Combining Biometric Scores in Identification Systems

Sergey Tulyakov and Venu Govindaraju, *Fellow, IEEE*

Both authors are with the State University of New York at Buffalo.

Abstract

Combination approaches in biometric identification systems usually consider only the matching scores related to a single person in order to derive a combined score for that person. We present the use of all scores received by all persons and explore the advantages of such an approach when enough training data is available. More fundamentally, we identify four types of classifier combinations determined by the numbers of trained combining functions and their input parameters. We prove that the improper choice of the combination type might result in only suboptimal performance of identification system. We investigate combinations, which consider all available matching scores and have only single trainable combination function. We introduce a particular kind of such combinations utilizing identification models, which account for dependencies between scores output by any one classifier. We present several experiments validating the advantage of our proposed combination algorithms for problems dealing with large number of classes, in particular, biometric person identification systems.

Index Terms

Combination of classifiers, biometric identification systems.

I. Introduction

Biometric applications can be operated in two modes: verification (1:1) mode and identification (1:N) mode. Common approaches to combining biometrics for (1:N) identification applications are actually a simple iterative use of the (1:1) verification system. The combined score for matching a set of biometrics to a particular enrolled person is usually obtained as a function of the matching scores of all biometrics for the particular person in either modes of operation. However, identification systems possess additional information that can be utilized for deriving the final score for a particular enrolled person. This additional information is available from the matching scores returned for other enrollees in the database.

In this paper we consider a combination of matchers in the identification system. In such system M multiple biometric matchers are used to produce MN matching scores, where N is the number of enrolled persons. We assume that M is small and N is large. Each biometric matcher in such setting is equivalent to the classifier assigning matching scores to each of N classes or persons. And the combination of biometric matchers in identification system can be viewed as a classifier combination problem with a large number of classes. We use terms

’matcher’ to refer to a classifier which outputs class matching scores, and ’identification’ to refer to the case dealing with large number of classes.

We assume the combination algorithm is producing a combined score for each class, and final matched class corresponds to the best combined score (see Figure 1). The combined score is determined by the combination function f which takes as parameters potentially whole set of match scores. Our categorization of the combination algorithm is determined by the construction properties of combined functions. In particular, combination algorithm can have only one combination function, and combined scores for different classes can be obtained by the permutation of input match scores. On the other hand, each class can have its own combination function, and combined scores are calculated differently for different classes. We might call the combination algorithm of first kind as class generic, and combination algorithms of the second kind as class specific. Another distinction between combination algorithms is based on the number of input parameters to each combination function. Some combination functions take as parameters only M match scores related to a particular class to calculate the combined score of this class. We call the combination algorithms with such combination functions as reduced parameter set combination algorithms, or local combination algorithms. Other types of combination functions might consider the whole set of MN match score to derive a combined score for one class. The combination algorithms with such functions could be called whole parameter set, or global combination algorithms.

If classifiers deal with a small number of classes, then the dependencies between scores assigned to different classes can be learned and used for combination purposes. For example, Xu et al. [1] used class confusion matrices for deriving belief values and integrated these values into combination algorithms in the digit classification problem. This algorithm has class specific and global combination functions. This is most general type of combinations and ideally we would use it for other problems. But learning class dependencies requires significant number of training samples of each class. Such data is not available for identification problems, where usually a single template is enrolled for each person. In addition, the database of enrolled persons can be frequently changed, and this makes learning class relationships infeasible.

As a consequence, combination approaches in identification systems usually consider only matching scores related to the single person in order to derive a combined score for that person. Though all MN matching scores are available, only M scores are used for combination. In this paper we investigate the question of whether it is possible to improve the performance of the identification system by using a whole set of scores for deriving combined matching score of each person. We argue that combination algorithms using class generic and global combination functions are well suited for combination problems in identification systems.

Some results of this work were presented in [2] and [3]. In this paper we provide additional insights into combination problems in identification systems, construct new example proving the difference between local and global combination types, and present a new global weighted sum combination algorithm. In addition, this paper contains the results of new experiments on bigger datasets and comparison of proposed 'second best score' identification model with identification model resulting from T-normalization[4].

In the next section we present the four types of combinations depending on the number of matching scores they consider and on the number of trainable combination functions. Then we give a review of the identification model concept. The model essentially keeps an information about dependencies between matching scores assigned by a single matcher to all enrolled persons. Using such models allows us to construct class generic and global combination functions. In the last sections we investigate the combinations in identification systems in detail, and show both theoretically (using example) and experimentally the advantages of combinations using identification models.

II. Complexity Types of Classifier Combinations

The general scheme for classifier combination is shown in figure 1. The final score for a class is derived from the scores received from all the classifiers for that class. The combination functions of such combination algorithms have reduced parameter set, and many well known combination methods (e.g. weighted sum of scores) fall into this category. It is also possible to consider a more general form of combination where derivation of a final score for a particular class includes all classifier scores, for that class as well as for other classes [1], [5]. For example, if the combination algorithm in this figure consisted of fully connected artificial neural network accepting MN input parameters and having N output parameters, it would present an example of most general, class specific and global combination function algorithm. The disadvantage of this more general approach is that it requires larger amount of training data, which might not be available in identification systems. This section describes this and another three types of combinations which

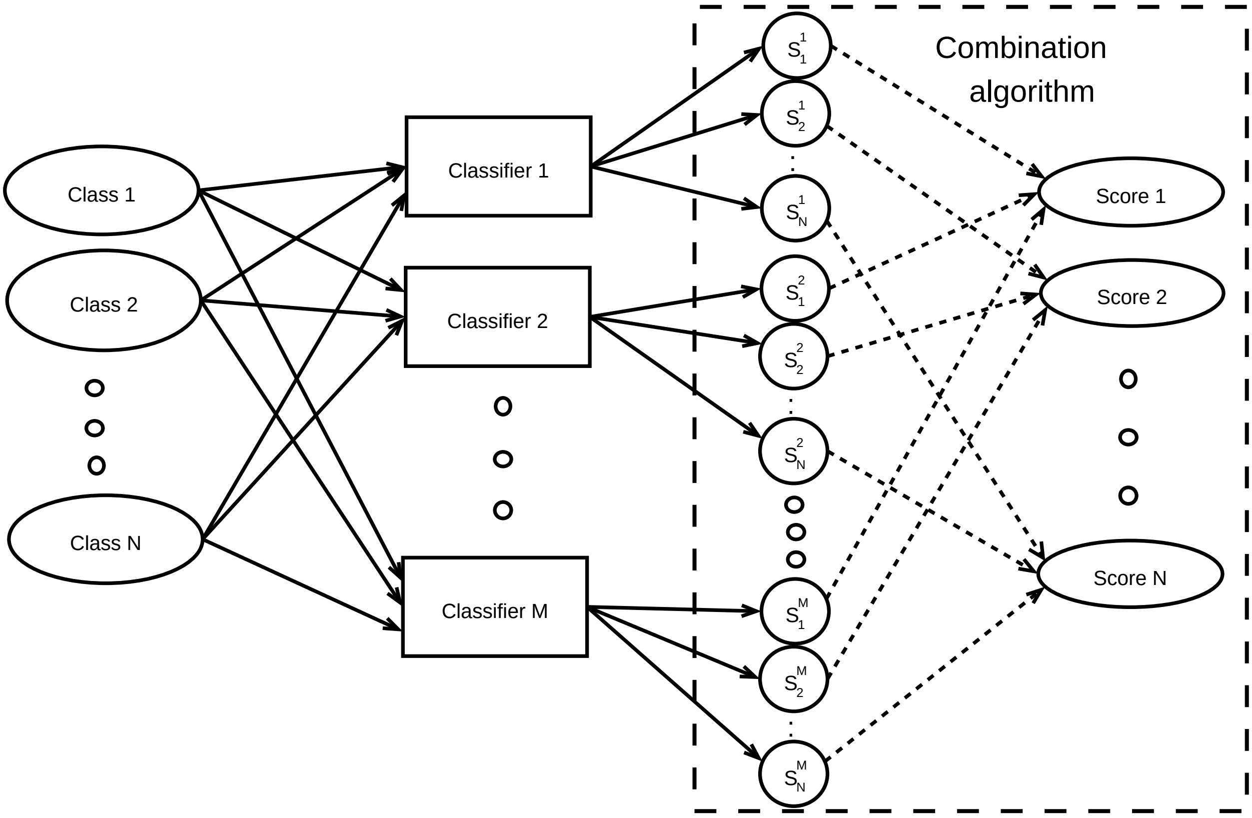


Fig. 1. Classifier combination takes a set of s_i^j - score for class i by classifier j and produces combination scores S_i for each class i.

might need smaller sets of training data. Ultimately, the problem characteristics and the size of training set would determine the most appropriate combination type for a particular problem.

A. Complexity Based Combination Types

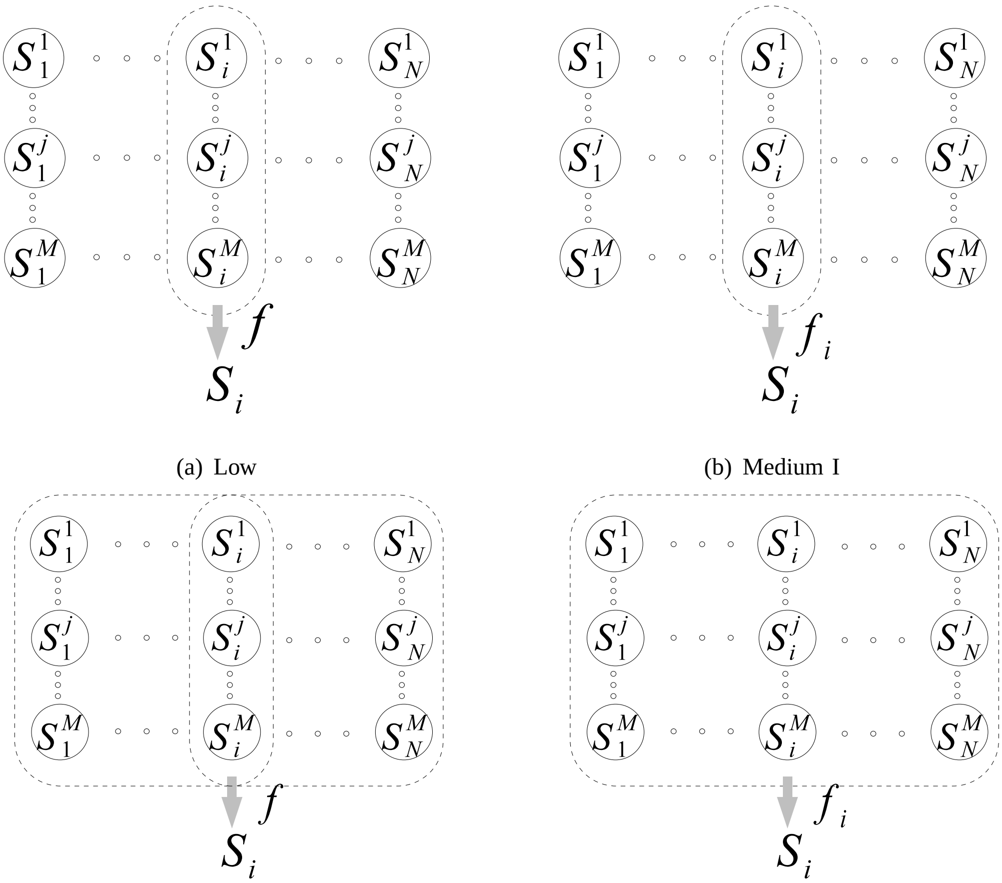
Combination algorithms can be separated into 4 different types depending on the number of classifier's scores they take into account and the number of combination functions required to be trained. As in Figure 1 i is the index for the N classes and j is the index for the M classifiers.

1) Low complexity combinations: $S_i = f(\{s_i^j\}_{j=1,\dots,M})$. Combinations of this type require only one combination function to be trained, and the combination function takes as input scores for one particular class as parameters. These methods use class generic and reduced parameter set (local) combination functions.

2) Medium complexity I combinations: $S_i = f_i(\{s_i^j\}_{j=1,\dots,M})$. Combinations of this type have separate score combining functions for each class and each such function takes as input parameters only the scores related to its class. These methods use class specific and reduced parameter set (local) combination functions.
3) Medium complexity II combinations: $S_i = f(\{s_i^j\}_{j=1,\dots,M}, \{s_k^j\}_{j=1,\dots,M;k=1,\dots,N,k\neq i})$. This combination function takes as parameters not only the scores related to this class, but all output scores of classifiers. Combination scores for each class are calculated using the same function, but scores for class i are given a special place as parameters. Applying function f for different classes effectively means permutation of the function's parameters. These combination functions are class generic and use whole parameter set (global).
4) High complexity combinations: $S_i = f_i(\{s_k^j\}_{j=1,\dots,M;k=1,\dots,N})$. Functions calculating final scores are different for all classes, and they take as parameters all output base classifier scores. These are class specific and whole parameter set (global) combination functions.

In order to illustrate the different combination types we can use a matrix score representation. Each row corresponds to a set of scores output by a particular classifier, and each column has scores assigned by classifiers to a particular class. The illustration of each combination type functions is given in Figure 2. In order to produce the combined score S_i for class i low complexity combinations (a) and medium I complexity (b) combinations consider only classifier scores assigned to class i (column i), and use local combination functions. Medium II (c) and high complexity (d) combinations consider all scores output by classifiers for calculating a combined score S_i for class i, and their combination functions are global.

Low (a) and medium II (c) complexity combinations have the same class generic combination functions f irrespective of the class for which the score is calculated. Note that medium II complexity type combinations have scores related to a particular class in a special consideration as indicated by the second ellipse around these scores. We can think of these combinations as taking two sets of parameters - scores for a particular class, and all other scores. The important property is that combination function f is same for all classes, but the combined scores S_i differ, since we effectively permute function inputs for different classes. Medium I (b) and high (d) complexity combinations have class specific combining functions f_i trained differently for different classes.



(a) Low (b) Medium I

(c) Medium II (d) High

Fig. 2. The range of scores considered by each combination type and combination functions.

It is interesting to compare our combinations types with previous categorization of combination methods by Kuncheva et al.[6]. In that work the score matrix has names ’decision profile’ and ’intermediate feature space’. It seems that using term ’score space’ makes more sense here. Kuncheva’s work also separates combinations into ’class-conscious’ set which corresponds to the union of ’low’ and ’medium I’ complexity types, and ’class-indifferent’ set which corresponds to the union of ’medium II’ and ’high’ complexity types. Again these terms might not be suitable since we can think of a combination method as being ’class-conscious’ if each class has its own combination function (class specific ’medium I’ and ’high’ complexity types), and ’class-indifferent’ if combination functions are same for all classes (class generic ’low’ and ’medium II’ complexity types). The continuation of this work [7] gave an example of the weighted sum rule having three different numbers of trainable parameters (and accepting different numbers of input scores), which correspond to 'low', 'medium I' and 'high' complexity types.

In contrast to Kuncheva's work, our categorization of combination methods is more general since we are not limiting ourselves to simple combination rules like weighted sum rule. Also we consider an additional category of 'medium II' type, which is missed there. An example of 'medium II' combinations are two step combination algorithms where in the first step the scores produced by a particular classifier are normalized (with possible participation of all scores of this classifier), and in the second step scores are combined by a function from 'low' complexity type. Thus scores in each row are combined first, and then the results are combined columnwise in the second step. Note, that it is still possible to have weighted sum combination method of medium II complexity type, and we give an example of such combination later.

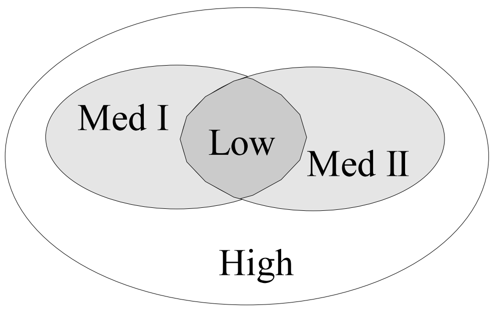


Fig. 3. The relationship diagram of different combination complexity types.

Figure 3 illustrates the relationships between presented complexity types of combinations. Medium complexity types are subsets of high complexity combinations, and the set of low complexity combinations is exactly the intersection of sets of medium I and medium II combination types. In order to avoid a confusion in terminology we will henceforth assume that a combination method belongs to a particular type only if it belongs to this type and does not belong to the more specific type.

In [8] we provided a stricter description of these complexity types using the concept of VC (Vapnik-Chervonenkis) dimension [9]. In particular, we derived the formulas for VC dimensions for each complexity type, and showed how complexity can be reduced either by adopting a lower complexity type combination, or by restricting the set of trainable combination functions.

The ability to use VC dimension for characterization of different combination types justifies our usage of term ’complexity types’.

Higher complexity combinations can potentially produce better classification results since more information is used. On the other hand the availability of training samples will limit the types of possible combinations. Thus the choice of combination type in any particular application is a trade-off between classifying capabilities of combination functions and the availability of sufficient training samples. In practice,we first see if a particular classifier combination problem can be solved with high complexity combinations as a most general combination type. If complexity is too big for the available training data size, number of classes N and the complexities of chosen combination functions, we consider lower complexity combinations. When the complexity is lowered it is important to see if any useful information is lost. If such loss happens, the combination algorithm should be modified to compensate for it.

Different generic classifiers such as neural networks, decision trees, etc., can be used for classifier combinations within each complexity class. From the perspective of this framework, the main effort in solving classifier combination problem consists in a justification for a particular chosen complexity type of combination and providing any special modifications to generic classifiers compensating for this chosen complexity type. As an example, the biometric person authentication systems we experimented with in this paper have a high number of enrolled classes (persons) N and a small number of classifiers (biometric face and fingerprint matchers) M. As a result medium I and high complexity combinations would have high complexity (VC dimension), and we will have problems training them. On the other hand low and medium II type combinations would have lower complexity (depending on the complexity of the set of trainable combination functions f), and we would be able to train them.

Most combinations algorithms in biometric applications are of low complexity type. Complexity framework suggests that it is possible to employ medium II combinations as well in these applications. In this work we are interested in developing such combinations. The main results presented here are the following. First, we prove that medium II complexity type is indeed different from low complexity type and low complexity combinations might have only suboptimal performance. Second, we investigate the reasons for this difference and suggest construction of so called identification model, which contains information available to medium II type combinations and not available to low complexity type combinations. Third, we derive combinations rules of medium II complexity type which are analogous to the traditional likelihood ratio and weighted sum combinations of low complexity type. Finally, the experiments on large biometric score sets confirm that suggested medium II complexity combinations have better performance than their counterparts of low complexity.

B. Verification and Identification Operating Modes

By our convention an identification system provides matching scores for N enrolled persons. We define an identification system as operating in identification mode if its purpose is to classify an input as belonging to any of N classes or persons. We assume that the classification decision is performed by applying $\arg\max$ operator to the N combined scores:

$$C = \arg\max_{1 \leq i \leq N} S_i$$

The correct identification rate, that is the frequency of correctly finding the true class of the input, is the natural measure of performance in this case, and we will use it in our experiments. Note, that there could be other performance measures for identification mode operation, such as Rank Probability Mass, Cumulative Match Curve, etc.[10], but we will not use them here.

Conversely, we define an identification system as operating in verification mode if its purpose is to decide whether an input belongs to some claimed person identity. In this case we can distinguish two classes: genuine and impostor verification attempts. The decision to accept is based on comparing a combined score of a claimed person identity i, S_i, to some threshold θ: $S_i > \theta$. The common way to describe the system performance in such two-class problems is to construct graphs showing the dependencies of errors on threshold θ: ROC curve, DET curve, etc.[10]. In our experiments we will use ROC curves.

Both identification and verification modes of operation can utilize combinations of all four complexity types described above. Typically, biometric systems operating in verification mode produce matching scores only for the claimed person identity, and not for other enrolled persons. The combinations using only such restricted sets of scores are necessarily of low or medium I complexity types. If we want to use combinations of other complexity types we have to additionally produce matching scores for other enrolled persons as well.

III. Previous Work in Identification System Combinations

If we have a combination algorithm for verification system, we can use it sequentially for all persons in identification system. Such algorithm will not utilize dependencies between scores output by a single matcher. Most of combination algorithms used in biometric applications are of this type. In our combination framework such combinations are of the low complexity type. Combination functions can also be user specific - f_i [11], [12] (medium I complexity type). Below we present approaches which do utilize score dependencies in identification trials: rank based combinations and some score normalization techniques.

A. Rank Based Combinations

The frequent approach to combination in identification systems is to use rank information of the scores. This approach transforms combination problems with measurement level output classifiers to combination problems with ranking level output classifiers ([1]). T.K. Ho has described classifier combinations on the ranks of the scores instead of scores themselves by arguing that ranks provide more reliable information about class being genuine [13], [14]. Thus, if the input image has low quality, then the genuine score, as well as the impostor scores will be low. Combining low score for genuine class with other scores could confuse a combination algorithm, but the rank of the genuine class remains to be a good statistic, and combining this rank with other ranks of this genuine class should result in true classification. Brunelli and Falavigna [15] considered a hybrid approach where traditional combination of matching scores is fused with rank information in order to achieve identification decision. Hong and Jain [16] consider ranks, not for combination, but for modeling or normalizing classifier output score. Behavior-Knowledge Space combination methods [17] are also based on ranks. Saranli and Demirekler [18] provide additional references for rank based combination and a theoretical approach to such combinations.

Rank-based methods do utilize the score dependencies in identification trials, and, as many authors suggest, these methods provide a better performance in identification systems. The problem with rank based methods, however, is that the score information is somewhat lost. Indeed, genuine score can be much better than second best score, or it could be only slightly better, but score ranks do not reflect this difference. It would be desirable to have a combination method which retains the score information as well as the rank information.

B. Score normalization approaches

Usually score normalization [19] means transformation of scores based on the classifier's score model learned during training, and each score is transformed individually using such a model. Thus the other scores output by a matcher during the same identification trial are not taken into consideration. If these normalized scores are later combined class-wise, then score dependence will not be accounted for by the combination algorithm.

Some score normalization techniques can use a set of identification trial scores output by classifier. For example, Kittler et al. [20] normalize each score by the sum of all other scores before combination. Similar normalization techniques are Z(zero)- and T(test)- normalizations [4], [21]. Z- normalization uses impostor matching scores to produce a class specific normalization. Z-normalization does not include the set of identification trial scores, and thus does not utilize score dependency. On the other hand, T-normalization does use a set scores produced during single identification trial, and can be considered as a simple form of identification model. T-normalization uses statistics of mean and variance of identification score set. Note that identification model implies some learning algorithm, but T-normalization is a predetermined routine with no training. Still, using this simple kind of score modeling turns out to be quite useful; for example, [22] argued for applying T-normalizations in face verification competition. There is also an argument[23] that useful classification information gets lost during such normalizations.

Score normalization techniques have been well developed in the speaker identification problem. Cohort normalizing method [24], [25] considers a subset of enrolled persons close to the current test person in order to normalize the score for that person by a log-likelihood ratio of genuine (current person) and impostor (cohort) score density models. Auckenthaler et al.[4] separated cohort normalization methods into cohorts found during testing (constrained) and cohorts dynamically formed during testing (unconstrained cohorts). Normalization by constrained cohorts utilizes only one matching score of each classifier and thus does not consider score dependencies. On the other hand, normalization by unconstrained cohorts potentially uses all scores of classifiers, and thus results in the construction of the identification model.

IV. Identification Model

Before describing in detail our combination method, we will give an overview of our previous investigation into performance characteristics of identification systems. It turns out that the traditional ways of describing the performance of verification systems - densities of the genuine and impostor scores, as well as ROC curve constructed with the help of these densities, do not fully represent the performance of identification systems. Densities of genuine and impostor scores disregard the fact that the scores produced by a single classifier and assigned to different classes are usually dependent. Thus, full description of the identification system performance requires reconstruction of the joint density of all scores related to different classes.

Since the number of classes in identification systems can be very large or variable, such reconstruction of joint density might not be possible. Thus we introduced [26] a concept of identification model - a model that represents identification system properties and performance. The identification system should be able to adequately represent the score distributions, and, in particular, help us to derive the mapping of scores into posterior class probabilities. Generally, we expect that any algorithm which works with classifier's scores (such as decision thresholding or classifier combination) should perform better if scores are remapped using identification model. Our previous research was using identification model for decision making, and in this section we summarize the results of this research.

A. Performance of identification systems

Suppose we have one matcher in the identification system with N classes. Let $s_1 > s_2 > \cdots > s_N$ be a set of matching scores we got in one identification attempt (bigger score means better match, and for this example we index scores by their rank rather than by class). How should we decide if the class corresponding to the best score s_1 is the true class of the input? One solution is to compare best score s_1 to some threshold θ and if $s_1 > \theta$ confirm identification success and accept class corresponding to s_1 as truth. But a little thought reveals that if the second-best score s_2 is close to the best score s_1 then there is big chance that class corresponding to s_2 might be a true class instead of class corresponding to s_1. Thus we definitely should include considering second-best score s_2 into our decision about accepting identification results. Similarly, considering third-best score s_3 or other scores might be beneficial as well.

In [26] and in [27] we investigated the benefits of utilizing the second best matching score for accepting identification results. The main results of those works:

- The performance improvements due to utilizing second best score for identification system decisions arise naturally, even if scores are statistically independent.

- The improvements can be bigger if matching scores are dependent.
- Real-life identification systems usually have dependent scores, that contributes favorably to identification system performance.

The following section considers an example of utilizing second best score s_2 to make a decision on accepting a person corresponding to the best matching score in biometric identification systems.

B. Identification Model for Acceptance Decision in Biometric Identification Systems

We consider the problem of person identification by means of fingerprint and face biometrics. We use the NIST Biometric Score Set, release 1 (BSSR1 [28]). We consider three subsets of scores: fingerprint li set which is produced for 6000 enrolled left index fingerprints and 6000 user input fingerprints, face recognizer C set and face recognizer G set which are produced for 3000 enrolled faces and 6000 user input faces. Thus all sets have 6000 identification trials with 1 genuine scores and 5999 (for fingerprints) or 2999 (for faces) impostor match scores.

We ran all experiments using leave-one-out method, that is for each identification trial we use all other 5999 trials for training and perform testing on the left out trial. All 6000 test runs are combined together in the ROC curve. For each test run we reconstruct densities of scores for two classes - genuine identification trials with the best score being the genuine match and impostor identification trials with the best score being impostor match. A test run score is given as a ratio of genuine density to the impostor density at a test trial score point (Bayesian classification).

Results of the experiments are shown in Figures 4 - 6. 1 top score thresholding means that we consider only the top scores for reconstructing the densities, and 2 top score thresholding means that we use both the best and second-best scores to reconstruct densities. In all cases making acceptance decision based on two scores has clear advantage over decisions based on just the first score.

The application of the identification model clearly improves the identification system performance. The biometric matching scores are dependent (see below Sec. VI), and this dependence is mainly caused by the quality of matched templates. For example, if an input fingerprint has small area, then it will contain small number of minutia, and consequently all minutia based matchers will produce low scores. And if the input fingerprint has many minutia, then not only the genuine match will have many matching minutia and high score, but impostor matches will

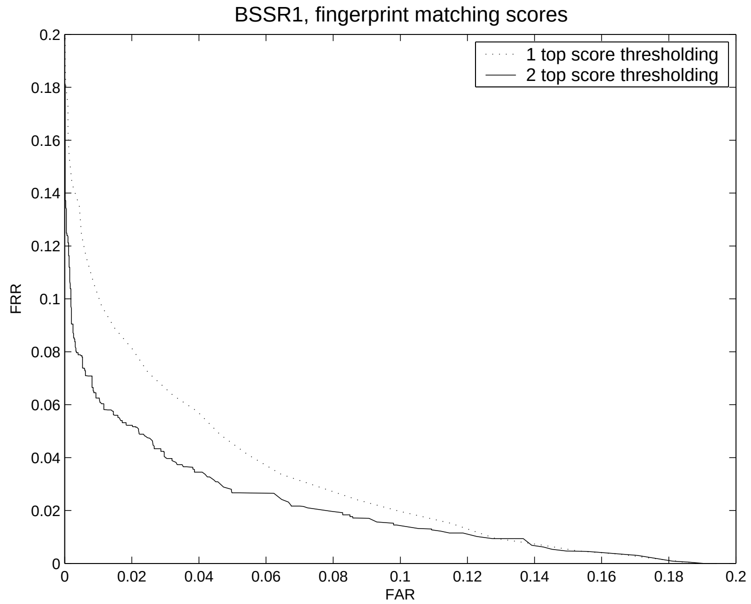


Fig. 4. ROC curves for optimal thresholding using and not using second-best score. BSSR1 set, fingerprint li scores.

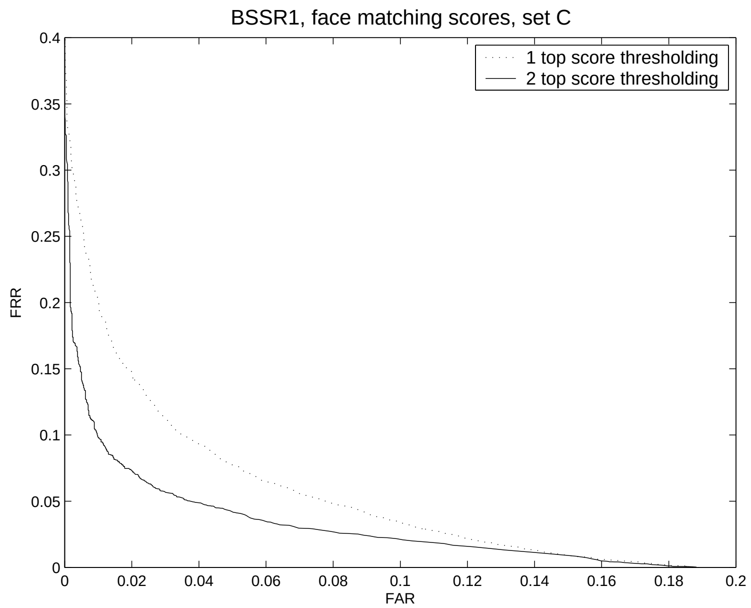


Fig. 5. ROC curves for optimal thresholding using and not using second-best score. BSSR1 set, face recognizer C scores.

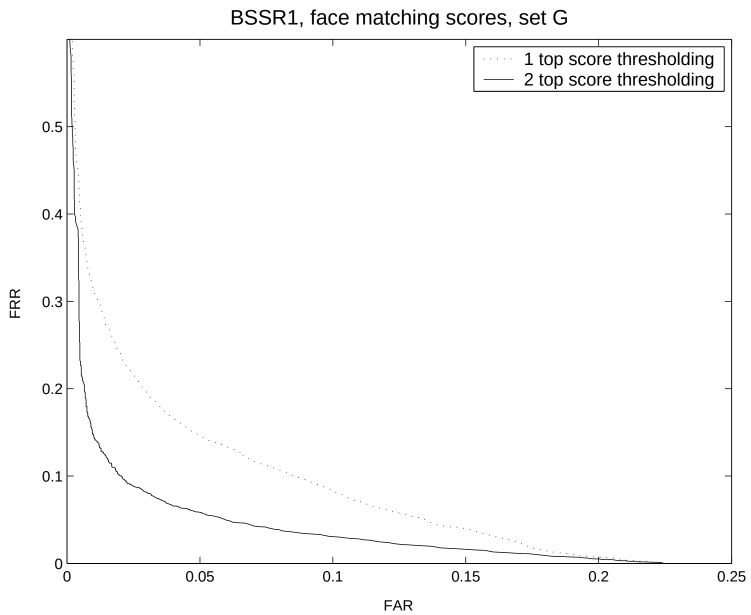


Fig. 6. ROC curves for optimal thresholding using and not using second-best score. BSSR1 set, face recognizer G scores.

also get a chance to produce a higher score. The following section gives an example clarifying the connection between matching score dependence and the benefits of utilizing second best score.

C. Example of dependent scores

Suppose we have an identification system with one matcher and, for simplicity, $N = 2$ classes. Also suppose we collected a data on the distributions of genuine and impostor scores and reconstructed score densities as shown in Figure 7.

Consider three possible scenarios on how these densities might have originated from the sample of the identification attempts:

1) In every observed identification attempt the score of the genuine class s_{gen} is chosen from the distribution of genuine scores, and the score of the impostor class s_{imp} is the additive inverse of s_{gen}: $s_{imp} = 1 - s_{gen}$. As we proved in [8], using second best score in addition to best score has no benefit for identification system performance in this case.
2) Both scores s_{gen} and s_{imp} are sampled independently from genuine and impostor distributions. The experiments with artificial densities showed[26] that utilizing the second best

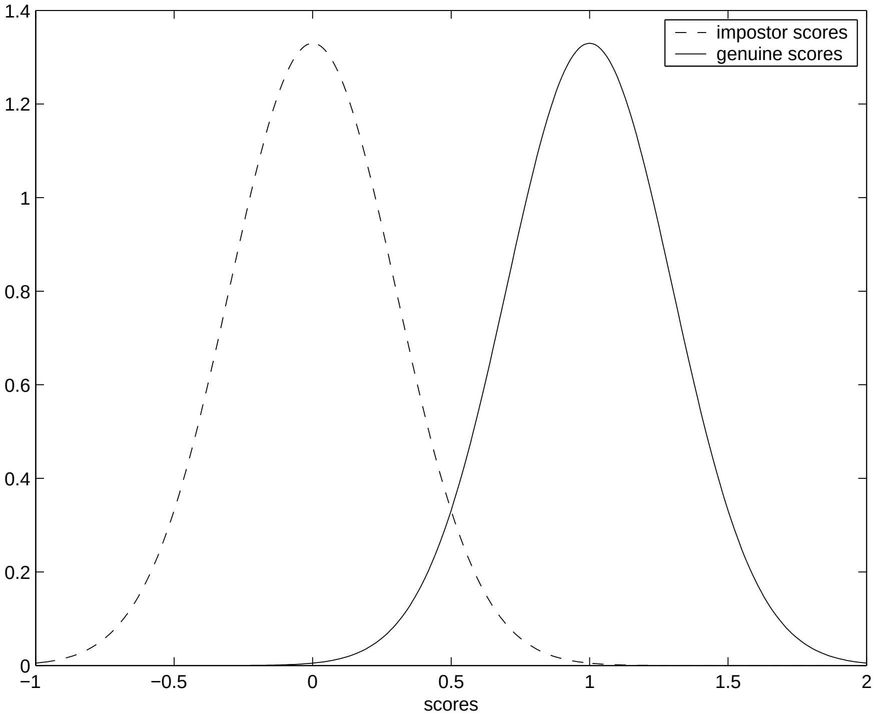


Fig. 7. Hypothetical densities of matching(genuine) and non-matching(impostors) scores.

score will give some improvement during the decision step.

3) In every observed identification attempt : $s_{imp} = s_{gen} - 1$. Thus in this scenario the identification system always correctly places genuine sample on top. Score distributions of Figure 7 do not reflect this fact. By using appropriate identification model (say, accept i if $s_i = \max_j s_j$) we can successfully separate all correct identification results from incorrect ones.

Matching scores are frequently dependent. Scores were dependent to some extent in all the applications we experimented with(see below Sec. VI). We can point out at least two causes of such dependence:

1) Recognizers usually incorporate some measure of input quality into matching score. If quality of the input is low we can expect all matching score to be low, and if quality is high, then matching scores also will be high.
2) In some applications, like character recognition, we expect images to belong to a definite set of classes, and if an image is in one class, it will be quite different from images in

other classes. When the distortion is small and the correct class is matched, the distance to other classes will dictate low impostor scores. But if impostor class is matched, the input sample probably lies somewhere in between classes, and the second best score is comparable to the first one.

Summarizing the above discussion, if matching scores are independent we expect to achieve average performance improvement by combining the second-best score. If scores are dependent, then any situation from no improvement to perfect decision is possible. Scores are usually dependent and therefore considering second-best score in decision is beneficial.

V. Low Complexity Combinations in Identification System

In this section we prove that the set of medium II complexity combinations is different from the set of low complexity combinations. Though this fact can be intuitively obvious, we present a strict mathematical proof by constructing examples of identification systems in which optimal low complexity combination performs worse than optimal medium II complexity combination.

As we discussed in [29] the optimal combination algorithms are different for verification and identification modes of system operation if there is a dependence between matching scores assigned to different classes. Thus we construct two examples, one for verification and another for identification operating modes, to prove the difference in complexity types for both operating modes.

For verification mode of operation we consider an identification system with only one matcher - the matcher described in the third scenario of the previous section. Let s_1 and s_2 be two matching scores assigned to two classes, one genuine and another impostor. The low complexity combination has to rely on only a single score of a claimed class. Suppose a class 1 is claimed and the only decision we can make is by comparing score s_1 to some threshold θ: accept verification attempt if $s_1 > \theta$. Clearly, there could be verification trials where class 1 is impostor, but its score is bigger than the threshold and it is accepted, or trials where class 1 is genuine, but it is rejected since its score is less than thresholds. So, $FRR(\theta) > 0$ and $FAR(\theta) > 0$ for any θ. On the other hand, we can consider the following decision algorithm of medium II complexity type: accept claimed identity i if $s_i > s_{3-i}$, and reject otherwise. Such algorithm will have $FRR = FAR = 0$. So we proved, that for verification mode of operation the optimal low complexity combination is not able to achieve same performance as optimal medium II complexity combination.

For identification mode of operation we consider an identification system with two matchers and two classes. Let these matchers be independent. Let the matching scores be only 0 or 1 and let the probabilities of score outputs for first (s_1^1, s_2^1) and second (s_1^2, s_2^2) matchers in case first class (ω_1) is a genuine class be given in table I. Let also assume, that if second class (ω_2) is genuine, then corresponding score pairs probabilities are the same (with permutation): $P(s_1^1, s_2^1|\omega_2 \text{ is genuine}) = P(s_2^1, s_1^1|\omega_1 \text{ is genuine})$ and $P(s_1^2, s_2^2|\omega_2 \text{ is genuine}) = P(s_2^2, s_1^2|\omega_1 \text{ is genuine})$.

s_1^1	s_2^1	$P(s_1^1, s_2^1\|\omega_1 \text{ is genuine})$
0	0	0.1
0	1	0.1
1	0	0.4
1	1	0.4

s_1^2	s_2^2	$P(s_1^2, s_2^2\|\omega_1 \text{ is genuine})$
0	0	0.1
0	1	0.2
1	0	0.5
1	1	0.2

TABLE I

PROBABILITIES OF MATCHING SCORE OUTPUTS FOR CLASSIFIERS IN IDENTIFICATION OPERATING MODE EXAMPLE.

Low complexity combination for such system operating in identification mode will be represented by the formula:

$$C = \arg\max_{i=1,2} f(s_i^1, s_i^2)$$

and since scores s_i^j are only of two values (0 and 1) we can enumerate all possible combination functions as having values $\{0, 1, 2, 3\}$ on different pairs $\{0, 0\}, \{0, 1\}, \{1, 0\}, \{1, 1\}$. After considering all such possible combination functions, we find that function f shown in Table II gives the best correct identification rate of 0.62.

Note, that due to the symmetry in original score distributions $P(s_1^1, s_2^1|\omega_1 \text{ is genuine})$ for different genuine classes ω_1 and ω_2, we can not have class-specific combinations. Thus the set of low complexity combinations coincides with the set of medium I combinations, and the set of medium II combinations coincides with the set of high complexity combinations. So, the optimal combination of medium II complexity type is exactly the optimal combination of high complexity type, and which is the optimal classification algorithm on the set of all scores $(s_1^1, s_1^2, s_2^1, s_2^2)$. Since we know distributions of scores assigned by our matchers,

s_i^1	s_i^2	$f(s_i^1, s_i^2)$
0	0	0
0	1	2
1	0	1
1	1	3

TABLE II

OPTIMAL LOW COMPLEXITY COMBINATION FUNCTION FOR IDENTIFICATION OPERATING MODE EXAMPLE.

using Bayes theorem and matcher independence assumption ($P(s_1^1, s_2^1, s_1^2, s_2^2|\omega_1 \text{ is genuine}) = P(s_1^1, s_2^1|\omega_1 \text{ is genuine})P(s_1^2, s_2^2|\omega_1 \text{ is genuine})$), we can calculate posterior probabilities of both classes for each combination of score outputs ($P(\omega_i \text{ is genuine}|s_1^1, s_2^1, s_1^2, s_2^2)$) and perform optimal Bayesian classification. Such classification achieves correct identification rate of .65. Thus, the medium II complexity combination achieves better performance in identification operating mode than optimal low complexity combination. Note, that in both cases we had the same undecided rate of .15, where the value of combination function is the same for both classes, or the posterior class probability is the same for both classes.

VI. COMBINATION COMPLEXITY FRAMEWORK AND IDENTIFICATION MODEL

The theoretical examples from the previous section showed that we are better to consider medium II or high complexity combinations for identification systems if match scores assigned by any classifier are dependent. Low complexity combinations are not capable to account for this dependence and might have suboptimal performance. In order to verify the dependence of match scores output by classifiers during identification trials, we measured the correlation between genuine score and different statistics of the sets of impostor scores. As table III shows, the scores produced by real life classifiers indeed can be dependent.

The presence of the dependence between match scores suggests that we might want to use medium II or high complexity combinations. High complexity combinations, though, require training a separate combination function for each person, and thus they need a significant amount of training data. On the other hand, medium II combination methods need only one combination function to be trained, and training data set can be quite small. Thus combination methods of

Matchers	$first_{imp}$	$second_{imp}$	$mean_{imp}$
li	0.3164	0.3400	0.2961
ri	0.3536	0.3714	0.3626
C	0.1419	0.1513	0.1440
G	0.1339	0.1800	0.1593

TABLE III

CORRELATIONS BETWEEN s_{gen} AND DIFFERENT STATISTICS OF THE IMPOSTOR SCORE SETS PRODUCED DURING IDENTIFICATION TRIALS IN NIST BSSR1 DATA.

medium II complexity type seem to be a good choice for identification problems.

As we discussed before, the previous combination approaches falling into medium II complexity type are rank based combination methods and combination methods involving particular normalizations of match scores followed by a simple combination function. The problem with rank based methods is that the score information is simply discarded. It is easy to construct an example where small difference in scores will result in big difference in ranks and will confuse the combination algorithm. Altincay and Demirekler [23] presented one such example.

Score normalization methods, which utilize the whole set of scores obtained during a current identification trial in order to normalize a score related to a particular class, followed by some combination algorithm, remain a viable approach to combination in identification systems. The question is, of course, what is the proper way to do such normalizations. The same paper by Altincay and Demirekler gives examples where normalization procedures frequently used result in a loss of information contained in a classifier's scores and yield suboptimal classification.

Identification systems produce matching scores for all persons in the database (we assume simple identification systems with no indexing). Experiments based on utilizing the second-best score for accepting identification results were presented in section IV. Our results show significant benefits resulting from using both the best and the second-best scores in order to accept or reject a class corresponding to the best score. Thus second best score alone can provide good amount of information for the identification model construction. We used similar statistic for identification models in our experiments. Additional advantage of using such statistic is that it is sometimes impossible to obtain all identification match score, but the second best match score will be usually available from the implementation of identification system.

VII. Derivation of Combination Rules Using Identification Model

In this section we present different combination methods of medium II complexity type utilizing identification model. The goal is to theoretically derive optimal combination algorithm with the assumption that the joint densities of the scores and score set statistics are known. We will also discuss the application of so called 'background model' and its relation to the identification model. Score normalization methods developed in the speaker identification research use the term 'background model' to describe the probabilities associated with the event that a considered class is an impostor class during the current identification attempt. Our term 'identification model' has a different meaning and describes the dependencies between scores output for all classes during any one identification attempt.

A. Likelihoods with Identification Model

Suppose that we combine M independent classifiers, and each classifier outputs N dependent scores. The optimal combination algorithm is the Bayesian classifier which accepts these NM scores and chooses the class which maximizes the posterior class probability. Thus the goal of optimal combination is to find

$$\arg\max_k P(C_k | \{s_i^j\}_{i=1,\dots,N;j=1,\dots,M})$$

Term C_k refers to the fact that the class k is the genuine class. By the Bayes theorem

$$P(C_k | \{s_i^j\}_{i=1,\dots,N;j=1,\dots,M}) = \frac{p(\{s_i^j\}_{i=1,\dots,N;j=1,\dots,M} | C_k) P(C_k)}{p(\{s_i^j\}_{i=1,\dots,N;j=1,\dots,M})}$$

and since the denominator is the same for all classes, our goal is to find

$$\arg\max_k p(\{s_i^j\}_{i=1,\dots,N;j=1,\dots,M} | C_k) P(C_k)$$

or, assuming all classes have the same prior probability,

$$\arg\max_k p(\{s_i^j\}_{i=1,\dots,N;j=1,\dots,M} | C_k)$$

By our current assumption, classifiers are independent, which means that any subset of scores produced by one classifier is statistically independent from any other subset of scores produced by another classifier. Hence, our problem is to find

$$\arg\max_k \prod_j p(\{s_i^j\}_{i=1,\dots,N}|C_k) \tag{1}$$

The problem now is to reliably estimate the densities $p(\{s_i^j\}_{i=1,\dots,N}|C_k)$, which is a rather hard task given that the number N of classes is large and we do not have many samples of each class for training. The last problem is solved by noticing that we do not construct class specific combination, and thus class indexes can be permuted. Thus all training samples pertaining to different genuine classes can be used to train only one density, $p(s_k, \{s_i^j\}_{i=1,\dots,N,i\neq k}|C_k)$. Now s_k^j is a score belonging to genuine match, and all other scores $\{s_i^j\}_{i=2,\dots,N}$ are from impostor matches. Since there are many impostor scores participating in this density, we might somehow try to eliminate them. Recall, that when considering identification models for decision, we relied on the second best score output by the classifier. Could we use similar consideration and rely only on one or two impostor scores?

Indeed, instead of $p(s_k, \{s_i^j\}_{i=1,\dots,N,i\neq k}|C_k)$ we can consider $p(s_k^j, t_k^j|C_k)$, where t_k^j is some statistics of all other scores besides s_k^j. In all subsequent experiments we were using statistics ”second best score besides current score”, sbs(s). More precisely, $t_k^j = sbs(s_k^j)$ means the second best score among current identification trial scores $\{s_i^j\}_{i=1,\dots,N,i\neq k}$ not including s_k^j. The final combination rule for this method is to find

$$\arg\max_k \prod_j p(s_k^j, t_k^j|C_k) \tag{2}$$

As our previous experiments showed[3] this algorithm does not perform as well as traditional likelihood ratio combination:

$$\arg\max_k \prod_j \frac{p(s_k^j|C_k)}{p(s_k^j|\overline{C_k})} \tag{3}$$

It seems that the score set statistics t_k^j of our identification model does not fully reflect the background information for score s_k^j, whereas the term $p(s_k^j|\overline{C_k})$ indeed contains such information. As an example, the genuine matching scores s_k^j can be very strong, but located in the region of low probability (both $p(s_k^j|C_k)$ and $p(s_k^j, t_k^j|C_k)$ are small), whereas $p(s_k^j|\overline{C_k})$ could be even smaller, and likelihood ratio can still succeed. In the next section we try to derive a combination rule which combines information from both identification model and background models.

B. Likelihood Ratios with Identification Model

As above we consider posterior class probability, apply Bayes formula, but now use independence of classifiers to decompose the denominator:

$$\begin{aligned} P(C_k|\{s_i^j\}_{i=1,\dots,N;j=1,\dots,M}) &= \frac{p(\{s_i^j\}_{i=1,\dots,N;j=1,\dots,M}|C_k)P(C_k)}{p(\{s_i^j\}_{i=1,\dots,N;j=1,\dots,M})} \\ = \frac{\prod_j p(\{s_i^j\}_{i=1,\dots,N}|C_k)P(C_k)}{\prod_j p(\{s_i^j\}_{i=1,\dots,N})} &= P(C_k)\prod_j \frac{p(\{s_i^j\}_{i=1,\dots,N}|C_k)}{p(\{s_i^j\}_{i=1,\dots,N})} \end{aligned} \tag{4}$$

The next step is similar to the step in deriving the algorithm for background speaker model [21]. We consider class $\overline{C_k}$ meaning some other class is genuine, and decompose $p(\{s_i^j\}_{i=1,\dots,N}) = P(C_k)p(\{s_i^j\}_{i=1,\dots,N}|C_k) + P(\overline{C_k})p(\{s_i^j\}_{i=1,\dots,N}|\overline{C_k})$. By assuming that N is large and $P(\overline{C_k}) \gg P(C_k)$, we can discard the first term and get the following classifier decision:

$$\arg\max_k \prod_j \frac{p(\{s_i^j\}_{i=1,\dots,N}|C_k)}{p(\{s_i^j\}_{i=1,\dots,N}|\overline{C_k})} \tag{5}$$

In comparison with decision 1 of the previous section we have an additional density $p(\{s_i^j\}_{i=1,\dots,N}|\overline{C_k})$. Such density can be viewed as a background of impostors for the genuine class C_k. As research in speaker identification suggests, utilizing such background model is beneficial for the system performance.

We estimated the ratios of equation 6 by additional modeling of $p(\{s_i^j\}_{i=1,\dots,N}|\overline{C_k})$. We used an approach similar to the previous section to estimate this density as $p(s_k^j, t_k^j|\overline{C_k})$ with $t_k^j = sbs(s_k^j)$ - the joint density of impostor scores s_k^j and corresponding identification trial statistics t_k^j. The final combination rule is

$$\arg\max_k \prod_j \frac{p(s_k^j, t_k^j|C_k)}{p(s_k^j, t_k^j|\overline{C_k})} \tag{6}$$

The technique described in this section can be characterized as a composition of identification model and background model. The identification model considers $p(s_k, t_k^j|C_k)$ and $p(s_k, t_k^j|\overline{C_k})$ instead of $p(s_k|C_k)$ and $p(s_k|\overline{C_k})$, and background model considers $p(s_k, t_k^j|\overline{C_k})$ or $p(s_k|\overline{C_k})$ in addition to $p(s_k, t_k^j|C_k)$ or $p(s_k|C_k)$. Thus, the identification model differs from background model by accounting for dependencies of scores in identification trials by using some statistic t_k^j.

Note, that traditional likelihood ratio (Eq. 3) is the optimal combination method for low complexity combinations operating in verification mode (see [29]). Thus, its extension by identification model (Eq. 6) should provide a good combination method of medium II complexity type for verification mode operations. Due to the derivation of 6 it also should provide a reasonable performance in identification mode. So, we will be testing this method for both verification and identification modes of operation of identification system.

C. Combinations of Dependent Classifiers

The combination algorithms presented in the previous two sections deal with independent classifiers. How should we address dependent classifiers?

By looking at the combination formulas 1 and 6 we can see that each classifier contributes terms $p(\{s_i^j\}_{i=1,\dots,N}|C_k)$ and $\frac{p(\{s_i^j\}_{i=1,\dots,N}|C_k)}{p(\{s_i^j\}_{i=1,\dots,N}|\overline{C_k})}$ correspondingly to the combination algorithm. Thus one can conclude that it is possible to model the same terms for each classifier, and then combine them by some other trainable function.

Note that any relationships between scores $s_{i_1}^{j_1}$ and $s_{i_2}^{j_2}$ where $i_1 \neq i_2$ and $j_1 \neq j_2$ will be essentially discarded. This seems to be inevitable for the current complexity type of combinations - medium II. If we wanted to account for such relationships, we would need class-specific combination functions, and thus higher complexity combinations.

D. Normalizations Followed by Combinations and Single Step Combinations

Figure 8 represents in graphical form the type of combinations we have presented thus far. All these combinations consist of two steps. In the first step, each score is normalized by using other scores output by the same matcher. In the second step, normalized scores are combined using a predetermined or trained combination function.

However, it is not necessary to have these two steps for combinations. For each normalization happening in the first step we use the same identification model statistic and the same trained density estimates. Thus the contribution of the particular classifier j to the whole combination algorithm's output for class i is calculated only from score s_i^j and statistic t^j (statistic though could vary for a class; in first case it was best or second best score; thus in fact two values are used). Figure 9 presents a diagram on how scores and statistics from all participating classifiers could be combined in a single combination step. The example of this combination is the weighted sum rule utilizing identification model described in the next section.

In the algorithm presented by this diagram the statistics t^j are extracted for each classifier j using its output scores by a predetermined and non-trainable algorithm. The scores related to a

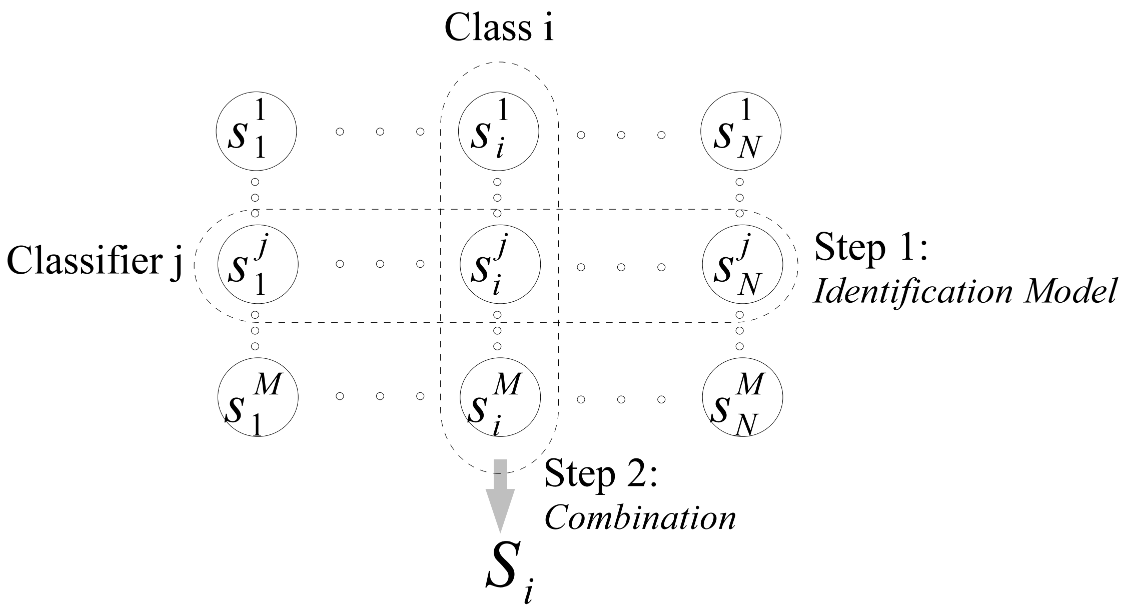


Fig. 8. 2-step combination method utilizing identification model.

particular class and statistics are combined together by a trainable function. This combination function is not class-specific and is easily trainable. This type of combinations are of medium II complexity type. In comparison, in low complexity type combinations only scores for a particular class are combined, and not statistics from identification models of classifiers.

E. Identification Model for Weighted Sum Combination

As an example of a single step combinations, we consider a weighted sum rule utilizing our second best score identification model. Weighted sum rule can be specifically trained to maximize correct identification rate for identification mode of operation, and it provides good performance in this case[29]. It is not optimal for verification mode though. Thus, we will test the performance of weighted sum rule with and without identification model modification in identification mode operation only.

The traditional weighted sum combination without identification model ('weighted sum local') is a low complexity combination which combines M scores from M biometric matchers assigned to a particular class i:

$$S_i = w_1 s_i^1 + \cdots + w_M s_i^M \tag{7}$$

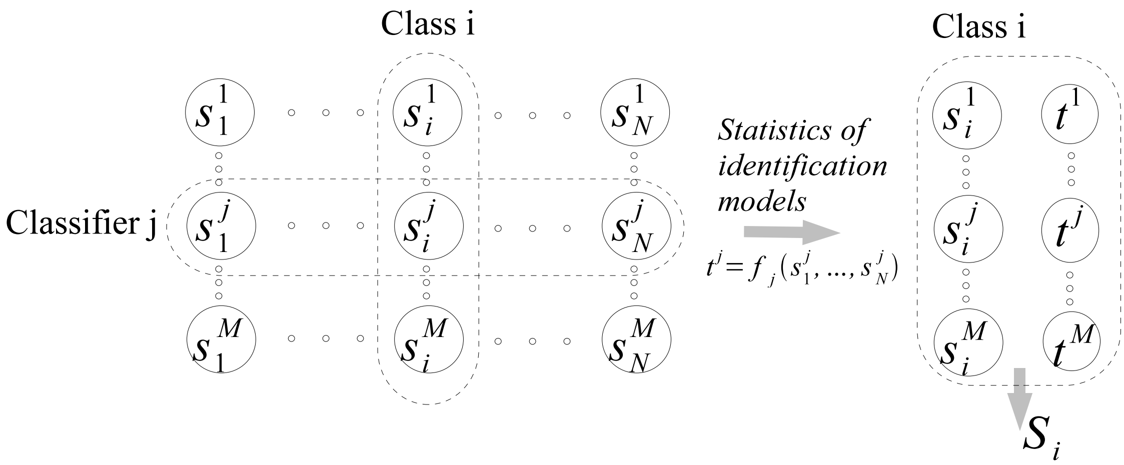


Fig. 9. 1-step combination method utilizing identification model.

The weighted sum rule with identification model ('weighted sum global') combines scores as well as statistics of score sets:

$$S_i = w_1 s_i^1 + w_2 sbs(s_i^1) + \cdots + w_{2M-1} s_i^M + w_{2M} sbs(s_i^M) \tag{8}$$

Note, that we use second best score statistics which showed good performance in decision making applications IV-B before. In both cases we train the weights so that the number of failed identification attempts is minimized.

VIII. EXPERIMENTS

We used biometric matching score set BSSR1 distributed by NIST[28]. This set contains matching scores for a fingerprint matcher and two face matchers 'C' and 'G'. Fingerprint matching scores are given for left index 'li' finger matches and right index 'ri' finger matches. Since we wanted to consider the case of independent matchers we performed four sets of experiments on combining fingerprint and face scores : 'li' & 'C', 'li' & 'G', 'ri' & 'C', and 'ri' & 'G'.

Though the BSSR1 score set has a subset of scores obtained from same physical individuals, this subset is rather small - 517 identification trials with 517 enrolled persons. In our previous experiments[2] we used this subset, but the number of failed identification attempts for most experiments was less than 10 and it is difficult to compare algorithms with so few negatives. In this work we use bigger subsets of fingerprint and face matching scores of BSSR1 by creating virtual persons; the fingerprint scores of a virtual person come from one physical person and the face scores come from another physical person. The scores are not reused, and thus we are limited to the maximum number of identification trials - 6000 and the maximum number of classes, or enrolled persons, - 3000. Some enrollees and some identification trials also needed to be discarded since all corresponding matching scores were invalid probably due to enrollment errors. In the end we split data in two equal parts - 2991 identification trials with 2997 enrolled persons with each part used as training and testing sets in two phases. The results for these two phases are added up later.

Two types of combinations are used in experiments - likelihood ratio and weighted sum. For likelihood ratio combinations we estimate score densities using Parzen window method with gaussian kernels. The kernel width is determined by the maximum likelihood method. For the weighted sum methods we employ a brute force approach to find the optimal weights maximizing the number of correct identification trials on the training sets. It is possible to use brute force search in this case since the maximum number of weights is 4. In both likelihood ratio and weighted sum experiments we used separate sets for training the combination method and testing it.

A. Performance in Identification Operating Mode

Table IV shows the numbers of failed identification trials among the total number of $2991*2 = 5982$ trials. 'LR' is the traditional likelihood ratio combination method of Eq. 3. 'LR+sbs' is the likelihood ratio augmented with second best score identification model of Eq. 6. 'WSum' is the traditional weighted sum combination of Eq. 7. 'WSum+sbs' is the weighted sum combination augmented with second best score identification model of Eq. 8. For comparison we also used traditional likelihood ratio and weighted sum combination methods with scores preprocessed by previously mentioned (Sec. III) T-normalization:

$$s_i^j(l) \rightarrow \frac{s_i^j(l) - \mu^j(l)}{\sigma^j(l)}$$

where $s_i^j(l)$ denotes a matching score given by classifier j to class i during identification trial l, $\mu^j(l)$ and $\sigma^j(l)$ are correspondingly the mean and the standard deviation of the set of scores produced by matcher j during identification trial l. T-normalization, as well as second best score identification model, makes our combinations of medium II complexity type, but, in contrast to second best score model, T-normalization is an untrainable method.

Matchers	LR	LR+T-norm	LR+sbs	WSum	WSum+T-norm	WSum+sbs
li & C	165	175	163	166	165	156
li & G	245	268	220	271	269	222
ri & C	106	126	112	114	116	107
ri & G	171	181	152	190	183	158

TABLE IV

EXPERIMENTS ON COMBINATIONS IN IDENTIFICATION SYSTEMS. ENTRIES ARE THE NUMBERS OF FAILED TEST IDENTIFICATION TRIALS.

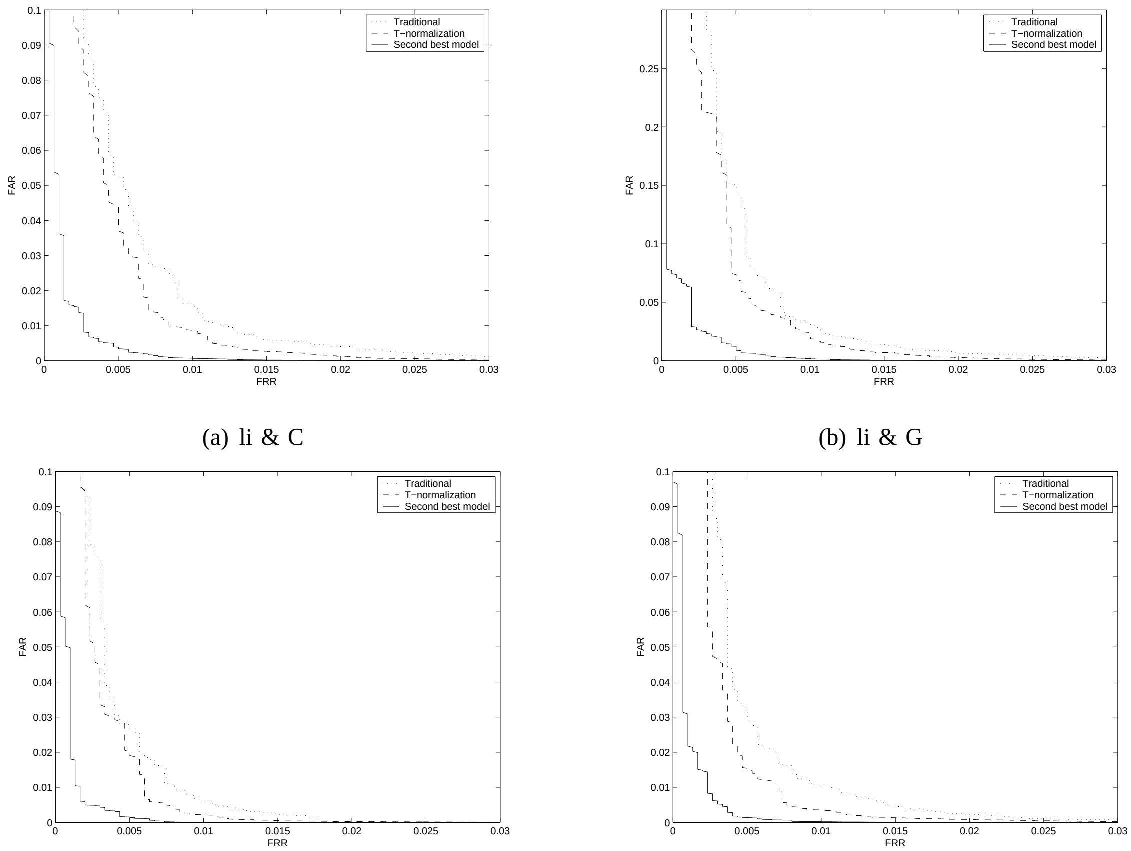
We can see that in all cases except one the addition of second best score identification model into corresponding low complexity algorithm resulted in performance improvement. The addition of T-normalization was detrimental for likelihood ratio combination method, and marginally positive for weighted sum combinations.

B. Performance in Verification Operating Mode

Although there are examples where score normalization techniques with background models are used for identification tasks[15], even more applications use such techniques for identification systems operating in verification mode[24], [30], [21]. We also applied the combinations utilizing identification models for biometric person verification tasks. The drawback of using either the background models or the identification models in verification tasks is that we have to produce not only one match per person and per matcher, but also some set of matching scores for other persons enrolled in the system, or some artificially modeled persons.

Figure 10 contains the results of experiments of our system operating in verification mode. The ROC performance curves were constructed using combinations of 2991 genuine and $2991 * (2997 - 1)$ impostor score sets (note that in contrast to identification mode experiments we only considered one training and one testing sets for each combination).

The results show that we were able to achieve significant improvement in the verification task performance by utilizing second best score identification model of Eq. 6. These improvements



(a) li & C (b) li & G

(c) ri & C (d) ri & G

Fig. 10. ROC curves for likelihood ratio combinations utilizing and not utilizing identification models in verification mode.

seem to be similar to the improvements achieved by using identification models for making acceptance decisions in biometric person identification in section IV-B. The T-normalization is also beneficial to the smaller extent in these experiments.

IX. Conclusion

Presented four complexity combination types originate naturally from the structure of the combination parameters - each score corresponds to some class and classifier, and the output of the combination algorithm corresponds to some class. Complexity combination types form a foundation for the classifier combination framework. The combination framework prompts the user to choose the combination complexity type first based on the numbers of classes and classifiers, and the number of training samples. Within a chosen complexity type one can use any generic classifier for combination. Finally, a generic classifier used for combination can be modified to account for a chosen complexity type and for any extraneous information about classifiers.

The problem of classifier combination in identification systems can be treated as an application of the combination framework. We observe that frequently the algorithms, employed for combining matchers in biometric identification systems, only use the scores related to one class to produce a final combination score. Instead of using low complexity combination algorithms in identification systems, we attempt to use medium II complexity type combinations, which utilize all available scores and require training only single combination function. Combination algorithms of low complexity type discard the dependency information between scores assigned to all classes by any single classifier. We gave examples, that if such information is discarded and low complexity type combinations are used instead of medium II complexity type combinations, then the combination is suboptimal.

In order to reflect the relationships between scores assigned by one classifier to different classes, we introduced the concept of the identification model. The identification model application is a score normalization algorithm where normalization depends on all scores output by a classifier in one identification trial, and the algorithm is the same for all classes. Thus our identification model has less complexity than similar attempts to normalization [31], [32]. In these previous attempts normalizations were class specific and required huge amount of training data. The combinations utilizing such normalizations will be similar to Behavior Knowledge Space combination [33], and they belong to high complexity combination type. Biometric identification problems can have large number of enrolled persons, and such combinations are not feasible due to the lack of training data. By restricting ourselves to non-class-specific normalizations of the identification model we are able to concentrate on combinations of medium II complexity type. Such combinations have significantly lower complexity, and result in efficient algorithms for identification systems.

In section IV we have shown how the identification model can be used in order to improve the performance of decision making in identification systems, and section VII contains examples of combination algorithms, likelihood ratio and weighted sum combination rules, utilizing identification models . The experiments show significant advantages for combinations of medium II type utilizing identification models, as opposed to less efficient low complexity type combinations, and non-feasible high complexity combinations.

REFERENCES

[1] L. Xu, A. Krzyzak, and C. Y. Suen, "Methods for combining multiple classifiers and their applications to handwriting recognition," *IEEE transactions on System, Man, and Cybernetics*, vol. 23, no. 3, pp. 418–435, 1992.

[2] S. Tulyakov and V. Govindaraju, "Classifier combination types for biometric applications," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), Workshop on Biometrics*, New York, USA, 2006.

[3] ——, "Identification model for classifier combinations," in *Biometrics Consortium Conference*, Baltimore, MD, 2006.

[4] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score normalization for text-independent speaker verification systems," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 42–54, 2000.

[5] D.-S. Lee, *Theory of Classifier Combination: The Neural Network Approach*. SUNY at Buffalo: Ph.D Thesis, 1995.

[6] L. I. Kuncheva, J. C. Bezdek, and R. P. W. Duin, "Decision templates for multiple classifier fusion: an experimental comparison," *Pattern Recognition*, vol. 34, no. 2, pp. 299–314, 2001.

[7] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Wiley InterScience, 2004.

[8] S. Tulyakov, "A complexity framework for combination of classifiers in verification and identification systems," Ph.D. dissertation, State University of New York at Buffalo, 2006.

[9] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[10] R. M. Bolle, J. H. Connell, S. Pankanti, N. K. Ratha, and A. W. Senior, *Guide To Biometrics*. New York: Springer, 2004.

[11] A. Jain and A. Ross, "Learning user-specific parameters in a multibiometric system," in *Image Processing. 2002. Proceedings. 2002 International Conference on*, vol. 1, 2002, pp. I–57–I–60 vol.1.

[12] J. Fierrez-Aguilar, D. Garcia-Romero, J. Ortega-Garcia, and J. Gonzalez-Rodriguez, "Bayesian adaptation for user-dependent multimodal biometric authentication," *Pattern Recognition*, vol. 38, no. 8, pp. 1317–1319, 2005.

[13] T. K. Ho, "A theory of multiple classifier systems and its application to visual word recognition," Ph.D Thesis, SUNY Buffalo, 1992.

[14] T. K. Ho, J. Hull, and S. Srihari, "Decision combination in multiple classifier systems," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 16, no. 1, pp. 66–75, 1994.

[15] R. Brunelli and D. Falavigna, "Person identification using multiple cues," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 17, no. 10, pp. 955–966, 1995.

[16] L. Hong and A. Jain, "Integrating faces and fingerprints for personal identification," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 12, pp. 1295–1307, 1998.

[17] Y. Huang and C. Suen, "A Method of Combining Multiple Experts for Recognition of Unconstrained Handwritten Numerals," *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 17, no. 1, pp. 90–94, 1995.

[18] A. Saranli and M. Demirekler, "A statistical unified framework for rank-based multiple classifier decision combination," *Pattern Recognition*, vol. 34, no. 4, pp. 865–884, 2001.

[19] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern Recognition*, vol. 38, no. 12, pp. 2270–2285, 2005.

[20] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On combining classifiers," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 3, pp. 226–239, 1998.

[21] J. Mariethoz and S. Bengio, "A unified framework for score normalization techniques applied to text independent speaker verification," *IEEE Signal Processing Letters*, vol. 12, 2005.

[22] P. Grother, "Face recognition vendor test 2002 supplemental report," NIST, Tech. Rep. NISTIR 7083, 2004.

[23] H. Altincay and M. Demirekler, "Undesirable effects of output normalization in multiple classifier systems," *Pattern Recognition Letters*, vol. 24, no. 9-10, pp. 1163–1170, 2003.

[24] A. Rosenberg and S. Parthasarathy, "Speaker background models for connected digit password speaker verification," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, vol. 1, 1996, pp. 81–84 vol. 1.

[25] J. Colombi, J. Reider, and J. Campbell, "Allowing good impostors to test," in *Signals, Systems & Computers, 1997. Conference Record of the Thirty-First Asilomar Conference on*, vol. 1, 1997, pp. 296–300 vol.1.

[26] S. Tulyakov and V. Govindaraju, "Combining matching scores in identification model," in *8th International Conference on Document Analysis and Recognition (ICDAR 2005)*, Seoul, Korea, 2005.

[27] ——, "Identification model with independent matching scores," in *Biometrics Consortium Conference*, Crystal City, VA, 2005.

[28] "Nist biometric scores set. http://www.nist.gov/biometricscores/."

[29] S. Tulyakov, V. Govindaraju, and C. Wu, "Optimal classifier combination rules for verification and identification systems," in *7th International Workshop on Multiple Classifier Systems*, Prague, Czech Republic, 2007.

[30] A. Schlapbach and H. Bunke, "Using hmm based recognizers for writer identification and verification," in *9th Intl Workshop on Frontiers in Handwriting Recognition (IWFHR-9 2004)*, 2004.

[31] D. Bouchaffra, V. Govindaraju, and S. Srihari, "A methodology for mapping scores to probabilities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, September 1999.

[32] K. Ianakiev, *Organizing Multiple Experts for Efficient Pattern Recognition*. SUNY at Buffalo: Ph.D Thesis, 2000.

[33] L. Lam and C. Y. Suen, "Optimal combinations of pattern classifiers," *Pattern Recognition Letters*, vol. 16, no. 9, pp. 945–954, 1995.