PATREC 3377
6 March 2004 Disk used
ARTICLE IN PRESS
No. of Pages 9, DTD = 4.3.1

ELSEVIER
Available online at www.sciencedirect.com
SCIENCE DIRECT®
Pattern Recognition Letters xxx (2004) xxx–xxx
Pattern Recognition Letters
www.elsevier.com/locate/patrec

On the structure of hidden Markov models

K.T. Abou-Moustafa [a,b,*], M. Cheriet [b], C.Y. Suen [a]

[a] Department of Computer Science, CENPARMI, Concordia University, GM-606, 1455 de Maisonneuve W., Montréal, QC, Canada H3G 1M8
[b] LIVIA, École de Technologie Supérieure, Univ. de Quebéc, 1100 Notre-Dame W., Montréal QC, Canada H3C 1K3

Received 22 July 2003; received in revised form 8 February 2004

Abstract

This paper investigates the effect of HMM structure on the performance of HMM-based classifiers. The investigation is based on the framework of graphical models, the diffusion of credits of HMMs and empirical experiments. Although some researchers have focused on determining the number of states, this study shows that the topology has a stronger influence on increasing the performance of HMM-based classifiers than the number of states.

Keywords: HMM structure; Graphical models; Credits diffusion; Ocham's razor; K-Means clustering

1. Introduction

Hidden Markov models (HMMs) (Baum and Petrie, 1966) are a class of stochastic processes that is capable of modeling time-series data. They belong to a larger class of models known as generative models. Though generative models are tools for data modeling, in the literature, HMMs were used in many classification problems such as speech recognition (Rabiner, 1989; Baker, 1975), handwritten word recognition (El-Yacoubi et al., 1999), object recognition (Cai and Liu, 2001), gesture recognition (Kim and Chien, 2001), bioinformatics (Bladi and Brunak, 1998) and modeling of biological sequences (Karplus et al., 1997).

Designing an HMM for data modeling to be a part of an HMM-based classifier means determining the structure (the number of states, and the topology) of the model. The topology in this context is meant to be the connections (transitions) between the states. The structure affects the modeling capability considerably and consequently the performance of the classifier. An estimation of the weight of each factor on the performance can point out to the main factor affecting the performance and consequently can lead to an improvement in the selection of values of this factor. Although some researchers focused on the problem of number of states and the topology, the goal of this paper is to investigate the effect of the number of states and the topology, each separately, on the performance of HMM-based classi-

* Corresponding author. Tel.: +1-514-848-7953; fax: +1-514-848-4522.
E-mail address: k_aboumo@cenparmi.concordia.ca (K.T. Abou-Moustafa).

0167-8655/$ - see front matter © 2004 Published by Elsevier B.V.
doi:10.1016/j.patrec.2004.02.005

fiers. Our research shows that the topology can improve the modeling capability greatly.

The investigation is based on (1) linking the theoretical results from model selection for graphical models with the diffusion of credits in Markovian models and, (2) supporting the results with empirical experiments for the recognition of unconstrained handwritten digits. The experiments compared the performance obtained from classifiers with different structures.

It is worth calling the readers' attention to two issues regarding this work. First, though HMMs are usually trained with time-series data with a long signal duration, such as the applications mentioned above, the experiments used isolated handwritten digits that have a short signal duration. The reason for that is to treat the HMM-based classifier like other classifiers such as multi layer perceptrons (MLPs) and support vector machines (SVMs) without any constraints on the data. Second, the goal of the paper is not to introduce a new state-of-the-art recognition result on the MNIST database using HMMs, but rather, to compare the performance of HMM-based classifiers under different structure conditions. It is well known from the literature that classifiers such as SVMs and MLPs can achieve very high recognition results (Dong, 2003; Simard et al., 2003; Liu et al., 2003) on this database.

For complete references on HMMs, the reader is required to read (Rabiner, 1989; Bengio, 1999). The paper uses the basic compact notation of HMMs defined as $\lambda = (A, B, \pi)$ where λ is the hidden Markov model, A is the transition probability matrix, B is the observation probability matrix and π is the initial state probability.

The rest of the paper is organized as follows: Sections 2 reviews related work in the literature and HMMs. Section 3 discusses the effect of the structure on the modeling capability of HMMs. Section 4 describes the experiments and results, and finally Section 5 concludes the paper.

2. Related work

A work directly related to this investigation is the problem of optimizing HMM structure in two forms; (1) application dependent methods, and (2) application independent methods. Application dependent methods use a priori knowledge from the application domain such as (El-Yacoubi et al., 1999), where they used information extracted from a character segmentation process to build a special HMM structure (number of states and the topology), (De Britto et al., 2001) modified the left-to-right model to enhance the performance of his proposed framework for numeral strings and (Lee et al., 2001) fixed the topology to be a left-to-right one and determined the number of states by reflecting the structure of a target pattern. The major drawback of these methods is that they are designed for specific applications and cannot be generalized to others.

On the other hand, application independent methods, although are more promising, yet they are not popularized. These methods include the work of (Stolcke and Omuhundro, 1992) and (Brants, 1996) where they proposed an incremental learning for the structure based on state merging and splitting, i.e., the structure is changed as new evidence is added to the model; (Lien, 1998) proposed a general method to determine the number of states and the connections between states in discrete left-to-right HMMs. Recently, Bicego et al. focused only on determining the number of states using probabilistic bisimulation (Bicego et al., 2001) and sequential pruning using Bayesian information criterion (BIC) (Bicego et al., 2003). Model selection approaches were also investigated for this purpose and recently (Biem, 2003) proposed a discriminative information criterion (DIC) framework and used it to optimize the HMM structure.

Different approaches for structure optimization can also be found. (Lyngso et al., 1999) focused on comparing HMMs in terms of state emission probabilities, (Bahlman et al., 2001) used Bayesian estimates of HMM states as a criterion for selecting HMMs and (Balasubramanian, 1993) selected HMMs based on equal probabilities of observation sequences only. By examining the above literature, and except for the work Stolcke and Brants, most of these methods use the left-to-right topology and the optimization targets only the number of states and the number of mixtures in cases of continuous HMMs. However, according to this investigation, the number of states may not affect the performance after a certain limit, but it can reduce the computational time for training and testing, while the topology can considerably affect the performance of HMMs.

3. HMM structure

In many applications that use HMMs, the number of states is manually predetermined prior to training. The connections between states, (topology) is determined by setting non-zero probabilities in the A matrix prior training. During training, the EM (Baum–Welch) algorithm improves the estimates of these probabilities from the data. Note that the EM algorithm cannot set 0 or 1 (can approach 0 or 1) probabilities in the A matrix, therefore it cannot be seen as an algorithm that learns the topology. In the following, we investigate the effect of the topology on the performance of HMM-based classifiers through two different perspectives: (1) using the graphical models framework and, (2) using the diffusion of credits while learning Markovian models.

3.1. Bayesian formulation for model selection

Determining the number of states and the topology of HMMs can be viewed as a model selection problem. The problem can be formulated as follows. Given the training set of examples Ψ and a criterion function Υ for the quality of the model on the data set Ψ, choose a model from a certain set of models, in such a way to maximize the expected value of this criterion function on new data (assumed to be sampled from the same unknown distribution from which the training data was sampled) (Bengio, 1999).

HMMs can be viewed as a special case of graphical models (Heckerman, 1996; Murphy, 2001). Model selection is one of the main problems in graphical models and much work has been introduced regarding this problem. The Bayesian approach, one of the main approaches for model selection, is a fundamental approach for model selection in graphical models. Following this approach means encoding the uncertainty about the structure of the HMM by using a discrete variable whose states correspond to the possible HMM structure hypotheses S^{h} and assessing it the a priori density $P(S^{\mathrm{h}})$. Given the training example set Ψ for the model λ and augmenting the model parameters A, B, π in a single parameter vector θ, the problem would be computing the posterior distribution for the HMM structures. This can be formulated as follows using Bayes theorem:

$$P(S^{\mathrm{h}}|\Psi) = \frac{P(\Psi|S^{\mathrm{h}})P(S^{\mathrm{h}})}{P(\Psi)} \tag{1}$$

where $P(\Psi)$ is a constant that does not depend on the network structure.

The maximum likelihood structure would be the complete graph (Murphy, 2001), i.e., the full ergodic model, since this has the greatest number of parameters, and hence can achieve the highest likelihood. On the other hand this increases the model's complexity and will let the model overfit the training data resulting in a poor generalization. In fact, the marginal likelihood in 1 plays an important role to prevent this overfit. From the definition of the marginal likelihood:

$$P(\Psi|S^{\mathrm{h}}) = \int P(\Psi|S^{\mathrm{h}}, \theta)P(\theta|S^{\mathrm{h}})\mathrm{d}\theta \tag{2}$$

it automatically penalizes more complex structures since they have more parameters and hence cannot give as much probability mass to the region of space where the data actually lies. In other words, a complex model is less believable and hence less likely to be selected. This phenomenon is known as Ocham's razor (Murphy, 2001) which favors simple models over complex ones. It can be seen that though the number of states may be fixed, the topology can affect the modeling capability in a serious way.

3.2. Diffusion of credits in Markovian models

The work in (Bengio and Frasconi, 1995) investigated the problem of diffusion in homogeneous and non-homogeneous HMMs and its effect on learning long term dependencies. Training HMMs requires propagating forward and backward probabilities and taking products of the transition matrix. Therefore, two types of diffusion exist, diffusion of influence in the forward path and diffusion of credit in the backward phase of training. The paper (Bengio and Frasconi, 1995) studied under which conditions these products of matrices will converge to a lower rank, thus harming learning long term dependencies. The difficulty of learning was measured by using the Dobrushin's ergodicity coefficient (Senta, 1986) defined as follows:

$$\tau(A) = \frac{1}{2} \sup_{i,j} \sum_{k} |a_{ik} - a_{jk}| \tag{3}$$

where $A = \{a_{ij}\}$ is the transition probability matrix. It was shown that in all cases, while training HMMs, the ergodicity coefficient will converge to 0 indicating a greater difficulty in learning, but the rate of convergence depends on the topology. Fig. 1 (Bengio and Frasconi, 1995) shows the convergence of four HMMs with the same number of states but with different topologies. It can be seen that the full ergodic model has the fastest convergence rate and that simpler models are slower. The final conclusion is that in order to avoid any kind of diffusion, most transition probabilities should be deterministic (0 or 1 probability). The result coincides with the Ocham's razor result obtained from the previous section and both prefer simple topologies over complicated ones.

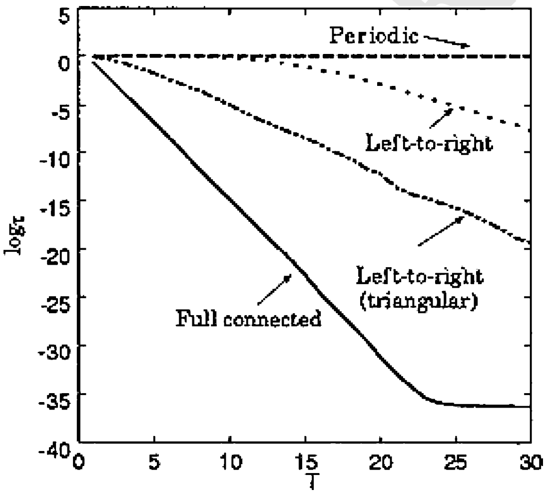

Fig. 1. Convergence of Dobrushin's coefficient for four different topologies.

4. Experiments

We were interested in investigating experimentally how the number of states and the topology can affect the performance of an HMM-based classifier. Two types of experiments were carried out, one to study the effect of number of states on the performance, and the other to study the effect of the topology on the performance.

4.1. The dataset and feature extraction

The dataset used in the experiments consists of images of unconstrained handwritten digits from the MNIST database (LeCun, 1998) which has a training set of 60,000 samples and a test set of 10,000 samples from approximately 250 writers. The digits are cropped and scaled to be contained in a 20×20 pixels images. The gray level values of the images were normalized to be from 0 to 1. The time series data were extracted from the digits using the sliding window technique (Cornell, 1996) with a width of 3 pixels, height equals the image height and an overlap of 2 pixels. A feature vector is extracted from each window by computing the average gray level value in each row of the window, i.e., the sum of gray level pixels in each row divided by the window width. This resulted in an observation sequence length of 18 vectors from each image.

4.2. HMM density type, initialization and codebook size

The experiments were conducted using discrete HMM (DHMM)-based classifiers where each consisted of 10 DHMMs. The number of states for each model was determined according to the goal of the experiment. Two topologies were used in the experiments, the left-to-right with self-state transitions (no jumps), and the ergodic topology. For the code book, the vector quantization (Gray, 1984) algorithm was used to construct seven different code books $(16, 32, \ldots, 1024)$. The initial parameters for B in all experiments were set using a uniform distribution. In our original investigation, all the experiments were conducted using the seven code books and with several initializations for the A matrix as will be shown later. However, due to the following reasons: (1) space limitation, (2) avoid redundancy, and (3) similarity of results and conclusions, we selected the clearest of these experiments for illustration.

4.3. Studying the effect of number of states

In studying the effect of number of states, two experiments were conducted. The first experiment used HMMs with a left-to-right topology and all models had an equal number of states. The experiment studied the relation between the performance and the increase in the number of states in the classifier. The second experiment studied the performance of classifiers with a varying number of states in each model. It compared the performance between models with an equal number of states and models with a varying number of states.

4.3.1. Experiment 1

This experiment was conducted using the seven code books, and for each experiment, the A matrix was initialized using three different initializations; (0.5 & 0.5, 0.7 & 0.3 and 0.9 & 0.1) for the ij and ii transitions, respectively. Fig. 2 illustrates the results for *Experiment 1* using three code books and the first initialization for the A matrix. It can be seen that increasing the number of states can increase the performance up to a certain limit, after that a saturation is reached whenever more unnecessary states are added. However, the saturation may be accompanied by a slight drop in the performance.

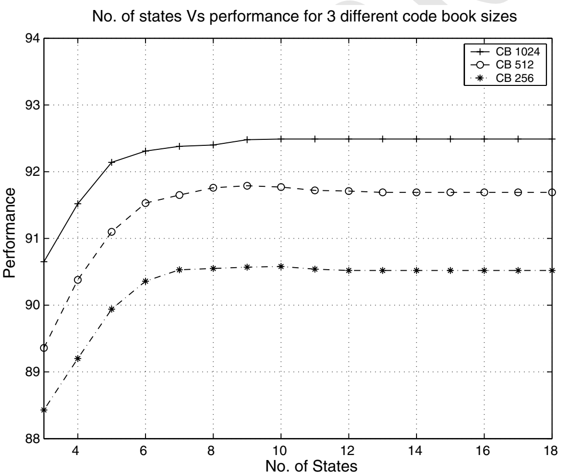

Fig. 2. The relation between performance and the number of states with different code book sizes.

The saturation may be explained as follows. The number of states N, is the number of values that the hidden variable can take and accordingly the emission of symbols change. Let the true (unknown) number of values of the hidden variable be N_0. If $N \ll N_0$ poor modeling will result and hence a classifier with poor performance. If $N \gg N_0$, additional states will introduce redundancy with no effect on the modeling capability and hence the performance is saturated. Adding more unnecessary states increases the complexity (time and computation) with no effect on the performance.

4.3.2. Experiment 2

The goal of the experiment was to measure the performance of classifiers with a different number of states in each model to see how comparable they are with classifiers having all models with an equal number of states. Two HMM classifiers were used. According to the previous experiments, the first classifier had 10 states per model, the second classifier had a different number of states in each model. Determining the number of states in each model will be described in the next subsection. As *Experiment 1*, this experiment was also conducted using the seven code books and the three different initializations. Fig. 3 illustrates the results of this experiment for three code book sizes and the first initialization. Models with an equal number of states are referred as (EQU) and models with a varying number of states are referred as (VAR).

Fig. 3 shows clearly how models with a varying number of states can achieve almost the same performance of models with an equal number of states with the advantage of a smaller number of states but paying the price of more epochs. The total number of states in the EQU models is 100, and the total for VAR models is 70 states. Achieving the same performance with a smaller number of states means a considerable reduction in complexity when it comes to large classification problems. However, as followed in the literature (El-Yacoubi et al., 1999; Augusting et al., 1998), a

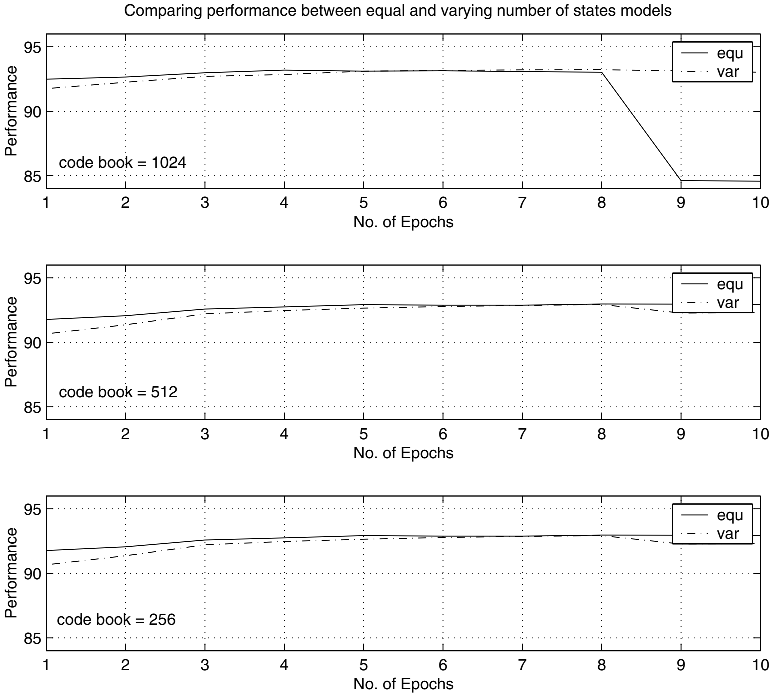

Fig. 3. Performance comparison between models with equal (EQU) and varying (VAR) number of states with different codebook sizes.

guaranteed performance with an easy design would be an HMM classifier with an equal number of states for all models. In Fig. 3, it is worth mentioning that the drop seen in the first graph is experienced in the other graphs for the EQU and VAR models but in late epochs not shown in the graphs. The reason for the drop is due to the overfit of models on the data and due to the diffusion of credits while learning.

4.3.3. Determining the number of states

As mentioned earlier, the number of states is usually fixed (manually predetermined). Exceptions are models that use automatic clustering algorithms that determine the number of states and their outputs, but this still leaves out the topology (Brants, 1996; Theodoridis and Koutroumbas, 1999). Clustering sequential data while neglecting the variations of the time factor, tends to discover the underlying structure of the data given that the number of clusters is known. To determine the number of states using clustering, we proposed the use a cluster validity index (Bezdek and Pal, 1998) to measure the goodness of different clustering configurations and then select the best number of clusters according to this cluster validity index.

In the experiments, the K-Means algorithm (Duda et al., 2001) was used to cluster the sequential data of each model. The algorithm was allowed to cluster the sequential data up to two different maximum number of clusters; (1) from three up to five clusters (first row in Table 1), and (2) from three up to nine clusters (second row in

Table 1
The number of states of each model

Model	0	1	2	3	4	5	6	7	8	9
No. of states (3–5)	5	5	5	5	4	4	4	5	3	4
No. of states (3–9)	6	5	8	8	9	6	8	8	3	9

Table 1). In order to overcome the problem of initialization of the K-means, the algorithm was run using 10 different initializations. For each clustering configuration, the DB-index (Bezdek and Pal, 1998) was used to measure the goodness of clustering. According to the DB-index measure, the number of states (clusters) in each model was determined according to the clustering configuration corresponding to the lowest value of the DB-index. Table 1 shows the number of states for each model for each clustering configuration.

4.4. Studying the effect of model topology

To study the effect of the model topology on the performance, two HMM-based classifiers were considered. Both classifiers had the same number of models and the same number of states in each model but the model topology was different in both classifiers. The first classifier had full ergodic (fully connected) models while the second had left-to-right topology as described earlier. The experiment was conducted using ten code books (previous 7 plus 3 more with size 1500, 1800, 2000), five different initializations for the A matrix; (0.5 & 0.5, 0.6 & 0.4, 0.7 & 0.3, 0.8 & 0.2 and 0.9 & 0.1) for the ij and ii transitions, respectively of the left-to-right model, and five different random initializations for the ergodic model. Fig. 4 illustrates the results obtained from this experiment on the ten code books and the first initialization of each model.

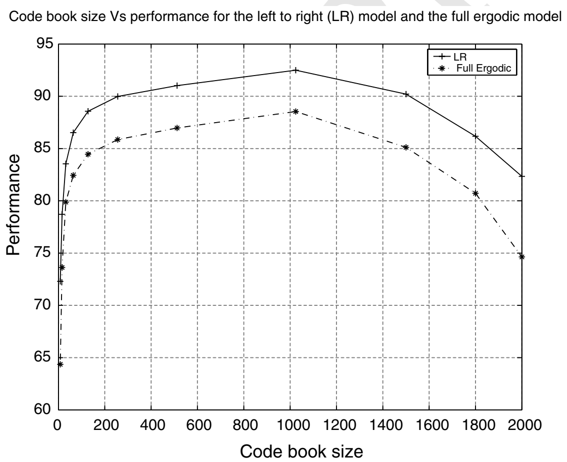

Fig. 4. Performance comparison between full ergodic and left-to-right models with different codebook sizes.

As expected, the results show that the simpler model; which is the left-to-right in that case, always outperforms the full ergodic model. The full ergodic model represents a fully connected graph and hence has the largest number of parameters. According to the Bayesian approach, the model has the highest likelihood of the data which led the model to overfit the training set and hence the poor performance on the test set. As for the diffusion of credits factor, the A matrix for the full ergodic model does not have deterministic (0 or 1 probabilities) transitions which made it difficult for the model to learn long range dependencies.

The degradation of performance in Fig. 4 is due to an accumulated effect from the vector quantization and the training of HMMs and it may be explained as follows. The vector quantization process was performed on the training set of the database and increasing the code book size led the algorithm to form smaller and finer (might be noise) clusters from the training set. Hence, the result is a well fitted code book for the training set and very sensitive to slight variations, i.e., over fitting. Next, the discrete HMMs used this sensitive but large code book for training, which implies that the HMMs were trained on very special sequences of symbols that may not occur in the test set. Consequently, the HMMs had over-fit the training set and will have a poor generalization on the test set. Hence, the fall in the two curves is due to the accumulated over-fit effect that started from the vector quantization and propagated to the HMM training.

5. Conclusion

We studied the effect of number of states and the topology on the performance of HMM-based classifiers. The Bayesian approach for model selection with the Ocham's razor showed that simpler models will have better generalization than full ergodic (fully connected) models. On the other hand, to avoid any diffusion of credits while learning HMMs, transition probabilities should be deterministic (0 or 1 probabilities). Both of these results supported with the empirical experiments show that the topology has a stronger influence than the number of states in improving the modeling capability of HMMs and hence increasing the performance of HMM-based classifiers. It can be seen from Figs. 2 and 4 that increasing the number of states from 3 to 6 increased the performance by almost 2% and changing the topology increased the performance by (4–5%). The result encourages us to design algorithms for HMMs, different than model selection techniques, that can learn the topology from the training data, i.e., set 0 or 1 transitions in the *A* matrix, especially in the absence of the a priori knowledge.

Acknowledgements

We would like to thank Incheol Kim from CENPARMI for useful discussions. Also, we would like to thank the Ministry of Education of Quebéc and NSERC of Canada for the financial support.

References

Augusting, E., Baret, O., Knerr, S., Price, D., 1998. Hidden Markov model based word recognition and its application to legal amount recognition on Frensh bank cheques. Comput. Vision Image Understand. 70 (3), 404–419.

Bahlman, C., Burkhardt, H., Ludwigs, A., 2001. Measuring HMM similarity with the Bayes probability of error and its application. In: Proc. 6th ICDAR, Seattle, Washington, USA. pp. 406–411.

Baker, J.K., 1975. The Dragon system—an overview. IEEE Trans. Acoust., Speech Signal Process 23 (11), 23–29.

Balasubramanian, V., 1993. Equivalence and reduction of hidden Markov models. MIT Artificial Intell. Lab. Tech. Report 1370.

Baum, L.E., Petrie, T., 1966. Statistical inference for probabilistic functions of finite state Markov chains. Ann. Math. Statist. 37, 1554–1563.

Bengio, Y., Frasconi, P., 1995. Diffusion of credits in Markovain model. Neural Informat. Process. System 7, 1251–1254.

Bengio, Y., 1999. Markovian models for sequential data. Neural Comput. Surveys 41 (1), 129–162.

Bezdek, J.C., Pal, N.R., 1998. Some new indexes of cluster validity. IEEE Trans. Systems Man Cybernet. Part B 28 (3), 301–315.

Bicego, M., Dovier, A., Murino, V., 2001. Designing the minimal structure hidden Markov model by bisimulation. In: Figueiredo, M., Zerubia, J., Jain, A.K. (Eds.), Energy Minimization Methods in Computer Vision and Pattern Recognition. Springer, pp. 75–90.

Bicego, M., Murino, V., Figueiredo, M., 2003. A sequential pruning strategy for the selection of the number of states in hidden Markov models. Pattern Recognition Lett. 24, 1395–1407.

Biem, A., 2003. A model selection criterion for classification: Application to HMM topology optimization. In: Proc. 17th ICDAR, Edinburgh, UK. pp. 104–108.

Bladi, P., Brunak, S., 1998. Bioinformatics, the Machine Learning Approach. MIT Press.

Brants, T., 1996. Estimating Markov model structures. In: Proc. ICLSP Philadelphia, PA.

Cai, J., Liu, Z.-Q., 2001. Hidden Markov models with spectral features for 2D shape recognition. IEEE Trans. PAMI 23 (12), 703–713.

Cornell, S., 1996. A comparison of hidden Markov model features for the recognition of cursive handwriting. Department of Computer Science, Michigan State University, Master Thesis, 1996.

De Britto, A.S., Sabourin, R., Bortolozzi, F., Suen, C.Y., 2001. An enhanced HMM topology in an LBA framework for the recognition of handwritten numeral strings. In: Internat. Conf. on Advances in Pattern Recognition. pp. 105–114.

Dong, J., 2003. Speed and accuracy: Large-scale machine learning algorithms and their applications, CENPARMI, Department of Computer Science, Concordia University, Montreal, Canada, Ph.D. Thesis, 2003.

Duda, R.O., Hart, P.E., Stork, D.G., 2001. Pattern Classification, second ed. Willey-Interscience.

El-Yacoubi, A., Gilloux, M., Sabourin, R., Suen, C.Y., 1999. An HMM based approach for off-line unconstrained handwritten word modeling and recognition. IEEE Trans. PAMI 21 (8), 752–760.

Gray, R., 1984. Vector quantization. IEEE Trans. ASSP, 4–29.

Heckerman, D., 1996. A tutorial on learning with graphical models, MSR-TR-9506, Microsoft Research.

Karplus, K., Sjolander, K., Barrett, C., Cline, M., Haussler, D., Hughey, R., Holm, L., Sander, C., 1997. Predicting protein structure using hidden Markov models. Proteins: Struct., Funct. Genet. 1 (1), 134–139.

Kim, I., Chien, S., 2001. Analysis of 3D hand trajectory gestures using stroke-based composite hidden Markov models. Appl. Intell. 15, 131–143.

LeCun, Y., 1998. The MNIST Database of Handwritten Digits, yann.lecun.com/exdb/mnist.

Lee, J., Kim, J., Kim, J.-H., 2001. Data-driven design of HMM topology for on-line handwritten recognition. Pattern Recognition 15 (1), 107–121.

Lien, J., 1998. Automatic recognition of facial expressions using hidden Markov models and estimation of expression intensity. CMU, School of Computer Science, Pittsburg, Ph.D. Thesis, 1998.

Liu, C.-L., Nakashima, K., Sako, H., Fujisawa, H., 2003. Handwritten digit recognition: Benchmarking of state-of-the-art techniques. Pattern Recognition 36, 2271–2285.

Lyngso, R., Pedersen, C., Nielsen, H., 1999. Metrics and similarity measures for hidden Markov models. In: Proc. Internat. Conf. on Intelligent Systems for Molecular Biology. pp. 178–186.

Murphy, K.P., 2001. A Introduction to Graphical Models, www.ai.mit.edu/murphyk/papers.html.

Rabiner, L.R., 1989. A tutorial on hidden Markov models and selected application in speech recognition. Proc. IEEE 77 (2), 257–286.

Senta, E., 1986. Nonnegative matrices and Markov chains. Springer, New York.

Simard, P., Steinkraus, D., Platt, J.C., 2003. Best practices for convolutional neural networks applied to visual document analysis. In: Proc. 17th ICDAR, Edinburgh, UK. pp. 962–965.

Stolcke, A., Omuhundro, S., 1992. Hidden Markov model induction by Bayesian model merging. In: Hanson, S., Cowan, J., Giles, C. (Eds.), Advances in Neural Information Processing 5. Morgan Kaufmann, pp. 11–18.

Theodoridis, S., Koutroumbas, K., 1999. Pattern Recognition. Academic Press.