A String Length Predictor to Control the Level Building of HMMs for Handwritten Numeral Recognition

Alceu de S. Britto Jr[a,b], Robert Sabourin[c,d], Flavio Bortolozzi[a], and Ching Y. Suen[d]

[a]Pontifícia Universidade Católica do Paraná (PUC-PR)
Rua Imaculada Conceição, 1155 - Curitiba (PR) 80215-901 - Brazil
[b]Universidade Estadual de Ponta Grossa (UEPG)
Pr. Santos Andrade S/N, Centro - Ponta Grossa (PR) 84100-000 - Brazil
[c]École de Technologie Supérieure (ETS),
1100 Rue Notre Dame Ouest - Montreal (QC) H3C 1K3 - Canada
[d]Centre for Pattern Recognition and Machine Intelligence (CENPARMI),
1455 de Maisonneuve Blvd. West, Suite GM 606 - Montreal (QC) H3G 1M8 - Canada

Abstract

In this paper a two-stage HMM-based method for recognizing handwritten numeral strings is extended to work with handwritten numeral strings of unknown length. We have proposed a Bayesian-based string length predictor (SLP) to estimate the number of digits in a string taking into account its width in pixels. The top 3 decisions of the SLP module are used to control the maximum number of levels to be searched by the Level Building (LB) algorithm. On 12,802 handwritten numeral strings and 2,069 touching digit pairs, this strategy has shown a small loss (0.91%) in terms of recognition performance compared to the results when the string length is considered as known.

1. Introduction

The LB search algorithm has been successfully used in the field of speech and text recognition to provide a way of avoiding a prior segmentation of words into characters [1,4,6]. We have used this algorithm to match individual numeral Hidden Markov Models (HMMs) against an unsegmented observation sequence with the objective of obtaining the *N* best segmentation-recognition paths for handwritten numeral strings of known length [2]. In this paper, we focus on a strategy to control the maximum number of levels in the LB search. The objective is to overcome the constraint related to the necessity of a priori knowledge of the string length (number of digits) for the recognition process.

In this direction, Procter and Elms have proposed an adaptative level building [6]. They have tried to control the LB search taking into account the probability of each level. To this end, the LB search should be terminated 2 or 3 levels further the point where the probability of the best match at the current level is lower than that of the previous level. However, this approach brought some loss in terms of recognition rate. They observed best results when the number of levels (*L* parameter) was fixed at 22. This value of *L* was appropriate for them (approximately two levels per digit), since it was enough to model each digit and all inter-digit spaces in experiments using strings composed of 2, 3, 4, 5, 6 and 10 digits extracted from the NIST database.

In our approach, the *L* parameter is defined taking into account some contextual information regarding the width of the string (in pixels). The string width is used to estimate the number of digits in the string. With this strategy, we define the *L* parameter of the LB search dynamically, without a significant loss in terms of recognition performance.

2. System overview

A general overview of our method for numeral string recognition is presented in Fig. 1. In the SCB (String Contextual-based) stage, a given numeral string is first preprocessed in order to correct slant, smooth the string contour and calculate the string bounding box. Subsequently, the FF (Foreground Feature) module scans the string image from left to right, while a feature vector based on foreground information is calculated for each column in the string bounding box. This vector is mapped to a discrete symbol available in a previously constructed codebook. The output of the FF module is a sequence of discrete observations representing the entire numeral string. The length of this sequence corresponds to the number of columns in the string bounding box. In the SR (Segmentation-Recognition) module, numeral HMMs trained on isolated digits ($\lambda_c^0, \lambda_c^1, ..., \lambda_c^9$), but considering string contextual information, are matched to the observation sequence provided by the FF module. The objective is to find the *N* best segmentation-recognition paths or hypotheses using an implicit segmentation-based strategy. Basically, this module consists of an LB algorithm whose number of levels to be constructed for each string depends on the String Length Predictor (SLP) outputs. For each segmentation-recognition hypothesis, the SR module provides: string length (number of digits), segmentation points, recognition result of each string segment and the global string recognition probability.

The segmentation-recognition hypotheses generated by the SCB stage are re-ranked in the Verification Stage. Basically, this second stage consists of an HMM-based digit classifier (VF) trained on isolated digits without taking into account any string contextual information. The FBF module (Foreground and Background Features) combines foreground and background information to improve the recognition performance of the numeral HMMs. Moreover, 10 additional numeral HMMs ($\lambda_{vr}^0, \lambda_{vr}^1, ..., \lambda_{vr}^9$) based on the rows of the numeral images are combined with the column-based models ($\lambda_{vc}^0, \lambda_{vc}^1, ..., \lambda_{vc}^9$) to ensure an accurate representation of the digit classes.

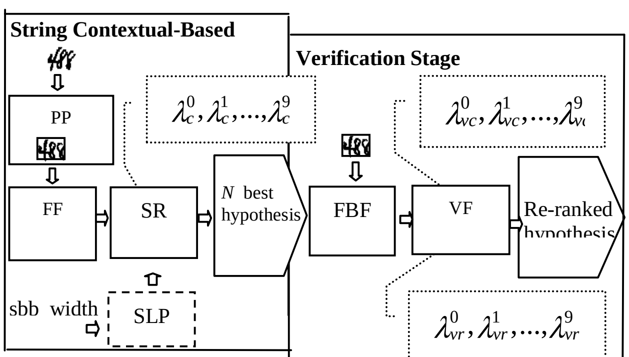

Figure 1. Overview of the proposed method

The verification process starts in the FBF module, in which the segmentation points provided by the first stage are used to define string segments and calculate their bounding boxes. Then, for a given segment, a feature vector combining foreground and background information is extracted from each column of the segment bounding box. This vector is mapped to a discrete symbol in a previously constructed codebook. A similar process is carried out for the rows of the segment. The output of the FBF module consists of two sequences of discrete observations for each segment: column-based and row-based sequences. In the Verification module (VF), the first step is to select the column ($\lambda_{vc}^0, \lambda_{vc}^1, ..., \lambda_{vc}^9$) and row ($\lambda_{vr}^0, \lambda_{vr}^1, ..., \lambda_{vr}^9$) models corresponding to the segment to be verified. This is done by using the recognition result of each string segment provided by the SCB stage. The selected models are scored using the corresponding sequence of observations extracted by the FBF module. Figure 2 shows the zoning scheme provided by the combination of column and row-based models, both trained on foreground and background features (relative position and orientation of each transition, projections and concavity configurations) [3].

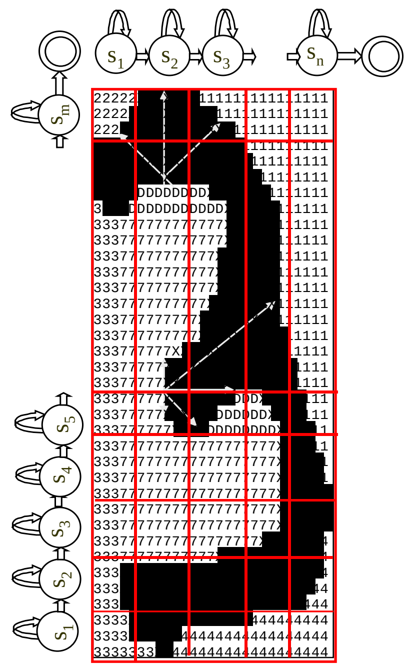

Figure 2: A zoning scheme provided by the combination of column and row-based HMMs

The output probabilities of both column and row models are combined by summing their logs. The resulting probability corresponds to the probability of the string segment to be correct. A string probability is obtained by summing the probability of each segment of the numeral string. This string probability is added to the probability of the corresponding segmentation-recognition hypothesis obtained from the SCB stage. The final resulting probability is used to re-rank the string segmentation-recognition hypothesis.

In the SCB stage of our method, we have used contextual information in different moments. First, for the construction of the individual numeral models with the objective of accurately represent numeral strings. The digits used as training samples are slant-corrected taking into account the slant estimated from their original strings [3]. In a second moment, the isolated numeral HMMs are trained on handwritten numeral strings in order to create a space-model inside of them, which consists of an additional state to represent the inter-digit spaces in numeral strings. The strategy of building the space model into the numeral models instead of using an independent model is an important one in the LBA framework, since possible future problems when the method is generalized for unknown-length strings will be avoided. An independent space model would represent one more model competing at each LBA level, which must be taken into account to estimate the string length (number of digits).

Following the idea of using, in the SCB stage, contextual information as much as possible, we have created the new module SLP in our system. The objective of this module is to predict the number of digits in the string taking into account the string width in pixels.

3. String Length Predictor

The String Length Predictor (SLP) is based on Bayes theory [5]. This module uses the minimum-error-rate decision rule to predict the string length (number of digits) given the width of the string bounding box (*sbb*) in terms of number of pixels. Let us consider, a set of string classes defined as:

$$w = \{2\_digit, 3\_digit, 4\_digit, 5\_digit, 6\_digit, 10\_digit\} \quad (1)$$

in which class $\#\_digit$ corresponds to strings composed of # digits. The a priori probabilities of these classes are considered ambiguous, *i.e.* $P(2\_digit) = P(3\_digit) = P(4\_digit) = P(5\_digit) = P(6\_digit) = P(10\_digit)$. The parameters of a Gaussian *pdf* are estimated for each class by using a training set composed of 44,256 handwritten numeral strings extracted from the NIST SD19 database. Then, a string length classifier is designed to classify the *sbb_width into* M classes of string lengths by using M discriminant functions $g_j(sbb\_width)$, computing the similarities between the unknown data *sbb_width* and each string class w_j and selecting the class w_i corresponding to

$$g_i(sbb\_width) > g_j(sbb\_width) \quad \text{for all } j \neq i . \quad (2)$$

Figure 3 shows the scheme of this classifier, where the decision rule is to maximize the a *posteriori* probability, while Figure 4 presents the probability distributions taking into account the *sbb_width* of each string class.

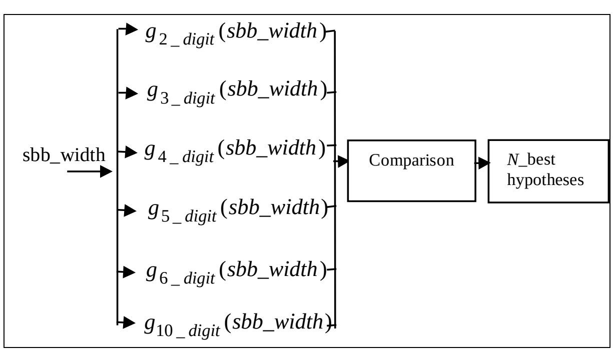

Figure 3. SLP architecture

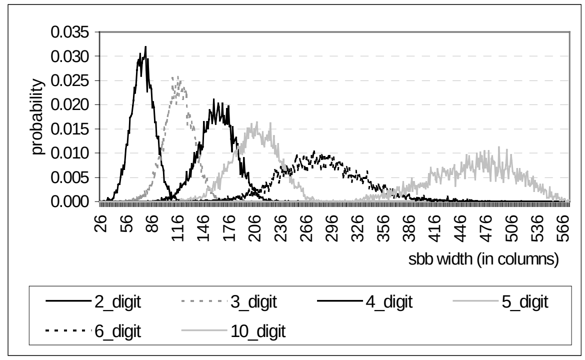

Figure 4. Probability distributions taking into account the *sbb_width* of each string class.

A test set composed of 12,802 numeral strings is used to evaluate the length predictor. Figure 5 shows the classification results. It is possible to observe that when the right decision is considered in the best 3 hypotheses, the performance of this predictor is very promising. Thus, we have used the top 3 decisions to determine the *L* parameter of the LBA instead of using a fixed value for all strings. For instance, if the top 3 decisions belong to the 3_*digit*, 4_*digit* and 5_*digit* classes, then five levels are constructed by the LBA and after that the probabilities of each segmentation-recognition hypothesis corresponding to these string lengths are compared.

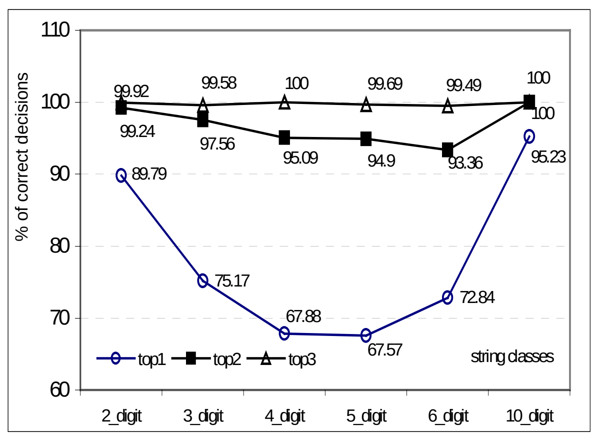

Figure 5. Classification results of the *sbb_width* into string length classes using the proposed predictor

4. Experimental Results

The experiments are based on 12,802 samples extracted from the hsf_7 series of NIST SD19 and distributed into 6 classes of strings: 2_digit (2,370), 3_digit (2,385), 4_digit (2,345), 5_digit (2,316), 6_digit (2,169) and 10_digit (1,217). These strings exhibit different problems, such as touching, overlapping and fragmentation. In addition, to evaluate the system in terms of touching digits we use a subset of data containing 2,069 touching digit pairs (TDPs) also extracted from NIST SD19.

During these experiments the SR module provides the 10 best segmentation-recognition paths for each numeral string. In the Verification stage, the FBF module uses the segmentation points of each path as delimiters in the preprocessed string image to calculate new features based on columns and rows for each string segment. The recognition result of the first stage is verified using the new set of features and numeral HMMs available in the Verification stage. We combine the recognition results of the SCB and Verification stages by summing the logs of their probabilities. Table 1 presents the top 5 recognition results for known-length numeral strings. The last line of this table shows the recognition results for Touching Digit Pairs (TDPs) using a database composed of 2,069 samples extracted from NIST database.

So far, the string recognition has been based on an informed strategy, *i.e.* the string length (number of digits) is known. This strategy is also important, since we can evaluate the system in different conditions, while at the same time adjusting some important aspects regarding string normalization, feature extraction and HMM parameters. However, this represents a strong constraint to the string recognition proposal.

Class	Top (1)	Top (2)	Top (3)	Top (4)	Top (5)
2_digit	95.23	97.59	98.35	98.48	98.57
3_digit	92.62	95.60	96.18	96.27	96.28
4_digit	92.11	95.35	95.95	96.03	96.12
5_digit	90.00	93.96	94.52	94.69	94.73
6_digit	90.09	94.05	94.88	94.92	95.02
10_digit	86.94	90.30	90.38	90.46	90.46
Global	91.57	94.86	95.47	95.57	95.63
TDPs	89.61	94.39	95.36	95.70	95.84

Table 1: Top 5 Recognition results for numeral string recognition of known length

To overcome this constraint, we have used the string length predictor proposed in this paper. Table 2 shows the recognition results.

Class	Top (1)	Top (2)	Top (3)	Top (4)	Top (5)
2_digit	94.81	97.17	97.93	98.05	98.14
3_digit	91.61	94.61	95.05	95.09	95.09
4_digit	91.25	94.29	94.84	94.93	94.97
5_digit	88.30	92.18	92.66	92.79	92.83
6_digit	89.07	92.81	93.55	93.59	93.68
10_digit	86.94	90.30	90.38	90.46	90.46
Global	90.66	93.87	94.41	94.50	94.54
TDPs	88.98	93.57	94.88	95.36	95.70

Table 2: Top 5 Recognition results for numeral string recognition of unknown length

As we can see, the scheme proposed to predict the string length provided a way of relax the constraint of the string recognition method, related to the a priori knowledge of the string length provided by the SLP module, with 0.91% of loss in the global recognition rate.

5. Conclusion

We proposed a Bayesian-based string length predictor (SLP) to estimate the number of digits in a string taking into account its width in pixels. The top 3 decisions of this predictor were used to control the maximum number of levels to be searched by the Level Building (LB) algorithm. With this strategy, it is possible to define the L parameter of the LB search dynamically, without a significant loss in terms of recognition performance. The final recognition rates of strings composed of 2, 3, 4, 5, 6, and 10 digits were: 94.81%, 91.61%, 91.25%, 88.30%, 89.07%, and 86.94% respectively. This means less than 1% of loss in the global recognition rate.

Acknowledgements: The authors wish to thank Fundação Araucária (Paraná-Brazil) and NSERC of Canada (grant 106456-00), which have supported this work.

References

[1] Bose C.B. and Kuo S-S. Connected and Degraded Text Recognition using Hidden Markov Model. Pattern Recognition, Vol. 27, No. 10, pp. 1345-1363, 1994.

[2] Britto A.S., Sabourin R., Bortolozzi F. and Suen C.Y., An Enhanced HMM Topology in an LBA Framework for the Recognition of Handwritten Numeral Strings, Proceedings of the International Conference on Advances in Pattern Recognition (ICAPR'2001), Vol 1, pp. 105-114, Rio de Janeiro-Brazil, March 2001.

[3] Britto A.S., Sabourin R., Bortolozzi F. and Suen C.Y., A two-stage HMM-based systems for recognizing handwritten numeral strings. Proceedings of the International Conference on Document Analysis and Recognition (ICDAR'01), pp. 396-400, Seattle, USA, September 2001.

[4] Elms A.J., Procter S. and Illingworth J. The Advantage of using an HMM-based Approach for Faxed Word Recognition. International Journal on Document Analysis and Recognition (IJDAR). pp. 18-36, 1998.

[5] Huang X.D., Ariki Y. and Jack M.A. Hidden Markov Models For Speech Recognition. Edinburgh Information Technology Series, Edinburgh University Press, Edinburgh, 276 p, 1993.

[6] Procter S. and Elms A. J. The recognition of handwritten digit strings of unknown length using hidden markov models. Proc. of the Fourteenth International Conference on Pattern Recognition (ICPR'98), pp. 1515-1517, 1998.