DATA DRIVEN DESIGN OF HMM TOPOLOGY FOR ON-LINE HANDWRITING RECOGNITION

Lee, J.L.; Kim, J.; Kim, J.H.

Published in:
EPRINTS-BOOK-TITLE

Publication date:
2004

DATA DRIVEN DESIGN OF HMM TOPOLOGY FOR ON-LINE HANDWRITING RECOGNITION

JAY J. LEE, JAHWAN KIM, AND JIN H. KIM

Dept. of Electrical Engineering & Computer Science, KAIST,
373-1, Kusong-dong, Yusong-gu, Taejon 305-701, Korea
E-mail: {joony, jahwan, jkim}@ai.kaist.ac.kr

Although HMM is widely used for on-line handwriting recognition, there is no simple and well-established way of designing the HMM topology. We propose a data-driven systematic method to design HMM topology. Data samples in a single pattern class are structurally simplified into a sequence of straight-line segments, and then these simplified representations of the samples are clustered. An HMM is constructed for each of these clusters, by assigning a state to each straight-line segments. Then the resulting multiple models of the class are combined to form an architecture of a *multiple parallel-path HMM*, which behaves as a single HMM. To avoid excessive growing of the number of the states, parameter tying is applied in that structural similarity among patterns is reflected. Experiments on on-line Hangul recognition showed about 19% of error reductions, compared to the previous intuitive design methods.

1 Introduction

As one of the major research directions for on-line handwriting recognition, hidden Markov model (HMM) is widely used because of the time sequential nature of on-line scripts as well as its capability of modeling shape variability in probabilistic terms. However, there has been no serious study or guidance in the design of HMM topology. Previous studies suggested that HMM should be designed depending on the signal being modeled [1, 2]. The model needs to have enough number of free parameters to accommodate complexity of target patterns and to reflect properties of the patterns. In practice, however, an arbitrary increment of the model parameters is not recommended, since available training samples are usually limited. Therefore, HMM topology should be determined based both on available training data and on the target pattern to be represented.

In this paper, we are focusing on two design parameters, *i.e.*, the number of states in HMM and the number of models for a class. Despite its importance, relatively little attention has been paid to the design of HMM topology. So far, suggested methods include intuition-based manual decision with empirical adjustment [3, 4], data-driven method by inferring the structural model of Markov network from a finite set of samples [5], and automatic state splitting method by maximum likelihood criterion [6].

We propose a data-driven method of design HMM topology for on-line handwriting recognition. Our design principle is that the HMM topology should be

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 239-248

constructed from the data, reflecting the structure of the target pattern. Here, we assumed that a target pattern is composed of straight-line segments. Accordingly, a sample of the target pattern can be structurally decomposed and simplified as a sequence of line segments. Then, the HMM has a state corresponding to each straight-line segment. To handle shape and writing-order variations present inside a class, sequences of straight-line segments, which are simplified representations of samples, are clustered to construct multiple models. The resulting multiple models for a single class are combined to form a single HMM architecture, called a *multiple parallel-path HMM*. For training, the initial observation probability distribution for each state is estimated from the distribution of corresponding straight-line segments, and then the Viterbi path training method is applied. When models for a single class have parts that are not simply similar in shape but *structurally* similar, corresponding states are tied. The number of parameters in the HMM are hence reduced to a manageable size.

The proposed method was evaluated using on-line Hangul (the Korean script) recognition, since Hangul graphemes are typically structured with line segments. Experiments showed that our method reduced about 20% of character recognition error compared to the previous design methods. We believe that the proposed design method can be applied to other scripts that are mostly composed of straight-line segments, such as Chinese characters.

The organization of the paper is as follows. Section 2 presents the data-driven design method of HMM topology, combining architecture of multiple models, and structural state-tying method. Section 3 introduces Hangul and the on-line Hangul recognition system briefly, then addresses the external duration modeling for performance improvement. Section 4 shows the experimental results of the proposed approach and analysis of the results. Conclusion is followed in Sect. 5.

2 Data-driven Design of HMM Topology

In this section, we will describe how to determine the number of states in HMM and the number of models for each pattern class, based on training samples. We will also explain how these multiple models for a single class are combined in the architecture of a *multiple parallel-path HMM*. Finally, the structure-based state-tying method to reduce the number of parameters will be explained.

2.1 Mapping Line Segment to HMM State

The number of HMM states roughly corresponds to the number and dynamics of signal `prototypes' being modeled. Thus, in modeling complex patterns, the number of states should be increased accordingly. However, the excessive number of states can generate the over-fitting problem when the number of training samples is insufficient compares to that of the model parameters [7].

In our design method, the number of HMM states is determined by the structural decomposition of the target pattern. Handwriting is structurally simplified as a sequence of the straight-line segments (see Fig. 1-(b)). After noise removal and smoothing operation, adjacent pen movements with similar directions are grouped into a single straight-line segment. An invisible pen-up movement between pen-down strokes is also inserted as an imaginary line. Average direction of the line segment with pen-down movement is encoded as one of the 16-direction codes, and the imaginary line is encoded by another 16-direction codes. We call the resulting sequence of direction code a *skeleton pattern* (Fig. 1-(c)). It is regarded as a simplified representation of the pen movement, since both hand vibrations and length variations are ignored but only directional information remains. It describes a time-sequential and global shape of the pen movements.

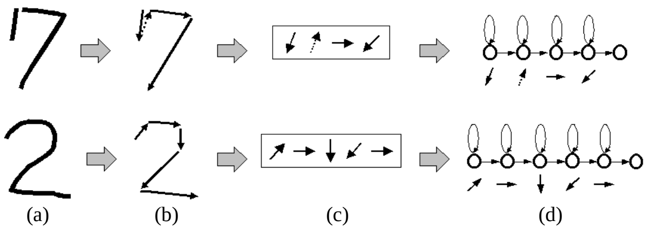


Figure 1: Examples of HMM topology design: (a) Handwriting sample, (b) Line segment approximation, (c) Skeleton pattern, (d) Resulting HMM

The structure of HMM is based on the skeleton pattern of a sample as shown in Fig. 1-(d). The transition structure of the model is a simple left-to-right one. The number of states is determined by mapping each straight-line segment into a single HMM state. Thus, each state assumes a uni-modal feature distribution only for the corresponding straight-line. Length variations of the straight-line are modeled in the self-loop of the state. As a consequence, each state of HMM corresponds to a straight-line segment of handwriting in time-sequential order. For this reason, an external knowledge can be utilized for the verification of the recognition result.

Since the maximum likelihood training method of HMM is a kind of steepest gradient search method, a good initial estimate, instead of random or uniform probability, is helpful for finding the global maximum of the likelihood function [1]. We can obtain initial observation probability distributions from the mapping relation between states and the skeleton pattern (see Fig. 2). The distribution of the line segments is accumulated from the training sample, and then its normalization is used as the initial parameter of the corresponding state. When the good initial observation probability distributions are given, the Viterbi path training method [8] works better and faster, compared to the usual Baum-Welch training method.

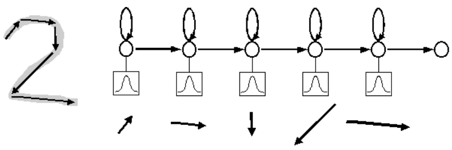


Figure 2: Initial observation distribution from skeleton pattern

2.2 Design of Multiple Models by Clustering

Whether or not having a single HMM to model whole patterns of a single class is also an important design decision. Skeleton patterns within a class, each of which is the simplified representation of sample, are clustered to determine the number of models for the class. The agglomerative clustering method [7] gathers skeleton patterns of the similar directional chain codes into a cluster. Since the skeleton patterns contain only principal pen movements, the proposed method collects the data samples of similar global shape as shown in Fig. 3-(c) and (d).

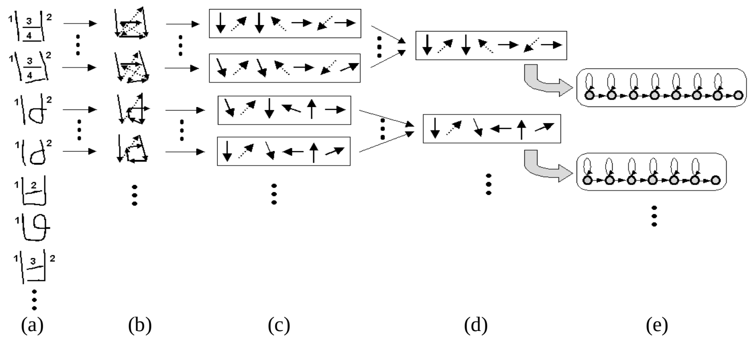


Figure 3: Example of multiple HMMs design – Hangul consonant 'ㅂ': (a) Handwriting sample, (b) Line segment approximation, (c) Skeleton pattern, (d) Representative pattern, (e) Resulting HMM

For the clustering, the distance between two skeleton patterns are defined as follows. The distance $D(X^1, X^2)$ of two skeleton patterns $X^1 = (x_1^1, x_2^1, \ldots, x_{L_1}^1)$ and $X^2 = (x_1^2, x_2^2, \ldots, x_{L_2}^2)$ is computed by *dynamic programming* using the recursion relation:

$$D(x_i^1, x_j^2) = d(x_i^1, x_j^2) + \min\{D(x_{i-1}^1, x_j^2), D(x_{i-1}^1, x_{j-1}^2), D(x_i^1, x_{j-1}^2)\},$$

$$D(X^1, X^2) = D(x_{L_1}^1, x_{L_2}^2)$$

where $d(x_i^1, x_j^2)$ is the directional difference of direction code x_i^1 and x_j^2.

The skeleton pattern that appears most frequently in each cluster is chosen as the *representative pattern* of that cluster (Fig. 3-(d)). An HMM is, then, constructed from each representative pattern. As a consequence, the number of representative patterns decides the number of models in a class, and the length of a representative pattern determines the number of states of the corresponding HMM. In our experiment, clusters containing only a small number of samples are disregarded to prevent generating too many models.

2.3 Combining Models to One Multiple Parallel-path HMM

One model for one class yields many benefits. It allows a modular design of the recognizer in that a model can be replaced with another easily. If different numbers of models exist in each class, models of the same class may compete for selection. It may hurt our attempt to select the top most labels for post-processing. In addition, *a prior* probability, which is obtained from language corpus, cannot be easily applied if there exist multiple models for one class label.

For these reasons, we propose a *multiple parallel-path HMM* (MPP-HMM) architecture, to combine the multiple models of the same class into a single HMM structure. Dummy initial and dummy final nodes are introduced and connected to multiple models of the same class. Then, these models are arranged in parallel. There is no connection between the multiple models. Thus, each constituent model forms one of the multiple paths from the dummy initial node to the dummy final one (see Fig. 4-(a)). *A prior* probability of each constituent model, $\Pr(\lambda_i)$, is assigned to the initial probability of the model π_i:

$$\pi_i = \Pr(\lambda_i) = K_i / K, \qquad \sum_{i=1}^{C} \pi_i = 1.0$$

where λ_i denotes the constituent HMM for cluster *i*, K_i is the size of data in cluster *i*, *K* is the size of all samples for the class, and *C* is the number of clusters in the class. Even though the MPP-HMM contains the multiple models inside, it behaves like a single HMM for the class. Re-estimation and Viterbi search algorithms still holds in this architecture. The structure of Fig. 4-(a) can be represented to the structure of Fig. 4-(b), which equals to the general left-to-right HMM structure.

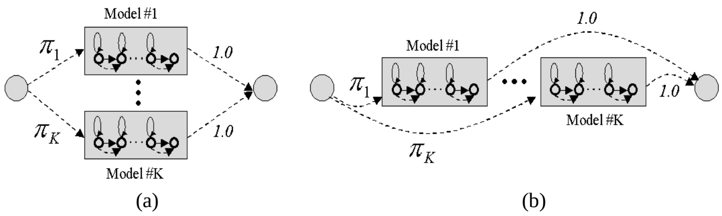


Figure 4: Architecture of multiple parallel paths HMM

2.4 State-Tying Based on Structural Similarity

Maintaining a balance of design between the model complexity and the amount of training data is critical for successful recognition system. Parameter-tying is one of the solutions for maintaining multiple models with limited training samples. Parameter-tying with HMM-based modeling is usually applied to states, actually observation probability distributions [2, 9].

Due to fast and sloppy writing, the pen movement is easily affected by the previous and the following pen directions, and therefore, simple shape similarity may not be robust for handwriting. Thus, not only the local closeness of output distributions but also their *structural* similarities are considered to determine the state-tying. The structural similarity is measured from the relative position of observation inside a pattern and from the global shape of the pattern. We can easily measure them by comparing representative patterns at the design phase.

The structural state-tying method is applied only to the states in the same class. The edit distance method (or the Levenshtein distance) [10] is used to compare representative patterns. Two representative patterns are considered for tying if their edit distance is within a threshold. Then, observation probability distributions are shared among the matched states. Figure 5 shows the example of the structural state-tying. The vertical line corresponding to the 1st state of a left model and the vertical line corresponding to the 1st state of a right model are tied. However, the same directional vertical line corresponding to the 3rd state of the right model is not tied to the 1st state of the left model because their relative positions are different.

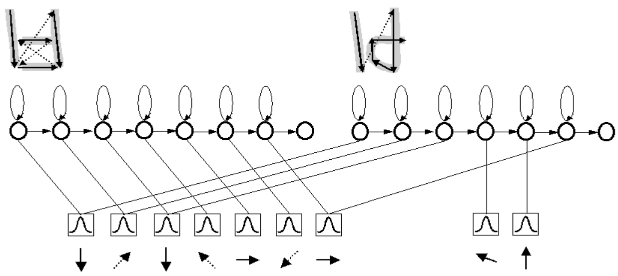


Figure 5: Example of state-tying based on structural similarity

3 Implementation for On-line Hangul Recognition

The proposed design method was evaluated using on-line Hangul recognition system. In this section, Hangul and on-line Hangul recognition system will be explained briefly. Also presented is the heuristic of external duration modeling which was introduced to achieve a level of duration modeling with small computational burden.

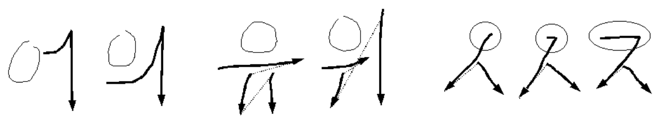
3.1 HMM Based On-line Hangul Recognizer

Hangul, the script used for Korean language, is a phonetic writing system. A character in Hangul, which corresponds to a single spoken syllable, is formed by spatial arrangement of either two or three graphemes: an initial consonant, a vowel, and a final consonant, if any. Each grapheme is formed by line segments consisting of sequential combination of horizontal, vertical, and/or diagonal lines. Therefore, a Hangul character can be easily decomposed and represented as a sequence of straight-line segments, although co-articulating effects deform the basic shapes. Several previous structural Hangul recognizers attempt to extract these basic line segments to recognize characters [11].

For the Hangul recognition system, we have used a HMM network-based approach in [3]. Discrete HMM was adopted to construct the grapheme and the ligature models. By sequential concatenation of grapheme and ligature HMMs as the order of writing characters, 5-layer finite state network was designed for all legal characters. Recognition is performed as finding the most likely path on this network by the modified Viterbi algorithm. For input code sequence, invisible pen-up movements as well as conventional pen-down strokes were encoded into observation symbols that consist of two sets of 16 direction codes.

3.2 External State Duration Modeling

States of normal HMM have exponential duration density inherently [1]. Such exponential duration characteristic is inappropriate for most physical signals. However, the durational information of the straight-line segments was not reflected on the proposed topology design method. Thus, patterns with the similar pen movements but largely different in their lengths can be often confused. Hangul has several confusing classes that have almost same pen movements as those of other classes (see Fig. 6-(a), (b)). Sometimes, handwritten input with a small noisy hook or serif may be mistaken to other class when no durational information is used (see Fig. 6-(c)).



(a) 'ㅓ' Vs. 'ㅢ' (b) 'ㅠ' Vs. 'ㅟ' (c) 'ㅅ' Vs. 'ㅅ' Vs. 'ㅈ'

Figure 6: Examples of confusing Hangul graphemes by similar pen movements

We have applied the external state duration method [12] in order to emphasize duration difference. This method adds random variables to the state duration outside the model. After model training, the duration information is calculated from the maximal paths of all training samples. During recognition, duration of each state is counted on the candidate Viterbi paths, and then, corresponding duration scores are added as the post-processing score. This explicit duration modeling needs a negligible amount of the computational overhead compares to the usual parametric or non-parametric state duration model. However, it is not profitable in case of a single model for a single class, because one state may model several different pen movements and thus hold various state durations.

4 Experiments

The data for experiments were collected without any constraint on writing style. Both printed and cursive handwriting styles were included in the data set. To train HMMs, we used about 85,000 Hangul characters in frequently used 2,350 Hangul character classes written by 84 writers. Since grapheme boundaries are not available, graphemes were manually segmented for the training.

For comparing our proposed design method against the previous design methods, the Hangul recognition system with a single HMM for each class [3] was used. The second row of Table 1 shows the results of this '*Single HMM*' system, in which the number of HMM states were tuned by the intuitive and empirical methods. Next, HMM was designed by the proposed method. The third row of Table 1 represents the recognition rates by the proposed multiple parallel-path HMM. The following fourth and fifth rows show the results after applying external state duration model and state-tying method, respectively.

Table 1: Correct recognition rates (%) of grapheme and character recognition tests

	Grapheme test	Character test
Single HMM	93.85 %	91.55 %
MPP-HMM	93.21 %	92.04 %
MPP-HMM + D	96.56 %	93.11 %
MPP-HMM + D + T	96.30 %	93.16 %

MPP-HMM: Multiple Parallel-Path HMM, D: external duration model, T: state-tying

Two kinds of tests were performed for evaluation of the performance of the proposed method. To examine how well each grapheme HMM was trained, recognition tests were performed for grapheme training data: 83,151 initial consonants of 19 classes, 83,536 vowels of 21 classes, and 49,431 final consonants of 27 classes. The second column of Table 1, labeled '*Grapheme test*', shows the recognition rates of these tests. Note that the use of state duration model gives remarkable performance improvement, *i.e.*, about 40 % error reduction. This improvement is due to the durational difference among confusing graphemes. Next, Hangul character recognition test was performed. The 12,140 characters were collected from 23 writers who did not participate in the training data. Two kinds of texts were used that are composed of 580 and 168 kinds of characters respectively. The third column of Table 1, labeled '*Character test*', shows the results. About 19% of recognition errors were reduced by the proposed method compared to the single model setting. Note that state-tying did not degrade the recognition accuracy.

Table 2: Increment of complexity (in SUN SPARC-II workstation)

	Single HMM	MPP-HMM	MPP-HMM after tying
Num. of models	107	258	258
Num. of different states	556	1070	562
Avg. recognition time	0.10 (sec/char)		0.19 (sec/char)

Table 2 shows the increment of complexity due to multiple models. The number of models was increased about 2.5 times with our method, compared to the single HMM system (an average of 2.5 models per class). The total number of the states was also increased about 2 times. However, the state-tying method reduced them to about a half. As a result, the number of free parameters for the observation probability distributions becomes almost the same as that of the single model setting. Next, the third row of Table 2 shows the increment of time complexity. Since search space of the recognition network broadened by the multiple model setting, average time for character recognition was increased about 2 times.

5 Conclusion

A data-driven systematic design method of HMM topology for on-line handwriting recognition was proposed. The number of models in each class and the number of HMM states in each model were determined by the structurally simplified sequences of the straight-line segments and their clusters. As a result, different handwriting styles were modeled by multiple HMMs, and their states were forced to correspond to the line segments of the target pattern in time sequential order. The multiple models in a class were combined in parallel to form the structure of the multiple parallel-path HMM, and then it behaves as a single HMM. States with structural similarity were tied, hence the number of HMM parameters were reduced. For the practical application of the system the external states duration modeling was applied. The experiments to on-line Hangul handwriting recognition showed that the proposed method reduced about 19% of the error rate compared to the intuitive design methods.

References

1. L. R. Rabiner and B. H. Juang, Fundamentals of Speech Recognition (Prentice Hall, 1993).
2. K. F. Lee, Automatic Speech Recognition (Kluwer Academic Publishers, 1989).
3. Bong-Kee Sin and Jin H. Kim, Ligature Modeling for Online Cursive Script Recognition, *IEEE Trans. on PAMI*, **19**, 6 (1997) pp. 623-633.
4. Jianying Hu, Sok Gek Lim, and Michael K. Brown, HMM Based Writer Independent On-line Handwritten Character and Word Recognition, *Proc. of IWFHR-6*, Taejon, Korea (1998) pp. 143-155.
5. M. G. Thomason and E. Granum, Dynamic Programming Inference of Markov Networks from Finite Sets of Sample Strings, *IEEE Trans. on PAMI*, **8**, 4 (1986) pp. 491-501.
6. Jun-ichi Takami and Shigeki Sagayama, A Successive State Splitting Algorithm for Efficient Allophone Modeling, *Proc. of ICASSP*, San Francisco (1992) pp.I. 573-576.
7. Richard O. Duda and Peter E. Hart, Pattern Classification and Scene Analysis (John Wiley & Sons, Inc., 1973).
8. L. R. Rabiner, J. G. Wilpon, and B. H. Juang, A Segmental k-means Training Procedure for Connected Word Recognition, *AT&T Technical Journal*, **65**, 3 (1986) pp. 21-31.
9. S. J. Young and P. C. Woodland, The Use of State Tying In Continuous Speech Recognition, *Proc. of EUROSPEECH '93, the 3rd European Conf. on Speech Communication and Technology*, Berlin, Germany (1993) pp. 2203-2206.
10. Robert A. Wagner, and Michael J. Fischer, The String-to-String Correction Problem, Journal of the Association for Computing Machinery, 21, 1 (1974) pp. 168-173.
11. H. Y. Kim and Jin H. Kim, Hierarchical Random Graph Representation of Handwritten Characters and its Application to Hangul Recognition, *Pattern Recognition* (in press).
12. L. R. Rabiner, B. H. Juang, S. E. Levinson, and M. M. Sondhi, Recognition of Isolated Digits Using Hidden Markov Models with Continuous Mixture Densities, *AT&T Technical Journal*, **64**, 6 (1985) pp. 1211-1234.