

# A Methodology for Mapping Scores to Probabilities

Djamel Bouchaffra, *Member, IEEE Computer Society*,  
Venu Govindaraju, *Senior Member, IEEE*, and  
Sargur Srihari, *Fellow, IEEE*

**Abstract**—This paper describes the derivation of probability of correctness from scores assigned by most recognizers. Derivation of probability values puts the output of different recognizers on the same scale; this makes comparison across recognizers trivial.

**Index Terms**—Recognizer, reestimation methods, feature space, classifier, probability.

## 1 INTRODUCTION

WE present in this paper the groundwork for the use of Bayesian methodology in integration of recognizers with any subsequent processing by deriving meaningful probabilistic measures from recognizers. We also address the important notion of scalability of scores [7] and show how scores from different recognizers can be compared. Such normalization of scores under a common scale promotes effective combination of recognizers. Finally, it is our conjecture that the probability values themselves are more precise in what they convey than the typically output scores of recognizers.

Previously, researchers have assumed, in majority of the work [10], that the recognizer merely provides a ranked list of classes for each input pattern and is associated with distance measures which are largely ignored by subsequent stages. Our interest is in deriving probabilistic correctness measures from word recognizers such that their output will be suitable for integration with subsequent stages such as linguistic processing in sentence recognition and classifier combination [7]. Further, we expect that the recognition rate to improve because of the additional retraining required by our methodology.

### 1.1 Word Recognition Background

In this paper, we will draw examples from handwritten word recognition to illustrate our point. However, the methodology described is equally applicable to all pattern classification tasks. The practical implementation of word recognizers use a lexicon of limited size (Fig. 1). Given the image of a handwritten word and a lexicon of possible words, the task is one of ranking the lexicon based on the “goodness” of match between each lexicon entry and the word image. Typically, the word recognizer computes a measure of “similarity” between each lexicon entry and the word image and uses this measure to sort the lexicon in descending order of the similarity measure [4], [9]. The lexicon entry with the highest similarity is the top choice of the recognizer. The top  $m$  choices are often referred to as the confusion set, as it contains the lexicon entries that are “similar” to actual lexicon entry that matches the truth in some feature space.

• The authors are with CEDAR, Department of Computer Science, State University of New York at Buffalo, Amherst, NY 14228.  
E-mail: {bouchaff, govind, srihari}@cedar.buffalo.edu.

Manuscript received 26 Apr. 1999.

Recommended for acceptance by R. Challappa.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 109827.

The similarity measures returned by recognizers are also referred to in the literature as “confidence scores” and “distance measures”:  $C(\omega_i|X)$ , the confidence of the recognizer on class  $\omega_i$  by analyzing pattern  $X$ . Given the same pattern, two recognizers may return the exact same ranking of the lexicon with different associated scores. Further, even the scores of the recognizers can be identical, but the information conveyed is still different. In order to interpret what the significance of a particular score returned by a recognizer is, one needs to study the behavior of the recognizer over several input patterns.

### 1.2 Intuitive Problem Description

Following is the analogy that should clarify the point being made. Teacher A and Teacher B want to evaluate the proficiency of students in Math. Both teachers give an exam to  $n$  students ( $S_1, \dots, S_n$ ) and grade the students’ responses. Teacher A examines the student’s response and gives a score of 80 percent to the student ( $S_j$ ) with the best answers (other students get lower scores). Teacher B gives a score of 90 percent to the same student ( $S_j$ ), which is also the highest score in his grading. Observing the above, the following questions become pertinent.

1. Is student  $S_j$  the most proficient in Math among the students examined?
2. Is the opinion of Teacher B about  $S_j$ ’s proficiency stronger than that of Teacher A, given that Teacher B gave the student a higher score?
3. When Teacher A (B) gives the best student a score of 80 percent (90 percent), is he correct in his selection of the best student?

An intuitive answer to the first question would be that, indeed, student  $S_j$  is the most proficient among the students  $S_1 \dots S_n$  if he scores the highest consistently over many exams. This eliminates the possibility of chance occurrence. However, it must be ensured that all the exams administered are of equal difficulty.

An intuitive answer to the second question is that the information given is insufficient to make the determination. The grading policies of both teachers have to be studied over a large number of tests to quantify their grading behavior. In one teacher’s mind, 80 percent can mean more than what 90 percent means to another teacher. The reason that comparison across teachers is difficult is because each teacher has a different notion of what a particular score means, i.e., the scores are on a different scale.

It is the third question that we address in this paper—the students are the lexicon entries, the teacher is the word recognizer, and the input image is one exam. Our objective is to assess to what degree the recognizer is correct in its ability to label word images. In particular, we are interested in the ability of the recognizer to choose the best class. The degree of correctness over a large number of trials should provide the probability of correctness. It is fair to assume that the probability of correctness increases as scores increase. However, the probability of correctness does not necessarily become 1 when the score of the top choice is 100 percent.

### 1.3 Motivation

No matter what the particular algorithm may be, all word recognizers invariably compute the “goodness” of match between the image and the symbolic representation of the word. While the distance measures returned by recognizers are adequate for most applications where recognition is the final stage of the application, we believe that there is a need for true probabilistic measures. The need for deriving probability values for the purpose of expressing signal and language information in a single framework has recently been underscored by Hull [6].

This research also makes possible the notion of having a common scale. Irrespective of how different recognizers, using

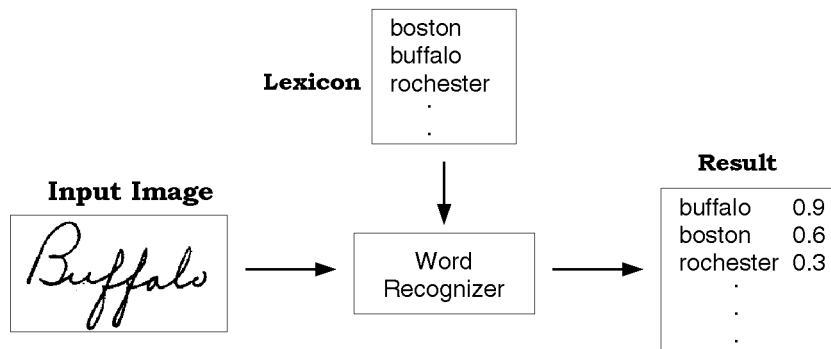


Fig. 1. I/O behavior of a word recognizer. Input is the word image and a lexicon of possible choices. Output is the lexicon sorted by some confidence measure. The top  $m$  choices form the neighborhood or the confusion set.

different paradigms, arrive at their distance scores (or confidence values), when they are converted to probabilities as described above, they are all at a common scale.

Finally, the most important implication of this research from a practical viewpoint is that it describes a methodology of reranking the output of a recognizer based on probability values (derived from the confidence scores), which opens the possibility of improving the recognition rate of the recognizer. In fact, our experiments with digit recognizers (reported in Section 4) show that the improvement is very pronounced in inherently poor recognizers. This is quite remarkable given the fact that we have no access to the features and classification process of the recognizer. The improvements are due to additional training brought to bear upon the problem which reveal insights into the behavior of the recognizer.

#### 1.4 Paper Organization

Section 2 outlines the precise statement of the problem in general mathematical terms and describes the difficulty in finding a solution. We elaborate the methodology developed for deriving probability values from confidence scores returned by recognizers. Section 3 presents the application of sentence recognition to support our claim that probability values from word recognizers allow natural integration with other stages of a recognition engine. Section 4 describes the experiments supporting our methodology for both words as well as digits.

## 2 DPS: DERIVING PROBABILITY GIVEN SCORE

Our task is to take a recognizer given as a blackbox, observe its behavior on an input pattern, and derive probability of correctness of its output. The general problem can be mathematically described as follows:

Let  $\Omega = \{\omega_1, \omega_2, \dots, \omega_c\}$  be the set of  $c$  classes and  $X$  be the input pattern. We define  $P_{\Omega}^X$  as the probability distribution on  $\Omega$  generated by the pattern  $X$ . Traditionally, a recognizer returns  $C(\omega_i|X)$ , where  $C$  is the "confidence" measure. Our objective is to derive the a posteriori probability  $P(\omega_i|X)$ , the probability for a class to be the true class (top choice) given the input pattern  $X$ . Using Bayes' rule:

$$P(\omega_i|X) = \frac{P(X|\omega_i) \times P(\omega_i)}{\sum_{j=1}^{j=m} P(X|\omega_j) \times P(\omega_j)}. \quad (1)$$

Without any prior information, we can reasonably assume that all classes  $\omega_i$  have the same chance of being the truth. In other

words, the "truthing" distribution of classes is uniform and  $P(\omega_i)$  can be approximated by  $\frac{1}{c}$ . However, it is difficult to compute the *state conditional* probability  $P(X|\omega_i)$ . It corresponds to the probability of an input pattern given a particular class as a truth. Infinitely many different patterns can qualify for matching with the same class  $\omega_i$ , hence, the difficulty of estimation.

We circumvent the difficulty in computing  $P(\omega_i|X)$  by falling back to the classic definition of probability. This definition records the frequency of an event of interest over many trials. However, given a particular pattern, a recognizer provides the same ranking of classes, no matter how many times the trials are repeated. In other words, there is no notion of *randomness* inherent in this process. One way of introducing randomness is to generate repeated trials with a large number of patterns that are "similar" to the input pattern. The notion of "similarity" used will be defined further. A set of similar patterns,  $\bar{X}$ , are generated to constitute the trials of the process.  $\bar{X}$  is built during a retraining phase (given that the recognizer in the blackbox was trained before) and contains the input pattern  $X$  and its neighbors. In our analogy, this is the process of constructing more exams for the students where all the exams are of the same difficulty level as the first. It is this "retraining" that brings additional "knowledge" to bear upon the problem, leading to potentially improved recognition rate.

Using the basic definition of probability, we can estimate  $P(\omega_i|X)$  by counting the number of times  $\omega_i$  is the top choice in the ranked lexicon during  $|\bar{X}|$  trials where  $|\bar{X}|$  is the number of input patterns contained in the set  $\bar{X}$ .

$$P(\omega_i|X) \simeq \hat{P}(\omega_i|X) = \frac{\sum_{u \in \bar{X}} \zeta_{\Omega}^u(\omega_i)}{|\bar{X}|}, \quad (2)$$

where

$$\zeta_{\Omega}^u(\omega_i) = \begin{cases} 1 & \text{if } \omega_i \in \Omega \text{ is the top choice given } u \in \bar{X} \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

Since the identity of  $X$  is not known, it is not easy to find the neighbors of  $X$ . The task becomes tractable if we make the assumption that the true class is always present in the confusion set  $\mathcal{A}$  (top  $m$  choices output by the recognizer). Based on this assumption, the neighborhood set  $\bar{X}$  can be written as

$$\bar{X} = \bigcup_{i=1}^{i=m} \bar{X}_j^i, \quad (4)$$

where  $\bar{X}_j^i$  represents the subset of images that belong to class  $\omega_i$ ,  $j$  is the center of the cluster,

TABLE 1

533 Images of Buffalo Divided into Five Clusters Based on the Confidence Scores Received During the Retraining Phase

Cluster <sub>j</sub>	Range of Confidence	Cluster Size	Samples
$\bar{X}_1^{Buffalo}$	0.00 ... 1.99	8	<i>Buffalo</i> <i>Buffalo</i> <i>BUFFALO</i>
$\bar{X}_2^{Buffalo}$	2.00 ... 3.99	326	<i>BUFFALO</i> <i>Buffalo</i> <i>BUFFALO</i>
$\bar{X}_3^{Buffalo}$	4.00 ... 5.99	173	<i>BUFFALO</i> <i>Buffalo</i> <i>Buffalo</i>
$\bar{X}_4^{Buffalo}$	6.00 ... 7.99	22	<i>Buffalo</i> <i>Buffalo</i> <i>Buffalo</i>
$\bar{X}_5^{Buffalo}$	8.00 ... 9.99	4	<i>Buffalo</i> <i>Bryant</i> <i>Buffalo</i>

$\delta$  used is 1.

$$C(\omega_i|X \equiv \text{image of class } \omega_i) \in [j - \delta, j + \delta],$$

and  $\delta$  is a small number described in the following subsection.

**2.1 Retraining**

$C(\omega_i|X \equiv \text{image } \in \mathcal{A} \text{ of class } \omega_i)$  is computed for every image of every class and serves as the image's tag. Thus, all the images belonging to a particular class can be quantized into clusters based on the range of score received. We can choose the  $\delta$  interval so that the images within the cluster of a particular class are all "similar" in the feature space of the recognizer used. One cluster from the images of each class in  $\mathcal{A}$  will form the set of trial images  $\bar{X}$  for retraining.

For the purpose of illustrating our methodology, we have adopted the following clustering procedure. The difference between the maximum and minimum scores received is used to find the range of scores for images of one class. This range was equally divided into equal intervals of size  $2\delta$ . Images with score  $j$  belong to cluster  $X_j^i$ , where  $X$  belongs to the class  $\omega_i$ .

Five hundred and thirty-three samples of images of the word "BUFFALO" were taken. A word recognizer [9] was invoked with the following two inputs: 1) the sample of image of "BUFFALO" and 2) a single lexicon entry: BUFFALO. The idea is to tag each image with the score  $C(\omega_{Buffalo}|X \text{ belongs to } \omega_{Buffalo})$ , the distance in feature space between the input sample and the word

recognizer's notion of a prototype of "Buffalo." Table 1 shows the clusters obtained by choosing  $2 \times \delta = 10$ .

Other clustering techniques [3], [5], [8] can be used as well. Some interesting observations become noteworthy. First, we notice (Table 1) that the word recognizer does reflect our intuitive sense of similarity. Writing styles seem to get sloppy as the cluster distance increases. Second, noise in the image contributes to a lower score even if the writing style is quite good.

If the classes in the confusion set  $\mathcal{A}$  are as seen in Fig. 2 with their respective scores, then  $\bar{X}$  is

$$\bar{X}_{5.32}^{Buffalo} \cup \bar{X}_{4.10}^{Beauty} \cup \bar{X}_{2.80}^{Bounty} \cup \bar{X}_{2.33}^{Niagara} \cup \bar{X}_{1.25}^{North}. \quad (5)$$

We can evaluate probability values as:

$$P(\omega_{Buffalo}|X \text{ unknown image}) \simeq \frac{Nb_{Buffalo}^{|\bar{X}|}}{|\bar{X}|}, \quad (6)$$

where  $Nb_{Buffalo}^{|\bar{X}|}$  is the number of times "Buffalo" is present as the top choice among  $|\bar{X}|$  trials.

**3 SENTENCE RECOGNITION APPLICATION**

Sentence recognition applications deal with: 1) an input sentence image, 2) word recognition results for each word image in the sentence, and 3) a language source expressed by the word/part-of-speech distributions [2], [1].

A sentence image  $I = i_1 i_2 \dots i_n$  is assigned to the sequence of clusters  $\bar{I} = \bar{I}_1 \bar{I}_2 \dots \bar{I}_n$ , where each  $\bar{I}_k$ ,  $\{k \in [1..n]\}$  is a cluster containing the  $k$ th input word image and  $n$  is the number of handwritten word images. This provides a mapping from the space of word images to the space of clusters of word images.

The problem consists of determining an optimal word/part-of-speech path  $(W, T)^*$  given the input sentence image  $I = i_1 i_2 \dots i_n$ . The optimal path  $(W, T)^*$  can be written as:

$$(W, T)^* = \underset{(W, T)}{\operatorname{argmax}} P((W, T)|I). \quad (7)$$

Equation (8) illustrates the use of a Bayesian framework to integrate signal and language can be written as a product of two terms.

<i>Buffalo</i>	
Buffalo	5.32
Beauty	4.10
Bounty	2.80
Niagara	2.33
North	1.25

Fig. 2. ASCII words that form set  $\mathcal{A}$  and their scores, given the word image "Buffalo".

TABLE 2  
Comparison of GSC and DPS

size of the test set	GSC		DPS	
	correctly recognized	% of correctly recognized	correctly recognized	% of correctly recognized
31000	29456	97.40	29472	97.45

TABLE 3  
Performance of GSC and DPS

class	size of the test set	GSC		DPS	
		correctly recognized	% of correctly recognized	correctly recognized	% of correctly recognized
0	6967	6832	98.06	6833	98.08
1	6524	6482	99.36	6481	99.34
2	3935	3756	95.46	3777	95.98
3	2228	2166	97.22	2168	97.31
4	3162	3051	96.50	3050	98.49
5	1539	1486	96.56	1483	96.36
6	1462	1430	97.81	1420	97.13
7	1438	1387	96.45	1387	96.45
8	2042	1944	95.20	1952	95.59
9	944	922	97.57	922	97.57

$$(W, T)^* = \arg \max_{(W, T)} \left[ \prod_{j=1}^{j=n} L((w_j, t_j), (w_{j-1}, t_{j-1}), (w_{j-2}, t_{j-2})) \times P(w_j | i_j) \right]. \quad (8)$$

The language part of the model is given by (9).

$$L((w_j, t_j), (w_{j-1}, t_{j-1}), (w_{j-2}, t_{j-2})) = \frac{P((w_j, t_j) | (w_{j-1}, t_{j-1}), (w_{j-2}, t_{j-2}))}{P(w_j, t_j)}. \quad (9)$$

## 4 EXPERIMENTAL RESULTS

We have conducted experiments with handwritten word recognizers and handwritten digit recognizers. In the case of handwritten word recognizers, we cannot report actual improvement of recognition rates because of the sparseness of data. It is difficult to find many samples of the same word. On the other hand, samples of handwritten digits are abundantly available.

### 4.1 Word Recognition

There were 1,621 images of 27 different words collected for the experiment. Following is the procedure adopted.

1. Clusters are created by submitting each image in the retraining data set (several samples of the same ASCII exist) to the word recognizer with only the exact truth in the lexicon.
2. The samples are put in five different clusters by simply dividing the range of scores into five zones and allowing each sample image to fall into a zone.
3. The word recognizer runs on each image in the test set.
4. For each of the top  $m$  choices returned, all samples of images corresponding to the same class and the score range are put together into a single group,  $\bar{X}$ .
5. Each of the sample images in this group are sent to the word recognizer.
6. Probability values are computed based on the presence of the word in the top choice.

The following tables show the output of the word recognizer on three examples. The truth (identity) is unknown to the testing

program. We display the probability value for each ASCII word and rerank the confusion set with respect to these probability values. The probabilities in the tables do not sum to 1 because all the lexicon entries are not shown.

TRUTH: "EVERYTHING"

TOP1	ASCII	SCORES	PROBABILITIES
1:	insurance	6.022	0.364
2:	everything	6.087	0.273
5:	deductible	7.036	0.273
4:	valuables	6.968	0.091
3:	completely	6.954	0.000

TRUTH: "ESTIMATES"

TOP1	ASCII	SCORES	PROBABILITIES
1:	insurance	6.340	0.357
4:	estimates	6.736	0.214
3:	valuables	6.693	0.071
2:	family	6.533	0.071
5:	expensive	6.967	0.000

TRUTH: "RUINED"

TOP1	ASCII	SCORES	PROBABILITIES
1:	ruined	5.387	0.353
3:	insurance	6.033	0.235
4:	valuables	6.101	0.235
2:	between	5.995	0.059
5:	caused	6.271	0.000

## 4.2 Digit Recognition

We have already described how the DPS changes results only for those samples that fall in the proximity of class boundaries. Hence, if the the original recognizer has a high recognition accuracy, DPS can only marginally affect the results. GSC digit recognizer [4] is such a recognizer with very high accuracy. DPS does improve the overall accuracy, albeit, by just 0.05 percent (Table 2).

The performance improves in most classes but falls in some. Our future work will be to analyze the cause of drop in accuracy in some of the classes (Table 3).

## 5 SUMMARY

We have presented a methodology for deriving probability values given recognition scores assigned to classes. We have argued that these probability values are more informative and useful in a variety of applications including, but not limited to, multiple classifier methodologies and statistical language modeling.

The methodology has many applications. It is very attractive because it enhances the performance of any classifier while treating it as a black box. Further, the only resource required to enable this method is a large data set for retraining. Our methodology will continue to improve the performance of a classifier as long as new data samples are added. Furthermore, each time the original classifier improves, our method can potentially further enhance the classifier.

Although we draw examples from handwritten word recognition and digit recognition to illustrate our point, the methodology described here is equally applicable to all pattern classification tasks.

## 6 FUTURE WORK

The research presented in this paper opens many new avenues for further work.

1. The size of the clusters depends on the size of  $\delta$ . Finding optimal values of  $\delta$  will lead to interesting results and improvements.
2. Statistical language modeling techniques can greatly benefit from the notion of deriving probability values as described. This will allow a Bayesian framework to combine recognition results at the word level to arrive at results at the sentence level.
3. Perhaps the most exciting area of research that will benefit from this work is classifier combination. The literature describes logistic regression, Borda count, and other schemes which all emphasize on the ranks of classes and not their scores. DPS will serve as a powerful tool for classifier combination methodologies.

## REFERENCES

- [1] D. Bouchaffra, V. Govindaraju, and S. Srihari, "A Methodology for Deriving Probabilistic Correctness Measures from Recognizers," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 930-935, Santa Barbara, Calif., 1998.
- [2] D. Bouchaffra, E. Koontz, V. Kripasundar, and R.K. Srihari, "Incorporating Diverse Information Sources in Handwriting Recognition Postprocessing," *Int'l J. Imaging Systems and Technology*, vol. 7, pp. 320-329, 1996.
- [3] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
- [4] J. Favata and S.N. Srihari, "Off-Line Sentence Level Recognition," *Proc. Int'l Workshop Frontier of Handwriting Recognition (IWFHR V)*, 1996.
- [5] J. Hartigan, *Clustering Algorithms*. New York: Wiley, 1975.
- [6] J.J. Hull, "Incorporating Language Syntax in Visual Text Recognition with a Statistical Model," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 12, pp. 1,251-1,256, Dec. 1996.
- [7] T.K. Ho, J.J. Hull, and S.N. Srihari, "Decision Combination in Multiple Classifier Systems," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 1, pp. ?, Jan. 1994.
- [8] A.K. Jain and R.C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, N.J.: Prentice Hall, 1988.
- [9] G. Kim and V. Govindaraju, "A Lexicon Driven Approach to Handwritten Word Recognition for Real-time Applications," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 366-379, Apr. 1997.
- [10] C.Y. Suen et al., "Computer Recognition of Unconstrained Handwritten Numerals," *IEEE Proc.*, vol. 80, pp. 1,162-1,180, 1992.