OFF-LINE CURSIVE SCRIPT RECOGNITION BASED ON CONTINUOUS DENSITY HMM

A.VINCIARELLI AND J.LUETTIN
IDIAP - Institut Dalle Molle d'Intelligence Artificielle Perceptive
Rue du Simplon 4, CP592 - 1920 Martigny, Switzerland
{*vincia,luettin*}*@idiap.ch*

A system for off-line cursive script recognition is presented. A new normalization technique (based on statistical methods) to compensate for the variability of writing style is described. The key problem of segmentation is avoided by applying a sliding window on the handwritten words. A feature vector is extracted from each frame isolated by the window. The feature vectors are used as observations in letter-oriented continuous density HMMs that perform the recognition. Feature extraction and modeling techniques are illustrated. In order to allow the comparison of the results, the system has been trained and tested using the same data and experimental conditions as in other published works. Performances comparable to those of more complex systems have been achieved.

1 Introduction

The off-line cursive script recognition (CSR) problem has been deeply studied in the last ten years. Although the range of proposed methods is very wide, these can be classified depending on two key properties: the size and nature of the lexicon involved, and whether or not a segmentation stage is present [1]. The lexicon is related to the application environment the recognizer is embedded in. Ambiguity over the last one, two or three digits of the zip code determines the size of the lexicon (between 10 and 1000) in postal applications [2,3]. Check amount recognition needs a small dictionary of numbers written in letters [4,5]. To read generic content documents involves very large lexica, in principle the whole dictionary of the language of the document [6,7,8]. The segmentation consists in isolating the single characters in a word so that each one of them can be separately recognized. Such task is difficult and error prone, since a character cannot be recognized before having been segmented, but cannot be segmented before having been recognized. This is referred to as the Sayre's paradox [1].
We use a sliding window that makes the segmentation unnecessary: by sliding the window column by column from left to right, frames of fixed width are extracted from the image. From each frame, a feature vector is extracted and the sequence of observations so obtained is used as the representation of the word. For each entry in the lexicon, a word-HMM is created by concatenating

In: L.R.B. Schomaker and L.G. Vuurpijl (Eds.), Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition, September 11-13 2000, Amsterdam, ISBN 90-76942-01-3, Nijmegen: International Unipen Foundation, pp 493-498

single letter HMMs. The use of letter models makes the system flexible with respect to changes in dictionary, and makes the use of large lexica possible since it doesn't require training examples of each word.
The paper is organized as follows: section 2 presents the preprocessing and normalization techniques, sections 3 and 4 describe the feature extraction process and the HMM based recognition, respectively, and the final section 5 illustrates results and conlusions.

2 Preprocessing and normalization

Before being processed, the word images are binarized, deslanted and desloped using techniques described in detail in [9]. The desloping technique consists in finding first a rough estimate of the core region, then in using the stroke minima close to its lower limit to fit the lower base line. The image is then rotated until the lower base line is horizontal. The first estimate of the core region is fundamental and is found by thresholding (with the Otsu method [10]) the distribution of the horizontal density (number of foreground pixels per line) values. The use of the distribution (instead of the density histogram itself) is preferred because the noise introduced by local features (e.g long horizontal strokes in ascenders or descenders) of the word becomes in it statistically irrelevant.
The deslanting is based on the hypothesis that, when the slant is minimum, the number of vertical strokes is maximum. A measure of the number of vertical strokes is given by the number of columns where the are no background pixels between the highest and lowest foreground pixel. If y_h and y_l are the heights of the highest and lowest foreground pixel respectively in such columns, the sum of the $(y_h - y_l)^2$ values can be calculated. After having applied a shear transform to the original image for each angle in a reasonable interval, the value of the sum is obtained. The angle corresponding to the shear transformed image that gave the highest sum is used as slant estimate. The described preprocessing does not involve any heuristic parameter and is completely adaptive.

3 Feature extraction

A sliding window that moves from left to right by one column isolates $nCol - width$ frames ($width$ is the width of the window and $nCol$ the number of columns in the image). The feature extraction process is not applied to the whole frame, but only to the area actually containing foreground pixels (eventual white lines at the top and at the bottom of the frame are discarded). Such

area is partitioned into 16 non overlapping cells (arranged in a 4×4 grid), each one containing a percentage f_i of foreground pixels in the frame, the vector $\mathbf{f} = (f_1, f_2, \ldots, f_{16})^T$ represents the feature vector.

4 HMM based word recognition

The limited space does not allow an exhaustive presentation of Hidden Markov Models, for a good introduction, see [11].
A HMM can be tought of as a probability density function over sequences of observations. In general, the handwritten data is fragmented into parts that are supposed to belong to some finite set of basic strokes. The vectors extracted from the fragments should form clusters corresponding to the elements of such set. The process that extracts the strokes from the word is called segmentation, at the end of the feature extraction process, the word is reduced to a sequence of symbols that is given as input to a discrete HMM. In our case, the segmentation is not performed, the observations are vectors extracted from blindly isolated frames and are supposed to be distributed more uniformly than in the previous case, that might result in a large quantization error due to the high variance within classes. For this reason, the use of continuous density HMMs has been preferred. The observations are modeled with mixtures of gaussians.
A model is built for each letter. This has mainly two advantages: the first one is the flexibility with respect to changes in dictionary, the second one is that it is not necessary to have training examples of each word in the lexicon. Due to the lack of capital letters in the training database, only small letters were modeled.
A word model is composed by a chain of letter models, when the final state of a letter is reached, the next step can be a self transition or a transition to the first state of the model of the following letter in the word. The letter models have a left right topology, i.e. only self transitions or transitions to the following state are allowed (see fig 1).
The training procedure, performed with the Baum-Welch algorithm [11], is not applied to single letter models but to word models and is based on the maximum likelihood criterion. This technique is called *embedded training* and avoids the need of knowing the boundaries between neighboring letters in the observation sequences. This is an advantage because otherwise a time consuming and error prone manual labeling of the training data would be necessary.
The trained models allow to calculate the probability of a certain sequence of letters for having generated the observed vectors. The recognition consists in

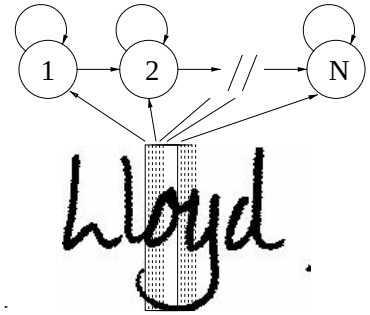

Figure 1. Markov modeling. The sequence of observations extracted from frames isolated by the window (that shifts column by column) is matched with the letter model using the Viterbi Algorithm.

finding the sequence of letters with the highest probability for a given observation sequence, in other words, in finding the most likely letter sequence.
When the sequence of letters is unconstrained, the problem of finding the most likely sequence is equivalent to finding the most likely path trough a ergodic model where each letter is connected to any other letter (path discriminant approach). When the sequence of letters is constrained to correspond to one of the words in the lexicon, the same problem is equivalent to finding the most likely word model (model discriminant approach). In both cases, the recognition is performed with the Viterbi algorithm [11].
The number of states is the same for all letter models as well as the number of mixtures of gaussians components used to compute the observation probability.

5 Results and conclusions

For training and testing, we used a database collected by Senior and Robinson[8] that is publicly available on the web[a]. The data consists of 4053 words extracted from a text that belongs to the LOB Corpus and has been written by a single person. The experimental conditions described in [8] have been reproduced: the database has been divided in training, validation and test set (2360, 675 and 1016 words respectively). The lexicon size is 1334. The performance is measured in terms of correctly classified letters and words (with and without lexicon).

[a] ftp://svr-ftp.eng.cam.ac.uk/pub/data

Several systems have been trained and tested to find the optimal configuration. State numbers from 1 to 10 and windows 10, 12 and 16 pixel wide have been used. The observations have been modeled with mixtures of gaussians with 1, 2 and 4 components. The best results have been achieved by a system with 9-state letter models, a window 10 pixels wide and a gaussian mixture with 2 components. The performance is reported in table 1.

The observation of the words uncorrectly classified, showed that horizon-

Table 1. Results. Recognition rate in terms of character and words (with and without lexicon.

character(%)	word (%)	word+lexicon(%)
67.55	10.88	82.45

tal strokes, in ascenders or descenders, wider than the character they belong to (see the lower part of the y in fig. 1) can cause noise in the frames of the neighboring letters. When such strokes are present it is possible to find empty rows between the base line (upper line) and the bottom (top) of the frame. The first empty rows below the base line and above the upper line are then detected and only the area between them is used to extract the features. The best performance for this technique has been achieved by a system with 9-state letter models, a window 10 pixel wide and a gaussian mixture with 1 component (see table 5). The best results obtained by Senior and Robinson

Table 2. Results after eliminating noise caused by horizontal strokes in ascenders and descenders. The first column reports the character recognition rate, the second one the word recognition rate without lexicon and the third one the word recognition rate with lexicon.

character(%)	word (%)	word+lexicon(%)
71.78	15.90	83.57

in [8] are 93.4% with lexicon and 58.9% without lexicon (no data are available at the character level), but their system is much more complex than the one described here. Other results over the same data can be found in [7], where linguistic models, working at the sentence level to improve the word recognition rate, are used. Continuous density HMMs are also used in [12], where a segmentation free system is used in conjunction with a segmentation based one in order to increase the performance.

This work presents a simple system for cursive script recognition. An effort has been made in order to limit as much as possible the use of heuristics.

Deslant and deslope techniques are adaptive and make no use of heuristic parameters. The features are robust with respect to the high variability of the input patterns and the system represents one of the few applications of the continuous density HMMs in the field of CSR.
The results, still to be improved with further work, are comparable to those of more complex systems.

References

1. T. Steinherz, E. Rivlin and N. Intrator, Off-Line Cursive Script Word Recognition - A Survey, Int. J. of Document Analysis and Recognition **2**, 1 (1999).
2. A. El-Yacoubi, M. Gilloux, R. Sabourin and C.Y. Suen, An HMM-Based Approach for Off-Line Unconstrained Handwritten Word Modeling and Recognition, IEEE Trans. on PAMI **21**, 752 (1999).
3. A. Kundu, Y. He and M.Y. Che, Alternatives to Variable Duration HMM in Handwriting Recognition, IEEE Trans. on PAMI **20**, 1275 (1998).
4. S. Knerr, E. Augustin, O. Baret and D. Price, HMM Based Word Recognition and its Application to Legal Amount Reading on French Checks,Computer Vision and Image Understanding **70**, 404 (1998).
5. T. Paquet and Y. Lecourtier, Recognition of Handwritten Sentences Using a Restricted Lexicon, Pattern Recognition **26**, 391 (1993).
6. H. Bunke, M. Roth and E.G. Schukat-Talamazzini, Off-Line Cursive Handwriting Using HMMs, Pattern Recognition **28**, 1399 (1995).
7. U. Marti and H. Bunke, Towards General Cursive Script Recognition, in *Proc. of 6^{t}h Int. workshop on Frontiers in Handwriting Recognition*, (Korea, 1998).
8. A.W. Senior and A.J. Robinson, An Off-Line Cursive Handwriting Recognition System, IEEE Trans. on PAMI **20**, 309 (1998).
9. A.Vinciarelli and J.Luettin, Off-Line Cursive Script Recognition Based on Continuous Density HMM, IDIAP Research Report IDIAP-RR 99-25, 1999.
10. R.M. Haralick and L.G. Shapiro, *Computer and Robot Vision*, (Addison wesley, USA, 1992).
11. L. Rabiner and B.H. Juang, *Fundamentals of Speech Processing*, (Prentice Hall, Englewood Cliffs, NJ, 1992).
12. M. Mohamed and P. Gader, Handwritten Word Recognition Using Segmentation-Free Hidden Markov Modeling and Segmentation Based Dynamic Programming Techniques, IEEE Trans. on PAMI **18**, 548 (1996).