Proc. of IEEE Computer Society Workshop on Biometrics at the Computer Vision and Pattern Recogniton (CVPR) conference, (Anchorage, USA), June 2008.

Decision-level Fusion Strategies for Correlated Biometric Classifiers

Kalyan Veeramachaneni[1], Lisa Osadciw[1], Arun Ross[2], Nisha Srinivas[1]
[1]Department of Electrical Engineering and Computer Science, Syracuse University.
[2]Department of Computer Science and Electrical Engineering, West Virginia University

Abstract

The focus of this paper is on designing decision-level fusion strategies for correlated biometric classifiers. In this regard, two different strategies are investigated. In the first strategy, an optimal fusion rule based on the likelihood ratio test (LRT) and the Chair Varshney Rule (CVR) is discussed for correlated hypothesis testing where the thresholds of the individual biometric classifiers are first fixed. In the second strategy, a particle swarm optimization (PSO) based procedure is proposed to simultaneously optimize the thresholds and the fusion rule. Results are presented on (a) a synthetic score data conforming to a multivariate normal distribution with different covariance matrices, and (b) the NIST BSSR dataset. We observe that the PSO-based decision fusion strategy performs well on correlated classifiers when compared with the LRT-based method as well as the average sum rule employing z-score normalization. This work highlights the importance of incorporating the correlation structure between classifiers when designing a biometric fusion system.

1. Introduction

Biometrics fusion, in the context of a verification system, may be posed as a binary hypothesis-testing problem involving multiple classifiers (i.e., matchers) [1]. In decision level fusion, each classifier operating under a binary hypothesis, applies a threshold on the match score and renders its decision regarding the presence (=1) or absence (=0) of a genuine individual. The decisions from multiple classifiers are then fused in order to generate the final decision.

Fusion at the decision-level is bandwidth efficient since only decisions, requiring a single bit, are transmitted to the fusion engine. Moreover, most commercial biometric classifiers grant access to decision-level information rather than score-level or feature-level information. Achieving optimality at the decision level, however, involves the selection of optimal decision thresholds *and* a fusion rule that minimize the classification error[1] [2]. There are 2^{2^N}possible fusion rules for an *N*-classifier system. Also, most decision fusion systems are designed under the assumption of independence between constituent classifiers. There have been efforts in recent literature addressing the problem of fusion in the presence of correlated biometric classifiers (e.g., see [4]). The procedures typically involve an exhaustive search for determining thresholds over a subset of monotonic fusion rules. The problem of jointly searching for the optimal thresholds and optimal fusion rule is NP hard [2].

In this paper, we present two formal designs for decision level fusion of correlated biometric classifiers. In the first strategy, decision thresholds are first estimated for each classifier prior to deducing the fusion rule. Thus, a two-step optimization procedure is adopted for this case. In the second strategy, an algorithm that can *jointly* optimize the thresholds and the fusion rule is designed. A particle swarm optimization (PSO) algorithm is proposed for this joint optimization. The rest of the paper is organized as follows. In section 2, decision level fusion for correlated biometric classifiers is discussed. Results observed on both a synthetic multivariate normal dataset and the NIST BSSR1 dataset are presented in Section 3. Section 4 concludes the paper.

2. Decision Level Fusion Strategies for Correlated Classifiers

The biometric verification problem may be posed as a binary hypothesis-testing problem with the match score(s) serving as observations. The two hypotheses are H_0: Score(s) indicates an imposter; H_1: Score(s) indicates a genuine user. In decision level fusion, each classifier applies a threshold on the match score and transmits the ensuing decision to the fusion engine. The threshold can, in theory, vary over the entire range of possible match scores. If a match score exceeds this threshold, the null hypothesis is rejected. If the match score falls below the threshold, the null hypothesis is accepted. This decision process using the threshold, λ_i, for sensor i can be summarized as,

$$u_i = \begin{cases} 1, & x_i >= \lambda_i \\ 0, & x_i < \lambda_i \end{cases} \quad \forall i . \qquad (1)$$

Let $[U]=[u_1,u_2,......u_N]$, be the binary vector of decisions generated by multiple classifiers based on decision thresholds $[\lambda_1,\lambda_2.....\lambda_N]$. These decisions can then be combined using a fusion rule of the form

$$u_f = f([U]) . \qquad (2)$$

The two errors, known as probability of false acceptance (P_{FA}) and probability of false rejection (P_{FR}) can be denoted as

$$P_{FA} = P(u_f = 1 / H_0) \quad (3)$$

$$P_{FR} = P(u_f = 0 / H_1) \quad (4)$$

where u_f is the final decision rendered by the fusion engine based on the decisions output by the individual classifiers. The goal is to minimize these errors.

TABLE 1: CONSTRUCTING THE FUSION RULE FOR TWO CLASSIFIERS

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

With two classifiers, the fusion rule consists of 4 bits, as shown in Table 1. In this table, u_1 is the decision output by the first classifier while u_2 is that of the second classifier. The global fusion rule is of length l bits where

$$l = \log_2 s . \quad (5)$$

$s = 2^{2^N}$ and N is the number of classifiers. The global decision rule replaces { d_0, d_1, d_2, d_3 } with 0's and 1's in their respective locations within f.

In order to formulate the problem, it is assumed that the probabilities of encountering an impostor or genuine score are the same (i.e, the prior probabilities are the same). Also, the costs of false accept and false reject are defined: C_{FA}: cost of false acceptance and C_{FR}: cost of false rejection. These are incorporated into a performance function for evaluating the fusion methodology. The Bayesian cost (error), which the paper intends to minimize, is

$$R = C_{FA} \times P(H_0) \times P_{FA} + C_{FR} \times P(H_1) \times P_{FR} \quad (6)$$

where

$$C_{FA} + C_{FR} = c \quad (7)$$

and c is a constant. In this paper we assume c=2. Here, (6) is a weighted linear multi-objective function, which needs to be minimized. The error probabilities (P_{FA}, P_{FR}) of the fused system can be estimated based on the fusion rule and the available training data set. They can be computed as:

$$P_{FA} = \sum_{i=0}^{l-1} d_i \times \left\{ P(u_1^{\,i}, u_2^{\,i}, ...u_N^{\,i} \mid H_0) \right\} \quad (8)$$

and

$$P_{FR} = \sum_{i=0}^{l-1} (1 - d_i) \times \left\{ P(u_1^{\,i}, u_2^{\,i}, ...u_N^{\,i} \mid H_1) \right\} \quad (9)$$

Here, u_j^i is used to indicate the value of u_j corresponding to d_i (see Table 1). Equations (8) and (9) require the calculation of joint probabilities. For example, for 2 classifiers there are 4 joint probabilities and for 5 classifiers there are 32 joint probabilities (Equation (5)). To reduce the computational effort involved we use the Bahadur-Lazarfeld expansion method [6, 7]. The method involves the estimation of only N-1 joint probability estimates for the N classifiers as shown below.

Correlation among classifiers is assumed under both hypotheses. Let $U = [u_1, u_2, \ldots u_n]$ be the vector of local decisions of individual classifiers. We first normalize the local decisions and generate a random variable z_j with zero mean and unit variance as

$$z_{j_h} = \frac{u_j - p_{j_h}}{\sqrt{p_{j_h} q_{j_h}}} \quad (10)$$

where $p_{j_h} = P(u_j = 1 \mid H_h)$, $q_{j_h} = 1 - p_{j_h}$ $\forall j$ and h denotes the hypothesis subscript '0' or '1'. Let

$$\overline{P}_{h=1}(U) = \prod_{j=1}^{N} (p_{j_1})^{u_j} (q_{j_1})^{(1-u_j)} \quad (11)$$

and

$$\overline{P_{h=0}}(U) = \prod_{j=1}^{N} (p_{j_0})^{(1-u_j)} (q_{j_0})^{u_j} \quad (12)$$

represent the joint probability estimates for independent classifiers for a given U. For correlated decisions, the joint probability estimates can be modified as

$$P_h(U) = \overline{P}_h(U)\left[1 + \sum_{i<j} \gamma_{ij_h} z_{i_h} z_{j_h} + \sum_{i<j<k} \gamma_{ijk_h} z_{i_h} z_{j_h} z_{k_h} + ...\right]. \quad (13)$$

The $\prod_{j=1}^{k} z_j$'s are the Bahadur-Lazarfeld polynomials given in [7]. The variable, γ, is the correlation coefficient,

$$\gamma_{123...n_h} = E\left(\prod_{j=1}^{n} z_{j_h} \right) \quad (14)$$

where

$$z_{j_h} = \frac{u_j - P(u_j = 1 / H_h)}{\sqrt{P(u_j = 1 / H_h) \times (1 - P(u_j = 1 / H_h))}} . \quad (15)$$

There are $2^N - 1$ correlation coefficients for N classifiers. However, only $2^N - 1 - N$ are calculated since the correlation coefficients are zero for N terms having single z's. Note that γ is independent of the values in U. It is the expected value of the product of z_j*'s* conditioned on the hypothesis. In (13), γ's are multiplied by z_j*'s*. z_j*'s* are a function of the values in U, as given in (15). An expansion of γ in the case of two classifiers results in

$$\gamma_{ij_h} = \frac{E(u_i u_j \mid H_h) - (P(u_i = 1 \mid H_h) \times P(u_j = 1 \mid H_h))}{\prod_{k=i,j} \sqrt{P(u_k = 1 \mid H_h) \times (1 - P(u_k = 1 \mid H_h))}} . \quad (16)$$

The fused error probabilities are estimated by replacing $\{P(u_1, u_2, ...u_n \mid H_h)\}$ in (8) and (9) with $\{P_h(U)\}$ as in (13).

2.1. Likelihood Ratio Test Based Decision Fusion Strategy

In this strategy, the optimal threshold for each classifier is found by minimizing (6) as in

$$\lambda_i = \arg\min_{\lambda_i} R^i \quad (17)$$

The joint probabilities are estimated for these fixed thresholds under both the hypotheses, and the optimal fusion rule is found using the likelihood ratio test (LRT). In the case of independent classifiers, it is merely the result of applying the Chair-Varshney rule (CVR) and is given by

$$\sum_{j=1}^{N}\left[u_j \log\left\{\frac{1-P_{FR_j}}{P_{FA_j}}\right\} + (1-u_j)\log\left\{\frac{P_{FR_j}}{(1-P_{FA_j})}\right\}\right] \underset{H_1}{\overset{H_0}{\lessgtr}} \log\left(\frac{C_{FA}}{2-C_{FA}}\right) \quad (18)$$

where u_j is the decision of the j^{th} classifier. For correlated classifiers, the optimal fusion rule is,

$$\Omega + \left[\log\left[\frac{1+\sum_{i<j}\gamma_{ij_1} z_{i_1} z_{j_1} + \sum_{i<j<k}\gamma_{ijk_1} z_{i_1} z_{j_1} z_{k_1} + ...}{1+\sum_{i<j}\gamma_{ij_0} z_{i_0} z_{j_0} + \sum_{i<j<k}\gamma_{ijk_0} z_{i_0} z_{j_0} z_{k_0} + ...}\right]\right] \underset{H_1}{\overset{H_0}{\lessgtr}} \Lambda(C_{FA}) \quad (19)$$

where Ω and $\Lambda(C_{FA})$ correspond to the left hand side and right hand side of (18) respectively. In the subsequent sections, the application of (19) is referred to as the LRT technique.

2.2. Particle Swarm Optimization based decision fusion Strategy

To jointly optimize the fusion rule and thresholds, a particle swarm optimization (PSO) based approach [3] is proposed. 'Particles' representing a possible solution to the multi-dimensional problem are 'flown' through the multi-dimensional search space. Each particle's fitness is evaluated using the Bayesian cost function (6). Each particle in the *N* classifier decision-level problem has *N+1* dimensions. The first N dimensions are the thresholds for each classifier and the *N+1*th dimension is the fusion rule. The classifier thresholds are assumed to be continuous. While the fusion rule is made of binary digits. Hence, each particle has two components, $\lambda_i = (\lambda_{i1}, \lambda_{i2} ....\lambda_{iN})$ representing continuous thresholds and $d_i = (d_{i1}, d_{i2} ....d_{il})$ representing the fusion rule, where the subscript i represents the particle number. Each dimension 'd_{iq}' in the binary component represents a binary bit corresponding to the fusion rule. An example of the fusion rule is given in Table 1. The particle for the binary component is l bits long.

The algorithm maintains in memory the state of the previous best position (as assessed using the cost function) found in the search space, called *'pbest'* represented as $\lambda_i^p = (\lambda_{i1}^p, \lambda_{i2}^p ....\lambda_{iN}^p)$. A velocity term along each dimension is defined as $V_i = (V_{i1}, V_{i2}, ..., V_{iN})$. The algorithm is given below.

1. The particles are initialized randomly. The "*pbest*" solutions are initially assigned the same values as the initial positions of the particles. The best position among all the particles, called "*gbest*", is determined based on their fitness function (6).

2. In each iteration, the velocity term, for both binary and continuous component, is updated. The continuous component of the particle is pulled in the direction of its own previous best position, λ_i^p, and the global best position, λ_g^p, found so far. This is apparent in the velocity update equation,

$$V_{iq}^{(t+1)} = \omega \times V_{iq}^{(t)} + U[0,1] \times \psi_1 \times (\lambda_{iq}^{p(t)} - \lambda_{iq}^{(t)}) + U[0,1] \times \psi_2 \times (\lambda_{gq}^{p(t)} - \lambda_{iq}^{(t)}) \quad \forall q \quad (20)$$

where U [0,1] is a sample from a uniform distribution, t represents a relative time index, ψ_1 and ψ_2, respectively, determine the impact of the previous best solution and global best solution on the particle's velocity. ψ_1, ψ_2, ω are set to 1, 1, and 0.8, respectively, in our simulation. For the binary PSO, the values of *'d'* are binary. A similar design and update strategy is used for the velocity vector. The position of the particle in the continuous space is updated using

$$X_{iq}^{(t+1)} = X_{iq}^{(t)} + V_{iq}^{(t+1)} . \quad (21)$$

For updating the position in binary space, the velocity is first transformed into a [0, 1] interval using the sigmoid function given by

$$S_{iq} = sig(V_{iq}) = \frac{1}{1+e^{-V_{iq}}} \quad (22)$$

where V_{iq} is the velocity of the i^{th} particle's q^{th} dimension. A random number is generated using a uniform distribution which is compared to the value generated from the sigmoid function and d_{iq} is estimated in the following manner.

$$d_{iq} = u(S_{iq} - U[0,1]) , \quad (23)$$

where u is a unit step function. The decision regarding d_{iq} is now probabilistic: that higher the value of V_{iq} the higher the value of S_{id}, thereby increasing the probability of deciding '1' for d_{iq}. As $V_{iq} \to \infty$, then $S_{iq} \to 1$ making it unlikely that d_{iq} will become zero again.

3. Once the particle is moved to the new location as determined by (20), (21), and (23), the particles representing the solution for the problem are evaluated using (8), (9) and (6). (8) and (9) are evaluated using the formulation in (13).

4. The "*pbest"* solution vector is updated if a better position, as assessed by the fitness function, is found.

Steps 1-4 are repeated until either convergence occurs or a preset performance is achieved. The algorithm can also be run for a fixed number of iterations. In our experiments, we used 10 particles and 1000 iterations to deduce the optimal fusion strategy. The final solution achieved by the PSO algorithm is $\lambda_g{}^p$ (the thresholds for the classifiers) and the $d_g{}^p$ (optimal fusion rule), where 'g' is the index of the *gbest* particle. Once the thresholds and fusion rule are obtained using the training data, they are applied on the test data.

The technique described above is referred to as the PSO technique in the subsequent sections.

2.3. Score level fusion using Z-norm

We compare the decision level fusion strategies described above to the score level fusion technique employing z-normalization [5]. Score level fusion typically employs a static threshold after combining the match scores originating from multiple classifiers.

3. Experimental Results

In this section, we present the results of the aforementioned schemes on two data sets. The first dataset is a synthetic bi-variate normal (2 classifiers) data set. Table 2 shows the values of the parameters for the marginal normal density functions under both the hypotheses (H_0 and H_1) for two classifiers ($N=2$). The covariance between the two classifiers is simulated by varying the Pearson correlation coefficients between –0.9 and 0.9 in steps of 0.2. It is assumed that both the genuine and impostor distributions have the same correlation coefficients. We call this symmetric correlation. In our experiments, 100,000 samples are generated using these distributions for training and an equal number is used for testing. The second data set is derived from the publicly available NIST BSSR database. We use the face classifier portion of this dataset that has match scores of 3000 subjects corresponding to two different face matchers. There are 6,000 genuine scores and ~18 million (6,000x2,999) impostor scores for each matcher. We use 50% of the genuine scores (3000) and 0.53% of the imposter scores (96,000), from each matcher, to compose the training set. The same number of scores is used in the test set. The Pearson's correlation under H_0 and H_1 for both training and testing data is approximately 0.18 and 0.476, respectively. The PSO technique is then used to generate the decision level fusion configuration for multiple C_{FA} values ranging from 0.1 –1.9. Subsequently, these configurations are applied to the test data set and the receiver-operating characteristic (ROC) computed.

TABLE 2. ARTIFICIAL PARAMETERS OF THE MARGINAL NORMAL DISTRIBUTION OF TWO CLASSIFIERS.

Parameter	Classifier 1	Classifier 2
μ_{H_0}	47.3	67.7
μ_{H_1}	144.5	251.2
σ_{H_0}	43.8	52.6
σ_{H_1}	12.8	23

3.1. Fusion Using the LRT Technique

The synthetic training data set is used to identify the optimal fusion rule using the formulation presented in Section 2.3. A search for the threshold for each classifier is first performed to minimize (6). The precision of the threshold search is varied with different step sizes. Four different step sizes, i.e., [0.01, 0.1, 1, 2], were used on the training data. We observed that a higher precision in step size did not necessarily lead to a better performance. The best performance was achieved with a precision of step size 1. Once the thresholds for individual classifiers were computed, their values were fixed and the optimal fusion rule estimated using (19). The resultant fusion configurations were then applied to the test data.

A similar procedure was used to identify the thresholds and optimal fusion rule for the NIST training data set as well.

3.2. Fusion Using the PSO Technique

The PSO algorithm generates the optimal thresholds *and* the optimal fusion rule using the training dataset. The threshold for classifier 1 for the synthetic dataset varied from 82, for a correlation of -0.9, to 101, for a correlation of +0.9. The threshold for classifier 2 varied from 110, for a correlation of -0.9, to 190 for a correlation of +0.9. The PSO scheme converged to the 'AND' rule as the optimal fusion rule across all the correlation factors. Multiple runs of PSO on the same training data set resulted in the same configuration.

The PSO technique was then run on the NIST BSSR training data set.

3.3. Results on test data

The fusion configurations deduced using the training data are then applied on the test data. Results in this section are reported on both datasets.

Synthetic dataset: Figure 1 compares the Bayesian error of the PSO and the LRT techniques. For low (both negative and positive) correlation values, the PSO technique outperforms the LRT technique. However, at high correlation values, the performance of the PSO deteriorates and approaches that of the LRT. The main reason for poor performance of LRT is the two-step optimization procedure adopted. First, the thresholds are deduced and then the optimal fusion rule is derived using the LRT. The PSO, however, jointly optimizes the thresholds for the classifiers and the fusion rule.

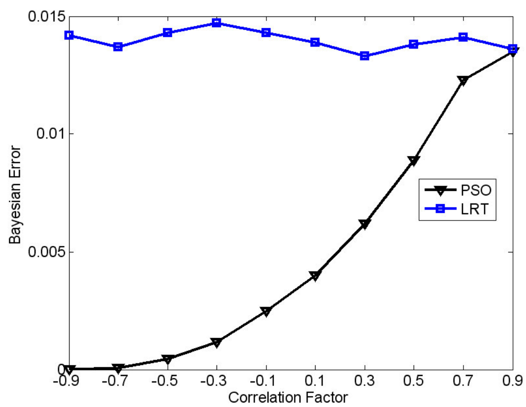

Figure 1: Bayesian error plot comparing the PSO technique and the LRT technique on the synthetic dataset

NIST-BSSR dataset: Table 3 presents the results observed on the NIST test set. Once again, the thresholds and the fusion rule derived from the training set were used. In this case also, the PSO technique outperforms the LRT technique for higher costs of false acceptance. The performance advantages of PSO over LRT are a function of the underlying correlation and the performance of constituent classifiers. In our experiments, the classifier, denoted as Face-2 dominates the classifier denoted as Face-1.

TABLE 3 COMPARING THE BAYESIAN ERROR FOR DIFFERENT DECISION LEVEL FUSION STRATEGIES ON THE NIST DATASET AFTER VARYING THE COST FUNCTION

C_{FA}	LRT	PSO
0.4	**0.0847**	0.08931
0.8	0.113200	**0.10408**
1	0.109541	**0.09922**
1.5	0.076512	**0.07027**

3.4. Comparison with Score Level Fusion Techniques

Figure 2 compares the PSO-technique with a simple score level fusion technique (sum-rule using z-normalization) on the synthetic dataset. We make the following observations: (a) the PSO technique outperforms the score-level fusion scheme when the classifiers are positively correlated, (b) the performance of the latter further deteriorates as the positive correlation value increases, and (c) the PSO technique and the sum rule fusion perform identically well for negatively correlated values. The PSO technique achieves an average of ~45% performance improvement over the score-level fusion technique across different positive correlation values.

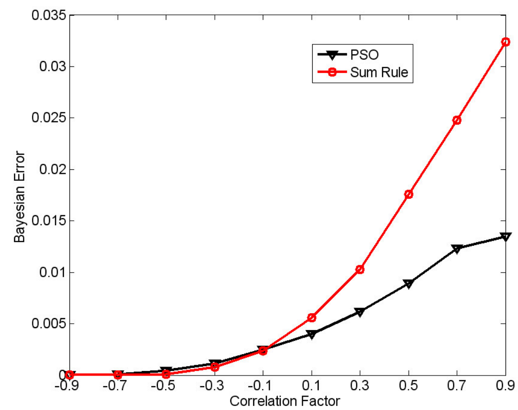

Figure 2: Bayesian error plot comparing the PSO technique with the Sum rule on the synthetic dataset

TABLE 4 COMPARING THE BAYESIAN ERROR OF PSO-BASED AND SUM RULE FUSION TECHNIQUES ON THE SYNTHETIC TEST DATASET FOR POSITIVE CORRELATION VALUES

Correlation	Sum Rule	PSO	% Improvement
0.1	0.0056	**0.004**	**28.57**
0.3	0.0103	**0.0062**	**39.80**
0.5	0.0176	**0.0089**	**49.43**
0.7	0.0248	**0.0123**	**50.40**
0.9	0.0324	**0.0135**	**58.33**

TABLE 5 COMPARING BAYESIAN ERRORS OF PSO-BASED AND.SUM RULE FUSION TECHNIQUES ON THE NIST BSSR DATASET

CFA	Sum Rule	PSO	% Improvement
0.5	0.11281	**0.09791**	**13.2%**
1	0.11409	**0.09922**	**13.03%**
1.4	0.09214	**0.08087**	**12.22%**

The receiver operating characteristic curves for the two classifiers, the PSO technique and the LRT technique are shown in Figure 3 for the NIST dataset. The PSO-technique outperforms the sum rule using z-normalization.

At lower false acceptance probabilities (higher C_{FA}) the PSO technique, LRT technique converge to the Face-2 classifier's performance. The Face-2 classifier is a better performing classifier between the two classifiers. Inspection of the fusion rule shows that both techniques employed 'Second classifier' only rule as the optimal fusion rule. This fusion rule ignores the decision from the classifier Face-1. Hence only Face-2 classifier is used. The sum rule utilizes information (scores) from both the classifiers and performs poorly. The LRT and PSO based techniques have capability to switch between fusion functions to achieve higher performance benefits. Table 5 presents the results in terms of the Bayesian error. It is observed that the PSO technique achieves a ~13% performance improvement across different C_{FA}s.

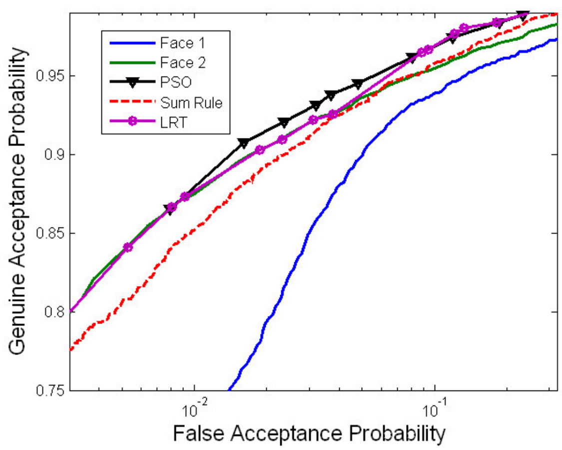

Figure 3: Receiver Operating Characteristic Curves obtained on the NIST BSSR dataset

4. Conclusions

In this paper we discussed two multi-classifier decision fusion methods for dealing with correlation in biometric verification. We presented the optimal fusion rule based on the Likelihood Ratio Test (LRT) where the classifier thresholds are first independently deduced. The fusion rule is based on estimating probability densities of the decision vector and applying the LRT after incorporating the correlation structure between classifiers. The performance of this optimal fusion rule was observed to be sensitive to the underlying thresholds of each classifier. To improve the performance we presented a PSO-based decision fusion strategy. PSO searches simultaneously for the optimal thresholds and the fusion rule based on the training data.

We also compared the PSO strategy with score level fusion. The PSO strategy performs better than the sum rule using z-norm. We achieve significant gains of ~45% on an average using the PSO on the synthetic test data for positively correlated classifiers, and a gain of ~13% on an average on the NIST-BSSR dataset.

Finally, Figure 4 summarizes our comparisons between score level fusion and decision level fusion using the symmetrically correlated synthetic data set. The PSO technique and the z-norm strategy perform identically well for the negative correlation factors. As correlation increases the performance of the score level strategy deteriorates. However, the score level strategy at lower correlation is better than the LRT technique.

In the future, we will employ the PSO and LRT based strategy to achieve optimal configurations under the Neyman Pearson criterion. The Neyman Pearson criterion will allow the designer to specify a particular false acceptance rate whilst designing the fusion module, thereby permitting flexibility.

References

[1] A. Ross, K. Nandakumar, and A. K. Jain, "Handbook of Multibiometrics", Springer Publishers, 1st edition, 2006.

[2] P. K Varshney, Distributed Detection and Data Fusion, Springer-Verlag New York, Inc., 1996.

[3] J. Kennedy, R. Eberhart and Y. H. Shi, Swarm Intelligence, Morgan Kaufmann Publishers, 2001.

[4] K.Venkataramanni, B. V. K Vijaya Kumar, "Role of Statistical Dependence Between Classifier Scores in Determining the Best Decision Fusion Rule for Improved Biometric Verification", MRCS 2006, pp. 489 –496.

[5] A. K. Jain, K. Nandakumar, A. Ross, "Score Normalization in Multimodal Biometric Systems , " Pattern Recognition, 38, pp. 2270 – 2285, 2005.

[6] M. Kam, 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.

[7] R. O Duda, P. Hart and D. Stork, Pattern Classification and Scene Analysis, Wiley, 1973, New York

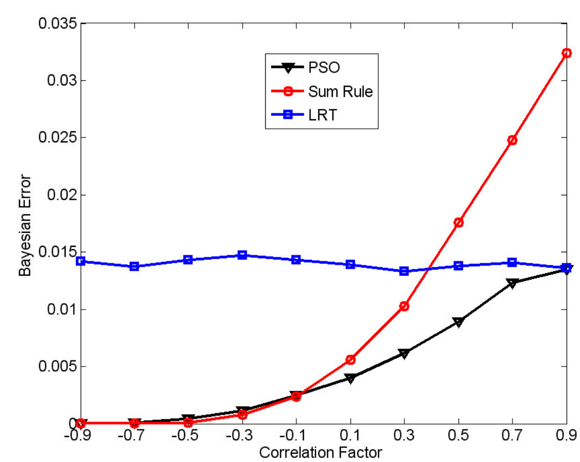

Figure 4: Bayesian error plot as a function of correlation on the synthetic dataset for the three different strategies discussed in this paper