Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Multicriteria Decision Making (MCDM 2007)

Improving Classifier Fusion Using Particle Swarm Optimization

Kalyan Veeramachaneni
Dept. of EECS
Syracuse University
Syracuse, NY, U. S. A
kveerama@syr.edu

Weizhong Yan
GE Global Research Center
Niskayuna, NY, U. S. A
yan@crd.ge.com

Kai Goebel
NASA Ames Research Center
Moffett Field, CA, U. S. A
goebel@email.arc.nasa.com

Lisa Osadciw
Dept of EECS
Syracuse University
Syracuse, NY, U. S. A
laosadci@syr.edu

Abstract - Both experimental and theoretical studies have proved that classifier fusion can be effective in improving overall classification performance. Classifier fusion can be performed on either score (raw classifier outputs) level or decision level. While tremendous research interests have been on score-level fusion, research work for decision-level fusion is sparse. This paper presents a particle swarm optimization based decision-level fusion scheme for optimizing classifier fusion performance. Multiple classifiers are fused at the decision level, and the particle swarm optimization algorithm finds optimal decision threshold for each classifier and the optimal fusion rule. Specifically, we present an optimal fusion strategy for fusing multiple classifiers to satisfy accuracy performance requirements, as applied to a real-world classification problem. The optimal decision fusion technique is found to perform significantly better than the conventional classifier fusion methods, i.e., traditional decision level fusion and averaged sum rule.

Keywords: Decision level fusion, multiple classifiers fusion, particle swarm optimization.

1 Introduction

Classifier design is a task of developing a classification system that optimizes performance with respect to requirements. Traditionally, design of classification systems is to empirically choose a single classifier through experimental evaluation of a number of different ones. The parameters of the selected classifiers are then optimized so that the specified performance is met. Single classifier systems have limited performance. For certain real-world classification problems, this single classifier design approach may fail to meet the desired performance even after all parameters/architectures of the classifier have been fully optimized. In these cases, classifier fusion , one of the most significant advances in pattern classification in recent years, proves to be effective and efficient [2]. By taking advantage of complementary information provided by the constituent classifiers, classifier fusion offers improved performance, (i.e., they are more accurate than the best individual classifier).

Classifier fusion can be done at two different levels, namely, score level and decision level. In score level fusion, raw outputs (scores or confidence levels) of the individual classifiers are combined in a certain way to reach a global decision. The combination can be performed either simply using the sum rule or averaged sum rule, or more sophisticatedly, using another classifier. Decision level fusion, on the other hand, arrives at the final classification decision by combining the decisions of individual classifiers. Majority voting rule and Chair-Varshney [13] optimal fusion rule are two examples of decision-level fusion schemes. Chair- Varshney [13] optimal decision fusion rule is achieved using the individual classifier performance indices. The optimal fusion rule can be majority-voting rule but is not limited to it.

There have been very few studies in optimizing fusion system performance. At each level of fusion, alternate strategies of fusion exist which can be explored to achieve the optimal performance across different costs of miss classification. In this paper, decision level fusion is chosen and optimization of decision level fusion to achieve the required performance is presented.

In decision level fusion, shown in Figure 1, each classifier under binary hypothesis gives its decision regarding the class of the observation. The decisions from multiple classifiers are fused at the fusion processor. The fusion processor uses a fusion rule to fuse the multiple decisions and produces a decision.

The most important problem for achieving optimum performance at decision level fusion becomes the optimal setting of individual decision thresholds. There are $2^{2^{\wedge}N}$ possible fusion rules for a binary hypothesis and N classifier system. Most of the classifier fusion work done in the past neglects all the possible rules that can be explored at decision level. Also the decision threshold for individual classifier is optimally set to minimize the error of the classifier [2]. This is done even before the fusion is carried out. This typically entails selection of an operating point from the Receiver Operating Characteristic (ROC) curve for the individual classifier, which will minimize the error for given costs of misclassification. Once the decision thresholds for individual classifiers are set, majority voting rule or the chair-varshney optimal fusion rule is used as the fusion rule. This method, however, does not guarantee optimum performance after fusion. Performance can be defined under Neymen Pearson criterion or Bayesian criterion.

In this paper the optimal thresholds and the corresponding fusion rule which results in optimum accuracy after fusion are found simultaneously. A particle swarm optimization based method is applied to achieve optimum decision thresholds for individual classifiers and the optimal fusion rule.

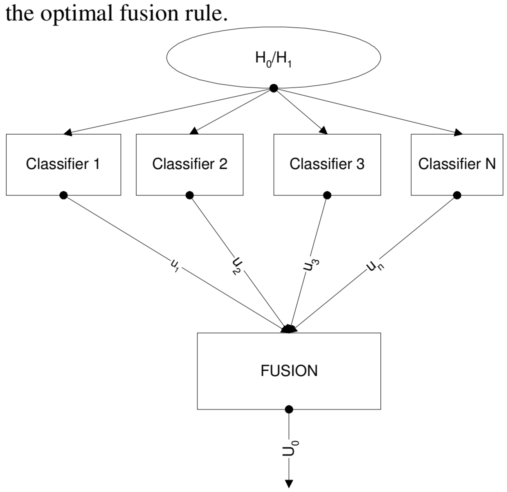

Figure 1. Decision Level Fusion for Multiple Classifier Systems

The rest of paper is organized as follows. In section 2, Decision level fusion under binary classification is detailed. Alternate decision fusion strategies are detailed for a system involving two classifiers. The resulting intractable problem is detailed. Particle Swarm Optimization is discussed in Section 3. Formulation of the particle for this problem, Bayesian cost function, the settings of the PSO used are also detailed in this section. Section 4 presents a case study for a real-world application problem. Results and discussion are given in Section 5. Comparisons of the results with the traditional decision level fusion and simple averaged sum rule fusion are presented in this section. Section 6 concludes the paper.

2 Decision Level Fusion

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

H_0 : Absence of the class

H_1 : Presence of the class

The two types of errors commonly known as probability of false positives and probability of false negatives (miss) are

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

$$P_{FN} = P(U_0 = 0 / H_1) \quad (2)$$

Also,

$$P_{TP} = 1 - P_{FN} \quad (3)$$

Where U_0 is the decision of the fusion processor, which takes in the decisions from the local classifiers and fuses them using the fusion rule. In the following sections, these error probabilities for fusion involving 2 classifiers and fusion involving N classifiers are derived.

2.1 Decision Fusion of Two Classifiers

For the fusion of 2 classifiers, there are 16 potential decision fusion rules possible as depicted in Table 1.

Table1: Possible Fusion Rules for 2 Classifiers

u_1	u_2	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1

u_1	u_2	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1

Binary strings represent the rules, since the decisions are binary (i.e. accept or reject the hypothesis). It has been shown that an optimum fusion rule must be monotonic so all other rules can be ignored [5]. The most commonly used rules are f_2 ("AND" rule) and f_8 ("OR" rule). The "AND" rule makes it more difficult for accepting hypothesis as both the classifiers should accept the hypothesis. The probability that both the classifiers result in an error is low and hence this minimizes the overall false positive probability. The "OR" rule eases acceptance of hypothesis since hypothesis is accepted as long as one of the classifiers says so. This reduces the probability of false negatives. The "NAND" rule or f_9 is rarely of interest since it typically performs poorly. This rule accepts the hypothesis if both the classifiers reject and reject otherwise. In most cases, this rule does not logically make sense and leads to poor performance. The f_1 rule simply rejects the hypothesis irrespective of individual classifiers decisions. Similarly, the f_{16} rule accepts the hypothesis irrespective of the individual classifiers decisions. These two rules are rarely of interest, however constitute the two ends of the receiver-operating characteristic curve. The system can completely ignore one classifier by using the f_4 (classifier 1) or f_6 (classifier 2) rules. These six rules constitute the monotonic set for two classifiers fusion. Number of monotonic rules increase as the number of classifiers increase. For example, for three classifiers there are 20 monotonic fusion rules. For more information on monotonic rules the reader is referred to [5].

2.2 The AND / OR Dichotomy

The "AND" and "OR" rules for 2 classifier fusion constitute a very important and dichotomous set. As the number of classifiers increase the fusion rules are constructed from these two operators. The third operator in a binary logic is the "NOT" operator which when introduced produces non-monotonic rules and lead to higher errors. In this section the dichotomy of "AND"/ "OR" rules is detailed. Independence of observations is assumed in this section.

The probability of false positives for 2 classifiers with the "AND" rule, becomes

$$P_{FP} = {P_{FP}}^1 \times {P_{FP}}^2 \quad (4)$$

and

$$P_{FN} = P_{FN}{}^{1} + P_{FN}{}^{2} - P_{FN}{}^{1} \times P_{FN}{}^{2} \quad (5)$$

where $P_{FP}{}^{i}$ is the probability of false positive of the i[th] classifier and $P_{FN}{}^{i}$ is the probability of false negatives for the i[th] classifier.

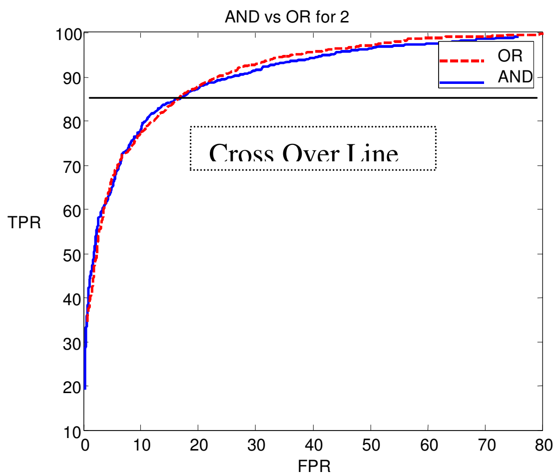

Figure 2: AND vs. OR rule for Two Classifiers.

It is apparent that if the error probabilities are improved or smaller for the new classifier this decreases global false positives while increasing the global false negatives. Thus, global performance improves if the decrease in false positives is greater than the increase in false negatives for equal costs assigned for both the errors. The degradation in false negatives can be significant if the two classifiers are not comparable in terms of accuracy. The OR rule reverses this effect giving

$$P_{FN} = P_{FN}{}^{1} \times P_{FN}{}^{2} \quad (6)$$

and

$$P_{FP} = P_{FP}{}^{1} + P_{FP}{}^{2} - P_{FP}{}^{1} \times P_{FP}{}^{2} \quad (7)$$

In Figure 2, the ROCs for these two rules, as applied to outputs of classifiers designed to detect pipeline defect is shown. Figure 2; support a comparison of the rules' performance. The "AND" rule achieves a much lower false positive rate (in percentages) at the expense of increasing false negative rate reflected in the decrease of True Positive rate on the y-axis. The "OR" rule however, works on improving the true positive rate at the expense of higher false positive rate. As can be seen that above the cross over line, shown in Figure 2, ROC pertaining to the "OR" rule dominates, while below the cross over line the "AND" rule ROC dominates. The crossover point is related to single classifier accuracy, implying that fusion is not required if we intend to operate in this region on ROC. On these points it would be beneficial to switch to a more accurate classifier. It should be noted here that identical thresholds for both the classifiers were set before fusing them with "AND" and "OR" rules. In case of 2 classifier fusion simultaneous improvements in FPR and TPR are impossible to achieve. One must choose to improve one of the two performance parameters, either the false positive rate or true positive rate. Global performance optimization can be done by searching for optimal thresholds for each classifier. Such an optimization will result in the maximizing the improvement of one error probability while minimally affecting the other error probability, resulting in a global minimum of a weighted cost function described later in Section 2.4, equation (18). Fusion of two classifiers is still beneficial than the single classifier system as one can argue that for the same false positive probability one can achieve higher true positive probability. However, the gain is always constrained by the less accurate classifier.

It should be easy to see how the increase in the false negative probability due to "AND" rule applied to 2 classifiers can be compensated by introducing a third classifier with an "OR" rule between it and the other two, i.e., (1 AND 2) OR 3. Careful selection of the individual thresholds can result in a significant reduction of the false negative probability while not affecting the false positive probability and hence can result in reduction of both false positive and false negative probabilities simultaneously when three classifiers are involved in fusion. However, the third classifier should be of comparable accuracy and should be diverse. Setting each classifier thresholds and the fusion rule to achieve this improvement is not trivial and becomes more and more complex as the number of classifiers involved in fusion is increased. This problem is a well-known intractable problem [6]. In this paper a particle swarm optimization based approach is used to achieve the optimal thresholds and the fusion rule.

2.3 Decision Level Fusion for N Classifier Systems

In this section the calculation of the error probabilities for a N classifier system given the individual thresholds and fusion rule is derived. P_{FP}, P_{FN} of the fused system is calculated from the fusion rule and individual classifier $P_{FP}{}^{i}$ and $P_{FN}{}^{i}$.

Table 2: Fusion Rule Formation for Two Classifiers

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 classifiers the fusion rule consists of 4 bits, as represented in Table 2. In Table 2, u_1 is the first classifier decision, and u_2 is the second classifier decision. The fusion rule is of length *l* bits where

$$l = \log_2 p \quad (8)$$

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

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

$$P_{FP} = \sum_{i=0}^{p-1} d_i \times \left\{ \prod_{j=1}^{N} E_j \right\} \quad (9)$$

where

$$E_j = \begin{cases} 1 - P_{FP}{}^{j}, & (u_j = 0) \\ P_{FP}{}^{j}, & (u_j = 1) \end{cases} \quad (10)$$

and

$$P_{FN} = \sum_{i=0}^{p-1} (1 - d_i) \times \left\{ \prod_{j=1}^{N} E_j \right\} \quad (11)$$

where

$$E_j = \begin{cases} P_{FN}{}^{j}, & (u_j = 0) \\ 1 - P_{FN}{}^{j}, & (u_j = 1) \end{cases} \quad (12)$$

$P_{FP}{}^{i}$ is the probability of false positives of the i^th^ classifier and is given by

$$P_{FP}{}^{i} = P(u_i = 1 / H_0) \quad (13)$$

$$P_{FP}{}^{i} = \int_{\lambda_i}^{\infty} P(z_i / H_0) \quad (14)$$

and

$$P_{FN}{}^{i} = P(u_i = 0 / H_1) \quad (15)$$

$$P_{FN}{}^{i} = \int_{-\infty}^{\lambda_i} P(z_i / H_1) \quad (16)$$

where λ_i is the threshold at which the classifier is set. z_i is the raw output of the classifier conditioned over H_1 or H_0. Hence, P_{FP} is a function of thresholds (local decision rules) of all the classifiers and the optimal fusion rule.

$$P_{FP} = g(\lambda_1, \lambda_2 ....\lambda_n, f) \quad (17)$$

Where λ_i is the decision threshold for the i^{th} classifier and '*f*' is the fusion rule. Similarly P_{FN} is also the function of local decision rules and fusion rule.

2.4 Bayesian Error in Decision Fusion

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

$$E = C_{FP} \times P_{FP} + C_{FN} \times P_{FN} \quad (18)$$

Where

$$C_{FP} = 2 - C_{FN} \quad (19)$$

Eq (18) 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 cost of true positive and true negatives is set to zero in Eq (18). The probability of false positives and probability of false negatives in (18) are the global false positives and false negative probabilities derived in (9), (11). In the previous section the calculation of these two for a 2 classifier system were discussed.

The aim of the optimization algorithm is to come up with the local decision thresholds such that (18) is minimized. The problem of optimal setting of local decision rules and the optimal fusion rule minimizing the total probability of error has been extensively studied. It has been shown in [5, 6] that finding optimal decision rules for each individual classifiers as well as corresponding optimal fusion rule is intractable. Person-by-Person Optimization (PBPO) has been traditionally used and suggested to solve this problem [14]. In PBPO approach, optimum decision threshold is found for a classifier while keeping others fixed at previously attained values. This process is done for each of the classifier and is repeated iteratively until convergence or in other words equilibrium is achieved. Equilibrium is attained when, for a set of such decision thresholds and fusion rule, no improvement is obtained by adjusting the decision threshold for any given classifier while leaving others fixed [4]. These are called person-by-person optimal solutions and hence the name PBPO. For any problem there maybe multiple equilibrium states. Hence, multiple initializations for the problem at the starting of the algorithm are required to increase the probability of reaching global optima [14]. With large number of classifiers the number of multiple initializations required also increases. In this paper particle swarm optimization algorithm is used to achieve optimal local thresholds and the optimal fusion rule.

2.5 Traditional Decision Level Fusion

Traditional decision level fusion strategies apply maximum likelihood ratio test as in (20) to derive the optimal threshold for each classifiers measurement. The threshold is applied to arrive at a hard decision.

$$\log \frac{P(z_j / H_1)}{P(z_j / H_0)} \underset{H_1}{\overset{H_0}{\gtrless}} \log \frac{C_{FN}}{C_{FP}} \quad (20)$$

The hard decisions can be combined using a majority voting rule or chair-varshney rule as in

$$\sum_{j=1}^{N} \left[u_j \log \left\{ \frac{1 - P_{FN_j}}{P_{FP_j}} \right\} + (1 - u_j) \log \left\{ \frac{P_{FN_j}}{(1 - P_{FP_j})} \right\} \right] \underset{H_1}{\overset{H_0}{\lessgtr}} \log \left(\frac{C_{FP}}{2 - C_{FP}} \right) \quad (21)$$

where u_i is the decision of the i^th^ classifier. In this paper, comparisons are done between the optimized decision level fusion and the traditional decision level fusion. Note that chair varshney (CV) rule assumes independence of classifiers. An equivalent of CV rule is presented for correlated classifiers in [15].

3 Particle Swarm Optimization

The particle swarm optimization 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 fairly simple computations and sharing of information within the algorithm as it derives its internal communications from the social behavior of individuals. The individuals, called particles henceforth, 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 $X_i = (x_{i1}, x_{i2}, ..., x_{iD})$, 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 $P_i = (p_{i1}, p_{i2}, ..., p_{iD})$ and a velocity along each dimension as $V_i = (v_{i1}, v_{i2}, ..., v_{iD})$. 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].

$$V_{id}^{(t+1)} = \begin{array}{l} \omega \times V_{id}^{(t)} + U[0,1] \times \psi_1 \times (p_{id}^{(t)} - x_{id}^{(t)}) + \\ U[0,1] \times \psi_2 \times (p_{gd}^{(t)} - x_{id}^{(t)}) \end{array} . \quad (22)$$

$$X_{id}^{(t+1)} = X_{id}^{(t)} + V_{id}^{(t+1)} \quad (23)$$

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].

3.1 PSO for Decision Fusion

Each particle in this problem has 'N+1' dimensions, where N is the number of classifiers in the classifier ensemble. Each of the N dimensions is a threshold at which a particular classifier is set. The 'N+1' th dimension is the fusion rule, which determines how all the decisions from the classifiers are fused. Hence the representation of each particle is, $X_i = (\lambda_{i1}, \lambda_{i2} ....\lambda_{in}, f_i)$. The classifier 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' classifiers, with a decimal value varying from $0 \le dec(f) \le p-1$. For binary search spaces, the binary decision model as described in [3] is being used. Hence the PSO used in the paper is a hybrid of both binary PSO and continuous PSO; binary for evolving the fusion rule, continuous for thresholds. The two objectives for this problem are given by (9), (11). The goal is to minimize both the P_{FP} and P_{FN}. At each iteration, the particles representing the solution for the problem are evaluated for these objectives using the weighted cost function (18). The memory of the particle is updated if it finds better minima. The particles are moved in the search space based on equations (22) and (23) and these steps are iteratively repeated till convergence occurs or the requirements are fulfilled.

4 A Real-world Application

To demonstrate the effectiveness of particle swarm optimization based decision-level fusion scheme proposed in this paper, we apply the fusion scheme to a real-world fault detection problem. The real-world application concerned in this paper is an automated defect detection system for non-destructive inspection. The system designed is for determining the conditions (normal or defected) of pipelines based on ultrasonic images. For classifier design, 5600 examples representing different conditions of the target object are used. Of the 5600 examples, 2600 are for defected condition while 3000 are for normal condition. To enhance classification performance, 370 features were extracted from the raw ultrasonic images using various methods from different domains. The classification performance requirement is less than 55% false positive rate with the true positive rate of greater than or equal to 98%.

Ten (10) neural networks are used as the base classifiers for classifier fusion. By using 10 networks we followed the suggestion of Opitz and Maclin [12] who reported that ensembles with as few as 10 base classifiers are adequate to sufficiently reduce error. The networks are fully connected feed-forward type with a single hidden layer. To increase the diversity of those individual classifiers, which is important for obtaining improved fusion performance, each network uses different number of hidden neurons. Additionally, each network uses different training data that are obtained by randomly sampling, with replacement, from the original data set. Furthermore, each network uses different features that are selected through a GA-based feature selection process. See (Yan et al., 2004) [9] for details on GA-based feature selection and the design of the 10 neural networks.

The conditional density functions, conditioned over hypothesis

$$H_0 \quad : \; No\ Defect$$
$$H_1 \quad : \; Defect$$

are formed for the 10 classifiers. These conditional density functions can be fitted to some well-known distributions and the models can be used to perform the optimization using PSO. For example, if it fits well one can fit the data to a normal distribution and use error functions to optimize the fusion strategy. In this paper, we apply the PSO to the actual data since the data size is within reach of the computational power used for simulation in this paper.

5 Results and Discussion

We applied the PSO based optimization technique for decision level fusion of two, three, five, seven classifiers. The thresholds and the corresponding fusion rule were generated for the cost (C_{FP}) values ranging from 0 to 2 to generate a Receiver Operating Characteristic Curve. The ROC curves achieved are shown in Figures 3,4,5,6,7. Each point on the ROC curve of the decision level fusion corresponds to a different fusion configuration defined by the thresholds for the individual classifiers and the fusion rule. This is found by PSO, which minimizes (18) for a particular cost (C_{FP}). Comparisons are done with the traditional decision fusion and averaged sum rule. In averaged sum rule all the classifier outputs are averaged and a threshold is applied to arrive at the final decision. All the outputs from the classifiers are averaged and the resulting distribution is subjected to varying decision threshold to generate the ROC for the averaged sum rule.

5.1 Traditional Decision fusion vs. Optimized Decision fusion

The data is split into three parts. One part is used as training data to arrive at optimal thresholds and the fusion rule. The other two datasets are used to test the fusion strategy achieved using the training data.

The thresholds and fusion rule are achieved using PSO and the traditional decision fusion strategies as in (20) and (21). The results achieved when applied to the test data are presented for different costs of false positives.

Table 3 Traditional Decision Fusion vs. Optimized Decision Fusion for C_{FP}=1

Appr. / #Classifiers	Traditional Decision Fusion		Optimized Decision Fusion	
	Train FPR, TPR %	Test FPR, TPR %	Train FPR, TPR %	Test FPR, TPR %
Five	12.2, 83.51	13.95, 82.89	12.8, 86.4	13.55, 85.71
Seven	12.5, 83.65	13.15, 82.96	12.4, 86.67	14.9, 85.85

In table 3, 4, 5 results are presented for traditional decision fusion and optimized decision fusion using PSO. Table 5 gives the Bayesian error values for the two approaches for different costs as applied to the test data. The optimized decision level fusion performed better for all the three costs. Note that traditional decision fusion has extremely biased performance due to the two-step optimization procedure adopted.

5.2 Comparisons with Averaged Sum Rule

Figure 3, 5, 7 show the comparisons for the 2, 3, 5 classifier scenario. The decision level fusion with PSO produces a dominating ROC when compared to the averaged sum rule. In figures 4, 6 the ROC between 96% and 99% of TPR is zoomed for 2 classifier and 3 classifier system. The performance enhancement by using optimal decision fusion strategy as generated by the PSO is seen more clearly in these graphs.

Table 4 Traditional Decision Fusion vs. Optimized Decision Fusion for C_{FA}=0.1

Appr. / # Classifiers	Traditional Decision Fusion		Optimized Decision Fusion	
	Train FPR, TPR	Test FPR, TPR	Train FPR, TPR	Test FPR, TPR
Five	88.2, 100	90.1, 99.93	57.3, 99.17	61.05, 98.55
Seven	91.8, 100	92.75, 99.93	61.2, 99.58	64.1, 98.49

Table 5 Traditional Decision Fusion vs. Optimized Decision Fusion for C_{FP}=1.9

Appr. /# Classifiers	Traditional Decision Fusion		Optimized Decision Fusion	
	Train FPR, TPR %	Test FPR, TPR %	Train FPR, TPR %	Test FPR, TPR %
Five	0.2, 19.91	0.1, 20.26	0.5, 41.62	0.85, 40.93
Seven	0.2, 16.34	0.1, 15.38	0.2, 37.22	0.5, 36.81

Table 5 Bayesian Errors for Traditional Decision Fusion vs. Optimized Decision Fusion (Testing)

C_{FA}	Traditional Decision Fusion		Optimized Decision Fusion	
	Five	Seven	Five	Seven
1	0.3106	0.3019	**0.2784**	**0.2905**
1.9	0.8164	0.08652	**0.07522**	**0.07269**
0.1	0.09142	0.09407	**0.0886**	**0.0927**

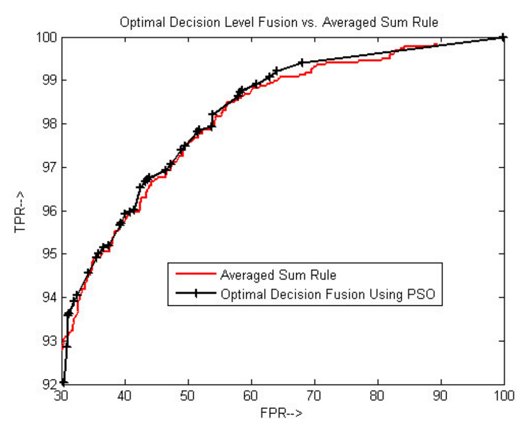

Figure 3: Optimal Decision Level Fusion vs. Averaged Sum rule for Two Classifiers

The results shown in the figures are for a TPR > 93%. It should be noted that it is very difficult to achieve any performance improvements in this region of ROC. Also

the output scores of classifiers are highly correlated which makes performance enhancements very hard to achieve.
Also the results presented in this section are derived using the entire data as training data. In future work, testing/cross validation will be done to establish the performance advantages of the optimized decision level fusion over averaged sum rule.

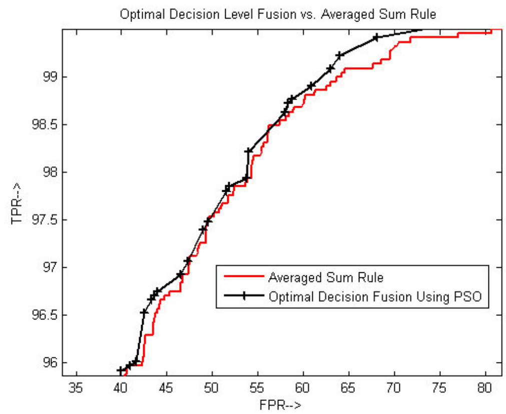

Figure 4: Optimal Decision Level Fusion vs. Averaged Sum rule for Two Classifiers

5.3 Constrained Performance Requirements

Constrained performance requirements can be specified in different forms, depending on which rate (FPR or FNR or both) needs to be set to a predetermined level. In this paper, we only consider the constrained performance requirement as being to minimize FPR while maintaining TPR (= 1-FNR) to a predetermined value. Let the predetermined large number be α, the constrained performance can then be expressed as

$$\min(FPR) \text{ Subject to } TPR \geq \alpha \qquad (22)$$

Alternatively, the constrained performance requirement is to minimize the quantity [10]

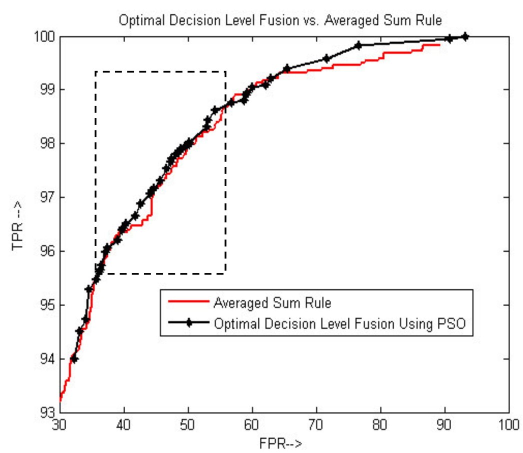

Figure 5: Optimal Decision Level Fusion vs. Averaged Sum Rule for 3 Classifiers.

$$\chi = FPR + \lambda \cdot (TPR - \alpha) \qquad (23)$$

where λ is a Lagrange multiplier.

Mathematically, the constrained performance requirement is a constrained optimization problem that is somewhat difficult to solve.

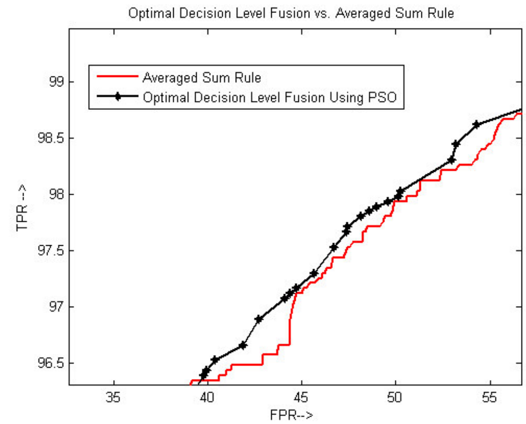

Figure 6: Optimal Decision Level Fusion vs. Averaged Sum Rule for 3 Classifiers.

In Table 6 results obtained for the constrained performance requirements by both the schemes is presented. The requirement on True Positive Rate was set to 98 % and the objective of fusion was to achieve minimum possible false positive rate. The point from the ROC of decision level fusion, which resulted in TPR > 98% was chosen to compare with the same from the averaged sum rule. Alternatively, one can modify the PSO cost function to reflect the constrained performance requirements and search for optimal fusion configuration. Such a point may or may not be in the set fusion strategies we evolved for different costs of false positives in the Bayesian error cost function (18). However, in this paper we ran PSO for costs of 1.6 to 2.0 in discrete steps of 0.1 and chose a strategy, which satisfied the performance requirements. The results achieved below show the performance enhancement using the PSO for the decision level fusion over the simple averaged sum rule.

Table 6. Meeting performance Requirements

Fusion Strategy	***TPR(>98 %)***	***FPR***
2 Classifiers Averaged Sum rule	*98.17*	*54.67*
2 Classifiers Decision Level fusion	***98.2143***	***53.9667***
3 Classifiers Averaged Sum rule	*98.03*	*51.17*
3 Classifiers Decision Level fusion	***98.0311***	***50.2333***

As can be seen from the above table decision level fusion with a PSO based optimization technique achieved better results in both the cases. When 2 classifier fusion is employed the decision level fusion achieved a perfectly dominating point i.e., TPR (DLF-PSO) > TPR(ASR) and FPR(DLF-PSO) < FPR (ASR). We achieved a significant performance improvement of 0.7 % in FPR while also improving the TPR by 0.04 %.
Also when three classifiers are employed the achievements have improved when compared to the 2-classifier system. An improvement of 0.94% was achieved in FPR while also improving the TPR by 0.0011 %. The aim here was to achieve a lower FPR for a TPR above 98%. Hence to summarize the improvements have been 0.7% for 2-

classifier system and 0.94 % for a 3-classifier system which are significant improvements for this application.

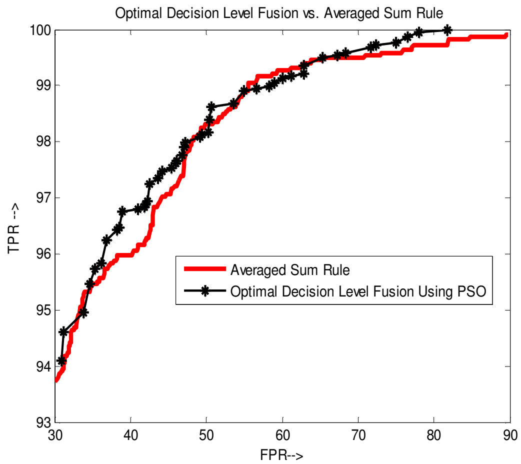

Figure 7: Optimal Decision Level Fusion vs. Averaged Sum Rule for 5 Classifiers.

6 Conclusions

The results presented in this paper show the performance advantage of using a decision level fusion scheme. The optimized decision level fusion is found to be performing better than the traditional decision level fusion. Often decision level fusion techniques are considered as suboptimal when compared to other fusion approaches. Such suboptimal performance is due to the two-layered optimization approach adopted by researchers while using decision fusion.

Also results presented show a promise of decision level fusion as compared to averaged sum rule. Also it is interesting to find that the combination of highly contrasting fusion rules as "AND" and "OR" result in better performance than the averaged sum rule. Particle swarm optimization problem is used to achieve the optimal decision fusion strategy, which is an intractable problem.

A similar fusion scheme can be devised at score level using the min and max rules. In future work a fusion scheme will be devised at the score level using a combination of min, max rules which are analogous to "AND" and "OR" rules.

Cross validation will be performed to establish the performance advantages due to optimization of decision fusion using PSO. The algorithm will also be applied to classifiers involving multiple modalities.

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.