Learning matching score dependencies for classifier combination

Sergey Tulyakov and Venu Govindaraju

Center for Unified Biometrics and Sensors (CUBS), SUNY at Buffalo, USA

The integration of recognition algorithms into a single document processing system might involve different available modules suitable for a single task. For example, we might possess few character or word recognition algorithms which all can be used in the system. One possible approach is to test these algorithms and to choose the one with the best performance. But practice shows that better approach is to try to use all available algorithms and to combine their outputs in order to achieve a better performance than any single algorithm. The combination problem consists in learning the behavior of given algorithms and deriving best possible combination function.

We assume that both the combined algorithms and the result of combination are classifiers. Thus a finite number of classes are distinguished in the problem, and the task is to find a class, which corresponds most to the input. As examples, classes might be a character set, a word lexicon, a person list, etc. Usually classifiers output the numeric matching scores corresponding to each class, and we will assume that these scores are available for combination. The combination algorithm is a function producing a final combined score for each class, and the final classifier selects class with the best combined score.

The purpose of this chapter is to investigate the different scenarios of combining classifiers, to show the difficulties in finding the optimal combination algorithms, and to present few possible approaches to combination problems. Generally, the classifier combination problem can be viewed as a construction of postprocessing classifier operating on the matching scores of combined classifiers. For many classifier combination problems, though, the number of classes or the number of classifiers and, consequently, the number of matching scores is too big, and applying generic pattern classification algorithms is difficult. Thus some scores are usually discarded from combination algorithm, or simplifying assumptions on score distributions are made and used in the combination algorithm. Though the dependency between classifiers is usually learned by the combination algorithms, the dependency between scores assigned to different classes by the same classifier is discarded. In this work we will show that accounting for score dependencies is essential for proper com-

bination of classifiers. The theory will be complemented by the experiments we perform on handwritten word recognizers and biometric person matchers.

1 Problem Description

Though the general theory presented in this chapter can be applied to any classifier combination task, we will mostly focus on two particular applications: handwritten word recognition and biometric person authentication. As a result, we are making few assumptions about combined classifiers. First we assume that each classifier assigns a matching score for each class, and we use these scores for combination. It would be convenient to call these classifiers 'matchers' or 'recognizers', in contrast to the general notion of classifiers making a decision and thus having selected class as their only output. Second, we assume that we only combine a small number of given matchers; in fact, for both applications we consider combinations of two matchers. Thus we separate ourselves from the so called 'classifier ensembles' having potentially large number of dynamically generated classifiers. Finally, we assume that the number of classes is large and may be variable. Indeed, the number of possible handwritten words defined by the corresponding lexicon or the number of enrolled persons in biometric database can be both large and variable. To be more specific, we describe both applications next.

1.1 Handwritten Word Recognizers

We consider the application of handwritten word recognizers in the automatic processing of United Kingdom mail. The destination information of the mail piece will usually contain the name of the postal town or county. After automatic segmentation of the mail piece image the goal of handwritten word recognizer is to match hypothesized town or county word image against a lexicon of possible names. Provided lexicon contains 1681 entries.

We use two handwritten word recognizers for this application: Character Model Recognizer (CMR)[6] and Word Model Recognizer (WMR)[11]. Both recognizers employ similar approaches to word recognition: they oversegment the word images, match the combinations of segments to characters and derive a final matching score for each lexicon word as a function of character matching scores. Still, the experiments (see Table 1) reveal that these matchers produce somewhat complementary results and their combination might be beneficial.

Our data consists of three sets of word images of approximately same quality (the data was provided as these three subsets and we did not regroup them). The images were manually truthed and only those images containing any of the 1681 lexicon words were retained. The word recognizers were run on these images and their match scores for all 1681 lexicon words were saved. Note, that both recognizers reject some lexicon entries if, for example, the

lexicon word is too short or too lengthy for presented image. We assume that in real systems such rejects will be dealt with separately (it is possible that the lexicon word corresponding to image truth will be rejected), but for our combination experiments we only keep scores of those lexicon words which are not rejected by any of the two recognizers. Thus for each image I_k we have a variable number N_k of score pairs (s_i^{cmr}, s_i^{wmr}), $i = 1, \ldots, N_k$ corresponding to non-rejected lexicon words. One of these pairs corresponds to the true word of the image and we will call these scores 'genuine', and other 'impostor' score pairs correspond to non-truth words.

After discarding images with non-lexicon words, and images where truth word was rejected by any recognizer, we are left with three sets of 2654, 1723 and 1770 images and related sets of score pairs. We will refer to the attempt of recognizing word image as identification trial. Thus each identification trial has a set score pairs (s_i^{cmr}, s_i^{wmr}), $i = 1, \ldots, N_k$ with one genuine score pair and $N_k - 1$ impostor pairs. The scores of each recognizer were also linearly normalized so that each score is in the interval $[0, 1]$ and bigger score means better match.

In order to get the general picture of the performance of considered recognizers we can count the numbers of identification trials where genuine score is better than all impostor scores of that trial. We summarized these counts in Table 1. The number of trials where first matcher (CMR) produced the genuine score bigger than all impostor scores is 3366, and second matcher (WMR) did the same 4744 times. Apparently, WMR has better performance, but still there are some identification trials ($5105 - 4744 = 361$), where CMR is correct and WMR is not. Since there is such distinction between recognizers, we strongly hope that their combination might achieve higher recognition rates.

Matchers	Total # of trials	1st matcher is correct	2nd matcher is correct	Both are correct	Either one is correct
CMR&WMR	6147	3366	4744	3005	5105
li&C	5982	4870	4856	3937	5789
li&G	5982	4870	4635	3774	5731

Table 1. Numbers of identification trials with any matcher having best score for the correct class.

Since our data was already separated into three subsets, we used this structure for producing training and testing sets. Each experiment was repeated three times, each time one subset is used as a training set, and two other sets are used as test sets. Final results are derived as averages of these three training/testing phases.

1.2 Biometric Person Matchers

We used biometric matching score set BSSR1 distributed by NIST[1]. 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. In this work we used both face matching scores and fingerprint 'li' scores and we do two types of combinations: 'li'&'C' and 'li'&'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[18] 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.

Table 1 shows the numbers of identification trials with genuine scores bigger than all impostor scores of that trial. The matchers now are more equal in strength and there is only a small number of trials where neither matcher correctly identified the genuine person.

2 Verification and Identification Tasks

Above described applications might include different operating scenarios. In one scenario the system generates a hypothesis of a true class of the input beforehand, and the task of the matchers is to verify if the input indeed of the hypothesized class. For example, a bank check recognition system might hypothesize about the value of the check based on the legal field, and numeric string recognition module must confirm that courtesy value coincides with the legal amount[7]. In biometric person verification systems a person presents a unique person identifier to the system, and biometric recognition module verifies if person's biometric scan matches the enrolled biometric template of claimed person's identity.

In another operating scenario a class of the input should be selected from a set of possible classes. Each lexicon word can be associated with a class for word recognition applications. In our considered application a set of UK postal town and county names serves as a lexicon for word recognizers. For biometric

person recognition a set of classes can coincide with the set of enrolled persons. The task of recognizer in this scenario is to select the class, which is the true class of input signal. We will assume that we deal with so called 'closed set identification', where the true class of input is included in the set of possible classes; in contrast 'open set identification' might not include true class in this set, and input needs to be rejected in this case.

We will call the system operating in the verification mode as verification system, and system operating in identification mode as identification system. Correspondingly, the problem solved by matchers or their combinations in the first case will be called verification task, and in the second case - identification task. Note that there could also be other operating scenarios involving considered matchers; as an example we have given open set identification.

2.1 Performance Measures

Different modes of operation demand different performance measures. For verification systems the performance is traditionally measured by means of Receiver Operating Characteristic (ROC) curves or by Detection Error Tradeoff (DET) curve. These curves are well suited for describing the performance of two-class pattern classification problems. In such problems there are two types of errors: the samples of first class are classified to belong to second class, and samples of second class are classified to be in first class. The decision to classify a sample to be in one of two classes is usually based on some threshold. Both performance curves show the relationship between two error rates with regards to a threshold (see [3] for precise definition of above performance measures).

In our case we will use ROC curves for comparing algorithm performance. If a matcher is used for verification task there are two classes: genuine if input belongs to the same hypothesized class, and impostor otherwise. The decision is traditionally based on the matching score of a recognizer assigned for hypothesis class.

For measuring performance of identification systems we will use ranking approach. In particular, we are interested in maximizing the rate of correctly identifying the input, first-rank-correct rate. If we look at identification task as a pattern classification problem, this performance measure will directly correspond to the traditional minimization of the classification error. Note that there are also other approaches to measure performance in identification systems[3], e.g. Rank Probability Mass, Cumulative Match Curve, Recall-Precision Curve. Though they might be useful for some applications, in our case we will be more interested in correct identification rate.

3 Verification Systems

The problem of combining matchers in verification systems can be easily solved with pattern classification approach. As we already noted, there are

two classes: genuine verification attempts and impostor verification attempts. The hypothesis class of the input is provided before matching. Each matcher j outputs a score s^j corresponding to a match confidence between input sample and hypothesis class. Assuming that we combine M classifiers, our task is to perform two-class classification (genuine and impostor) in M-dimensional score space $\{s^1, \ldots, s^M\}$. If the number of combined classifiers M is small, we will have no trouble in training pattern classification algorithm.

We employ the Bayesian risk minimization method as our classification approach[17]. This method states that the optimal decision boundaries between two classes can be found by comparing the likelihood ratio

$$f_{lr}(s^1, \ldots, s^M) = \frac{p_{gen}(s^1, \ldots, s^M)}{p_{imp}(s^1, \ldots, s^M)} \tag{1}$$

to some threshold θ where p_{gen} and p_{imp} are M-dimensional densities of score tuples $\{s^1, \ldots, s^M\}$ corresponding to two classes - genuine and impostor verification attempts. In order to use this method we have to estimate the densities p_{gen} and p_{imp} from the training data. For our applications the number of matchers M is 2 and the number of training samples is large (bigger than 1000), so we can successfully estimate these densities.

In our data each identification trial has one genuine and $N_k - 1$ impostor score pairs, so the total number of genuine score pairs is $T = K$ (K is the number of identification trials in the training set) and the total number of impostor score pairs is $T = \sum_{k=1}^{K}(N_k - 1)$. We approximate both densities as the sums of 2-dimensional gaussian Parzen kernels

$$\hat{p}(s^1, s^2) = \frac{1}{T}\sum_{t=1}^{T} \frac{1}{2\pi\sigma^2} e^{-\frac{(s^1 - s_t^1)^2 + (s^2 - s_t^2)^2}{2\sigma^2}}$$

where $\{s_t^1, s_t^2\}_{t=1,\ldots,T}$ are the set of training score pairs. The window parameter σ is estimated by the maximum likelihood method on the training set[16] using leave-one-out technique. Note that σ is different for genuine and impostor density approximations.

For a given threshold θ we calculate the number of misidentified samples from the test data set of each class. The genuine samples (s^1, s^2) are misidentified as impostor samples if $\hat{f}_{lr}(s^1, s^2) = \frac{\hat{p}_{gen}(s^1, s^2)}{\hat{p}_{imp}(s^1, s^2)} < \theta$ (false rejects), and impostor samples misidentified as genuine if $\hat{f}_{lr}(s^1, s^2) \geq \theta$ (false accepts). Thus for each θ we calculate false reject and false accept rates, $FRR(\theta)$ and $FAR(\theta)$, and construct ROC curve, which is a graph of $FRR(\theta)$ versus $FAR(\theta)$. The resulting ROC curves for original matchers and for their combinations with likelihood ratio method are shown in Figures 1, 2 and 3.

As we expected, the combination has better performance than any of the individual matchers. Biometric matchers are based on different modalities and thus better complement each other than word recognizers. This is indicated

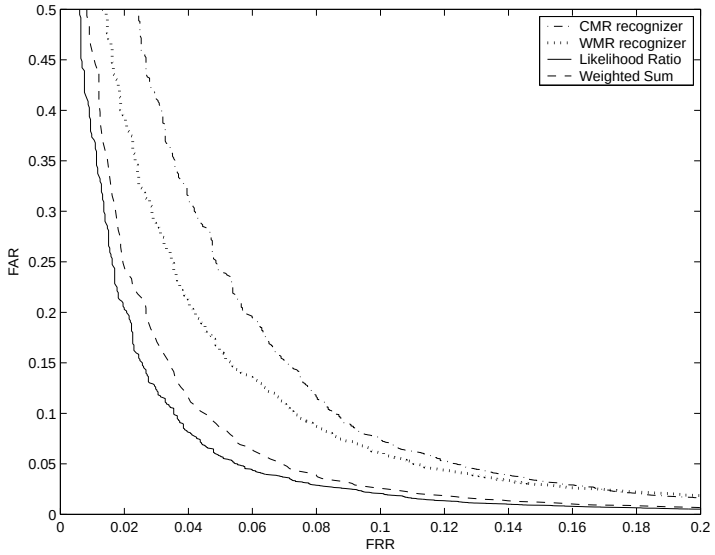


Fig. 1. ROC curves for two handwritten word recognizers (WMR and CMR) and their combinations by likelihood ratio and weighted sum methods.

by the performance graphs: the improvement is bigger in the case of biometric matchers.

The likelihood ratio combination method is theoretically optimal for verification systems and its performance only limited by our ability to correctly estimate score densities. The density estimation is known to be a difficult task; working with many-dimensional data, having heavy tailed distributions or discreteness in the data can lead to very poor density estimates. In our experiments we had sufficient number of training samples in 2-dimensional space and the task was relatively easy, but still we had to make adjustments for the discreteness of fingerprint scores represented by the integer numbers in the range $0 - 350$.

Since our problem is the separation of genuine and impostor classes, we could apply many existing pattern classification techniques. For example, support vector machines have shown good performance in many tasks, and can be definitely used to improve the likelihood ratio method. In [20] we performed some comparisons of likelihood ratio method with SVMs on an artificial task and found that on average (over many random training sets) SVMs do have slightly better performance, but for a particular training set it might not be true. The difference in performance is quite small and decreases with the increasing number of training samples. Also note that many pattern classification algorithms provide only a single decision boundary (separating hyperplane in the kernel mapped space for SVMs), and this effectively results in

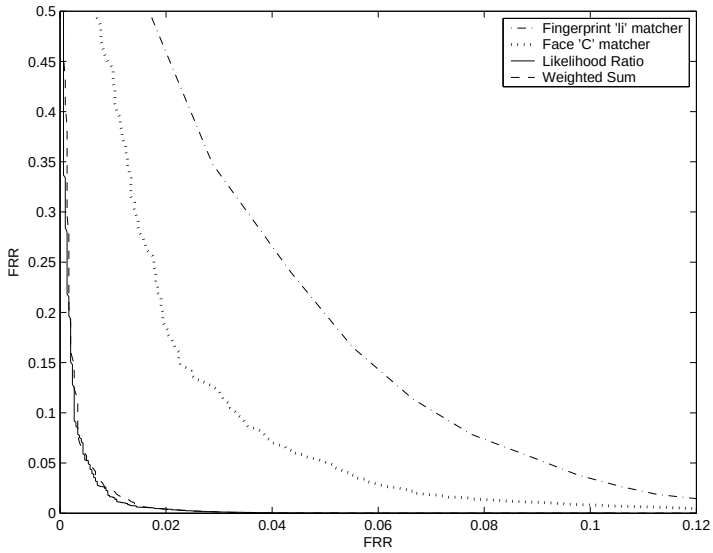


Fig. 2. ROC curves for two biometric matchers (fingerprint 'li' and face 'C') and their combinations by likelihood ratio and weighted sum methods.

the single point of FAR-FRR plane instead of ROC curve. The advantage of likelihood ratio combination method is that we get the whole range of solutions by varying threshold parameter θ and which are represented by ROC curve.

4 Identification Systems

In identification systems a hypothesis of the input sample is not available and we have to choose the input's class among all possible classes. Denote N as the number of classes. The total number of matching scores available for combination now is MN: N matching scores for each class from each of M combined classifiers. If numbers M and N are not big, then we can use generic pattern classifiers in MN-dimensional score space to find the input's class among N classes. For some problems, e.g. digit or character recognition, this is an acceptable approach; the number of classes is small and usually there is a sufficient number of training samples to properly train pattern classification algorithms operating in MN score space.

But for our applications in handwritten word recognition and biometric person identification the number of classes is too big and the number of training samples is too small (there might be even no training samples at all for a particular lexicon word), so the pattern classification in the MN-dimensional score space seems to be out of the question. The traditional approach in this

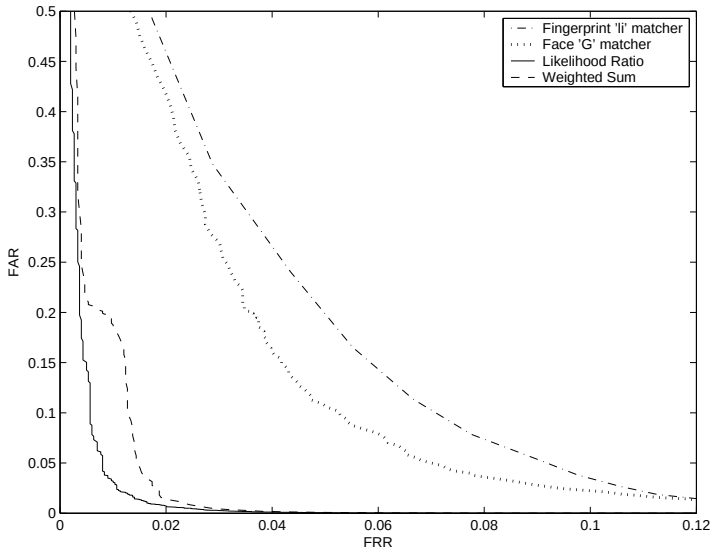


Fig. 3. ROC curves for two biometric matchers (fingerprint 'li' and face 'G') and their combinations by likelihood ratio and weighted sum methods.

situation is to use some combination rules. The combination rule implies the use of some combination function f operating only on M scores corresponding to one class, $f(s^1, \ldots, s^M)$, and it states that the decision class C is the one which maximizes the value of a combination function:

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

Note that in our notation the upper index of the score corresponds to the classifier, which produced this score, and lower index corresponds to the class for which it was produced. The names of combination rules are usually directly derived from the names of used combination functions: the sum function $f(s^1, \ldots, s^M) = s^1 + \cdots + s^M$ corresponds to the sum rule, the product function $f(s^1, \ldots, s^M) = s^1 \ldots s^M$ corresponds to the product rule and so on.

Many combination rules have been proposed so far, but there is no agreement on the best one. It seems that different applications require different combination rules for best performance. Anyone wishing to combine matchers in real life has to test few of them and choose the one with best performance. Combination rules are also frequently used for verification problems to find the final score, which is compared with threshold and the decision is based on this comparison. But there is no real need to do it - the plethora of pattern classification algorithms is available for solving combinations in verification problems.

Our main interest in this chapter is to investigate the problem of finding the optimal combination function for identification systems. This problem appears to be much more difficult in comparison to combinations in verification systems.

4.1 Likelihood Ratio Combination Rule

As we already know, likelihood ratio function is the optimal combination function for verification systems. We want to investigate whether it will be optimal for identification systems. Suppose we performed a match of the input sample by all M matchers against all N classes and obtained MN matching scores $\{s_i^j\}_{i=1,\ldots,N;j=1,\ldots,M}$. Assuming equal prior class probabilities, the Bayes decision theory states that in order to minimize the misclassification rate the sample should be classified as one with highest value of likelihood function $p(\{s_i^j\}_{i=1,\ldots,N;j=1,\ldots,M}|\omega_i)$. Thus, for any two classes ω_1 and ω_2 we have to classify input as ω_1 rather than ω_2 if

$$p(\{s_i^j\}_{i=1,\ldots,N;j=1,\ldots,M}|\omega_1) > p(\{s_i^j\}_{i=1,\ldots,N;j=1,\ldots,M}|\omega_2) \tag{3}$$

Let us make an assumption that the scores assigned to each class are sampled independently from scores assigned to other classes; scores assigned to genuine class are sampled from M-dimensional genuine score density, and scores assigned to impostor classes are sampled from M-dimensional impostor score density:

$$\begin{aligned} &p(\{s_i^j\}_{i=1,\ldots,N;j=1,\ldots,M}|\omega_i) \\ &= p(\{s_1^1,\ldots,s_1^M\},\ldots,\{s_{\omega_i}^1,\ldots,s_{\omega_i}^M\},\ldots,\{s_N^1,\ldots,s_N^M\}|\omega_i) \\ &= p_{imp}(s_1^1,\ldots,s_1^M)\ldots p_{gen}(s_{\omega_i}^1,\ldots,s_{\omega_i}^M)\ldots p_{imp}(s_N^1,\ldots,s_N^M) \end{aligned} \tag{4}$$

After substituting 4 into 3 and canceling out common factors we obtain the following inequality for accepting class ω_1 rather than ω_2:

$$p_{gen}(s_{\omega_1}^1,\ldots,s_{\omega_1}^M)p_{imp}(s_{\omega_2}^1,\ldots,s_{\omega_2}^M) > p_{imp}(s_{\omega_1}^1,\ldots,s_{\omega_1}^M)p_{gen}(s_{\omega_2}^1,\ldots,s_{\omega_2}^M)$$

or

$$\frac{p_{gen}(s_{\omega_1}^1,\ldots,s_{\omega_1}^M)}{p_{imp}(s_{\omega_1}^1,\ldots,s_{\omega_1}^M)} > \frac{p_{gen}(s_{\omega_2}^1,\ldots,s_{\omega_2}^M)}{p_{imp}(s_{\omega_2}^1,\ldots,s_{\omega_2}^M)} \tag{5}$$

The terms in each part of the above inequality are exactly the values of the likelihood ratio function f_{lr} taken at the sets of scores assigned to classes ω_1 and ω_2. Thus, the class maximizing the MN-dimensional likelihood function of inequality 3 is the same as a class maximizing the M-dimensional likelihood ratio function of inequality 5. The likelihood ratio combination rule is the optimal combination rule under used assumptions.

Table 2 shows the performance of this rule on our data sets. Whereas the combinations of biometric matchers have significantly higher correct identification rates than single matchers, the combination of word recognizers has

Matchers	1st matcher is correct	2nd matcher is correct	Either one is correct	Likelihood Ratio Rule	Weighted Sum Rule
CMR&WMR	3366	4744	5105	4293	5015
li&C	4870	4856	5789	5817	5816
li&G	4870	4635	5731	5737	5711

Table 2. Correct identification rate for likelihood ratio and weighted sum combination rules.

lower correct identification rate than a single WMR matcher. This fact is rather surprising: the calculation of the combined scores by the likelihood ratio is exactly the same as we did for combinations in verification systems which gave us significant improvements in all cases (Figures 1, 2 and 3).

Few questions arise after reviewing the results of these experiments:

- If likelihood ratio combination rule was not able to improve correct identification rate of word recognizers, is there any other rule which will succeed?
- What are the reasons for the failure of seemingly optimal combination rule?
- What is the true optimal combination rule, and can we devise an algorithm of learning it from the training data?

In the rest of this chapter we will investigate these questions.

4.2 Weighted Sum Combination Rule

One of the frequently used rules in classifier combination problems is the weighted sum rule with combination function $f(s^1, \ldots, s^M) = w_1 s^1 + \cdots + w_M s^M$. The weights w_j can be chosen heuristically with the idea that better performing matchers should have bigger weight, or they can be trained to optimize some criteria. In our case we train the weights so that the number of successful identification trials on the training set is maximized. Since we have two matchers in all configurations we use brute-force method: we calculate the correct identification rate of combination function $f(s^1, s^2) = ws^1 + (1-w)s^2$ for different values of $w \in [0, 1]$, and find w corresponding to highest rate.

The numbers of successful identification trials on the test sets is presented in Table 2. In all cases we see an improvement over the performances of single matchers. The combination of word recognizers is now successful and is in line with the performance of other combinations of matchers.

We also investigated the performance of this method in the verification task. Figures 1, 2 and 3 contain ROC curves of the weighted sum rule used in verification task with the same weights as in identification experiments. In all cases we get slightly worse performance from the weighted sum rule than from the likelihood ratio rule. This confirms our assertion that the likelihood ratio is the optimal combination method for verification systems.

4.3 Explaining Identification System Behavior

The main assumption that we made while deriving likelihood ratio combination rule in section 4.1 is that the score samples in each identification trial are independent. That is, genuine score is sampled from genuine score distribution and is independent from impostor scores which are independent and identically distributed according to impostor score distribution. We can verify if this assumption is true for our matchers.

Matchers	$first_{imp}$	$second_{imp}$	$third_{imp}$	$mean_{imp}$
CMR	0.4359	0.4755	0.4771	0.1145
WMR	0.7885	0.7825	0.7663	0.5685
li	0.3164	0.3400	0.3389	0.2961
C	0.1419	0.1513	0.1562	0.1440
G	0.1339	0.1800	0.1827	0.1593

Table 3. Correlations between s_{gen} and different statistics of the impostor score sets produced during identification trials for considered matchers.

Table 3 shows correlations between genuine score and some functions of the impostor scores obtained in the same identification trial. $first_{imp}$ column has correlations between genuine and the best impostor score, $second_{imp}$ and $third_{imp}$ consider second-best and third-best impostor scores, and $mean_{imp}$ has correlations between the mean of all impostor scores obtained in an identification trial and a genuine score. Non-zero correlations indicate that the scores are dependent. The correlations are especially high for word recognizers, and this might be the reason why the likelihood ratio combination rule performed poorly there.

The dependence of matching scores obtained during a single identification trial is usually not taken into account. One of the reasons might be that as a rule all matching scores are derived independently from each other: the same matching process is applied repeatedly to all enrolled biometric templates or all lexicon words, and the matching score for one class is not influenced by the presence of other classes or the matching scores assigned to other classes. So it might seem that the matching scores are independent, but it is rarely true. The main reason for this is that all matching scores produced during identification trial are derived using the same input signal. For example, a fingerprint matcher, whose matching score is derived from the number of matched minutia in enrolled and input fingerprint, will produce low scores for all enrolled fingerprints if the input fingerprint has only few minutias.

The next three examples will illustrate the effect of score dependences on the performance of identification systems. In particular, second example confirms that if identification system uses likelihood ratio combination, then its performance can be worse than the performance of a single matcher.

Example 1

Suppose we have an identification system with one matcher and, for simplicity, $N = 2$ classes. During each identification attempt a matcher produces two scores corresponding to two classes, and, since by our assumption the input is one of these two classes (closed set identification), one of these scores will be genuine match score, and another will be impostor match score. Suppose we collected a data on the distributions of genuine and impostor scores and reconstructed score densities (let them be gaussian) as shown in Figure 4.

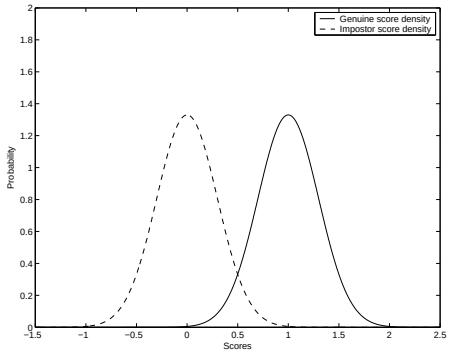


Fig. 4. Hypothetical densities of matching(genuine) and non-matching(impostors) scores.

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

1. Both scores s_{gen} and s_{imp} are sampled independently from genuine and impostor distributions.
2. In every observed identification attempt : $s_{imp} = s_{gen} - 1$. Thus in this scenario the identification system always correctly places genuine sample on top. There is a strong dependency between scores given to two classes, and score distributions of Figure 4 do not reflect this fact.

If a system works in verification mode and we have only one match score to make a decision on accepting or rejecting input, we can only compare this score to some threshold. By doing so both scenarios would have same performance: the rate of false accepts (impostor samples having match score higher than threshold) and the rate of false rejects (genuine samples having match score lower than threshold) will be determined by integrating impostor and genuine densities of Figure 4 no matter what scenario we have. If system works in identification mode, the recognizer of the second scenario will be a clear winner: it is always correct while the recognizer of first scenario can make mistakes and place impostor samples on top.

This example shows that the performance of the matcher in the verification system might not predict its performance in the identification system. Given two matchers, one might be better for verification systems, and another for identification systems.

Example 2

Consider a combination of two matchers in two class identification system: one matcher is from the first scenario, and the other is from the second scenario. Assume that these matchers are independent. Let the upper score index refer to the matcher producing this score; s_i^j is the score for class i assigned by the classifier j. From our construction we know that the second matcher always outputs genuine score on the top. So the optimal combination rule for identification system will simply discard scores of first matcher and retain scores of the second matcher:

$$f(s^1, s^2) = s^2 \tag{6}$$

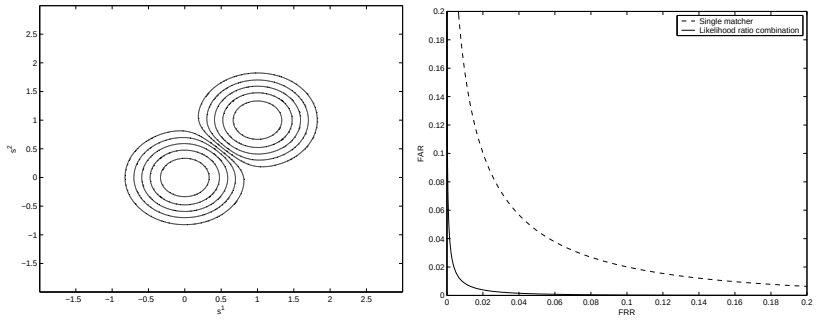
The input will always be correctly classified as $\arg\max_i s_i^2$.

Let us now use the likelihood ratio combination rule for this system. Since we assumed that matchers are independent, the densities of genuine $p_{gen}(s^1, s^2)$ and impostor $p_{imp}(s^1, s^2)$ scores are obtained by multiplying corresponding one-dimensional score densities of two matchers. In our example, impostor scores are distributed as a Gaussian centered at $(0, 0)$, and genuine scores are distributed as a Gaussian centered at $(1, 1)$. Figure 5(a) contains the contours of function $|p_{gen} - p_{imp}|$ which allows us to see the relative position of these gaussians. The gaussians have same covariance matrix, and thus the optimal decision contours are hyperplanes[17] - lines $s^1 + s^2 = c$. Correspondingly, the likelihood ratio combination function is equivalent to the combination function $f = s^1 + s^2$ (note, that true likelihood ratio function will be different, but if two functions have same contours, then their combination rules will be the same). Such combination improves the performance of the verification system relative to any single matcher; Figure 5(b) shows corresponding ROC curves for any single matchers and their combination.

Suppose that (s_1^1, s_1^2) and (s_2^1, s_2^2) are two score pairs obtained during one identification trial. The likelihood ratio combination rule classifies the input as a class maximizing likelihood ratio function:

$$\arg\max_{i=1,2} \frac{p_{gen}(s_i^1, s_i^2)}{p_{imp}(s_i^1, s_i^2)} = \arg\max_{i=1,2} \; s_i^1 + s_i^2 \tag{7}$$

Let the test sample be $(s_1^1, s_1^2) = (-0.1, 1.0)$, $(s_2^1, s_2^2) = (1.1, 0)$. We know from our construction that class 1 is the genuine class, since the second matcher assigned score 1.0 to it and 0 to the second class. But the class 2 with scores $(1.1, 0)$, has combined score $s_2^1 + s_2^2 = 1.1 + 0 = 1.1$, which is bigger than combined score for class 1, $s_1^1 + s_1^2 = -0.1 + 1.0 + 0 = 0.9$. Hence class 2 has bigger ratio of genuine to impostor densities than class 1, and the likelihood



(a) (b)

Fig. 5. (a) Two-dimensional distributions of genuine and impostor scores for examples 2 and 3 (b) ROC curves for single matchers and their likelihood ratio combination.

ratio combination method would incorrectly classify class 2 as the genuine class.

Thus the optimal for verification system likelihood ratio combination rule (7) has worse performance than a single second matcher. On the other hand, the optimal for identification system rule (6) does not improve the performance of the verification system. Recall, that in section 4.1 we showed that if scores assigned by matchers to different classes are independent, then likelihood ratio combination rule is optimal for identification systems, as well as for verification systems. Current example shows that if there is a dependency between scores, this is no longer a case, and the optimal combination for identification systems can be different from the optimal combination for verification systems.

It seems that this example is analogous to our experiments with the combination of word recognizers. Our better performing word recognizer, WMR, has strong dependence between scores assigned to different classes (Table 3), and the resulting combination by likelihood ratio rule has worse performance than WMR's.

Example 3

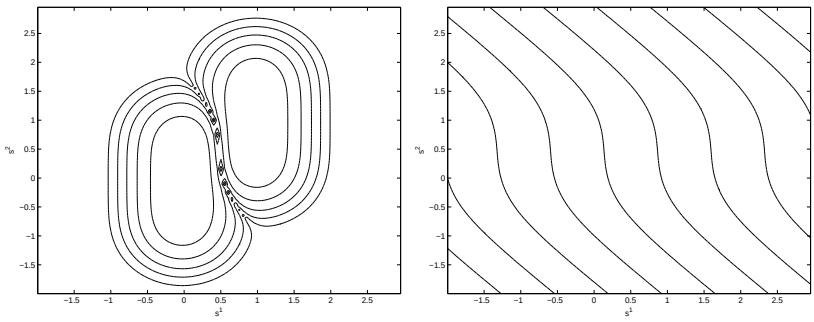
The problem of finding optimal combination function for verification systems was a relatively easy task: we needed to approximate the densities of genuine and impostor scores and take their ratio. It turns out that the problem of finding optimal combination function for identification systems is considerably more difficult - we are not able to express it in such simple form. In fact, it is even difficult to construct an artificial example where we would know what this function is. Here we consider one such example.

Let X_{gen}, X_{imp} and Y be independent two-dimensional random variables, and suppose that genuine scores in our identification system are sampled as

a sum of X_{gen} and Y: $\mathbf{s_{gen}} = \mathbf{x_{gen}} + \mathbf{y}$, and impostor scores are sampled as a sum of X_{imp} and Y: $\mathbf{s_{imp}} = \mathbf{x_{imp}} + \mathbf{y}$, $\mathbf{x_{gen}} \sim X_{gen}$, $\mathbf{x_{imp}} \sim X_{imp}$ and $\mathbf{y} \sim Y$, bold symbols here denote two-dimensional vector in the space (s^1, s^2). The variable Y provides the dependence between scores in identification trials; we assume that its value $\mathbf{y}$ is the same for all scores in one identification trial.

Let X_{gen} and X_{imp} have gaussian densities $p_{X_{gen}}(s^1, s^2)$ and $p_{X_{imp}}(s^1, s^2)$ as in the previous example and in the Figure 5(a). For any value of $\mathbf{y}$ conditional densities of genuine and impostor scores $p_{X_{gen}+Y|Y=\mathbf{y}}(s^1, s^2)$ and $p_{X_{imp}+Y|Y=\mathbf{y}}(s^1, s^2)$ are also gaussian and independent. As we discussed in the previous example, the likelihood ratio combination rule results in the combination function $f(s^1, s^2) = s^1 + s^2$, and this rule will be optimal for every identification trial and its associated value $\mathbf{y}$. The rule itself does not depend on the value of $\mathbf{y}$, so we can use it for every identification trial, and this is our optimal combination rule for identification system.

On the other hand, this rule might not be optimal for the verification system defined by the above score distributions. For example, if Y is uniformly distributed on the interval $0 \times [-1, 1]$, then the distributions of genuine and impostor scores $X_{gen}+Y$ and $X_{imp}+Y$ will be as shown in the Figure 6(a) and the optimal combination rule separating them will be as shown in the Figure 6(b). By changing the distribution of Y and thus the character of dependence between genuine and impostor scores we will also be changing optimal combination rule for verification system. At the same time, the optimal combination rule for identification system will stay the same - $f(s^1, s^2) = s^1 + s^2$.



(a) (b)

Fig. 6. (a) Two-dimensional distributions of genuine and impostor scores for example 3 (b) Contours of the likelihood ratio combination function.

If we knew only the overall score distributions as in the Figure 6(a) we would not have enough information to find the optimal combination function for identification system. If score vectors having distributions of Figure 6(a) are in its own turn are independent, then likelihood ratio combination of Figure 6(b) will be optimal for identification system. Or, if scores are generated

by the initial construction, linear combination function is the optimal one. Thus, there could be different optimal combination functions for identification systems with scores distributed as in the Figure 6(a), and the difference is determined by the nature of the score dependencies in identification trials.

5 Estimating Optimal Combination Function for Identification Systems

As we saw in the example 3 of the previous section, it is rather difficult to say from the training samples what is the optimal combination function for the identification system. The densities of genuine and impostor matching scores are of little help, and might be useful only if the scores in identification trials are independent. For dependent scores we have to consider the scores in each identification trial as a single training sample, and train the combination function on these samples.

This was precisely the technique we used to train the weighted sum rule for identification systems in section 4.2. For each training identification trial we checked whether the genuine score pair produced bigger combined scores than all impostor score pairs. By counting the numbers of successful trials we were able to choose the proper weights.

Though the weighted sum rule provides a reasonable performance in our applications, its decision surfaces are linear and might not completely separate generally non-linear score distributions. We might want our combination function to be more complex, trained with available training set and possibly approaching ideal optimal function when the size of the training set is increased. In this section we present two ideas on learning such combination functions. Since we do not know the exact analytical form of optimal combination function, the presented combination methods are rather heuristic.

5.1 Learning Best Impostor Distribution

The likelihood ratio combination function of section 4.1 separates the set of genuine score pairs from the set of all impostor score pairs. But we might think that for identification systems it is more important to separate genuine score pairs from the best impostor score pairs obtained in each identification trial. There is a problem, though, that we do not know which score pair is the best impostor in each identification trial. The best impostor score pair can be defined as one having biggest combined score, but the combination function is unknown.

To deal with this problem we implemented an iterative algorithm, where the combination function is first randomly initialized and then updated depending on found best impostor score pairs. The combination rule is based on the likelihood ratio function with the impostor density trained only on the set of found best impostor score pairs. The exact algorithm is presented below:

1. Make initialization of $f(s^1, s^2) = \frac{\hat{p}_{gen}(s^1,s^2)}{\hat{p}_{imp}(s^1,s^2)}$ by selecting random impostor score pairs from each training identification trial for training $\hat{p}_{imp}(s^1, s^2)$.
2. For each training identification trial find the impostor score pair with biggest value of combined score according to currently trained $f(s^1, s^2)$.
3. Update $f(s^1, s^2)$ by replacing impostor score pair of this training identification trail with found best impostor score pair.
4. Repeat steps 2-3 for all training identification trials.
5. Repeat steps 2-4 for predetermined number of training epochs.

The algorithm converges fast - after 2-3 training epochs, and found best impostor score pairs change little in the subsequent iterations. The trained combination function subsequently gets tested using a separate testing set. Table 4 (Best Impostor Likelihood Ratio method) provides the results of the experiments.

Matchers	Likelihood Ratio Rule	Weighted Sum Rule	Best Impostor Likelihood Ratio	Logistic Sum Rule	Weighted Sum + Ident Model
CMR&WMR	4293	5015	4922	5005.5	5025.5
li&C	5817	5816	5803	5823	5826
li&G	5737	5711	5742	5753	5760

Table 4. Correct identification rate for all considered combination methods.

The method seems to perform well, but weighted sum combination rule is still better for word recognizers and biometric li&C matchers. This method is not able to fully account for the dependence of scores in identification trials, and the learning of the optimal combination function will not be probably achieved with it.

5.2 Sum of Logistic Functions

Generally, the matching score reflects the confidence of the match, and we can assume that if the score is bigger, then the confidence of the match is higher. When the scores are combined, the higher score should result in higher combination score. Thus, the combination function $f(s^1, s^2)$ should be monotonically nondecreasing in both of its arguments. One type of monotonic functions, which are frequently used in many areas, are logistic functions:

$$l(s^1, s^2) = \frac{1}{1 + e^{-(\alpha_1 s^1 + \alpha_2 s^2 + \alpha_3)}}$$

If $\alpha_1 \geq 0$ and $\alpha_2 \geq 0$, then $l(s^1, s^2)$ is monotonically nondecreasing in both of its arguments. Our goal is to approximate the optimal combination function as a sum of such logistic functions. The sum of monotonically nondecreasing functions will also be monotonically nondecreasing.

Suppose we have one identification trial and $\mathbf{s_1} = (s_1^1, s_1^2)$ and $\mathbf{s_2} = (s_2^1, s_2^2)$ are two score pairs of this trial. Let $\mathbf{s_1}$ be a genuine score pair, and $\mathbf{s_2}$ be an impostor score pair. Suppose also that we have some initial sum of logistic functions as our combination function. If both matchers gave a higher score to the genuine class and $s_1^1 > s_2^1$ and $s_1^2 > s_2^2$, then by our construction the combination score for genuine class will be higher than the combination score for impostor class. There is no need to do any modifications to our current combination function. If both matchers gave a lower score to the genuine class and $s_1^1 < s_2^1$ and $s_1^2 < s_2^2$, then we can not do anything - any monotonically nondecreasing function will give a lower combination score to the genuine class.

If one matcher gave a higher score to the genuine class and another matcher gave a higher score to the impostor class, we can adjust our combination function by adding corresponding logistic function to the current sum. For example, if $s_1^1 > s_2^1$ and $s_1^2 < s_2^2$ logistic function $l(s^1, s^2) = \frac{1}{1+e^{-(\alpha_1 s^1 + \alpha_3)}}$ will be increasing with respect to the first argument and constant with respect to the second argument. The input sample will be assigned genuine class since first matcher correctly identified it. We choose parameters α_1 and α_3 relative to the training sample:

$$l(s^1, s^2) = \frac{1}{1 + e^{-\frac{1}{h}\frac{1}{a-b}(s^1 - \frac{a+b}{2})}} \tag{8}$$

where $a = s_1^1$ and $b = s_2^1$, and h is the smoothing parameter. If a and b are close to each other, we get a steeper logistic function, which will allow us better separate genuine and impostor score pair. Similar logistic function is added to the current sum if second matcher is correct, and first is not: we replace s^1 by s^2 in equation (8), and $a = s_1^2$,$b = s_2^2$.

The overall training algorithm is similar to the training we did for best impostor likelihood ratio in the previous section:

1. Make initialization $f(s^1, s^2) = s^1 + s^2$, $n = 1$.
2. For each training identification trial and for each impostor score pair in this trial check if its combined score is higher than combined score of the genuine pair.
3. Update $f(s^1, s^2)$ by adding described above logistic function: $f(s^1, s^2) = \frac{1}{n+1}(nf(s^1, s^2) + l(s^1, s^2))$, $n = n + 1$.
4. Repeat steps 2-3 for all training identification trials.
5. Repeat steps 2-4 for predetermined number of training epochs.

The smoothing parameter h is chosen so that the performance of the algorithm is maximized on the training set. The convergence of this algorithm is even faster than the convergence of the best impostor likelihood ratio algorithm. Table 4 (Logistic Sum method) presents correct identification rate for this method.

The method outperforms weighted sum method for both biometric combinations, but not for the combination of word recognizers. This suggests that

our heuristic was quite good, but still can be improved somehow. We can also see that the advantage of this method for second biometric combination outweighs its disadvantage for the combination of word recognizers, and thus we can consider it as the best combination rule so far.

6 Utilizing Identification Model

The previous two section investigated the usage of the so called combination rules in identification systems. We defined the combination rules by equation (2) and mentioned that such combination rules are a specific type of a classifiers operating in MN-dimensional score space and separating N classes, M is the number of classifiers. By considering the combinations of this restricted type we are able to significantly reduce the difficulty of training combination function, but at the same we might not get the best possible performance from our system.

We discussed this topic in length in [18] (see also the chapter on the review of combination methods). It turns out that besides two already mentioned types of combinations (combination rules of equation (2), *low complexity* combinations, and all possible N-class pattern classification methods in MN-dimensional score space, *high complexity* combinations) we can distinguish two additional types of classifier combinations in between. *Medium I complexity* combinations make the combination function class-specific:

$$C = \arg \max_{i=1,\ldots,N} f_i(s_i^1, \ldots, s_i^M) \tag{9}$$

while *medium II complexity* combinations remain class-generic and derive the combination score for each class not only from M scores assigned to this class but from potentially all available MN scores:

$$C = \arg \max_{i=1,\ldots,N} f(s_i^1, \ldots, s_i^M; \{s_k^j\}_{j=1,\ldots,M;k=1,\ldots,N;k\neq i}) \tag{10}$$

Generally, it is possible to use both medium I and medium II complexity type combinations for our applications, but we will concentrate on medium II complexity type. Since the combination functions of this type consider scores for all classes in order to derive a combined score for a particular type, we have a fair chance to properly learn the dependency between scores assigned to different classes, and train the combination function with this dependency in mind.

6.1 Identification Models

The goal of constructing an identification model is to somehow model the distributions of scores in identification trials. Better model will provide more information to the combination algorithm and result in better performance.

We can use different heuristics in order to decide on which identification model might work best in a given application. For example, we might want the identification model to provide a good estimate for posterior class probability for a score from a current set of identification scores.

Consider our third example from the section 4.3. Recall, that genuine and impostor distributions are represented as sums of two random variables: $X_{gen} + Y$ and $X_{imp} + Y$. If each identification trial has many impostor samples, we can estimate the current value of Y as sum of all scores in this trial: $\hat{\mathbf{y}} = \sum_{i=1,\ldots,N} \mathbf{s_i}$ (note, that the mean of X_{imp} is 0). The identification model in this case could state that instead of scores $\mathbf{s_i}$, we have to take their transformations: $\mathbf{s'_i} = \mathbf{s_i} - \hat{\mathbf{y}}$. If the combination rule is trained to use $\mathbf{s'_i}$ instead of $\mathbf{s_i}$, we will achieve near-optimal combination.

The identification model produced for this example is non-trainable, and it is only justified by the assumption that genuine and impostor scores are the sums of two random variables. If the assumption is not true, then the identification model might not perform well. In our research we are interested in designing general identification models which can be learned from the training data and which perform well for any applications.

There might be two approaches on using identification models as represented in Figures 7 and 8. In the first approach the identification model is applied to each score before the actual combination. Thus the score is normalized using identification model and the other identification trial scores. In the second approach identification model provides some statistics about current identification trial, and these statistics are used together with the scores in a single combination step. For our example, we can normalize scores $\mathbf{s'_i} = \mathbf{s_i} - \hat{\mathbf{y}}$ and use normalized score $\mathbf{s'_i}$ in subsequent combination. This will be a two step combination approach. Alternatively, we can use both $\mathbf{s_i}$ and $\hat{\mathbf{y}}$ as an input to the 1-step combination algorithm.

6.2 Related Research

We can list two general approaches in classifier combination research, which implicitly use the concept of identification model. These are the combination approaches based on rank information and combinations utilizing score normalization with current identification trial scores.

Rank based approaches replace the matching scores output by classifiers by their rank among all scores obtained in the current identification trial. Such transformation is performed for each classifier separately, and the ranks are combined afterward. 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 [9]. If there is a dependence between identification trial scores as for second matcher in our first example of section 4.3 (where the top score always belongs to the genuine class), then the rank of the class will be a perfect indicator if the class is genuine or not. Combining low score for genuine class with other scores as

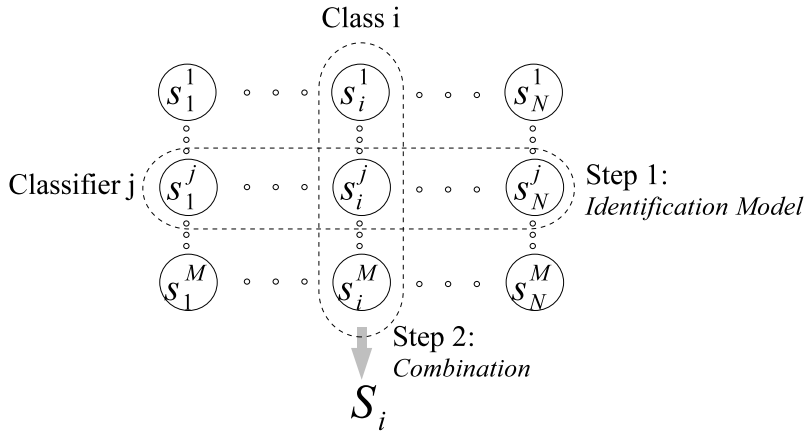


Fig. 7. 2-step combination method utilizing identification model.

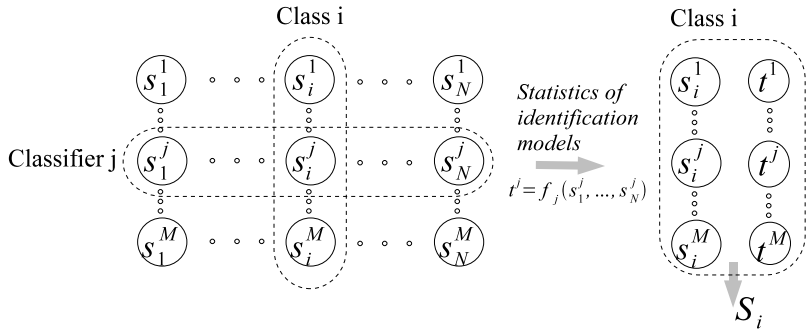


Fig. 8. 1-step combination method utilizing identification model.

in the second example could confuse a combination algorithm, but the rank of the genuine class is still good, and using this rank should result in true classification. Brunelli and Falavigna [4] considered a hybrid approach where traditional combination of matching scores is fused with rank information in order to achieve identification decision. Saranli and Demirekler [15] provide additional references for rank based combination and a theoretical approach to such combinations.

Another approach for combinations, which might use the identification model, is a score normalization followed by some combination rule. Usually score normalization [10] means transformation of scores based on the classifier's score model learned during training, and each score is transformed individually using such a model. Such normalizations do not use the information about scores in identification trial, and the combinations using them can still be represented as a combination rule of equation (2). But some score nor-

malization techniques indeed use a dynamic set of identification trial scores. For example, Kittler et al. [12] normalize each score by the sum of all other scores before combination. The combinations employing such normalizations are medium II complexity type combinations and can be considered as implicitly using an identification model.

Score normalization techniques have been well developed in the speaker identification problem. Cohort normalizing method [14, 5] 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. [2] separated cohort normalization methods into cohorts found during training (constrained) and cohorts dynamically formed during testing (unconstrained cohorts). Normalization by constrained cohorts followed by low complexity combination amounts to medium I combination types, since whole combination method becomes class-specific, but only one matching score of each classifier is utilized. On the other hand, normalization by unconstrained cohorts followed by low complexity combination amounts to medium II or high complexity combinations, since now potentially all scores of classifiers are used, and combination function can be class-specific or non-specific.

The related normalization techniques are Z(zero)- and T(test)- normalizations [2, 13]. Z- normalization is similar to constrained cohort normalization, since it uses impostor matching scores to produce a class specific normalization. Thus Z-normalization used together with low complexity combination rule results in medium I combination. T-normalization uses a set scores produced during single identification trial, and used together with low complexity combination rule results in medium II combination (note that this normalization is not class-specific).

Medium II combinations seem to be the most appropriate type of combinations for identification systems with large number of classes. Indeed, it is usually hard to train class-specific combination types of medium I and high complexity since the number of training samples for each class can be too small. As an example justifying medium II combinations in biometrics, [8] argued for applying T-normalizations in face verification competition. Ranks, T-normalization and many other investigated score normalization approaches are usually non-trainable. The concept of identification model implies that there is some training involved.

6.3 Identification Model for Weighted Sum

We will use the following idea for our identification model in this section. The confidence of a matching score is determined by the score itself and by the other scores in the same identification trial. If for a given score of a classifier there is another score in the same trial which is higher, then we have less confidence that the score belongs to the genuine class. Conversely, if all other

scores are lower than a given score, we have more confidence that the score belongs to the genuine class.

The identification model in this case will consists in considering the following function of the identification trial scores: $sbs(s_i^j)$ - the best score besides score s_i^j in set of the current identification trial scores $\{s_i^j\}_{i=1,\ldots,N}$ of classifier j:

$$sbs(s_i^j) = \max_{k=1,\ldots,N;k\neq i} s_k^j \tag{11}$$

We use the 1-step identification model combination with weighted sum combination function. It means that instead of using only matching scores s_i^j, $j = 1,\ldots,M$ for producing combined score S_i of class i, we will be using both s_i^j and $sbs(s_i^j)$. For two classifiers in our applications we will have the following combination function:

$$S_i = w_1 s_i^1 + w_2 sbs(s_i^1) + w_3 s_i^2 + w_4 sbs(s_i^2) \tag{12}$$

The number of considered input parameters for this method is two times bigger than the number of input parameters to the original weighted sum rule. We can still use the brute force approach to train the corresponding weights. Note, that though the number of weights is increased, the increase is rather small in comparison to the total number of classes (thousands). Thus we achieved the good trade-off between taking into consideration all scores produced by classifiers and the simplicity of training combination function.

The results of the experiments are presented in the Table 4 (Weighted Sum + Ident Model). The method outperforms all other methods for identification tasks. Note, that as in all our experiments, we used separate data sets for training weights and testing the trained method; thus the performance improvement is due not to more possibilities for training, but due to more complex combination function.

6.4 Identification Model for Verification Systems

Although most verification systems use only matching scores for one given class to make combinations and decisions on whether the class is genuine or impostor, there is an idea that the performance can be improved if the matching scores for other classes are taken into consideration. In fact, most of the cohort score normalization methods, which we referenced above, employ a superfluous set of matching scores for a cohort of a given class in order to make verification decision. These scores might be available naturally in identification system, but the verification system has to do additional matches to create these scores.

If the scores for other classes are available in addition to the score for a given class, they can provide significant amount of information to the combination algorithm. Indeed, as we discussed before, the matching scores are usually dependent and the dependence is caused by the quality of the input sample. Scores for other classes can implicitly provide us the information

about the input sample quality. Consequently, we can view the application of identification model as score normalization with respect to the input sample.

The information supplied by the identification model can be considered as a predictor about the given score we consider in the verification task. We imply that this score is genuine, and the goal of the identification model is to check if this score is reasonable in comparison with scores we get for other, impostor, scores. Thus we can check the correlations of the genuine score with different functions of the impostor scores in order to find the statistics, which best predict the genuine score. Table 3 contains the correlation measurements for our matchers, and these measurements can be used to determine which statistics of impostor scores the identification model should include. In our experiments we considered first and second best impostor statistics. They seem to be good predictors according to Table 3, and, as an additional advantage, first few best scores are usually available due to the utilizing indexing methods in identification systems.

The application of the identification model in verification system is clear now. Instead of taking a single match score for a given class from a particular matcher, take few additional match scores for other class, and calculate some statistics from them. Then use these statistics together with a match score for a designated class in the combination. Since the likelihood ratio method is optimal for verification tasks, we use it here. If we employ the statistic of second best score from the previous section, our combination method will be written as

$$f_{lr}(s_i^1,\ldots,s_i^M;\{s_k^j\}_{j=1,\ldots,M;k=1,\ldots,N;k\neq i}) = \frac{p_{gen}(s_i^1,sbs(s_i^1),\ldots,s_i^M,sbs(s_i^M))}{p_{imp}(s_i^1,sbs(s_i^1),\ldots,s_i^M,sbs(s_i^M))} \tag{13}$$

Note that we are dealing with the verification task, so we only produce the combined score for thresholding, and do not select among classes with arg max. Also, during our experiments we used a little different statistics than the statistics $sbs = \max_{k=1,\ldots,N;k\neq i} s_k^j$ from the previous section - we selected the second ranked score from $\{s_k^j, k = 1,\ldots,N; k \neq i\}$.

Figure 9 contains the resulting ROC curve from utilizing identification model by equation 13 in the combination of word recognizers. Note, that this method performs significantly better than the original likelihood ratio method. We have also reported similar improvements for the biometric matchers before[19].

If we look at the verification task as the two class pattern classification problem in the M-dimensional score-feature space, then using identification model corresponds to expanding the feature space by the statistics of identification trials. The achieved improvements confirm the usefulness of these additional features.

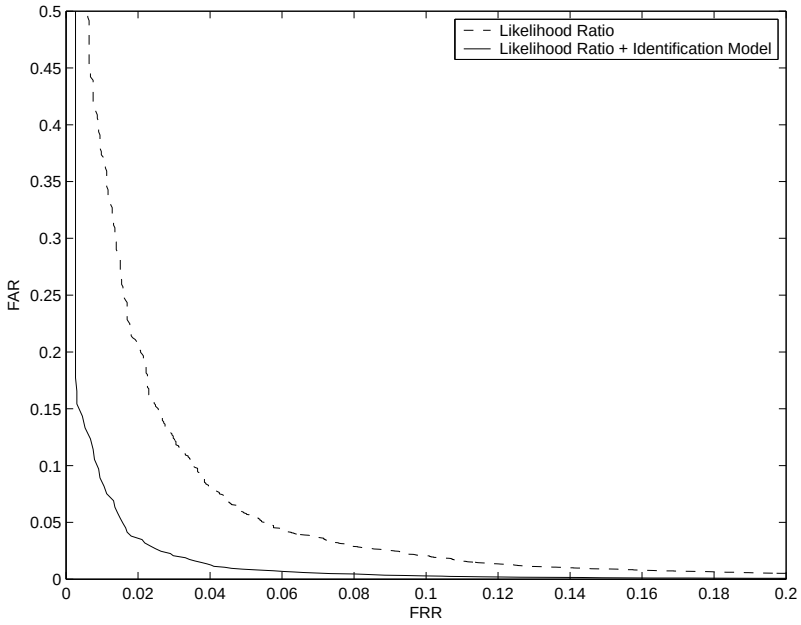


Fig. 9. The effect of utilizing identification model in the likelihood ratio combination function for handwritten word recognizers.

7 Summary

In this work we considered combinations of handwritten word recognizers and biometric matchers. There can be different operating scenarios for the applications involving these matchers, and we considered two of them - verification and closed set identification. Different operating scenarios require different performance measures: ROC curves for verification problems, and correct identification rate for identification problems.

It turns out that for different scenarios we need to construct different combination algorithms in order to achieve optimal performance. This need is caused by the frequent dependence among scores produced by each matcher during a single identification trial. The optimal combination algorithm for verification systems corresponds to the likelihood ratio combination function. It can be implemented by the direct reconstruction of this function with genuine and impostor score density approximations. Alternatively, many generic pattern classification algorithms can be used to separate genuine and impostor scores in the M-dimensional score space, M is the number of combined matchers.

The optimal combination algorithm for the identification systems is more difficult to realize. We do not know how to express analytically the optimal combination function, and can only speculate on the heuristics leading to its

construction. We described two possible approaches for approximating the optimal combination function in identification systems and compared them with traditionally used weighted sum combination method. The results are promising, but it is clear, that further development is needed in this area.

The concept of identification model provides a different point of view on the combinations in identification systems. The score dependence in identification trials can be explicitly learned in these models. The combination algorithm utilizing identification model uses more information about identification trial scores than traditional combination methods relying on a single match score for designated class. As a result it is possible to achieve significant improvements using these models.

References

1. Nist biometric scores set. http://www.nist.gov/biometricscores/.
2. Roland Auckenthaler, Michael Carey, and Harvey Lloyd-Thomas. Score normalization for text-independent speaker verification systems. *Digital Signal Processing*, 10(1-3):42–54, 2000.
3. Ruud M. Bolle, Jonathan H. Connell, Sharath Pankanti, Nalini K. Ratha, and Andrew W. Senior. *Guide To Biometrics*. Springer, New York, 2004.
4. R. Brunelli and D. Falavigna. Person identification using multiple cues. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(10):955–966, 1995.
5. J.M. Colombi, J.S. Reider, and J.P. Campbell. Allowing good impostors to test. In *Signals, Systems & Computers, 1997. Conference Record of the Thirty-First Asilomar Conference on*, volume 1, pages 296–300 vol.1, 1997.
6. J.T. Favata. Character model word recognition. In *Fifth International Workshop on Frontiers in Handwriting Recognition*, pages 437–440, Essex, England, 1996.
7. G.Kim and V.Govindaraju. Bank check recognition using cross validation between legal and courtesy amounts. *Int'l J. Pattern Recognition and Artificial Intelligence*, 11(4):657–674, 1997.
8. Patrick Grother. Face recognition vendor test 2002 supplemental report, nistir 7083. Technical report, NIST, 2004.
9. T. K. Ho, J. J. Hull, and S. N. Srihari. Decision combination in multiple classifier systems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(1):66–75, 1994.
10. Anil Jain, Karthik Nandakumar, and Arun Ross. Score normalization in multimodal biometric systems. *Pattern Recognition*, 38(12):2270–2285, 2005.
11. G. Kim and V. Govindaraju. A lexicon driven approach to handwritten word recognition for real-time applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(4):366–379, 1997.
12. J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas. On combining classifiers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, pages 226–239, March 1998.
13. J. Mariethoz and S. Bengio. A unified framework for score normalization techniques applied to text independent speaker verification. *IEEE Signal Processing Letters*, 12, 2005.

14. A.E. 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*, volume 1, pages 81–84 vol. 1, 1996.
15. Afsar Saranli and Mubeccel Demirekler. A statistical unified framework for rank-based multiple classifier decision combination. *Pattern Recognition*, 34(4):865–884, 2001.
16. B W Silverman. *Density estimation for statistics and data analysis*. Chapman and Hall, London, 1986.
17. S. Theodoridis and Koutroumbas K. *Pattern Recognition*. Academic Press, 1999.
18. 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.
19. S. Tulyakov and V. Govindaraju. Identification model for classifier combinations. In *Biometrics Consortium Conference*, Baltimore, MD, 2006.
20. S. Tulyakov and Govindaraju V. Using independence assumption to improve multimodal biometric fusion. In *6th International Workshop on Multiple Classifiers Systems (MCS2005)*, Monterey, USA, 2005. Springer.

Index