Evaluation of different strategies to optimize an HMM-based character recognition system

Murilo Santos[1], Albert Ko[2], Luis S. Oliveira[3], Robert Sabourin[2], Alessandro L. Koerich[1] and Alceu S. Britto Jr[1,4]

[1]Pontifícia Universidade Católica do Paraná (PUCPR), Curitiba (PR), Brasil
[2]École de Technologie Supérieure (ETS), Montreal (QC), Canada
[3]Universidade Federal do Paraná (UFPR), Curitiba (PR), Brasil
[4]Universidade Estadual de Ponta Grossa (UEPG), Ponta Grossa (PR), Brasil

Abstract

Different strategies for combination of complementary features in an HMM-based method for handwritten character recognition are evaluated. In addition, a noise reduction method is proposed to deal with the negative impact of low probability symbols in the training database. New sequences of observations are generated based on the original ones, but considering a noise reduction process. The experimental results based on 52 classes of alphabetic characters and more than 23,000 samples have shown that the strategies proposed to optimize the HMM-based recognition method are very promising.

1. Introduction

Many efforts have been made to provide solutions for the recognition of handwriting texts. Some maturity can be observed for isolated digit recognition. Recognition rates above 99.0% for isolated digits [1-3] have been reached by using different methods and strategies. However, when the focus is the recognition of alphabetic characters the results are not in the same level, since such a problem is more complicated. The most obvious difference is the number of classes that can be up to 52 depending if uppercase (A–Z) and lowercase (a–z) characters are distinguished from each other. Consequently, there are a larger number of ambiguous alphabetic characters other than numerals. Character recognition is further complicated by other differences such as multiple patterns to represent a single character, cursive representation of letters, and the number of disconnected and multi–stroke characters [4]. In fact, we can say that character recognition is still an open problem.

Among the important contributions available in the literature, we can find the works proposed by Oh and Suen [5], Dong et al. [6], Koerich et al. [7] and Britto et al. [8]. The first three referred works are based on Neural Networks and the last is based on Hidden Markov Models (HMMs). The use of HMMs has shown to be a promising strategy since this kind of stochastic models may be dynamically combined for the recognition of numeral strings or even words [8,9]. In addition, it is possible to add contextual information related to the interaction of adjacent characters during the training process. However, the modeling process using HMMs is quite complex and different aspects must be taken into account, such as: the model topology, number of states, number of symbols per state, the strategy used to represent the features inside the models, and so on. The method proposed in [8] is a typical sample of this kind of recognition approach, and it has been successfully used for the recognition of isolated digits and characters, numeral strings and also cursive words. However, there are still some points to be improved in this method.

Thus, the objective of this work is to investigate different approaches to combine the complementary features used in the HMM-based method proposed in [8], and also to evaluate a strategy to reduce the negative impact of noise usually present in the discrete observation sequences used for training the character HMMs.

This work is organized into 5 sections. Section 2 presents a general overview of the original recognition method, which is improved in this work. Section 3 shows the optimization strategies: the different schemes to combine the complementary features in the character HMMs, and the strategy used to reduce the impact of noise in the recognition process by manipulating the discrete observation sequences used in the training step. In the Section 4, we present the experimental results, while Section 5 shows our conclusions and future work.

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.230

2. System overview

This section presents the original system which is improved in this work in some specific aspects described in Section 3.

2.1 Feature extraction method

The extraction method consists of scanning the character image from left-to-right (column-based features) and from bottom-to-top (row-based features). Foreground and background information are combined in a vector of 47 features: 34 foreground plus 13 background features.

Foreground features (FF): The FF vector consists of local and global features calculated taking into account the foreground pixels of the image columns or rows. The local features are based on transitions from background to foreground pixels and vice versa.

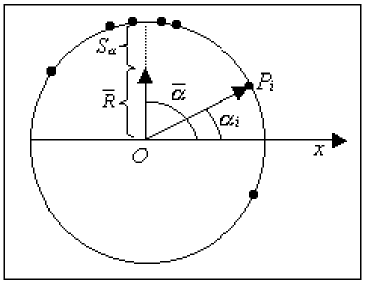

Figure 1. Circular mean direction $\overline{\alpha}$ and variance S_α for a distribution $F(\alpha_i)$

For each transition, the mean direction and corresponding variance are obtained by means of statistic estimators. These estimators are more suitable for directional observations, since they are based on a circular scale. For instance, given the directional observations $\alpha_1 = 1^o$ and $\alpha_2 = 359^o$, they provide a mean direction ($\overline{\alpha}$) of 0^o instead of 180^o calculated by conventional estimators. Let $\alpha_1, \ldots, \alpha_i, \ldots, \alpha_N$ be a set of directional observations with distribution $F(\alpha_i)$ and size *N*. Figure 1 shows that α_i represents the angle between the unit vector $\overline{OP_i}$ and the horizontal axis, while P_i is the intersection point between $\overline{OP_i}$ and the unit circle. The Cartesian coordinates of P_i are defined as:

$$(\cos(\alpha_i), \sin(\alpha_i)) \tag{1}$$

The circular mean direction $\overline{\alpha}$ of the *N* directional observations on the unit circle corresponds to the direction of the resulting vector $(\overline{R})$ obtained by the sum of the unit vectors $(\overline{OP_1}, \ldots, \overline{OP_i}, \ldots, \overline{OP_N})$. The center of gravity $(\overline{C}, \overline{S})$ of the *N* coordinates $(cos(\alpha_i), sin(\alpha_i))$ is defined as:

$$\overline{C} = \frac{1}{N} \sum_{i=1}^{N} cos(\alpha_i) \tag{2}$$

$$\overline{S} = \frac{1}{N} \sum_{i=1}^{N} sin(\alpha_i) \tag{3}$$

These coordinates are used to estimate the mean size of $\overline{R}$, as:

$$\overline{R} = \sqrt{\left(\overline{C}^2 + \overline{S}^2\right)} \tag{4}$$

Then, the circular mean direction can be obtained by solving one of the following equations:

$$cos(\overline{\alpha}) = \frac{\overline{C}}{\overline{R}}, \qquad sin(\overline{\alpha}) = \frac{\overline{S}}{\overline{R}} \tag{5}$$

Finally, the circular variance of $\overline{\alpha}$ is calculated as:

$$S_\alpha = 1 - \overline{R} \qquad\qquad 0 \le S_\alpha \le 1 \tag{6}$$

To estimate $\overline{\alpha}$ and S_α for each transition of a numeral image, we have considered $\{0^o, 45^o, 90^o, 135^o, 180^o, 225^o, 270^o, 315^o\}$ as the set of directional observations, while $F(\alpha_i)$ is computed by counting the number of successive black pixels over the direction α_i from a transition until the encounter of a white pixel. In Figure 2 the transitions in a column of numeral 5 are enumerated from 1 to 6, and the possible directional observations from transitions 3 and 6 are shown.

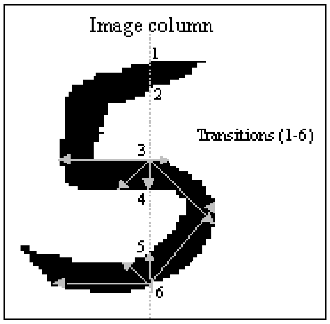

Figure 2. Transitions in a column image of numeral 5, and the directional observations to estimate the mean direction for transitions 3 and 6

In addition to this directional information, we have calculated two other local features: a) relative position of each transition, taking into account the top of the character bounding box, and b) whether the transition belongs to the outer or inner contour, which shows the presence of loops in the character image. Since for each column we consider 8 possible transitions, at this point our feature vector is composed of 32 features.

The global features are based on vertical projection (VP) of black pixels for each column, and the derivative of VP between adjacent columns. This constitutes a total of 34 features normalized between 0 and 1.

Background features (BF): The BF vector is based on concavity information. These features are used to highlight the topological and geometrical properties of the character classes. Each concavity feature represents the number of white pixels that belong to a specific concavity configuration.

The label for each white pixel is chosen based on the Freeman code with four directions. Each direction is explored until the encounter of a black pixel or the limits imposed by the character bounding box. A white pixel is labeled if at least two consecutive directions find black pixels. Thus, we have 9 possible concavity configurations. Moreover, we consider four more configurations, in order to detect more precisely the presence of loops.

The total length of this feature vector is then 13. The concavity vector is normalized between 0 and 1, by the total of the concavity codes computed for each column or row of the digit image.

Combination of FF and BF features: a single feature vector composed of foreground and background features is extracted from each column and row of the character image. Each feature vector is mapped to one of 256 possible discrete symbols available in a codebook previously constructed by using the K-means algorithm [8]. Thus, the output of the feature extraction method consists of two sequences of discrete observations for each digit: column-based and row-based sequences.

2.2 Hidden Markov models

In the proposed classifier each character class is represented by two HMMs: one based on columns ($\lambda_c^a, \lambda_c^b, ..., \lambda_c^z$) and other based on rows ($\lambda_r^a, \lambda_r^b, ..., \lambda_r^z$) of the character image. These column- and row-based models provide a way of combining foreground and background features in a zoning scheme as shown in Figure 3.

The topology of the numeral models is defined taking into account the recognition of handwritten text. This means a left-right model with number of states defined take into account durational statistics calculated from the training database [8].

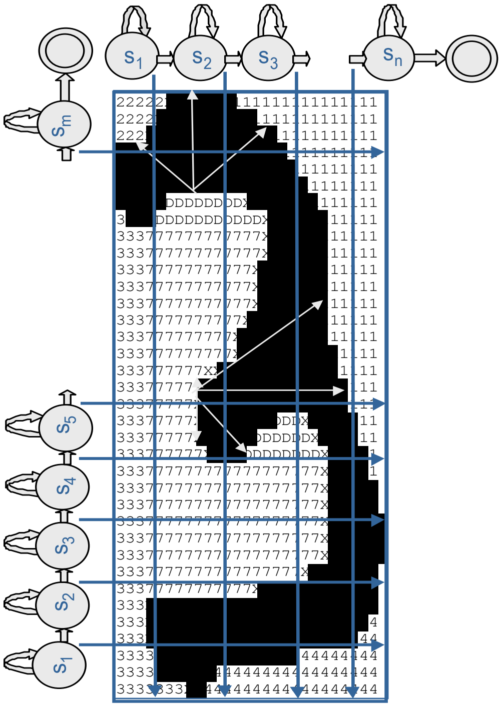

Figure 3. Complementary information extracted from columns and rows of character images and combined in an HMM-based method

3. Strategies to improve the system

Two strategies to combine the complementary features described in the previous section were evaluated, and also the scheme used to reduce the impact of possible noise in the training set. Similar to the LOOT (Leave-One-Out) method proposed in [11], a new sequence is generated by leave one observation out of the original sequence. However, here a non-repetition scheme (NR) is used. In addition to the NR scheme, a filter (NRF) is used to carry out this process.

3.1 Combination of the Complementary Features (FF and BF)

As described before, the FF and BF features are combined in a single feature vector. In this work, we have investigated two different strategies to combine these complementary features, as follows:

Multiple *B* matrices: the *B* matrix of a discrete HMM contains the probability symbol distribution. In this system configuration, the HMMs have two different B matrices – one containing the probabilities related to the foreground features, and other containing the probabilities related to the background features. Thus, we have the same two models per class: column-based and row-based models. However, each model has the following parameters: $\lambda=\left(A,B^0,B^1,\pi\right)$ - the state transition probability distribution (*A*), two *B* matrices – B^0 (foreground-based) and B^1 (background based), and the initial state probability distribution(π). The objective is to better represent the features in the character HMMs. In the training process (Baum-Welch algorithm [10]) and in the recognition process (Viterbi algorithm [10]) $bj(k)$ is replaced by $bj\left(k^0,k^1\right)$ which is computed as:

$$bj\left(k^0,k^1\right)=b^0j\left(k^0\right)\times b^1j\left(k^1\right) \tag{7}$$

Multiple models: in this strategy each subset of feature (foreground and background-based) are used for training a specific character HMM. Thus, the system configuration is composed of four character models: foreground column-based HMM, background column-based HMM, foreground row-based HMM and background row-based HMM. During the recognition the models are combined by sum the *log* of their probabilities.

3.2 Dealing with noise in the training set

Similar to the leave-one-out method proposed in [11], we increase the number of observation sequences in the training database trough the generation of new sequences from the original ones. The idea is to minimize the negative impact of noise in the set of observation sequences used for training the HMM characters. However, we have evaluated two different strategies to carry out the generation of new sequences: a) the use of a non-repetition scheme (LOT-NR) and b) the use of a non-repetition plus a filtering process (LOT-NRF).

LOT-NR (Leave-one-out with non-repetition): instead of considering all new sequences generate by the leave-one-out scheme, we avoid the generation of repeated sequences. For instance, consider the following hypothetical sample:

Hypothetical observation sequence: (a,b,b,a,a,c)	
Leave-one-out strategy	Leave-one-out with non-repetition (LOT-NR)
New sequences (b,b,a,a,c) (a,b,a,a,c) (a,b,a,a,c) (a,b,b,a,c) (a,b,b,a,c) (a,b,b,a,a)	New sequences (b,b,a,a,c) (a,b,a,a,c) (a,b,b,a,c) (a,b,b,a,a)

We can see that with the use of the LOT-NR scheme (second column) only different sequences are created.

LOT-NRF: (Leave-one-out with non-repetition and filtering): a filtering process is added to the LOT-NR scheme. With such a filter, only observations which low frequency in the training set considering the corresponding character class is used. The idea is to leave out just the observations with low frequency in the training set for a specific character class. A threshold value (empirically defined) is used to define which observations will be considered. See the following hypothetical sample:

Original sequence: (a,b,a,a,c)
Frequency of each observation (calculated by class in the training set): a=0.3%, b=0.03%, c=0.01%
Threshold value: 0.1%
New sequences: (a,a,a,c), (a,b,a,a)

4. Experimental Results

The experiments undertaken for the evaluation of the proposed strategies were done using isolated characters from the NIST SD19. We have used 74,880 character samples from *hsf_0, hsf_1, hsf_2,* and *hsf_3* for training, 23,670 from *hsf_7* for validation and 23,941 from *hsf_4* for testing.

Table 1. **Different strategies to combine the FF and BF features in the HMM-based system (rec. rates in %)**

Samples	Single vector	Multiple B matrices	Multiple models
lowercase	84.0	85.03	84.94
uppercase	90.0	91.44	91.14
52 classes	87.0	88.27	87.76

Table 1 presents the recognition performance of the different strategies used to combine the FF and BF features: in a single vector (original system), using multiple B matrices and multiple HMMs. As we can see, the separation of the two vectors has shown to be a promising strategy in both investigations. The better results were achieved by using multiple B matrices.

We have used the multiple models configuration to investigate our strategies to minimize the impact of possible noise in the training set. We could not use the multiple *B* matrices approach since for combining two sequences of observations it is mandatory to have the same number of observations. Table 2 presents the recognition rates when we consider our two modified leave-one-out strategies: LOT-NR and LOT-NRF. The best results when using LOT-NRF were found by setting the threshold values to 0.12 and 0.16 for upper and lowercase, respectively. As we can see in Table 3, we have improved the original system performance presented in [8], and also we obtained similar results than those based on Neural Networks, but with the advantage that the proposed HMMs have been used for numeral strings and words without any modification.

Table 2. Reduction of the negative impact of noise in the training set (rec. rates in %)

Samples	LOT-NR	LOT-NRF	
lowercase	84.0	85.67	Threshold :0.12
uppercase	90.0	91.62	Threshold :0.16
52 classes	87.0	88.22	

Table 3: Related works (results on NIST database)

Method	# Tr	# Val	# Test	Rec. (%)
(Oh, 1998) [5] lowercase (26 classes)	26,000	-	11,941	90.0
(Dong, 2001b) [6] lowercase (26 classes)	23,937	-	10,688	92.3
(Koerich, 2002) [4]				
uppercase (26 classes)	37,440	12,092	11,941	92.3
lowercase (26 classes)	37,440	11,578	12,000	84.6
both (52 classes)	74,880	23,670	23,941	85.5
(Britto, 2004) [8]				
uppercase (26 classes)	37,440	12,092	11,941	90.0
lowercase (26 classes)	37,440	11,578	12,000	84.0
Both (52 classes)	74,880	23,670	23,941	87.0

5. Conclusions

We have investigated different strategies to improve the character recognition performance of a HMM-based system. Some improvement was achieved by representing the complementary features of the original system in different B matrices or even in different character HMMs. In addition, the scheme used to generate new observation sequences from the original ones has shown to be an interesting strategy to reduce the impact of possible noise in the training set. Further work may be done by evaluating different ways to combine the model outputs in the new system configuration.

6. References

1. Oliveira, L.S. and Sabourin, R. Support Vector Machines for Handwritten Numerical String Recognition, 9th Int.. Workshop on Frontiers in Handwriting Recog. (IWFHR-9), Kokubunji, Tokyo, Japan, pp 39-44, 2004.
2. Dong J.; Krzyzak A. and Suen C. Y. A muti-net learning framework for pattern recognition. Proc. of the Sixth Int. Conf. on Doc. Analysis and Recog., pp. 328-332, 2001.
3. Suen C. Y., Xu Q., and Lam L. Automatic recognition of handwritten data on cheques - fact or fiction? Pattern Recog. Letters, 20(13):1287-1295, 1999.
4. Koerich A. L. Large Vocabulary Off-Line Handwritten Word Recognition. PhD thesis, École de Technologie Supérieure, Montreal-Canada, August, 2002.
5. Oh I.S. and Suen C.Y. Distance features for neural network–based recognition of handwritten characters. Int. Journal on Document Analysis and Recognition, 1(2):73–88, 1998.
6. Dong J.; Krzyzak A.; and Suen C.Y. Local learning framework for recognition of lowercase handwritten characters. In Proc. Int. Workshop on Machine Learning and Data Mining in Pattern Recognition, Leipizig, Germany, pp. 226-238, 2001.
7. Koerich A. L.; Sabourin R.; and Suen C. Y. Lexicon-Driven HMM Decoding for Large Vocabulary Handwriting Recognition with Multiple Character Models. *Int. Journal on Doc. Analysis and Recognition (IJDAR*), Vol.6 No.2, pp.126-144, October 2003.
8. Britto JR., A. S.; Sabourin R.; Bortolozzi F.; and Suen C.Y. Foreground and Background Information in an HMM-Based Method for Recognition of Isolated Characters and Numeral Strings. 9th Inter. Workshop on Frontiers in Handwriting Recognition (IWFHR-9), Tokio Japan, pp. 371-376, 2004.
9. Cavalin, P.; Britto JR, A. S.; Oliveira, L. E. S.; Sabourin, R.; and Bortolozzi, F. An Implicit Segmentation based Method for Recognition of Handwritten Strings of Characters. In: Proc. of the 21st Annual ACM Symposium on Applied Computing, 2006. v. 1. pp. 836-840.
10. Rabiner, L. R. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proc. of the IEEE, Vol. 77, No. 2, pp.257-286, 1989.
11. Ko, A.; Sabourin, R.; and Britto Jr, A. S. Leave-One-Out-Training and Leave-One-Out-Testing Hidden Markov Models for a Handwritten Numeral Recognizer: the Implication of Single Classifier and Multiple Classifications. IEEE Trans. on Pattern Analysis and Machine Intelligence, Oct.,2008.