Combining Matching Scores in Identification Model

Sergey Tulyakov and Venu Govindaraju
Center of Excellence for Document Analysis and Recognition (CEDAR)
Department of Computer Science & Engineering
State University of New York at Buffalo, USA
tulyakov@cedar.buffalo.edu

Abstract

The paper discusses a problem of combining recognition scores for different classes produced by one recognizer during one recognition attempt. This problem arises in identification problems which we define as 1:N classification problems with big or variable N. By using artificial example we show that intuitive solution of making identification decision based solely on the best matching score is frequently suboptimal. Paper presents reasons for such behavior, and draws parallels with score normalization technique used in speaker identification. Two examples of real life applications illustrate the possible benefits of properly combining recognition scores.

1. Introduction

The identification problem of pattern recognition can be loosely defined as a classification problem with variable classes. One of the applications is identifying the person using speech, handwriting or any other biometric samples among n enrolled persons. The number of enrolled persons and their identities change arbitrarily, thus class relationships are frequently discarded. Other applications which we will consider in this paper are handwriting recognition with variable lexicon, and barcode recognition.

The class relationships are usually discarded due to problem intractability. For example, the number of persons or handwritten words can be big, and the number of samples available for training can be 1 - for biometric templates, or none - for word recognition. Thus it is reasonable that recognition algorithms do not include relationships between classes into their training procedure. Correspondingly, during recognition stage classes are matched individually - in contrast to usual pattern classification methods. The class whose matching produced best score is declared as winner.

The problem appears when we want to make sure that the winner class is indeed the truth for the input sample. The quick solution appears to be setting up some threshold θ and accepting winner only if the matching score is better than threshold. But it easy to notice that if there is a second candidate with score close to the best score there is a bigger chance that true class might be the one with second score instead of first. Thus instead of condition $s_1 > \theta$, where s_1 is the best score and $>$ means better score, we might want to use condition $s_1 > \theta_1$ and $s_1 - s_2 > \theta_2$, s_2 is the second best score. The second condition takes into consideration not only individual matching scores, but also relations between scores of different classes. Thus this condition partly restores class relationship information into classification task.

We can identify multiple questions arising with regards to described problem:

1. How thresholds θ_1 and θ_2 can be found?
2. Is there better way to combine s_1 and $s_1 - s_2$ to get classification confidence?
3. Can we use $s_3, s_4, \ldots$ to improve confidence estimates?
4. How problem parameters, e.g. number of classes, influence performance gain of using interclass score relationships?

Expanding identification decision to include second best score is very intuitive, and undoubtedly many researchers and system integrators attempted to do it. For example, Brunelli and Falavigna[1] used the ratio of normalized best and second best scores in the decision on person identification based on speech and facial features. One recent paper[8] uses first score and average of N next scores to make decision on writer identification.

On the other hand, the theoretical properties of using second-best scores got little attention so far. For example, Grother and Phillips[3] use simple threshold condition of first type while building the model of identification performance. Our paper attempts to get insight into proper way of combining matching scores for identification problem.

The related idea of combining matching scores is to somehow use local neighborhood information of the matched pattern. For example, the technique of score normalization in speaker verification [7, 2, 5] uses matching scores of a set of similar speakers (cohort) to improve verification confidence. Also there are works where this technique is successfully applied for speaker identification[4]. In a sense, using background models[7] for speaker verification implies some distribution of non-matching classes, and normalized score rather approximates optimal identification score - a Bayes posterior probability of matching input to particular class. This implicit assumption about non-matching score distribution might explain the difficulty in deriving strict mathematical framework for score normalization in verification problem.

In this paper we utilize artificial example to illustrate the effects of using second-best scores in identification process. Later we give examples of using second-best scores in real life applications.

2. Artificial Example

Let N be the number of classes in our identification problem, $s_1 > s_2 > \cdots > s_N$ are the scores produced during identification attempt (index shows the order of scores and not class number), one of the scores s_i is produced by matching with truth class, $id(input) = id(i)$. Let p_m and p_n denote the densities of of matching and non-matching scores. We fix the densities for our example as

$$p_m(s) = c_m e^{-\frac{(1-s)^2}{2\sigma_m^2}}$$
$$p_n(s) = c_n e^{-\frac{s^2}{2\sigma_n^2}}$$

where $s \in [0,1], \sigma_m = .4, \sigma_n = .2$ and c_m and c_n are normalizing constants. The densities for matching and non-matching scores are shown in figure 1.

The main assumption which we make in this example is that the scores s_i produced during matching input pattern against all classes are independent random variables. One score from the truth class is sampled from p_m and remaining $N-1$ scores are sampled from p_n. This assumption is rather restrictive and generally it is not true. For example, frequently matching score includes some measure of input signal quality. Since the quality of input is the same for all matching attempts, we expect that scores $s_1, s_2, \ldots, s_N$ will be dependent.

Using independence assumptions we are able to calculate the joint density of best and second-best scores under two conditions: best score comes from matching truth class and best score comes from matching non-truth class. These are the formulas for densities in case $N = 2$:

$$\begin{aligned} p(s_1, s_2 | id(input) = id(1)) &= p_m(s_1) * p_n(s_2) \\ p(s_1, s_2 | id(input) \neq id(1)) &= p_n(s_1) * p_m(s_2) \end{aligned} \quad (1)$$

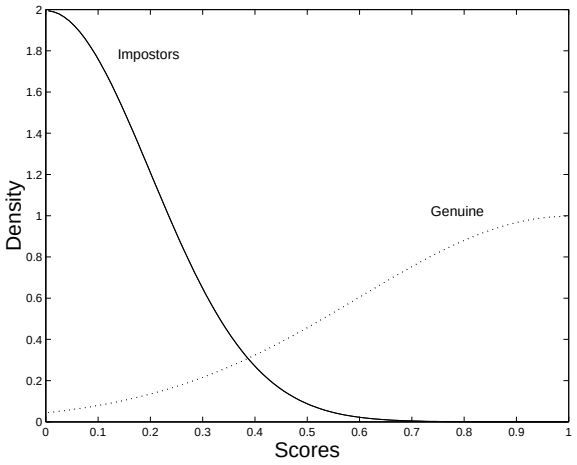

Figure 1. Chosen densities of matching(genuine) and non-matching(impostors) scores.

Bayes decision theory holds that optimal decision surfaces are defined by the likelihood ratio:

$$L = \frac{p(s_1, s_2 | id(input) = id(1))}{p(s_1, s_2 | id(input) \neq id(1))} = \frac{p_m(s_1) * p_n(s_2)}{p_n(s_1) * p_m(s_2)} \quad (2)$$

Sample decision surfaces are shown in figure 2. Note that if

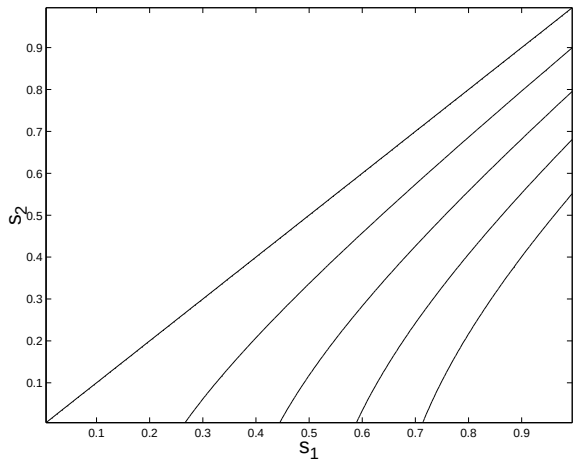

Figure 2. Bayes decision boundaries for N=2 with contours drawn for L=1,10,100,1000 and 10000.

we used only best score s_1 for making decisions, we would

get vertical lines as decision boundaries. Thus decisions involving second best score substantially differ from decisions based solely on s_1. In figure 3 we showed the ROC

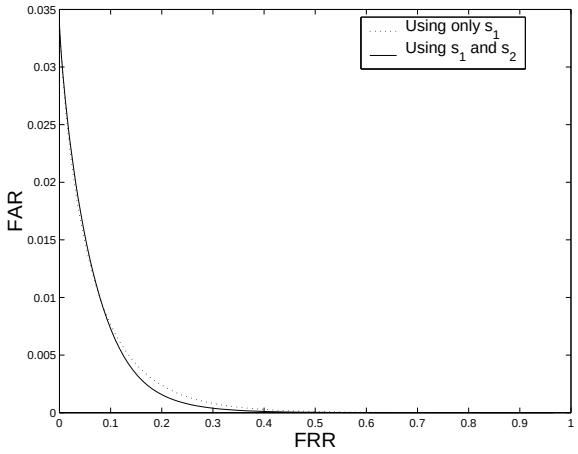

Figure 3. ROC curves for optimal thresholding using and not using second-best score.

curves for decisions utilizing second-best score and not utilizing it. It can be seen that good reduction in false acceptance rate (FAR) is achieved when false reject rate (FRR) is around $0.1 - 0.4$. Second-best score turns out to be good feature for making decision about two cases - best matched class is the truth or best matched class is not a truth.

2.1. Dependent Scores

The main assumption used in our example is the independence of matching scores. But as we noted this is rarely a case in real life matchers. Is it still beneficial to use second-best score for making identification decision if scores are dependent?

Using second-best score in addition to best score amounts to simply adding one more feature in two-class classification problem. In ideal Bayes framework which we consider in artificial example adding more features can not make performance worse. Thus it is possible to only improve results by considering second-best score, as well as all other scores. Of course, in real life we do not have this luxury, and decision on using second-best or other scores will greatly depend on the number of training samples. In our ideal situation we can identify two extreme cases of score dependency with regard to using second-best score.

The worst case is where no improvement can be achieved. As an example of such case consider matcher normalizing scores to posterior probabilities of participating classes, $s_i = P(\omega_i|input)$. Since the task of combining output scores dealt in this paper is to try to approximate such probabilities based on output scores, it is clear that no improvement is possible. More specifically for $N = 2$, if always $s_2 = 1 - s_1$, then considering s_2 for score combination will be useless.

The best case is where combination of s_1 and s_2 achieves perfect separation of successful and unsuccessful identification events. For example, suppose that correct identifications (the best score belongs to genuine class) always have $s_1 - s_2 > .1$, and incorrect identifications have $s_1 - s_2 < .1$. This is quite realistic situation: when the correct class is matched, the distance to other classes will dictate big difference in scores, and if incorrect class is matched, the input sample is most probably lies somewhere in between classes, and distances to classes will be more comparable. In this situation, taking $s_1 - s_2 = .1$ as decision boundary will give us FAR=0 and FRR=0. Trying to use only score s_1 or independence assumption as in formula 1 will fail to give such results.

Summarizing above discussion, if matching scores are independent we expect to achieve average performance improvement from including second-best score into consideration. If scores are dependent, then any situation from no improvement to perfect decision about identification results is possible.

3. Examples of Using Second-Best Matching Scores

In this section we present two examples of incorporating second-best matching score for identification problem.

3.1. Handwritten word recognition

The application which we consider is the recognition of handwritten street names in the automated mail processing system. The handwritten destination address is first automatically segmented and recognition of zip codes and house numbers is performed. Based on these two numbers the database with street names is queried, and results of the query are used as lexicon to the word recognizer [9]. The lexicon could contain as little as one and up to few hundred phrases. The identification of street name by word recognizer serves as a decision for accepting mail piece recognition.

The truth phrase of the recognized image is not always contained in the lexicon. Thus we are dealing not strictly with identification problem, but rather with the mixture of identification and verification. Nevertheless, assuming some probability that this automatically generated lexicon contains truth, we are dealing with identification problem.

The used word recognizer does not do any postprocessing of its results, and has matching distances as output scores. It can be easily verified, since score assigned to a particular class does not change if lexicon is changed (but still includes this particular class). Thus incorporating second-best score fits this application well.

To incorporate second-best score we used logistic regression on (s_1, s_2) samples. Logistic regression is a simple way to model posterior probability

$$P(\text{correct identification}|s_1, s_2) \sim L(s_1, s_2) = \frac{1}{1 + e^{as_1+bs_2+c}} \quad (3)$$

We used total 7615 samples for both training (finding best values of a, b, c) and testing. Among them 2427 samples are incorrectly identified - best score did not correspond to the truth class.

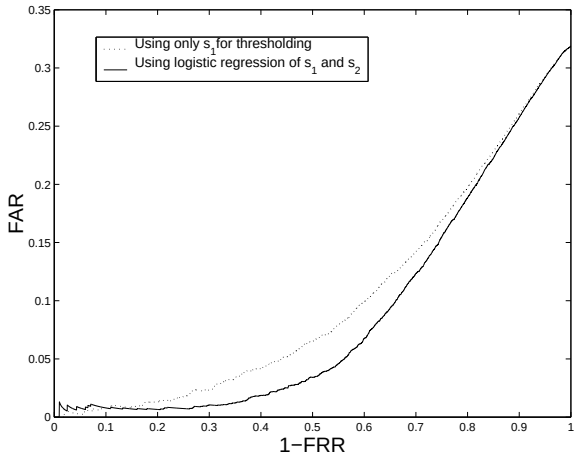

Figure 4. ROC curves for handwritten word recognition.

Figure 4 shows the ROC curves based on thresholding using only first score s_1 and on thresholding the value of logistic function. We can see from the graph that significant reduction in false acceptance rate can be achieved for acceptance rate (1-FRR) from 0.2 to 0.7.

3.2. Barcode Recognition

In this problem we deal with automated recognition of the barcodes on mail pieces. We consider 4-state type barcodes with 65 bars. Each bar can be represented by a pair of bits $(b_1, b_2), b_i \in \{0, 1\}$ indicating the presence ($b_i = 1$) or absence ($b_i = 0$) of upper (b_1) or lower (b_2) part of the bar. Thus whole barcode can be represented by a sequence of 65*2=130 bits $B = (b_1, b_2, \ldots, b_{130})$.

Barcode employs error correction using Reed-Solomon encoding [6] with symbols over Galois field GF(64). It takes 6 bits to encode one symbol, thus barcode can be represented as $\lceil \frac{130}{6} \rceil = 22$ symbols (one symbol is shorted). Out of these 22 symbols we have 4 error-correction symbols. The property of Reed-Solomon encoding is that minimum distance between two codewords is the number of error-correction symbols plus 1. Thus in our case the minimum distance is $4 + 1 = 5$. This means that given one valid barcode we have to change at least 5 symbols (6-bit sequences) to get another valid barcode. Correspondingly, at least 5 bits should be flipped to change one valid barcode into another valid barcode.

The noise model is introduced where it is assumed that any bit can be corrupted and have new float value in the interval $[0, 1]$. Denote corrupted barcode as sequence of 130 float numbers $F = (f_1, f_2, \ldots, f_{130})$. The problem of barcode recognition is given corrupted barcode F find some valid barcode B which is closest to F in some sense. Reed-Solomon decoding algorithm operating on binary strings is able to correct 2 corrupted symbols. Decoding of barcode F with float numbers reflecting the probability of upper or lower part of the bar presence involves making hypothesis about proper binary form of the barcode, and combining it with binary Reed-Solomon decoding. In fact, by means of accepting a multitude of hypothesis on what bars are and which bars are corrupted, we are searching through a set of close valid binary barcodes and find the closest one. We do not present noise model and particular distance function $dist(B, F)$ used, since they are irrelevant for current paper.

The performance of the decoding algorithm is measured by the cost function $Cost$ which is a linear combination of the cost of rejecting recognition results, $RejectRate$, and the cost of accepting incorrectly recognized barcode, $MisreadRate$. Three cost functions were considered: $Cost = RejectRate + k * MisreadRate, k = 2, 10, 100$.

To minimize the cost of barcode recognition we need to find best possible decision algorithm on whether barcode with the best score will be accepted as a recognition result or not. One of the decisions is based on the comparison of this best score s_1 with some preset threshold. Another decision is based on finding two closest valid barcodes and comparing linear combination of corresponding scores $\alpha_1 s_1 + \alpha_2 s_2$ with preset threshold. The thresholds and parameters α_i were found so that $Cost$ is minimized. Table 1 presents the results of the experiments. The numbers in the table show minimum values of $Cost$ (expressed in %) given optimal parameters minimizing that cost.

We can see that incorporating the score of second-best matched barcode allows reducing recognition cost in half. This is very impressive improvement given that recogni-

Cost model	Using s_1	Using s_1 and s_2
k=2	0.3261	0.1998
k=10	0.5287	0.2449
k=100	0.9271	0.5194

Table 1. Costs of barcode recognition are significantly reduced when s_2 is used for thresholding.

tion algorithm was only slightly modified. Barcode recognition is exactly identification problem, since we know all the classes - all valid barcodes (2^{106} total), and we are pretty certain during recognition that corrupted barcode has its truth among all these barcodes. Due to linearity property, each valid barcode has similar neighborhood, and as we discussed in section 2.1 the recognition scores s_1 and s_2 are dependent It is quite possible that dependence of scores turned out to be positive factor in the improvement.

4. Conclusion

The purpose of artificial example and independence condition on matching scores was to reveal that the benefit from using a combination of scores, instead of only one best score, comes naturally. The improvement is explained by explicitly stating that we deal with identification process - the true class is one among N matched classes. In case of dependent scores total improvement of using score combination can be considered as composite of two parts: improvement due to identification process assumption and improvement due to score dependency.

As we noted before, the score normalization technique widely used in speaker verification and identification makes implicit assumptions on matching and non-matching score distributions. Making such assumptions results in implicitly accepting identification model. Consequently, improvements from using score normalizations techniques can be explained directly by utilizing benefits of identification process. Part of the improvements, of course, can be explained by utilizing score dependencies.

The interesting questions which were not answered in this work is what really happens when the number of classes changes and whether it makes sense to use scores $s_3, s_4, \ldots$ in combination. Note that the first practical application has variable number of classes. Clearly, the distribution of the best non-truth score changes with N. Thus it makes sense to try use different decisions for different numbers of N. We performed few experiments with this example by constructing three bins for different sizes of N and thus training three logistic models. Unfortunately, the improvements were insignificant and we had to discard changes. Similar results were achieved by trying to use score s_3 together with scores s_1 and s_2 to construct 3-dimensional logistic regression. Though both ideas are valid and could improve identification, they require significant increase in the number of training samples to produce reliable improvements.

References

[1] R. Brunelli and D. Falavigna. Person identification using multiple cues. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(10):955–966, 1995.

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

[3] P. Grother and P. Phillips. Models of large population recognition performance. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–68–II–75 Vol.2, 2004.

[4] J.-H. Kim, G.-J. Jang, S.-J. Yun, and Y. H. Oh. Candidate selection based on significance testing and its use in normalisation and scoring. In *5th International Conference on Spoken Language Processing (ICSLP-1998)*, 1998.

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

[6] W. Peterson and E. Weldon. *Error-Correcting Codes*. MIT Press, Cambridge, USA, 2nd edition, 1972.

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

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

[9] P. Slavik and V. Govindaraju. An overview of run-length encoding of handwritten word images. Technical Report 2000-09, State University of New York at Buffalo, August 2000.