Text Recognition from Grey Level Images Using Hidden Markov Models

Kjersti Aas Line Eikvil Tove Andersen
Norwegian Computing Center, P.O. Box 114 Blindern, N-0314 Oslo, Norway
e-mail: Kjersti.Aas@nr.no Tel: (+47) 22 85 25 00 Fax: (+47) 22 69 76 60

Abstract. The problems of character recognition are today mainly due to imperfect thresholding and segmentation. In this paper a new approach to text recognition is presented which attempts to avoid these problems by working directly on grey level images and treating an entire word at the time. The features are found from the grey levels of the image, and a hidden Markov model is defined for each character. During recognition the most probable combination of models is found for each word by the use of dynamic programming.

1 Introduction

One of the remaining problems in character recognition, is segmentation. Both segmentation of text from background in the grey level image, and segmentation of the constituents of each character in the binary image, is a problem. In this paper we present a method which attempts at avoiding these segmentation problems by working directly on the grey level image and perform recognition word by word instead of character by character.
Little work has been done in the area of text and character recognition from grey level images, but two examples can be found in [1] and [2]. The latter also gives an overview of previous work in this field. The method we have used in this study is very simple and straight forward. Word recognition has been payed more attention, and in [3] [4] [5], some studies based on the use of hidden Markov models (HMMs) can be found. We have chosen a method similar to that presented in [5], where HMMs are used to describe each character class. The word is then matched with all possible combinations of models, using dynamic programming to find the string that gives the best match.

2 Hidden Markov models

Hidden Markov models (HMM) together with dynamic programming have been widely used in the field of speech recognition [7] [8]. Lately these methods have also appeared within the field of optical character recognition. Some examples of applications are handwritten word recognition [4] and keyword spotting [3].
Two main approaches have been used, constructing a Markov model either for each word or for each character. The latter approach is chosen for our application, because it does not require a limited vocabulary.

A hidden Markov model is a doubly stochastic process with an underlying stochastic process that is not observable (hidden) but can only be observed through another set of stochastic processes that produce sequence of observations [8]. The hidden Markov model requires 3 probability measures to be defined, the transition probabilities of the underlying Markov chain A, the observation probabilities B and the initial probability vector Π. All the HMMs considered in this paper are assumed to be first order, left-to-right models with N states.

3 Training

Samples from each character class are first manually labelled. The grey level image is temporarily thresholded, and connected components are extracted and labelled with the correct class label. The characters are represented by their bounding box and location in the image, and features are extracted from the original grey level image. A class description containing estimates of the parameters needed to define the hidden Markov model, is computed for each character, based on the extracted features.

3.1 Feature extraction

The location of the binary character and its bounding box are used as a reference to the corresponding grey level image. However, prior to the feature extraction, the position and size of the circumscribing box must be normalized. We require that this box should have the same height for all characters. During training, the class membership of each character is known, and from this the position of the normalized box is found. They grey levels of the subimage, defined by the normalized box, are then calibrated before the features are extracted.
Features are extracted for each column of the subimage, and for each pair of consecutive grey levels within a column, the minimum value is chosen. The features were mainly chosen for their simplicity. They are easily extracted and robust, as no complicated computations are required. Furthermore, the features are easily combined with the Markov models for the characters. Similar features, although extracted from binary images, are used in [6]. In our study, we have not made attempts at making the features invariant to scale and rotation. Invariance to fonts, can to a certain extent be obtained through careful design of the Markov models [6].
A disadvantage of these features is that the amount of data will be relatively large and may increase the processing time needed for the classification. However, this will be partly compensated for as the computations needed for the feature extraction are very simple.

3.2 Parameter estimation

For each character model, the set of parameters $c = (A, B, \Pi)$ has to be estimated from a training set with a sufficient number of samples from each character

class. In addition, the number of states per model must be defined. To simplify the training procedure, avoiding careful manual design of the models, an equal number of states per model was desirable. Each state in the HMM has a geometric duration probability, and experience has shown that exponentials are not good models for state duration probabilities. Hence, we chose to define a relatively large number of states per model, and initially designed each model to have 6 states. However, as we do not allow any state to be skipped, the number of frames for a character must be equal to or exceed the number of states for the corresponding models. Therefore, models with a smaller number of states were designed for the shortest characters; "l", "i", ",", and ".".
The transition probabilities were constrained to

$$a_{ij} = 0, \quad j < i, \quad j > i + 1,$$

meaning that the states proceed from left to right and no states are allowed to be skipped. Working on grey level images, we do not expect any parts of the characters to be missing. In view of this, the initial state probabilities were assumed to be

$$\pi_i = \begin{cases} 1 \text{ if } i = 1 \\ 0 \text{ otherwise} \end{cases}$$

The feature vectors were assumed to be continuous with a Gaussian density of the form

$$b_j(O_t) = \frac{\exp^{-\sum_{d=1}^{D}(O_{td}-\mu_{jd})^2}}{(2\pi)^{D/2}} \tag{1}$$

where D is the dimension of the feature vector. That is, the components of O_t were assumed to be uncorrelated. This assumption is not necessarily correct, but due to the extra processing time needed for the conversion of covariance matrices and the risk of numerical instabilities in the computations, we still chose this solution.
The means in equation 1, μ_{jd}, $1 \leq j \leq N, 1 \leq d \leq D$ have to be estimated. For this the the segmental K-means method [8] was used. Although all the parameters Π, A, B can be estimated by this method, it was only used for the estimation of the means in equation 1. The transition and prior probabilities were kept fixed during the training.

4 Recognition

Based on the grey level image, a sequence of feature vectors are extracted for an entire word at the time. A modified Viterbi algorithm is used to match these sequences of frames against the HMMs of the single characters. A level building algorithm keeps track of the path yielding minimum distance for the string up to any segment, enabling the final identification of the optimal sequence of models. Section 4.1 describe the feature extraction, while the level building algorithm is described in section 4.2.

4.1 Feature extraction

A grey level image is thresholded, and from this binary image connected components are identified and grouped into words. The words are sorted according to their location in the document. Based on the knowledge of the words' location, the further processing can now be performed on the original grey level image.
The bounding box of each word is normalized before the features are extracted, and the grey values of the resulting box are calibrated. At this point the class membership of the characters is unknown. Therefore the normalization is performed relative to the baseline, which is found by investigating the grey levels of each scanline of the normalized box, starting from the bottom. When the average grey level goes below a limit and the difference from the average grey level of the scanline below is small, we assume that we have found the baseline. The limits are estimated from the grey levels of the image.
From the normalized box the features are extracted for each column, in the same way as during training, resulting in a list of frames. Before the frames are sent to recognition, empty frames are removed. By empty frames we mean the background frames between characters. These are identified by investigating the minimum grey value of the frames. The reason for removing these frames, is that the training was performed only on single characters where there are no empty frames. Although the frames are removed, the information about where the empty frames were found is stored to be used later.

4.2 Level Building

The frames found during the feature extraction are input to a level building procedure [7], where they are matched against the single character HMMs through a sequence of Viterbi matches [8]. The issue of the level building is to determine the optimum sequence of HMMs. At the first level possible candidates for the first character of the word are found, at the second level possible candidates for the second character are found and so on. This process is repeated through a number of levels corresponding to the maximum expected number of characters, L, for the current word. Except for the initialization step, the match is identical at each level.
Through the level building, contextual knowledge, like the transition probabilities between characters, is incorporated. We have used the probabilities reported by Konheim [9], which give the probabilities for 1-state transitions between two sucessive letters in the English language. Transitions from small letters to capital letters were not allowed, nor were transitions between capital letters and period or comma. Transition from a small letter to a comma or a period is assigned the same probability as the most likely transition from that letter to another small letter. The words to be recognized were sorted and appeared in the same order as in the document, and we were thereby also able to look at character transition probabilities between words. This was utilized for the case where a word ended with the character ".". In this case we required the next word to start with a capital letter.

It is possible to expand or contract a hidden Markov model so that it accounts for a large or small part of the of the frame sequence, even though characters on which the model was trained might never have been as long or as short as the obtained matches. Such problems can be eliminated by building durational constraints into the algorithm. The character duration density, $P_{dur}(\tau, c)$ was defined to be 1 if the duration of the current model was within the prescribed linmits and 0 otherwise.

For each model we also check whether it is legal relative to the current interval of frames. The current interval will not have any legal models, if a frame has been removed within the interval. The frame vectors in the interval are also checked for ascenders and descenders, and from this the current model is classified as legal or illegal.

For each HMM, c, and at each level l we do a Viterbi match against the frames sequence and retain for each possible t $P(l,t,c)$ and $B(l,t,c)$. The first is the accumulated log probability to frame t at level l for HMM c along the optimal path, while the latter is a backpointer indicating where the path started at the beginning of level.

At the end of each level l, a maximization over c is performed to get the character model which gave the best match for position t, $\hat{W}(l,t)$, the probability for this match, $\hat{P}(l,t)$ and the backpointer of the best character model, $\hat{B}(l,t)$.

This process is repeated through the number, L, of levels equivalent to the maximum number of characters likely to appear in the current string. The best solution will be the maximum of $\hat{P}(l,T)$ over all levels l. The best character string is obtained by backtracking using the pointers in $\hat{B}(l,t)$.

5 Experimental Results

The training was performed on single labelled characters extracted from pages of printed text. The text was printed in times roman, with a fontsize of 10 pt, and was scanned at a resolution of 300 dpi with 256 greylevels. We had 54 different classes of characters, which included the small and capital english letters (a–z, A–Z), and point (".") and comma (",").

The data used in the recognition were fetched from a text page of the same font and fontsize as that of the training set. The page was scanned at 300 dpi with 256 grey levels, and contained 610 words, with a length varying from one to 15 characters. The total number of characters (including commas and points) were 3150.

25 of the words and 29 of the characters were wrongly classified, giving correct classification rates of 95.9 % and 99.06 %, respectively. A Sun SPARCstation 5 was used for training and testing the recognition system. The recognition phase, including the feature extraction, takes approximately 1.0 s per character, when no effort has been laid on increasing the efficiency of the code.

6 Summary and Discussion

The work presented in this paper shows that the use of HMM with Viterbi matching and level building is a promising technique for recognition of text in grey level images. Currently the efficiency of the algorithm is not sufficient, and this will be a subject for further studies. There are several ways of increasing the speed of the recognition, both by reducing the dimension and number of feature vectors and by making the implementation of the algorithm more efficient.
We think that a better result could have been obtained, if the Markov model for each of the characters had been manually designed. Also incorporation of probabilities for trigrams of characters, may increase the recognition rate. In this study, we have tested the method on a data set fetched only from one single document. In the future we would like to do more testing on documents of varying grey levels and contrast. We would also like to look more into the problems of invariance to scale and font.

Acknowledgements

Support for most of this research was provided by the Norwegian Research Council (NFR).

References

1. J. Rocha & T. Pavlidis: "A Shape Analysis Model with Applications to a Character Recognition System" *IEEE Trans. Pattern Machine Intell.*, Vol. 16, No. 4, 1994
2. L. Wang & T. Pavlidis: "Direct Gray-Scale Extraction of Features for Character Recognition" *IEEE Trans. Pattern Machine Intell.*, Vol. 15, No. 10, 1993
3. F. R. Chen, L. D. Wilcox & D. S. Bloomberg: "Detecting and Locating Partially Specified Keywords in Scanned Images using Hidden Markov Models" *Proc. Sec. Int. Conf. Doc. Anal. Recog.*, pp. 133–138, 1993.
4. M-Y. Chen, A. Kundu & J. Zhou: "Off-Line Handwritten Word Recognition Using a Hidden Markov Model Type Stochastic Network" *IEEE Trans. Pattern Machine Intell.*, Vol 16, No. 5, pp 481–496, 1994.
5. C. B. Bose & S. Kuo: "Connected and Degraded Text Recognition using Hidden Markov Model." *Pattern Recognition*, Vol 27, No. 10, pp. 1345–1363, 1994.
6. O. E. Agazzi & S-S. Kuo: "Hidden Markov Model based Optical Character Recognition in the Presence of Deterministic Transformations" *Pattern Recognition*, Vol 26, No. 12, pp. 1813–1826, 1993.
7. L. R. Rabiner and S. E. Levinson: "A Speaker-Independent, Syntax-directed, Connected Word Recognition System Based on Hidden Markov Models and Level Building." *IEEE Transaction on Acoustics, Speech and Signal Processing,* vol ASSP-33, No 3, p. 561–573.
8. L. R. Rabiner: "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition." *Proceeding of the IEEE,* vol 77, No 2.
9. A. G. Konheim: "Cryptography: A Primer.", John Wiley, New York, 1982