optimization strategies as in PBPO is part of the future work. The results presented in this paper demonstrate PSO as a strong computational paradigm to optimize the distributed detection network in presence of correlation.

REFERENCES

[1] Kalyan Veeramachaneni, Lisa Osadciw, "Dynamic Sensor Management Using Multi Objective Particle Swarm Optimizer", Multisensor, Multisource Information fusion: Architectures, Algorithms, and Applications 2004, edited by B. V. Dasarathy, Proceedings of SPIE Vol. 5434.
[2] Josef Kittler, Mohamad Hanef, Robert P. W. Duin, Jiri Matas, " On Combining Classifiers", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 3, March 1998.
[3] James Kennedy, Russell Eberhart and Shi, Y.H., Swarm Intelligence , Morgan Kaufmann Publishers, 2001.
[4] Randy Cogill, Sanjay Lall, "Decentralized Stochastic Decision Problems and Polynomial Optimization", Proceedings of the 2004 Annual Allerton Conference on Communication, Control, and Computing.
[5] Pramod K Varshney, Distributed Detection and Data Fusion, Springer-Verlag New York, Inc., 1996.
[6] J. N. Tsitsiklis and M. Athans, " On Complexity of decentralized decision making and detection problems", IEEE Transactions on Automatic Control, vol. AC-30, pp.. 440-446, 1985.
[7] Kalyan Veeramachaneni, "An Evolutionary Algorithm Based Dynamic Thresholding for Multimodal Biometrics" Masters Thesis, School of Electrical and Computer Engineering, Syracuse University, 2003.
[8] Arun Ross, Anil Jain, 'Information Fusion in Biometrics', Pattern Recognition Letters 24, 2003, pp 2115 –2125.
[9] Yan, W. and Goebel, K., "Designing classifier ensembles with constrained performance requirements" in Multisensor, Multisource Information fusion: Architectures, Algorithms, and Applications 2004, edited by B. V. Dasarathy, Proceedings of SPIE Vol. 5434, pp59-68.
[10] Therrien, C.W. (1989), Decision estimation and classification: An introduction to pattern recognition and related topics, John Wiley and Sons, NY, 1989.
[11] Kalyan Veeramachaneni, Thanmaya Peram, Lisa Ann Osadciw, Chilukuri Mohan, "Optimization Using Particle Swarms with Near Neighbor Interactions", Lecture Notes in Computer Science, Springer Verlag, Vol. 2723, 2003.
[12] Optiz, D.W. (1999), "Feature selection for ensembles", Proceedings of 16th International Conference on Artificial Intelligence, pp 379-384.
[13] Chair Z., P. K. Varshney, "Optimal Data Fusion in Multiple Sensor Detection Systems," IEEE Trans. on Aerospace and Elect. Systems, Vol. AES-22, No. 1, pp. 98-101, Jan. 1986.
[14] Tang, Z. -B., K. R. Pattipati, and D. L. Kleinman, "An Algorithm for Determining the Decision Thresholds in a Distributed Detection Problem," IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-21, pp. 231-237, Jan./Feb. 1991.
[15] Kam., M., Q. Zhu., and W. S. Gray, "Optimal Data Fusion of Correlated Local Decisions in Multiple Sensor Detection Systems," IEEE Transactions on Aerospace and Elect. Syst., Vol. 28, pp. 916-920, July 1992.
[16] Genz, A. (1992) "Numerical Computation of Multivariate Normal Probabilities, " J. Comp. Graph. Stat, 1, 141-149 1992.
[17] Peter Willet, Peter F. Swaszek, Rick S. Blum, « The Good, Bad, and Ugly : Distributed Detection of Known Signal in Dependent Gaussian Noise », IEEE Transactions on Signal Processing, Vol. 48, No. 12, December 2000.
[18] Shanti S. Gupta, « Probability Integrals of Multivariate Normal and Multivariate t, » The Annals of Mathematical Statistics, 34, 792-828.
[19] Kalyan Veeramachaneni, Lisa Osadciw, Pramod Varshney, « Adaptive Multimodal Biometric Management Algorithm., » IEEE Transactions on Systems Man and Cybernetics, Vol. 35, August, 2005.
[20] Tsitsiklis, J. N., and M. Athans, « On Complexity of Decentralized Decision Making and Detection Problems, » IEEE Trans. On automatix Control, Vol. AC-30, pp. 440 –446, May 1985.
[21] Richard O Duda, Pattern Classification and Scene Analysis, Wiley, 1973, New York.
[22] Eugene Charniak, « Bayesian Networks without Tears », AI Magazine, Volume 12, Issue 4, Pages 50-63, 1991.
[23] Caren Marzban, "A Comment on the ROC Curve and the Area Under it as Performance Measures", Weather and Forecasting, Vol. 19, No. 6, 1106-1114.
[24] Tang, Zhuang-Bo, « Optimization of Detection Networks », Ph. D Dissertation, University of Connecticut, 1990.
[25] Kalyan Veeramachaneni, Lisa Osadciw, « Dynamic Sensor Management Using Multi Objective Particle Swarm Optimizer », SPIE Defence and Security Symposium, Orlando, Florida, April 2004.

APPENDIX

A. *Sensor Suite used in this paper*

The mean and standard deviations of the sensors/classifiers used in this paper are given here. The covariance matrix is constructed based on the correlation coefficients assumed which are also given. Table III gives the means and the standard deviations under hypothesis H_0 and H_I

TABLE 3 MEAN AND STANDARD DEVIATIONS OF SENSORS USED IN THIS PAPER

Hypothesis/ Paremeter	H_0/μ_0	H_0/σ_0	H_1/μ_1	H_1/σ_1
Sensor 1	47.375	43.864	144.514	12.843
Sensor 2	67.755	52.633	251.209	23.008
Sensor 3	50.417	26.206	167.464	10.189

The correlation coefficients under each hypothesis are given for all the possible pairs of sensors and are given as ρ_h^{12}, ρ_h^{13}, ρ_h^{23}. The covariance matrix is given by,

$$\Sigma^h = \begin{bmatrix} (\sigma_h^1)^2 & (\sigma_h^1)(\sigma_h^2)\rho_h^{12} & (\sigma_h^1)(\sigma_h^3)\rho_h^{13} \\ (\sigma_h^2)(\sigma_h^1)\rho_h^{12} & (\sigma_h^2)^2 & (\sigma_h^2)(\sigma_h^3)\rho_h^{23} \\ (\sigma_h^3)(\sigma_h^1)\rho_h^{13} & (\sigma_h^3)(\sigma_h^2)\rho_h^{23} & (\sigma_h^3)^2 \end{bmatrix} \quad (33)$$

The correlation coefficients used in this paper under both the hypothesis are given in Table IV.

TABLE 4: CORRELATION COEFFICIENTS UNDER BOTH THE HYPOTHESIS FOR ASYMMETRIC CASE

Hypothesis	$h=0$	$h=1$
ρ_h^{12}	0.76036	0.0185
ρ_h^{13}	0.52982	0.8214
ρ_h^{23}	0.60286	0.4102