

Optimizing the number of states for HMM-based on-line handwritten Whiteboard recognition

Jürgen Geiger, Joachim Schenk, Frank Wallhoff and Gerhard Rigoll
Institute for Human-Machine Communication
Technische Universität München
80290 Munich, Germany
{geiger, schenk, wallhoff, rigoll}@mmk.ei.tum.de

Abstract

In this paper, we present a novel way to determine the number of states in Hidden-Markov-Models for on-line handwriting recognition. This method extends the Bakis length modeling method which has successfully been applied to off-line handwriting recognition. We propose a modification to the Bakis method and present a technique to improve the topology with a small number of iterations. Furthermore, we investigate the influence of state tying.

In an experimental section, we show that our improved system outperforms a system with Bakis length modeling by 1.5 % relative and with fixed length modeling by 5.1 % relative on the IAM-On-DB-t1 benchmark.

1. Introduction

Adopted from automatic speech recognition (ASR), Hidden-Markov-Models (HMMs, [10]) have become quite popular for on-line handwriting recognition [9]. More recently, HMMs have also been introduced for on-line handwritten whiteboard note recognition [7], a special sub-domain of handwriting recognition.

One major challenge of HMMs is that the topology (i. e. the number of states and the transitions between these states) has to be determined prior to the training and remains fixed during the training phase. Training with the EM algorithm optimizes the parameters of the HMMs while the topology remains untouched. It is therefore essential to specify a good topology in advance.

An easy way to determine the number of states is to use the same number of states for each character model. This is referred to as fixed length modeling [14]. The

effort is kept at a minimum, but this is also reflected in the experimental results, which are mediocre. Since different characters and symbols have different lengths (with regard to sample points), it is advantageous to assign a unique number of states to each of the character models depending on the characteristics of the symbol. This is realized in the Bakis length modeling method [14], where the number of states of a symbol is set to a fraction of the average length of the symbol, where the length of a symbol is the average number of sample points contained in the realisations of this symbol. However, using this method, there is only a simple dependency between the length of a symbol and the number of states of the corresponding character model. A weakness of the Bakis length modeling method is that it can lead to some character models having extreme large number of states, which, in turn, leads to weak recognition performance for these models.

In [5], a data-driven method was used to optimise HMM topology for hangul handwriting recognition. Other simple methods to determine the number of HMM states were used in [4] and [8].

The contribution of our paper is three-fold. First, we introduce a more sophisticated approach for deciding on the needed number of states, which is based on the Bakis length modeling method. Second, this approach is further enhanced by an iterative optimization, where the recognition process and the state estimation are put in a loop. As we show, our novel approach leads to both high recognition accuracy and a small overall number of states. Therefore, we finally perform state tying with our systems, which leads to a further improvement in case of the enhanced length modeling, but leaves our baseline system unaffected.

The next section gives a short overview on the implemented recognition system. Sec. 3 summarizes fixed length modeling and the Bakis length modeling method.

In Sec. 4, we present our improved way to choose the number of HMM states. Experiments are conducted in Sec. 5. Finally we draw conclusions and give an outlook in Sec. 6.

2. System overview

This section briefly summarizes the recognition system used for the final experiments. Further details can be found in [11]. The handwritten whiteboard data is first recorded using the EBEAM-system deriving sample points $s(t) = (x(t), y(t), p(t))^T$, where $x(t)$ denotes the x -coordinate, $y(t)$ the y -coordinate, and $p(t)$ the pressure of the pen at time instance t . Afterwards, the recorded data is heuristically segmented into lines [7] and resampled in order to achieve a space-equidistant sampling. Then, a histogram-based skew- and slant-correction is performed, and all text lines are normalized to meet a distance of “one” between the corpus and the base line using a histogram-based projection approach.

Following the preprocessing and normalization, 24 state-of-the-art *on-line* and *off-line* features are extracted. The extracted on-line features are: the pen’s “pressure”, indicating whether the pen touches the whiteboard surface (f_1); a velocity equivalent, which is computed *before* resampling (f_2); the x - and y -coordinate after resampling ($f_{3,4}$); the “writing direction”, i. e. the angle α of the strokes, coded as $\sin \alpha$ and $\cos \alpha$ ($f_{5,6}$); and the “curvature”, i. e. the difference of consecutive angles $\Delta\alpha = \alpha(t) - \alpha(t - 1)$, coded as $\sin \Delta\alpha$ and $\cos \Delta\alpha$ ($f_{7,8}$).

Besides, on-line features describing the relation between the sample point $s(t)$ to its neighbors are used: a logarithmic transformation of the “vicinity aspect” v : $\text{sign}(v) \cdot \lg(1 + |v|)$ (f_9); the “vicinity slope”, i. e. the angle φ between the line $[s(t - \tau), s(t)]$, whereby $\tau < t$ denotes the τ^{th} sample point before $s(t)$, and the bottom line, coded as $\sin \varphi$ and $\cos \varphi$ ($f_{10,11}$); and the “vicinity curliness”, the length of the trajectory normalized by $\max(|\Delta x|; |\Delta y|)$ (f_{12}). Finally the average square distance to each point in the trajectory and the line $[s(t - \tau), s(t)]$ is given (f_{13}). The off-line features are: a 3×3 “context map” to incorporate a 30×30 partition of the currently written letter’s image (f_{14-22}); and “ascenders” and “descenders” (the number of pixels above and beneath the current sample point, respectively) ($f_{23,24}$).

As the values of the features vary in different ranges, each dimension d of the feature vector is normalized to a mean of $\mu_d = 0$ and variance of $\text{var}_d = 1$.

Vector Quantisation (VQ, [3]) is used to map the continuous feature vectors to discrete symbols. Then, the handwritten data is recognized by a discrete

HMM-based recognizer: each symbol is modeled by one linear left-to-right-HMM. The number of states of the HMMs is the subject of this work. It will be examined how the number of HMM states can be optimised. Training of the HMMs is performed by the well-known EM algorithm, in the case of HMMs known as Baum-Welch-algorithm [2]. Using the Viterbi algorithm, the handwritten data is recognized and segmented [10].

3. Fixed length modeling and Bakis length modeling

3.1. Fixed Length Modeling

The easiest way to determine the number of states n for an HMM-based recognition system is to use the same number of states c for every character model s :

$$n(s) = c \quad \forall s. \quad (1)$$

One can globally optimize the number of states but cannot adopt to the complexity of individual symbols at the same time. However, since in handwriting different symbols have different lengths, this is a very suboptimal way of determining the number of HMM states. There are large differences in the average number of sample points of different symbols. Symbols with a large number of sample points can be better recognized with a model that has a large number of states. Conversely, models with a smaller number of states lead to better recognition of short symbols. Therefore, it is more beneficial to assign a unique number of states to each character, depending on the average length of the character. This is usually done using the Bakis length modeling method, for which the average length of every symbol needs to be calculated. This can be done with a forced alignment.

3.2. Bakis Length Modeling

The Bakis length modeling method [1] is based on the computation of a so-called forced alignment. The forced alignment of a line of text is the optimal mapping between the observations and the corresponding HMM states. To compute the forced alignment for a line of text, the trained models, the feature vectors $\mathbf{X} = (X_1, X_2, \dots, X_i, \dots, X_n)$ of the observations and the transcription (i.e. the sequence of models) are needed. Since we use linear left-to-right HMMs, the concatenation of models according to the transcription leads to a unique sequence of HMM states

$Q = (q_1, q_2, \dots, q_j, \dots, q_m)$. The optimal alignment between observations and model states is acquired applying the Viterbi algorithm:

$$\hat{A} = \underset{A}{\operatorname{argmax}} P(X, A|Q), \quad (2)$$

where $\hat{A} = ((X_1, q_1), \dots, (X_i, q_j), \dots, (X_n, q_m))$ represents the most likely mapping between the observations and the HMM states. The number of state occupations equals the number of sample points representing the character, and hence, its length. Applying this method to every line of text in the training set leads to the average length $\bar{\ell}(s)$ for every symbol s , which is needed for the Bakis length modeling method.

For Bakis length modeling, it is assumed that the number of HMM states should correspond to the length of the character and the Bakis length modeling method follows the statement that each character model gets its own number of model states depending on the average length of the character. The number of states n for the symbol s is set to

$$n(s) = f \cdot \bar{\ell}(s), \quad (3)$$

where $\bar{\ell}(s)$ is the average length of the character s and f is a fixed factor that has to be optimized.

With increasing values of f , both the average number of states as well as the variance in the number of states become larger. In the domain of handwritten whiteboard note recognition, a large number of states is required. To accomplish this, however, a high factor f has to be used. This leads to outliers with a very large or small number of states, compared to the fixed length modeling method. The symbols corresponding to these models show weak recognition results. The improvement in performance that can be achieved by using a unique number of states for each model is limited by this fact. This could make you think that the Bakis length modeling method is not the optimal way to choose the number of HMM states. In the next section, we present our more sophisticated approach to calculate the number of HMM states.

4. New length modeling technique

As pointed out in the previous section, in contrast to fixed length modeling, the Bakis length modeling method allows for choosing the number of HMM states for each symbol individually. However, this leads to a higher variance of the number of states, which is not the case for fixed length modeling. In this section, we enhance the Bakis method in a way such that the number

of states is chosen individually for each symbol and at the same time it is prevented that extreme outliers with very large or small numbers of states exist.

4.1. New Approach for Estimating the Number of States

In order to be able to better model the number of HMM states, we propose to add a constant additive variable in Eq. 3. The resulting formula is

$$n(s) = c + f \cdot \bar{\ell}(s), \quad (4)$$

where c is an additive constant which has to be optimized heuristically by experiments. With this formula, the advantages of fixed length modeling and Bakis length modeling are combined: each model gets its own number of states but, due to the constant variable c , no models with too few states are generated. Also, a smaller factor f can be used (compared to the Bakis method) and thus, less models with extreme high number of states exist: the variance in the number of states is reduced. Our formula mathematically contains fixed length modeling and the Bakis method: Setting $f = 0$ while $c > 0$ implements the fixed length modeling method (Eq. 1); when setting $c = 0$ and $f > 0$, our approach equals the Bakis method (Eq. 3).

Due to the additive constant c in Eq. 4, the factor f can be chosen to be smaller than in Eq. 3. This leads to less outlier symbols with extreme state numbers, the number of states can be modeled more appropriately.

4.2. Iterative Refinement of the Number of States

In order to obtain the number of HMM states according to Eq. 4, a forced alignment as explained in section 3.2 is needed to determine the average length of every symbol. For initialization, the forced alignment is calculated on the basis of trained models that use the fixed length modeling method to determine the number of states. Applying Eq. 4 leads to a new topology, which results in a better recognition system. With this improved system, it is possible to calculate a new, better forced alignment, which in turn leads to an even superior recognition system. This mechanism can be performed in several iterations to improve the system. This is depicted in Fig. 1.

4.3. The Influence of State Tying

When a large total number of states occurs in a recognition system, not all parameters can be trained

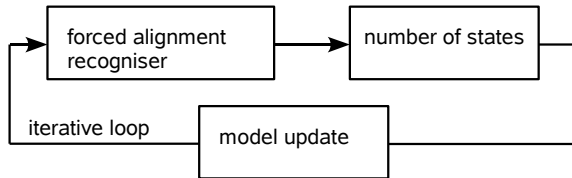


Figure 1. Iterative mechanism to improve the number of states

sufficiently due to sparse data [10]. One way to prevent this is to increase the size of the training database. Another possibility is the application of state tying. Since the number of training samples is fixed given the database, in this paper we concentrate on state tying. With state tying, the amount of states can effectively be reduced and therefore the ratio between training data and free model parameters is increased.

Similar states within the trained system are clustered together as described in [13], and in a new training cycle, the parameters of these states are jointly trained given the same training samples. With this technique, it is prevented that some states are trained by only a few samples.

We expect state tying to have a larger impact on our modified recognition system than on the system which used fixed length modeling, because a larger overall number of states is used. With fixed length modeling, a total number of 855 states is used. Our new approach uses 1565 states, which is almost twice as much. The impact of state tying will be evaluated in the next section.

5. Experiments

We conduct experiments on the IAM-onDB-t1 benchmark of the IAM-OnDB, a database containing handwritten whiteboard notes, [6].

In the first experiment (*Exp. 1*), the baseline system, the recognition system using fixed length modeling is evaluated. The highest accuracy on the test set, 64.9 %, is achieved with 15 states.

With the second experiment (*Exp. 2*), the influence of state tying on the baseline system is evaluated. The performance could be improved to 65.1 % with state tying.

The Bakis length modeling method is evaluated in the third experiment (*Exp. 3*). Factor f was optimised on the validation set. The best results could be achieved with a factor of $f = 0.5$ with 67.2 % on the test set. The system which uses the Bakis length modeling method

outperforms the system with fixed length modeling by 3.5% relative.

The performance of our new approach for the computation of the number of HMM states with the iterative mechanism is measured in the fourth experiment (*Exp. 4*). The parameters f and c of Eq. 4 were optimized on the validation set of the database and the best parameters ($f = 0.4$ and $c = 3$) were taken for evaluation on the test set. In Fig. 2 the results for different values of f and c (after the first iteration) are shown.

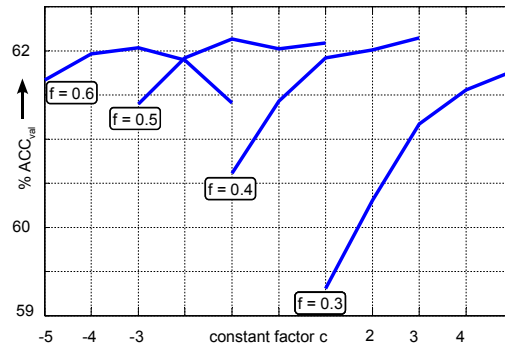


Figure 2. Results on the validation set for different values of the parameters f and c . The maximum appears at $f = 0.4$ and $c = 3$.

The best performing system of experiment *Exp. 1* was taken as an initialization to compute the first forced alignment. The best result, after iteration 4 (68.0 % on the test set), is a relative improvement of 1.2% compared to the regular Bakis length modeling.

The fifth experiment (*Exp. 5*) examines the influence of state tying on the best system of experiment *Exp. 4*. With state tying, the system performance climbs up to 68.4 %. State tying has a much higher impact on the system of experiment *Exp. 4* than it has on experiment *Exp. 1*. This is because a larger overall number of states is used in experiment *Exp. 4* (1565 states) than in *Exp. 1* (855 states).

Finally, experiment six (*Exp. 6*) investigates the influence of the initial system (with fixed length modeling) with which the forced alignment is computed. For Bakis length modeling as well as our new approach, a forced alignment is needed to get the average length of every symbol. This forced alignment is calculated based on a system with fixed length modeling, for which the optimal number of states has to be found. This leads to more effort in total. It might be possible, however,

that the result of our new approach is independent of the number of states of the initial system. In this case, the step of optimizing the initial system's state number could be omitted. To test this, three different systems with fixed length modeling were used to compute the forced alignment that is needed to get the new state numbers according to Eq. 4. See table 1 for the results. The results are similar, regardless of the num-

# states initial system	5	9	19
acc.	67.8 %	67.9 %	67.1 %

Table 1. Accuracy on character level for Exp. 6: influence of the initialization

ber of states of the initial system. Therefore time can be saved by omitting the step of searching for the best number of states for fixed length modeling before computing the forced alignment.

Table 2 summarizes the results of the experiments.

Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5
64.9 %	65.1 %	67.2 %	68.0 %	68.4 %

Table 2. Accuracy on character level on the test set for experiments Exp. 1 through Exp. 5

Shown in table 2 are the results of Exp. 1 through Exp. 5 on the test set. Experiment Exp. 5, our new approach in addition with state tying, shows the best results.

6. Conclusion

In this work, we evaluated different methods to determine the number of HMM states in a handwriting recognition system. We compared fixed length modeling and Bakis length modeling and proposed an extension to the Bakis length modeling method to decide on the number of states for linear left-to-right HMMs. We introduced a more sophisticated method to estimate the number of states for each character model individually and presented a technique to improve the number of states iteratively. Our new approach is a combination of fixed length modeling and the Bakis method.

After applying state tying to all systems, our improved system shows a relative improvement of 5.1% compared to a system with fixed length modeling and 1.5% compared to a system with regular Bakis length modeling. We also investigated the number of states

of the initial system with which the forced alignment (which is needed for the Bakis method as well as for our new approach) is computed. We showed that the results of our length modeling method are independent of the number of states of the initial system with which the forced alignment is calculated.

One possibility to improve our system could be to introduce a technique to automatically decide on the parameters of Eq. 4. For example, an optimality criterion could require the variance in the number of states to be minimised. There exist more possible ways to automate the step of finding optimal parameters.

In future work, we plan to adapt our novel technique to other domains where HMMs with a large number of states are used. In addition, we plan to investigate the influence of the VQ by applying the VQ technique presented in [12].

References

- [1] R. Bakis. Continuous Speech Recognition via Centisecond Acoustic States. In *Proc. 91st Meeting of the Acoustic Society in America*, volume 59, page S97, 1976.
- [2] L. Baum and T. Petrie. Statistical Inference for Probabilistic Functions of Finite State Markov Chains. In *Annals of Mathematical Statistics*, volume 37, pages 1554 – 1563, 1966.
- [3] R. M. Gray. Vector Quantization. In *IEEE ASSP Magazine*, pages 4 – 29, 1984.
- [4] J. HA, S. Oh, and J. Kim. Recognition of unconstrained handwritten English words with character and ligature modeling. *International journal of pattern recognition and artificial intelligence*, 9(3):535–556, 1995.
- [5] J. Lee, J. Kim, and J. Kim. Data-driven design of hmm topology for on-line handwriting recognition. In *The 7th International Workshop on Frontiers in Handwriting Recognition*, pages 107–121, 2000.
- [6] M. Liwicki and H. Bunke. IAM-OnDB – an On-Line English Sentence Database Acquired from Handwritten Text on a Whiteboard. In *Proc. of the ICDAR*, volume 2, pages 1159 – 1162, 2005.
- [7] M. Liwicki and H. Bunke. HMM-Based On-Line Recognition of Handwritten Whiteboard Notes. In *Proc. of the IWFHR*, pages 595 – 599, 2006.
- [8] K. Nathan, H. Beigi, J. Subrahmonia, G. Clary, and H. Maruyama. Real-time on-line unconstrained handwriting recognition using statistical methods. In *Proc. of the ICASSP*, volume 4, pages 2619–2619, 1995.
- [9] R. Plamondon and S. Srihari. On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey. In *IEEE Trans. Pattern Anal. Mach. Intell.*, volume 22, pages 63 – 84, 2000.
- [10] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proc. of the IEEE*, volume 77, pages 257 – 285, 1989.

- [11] J. Schenk, J. Lenz, and G. Rigoll. Novel Script Line Identification Method for Script Normalization and Feature Extraction in On-Line Handwritten Whiteboard Note Recognition. In *Pattern Recognition Journal*, volume 42, pages 3383 – 3393, 2009.
- [12] J. Schenk and G. Rigoll. Neural Net Vector Quantizers for Discrete HMM-Based On-Line Handwritten Whiteboard-Recognition. In *Proc. of the ICPR*, 2008.
- [13] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. *The HTK Book Version 3.0*. Cambridge University Press, 2000.
- [14] M. Zimmermann and H. Bunke. Hidden Markov Model Length Optimization for Handwriting Recognition Systems. In *Eighth International Workshop on Frontiers in Handwriting Recognition*, pages 369 – 374, 2001.