Hidden Markov Model Length Optimization for Handwriting Recognition Systems

Matthias Zimmermann, Horst Bunke
University of Bern
Institute of Informatics and Applied Mathematics
Neubrueckstrasse 10, CH-3012 Bern, Switzerland
{zimmerma,bunke}@iam.unibe.ch

Abstract

This paper investigates the use of three different schemes to optimize the number of states of linear left-to-right Hidden Markov Models (HMM). As the first method we describe the fixed length modeling scheme where each character model is assigned the same number of states. The second method considered is the Bakis length modeling where the number of model states is set to a given fraction of the average number of observations of the corresponding character. In the third length modeling scheme the number of model states is set to a specified quantile of the corresponding character length histogram. This method is called quantile length modeling. A comparison of the different length modeling schemes has been carried out with a handwriting recognition system using off-line images of cursively handwritten English words from the IAM database. For the fixed length modeling a recognition rate of 61% has been achieved. Using Bakis or quantile length modeling the word recognition rates could be improved to over 69%.

1 Introduction

After the successful application of Hidden Markov Models (HMM) to speech recognition and more recently to on-line handwriting recognition, HMM based recognition systems become more and more popular in off-line handwriting systems as well [7, 12, 19]. One of the advantages of the HMM framework lies in its capability to perform segmentation and recognition at the same time. This is a significant advantage over systems relying on a segmentation-followed-by-recognition scheme[1]. A further advantage of HMMs lies in the fact that individual models can be trained using global data only. A prior segmentation of the data into its model-specific parts, like characters or words, is not required. It is sufficient to provide the data and the corresponding label sequence. The training of the HMM using either the Baum-Welch or the Viterbi training algorithm will then learn the parameters of the provided models.

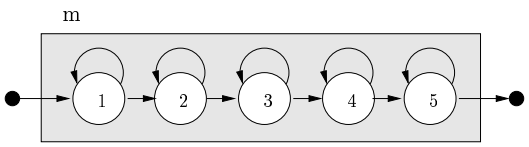

Figure 1. A character HMM

In contrast to the optimization of the parameters of an HMM, its topology (number of model states and transitions between these states) needs to be specified in advance and remains fixed during the training. The HMM framework does not provide any optimization of the topology. Although it has been suggested that the selection of the model topology should depend on the available training data, no general solution to this problem is available so far. Among the few published attempts to derive HMM topologies directly from the data the following references will be mentioned here. A model merging strategy is described in [18] which has been applied to the recogniton of spoken words. In the case of on-line handwriting recognition different strategies are proposed. The construction of a multi-branch HMM in combination with a state tying scheme is documented in [8].

The most commonly used HMM topology in speech recognition as well as in on- and off-line handwriting recognition is a simple linear left-to-right topology. Only transitions allowing to remain in a state or move to the next state are defined. An example of the left-to-right model is shown in Fig. 1. References for the use of strictly linear topologies can be found for the recognition of on-line handwritten

[1] Such systems are strongly affected by Sayre's paradox [16]: "To recognize a letter, one must know where it starts and where it ends, to isolate a letter, one must recognize it first".

Hangul characters [17], the recognition of faxed machine printed words [3], off-line handwritten numeral strings [6] and off-line cursive handwriting [2, 13]. Sometimes additional transitions are added to allow the model to skip one or more states [5, 10]. In a few cases more complex topologies are used, called multiple parallel-path HMMs [8] or multi-branch HMMs [19], which both use HMMs with left-to-right topologies as their building blocks.

In the case of a recognition system for isolated (spoken) words using left-to-right HMMs two schools of thought regarding the selection of the number of states are mentioned in [15]. The first is based on the idea to let the number of states roughly correspond to the number of sounds (phonemes) within the word. The other idea is named after R. Bakis and consists in selecting the number of model states proportionally to the average number of observations of the corresponding training samples [1]. In speech and on-line handwriting recognition systems where a sequential, time dependent signal has to be recognized, the first idea is adopted in most cases. In such applications the model states are normally related to the stationary parts of a time depending signal. In the case of speech recognition systems phones are defined by the stationary parts of the signal. For on-line handwriting recognition systems strokes are normally associated with the stationary parts of the signal. In theory one state per phone or stroke model whould suffice since a single output density function can model any stationary signal. While such models are very simple they produce an exponential state duration distribution which is normally not adequate to model phone durations or stroke lengths. In order to cope with this deficiency an explicit state duration modeling has been suggested in [4] and [9].

In the case of off-line handwriting recognition systems no time depending signal is available. A sliding window mechanism is often used to produce a sequence of observations (feature vectors). Following this approach time dependent observations are replaced by observations depending on the horizontal position of the window. As a consequence the mapping between the model states and the stationary parts of the signal is not obvious anymore and no commonly accepted idea exists what a HMM state should correspond to in the case of off-line handwriting recognition. In a black box approach characters are most frequently modeled by a single HMM which consist of a sequence of states (as shown in Fig. 1). This provides also a sequence of output density functions which can cope with complex character shapes and a flexible character duration modeling.

For the commonly used sliding window technique, character HMMs in both [14] and [12] have a fixed (globally optimized) number of states. The use of model specific numbers of states using the Bakis-model is mentioned in [5], [3] and [8]. Although [8] investigates on-line recognition of Hangul characters it was the only publication found by the authors which provided a direct comparison of the recognition performance for both a fixed and a character specific number of states.

This paper compares three different methods to optimize the number of states for the linear left-to-right topology. The first method optimizes the HMMs using a fixed number of states for all character models. The second method represents the Bakis-model where a given fraction of the average character length determines the number of states for the corresponding HMM. The third method is called quantile modeling and is motivated by the concept of minimum duration modeling where the number of states for each character HMM is defined by a specified quantile of the character length histogram. The investigation of the different length modeling methods is carried out using an HMM word recognition system for off-line cursive handwriting. The system corresponds to the recognition system published in [12] which has been adapted to the recognition of isolated words.

The remaining parts of this paper are organized as follows. The next section describes the length modeling methods. Its first subsection provides an introduction to forced alignment and the measuring of the length of sample characters. The following three subsections explain the different modeling methods in detail. Sec. 3 documents the experimental results, while Sec. 4 draws some conclusions and points out possible extensions of the presented work.

2 Length Modeling

2.1 Forced Alignment

If the Viterby decoder of an HMM based recognition system is used in forced alignment mode, the trained HMMs, a sequence of observations (feature vectors), and its transcription (in the form of a sequence of models) are provided to the decoder. Based on this information it is then the task of the decoder to find an optimal mapping between the observations and the model states representing the transcription. Before we extract the feature vector sequence $X = (X_1, X_2, \ldots, X_i, \ldots, X_n)$ of a word image, it is normalized in order to reduce the variability present in words written by different writers. An example of the text normalization and the extraction of the corresponding feature vector (observation) sequence $X = (X_1, X_2, \ldots, X_i, \ldots, X_n)$ is provided in Fig. 2. Using a sliding window technique a single feature vector X_i is extracted for each column of the normalized text line image. Consequently the horizontal coordinates of the image columns are represented by the indices of the corresponding feature vectors. This fact directly supports the estimation of character widths as we will see later.

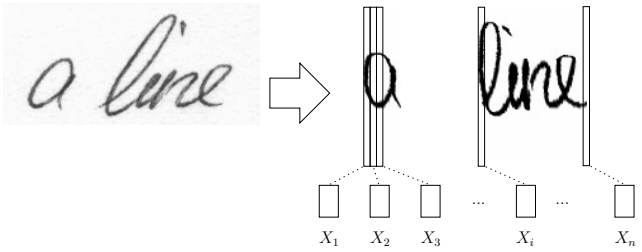

Figure 2. Text normalization and extraction of a feature vector sequence

Figure 3. Character segmentation using forced alignment

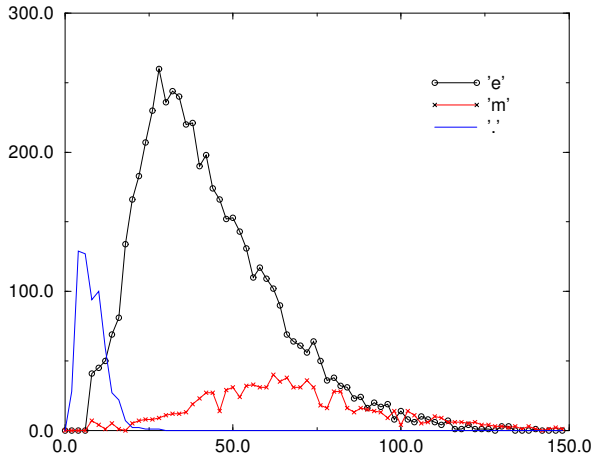

Figure 4. Length histograms for some characters

The HMM state sequence $Q = (q_1, q_2, \ldots, q_j, \ldots, q_m)$ used in the forced alignment mode can be generated by the concatenation of the character HMMs according to the transcription of the text line. Providing both the feature vector sequence X and the state sequence Q to the Viterby decoder, an optimal alignment $\hat{A}$ can be found by dynamic programming as follows.

$$\hat{A} = \arg\max_{A} P(X, A|Q)$$

$\hat{A} = ((X_1, q_1), \ldots (X_i, q_j) \ldots (X_n, q_m))$ represents therefore the most likely assignment of each X_k to a state q_l. Because each feature vector X_k has to be absorbed by exactly one state and each state q_l consumes at least one feature vector it follows that each A considered in the computation of $\hat{A}$ has to start with the assignment (X_1, q_1) and must end with (X_n, q_m). All elements of the path between these two positions have to respect the restrictions imposed by the linear left-to-right topology of the character HMMs involved in the state sequence Q.

Once an optimal alignment $\hat{A}$ is found the position and lengths of the individual models (in our case characters) can easily be determined using the feature vector indices[2]. Fig. 3 gives two examples of the character segmentation using forced alignment.

Using the estimated character lengths of 9'929 word images from the IAM database the obtained length histograms for the three characters 'e' 'm' and '.' are plotted in Fig. 4 where the x-axis corresponds to the character width in pixels and the y-axis to the number of samples. The different heights of the three curves reflect the fact that roughly 5000 samples of 'e', 1000 samples of 'm' and 600 samples of '.' were present in the set of the 9'929 words.

[2]The smallest index of all feature vectors assigned to the first state of a given character HMM directly represents the horizontal coordinate for this character. The individual lengths can then be estimated by the distance between the beginnings of two consecutive characters.

2.2 Fixed Length Modeling

For the fixed length modeling no assumptions are made at all. Instead of trying to predict a good length for the individual character HMM all models are assigned the same length (number of states).

The optimal number of states can then be found by measuring the recognition rate of the word recognizer for each possible number of states. To speed up the method, the search range can be constrained by the use of empirical knowledge.

2.3 Bakis Length Modeling

For the Bakis modeling method we assume that the length of the HMM should depend on the available training data for the individual models in the following way. For each HMM the length (number of states) is set to a fraction of the average number of observations (feature vectors) of the corresponding samples in the training data.

The optimal number of states per model can then be found by measuring the recognition rate of the word recognizer for different fractions of the average character lengths.

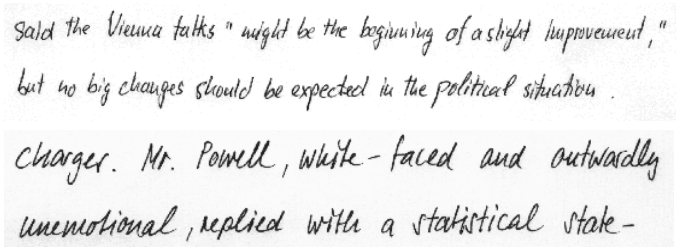

Figure 5. Some text lines from the IAM database

To speed up the search for the optimal fraction it can be considered that for the left-to-right topology a model will only accept samples containing at least as many feature vectors as its number of states. Consequently a fraction of 1.0 would reject already half of the samples. Therefore the optimal fraction can be found between 0.0 and a number significantly smaller that 1.0.

2.4 Quantile Length Modeling

The quantile length modeling method can be seen as a statistical variant of the minimum duration modeling where each HMM only accepts samples which are at least as long as the shortest samples observed in the training data. In this method the length (number of states) of each HMM is set to a specified quantile of the corresponding character length histogram[3].

3 Experimental Results

This section presents the experiments which have been carried out in order to compare the three different length modeling methods. The data used for both training and testing the different systems comes from the IAM database which was first described in [11]. Some example textlines from the IAM database are shown in Fig. 5. The words used for all experiments have been automatically extracted from the form images in the database using the method described in [20].

For the word recognition system used in the following experiments the sytem described in [12] has been adapted for the recognition of isolated words[4]. The training and testing of all systems has been carried out on a total amount of 10'929 isolated words. As test set, a random selection of 1'000 words have been put aside. The lexicon used in the experiments contained 2'296 words and was defined by the set of words occuring in the 10'929 isolated word images.

States	Size	Recognition rate
8	584	57.2%
10	730	58.9%
12	876	59.7%
14	1022	60.7%
16	1168	61.0%
18	1314	58.6%
20	1460	56.7%
22	1606	53.7%
24	1752	50.9%

Table 1. Recognition rates using a fixed number of states

Frac.	Size	R. rate
0.20	702	60.5%
0.30	1049	67.3%
0.36	1258	68.3%
0.38	1331	69.0%
0.40	1398	69.2%
0.42	1465	68.8%
0.44	1538	68.6%
0.50	1752	66.0%
0.60	2099	59.8%

Quant.	Size	R. rate
0.00	1207	56.6%
0.01	1323	65.3%
0.02	1462	69.0%
0.03	1568	69.1%
0.04	1656	69.6%
0.05	1721	68.8%
0.10	2014	65.5%
0.20	2366	57.8%
0.50	3262	29.6%

Table 2. Recognition rates for the Bakis (left) and the quantile method (right)

Each of the presented length modeling method can be optimized using different settings of a single parameter. The resulting number of model states and the corresponding recognition rates for each method are summarized in Tab. 1 and 2.

Tab. 1 documents the resulting recognition rates for different (fixed) number of states. On the left side of Tab. 2 the recognition rates for the Bakis length modeling method using different fractions of average sample lengths (column Frac.) are given. The results for the quantile length modeling method using the indicated quantiles (column Quant.) are presented on the right side.

To explain the significant improvement of both the Bakis and the quantile length modeling methods over the fixed length modeling approach some additional experiments have been carried out.

[3] The character length histogram is computed by the estimation of the number of observations (feature vectors) for each corresponding sample in the training data.

[4] The adaption of the system was trivial. It consisted in disabling the language model which was designed to recognize a sequence of words. After this step the resulting system has been trained and tested using isolated words only. In each case all HMMs were modeled with a linear left-to-right topology using a single Gaussian as continuous output function in each state. Four iterations of the Baum-Welch alogrithm have been used for the training of all HMMs.

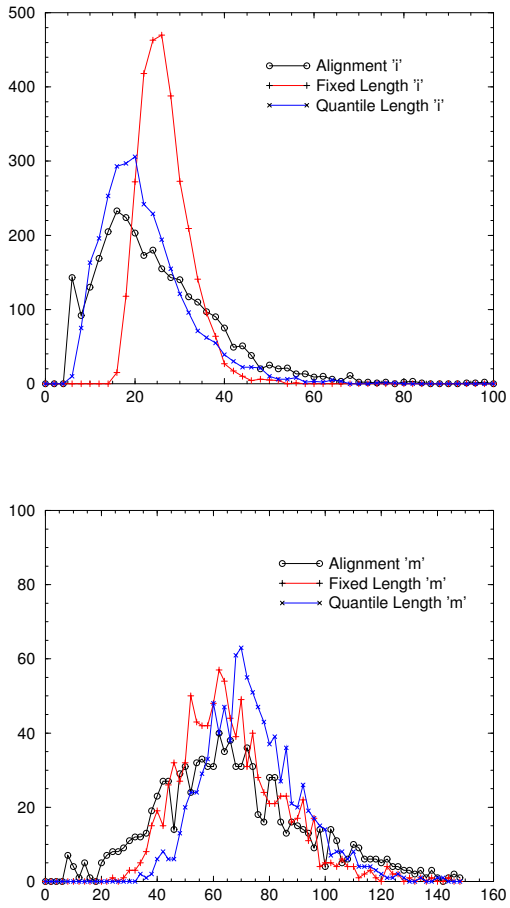

Figure 6. Comparison of estimated and generated character lengths for the characters 'i' and 'm'

For the comparison of the different length modeling approaches some properties of the fixed length modeling (using 16 states) and the quantile modeling (using the 2% quantile) have been selected. After training of the HMMs, the models for the characters 'i' and 'm' have been used to generate length histograms for the corresponding characters[5] where the x-axis correspond to the character width in pixels and the y-axis to the number of samples.

In Fig. 6 the resulting length distributions for the fixed length modeling and the quantile modeling method are compared with the 'true' length histograms which were estimated using forced alignment as described in Sec. 2.1.

[5]This has been done using the stochastic Monte Carlo method to simulate runs through the finite state automata defined by the HMM transition probabilities.

Fixed	Bakis	Quantile	T. Bakis	T. Quantile
16	0.4	0.04	16/0.4	16/0.04
1168	1398	1656	1088	1092
61.0%	69.2%	69.6%	68.1%	67.9%

Table 3. Comparison of the different length modeling schemes

It can be observed that especially for short characters (represented by 'i') the fixed length modeling approach produces length distributions which do not adequatly model the estimated length histograms. In case of the character 'm' both the fixed length and the quantile length modeling produce similar length distributions with a peak which is roughly at the position of the peak of the estimated length histograms.

Based on this observation two additional experiments were carried out to verify the importance of the correct modeling of short characters. Both the Bakis and the quantile length modeling scheme were combined with the fixed length modeling in the following way. After the length calculation of each character using either the Bakis or the quantile method, characters with more than 16 states (the optimal length obtained using the fixed length modeling scheme) have been truncated to a maximal length of 16 states.

Both the truncated Bakis and the truncated quantile methods did not only produce higher word recognition rates than the system using fixed length modeling but also resulted in significantly smaller HMM systems. This is shown in Tab. 3 which summarizes the achieved recognition rates (third row) of the different length modeling schemes. The results for the truncated Bakis method are reported in column T. Bakis and the results for the truncated quantile method in column T. Quantile. In the first row the corresponding parameter settings are provided, the second row lists the number of resulting model states and the last row the corresponding recognition rates.

4 Conclusion and Future Work

In this paper three different length modeling schemes to optimize the number of states in left-to-right HMMs were presented. The investigated methods included the frequently used fixed length modeling and the Bakis length modeling schemes. As a third method the authors propose the quantile length modeling which is based on character length histograms and motivated by the minimum duration modeling idea.

Using an HMM based word recognition system for offline handwriting recognition it could be shown that the

quantile length modeling is comparable with the Bakis length modeling, although more states per HMM were produced with the quantile length modeling in average. Both the Bakis and the quantile length modeling produced significantly higher recognition rates than the fixed length modeling approach. Using a combination of length modeling schemes it could also be demonstrated that it is possible to build an HMM based word recognition system which is significantly smaller (in terms of model states) and still achieves higher word recognition rates than the recognition system constructed with fixed length modeling only.

Future work should investigate the method presented in [10] which optimizes both the number of states per HMM and the number of Gaussian mixtures per model state within a single framework.

References

[1] R. Bakis. Continuous speech word recognition via centisecond acoustic states. In *Proc. of ASA Meeting*, Washington DC, USA, Apr. 1976.

[2] H. Bunke, M. Roth, and E. Schukat-Talamazzini. Off-line handwriting recognition using hidden Markov models. *Pattern Recognition*, 55(1):75–89, 1995.

[3] A. J. Elms, S. Procter, and J. Illingworth. The advantage of using an HMM-based approach for faxed word recognition. *Int. Journal on Document Analysis and Recognition*, 1(1):18–36, Feb. 1998.

[4] J. Ferguson. Variable duration models for speech. In *Proc. of Symposium on Application of Hidden Markov Models to Text and Speech*, pages 143–179, Oct. 1980.

[5] D. Guillevic and C. Suen. HMM word recognition engine. In *Fourth Int. Conference on Document Analysis and Recognition, Ulm, Germany*, volume 2, pages 544–547. IEEE, IEEE Computer Society Press, 1997.

[6] A. S. B. Jr, R. Sabourin, F. Bortolozzi, and C. Y. Suen. A two-stage HMM-based system for reconizing handwritten numeral strings. In *6th Int. Conference on Document Analysis and Recognition, Seattle WA, USA*, pages 396–400, 2001.

[7] A. Kundu. Handwritten word recognition using hidden Markov model. In H. Bunke and P. Wang, editors, *Handbook of Character Recognition and Document Image Analysis*, chapter 6, pages 157–182. World Scientific, 1997.

[8] J. J. Lee, J. Kim, and J. H. Kim. Data-driven design of HMM topology for online handwriting recognition. In H. Bunke and T. Caelli, editors, *Hidden Markov Models: Applications in Computer Vision*, volume 45 of *Machine Perception and Artificial Intelligence*, pages 107–121. World Scientific, 2001.

[9] S. Levinson. Continuously variable duration hidden markov models for automatic speech recognition. *Computer Speech and Language*, 1:29–45, Mar. 1986.

[10] D. Li, A. Biem, and J. Subrahmonia. HMM topology optimization for handwriting recognition. In *26th Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2001.

[11] U.-V. Marti and H. Bunke. A full English sentence database for off-line handwriting recognition. In *5th Int. Conference on Document Analysis and Recognition 99, Bangalore, India*, pages 705–708, 1999.

[12] U.-V. Marti and H. Bunke. Handwritten sentence recognition. In *15th Int. Conference on Pattern Recognition*, volume 3, pages 467–470, Barcelona, Spain, 2000.

[13] U.-V. Marti and H. Bunke. Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 15:65–90, 2001.

[14] M. Mohamed and P. Gader. Handwritten word recognition using segmentation-free hidden Markov modeling and segmentation-based dynamic programming techniques. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(5):548–554, 1996.

[15] L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[16] K. M. Sayre. Machine recognition of handwritten words: A project report. *Pattern Recognition*, 5(3):213–228, 1973.

[17] B.-K. Sin and J. H. Kim. Ligature modeling for online cursive script recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(6):623–633, 1997.

[18] A. Stolke and S. Omohundro. Best-first model merging for hidden Markov model induction. Technical Report TR-94-003, International Computer Science Institute, Berkeley CA, USA, 1994.

[19] W. Wang, A. Brakensiek, A. Kosmala, and G. Rigoll. Multi-branch and two-pass HMM modeling approaches for off-line cursive handwriting recognition. In *6th int. Conference on Document Analysis and Recognition, Seattle WA, USA*, pages 231–235, 2001.

[20] M. Zimmermann and H. Bunke. Automatic segmentation of the IAM off-line handwritten English text database. In *16th Int. Conference on Pattern Recognition*, volume 4, pages 35–39, Quebec, Canada, Aug. 2002.