Combining multiple biometric traits with an order-preserving score fusion algorithm

Yicong Liang[a,b,*], Xiaoqing Ding[a], Changsong Liu[a], Jing-Hao Xue[b]

[a] *Department of Electronic Engineering, Tsinghua University, Beijing 100084, China*
[b] *Department of Statistical Science, University College London, London WC1E 6BT, UK*

Abstract

Multibiometric systems based on score fusion can effectively combine the discriminative power of multiple biometric traits and overcome the limitations of individual trait, leading to a better performance of biometric authentication. To tackle multiple adverse issues with the established classifier-based or probability-based algorithms, in this paper we propose a novel order-preserving probabilistic score fusion algorithm, Order-Preserving Tree (OPT), by casting the score fusion problem into an optimisation problem with the natural order-preserving constraint. OPT is an algorithm fully non-parametric and widely applicable, not assuming any parametric forms of probabilities or independence among sources, directly estimating the posterior probabilities from maximum likelihood estimation, and exploiting the power of tree-structured ensembles. We demonstrate the effectiveness of our OPT algorithm by comparing it with many widely-used score fusion algorithms on two prevalent multibiometric databases.

Keywords: Score fusion, Multibiometric system, Order-preserving algorithm, Tree-structured ensemble

---

1. Introduction

Biometric systems have found an increasingly wide range of applications in both science and industry. However in many of these applications, unibiometric systems, which exercise only a single biometric trait, often cannot fulfil

---

[*] Corresponding author: Tel.: +86-15010266579;
Email address: `liangyc04@gmail.com` (Yicong Liang)

Preprint submitted to Neurocomputing *February 29, 2016*

the biometric authentication tasks. This is mainly due to their limited abilities to represent subjects and prevent spoofs. To overcome such limitations, multibiometric systems have been developed. From fusing several different types of complementary biometric traits together, multibiometric systems can benefit substantially in representing and discriminating subjects, as well as in preventing spoofs since it is much more difficult to cheat simultaneously in all the information sources than to deceive a unibiometric system.

In a multibiometric system, there are four stages in which information fusion can be implemented, namely the sensor stage, feature stage, score stage and decision stage, listed from the earliest to the latest. To fuse information in a later stage means the ease of implementation, at a cost of more information loss. To date, fusion in the score stage is generally considered to provide an appropriate trade-off and preferred by many researchers [1, 2, 3].

Existing score fusion algorithms can be divided into two main categories: classifier-based algorithms and probability-based algorithms.

Classifier-based algorithms tackle the fusion problem as a pattern classification task. In this framework, source scores of a sample are used as the input features of a classifier to obtain the predicted class. The classifier is trained on the training samples to minimise the training error by using traditional pattern recognition algorithms. Traditional classifier-based approaches include linear classifiers with minimised least squares error [4] or L_1-norm soft margin error [5, 6], reduced multivariate polynomial classifier [7], support vector machine [8] and single hidden layer feedforward neural network [9]. Other advanced techniques in the pattern recognition society, such as semi-supervised learning [10, 11], ensemble learning [12] and kernel tricks [6], can also be transferred without difficulty to solve the fusion problem. Two recent examples of classifier-based algorithms are FWOT [13] and minCq [14, 15]. FWOT optimises an objective function that is a combination of the squared hinge loss and a 2-norm regulariser. The classifier structure of FWOT is a close resemblance of a single-layer neural network. The algorithm minCq, on the other hand, tailors the score fusion problem into a PAC-learning framework and obtains an optimal linear fusion algorithm by minimising an upper bound of the true error risk. Both algorithms hold good ability of generalisation.

Classifier-based algorithms can directly exploit the great progress made in the pattern recognition realm. However, an issue with these algorithms is that they are suboptimal in score fusion, because, different from the usual features found in a pattern recognition task, a source score is by nature that a higher score implies a higher probability of the sample belonging to the genuine class. This intrinsic characteristic, as informative prior knowledge, has been largely neglected by the classifier-based algorithms. Particularly, when the training samples are insufficient, the neglect of this prior knowledge may worsen overfitting and thus the performance of biometric authentication. Another issue with these algorithms is that in the design of a classifier-based fusion algorithm there are often tuning parameters, the optimisation of which may not be a trivial problem and issues such as local optimum or overfitting may exist.

Probability-based algorithms tackle the fusion problem in a probabilistic manner. These algorithms usually consist of two steps. In the first step, they normalise all source scores to make their values between $[0, 1]$, and treat the normalised score as the posterior probability of the genuine class given the corresponding source as evidence. In the second step, they merge multiple posterior probabilities into a single posterior probability given all sources as evidence. To make the merge work, these algorithms usually need extra assumptions, made a priori or with the help of training samples or as a combination of the former two. For example, Kittler et al. [16] show that, with the assumption that all source scores are mutually conditionally independent, the commonly used Product rule can be derived. On the other hand, given the assumption that posterior probabilities of each classifier do not deviate dramatically from the priors, the Sum rule can be induced. Other off-the-shelf fusion rules, such as the Max, Min, Median and Majority Vote rules, can all be derived as the midway rules of the Sum and Product rules. Terrades et al. [17] also assume that source scores are mutually independent. Prabhakar and Jain [18] and Nandakumar et al. [1] estimate the probability density function of the training samples and use this function to assist the determination of the merged posterior probability. Ma et al. [19, 20] assume parametric forms of the merged posterior probability, which take the dependency between scores into consideration, and use the training samples to estimate the parameters. As a recent example, Cheema et al. [21] assume that the merged posterior probability is a linear combination of source probabilities. Their method solves a constrained quadratic optimisation problem to decide the optimal weights and can deal with the cases with more than two classes.

There are also several issues of concern to these probability-based algorithms. Firstly, except the density estimation algorithms [18, 1], these probability-based algorithms all make some assumptions about the merged posterior probability, which may not be fulfilled in practice and thus may limit the generalisability of them. Secondly, as for [18, 1], the density estimation procedures will induce hyper-parameters, such as the number of Gaussian mixtures in [1] and the Parzen window width in [18], and the estimation results may be unreliable when the sample dimension is high. Thirdly, in the score normalisation step, most of these algorithms apply heuristic techniques such as the min-max, z-score and tanh algorithms [22], and it is in question how well the normalised result reflects the true posterior probability given the score.

To tackle these adverse issues, in this paper we propose a novel probability-based score fusion algorithm, termed Order-Preserving Tree (OPT). The advantages of OPT are threefold. Firstly, OPT treats both the score normalisation and the posterior probability merging procedure as an constrained optimisation problem. The only constraint in optimisation, which is also the only assumption that OPT makes, is the intrinsic characteristic of order preserving: For any two samples A and B, if every source score suggests that the A is no less likely than B to belong to the genuine class, then the fusion result should also give the same suggestion. Secondly, OPT does not assume any parametric form of probabilities, making itself enjoy widely applicability. Moreover, being fully non-parametric, OPT has no hyper-parameters that need to be tuned, which makes the training procedure efficient. Thirdly, OPT bypasses the procedure of probability density estimation of samples; it instead directly estimates the posterior probabilities themselves. This not only can avoid the issues with density estimation, but also according to Occam's Razor can be more suitable for a task like score fusion. To avoid the problem of the curse of dimensionality, we adopt a tree-structured ensemble to hierarchically merge multiple source scores.

To demonstrate the effectiveness of our OPT algorithm, we conduct extensive experiments on the NIST-BSSR1 and XM2VTS databases, two public-domain databases specially designed to evaluate score fusion algorithm in biometric authentication. Our algorithm demonstrates superior performance compared with many off-the-shelf, classifier-based and probability-based score fusion techniques.

The remainder of this paper is organised as follows. In Section 2 and Section 3 we give the basic framework and implementation details of our OPT algorithm, respectively. The experimental results are summarised in Section 4. The work is concluded in Section 5.

2. Algorithmic framework

In this paper, we will focus on the two-class problem for simplicity. This is the typical case for the multibiometric verification system, where the target is to predict whether a pair of samples belong to the same subject, given a set of biometric similarity scores. We start by establishing notation.

2.1. Notation

The genuine class and the imposter class are denoted by ω_+ and ω_-, respectively. Suppose that there are N training samples, denoted by $x_1, \ldots, x_N$, with corresponding class labels $y_1, \ldots, y_N$, where $y_i = 1$ if $x_i \in \omega_+$ and $y_i = 0$ otherwise. Suppose for each sample x there are K source scores and use $S_i(x)$ to denote the ith source score. We assume that a higher score indicates a higher posterior probability (suggested by the score) of belonging to ω_+. We use $P_{i_1 \ldots i_k}(x)$ to denote the posterior probability $Pr(x \in \omega_+ | S_{i_1}(x), \ldots, S_{i_k}(x))$ for short.

2.2. Overall structure

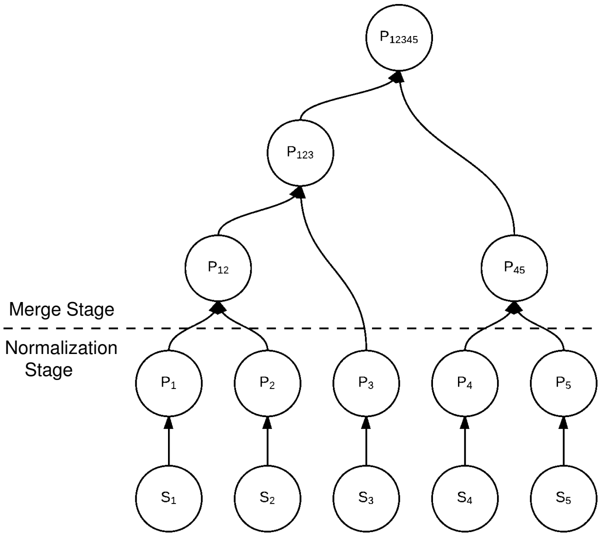


Figure 1: A realisation of our OPT algorithm with five source scores. S_i indicates the ith score, and $P_{i_1 \ldots i_k}$ denotes the posterior probability $Pr(\omega_+ | S_{i_1}, \ldots, S_{i_k})$.

The overall structure of our OPT algorithm is illustrated in Fig. 1. It is divided in two stages: a normalisation stage and a merge stage. In the normalisation stage, we transform each source score into a posterior probability suggested by the score, i.e. we calculate P_i for every i given S_i. In the merge stage, we merge the information given by all P_is together and obtain the final posterior probability $P_{1,\ldots,K}$, i.e. we calculate the conditional probability $Pr(x \in \omega_+|P_1(x), \ldots, P_K(x))$.

The support of the conditional probability $Pr(x \in \omega_+|P_1(x), \ldots, P_K(x))$ is K-dimensional. Since we do not give the functional any parametric form, it will encounter the curse of dimensionality to directly estimate the probability. We apply a tree-like hierarchical structure to circumvent this problem, as illustrated in Fig. 1. Using this hierarchical structure, we only need to calculate a two-dimensional conditional probability function in each node.

We propose the methodology of a score normalisation algorithm and a two-dimensional merge algorithm in Sections 2.3 and 2.4, respectively. We present the implementation details of our algorithms in Section 3.

2.3. Score normalisation

Different from previous heuristic score normalisation approaches such as the min-max, z-score or tanh algorithms, our approach tackles the score normalisation problem by using a well-founded probabilistic framework. Specifically, we obtain the maximum likelihood estimates of the posterior probabilities of all training samples, and expand these values to a function by interpolation. For the kth source score, we simultaneously estimate $P_k(x_i)$, for $i = 1, \ldots, N$. We assume that all training scores are independently sampled. According to the definition of $P_k(\cdot)$, for every $x_i \in \omega_+$, the probability of the presence of its label is $P_k(x_i)$, while for every $x_i \in \omega_-$, the probability of the presence of its label is $1 - P_k(x_i)$. Therefore, the probability of the presence of all training labels is

$$\prod_{x_i \in \omega_+} P_k(x_i) \prod_{x_i \in \omega_-} (1 - P_k(x_i)) \ .$$

Therefore we can estimate all $P_k(x_i)$s by maximising the logarithm of the above equation, which is

$$\mathcal{L}_k(P_k(\cdot)) = \sum_{i=1}^{N} (y_i \ln P_k(x_i) + (1 - y_i) \ln(1 - P_k(x_i))) \ . \tag{1}$$

The only constraint that all $P_k(x_i)$s should obey is the intrinsic order-preserving constraint, which can be represented as

$$(P_k(x_i) - P_k(x_j))(S_k(x_i) - S_k(x_j)) \geq 0,\ \forall 1 \leq i < j \leq N\ . \tag{2}$$

We maximise (1) with constraint (2) to obtain all $P_k(x_i)$s. Afterwards, and interpolate these values to obtain the normalisation function for the kth source score, i.e., we build a new function $f_k(\cdot)$ whose domain is $\mathbb{R}$, and for every training sample x_i, we have $f_k(S_k(x_i)) = P_k(x_i)$. Then for every test samples x, we simply let $P_k(x) = f_k(S_k(x))$. The details of the optimisation and interpolation procedure will be presented in Section 3.1.

2.4. Two-dimensional merge

At every node of our tree-structured probability merger, the task is to obtain a posterior probabilistic function $P_{\mathcal{KL}}$ based on the outputs of its two branches $P_{\mathcal{K}}$ and $P_{\mathcal{L}}$, where $\mathcal{K}$ and $\mathcal{L}$ can be single integers (indicating leaves of the tree) or sets of integers (indicating branch nodes). Like in Section 2.3, we solve the problem through maximum likelihood estimation, maximising

$$\mathcal{L}_{\mathcal{KL}}(P_{\mathcal{KL}}(\cdot)) = \sum_{i=1}^{N} \left(y_i \ln P_{\mathcal{KL}}(x_i) + (1 - y_i)\ln(1 - P_{\mathcal{KL}}(x_i))\right)\ . \tag{3}$$

The order-preserving constraint is, nevertheless, different in form from the case of the unary operation in Section 2.3:

$$(P_{\mathcal{KL}}(x_i) - P_{\mathcal{KL}}(x_j))\mathbf{1}_{\{P_{\mathcal{K}}(x_i) \geq P_{\mathcal{K}}(x_j), P_{\mathcal{L}}(x_i) \geq P_{\mathcal{L}}(x_j)\}} \geq 0,\ \forall i, j, \tag{4}$$

where $\mathbf{1}_{\{X\}}$ is the indicator function of the proposition X. Here the constraint is that, when both source probabilities agree, the merged probability also must agree with them.

In short, when estimating $P_{\mathcal{KL}}(\cdot)$ for every training sample, we try to let its value on positive sample as large as possible and its value on negative samples as small as possible, the only constraint is that it will preserve the order of two samples if such order is agreed by both the source scores.

Similarly to Section 2.3, we maximise (3) with constraint (4) to obtain all $P_{\mathcal{KL}}(x_i), i = 1, \ldots, N$. The details of the optimisation procedure can be found in Section 3.2.

3. Implementation details

In this section, we present the implementation details of our OPT algorithm, which include the optimisation procedures to solve (1) and (3).

3.1. Maximisation of (1) with constraint (2)

We firstly sort all samples by their scores in the ascending order. To simplify the notation, we assume that $S_k(x_1) \leq S_k(x_2) \leq \ldots \leq S_k(x_N)$. Therefore, according to the order-preserving constraint (2), we have $P_k(x_1) \leq P_k(x_2) \leq \ldots \leq P_k(x_N)$.

Then we partition all $P_k(x_i)$s based on whether the inequality relations are strict. Specifically, we define a *partition* as follows.

Definition 1. *A partition related to the posterior function, denoted by* $Par(P_k)$, *is a partition of the interval* $\{1, \ldots, N\}$ *into* $c+1$ *intervals* $\{1, \ldots, j_1\}, \{j_1 + 1, \ldots, j_2\}, \ldots, \{j_c + 1, \ldots, N\}$, *such that*

$$\begin{aligned} &P_k(x_{j_l+1}) = P_k(x_{j_l+2}) = \ldots = P_k(x_{j_{l+1}}),\ l = 0, \ldots, c\ , \\ &P_k(x_{j_1}) < P_k(x_{j_2}) < \ldots < P_k(x_N)\ , \end{aligned} \tag{5}$$

where $j_0 = 0, j_{c+1} = N$.

It follows that, if P_k is the solution of (1), $Par(P_k)$ will have the following two properties:

1.
$$P_k(x_{j_l+1}) = \frac{\sum_{u=j_l+1}^{j_{l+1}} y_u}{j_{l+1} - j_l},\ l = 0, \ldots, c\ . \tag{6}$$

2. For $l = 0, \ldots, c$, there does not exist an integer $t \in (j_l, j_{l+1})$, such that
$$\frac{\sum_{u=j_l+1}^{t} y_u}{t - j_l} < \frac{\sum_{u=t+1}^{j_{l+1}} y_u}{j_{l+1} - t}\ . \tag{7}$$

For short, we denote the right hand of (6) by $\overline{[j_l + 1, j_{l+1}]}$.

The proof of these two properties is straightforward. If there exists an l which violates (6), then we can slightly modify all posterior probabilities in the corresponding interval towards $\overline{[j_l + 1, j_{l+1}]}$, which will make (1) larger without violating the order-preserving constraint. In addition, if there exist an l and a t which violate (7), then we can separate $\{j_l+1,\dots,j_{l+1}\}$ into two smaller intervals $\{j_l+1,\dots,t\}$ and $\{t+1,\dots,j_{l+1}\}$, and slightly decrease the posterior probabilities of the former interval and increase the posterior probabilities of the latter interval, which will also make (1) larger without violating the order-preserving constraint.

Moreover, we have the following proposition:

Proposition 1. *The partition which satisfies both (6) and (7) is unique.*

Proof: If we have two different partitions P_k and P'_k both satisfy (6) and (7), we firstly find the first different intervals of these two partitions. To simplify the notation, we assume their first intervals are different, i.e. $j_1 \neq j'_1$, we assume $j_1 > j'_1$, and we note l as the largest number such that $j'_l < j_1$. Then we must have $\overline{[j'_l+1, j_1]} \leq \overline{[1, j'_l]}$, as otherwise P_i will violate (7). However, as $\overline{[1, j'_l]} < \overline{[j'_l+1, j'_{l+1}]}$, it follows that j_1 can separate $[j'_l+1, j'_{l+1}]$ into two smaller intervals such that $\overline{[j'_l+1, j_1]} < \overline{[j_1+1, j'_{l+1}]}$, which indicates P'_i violates (7). ■

According to Definition 1 and Proposition 1, we can confirm that: If we find a partition which satisfies both (6) and (7), then the posterior probability function that it indicates is the maximum likelihood solution of (1) with constraint (2). The algorithm to find such an partition is displayed in Algorithm 1.

In the test phase, we should interpolate $P_k(x_i)$s into a probability function with support $\mathbb{R}$. In order to maintain the interval number c, we apply the nearest-neighbour interpolation method, i.e. for each test sample x, $P_k(x) = P_k(x_i)$ if $S_k(x_i)$ is the nearest to $S_k(x)$ among all the training scores.

3.1.1. Complexity analysis

The number of intervals outputted by Algorithm 1, c, plays a critical role in the complexity of our OPT algorithm. The complexity of Algorithm 1 is at most $O(N \times c)$ and, as will be seen in Section 3.2, the complexity of the 2-D merging algorithm will highly depend on the value of c.

We conduct an experiment to illustrate the relationship of c, N and the running time of Algorithm 1. We randomly generate multiple positive and negative sample sets with various sizes from 10^4 to 10^8. In every sample set, the positive and negative sample scores are generated from two normal distributions with the same variance and different means. The difference between the positive score mean and the negative score mean is controlled

Algorithm 1 Optimal partition search: one-dimensional

```
Input:
    Scores $S_k(x_1) \leq \ldots \leq S_k(x_N)$, corresponding labels $y_1, \ldots, y_N$.
Output:
    A set of integers $j_1, \ldots, j_c$, such that the partition $\{1, \ldots, j_1\}, \{j_1 + 1, \ldots, j_2\}, \ldots, \{j_c + 1, \ldots, N\}$ satisfies (7);
    $P_k(x_i), i = 1, \ldots, N$.
Initialise $c \leftarrow 0$, $r \leftarrow 0$, $u_c \leftarrow 0$, $v_c \leftarrow 0$;
for $l = 1$ to $N$ do
    if $l = 1$ or $S_k(x_l) > S_k(x_l - 1)$ then
        $u \leftarrow 1$, $v \leftarrow y_l$;
    else
        $u \leftarrow u + 1$, $v \leftarrow v + y_l$;
    end if
    if $l \neq N$ and $S_k(x_l) = S_k(x_l + 1)$ then
        go to the next loop;
    end if
    while $c > 0$ and $v/u \leq v_c/u_c$ do
        $u \leftarrow u + u_c$, $v \leftarrow v + v_c$;
        $c \leftarrow c - 1$;
    end while
    $c \leftarrow c + 1$;
    $j_c \leftarrow l$;
    $u_c \leftarrow u$;
    $v_c \leftarrow v$;
end for
Calculate $P_k(x_i)$, $i = 1, \ldots, N$, using (6).
```

to let the equal error rate of the score be 10%. The ratio between positive and negative samples is 1:999. Algorithm 1 is performed on each score set to evaluate the number of intervals c and the running time. For every specific N, 10 sample sets are independently generated to estimated the variance of the concerned indicators. The experiment is conducted on a MacBook Air.

Table 1: The outputted number of intervals c and the running time T of Algorithm 1 for different sample sizes N.

N	10^4	10^5	10^6	10^7	10^8
c	8.4 ± 1.1	24.8 ± 2.4	59.1 ± 3.8	135 ± 8.9	309.8 ± 6.1
T(s)	0.032 ± 0.001	0.319 ± 0.012	3.461 ± 0.164	38.61 ± 1.53	468.8 ± 40.1

The experimental results are summarised in Table 1, from which we can observe that c is far smaller than N, and the running time of Algorithm 1 is nearly linear with N.

3.2. Maximisation of (3) with constraint (4)

Let c_k and c_l denote the interval numbers of the partitions corresponding to $P_{\mathcal{K}}$ and $P_{\mathcal{L}}$, respectively. Since the same values of $(P_{\mathcal{K}}(x), P_{\mathcal{L}}(x))$ will result in the same $P_{\mathcal{KL}}(x)$, we will only consider $c_k \times c_l$ cases instead of $N \times N$ different score pairs, which leads to a much reduced computational complexity.

We firstly build two $c_k \times c_l$ matrices M_+ and M_-, where element $M_+(u, v)$ is the fraction of genuine training samples, that have been partitioned into the uth interval according to $P_{\mathcal{K}}$ and into the vth interval according to $P_{\mathcal{L}}$, and M_- is the matrix similarly constructed for the imposter training samples. What we will do is to build another $c_k \times c_l$ matrix R, where element $R(u, v)$ is the corresponding merged probability.

Following the methodology in Section 3.1, we also partition R into a set of two-dimensional intervals, where each interval has the same merged probability. To achieve this, we first make the following three definitions of two-dimensional intervals.

Definition 2. *A two-dimensional interval $\mathcal{I}$ of a matrix is a set of elements of this matrix, such that $\forall (u_1, v_1) \in \mathcal{I}$, $\forall (u_2, v_2) \in \mathcal{I}$, if $u_1 \le u_2$ and $v_1 \le v_2$, then $\forall u \in [u_1, u_2]$, $\forall v \in [v_1, v_2]$, we have $(u, v) \in \mathcal{I}$.*

Definition 3. *A lowerleft neighbour of a two-dimensional interval* $\mathcal{I}$ *is another two-dimensional interval* $\mathcal{J}$*, which has no intersection with* $\mathcal{I}$ *and has at least one element* (u_0, v_0) *such that* $(u_0 + 1, v_0) \in \mathcal{I}$ *or* $(u_0, v_0 + 1) \in \mathcal{I}$*. Besides,* $\forall (u, v) \in \mathcal{J}$*,* $\exists (s, t) \in \mathcal{I}$*, such that* $s \geq u$ *and* $t \geq v$*.*

Definition 4. *A lowerleft sub-interval of a two-dimensional interval* $\mathcal{I}$ *is another two-dimensional interval* $\mathcal{J}$*, where* $\mathcal{J} \subsetneqq \mathcal{I}$*, and* $\forall (u, v) \in \mathcal{J}$*,* $\forall (s, t) \in \mathcal{I}$*,* $s \leq u$ *and* $t \leq v$*, we have* $(s, t) \in \mathcal{J}$*.*

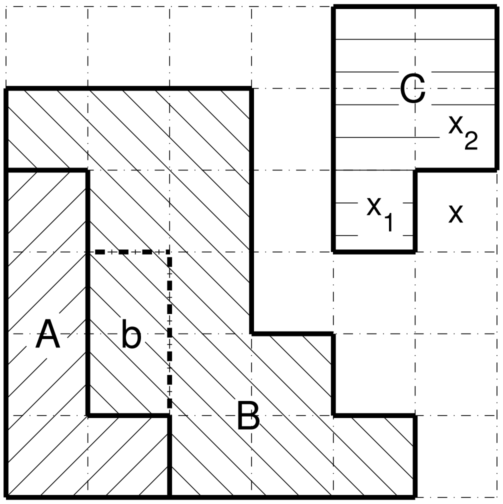


Figure 2: Illustration of two-dimensional interval, lowerleft neighbor and lowerleft sub-interval.

We illustrate the corresponding concepts in Figure 2. In the figure, A and B (with b as a part of it) are both two-dimension intervals, while C is not a two-dimension interval because it includes x_1 and x_2 but does not include x. A is a lowerleft neighbour of B, and b is a lowerleft sub-interval of B (A is not a lowerleft neighbour of b).

Similarly to the one-dimensional case, this partition should has the following two properties:

1. For each interval $\mathcal{I}$, denote the merged probability of every element in $\mathcal{I}$ by $R_{\mathcal{I}}$, then

$$R_{\mathcal{I}} = \frac{\sum_{(u,v)\in\mathcal{I}} M_+(u,v)}{\sum_{(u,v)\in\mathcal{I}} M_+(u,v) + \sum_{(u,v)\in\mathcal{I}} M_-(u,v)} \ . \tag{8}$$

   For short, we denote the right hand of (8) by $\overline{\mathcal{I}}$, termed the average ratio of $\mathcal{I}$.

2. For each interval $\mathcal{I}$, and for any of its lowerleft sub-interval $\mathcal{J}$, $\overline{\mathcal{J}} \geq \overline{\mathcal{I}}$.

To find a partition that satisfies these two properties, we design an iterative algorithm, as shown in Algorithm 2.

```
Algorithm 2 Optimal partition search: two-dimensional
Input:
    Matrices $M_+$, $M_-$.
Output:
    Merged probability matrix $R$.
Initialize $c \leftarrow 1$, $\mathcal{I}_1 \leftarrow \{R\}$, $flag_{exit} \leftarrow 0$;
while $flag_{exit} = 0$ do
    for $l = 1$ to $c$ do
        Find every lowerleft neighbours of $\mathcal{I}_l$ whose average ratio is no less than $\overline{\mathcal{I}_l}$;
        Merge these neighbours into $\mathcal{I}_l$;
    end for
    Reorder all intervals, update $c$.
    for $l = 1$ to $c$ do
        If there exists an lowerleft sub-interval $\mathcal{I}_{l1}$ of $\mathcal{I}_l$ and its complementary interval $\mathcal{I}_{l2}$ such that $\overline{\mathcal{I}_{l1}} < \overline{\mathcal{I}_{l2}}$, divide $\mathcal{I}_l$ into two intervals $\mathcal{I}_{l1}$ and $\mathcal{I}_{l2}$;
    end for
    Reorder all intervals, update $c$;
    if No change of partition has been made in this while loop then
        $flag_{exit} \leftarrow 1$;
    end if
end while
Calculate $R$ using (8).
```

In the test phase, since every source score of each test sample is guaranteed to fall into a one-dimensional interval by the nearest-neighbour interpolation in the normalisation stage (see Section 3.1), no further interpolation is needed in the merge stage.

3.2.1. Convergence analysis and complexity analysis

Every iteration in Algorithm 2 is composed of two stages. The first stage, which merges valid lowerleft neighbours for every current interval, has the complexity $O(c^2)$. For the second stage, which divides intervals into subintervals, we can use a dynamic programming algorithm to accomplish the task, and the complexity is $O(c_k \times c_l)$. Therefore, the complexity of Algorithm 2 is $O(N_{iter} \times (c^2 + c_k \times c_l))$, where N_{iter} is the number of iterations. In all practical experiments we conducted in Section 4, c, c_k and c_l are all less than 300, N_{iter} is less than 30, and the running time of the whole two-dimensional merging algorithm on a Macbook Air is less than one second.

Algorithm 2 will output a set of intervals which have the two properties just described. If we can prove that the partition with these properties is unique, we can conclude that Algorithm 2 will converge to the optimal solution of (3). However, since the two-dimensional space does not have a total order structure like the one-dimensional space has, this uniqueness proposition cannot be proved in the same way as the proof of its one-dimensional counterpart in Section 3.1. We note that as yet we have not accomplished in proving this proposition. Nevertheless, we also have not found a counter-example having two partitions both satisfying the two properties. In practice, we have tried using different initialisations of Algorithm 2 and/or reversing the positive/negative definitions of instances. It turns out that all settings lead to exactly the same partition. Therefore we believe that Algorithm 2 holds a global convergence property.

4. Experiments

To demonstrate the effectiveness of our OPT algorithm, we apply it to two public multibiometric databases, NIST-BSSR1 [23] and XM2VTS-benchmark [24]. Three sub-databases are extracted from NIST-BSSR1. These databases are summarised in Table 2.

Firstly in Section 4.1, we illustrate our OPT algorithm through visualising the results obtained from applying it to the NIST-face database. Then in

Table 2: Summary of multibiometric databases. N_+: the number of genuine samples; N_-: the number of imposter samples; FaM: face matchers; FiM: fingerprint matchers; SpM: speech matchers.

Database	Traits	K	N_+	N_-
NIST-multimodal	2 FaM; 2 Fim	4	517	266,772
NIST-face	2 FaM	2	6,000	17.994 million
NIST-fingerprint	2 FiM	2	6,000	35.994 million
XM2VTS	5 FaM; 3 SpM	8	1,000	151,800

Section 4.2, we present empirical studies of comparing OPT with many off-the-shelf, classifier-based and probability-based score fusion techniques.

4.1. Visualisation of OPT on the NIST-face database

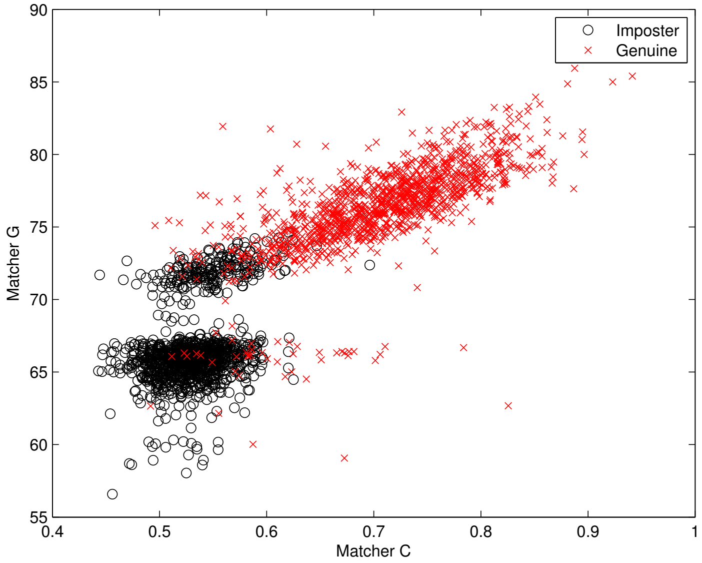


Figure 3: Scatter plot for the NIST-face database.

The NIST-face database contains the results of two face-verification algorithms as two source scores, namely matcher C and matcher G. The scatter plot of the samples is depicted in Fig. 3. The results of the OPT normalisation for the two matchers, i.e. the posterior probability functions $P_k(x)$

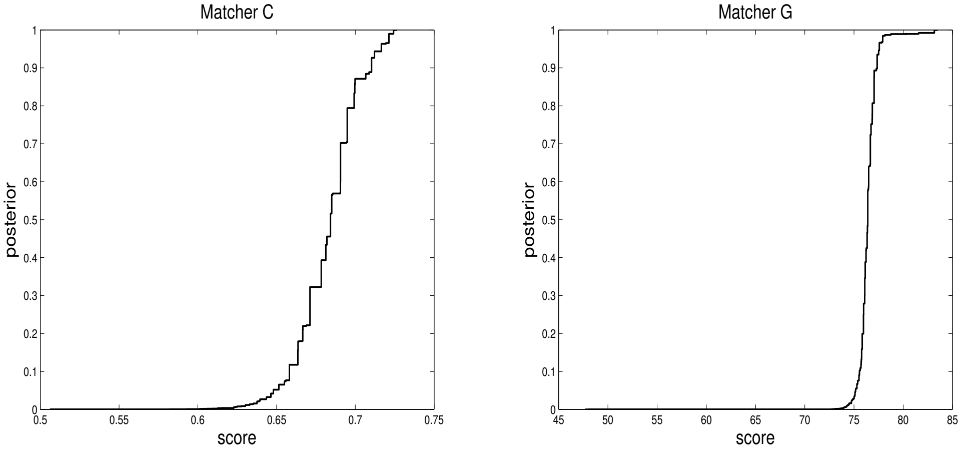


Figure 4: Posterior probabilities corresponding to the two source scores for the NIST-face database.

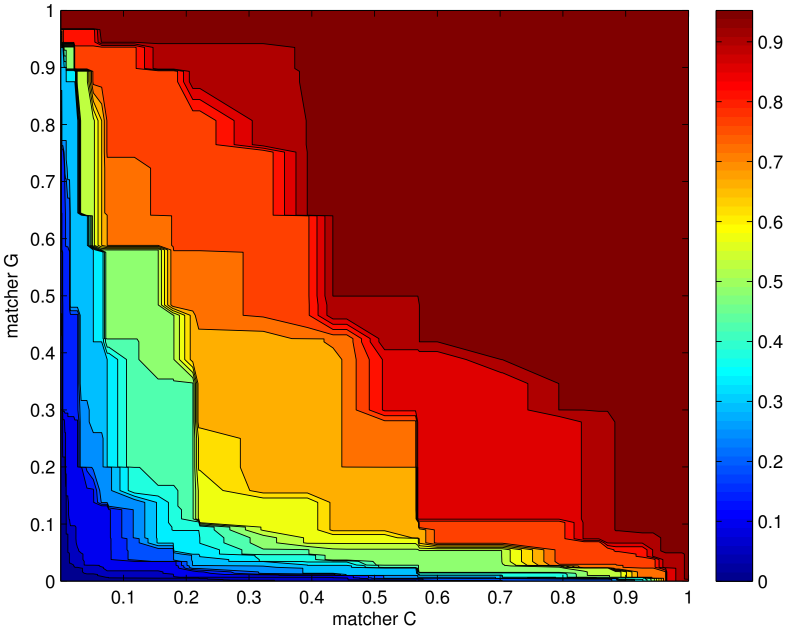


Figure 5: Merged posterior probabilities for the NIST-face database.

versus $S_k(x)$, are plotted in Fig. 4, and the result of the OPT merge, i.e. the matrix R, is shown in Fig. 5.

From Fig. 4, we can observe that the OPT normalisation results are similar to the tanh function, different from those obtained by linear normalisation techniques, such as the min-max and z-score algorithms. By inspecting Fig. 5, we can also find that the merged posterior probability function obtained by OPT is quite different from those obtained by some widely-used heuristic rules, such as the sum and product rules.

4.2. Comparative experimental results

For the experiments on the NIST-BSSR1 databases, we randomly choose half of the genuine and impostor samples to form the training set, and leave the other half samples for testing. This dataset split procedure is repeated for 100 times, and the average performances are recorded. For the experiments on the XM2VTS-benchmark database, the training-test split has already been predetermined by the database protocol, where the number of training (test) samples is 600 (400) for the genuine class and 40,000 (111,800) for the impostor class. To decide the values of the parameters of the other algorithms (such as the scale factor in tanh normalisation algorithm and the kernel's scale in the RBF-SVM algorithm), we simply scan the parameter space and choose the parameter values which lead to the best test performance. Compared with the standard procedure such as cross-validation, this procedure is much faster but the performance is an overoptimistic estimation.

Here we adopt the genuine accept rate (GAR) and the false accept rate (FAR) to evaluate the performance of score fusion algorithms. In particular, we compare the ROC curves (GAR versus FAR) of the algorithms, as well as their GAR when their FAR is at 0.01%. The results for the NIST-face database are shown in Fig. 6 and Table 3, for the NIST-fingerprint database in Fig. 7 and Table 4, for the NIST-multimodal database in Fig. 8 and Table 5, and for the XM2VTS database in Fig. 9 and Table 6, respectively.

From Fig. 6 we can find that, for the NIST-face database, the ROC curve of our OPT algorithm is well above the curves of two individual matchers, about 10% (4%) over that of matcher C and 20% (7%) over that of matcher G when FAR equals 0.001% (1%).

Moreover, we compare OPT with some off-the-shelf rules of normalisation, which includes the min-max, z-score, tanh and reduction of high-scores effect (RHE) normalisations with the sum rule [2], as well as with three probabilistic fusion algorithm (LR-based [1], IN and DN [17]) and a classification-based

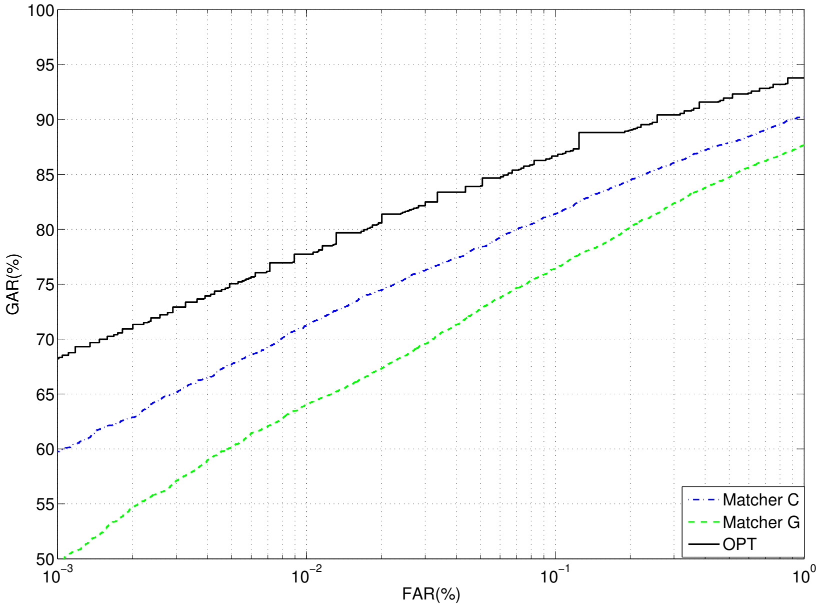


Figure 6: Performance gain obtained by OPT on the NIST-face database.

Table 3: Performance on the NIST-face database.

Method	GAR
min-max + sum [2]	77.2%
z-score + sum [2]	76.8%
tanh + sum [2]	74.6%
RHE [2]	77.5%
LR-based [1]	77.2%
IN [17]	75.1%
DN [17]	72.5%
minCq [15]	69.2%
OPT	77.8%

fusion algorithm (minCq [15]). The GAR of our OPT algorithm is superior to all of others, as listed in Table 3 with FAR equal to 0.01%.

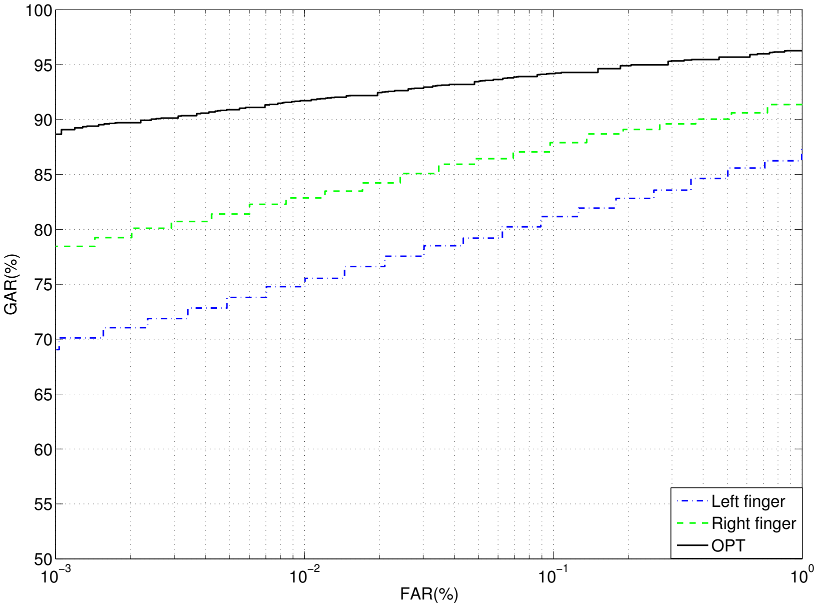


Figure 7: Performance gain obtained by OPT on the NIST-fingerprint database.

A similar pattern can be observed for the NIST-fingerprint database in Fig. 7 and Table 4, where the two individual matchers are the left-finger matcher and the right-finger matcher.

Different from the NIST-face and NIST-fingerprint databases, the NIST-multimodal database has four source scores (two for face and the other two for fingers). Hence there are 15 different ways of constructing the tree-structured ensemble merger. In our experiment, we trial all of them, and plot in Fig. 8 the ROC curve of an OPT merger with the median performance along with the curves of the four individual matchers (8(a)) and other established methods (8(b)), which include the heuristic rules of normalisation (min-max, z-score, tanh and RHE), three probability-based method (LR-based, IN and DN) and two classifier-based methods (RBF-SVM [8] and minCq). We also list in Table 5 the performances (in GAR) of the best and the worst OPT trees, as well as the average OPT performance, compared with the other methods.

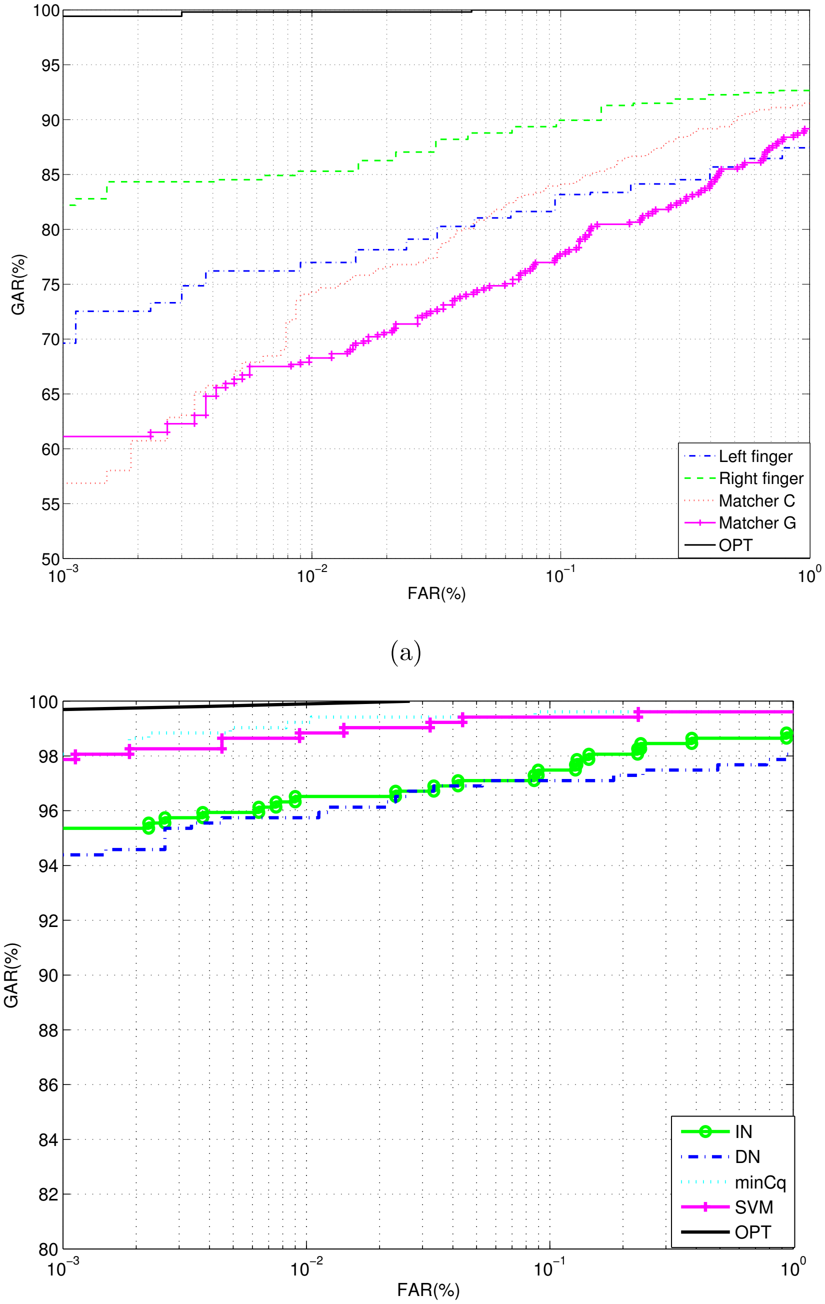
From Fig. 8 we can observe that the ROC curve of OPT is nearly perfect,

Table 4: Performance on the NIST-fingerprint database.

Method	GAR
min-max + sum [2]	91.0%
z-score + sum [2]	91.1%
tanh + sum [2]	90.3%
RHE [2]	91.2%
LR-based [1]	91.4%
IN [17]	91.3%
DN [17]	91.2%
minCq [15]	91.2%
OPT	91.8%

Table 5: Performance on the NIST-multimodal database.

Method	GAR
min-max + sum [2]	97.9%
z-score + sum [2]	98.2%
tanh + sum [2]	97.7%
RHE [2]	99.4%
LR-based [1]	99.1%
RBF-SVM [8]	98.8%
IN [17]	96.5%
DN [17]	95.7%
minCq [15]	99.2%
OPT(best tree)	99.9%
OPT(worst tree)	99.7%
OPT(average)	99.8%



(a)

(b)

Figure 8: (a) Performance gain obtained by OPT on the NIST-multimodal database. (b) Performance comparison with other methods on the NIST-multimodal database

significantly above the curves of individual matchers and also outperforms the other methods. We can also find from Table 5 that our OPT algorithm performs the best on average (99.84% to two decimal place accuracy), and we note that in fact 12 of the 15 trees achieve a performance higher than 99.80%.

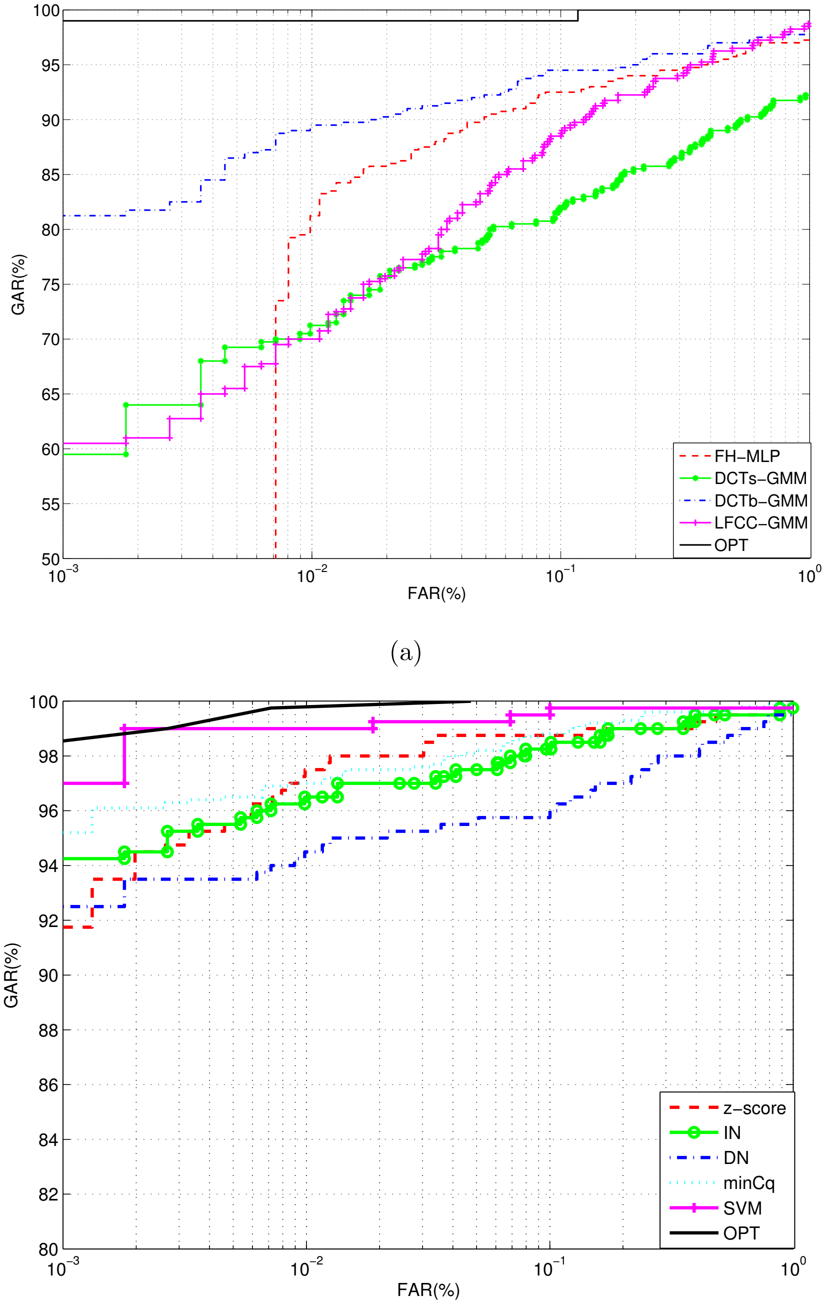
Table 6: Performance on the XM2VTS database.

Method	GAR
min-max + sum	96.50%
z-score + sum	97.25%
tanh + sum	96.50%
LR-based [1]	98.75%
RBF-SVM [8]	99.00%
IN [17]	96.50%
DN [17]	94.50%
minCq [15]	97.00%
OPT(best tree)	100.00%
OPT(worst tree)	97.75%
OPT(average)	99.35%

The XM2VTS database has eight source scores, hence there are 135,135 different ways to build the tree-structured merger. We trial all of these merger trees.

In Fig. 9, we can see that, similarly to Fig. 8, the ROC curve of the median OPT merger is greatly superior to the curves of the best four individual matchers and outperforms the other off-the-shelf and state-of-the-art methods.

In Fig. 10, we illustrate the histogram of GAR (%) constructed over all the OPT merger trees for the XM2VTS database when FAR is at 0.01%. From Fig. 10 and Table 6, we can observe that, as with Table 5, our OPT performs the best on average, and we note that 125,248 of 135,135 trees achieves the performance higher than or equal to 99.00%. In other words, when we randomly pick a tree structure, OPT will achieve the highest performance of other methods in comparison, at a probability of 92.7% for the XM2VTS database (and 80% for the NIST-multimodal database).



(a)

(b)

Figure 9: (a) Performance gain obtained by OPT fusion on the XM2VTS database. (b) Performance comparison with other methods on the XM2VTS database

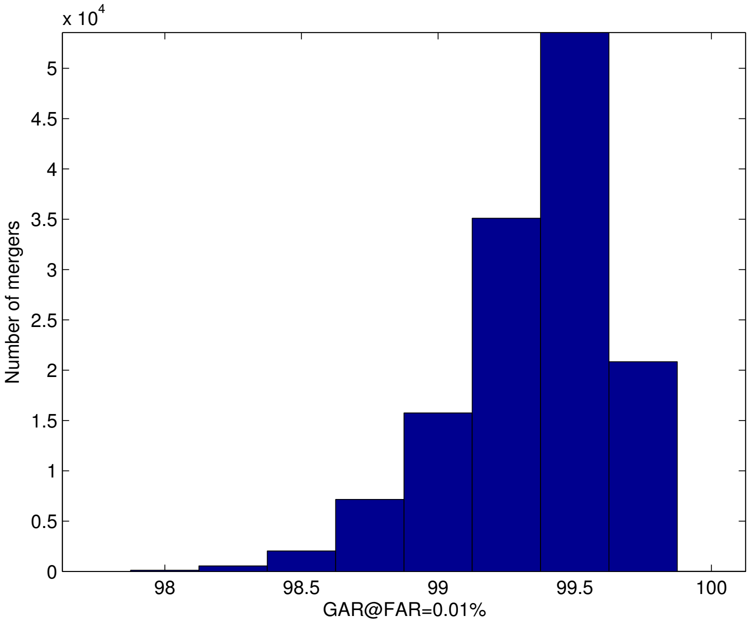


Figure 10: Histogram of GAR (%) of the OPT merger trees for the XM2VTS database when FAR is at 0.01%.

4.3. Discussion on the merging order of sources

We note that, as shown in Table 5, Table 6 and Fig. 10, for a database with more than two sources such as the NIST-multimodal database and the XM2VTS database, the merging order of sources will marginally impact the performance of OPT. Although it is possible to trial all potential merging orders for the NIST-multimodal and XM2VTS databases to find the best order, it becomes infeasible when the number of sources is sufficiently large. Therefore, for computational convenience, it is beneficial to pre-determine some favourable merging orders or ideally the optimal merging order.

To achieve this goal, we try to infer some properties of the merging order which might lead to good classification performance, by investigating the statistical difference between a group of some favourable merging orders and a group of some undesirable merging orders. Specifically, among all the 135,135 trees that merge the eight sources of the XM2VTS database, we compare the 20,860 trees whose performances are better than or equal to 99.75% and the 25,618 trees whose performance are worse than or equal to 99.00%, by counting the source pairs that they directly merged. We summarise the statistical results in Table 7.

Table 7: Source pairs used by good merging trees and bad merging trees. N_T: the number of the trees that directly merge the corresponding source pairs. Rk: the rank based on N_T.

Good merging trees			Bad merging trees		
Rk.	Source pair	N_T	Rk.	Source pair	N_T
1	FH-MLP & LFCC-GMM	3037	1	DCTs-GMM & LFCC-GMM	4354
2	DCTb-GMM & DCTs-MLP	2852	2	PAC-GMM & SSC-GMM	4228
3	DCTs-MLP & DCTb-MLP	2216	3	DCTs-MLP & LFCC-GMM	3463
⋮	⋮	⋮	⋮	⋮	⋮
26	DCTb-GMM & SSC-GMM	213	26	DCTb-GMM & DCTs-MLP	1218
27	PAC-GMM & SSC-GMM	151	27	DCTs-GMM & DCTs-MLP	1125
28	DCTs-GMM & SSC-GMM	30	28	FH-MLP & LFCC-GMM	69

From Table 7 we can observe that some patterns of how to choose source pairs to merge first indeed exist. For example, merging FH-MLP and LFCC-GMM will be beneficial, while merging PAC-GMM and SSC-GMM will be detrimental. However, as yet we have not managed to induce convincing statistical principles from the difference between these two groups of source pairs.

Hence, it remains an open problem to produce a (theoretically or statistically) convincing, reliable and non-exhaustive scheme to pre-determine an optimal merging order for multiple-source databases. Nevertheless, from Fig. 10 we can see that the variance of the performance of mergers of different merging orders is relatively small, and we can reach a classifier better than the state-of-the-art (98.75%) for the XM2VTS database with high probability. In practice, people may use a validation dataset and randomly try sufficiently many different merging orders and select the one with the best validating performance. Furthermore, people may design some heuristic but proper schemes, which can be based on the diversity of different biometric traits for example or their other prior knowledge about the relative importance and interaction of the traits, to attain a classifier with performance beyond the average over all merging orders.

5. Conclusion

We have proposed a novel probability-based score fusion algorithm, OPT, which is fully non-parametric. It treats both the score normalisation and the posterior probability merge as an constrained optimisation problem with only the natural order-preserving constraint. We have designed an effective algorithm to solve the optimisation problem and induced a tree-structured ensemble to bypass the curse of dimensionality. The effectiveness of our OPT algorithm has been demonstrated by experiments on both the NIST-BSSR1 and XM2VTS-benchmark databases.

Acknowledgment

This work was partially supported by the National Basic Research Program of China (973 program) under Grant No. 2013CB329403.

References

[1] K. Nandakumar, Y. Chen, S. C. Dass, A. K. Jain, Likelihood ratio-based biometric score fusion, Pattern Analysis and Machine Intelligence, IEEE Transactions on 30 (2) (2008) 342–347.

[2] M. He, S.-J. Horng, P. Fan, R.-S. Run, R.-J. Chen, J.-L. Lai, M. K. Khan, K. O. Sentosa, Performance evaluation of score level fusion in multimodal biometric systems, Pattern Recognition 43 (5) (2010) 1789–1800.

[3] M. Hanmandlu, J. Grover, A. Gureja, H. Gupta, Score level fusion of multimodal biometrics using triangular norms, Pattern Recognition Letters 32 (14) (2011) 1843–1850.

[4] N. Ueda, Optimal linear combination of neural networks for improving classification performance, Pattern Analysis and Machine Intelligence, IEEE Transactions on 22 (2) (2000) 207–215.

[5] A. Demiriz, K. P. Bennett, J. Shawe-Taylor, Linear programming boosting via column generation, Machine Learning 46 (1-3) (2002) 225–254.

[6] P. Gehler, S. Nowozin, On feature combination for multiclass object classification, in: Computer Vision (ICCV), IEEE International Conference on, IEEE, 2009, pp. 221–228.

[7] K.-A. Toh, W.-Y. Yau, X. Jiang, A reduced multivariate polynomial model for multimodal biometrics and classifiers fusion, Circuits and Systems for Video Technology, IEEE Transactions on 14 (2) (2004) 224–233.

[8] C. Bergamini, L. S. Oliveira, A. L. Koerich, R. Sabourin, Combining different biometric traits with one-class classification, Signal Processing 89 (11) (2009) 2117–2127.

[9] Y. Kim, K.-A. Toh, A. B. J. Teoh, H.-L. Eng, W.-Y. Yau, An online learning network for biometric scores fusion, Neurocomputing 102 (2013) 65–77.

[10] Y. Yang, J. Song, Z. Huang, Z. Ma, N. Sebe, A. G. Hauptmann, Multi-feature fusion via hierarchical regression for multimedia analysis, Multimedia, IEEE Transactions on 15 (3) (2013) 572–581.

[11] D. Liu, K.-T. Lai, G. Ye, M.-S. Chen, S.-F. Chang, Sample-specific late fusion for visual category recognition, in: Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, IEEE, 2013, pp. 803–810.

[12] J. Liu, S. McCloskey, Y. Liu, Local expert forest of score fusion for video event classification, in: Computer Vision–ECCV 2012, Springer, 2012, pp. 397–410.

[13] Z. Xu, Y. Yang, I. Tsang, N. Sebe, A. G. Hauptmann, Feature weighting via optimal thresholding for video analysis, in: Computer Vision (ICCV), IEEE International Conference on, IEEE, 2013, pp. 3440–3447.

[14] F. Laviolette, M. Marchand, J.-F. Roy, From PAC-Bayes bounds to quadratic programs for majority votes, in: Proceedings of the 28th International Conference on Machine Learning (ICML), 2011, pp. 649–656.

[15] E. Morvant, A. Habrard, S. Ayache, Majority vote of diverse classifiers for late fusion, arXiv preprint arXiv:1404.7796.

[16] J. Kittler, M. Hatef, R. P. Duin, J. Matas, On combining classifiers, Pattern Analysis and Machine Intelligence, IEEE Transactions on 20 (3) (1998) 226–239.

[17] O. R. Terrades, E. Valveny, S. Tabbone, Optimal classifier fusion in a non-Bayesian probabilistic framework, Pattern Analysis and Machine Intelligence, IEEE Transactions on 31 (9) (2009) 1630–1644.

[18] S. Prabhakar, A. K. Jain, Decision-level fusion in fingerprint verification, Pattern Recognition 35 (4) (2002) 861–874.

[19] A. J. Ma, P. C. Yuen, J.-H. Lai, Linear dependency modeling for classifier fusion and feature combination, Pattern Analysis and Machine Intelligence, IEEE Transactions on 35 (5) (2013) 1135–1148.

[20] A. J. Ma, P. C. Yuen, Reduced analytic dependency modeling: Robust fusion for visual recognition, International Journal of Computer Vision 109 (3) (2014) 233–251.

[21] M. S. Cheema, A. Eweiwi, C. Bauckhage, A stochastic late fusion approach to human action recognition in unconstrained images and videos, in: Pattern Recognition, Springer, 2014, pp. 616–628.

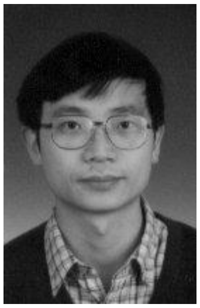
[22] A. Jain, K. Nandakumar, A. Ross, Score normalization in multimodal biometric systems, Pattern recognition 38 (12) (2005) 2270–2285.

[23] National Institute of Standards and Technology, NIST Biometric Scores Set – Release 1, `http://www.itl.nist.gov/iad/894.03/biometricscores` (2004).

[24] N. Poh, S. Bengio, Database, protocols and tools for evaluating score-level fusion algorithms in biometric authentication, Pattern Recognition 39 (2) (2006) 223–233.

Yicong Liang was born in Nanjing, China, in 1986. He received the B.S. degree from Tsinghua University, Beijing, China, in 2008. He is currently working toward the Ph.D. degree with the Department of Electronic Engineering in Tsinghua University, Beijing, China. His research interests include pattern recognition, machine learning along with their applications in face recognition and face verification.

Xiaoqing Ding received the B.E. degree from Tsinghua University, Beijing, China, in 1962. She is currently a Professor and a Ph.D. Supervisor with the Department of Electronic Engineering, Tsinghua University. She has published more than 550 papers, and is the holder of 25 patents. Her research interests include pattern recognition, image processing, character recognition, biometric identification, computer vision and video surveillance, etc. Prof. Ding is an IEEE Fellow and IAPR Fellow. She was a recipient of a series of achievements on Chinese/multilanguage character recognition, face recognition, etc. She was a recipient of the most prestigious National Scientific and Technical Progress Awards in China in 1992, 1998, 2003, and 2008.



Changsong Liu is an Associate Professor in the Department of Electronic engineering at Tsinghua University. He received the B.S. degree in both mechanics engineering and electronic engineering in 1992, the Master degree in electronic engineering in 1995, the Ph.D. degree in 2007 from Tsinghua university, China. His fields of interests include image processing, pattern recognition, and nature language processing. He has published more than 80 papers.

Jing-Hao Xue received the B.Eng. degree in telecommunication and information systems in 1993 and the Dr.Eng. degree in signal and information processing in 1998, both from Tsinghua University, the M.Sc. degree in medical imaging and the M.Sc. degree in statistics, both from Katholieke Universiteit Leuven in 2004, and the degree of Ph.D. in statistics from the University of Glasgow in 2008. Since 2008, he has worked in the Department of Statistical Science at University College London as a Lecturer and Senior Lecturer. His current research interests include statistical classification, high-dimensional data analysis, computer vision and pattern recognition.