Use of Identification Trial Statistics for the Combination of Biometric Matchers

Sergey Tulyakov and Venu Govindaraju, *Fellow, IEEE*

***Abstract*—Combination functions typically used in biometric identification systems consider as input parameters only those matching scores which are related to a single person in order to derive a combined score for that person. We discuss how such methods can be extended to utilize the matching scores corresponding to all people. The proposed combination methods account for dependencies between scores output by any single participating matcher. Our experiments demonstrate the advantage of using such combination methods when dealing with a large number of classes, as is the case with biometric person identification systems. The experiments are performed on the National Institute of Standards and Technology BSSR1 dataset and the combination methods considered include the likelihood ratio, neural network, and weighted sum.**

***Index Terms*—Biometric identification systems, combination of classifiers.**

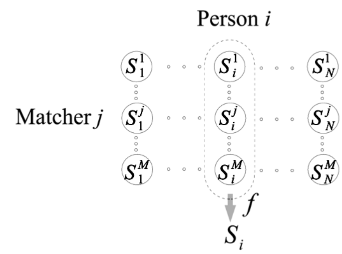


Fig. 1. Set of scores available for combinations in identification systems includes all MN matching scores from M matchers and assigned to all N persons. The combination functions f usually only utilize the set of scores related to one person i in order to calculate the combined matching score for this person.

I. Introduction

BIOMETRIC applications operate in two modes: 1) verification (1:1) mode and 2) identification (1:N) mode. Common approaches to combining biometrics for (1:N) identification applications are usually a simple iterative use of the (1:1) verification system. The combined score assigned to a particular enrolled person is obtained as a function of the scores assigned to that person by all of the biometric matchers in either mode of operation. However, in the identification mode, additional information is available for deriving the combined score for any person in the database of enrollees. This additional information is available from the matching scores returned to the enrollees other than the target person.

We consider M multiple biometric matchers used to produce MN matching scores (Fig. 1), where N is the number of enrolled persons. We assume that M is small and N is large. Each biometric matcher in such a setting is equivalent to a classifier assigning matching scores to each N class or person. And the combination of biometric matchers can be viewed as a classifier combination problem with a large number of classes.

Combination methods can be categorized based on the construction properties of the combination functions f. When methods use a single common combination function, they are called class generic methods. When each class has its own combination function, so that the combined scores are calculated differently for different classes, the methods are called class specific.

Local methods take as parameters only the M match scores related to a particular class (single column in Fig. 1) whereas the global methods consider the whole set of MN match scores (all columns in Fig. 1) to derive the combined score for any one class. In this paper, we explore global methods whose combination functions use the additional information (all columns) when computing the integrated score for each person.

When classifiers deal with a small number of classes, the dependencies between the 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. It is the most general type of combination method allowing optimal performance. However, learning class dependencies requires a significant number of training samples for each class. Such data might not be available for the 1:N identification mode systems, where usually a single template is enrolled for each person. In addition, the database of enrolled persons can be frequently changed making learning class relationships infeasible.

As a consequence, combination approaches in 1:N identification systems have considered only the local methods even when all of the MN scores are available. In this paper, we investigate the question of whether it is possible to improve the performance of the identification system by using all of the MN matching scores for deriving the combined score for each person [2], [3].

Manuscript received September 14, 2007; revised June 09, 2008. Current version published November 19, 2008. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Vijaya Kumar Bhagavatula.

The authors are with the Center for Unified Biometrics and Sensors, State University of New York (SUNY) at Buffalo, Buffalo, NY 14260 USA (e-mail: tulyakov@cubs.buffalo.edu; govind@cubs.buffalo.edu).

Digital Object Identifier 10.1109/TIFS.2008.2004287

II. Previous Work in Identification System Combinations

Traditionally, two types of biometric person authentication systems are defined: 1) verification (1:1) and 2) identification

(1:N) systems. It is usually implied that verification systems have only the matching scores related to one enrolled person available to the combination method. However, it is possible that a verification system additionally uses matching scores related to other people. For example, in [4], the authors performed "identification based verification" by utilizing matching scores of other enrolled peoples while making verification decisions on a particular person.

In order to avoid confusion, we define an identification system as a system which provides matching scores for all N enrolled persons. As in [4], such systems can operate in verification mode also. An identification system operates in identification mode if its purpose is to classify an input as belonging to any of the N classes or persons. We assume that the classification decision is performed by applying the $\arg\max$ operator to the N combined scores

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

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 and cumulative match curve [5].

When an identification system operates in verification mode, we can distinguish two classes: 1) genuine and 2) 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 ROC curves showing the dependencies of errors on threshold θ (or DET curve [5]).

If we have a combination algorithm for verification systems, it can be sequentially applied for all people to operate in the identification mode [6]. However, this approach does not utilize dependencies between scores output by a single matcher (i.e., the dependencies between the scores along the rows in the score matrix of Fig. 1). It is essentially a local method which considers only a single column of scores as input parameters to combination functions. Most combination algorithms used in biometric applications are of this type and are sometimes also user specific [7], [8].

Here, we present previous approaches which utilize score dependencies in the identification mode.

A. Rank-Based Combinations

T.K. Ho has used classifier combinations on the ranks of the scores instead of scores themselves by arguing that ranks provide more reliable information about a class being genuine [9], [10]. Thus, if the input image has low quality, then the genuine score as well as the impostor scores will be low. Combining the low score for genuine class with other scores could confuse a combination algorithm, but the rank of the genuine class remains as a stable statistic, and combining this rank with other ranks of the genuine class should result in true classification. Brunelli and Falavigna [11] considered a hybrid approach where the traditional combination of matching scores is fused with the rank information in order to achieve the identification decision. Hong and Jain [12] consider ranks, not for combination, but for modeling or normalizing the output score of a classifier. Behavior–knowledge space combination methods [13] are also based on ranks. Saranli and Demirekler [14] provide additional references for rank-based combination methods.

The problem with rank-based methods is that the score information is lost. Indeed, the best score can be much better than the 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 [15] refers to a transformation of scores based on a 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 (rows in the score matrix of Fig. 1) are not taken into consideration. If these normalized scores are later combined classwise (columnwise), then score dependence is not accounted for by the combination algorithm.

Some score normalization techniques can use a set of identification trial scores output by the classifier. For example, Kittler *et al.* [16] normalize each score by the sum of all the other scores before combination. Similar normalization techniques are used in zero (Z)- and test (T)-normalizations [17], [18]. Z-normalization uses impostor matching scores to produce a class-specific normalization. Z-normalization does not include the set of identification trial scores (rows in Fig. 1) and, thus, does not utilize the score dependency. On the other hand, T-normalization uses a set of scores produced during a single identification trial by utilizing statistics of mean and variance of the identification score set. Note that T-normalization is a predetermined routine without training. Still, using this simple kind of score modeling turns out to be quite useful; for example, [19] argued for applying T-normalizations in face verification. There is also a counterargument [20] that says useful classification information gets lost during such normalizations.

Score normalization techniques have been well developed in the speaker identification literature. The cohort normalizing method [21], [22] considers a subset of enrolled people close to the current test person in order to normalize the score for that person by a log-likelihood ratio of the genuine (current person) and impostor (cohort) score density models. Auckenthaler *et al.* [17] separated cohort normalization methods into cohorts found during training (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 all classifiers.

III. Complexity Types of Classifier Combinations

This section describes four types of combination methods and their requirements of training data. Ultimately, the problem characteristics and the availability of training scores determine

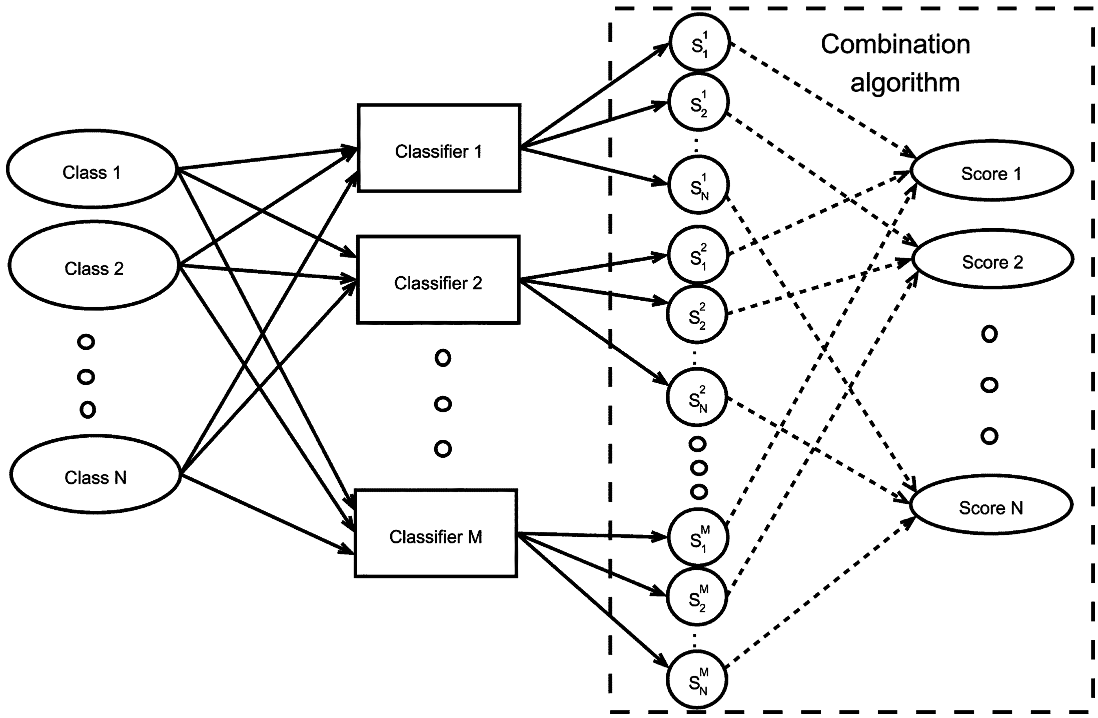


Fig. 2. Classifier combination takes a set of s_i^j—score for class i by classifier j and produces combination scores S_i for each class i. i is the index for the N classes and j is the index for the M classifiers.

the type of combination method which is appropriate for a particular problem.

Fig. 2 gives a different view of the problem of integrating scores in identification systems, for the purpose of formally categorizing the combination methods. The combination functions of local methods have a reduced parameter set (as connections in Fig. 2 show), and many well-known combination methods (e.g., weighted sum of scores) fall into this category. A fully connected artificial neural network (ANN) accepting MN input parameters and having N output parameters would present an example of the most general, class-specific, and global combination function algorithm [1], [23]. The disadvantage of this more general approach is that it requires a very large amount of training data, which might not be always available in identification systems.

A. Types of Combination Methods

Here, we develop a formal framework for combination methods further categorizing the local and global combination functions that are required to be trained. The first two categories correspond to local and the remaining two correspond to global methods.

1) Low complexity methods $S_i = f\left(\left\{s_i^j\right\}_{j=1,\ldots,M}\right)$. Methods of this type require that only one combination function is to be trained, and the combination function accepts input scores for one particular class as parameters. It represents class-generic and reduced-parameter set combination functions.
2) Medium complexity I methods $S_i = f_i\left(\left\{s_i^j\right\}_{j=1,\ldots,M}\right)$. Methods 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. It represents class-specific and reduced-parameter set combination functions.
3) Medium complexity II methods $S_i = f\left(\left\{s_i^j\right\}_{j=1,\ldots,M}, \left\{s_k^j\right\}_{j=1,\ldots,M;k=1,\ldots,N,k\neq i}\right)$. Methods of this type take as parameters not only the scores related to the same class, but all output scores of the classifiers. Combination scores for each class are calculated by using the same function, but scores for class i are given a special place in the parameter list. Applying function f for different classes effectively means permutation of the function's parameters. These combination functions are class generic and use the whole parameter set.
4) High complexity methods $S_i = f_i\left(\left\{s_k^j\right\}_{j=1,\ldots,M;k=1,\ldots,N}\right)$. Functions calculating final scores are different for all classes, and they accept as parameters all of the scores output by the base classifiers. This represents class-specific and whole parameter-set combination functions.

We can illustrate the different combination types using the matrix score representation (Fig. 1) as well. Each row corresponds to a set of scores output by a particular classifier, and each column corresponds to scores assigned by classifiers to a particular class. The illustration of each combination type functions is given in Fig. 3. In order to produce the combined score S_i for class i, low complexity methods (Fig. 3(a)) and medium I complexity (Fig. 3(b)) combinations consider only those classifier scores which are assigned to class i (column i), reflecting the property of local combination functions. Medium II [Fig. 3(c)]

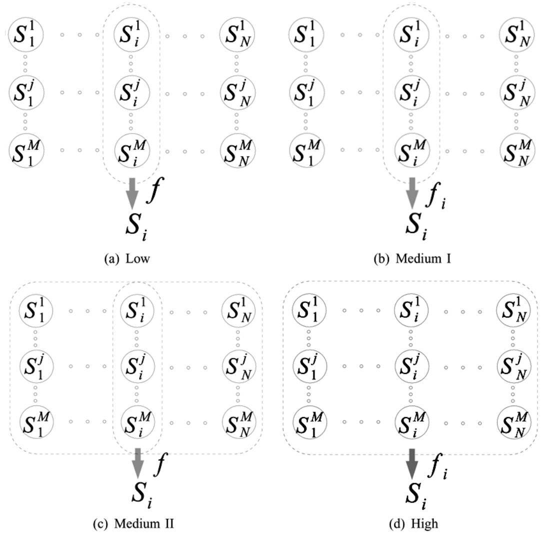


Fig. 3. Range of scores considered by each combination type and combination functions.

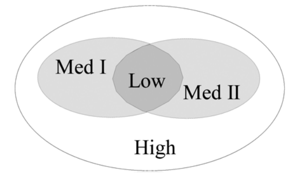


Fig. 4. Relationship diagram of different combination complexity types.

and high complexity [Fig. 3(d)] methods consider all of the scores output by classifiers for calculating a combined score S_i for class i, reflecting the property of global combination functions.

Low [Fig. 3(a)] and medium II [Fig. 3(c)] complexity methods have the same class-generic combination functions f irrespective of the class for which the score is calculated. Note that medium II complexity-type methods have scores related to a particular class in 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 in these methods is that the combination function f is the same for all classes, but the combined scores S_i differ since we effectively permute function inputs for different classes. Medium I [Fig. 3(b)] and high [Fig. 3(d)] complexity methods have class-specific combination functions f_i trained differently for different classes.

It is interesting to compare our combinations types with previous categorization of combination methods by Kuncheva *et al.* [24], who refer to the score matrix as "decision profile" and "intermediate feature space." Kuncheva's work also separates combinations into a "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. The continuation of this work [25] 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 such as the weighted sum rule. Further, we consider an additional category of the "medium II" type. An example of the "medium II" combination is the two-step combination algorithm 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 the "low" complexity type. Thus, scores in each row are combined first, and then the results are combined columnwise in the second step.

Fig. 4 illustrates the relationships between the types of combination methods. Medium complexity types are subsets of high complexity type, and the set of low complexity methods is exactly the intersection of sets of medium I and medium II combination methods. 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 [26], we provide a description of these complexity types using the concept of Vapnik–Chervonenkis (VC) dimension [27]. The ability to use the VC dimension for characterization of different combination types justifies our usage of the term "complexity types."

Higher complexity methods can potentially produce better classification results since more information is used. However, the availability of training samples limits the types of possible combinations to choose from. Thus, the choice of a combination method in any particular application is a tradeoff between classifying capabilities of the combination functions and the availability of sufficient training samples. Different generic classifiers, such as neural networks, decision trees, etc., can be used in combination within each complexity class. From the perspective of our framework, the main effort in solving the classifier combination problem consists of identifying the complexity type and modifying the generic classifiers (if needed) to compensate for a mismatched function complexity type for inadequate training data reasons.

The biometric person authentication systems we experimented with in this research have a high number of enrolled classes (people) N and a small number of classifiers (biometric face and fingerprint matchers) M. Most combinations methods described in the literature for biometric applications are of low complexity type. In this paper, we are interested in exploiting higher complexity combinations. We will derive combinations rules of the medium II complexity type which are analogous to the traditional likelihood ratio, neural-network, and weighted sum combinations of the low complexity type. Our experiments on large biometric score sets confirm that the medium II complexity combinations have better performance than their counterparts of low complexity.

Both identification and verification modes of operation can utilize combinations of all four complexity types. In our experiments, we compare the combination methods of low and medium II complexity types and report performance for both, verification and identification, modes of operation.

IV. Derivation of Combination Rules Using Identification Trial Statistics

In this section, we present different combination methods of the medium II complexity type by utilizing the statistics of the identification trial score set for normalization purposes. Our goal is to theoretically derive an 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 the "background model" (used in speaker identification [18]) and its relation to our approach.

A. Likelihoods With the Use of Identification Trial Score Set Statistics

Suppose 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 the optimal combination method is to find

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

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

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

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

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

or, assuming all classes have the same prior probability

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

Given the assumption that 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, our problem is to find

$$\arg\max_k \prod_j p\left(\left\{s_i^j\right\}_{i=1,\ldots,N} |C_k\right). \quad (1)$$

The goal is to reliably estimate the densities $p\left(\left\{s_i^j\right\}_{i=1,\ldots,N} |C_k\right)$, which is a hard task given that the number N of classes is large and we do not have many samples of each class for training. Since we do not want to construct a class-specific combination method, the class indexes can be permuted. Thus, all of the training samples pertaining to different genuine classes can be used to train only one density $p\left(s_k^j, \left\{s_i^j\right\}_{i=1,\ldots,N,i\neq k} |C_k\right)$. Now s_k^j is a score belonging to a genuine match, and all other scores $\left\{s_i^j\right\}_{i=2,\ldots,N}$ are from impostor matches. In order to keep the problem tractable, instead of $p\left(s_k, \left\{s_i^j\right\}_{i=1,\ldots,N,i\neq k} |C_k\right)$, we can consider $p\left(s_k^j, t_k^j |C_k\right)$, where t_k^j is some statistic of all the other scores besides s_k^j. The final combination rule for this method is to find

$$\arg\max_k \prod_j p\left(s_k^j, t_k^j |C_k\right). \quad (2)$$

As our previous experiments have shown [3], this algorithm does not perform as well as the traditional likelihood ratio combination

$$\arg\max_k \prod_j \frac{p\left(s_k^j|C_k\right)}{p\left(s_k^j|\overline{C_k}\right)}. \quad (3)$$

One reason for the lower performance could be that the score set statistics t_k^j do not fully reflect the background information for score s_k^j, whereas the term $p\left(s_k^j|\overline{C_k}\right)$ contains such information. For example, the genuine matching scores s_k^j can be very strong, but located in the region of low probability (both $p\left(s_k^j|C_k\right)$ and $p\left(s_k^j, t_k^j|C_k\right)$ are small), whereas $p\left(s_k^j|\overline{C_k}\right)$ could be even smaller, and the likelihood ratio can still succeed. In Section IV-B, we will derive a combination rule which combines the use of the score set statistics and background models [21].

B. Likelihood Ratios With the Use of Identification Trial Score Set Statistics

We consider the posterior class probability, apply the Bayes formula as before, but now use the independence of classifiers to decompose the denominator

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

The next step is similar to the step in deriving the algorithm for the background speaker model [18]. We consider the class $\overline{C_k}$ to mean that some other class is genuine, and decompose $p\left(\left\{s_i^j\right\}_{i=1,\dots,N}\right) = P(C_k)p\left(\left\{s_i^j\right\}_{i=1,\dots,N}|C_k\right) + P(\overline{C_k})p\left(\left\{s_i^j\right\}_{i=1,\dots,N}|\overline{C_k}\right)$. By assuming that N is large and $P(\overline{C_k}) \gg P(C_k)$, we can discard the first term and represent 4 as

$$\frac{P(C_k)}{P\left(\overline{C_k}\right)^M}\prod_j \frac{p\left(\left\{s_i^j\right\}_{i=1,\dots,N}|C_k\right)}{p\left(\left\{s_i^j\right\}_{i=1,\dots,N}|\overline{C_k}\right)}.$$

Assuming that all classes have the same probability of occurring ($P(C_k) = P(C_i)$ and $P(\overline{C_k}) = P(\overline{C_i})$), we obtain the following classifier decision:

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

In comparison with decision 1 of the previous section, we have an additional density $p\left(\left\{s_i^j\right\}_{i=1,\dots,N}|\overline{C_k}\right)$. This density can be viewed as a background of impostors for the genuine class C_k. As research in the speaker identification suggests [21], utilizing such a background model is beneficial for system performance.

We estimate the ratios of (5) by additional modeling of $p\left(\left\{s_i^j\right\}_{i=1,\dots,N}|\overline{C_k}\right)$. We use an approach similar to the previous section to estimate this density as $p\left(s_k^j, t_k^j|\overline{C_k}\right)$ with t_k^j as the joint density of impostor scores s_k^j and corresponding identification trial statistics t_k^j. The final combination rule is then

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

The use of the identification trial score set statistics considers $p\left(s_k^j, t_k^j|C_k\right)$ and $p\left(s_k^j, t_k^j|\overline{C_k}\right)$ instead of $p\left(s_k^j|C_k\right)$ and $p\left(s_k^j|\overline{C_k}\right)$, and the background model considers $p\left(s_k^j, t_k^j|\overline{C_k}\right)$ or $p\left(s_k^j|\overline{C_k}\right)$ in addition to $p\left(s_k^j, t_k^j|C_k\right)$ or $p\left(s_k^j|C_k\right)$. Thus, the use of the identification trial score statistics differs from the background model in being able to account for dependencies of scores in identification trials by using the statistic t_k^j.

Note, that the traditional likelihood ratio ((3)) is the optimal combination method for low complexity combinations operating in verification mode (see [28]). Thus, its extension by (6) should provide a good combination method of medium II complexity type for verification mode operations.

C. Statistics of Identification Trial Scores

The important question which we have to decide is what particular identification trial score statistics t_k^j will be most suitable to replace the set of scores $\left\{s_i^j\right\}_{i=1,\dots,N,i\neq k}$. The likelihood ratio incorporating score statistics ((6)) will be more discriminating than the traditional likelihood ratio (3) if t_k^j provides an information on whether the score s_k^j is genuine or impostor. We use the term "'identification model"' to denote a particular way of choosing identification trial score set statistics t_k^j and using this statistic together with scores s_k^j.

One of the identification models we previously presented was the second best score model [29], where statistic $t_k^j = sbs\left(s_k^j\right)$ is calculated as the best score in the set $\left\{s_i^j\right\}_{i=1,\dots,N,i\neq k}$ ("second best" after s_k^j). We can reason that if the second best score is big (e.g., bigger than current score s_k^j, so s_k^j is not the best score), then we have less confidence that s_k^j is a genuine score, and more confidence that this is the impostor score. And if it is small (so s_k^j is big relative to all other scores), we have more confidence that s_k^j is genuine. Originally, we used $sbs\left(s_k^j\right)$ for accepting first-choice decisions in open-set identification systems [29]. In this case, $sbs\left(s_k^j\right)$ exactly coincides with second best score of the identification trial score set.

T-normalization can be considered as another identification model. It is expressed as a transformation of all matching scores s_k^j by the formula

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

where $\mu^j(l)$ and $\sigma^j(l)$ are correspondingly the mean and the standard deviation of the set of scores produced by matcher j during the identification trial l. In contrast to the second best score model, T-normalization uses different statistics—μ^j and σ^j which are the same for all scores s_k^j in the current identification trial, and it performrns predetermined transformation using these statistics.

Clearly, there might be many variations on calculating statistics t_k^j—it may or may not be dependent on k, it might include mean, variance, nth ranked score, or any other statistics of a score set. It seems that for different applications, the most useful statistics will be different, and it would be desirable to have an automatic way of determining useful score statistics. In our experiments, we limited ourselves to only using second best score statistics and T-normalization.

One approach to choose a best statistics of identification trial score sets is to look at the dependence between genuine and impostor scores. In order to verify the dependence of match scores, we measured the correlation between the genuine score and different statistics of the sets of impostor scores. Table I contains a small part of measured correlations corresponding to the $first_{imp}$—first ranked impostor score, $second_{imp}$—second ranked impostor score, and $mean_{imp}$—the mean of impostor scores. As the results of Table I show, the scores produced by real-life classifiers are indeed dependent.

The correlations between genuine and impostor set statistics indicate the usefulness of a given statistics—bigger correlation means that this statistic is better able to predict whether the score is genuine or not. So we might want to calculate such correlations for many different statistics and choose statistics with bigger correlations. Second best score statistics seems to provide a good prediction on the strength of genuine score, and this is the additional reason why we used it in our experiments. Note

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

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

that $sbs\left(s_k^j\right)$ used in our experiments is calculated with respect to s_k^j and if s_k^j is an impostor score, it might not be $first_{imp}$ or $second_{imp}$. During testing, we do not know what the exact set of impostors is, so instead of $first_{imp}$ or $second_{imp}$, we are forced to use $sbs\left(s_k^j\right)$.

D. 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 (1) and (6), we can see that each classifier contributes terms $p\left(\left\{s_i^j\right\}_{i=1,\dots,N}|C_k\right)$ and $p\left(\left\{s_i^j\right\}_{i=1,\dots,N}|C_k\right)/p\left(\left\{s_i^j\right\}_{i=1,\dots,N}|\overline{C_k}\right)$ correspondingly to the combination algorithm. Thus, one can conclude that it is possible to model the same terms for each classifier with the help of identification trial score statistics $p\left(s_k^j, t_k^j|C_k\right)$ and $p\left(s_k^j, t_k^j|C_k\right)/p\left(s_k^j, t_k^j|\overline{C_k}\right)$, 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 combination methods.

Another way to construct the combinations of medium II complexity type for dependent classifiers is presented in Section IV-E.

E. Normalizations Followed by Combinations and Single-Step Combinations

Fig. 5 represents, in graphical form, the type of combinations we have presented thus far. All of these combinations consist of two steps. In the first step, each score is normalized by using the other scores' output by the same matcher. In the second step, normalized scores are combined by using a predetermined or trained combination function.

Score normalization based on modeling the joint densities of scores and statistics, $p\left(s_k^j, t_k^j|C_k\right)$ and $p\left(s_k^j, t_k^j|C_k\right)/p\left(s_k^j, t_k^j|\overline{C_k}\right)$ might not correctly represent the original terms of (1) and (5), $p\left(\left\{s_i^j\right\}_{i=1,\dots,N}|C_k\right)$ and $p\left(\left\{s_i^j\right\}_{i=1,\dots,N}|C_k\right)/p\left(\left\{s_i^j\right\}_{i=1,\dots,N}|\overline{C_k}\right)$. Approximating densities might also be unreliable if few statistics are used.

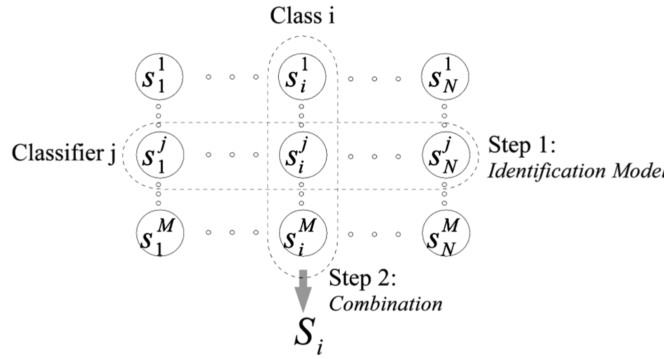


Fig. 5. Two-step combination method utilizing the identification model.

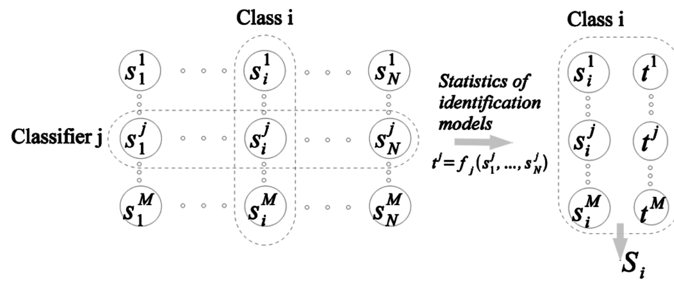


Fig. 6. One-step combination method utilizing the identification model.

However, it is not necessary to have the two steps for combinations. 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_i^j. Fig. 6 illustrates how scores and statistics from all of the participating classifiers could be combined in a single combination step.

In the algorithm described by Fig. 6, the statistics t^j are extracted for each classifier j using its output scores by a predetermined and nontrainable algorithm. The scores related to a 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 combination is of medium II complexity type. In comparison, for the low complexity-type combinations, only the scores for a particular class are combined, and statistics from other classes are not considered.

F. Neural Network and Weighted Sum Combinations Using Second-Best Score

As an example of single-step combinations, we consider two combination methods incorporating the second best score statistics: the neural network and the weighted sum rule.

The traditional neural network corresponding to the low complexity combination type can be represented as a function $S_i = f\left(s_i^1, \dots, s_i^M\right)$. Following the diagram of Fig. 6, the neural-network combination of medium II complexity type will have the form $S_i = f\left(s_i^1, sbs\left(s_i^1\right), \dots, s_i^M, sbs\left(s_i^M\right)\right)$. We used multilayer perceptron trained by a traditional backpropagation method and optimizing MSE.

The traditional weighted sum combination without the use of second-best score ('weighted sum local') is a low complexity combination which combines M scores from M biometric matchers assigned to a particular class i

$$S_i = w_1 s_i^1 + \cdots + w_M s_i^M. \quad (7)$$

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

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

The weighted sum rule can be specifically trained to maximize the correct identification rate in the identification mode of operation [28]. However, it is not optimal for the verification mode. Thus, we will test the performance of the weighted sum rule with and without the second-best score model modification for the identification-mode operation only. The neural network, on the other hand, might not be optimal for the identification mode due to MSE minimization criteria, but gives an output approximating likelihood ratio. We give the performance of the neural network method for identification and verification modes.

V. Experiments

We have used the biometric matching score set BSSR1 distributed by NIST [30]. This set contains matching scores for a fingerprint matcher and two face matchers "C" and "G." Fingerprint matching scores are given for the left index "li" finger matches and right index "ri" finger matches. For each combination method we performed, six sets of experiments on combining any two pairs of scores : "C" & "G", "li" & "ri", "li" & "C", "li" & "G", "ri" & "C", and "ri" & "G".

Although the BSSR1 score set has a subset of scores obtained from the same physical individuals, this subset is rather small—517 identification trials with 517 enrolled people. 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 a different physical person. Note that for pairs of face scores and for pairs of fingerprint scores, we retain the correspondence of scores to real people as specified in the database. 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 people—3000. Some enrollees and some identification trials had to be discarded due to enrollment errors. We use a bootstrap testing procedure: for 100 iterations, we randomly split the data in two parts—2991 identification trials with 2991 enrolled people in each part used as separate training and testing sets. The results of 100 training/testing iterations are averaged at the end.

In order to achieve good performance of training algorithms, all of the scores were normalized using a simple min-max algorithm to interval [1]. When we used T-normalization, additional min-max normalization was performed after it.

A. Description of Used Algorithms

The goal of our experiments is to compare three general architectures for classifier combination—traditional low complexity combinations which do not use any identification model, medium II complexity combinations based on T-normalized scores, and medium II complexity combinations using the second best score model. Three types of learning algorithms are used in the experiments: 1) likelihood ratio; 2) neural network; and 3) weighted sum. In order to make the comparison objective, we utilize each learning algorithm in each of three architectures. Each classifier in traditional and T-normalization methods supplies only a single score and the learning function depends on two input parameters $f(s^1, s^2)$. On the other hand, the second best score model has additional score statistics $sbs(s^1)$ and $sbs(s^2)$, and the learning function depends on four input parameters $f(s^1, sbs(s^1), s^2, sbs(s^2))$.

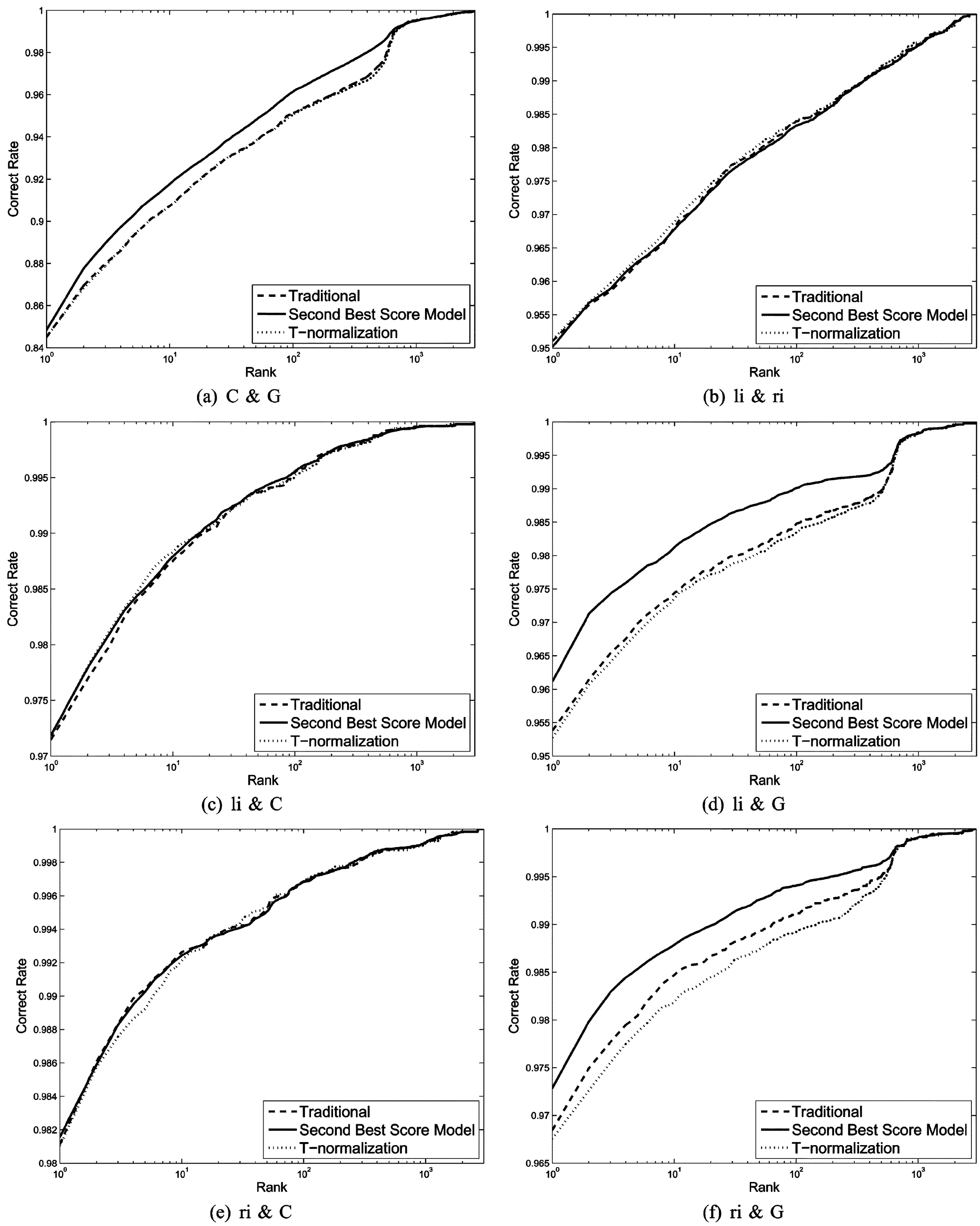
TABLE II
Correct Identification Rates of Combinations in Identification Systems. The Standard Deviations of These Rates Estimated From Bootstrap Samples are Given in Parentheses

Matchers	NN	NN+T	NN+sbs	WS	WS+T	WS+sbs
C & G	83.49	83.59	**83.86**	84.51	84.53	**84.85**
(σ)	(0.65)	(0.84)	(0.62)	(0.50)	(0.50)	(0.50)
li & ri	95.12	95.11	**95.17**	95.11	**95.13**	95.02
(σ)	(0.30)	(0.30)	(0.29)	(0.29)	(0.32)	(0.32)
li & C	96.44	**97.13**	96.21	97.15	97.17	**97.19**
(σ)	(0.93)	(0.24)	(0.78)	(0.23)	(0.23)	(0.25)
li & G	95.38	94.65	**95.73**	95.38	95.28	**96.12**
(σ)	(0.35)	(0.80)	(0.43)	(0.30)	(0.26)	(0.29)
ri & C	97.51	**98.10**	97.39	98.11	98.10	**98.16**
(σ)	(0.63)	(0.17)	(0.41)	(0.16)	(0.17)	(0.22)
ri & G	96.69	96.09	**97.03**	96.85	96.76	**97.29**
(σ)	(0.29)	(0.54)	(0.26)	(0.23)	(0.21)	(0.25)

For likelihood-ratio combinations, we estimate score densities using the Parzen window method with Gaussian kernels. The kernel width is determined by the maximum-likelihood method. We use only 1000 identification trial scores for reconstructing densities, and the remainder of the training set (2991–1000 trials) is used for validating kernel widths. Note that for each identification trial, there is one genuine score and 2990 impostor scores. In order to make our implementation faster, we only used a single random impostor score from a trial for training. We did not utilize the statistical independence of data when combining matchers of different modalities, and in all experiments, we approximated either 2-D densities of genuine and impostor scores—$p_{\text{gen}}(s^1, s^2)$ and $p_{\text{imp}}(s^1, s^2)$, or four dimensional densities—$p_{\text{gen}}(s^1, sbs(s^1), s^2, sbs(s^2))$ and $p_{\text{imp}}(s^1, sbs(s^1), s^2, sbs(s^2))$.

The neural network is multilayer perceptron trained by the backpropagation algorithm. The neural network has two hidden layers with eight and nine nodes and an output layer with one node in all cases. The input layer has two nodes for traditional training (no identification model) and T-normalized training, and four nodes for training with the second best score model. As for the likelihood-ratio method, we used 1000 training samples (one genuine and one random impostor score from the identification trial) for backpropagation training and remaining 2991-1000 training samples for validation. The training was stopped when the MSE on the validation set achieved minimum.

For the weighted sum methods, we need to find the optimal weights maximizing the number of correct identification trials



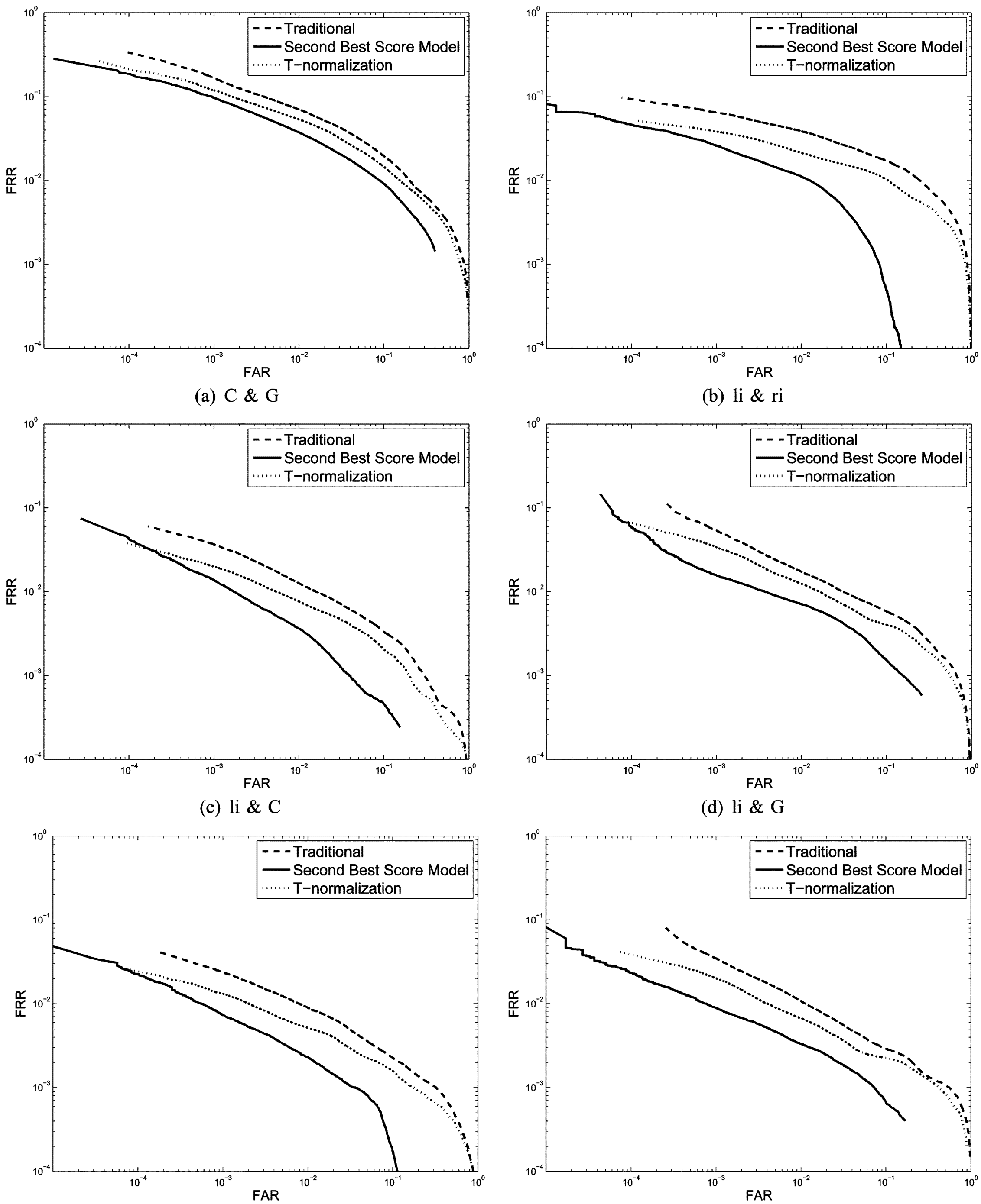
(a) C & G (b) li & ri (c) li & C (d) li & G (e) ri & C (f) ri & G

Fig. 7. CMC curves for weighted sum combinations utilizing and not utilizing identification models in the identification mode.

on the training sets. Though previous research proposing solutions for this problem exists (e.g., [31], [32]), it deals with the case of small number of classes and is not directly applicable to our case. The key idea of learning algorithms minimizing classification error is to replace the discrete misclassification cost function with some smooth approximation in order to be able to take a derivative of the cost function and perform gradient descent optimization. For our experiments, we implemented a simpler approach of random modification of weights and accepting new weights if classification performance improves. Though our approach takes more training time than the gradient descent method would have taken, it does not depend on the smoothing parameters and it is sufficiently fast.

B. Performance in Identification Operating Mode

Table II shows the obtained correct identification rate for experiments with the neural network and weighted sum combination methods. The correct identification means that the genuine combined score was better than 2990 impostor-combined scores (there is a total of 2991 enrollees). In this table, "NN" is the traditional neural-network combination method of low complexity type, "NN + T" is the neural network operating



(a) C & G (b) li & ri (c) li & C (d) li & G (e) ri & C (f) ri & G

Fig. 8. ROC curves for likelihood-ratio combinations utilizing and not utilizing identification models in verification mode.

on T-normalized scores and "$\mathrm{NN} + \mathrm{sbs}$" is the neural network augmented with the second-best score model. Similarly,"WS" is the traditional weighted sum combination of (7), "$\mathrm{WS} + \mathrm{T}$" is the weighted sum operating on T-normalized scores and "$\mathrm{WS} + \mathrm{sbs}$" is the weighted sum combination augmented with the second-best score model of (8).

We also provided the CMC graphs showing the performance of "WS," "$\mathrm{WS} + \mathrm{T}$," and "$\mathrm{WS} + \mathrm{sbs}$" methods in Fig. 7. As we discussed in Section IV-F, neural-network training is not optimized for best rank performance and we chose to not include similar CMC graphs for it.

We can see that in all cases, the addition of either the T-normalization or the second-best score statistic into the corresponding low complexity algorithm results in performance improvement. The weighted sum has generally better performance than the neural-network combination method, and the second best score statistics mostly outperform T-normalization.

C. Performance in Verification Operating Mode

Although there are examples where score normalization techniques with background models have been used for identification tasks [11], even more applications use such techniques for

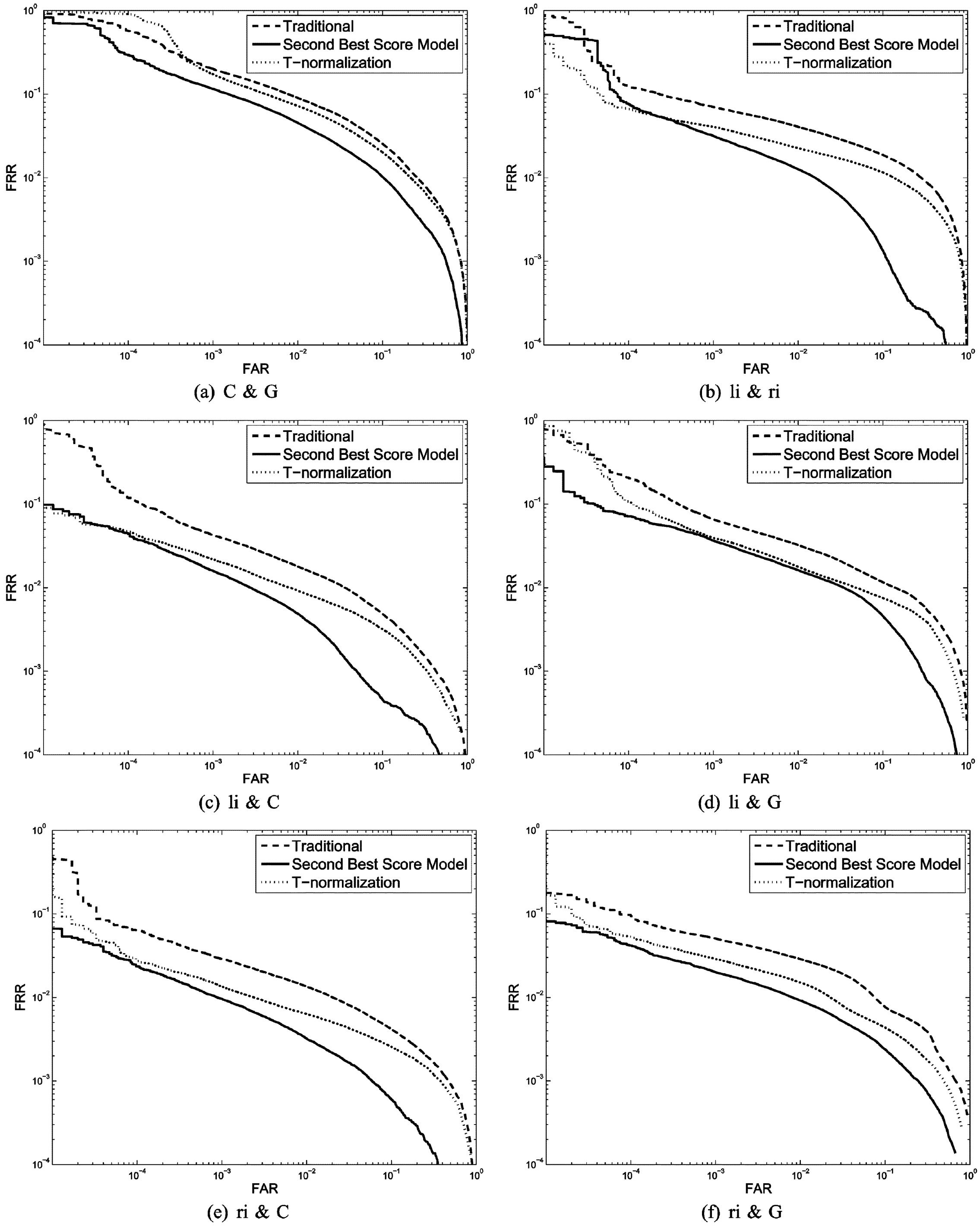


Fig. 9. ROC curves for neural-network combinations utilizing and not utilizing identification models in verification mode.

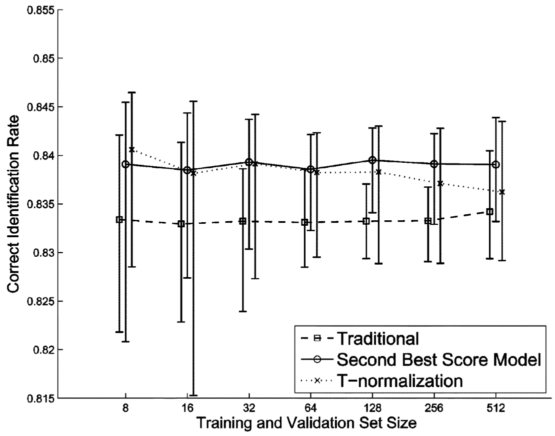
identification systems operating in verification mode [21], [18], [33]. We also applied the combinations utilizing identification models for biometric person verification tasks. The drawback of using either the background models or the second-best score statistic in verification tasks is that we have to produce not only one match per person and per matcher, but also some set of matching scores for other people enrolled in the system (or some artificially modeled people).

Figs. 8 and 9 contain the results of experiments when operating in the verification mode for the likelihood-ratio and neural-network combination methods. The ROC performance curves were constructed using combinations of 2991 $\times$ 100 (test trials $\times$ iterations) genuine and impostor score sets. Note that only a single random impostor was used from each test identification trial.

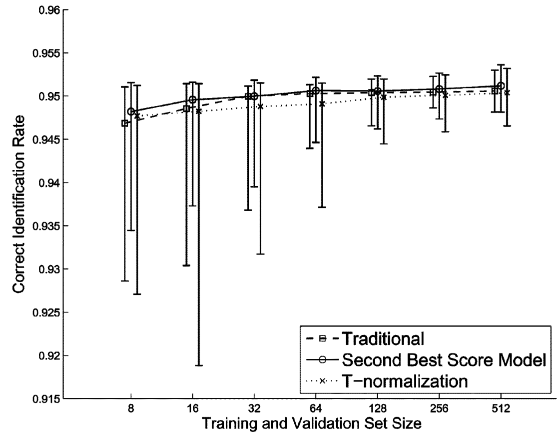
We were able to achieve significant improvement in the verification task performance as well by utilizing the second-best score statistic. The T-normalization is also beneficial, but to a smaller extent in these experiments.

D. Dependence of the Performance on the Number of Training Samples

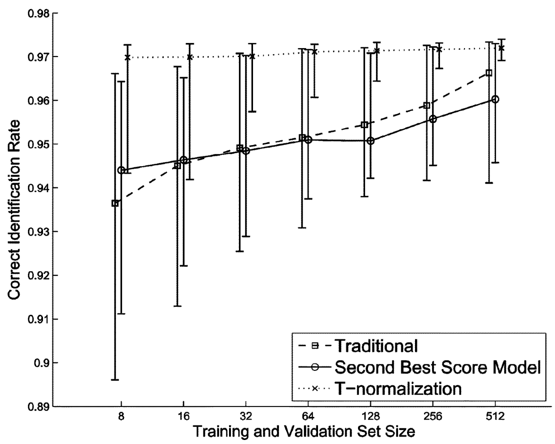
Since the use of the second best score model requires learning combination functions with a bigger number of parameters, the



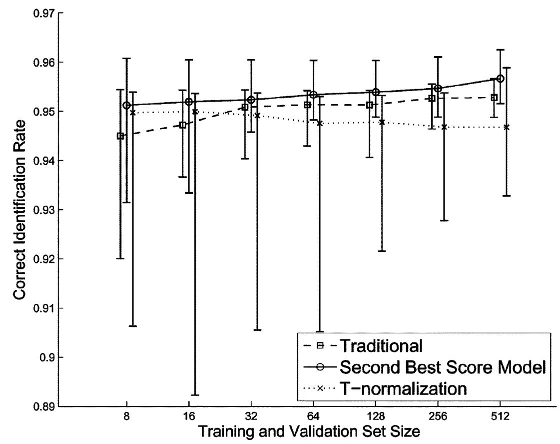
(a) C & G



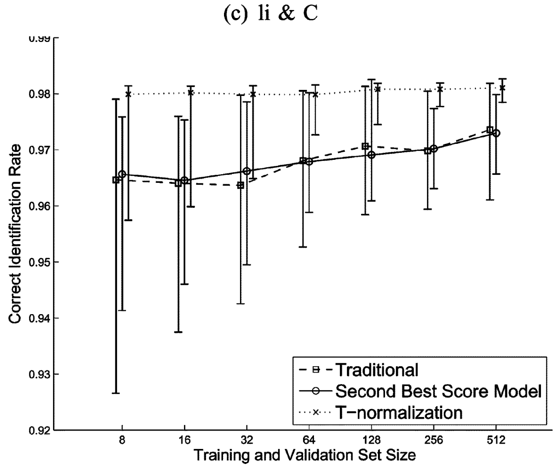
(b) li & ri



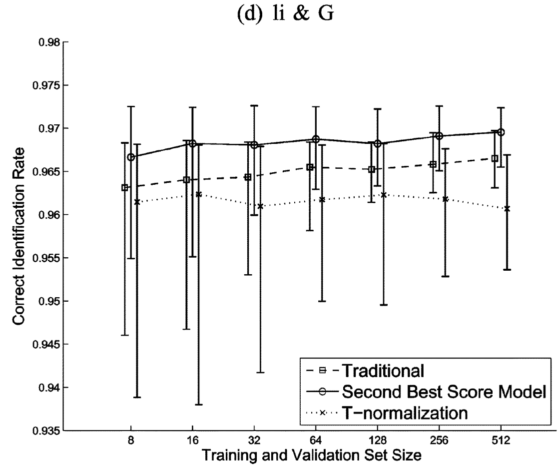
(c) li & C



(d) li & G



(e) ri & C



(f) ri & G

Fig. 10. Correct identification rates together with 90% bootstrap confidence estimates for different numbers of training and validation samples in neural-network combination methods.

errors associated with the learning algorithm might increase and negate the benefits of additional model information. In order to clarify the impact of additional training demand on the proposed methods, we conducted experiments with different numbers of training samples supplied to the learning algorithm. Figs. 10 and 11 present results of these experiments for neural-network and likelihood-ratio combination methods. Same numbers of training and validation samples are chosen here: $8, 16, \ldots, 512$.

Fig. 10 presents the correct identification rate together with 90% confidence intervals estimated from bootstrap samples (extreme 5% of bootstrap samples were discarded from each end) for neural-network combinations. The performance results agree with the results presented in Table II—combinations involving "C" are well handled by the T-normalization method, and combinations involving "G" have better performance when using the second best score model. The size of training and validation sets have little impact on the average correct identification rate, though it tends to slightly increase with the increasing size of these sets. The impact on the spread of rate measurements is more significant. If we want to avoid the accidental bad performance of a particular learned algorithm,

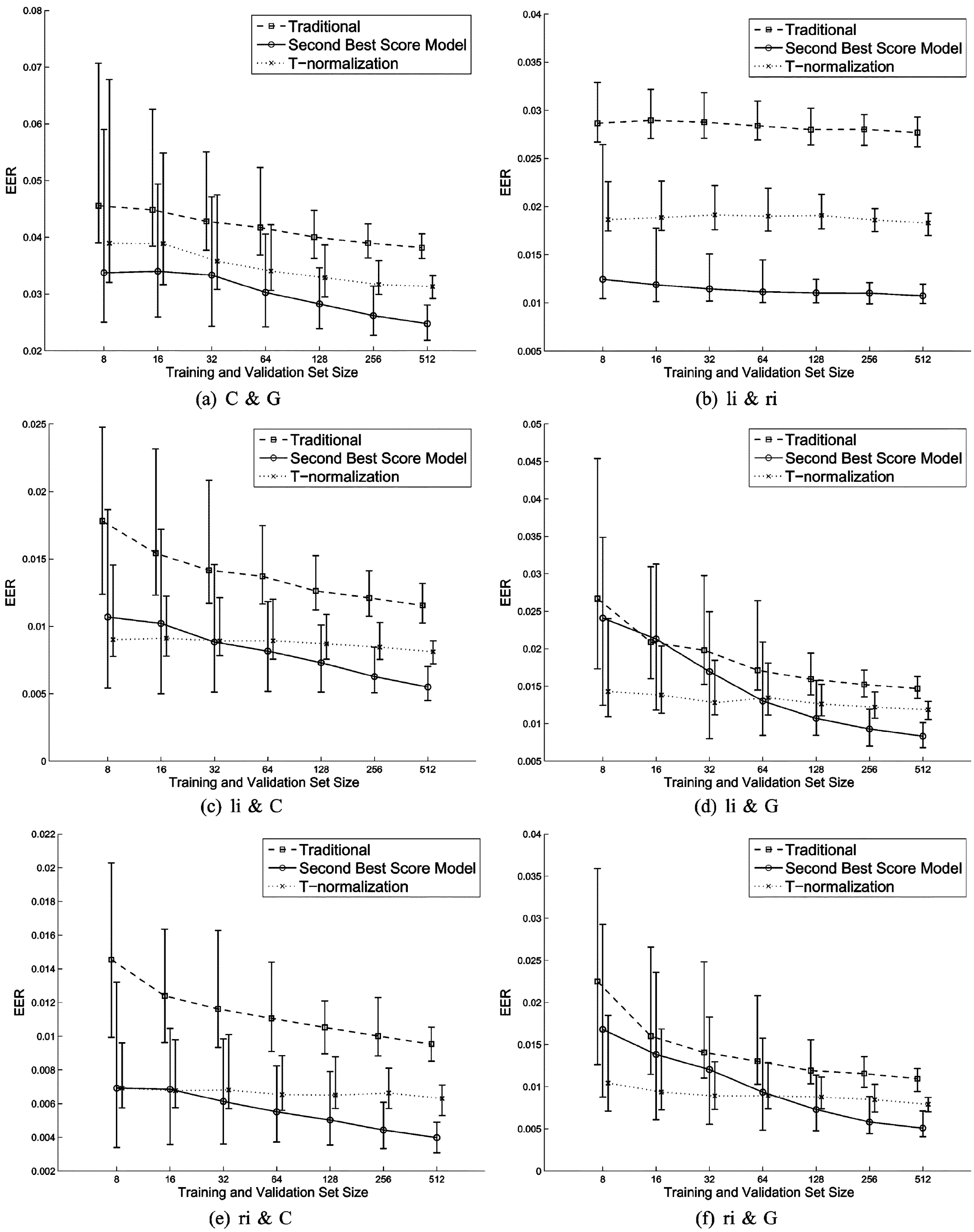


Fig. 11. Equal error rates (EERs) together with 90% bootstrap confidence estimates for different numbers of training and validation samples in likelihood-ratio combination methods.

we need to ensure that a sufficient number of training samples are used.

Fig. 11 presents the EERs together with 90% confidence intervals for likelihood-ratio combination methods. The increase of the training sample size has big impact on the spread of error rates, and a lesser impact on the average error rate. In some cases, the second best score model has worse performance than the T-normalization method when the number of training samples is small. For a larger number of training samples, the second best score model overtakes T-normalization. This observation confirms that learning the 4-D score densities for the second best score model can result in a worse performance than the approaches requiring learning 2-D densities, such as T-normalization. When the number of training samples is sufficiently large (more than 100 in this case), the density approximations for the second best score method are good enough to outperform T-normalization.

VI. Conclusion

We have presented four complexity combination types that originate naturally from the structure of the constructed combination method. We showed the usefulness of differentiating these four combination types to better understand the problem of classifier combination and for constructing well-performing combination algorithms. We observe that often the algorithms used for combining matchers in biometric identification systems only utilize the scores related to one class to produce the final combination score. Combination algorithms of low complexity type discard the dependency information between scores assigned to all classes by any single classifier. Instead of using low complexity combination algorithms in identification systems, we describe the use of medium II complexity-type combinations, which utilize all of the available scores and require the training of only a single combination function.

In order to use the relationships between scores assigned by one classifier to different classes, we have introduced the concept of the second-best score statistic. It is a way of score normalization where the normalization depends on all of the scores output by a classifier in any one identification trial, and the method is the same for all classes. This approach has less complexity than previous attempts of normalization [34], [35]. In these previous attempts, normalizations were class specific and required a huge amount of training data. The combinations utilizing such normalizations are similar to the behavior knowledge space combination [36], and belong to the high complexity combination type. Biometric identification problems can have a large number of enrolled people, and such combinations are not feasible due to the lack of training data. By restricting ourselves to nonclass-specific normalizations, we are able to concentrate on combinations of the medium II complexity type. Such combinations have significantly lower complexit and result in efficient algorithms for identification systems.

REFERENCES

[1] L. Xu, A. Krzyzak, and C. Y. Suen, "Methods for combining multiple classifiers and their applications to handwriting recognition," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 3, pp. 418–435, May/Jun. 1992.

[2] S. Tulyakov and V. Govindaraju, "Classifier combination types for biometric applications," presented at the IEEE Comput. Soc. Conf. Computer Vision and Pattern Recognition Workshop on Biometrics, New York, 2006.

[3] S. Tulyakov and V. Govindaraju, "Identification model for classifier combinations," presented at the Biometrics Consortium Conf., Baltimore, MD, 2006.

[4] N. Fan, J. Rosca, and R. Balan, "Speaker verification with combined threshold, identification front-end, and UBM," in *Proc. 4th IEEE Workshop on Automatic Identification Advanced Technologies*, 2005, pp. 112–117.

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

[6] Y. Lee, K. Lee, H. Jee, Y. Gil, W. Choi, D. Ahn, and S. Pan, "Fusion for multimodal biometric identification," in *Proc. Audio- and Video-Based Biometric Person Authentication*, 2005, pp. 1071–1079.

[7] A. Jain and A. Ross, "Learning user-specific parameters in a multibiometric system," in *Proc. Int. Conf. Image Processing*, 2002, vol. 1, pp. I–57.

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

[9] T. K. Ho, "A theory of multiple classifier systems and its application to visual word recognition," Ph.D. dissertation, State Univ. New York Buffalo, Buffalo, NY, 1992.

[10] T. K. Ho, J. Hull, and S. Srihari, "Decision combination in multiple classifier systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 1, pp. 66–75, Jan. 1994.

[11] R. Brunelli and D. Falavigna, "Person identification using multiple cues," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 10, pp. 955–966, Oct. 1995.

[12] L. Hong and A. Jain, "Integrating faces and fingerprints for personal identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 12, pp. 1295–1307, Dec. 1998.

[13] Y. Huang and C. Suen, "A method of combining multiple experts for recognition of unconstrained handwritten numerals," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 1, pp. 90–94, Jan. 1995.

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

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

[16] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, pp. 226–239, Mar. 1998.

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

[18] J. Mariethoz and S. Bengio, "A unified framework for score normalization techniques applied to text independent speaker verification," *IEEE Signal Process. Lett.*, vol. 12, no. 7, pp. 532–535, Jul. 2005.

[19] P. Grother, Face Recognition Vendor Test 2002 Supplemental Rep. NIST, 2004, Tech. Rep. NISTIR 7083.

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

[21] A. Rosenberg and S. Parthasarathy, "Speaker background models for connected digit password speaker verification," in *, Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, 1996, vol. 1, pp. 81–84.

[22] J. Colombi, J. Reider, and J. Campbell, "Allowing good impostors to test," in *Proc. 31st Conf. Rec. Signals, Systems Computers*, 1997, vol. 1, pp. 296–300.

[23] D.-S. Lee, "Theory of classifier combination: The neural network approach," Ph.D. dissertation, State Univ. New York Buffalo, Buffalo, NY, 1995.

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

[25] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Hoboken, NJ: Wiley, 2004.

[26] S. Tulyakov, "A complexity framework for combination of classifiers in verification and identification systems," Ph.D. dissertation, State Univ. New York (SUNY) Buffalo, Buffalo, NY, 2006.

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

[28] S. Tulyakov, V. Govindaraju, and C. Wu, "Optimal classifier combination rules for verification and identification systems," presented at the 7th Int. Multiple Classifier Systems, Prague, Czech Republic, 2007.

[29] S. Tulyakov and V. Govindaraju, "Combining matching scores in identification model," presented at the 8th Int. Conf. Document Analysis and Recognition, Seoul, Korea, 2005.

[30] "NIST biometric scores set." [Online]. Available: http://www.nist.gov/biometricscores/.

[31] B.-H. Juang and S. Katagiri, "Discriminative learning for minimum error classification [pattern recognition]," *IEEE Trans. Signal Process.*, vol. 40, no. 12, pp. 3043–3054, Dec. 1992.

[32] D. Miller, A. Rao, K. Rose, and A. Gersho, "A global optimization technique for statistical classifier design," *IEEE Trans. Signal Process.*, vol. 44, no. 12, pp. 3108–3122, Dec. 1996.

[33] A. Schlapbach and H. Bunke, "Using hmm based recognizers for writer identification and verification," presented at the 9th Int. Workshop on Frontiers in Handwriting Recognition, Tokyo, Japan, 2004.

[34] D. Bouchaffra, V. Govindaraju, and S. Srihari, "A methodology for mapping scores to probabilities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 9, pp. 923–927, Sep. 1999.

[35] K. Ianakiev, "Organizing multiple experts for efficient pattern recognition," Ph.D. dissertation, State Univ. New York Buffalo, Buffalo, NY, 2000.

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

Sergey Tulyakov received the Ph.D. degree in computer science and engineering from the University at Buffalo, Buffalo, NY, in 2006.

Currently, he is a Research Scientist at the Center for Unified Biometrics and Sensors, University at Buffalo. His research interests include handwriting recognition, fingerprint and face biometric person authentication, and combination of pattern classifiers.

Venu Govindaraju (F'06) received the B.Tech. (Hons.) degree from the Indian Institute of Technology (IIT), Kharagpur, India, in 1986, and the Ph.D. degree from the University of Buffalo, Buffalo, NY, in 1992.

Currently, he is a Professor of Computer Science and Engineering at the State University of New York at Buffalo (SUNY Buffalo). In a research career spanning more than 20 years, he has made significant contributions to many areas of pattern recognition which is a major branch in the field of artificial intelligence within computer science. Much of his early work focused on the automated recognition of written language, both machine-printed and handwritten text, and more recently, his research has expanded to information retrieval and biometrics. He has authored more than 270 scientific papers including 50 journal papers. His seminal work in handwriting recognition was at the core of the first handwritten address interpretation system used by the U.S. Postal Service. He was also the prime technical lead responsible for technology transfer to Lockheed Martin and Siemens Corporation for deployment by the U.S. Postal Service, Australia Post, and U.K. Royal Mail. He has been the Principal or Co-Principal Investigator of projects funded by government and industry for about U.S.$50 million. The Center for Unified Biometrics and Sensors (CUBS) that he founded in 2003 has since received more than U.S.$8 million of research funding covering several projects. He has given more than 75 invited talks and has supervised the dissertation of 17 doctoral students.

Dr. Govindaraju has served on the editorial boards of premier journals in his area and has chaired several technical conferences and workshops. He has won several awards for his scholarship including the prestigious MIT Global Technovator Award. He is a Fellow of the International Association of Pattern Recognition (IAPR).