Model-based clustering with Hidden Markov Models and its application to financial time-series data

Bernhard Knab[1], Alexander Schliep[2], Barthel Steckemetz[3], and Bernd Wichern[4]

[1] Bayer AG, D-51368 Leverkusen, Germany
[2] Department Computational Molecular Biology, Max-Planck-Institut für Molekulare Genetik, D-14195 Berlin, Germany
[3] Science Factory GmbH, D-50667 Köln, Germany
[4] ifb AG, D-50667 Köln, Germany

Abstract. We have developed a method to partition a set of data into clusters by use of Hidden Markov Models. Given a number of clusters, each of which is represented by one Hidden Markov Model, an iterative procedure finds the combination of cluster models and an assignment of data points to cluster models which maximizes the joint likelihood of the clustering.

To reflect the non-Markovian nature of some aspects of the data we also extend classical Hidden Markov Models to employ a non-homogeneous Markov chain, where the non-homogeneity is dependent not on the time of the observation but rather on a quantity derived from previous observations.

We present the method, a proof of convergence for the training procedure and an evaluation of the method on simulated time-series data as well as on large data sets of financial time-series from the Public Saving and Loan Banks in Germany.

1 Introduction

Grouping of data, or clustering, is a fundamental task in data analysis as well as a prerequisite step for classification of unlabeled data. Methods for clustering have been widely investigated [7] and can be coarsely categorized into two classes: distance- and model-based approaches. The former base the decision whether to group two data points on their distance, the latter assign a data point to a cluster represented by a particular statistical model based on its likelihood under the model.

For a number of reasons, model-based clustering is better suited for time-series data [14]. Usually, there is no natural distance function between time-series. A number of non-critical variances of signals — a delay, an overall slower rate, a premature cutoff — will be overly emphasized by, say, Euclidean distance. Capturing the essential *qualitative* behavior of time-series is difficult to achieve with a distance function.

Using stochastic models to represent clusters changes the question at hand from how close two given data points are to how likely one particular data point is under model. Often, the latter question is easier to answer as we will demonstrate for our particular application. Also, one can expect a larger robustness with respect to noise in the data virtue of the stochastic model. As it is straight-forward to generate artificial data given a model-based clustering, an analysis of the clustering quality based on the *predictive* performance of the inferred set of models becomes possible.

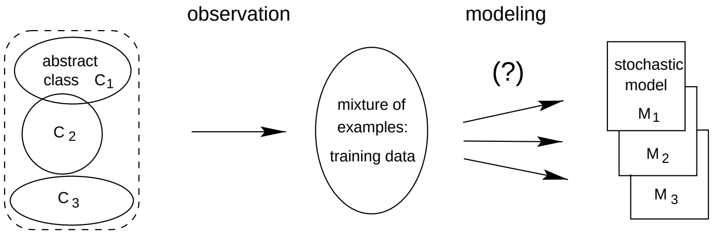

Fig. 1. The general assumption regarding the nature of the data: A mixture of realizations from unobservable abstract classes is observed. Stochastic models corresponding to those abstract classes should then be inferred.

Our approach of using Hidden Markov Models (HMMs) as clusters is motivated by the well known k-Means algorithm [7]. In [9] it has been realized that an analogon of k-means can be used for model-based clustering. This has been implemented for multi-variate Gaussians in [15]. In the k-Means algorithm the median is used to represent a cluster and a clustering is computed by an iterative application of the following steps.

1. Assign each data point to its closest median, and
2. Re-compute the median for each of the clusters.

When going over to HMMs as cluster representatives two modifications are necessary. The criterion for the re-assignment of data points to clusters is maximization of the likelihood of the data points. The re-computation of clusters is done by training the cluster models with the Baum-Welch reestimation algorithm [1,2]. From a computational point of view the main difference lies in the necessity of the nested iterative procedure.

The savings and loan bank application we considered implied contractual constraints which violated the Markovian assumption inherent in *classical* HMMs. We could account for these constraints by a model extension, which can be thought of as a HMM based on a non-homogeneous Markov chain. However, the non-homogeneity was not conditioned on the time of the observation in the time-series but rather on a function summarizing and, hence, dependent on previous observations. This extension required only minor modifications to the various relevant HMM algorithms such as Baum-Welch. The clustering with the extended model provided a powerful modeling and analysis framework which improved the quality of the modeling substantially when compared with the methods previously used.

This paper is organized as follows: After establishing notation and necessary concepts in Sec. 2 we introduce the algorithm, analyze its computational complexity and discuss implementational questions in Sec. 3. The setting of the application problem and the data used in the experimental validation is subsequently described. This motivates the following extension to non-homogeneous HMMs introduced in Sec. 5. Experimental results and a discussion conclude the paper.

2 Definitions and Notation

Hidden Markov Models (HMMs) can be viewed as probabilistic functions of a Markov chain [4,16], where each state of the chain independently can produce emissions according to so-called emission probabilities or densities. We will restrict ourselves to univariate emission probability densities. Extensions to multivariates or mixtures thereof as well as discrete emissions are routine.

Definition 1 (Hidden Markov Model). Let $O = (O_1, \ldots)$ be a sequence over an alphabet Σ. A Hidden Markov Model λ is fully determined by the following parameters:

- S_i, the states $i = 1, ..., N$,
- π_i, the probability of starting in state S_i,
- a_{ij}, the transition probability from state S_i to S_j, and
- $b_i(\omega)$, the emission probability density of a symbol $\omega \in \Sigma$ in state S_i.

The obvious stochasticity constraints on the parameters apply. A more thorough and very readable introduction to HMMs can be found in [17], respectively in one of several books [14,10,5,3]. The problem we will address can be formally defined as follows.

Definition 2 (HMM Cluster Problem). Given a set $\mathcal{O} := \{O^1, O^2, \ldots, O^n\}$ of n sequences, not necessarily of equal length, and a fixed integer $K \ll n$. Compute a partition $\mathcal{C} = (C_1, C_2, \ldots, C_K)$ of $\mathcal{O}$ and HMMs $\lambda_1, \ldots, \lambda_K$ as to maximize the objective function

$$f(\mathcal{C}) = \prod_{k=1}^{K} \prod_{O^i \in C_k} L(O^i | \lambda_k)\,. \tag{1}$$

Here, $L(O^i|\lambda_k)$ denotes the likelihood function, i.e. the probability density for generating sequence O^i by model λ_k:

$$L(O^i|\lambda_k) := P(O^i|\lambda_k). \tag{2}$$

It has been implicitely discovered before, e.g. [20], that the problem of computing a k-means clustering can be formulated as a joint likelihood maximization problem.

3 The clustering algorithm

Adapting the general outline of the k-means algorithm, we propose the following maximum likelihood algorithm to solve a HMM Cluster Problem, given K initial HMMs $\lambda_1^0, \dots, \lambda_K^0$.

1. **Iteration ($t \in \{1, 2, \dots\}$):**
   (a) Generate a new partitioning of the sequences by assigning each sequence O_i to the model k for which the likelihood $L(O_i | \lambda_k^{t-1})$ is maximal.
   (b) Calculate new parameters $\lambda_1^t, \dots, \lambda_K^t$ using the re-estimation algorithm for each model with their start parameters $\lambda_1^{t-1}, \dots, \lambda_K^{t-1}$ and their assigned sequences.
2. **Stop**, if the improvement of the objective function (1) is below a given threshold, ε, the grouping of the sequences does not change or a given iteration number is reached.

As there is a one-to-one correspondence between clusters and models we will use the terms interchangeably in the following.

Convergence

The nested iteration scheme does indeed converge to a local maximum. How to avoid the usual practical problems with local maximization is described later.

Theorem 1. *The objective function* (1) *of the maximum likelihood algorithm is non-decreasing.*

Proof: Given the partitioning $\mathcal{C}^t$ after the iteration t, the corresponding trained model parameters λ_k^t and the logarithm of objective function (1), $\log f(\mathcal{C})$. Then,

$$\begin{aligned}
\log f(\mathcal{C}^t) &= \sum_{k=1}^{K} \sum_{O^i \in C_k^t} \log L(O^i | \lambda_k^t) \\
&\leq \sum_{k=1}^{K} \sum_{O^i \in C_k^t} \max_{l=1 \dots K} \log L(O^i | \lambda_l^t) \\
&= \sum_{k=1}^{K} \sum_{O^i \in C_k^{t+1}} \log L(O^i | \lambda_k^t) \\
&\leq \sum_{k=1}^{K} \sum_{O^i \in C_k^{t+1}} \log L(O^i | \lambda_k^{t+1}) \\
&= \log f(\mathcal{C}^{t+1}) \, ,
\end{aligned}$$

and thus $f(\mathcal{C}^t) \leq f(\mathcal{C}^{t+1})$. The last inequality above follows as the likelihood of a single model is non-decreasing in re-estimation [17], i.e. $L(O|\lambda_k^t) \leq L(O|\lambda_k^{t+1})$.

□

Complexity

Given n sequences with maximal length T and K models with at most N states (and univariate density functions), the computational complexity, in slight abuse of the usual notation, of the re-estimation of the K models in step 1(b) is $O(I_r\, n\, T\, N^2)$, where I_r is an upper bound, typically dependent on the size of the input, for the number of iterations in the applied re-estimation algorithm. Assigning all sequences to the model with the highest likelihood needs $O(n\, K\, T\, N^2)$ steps. Therefore, the complexity of the complete clustering algorithm is $O(I_c\, n\, T\, N^2\, (K + I_r))$, where I_c is another bounding constant for the outer iteration. The bounds I_r and I_c seem both artificial and unjustly taken constant in the input. However, in practical HMM training over-fitting is often a more pressing problem than getting stuck in optima which are local but not global. This is routinely dealt with by limiting the number of Baum-Welch steps, which supports taking at least I_r constant.

Implementation

The relevant data structures and algorithms are freely available in a C-library, the GHMM [13], licensed under the Library Gnu General Public License (LGPL). The software has been compiled and used on a wide range of hardware as well as operation systems. It is currently in use at a number of different institutions for other problem domains.

Initialization

A suitable model topology, i.e. the number of states and the allowed transitions (the non-zero transition probabilities), and the number of initial models should be motivated by the application. Note that the topology remains unchanged during the training process.

Since the clustering algorithm will only converge to a local maximum the choice of the model's start parameters will have an impact on the maximum computed. The simplest approach is to set all parameter to random values subject to stochasticity constraints. This can easily lead to an very uneven, w.r.t. cluster size, assignment of sequences to models, as some random model might have near zero probabilities of producing any sequences in the set at all. Alternatively, one can initially train one HMM with all sequences, and subsequently use K copies of that model as the input for the clustering, after adding small random perturbations to the parameters of the K copies individually. Training can be thought of in this case as seeing a divergence of clusters. In any case, one has to pay attention that in the first iteration step each sequence can be build from at least one model — i.e., the likelihood of the set of sequences may not be zero. If there is only a very limited amount of training data available, pseudo-counts or, more rigorously, Dirichlet priors [19] as background distributions in a Bayesian setting can be employed to dispatch with this over-fitting problem effectively.

4 Application to loan bank data

To evaluate the proposed clustering method on real data, we use financial time-series data obtained from the public saving and loan banks in Germany for an ongoing co-operative research project [11]. The fundamental concept behind saving and loan banks is to combine a period of saving money, usually until some threshold D has been reached, which is the prerequisite for taking out a loan, which then has to be re-payed in fixed installments. Contractual details vary widely, but manual inspection suggested a number of prototypical contract histories making clustering appear feasible.

Each of the data points corresponds to an individual saving and loan contract. It consists of a time-series of feature vectors recorded in yearly intervals. Depending on the respective bank, there might be as many as 3 million data points available.

There are about 40 individual quantities recorded in one feature vector. Out of those, we mainly consider the relative savings amount (RSA). The RSA quantifies the amount of money saved over the last period of twelve months relative, in percent, to the total volume of the loan. It is the most important feature of the time-series, since it is the dominant factor for the further development of the contract, and hence the other recorded quantities, except demographical data etc., depend on it directly or indirectly. Modeling all 40 quantities can be easily accommodated in the HMM-Clustering framework we propose.

In the RSA time-series data a number of typical patterns can be observed, which correspond to different types of behavior. This motivates a theoretical interest in classifying and clustering this data. From a practical point of view, the clustering process is highly relevant as it is the first step towards simulation of the whole collection of contracts. Simulation is used for liquidity forecasting and hence as the basis for executive decisions such as investment strategies or contract design.

The observed time-series exhibit global patterns that correspond to certain deterministic constraints imposed by the terms and regulations of loan banking (e.g. the threshold D which specifies the end of the saving period). For a good model it is necessary that generated sequences also obey those constraints. In the next section we demonstrate how this non-Markovian behavior can be accounted for in HMM modeling.

5 Model extensions

The basic idea of our Model extension is to allow transition probabilities to vary, similarly to time inhomogeneous Markov chains. However, in our case the transition probabilities do not depend on time but rather on the partial sequence observed so far. As an example we consider a sequence of savings which, when summed, exceed the threshold D. In most cases the sequence will enter a state corresponding to amortizations instead of remaining in a saving phase state in the next time step.

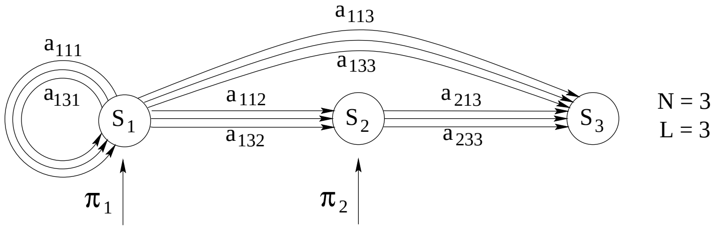

Fig. 2. Graph of an extended HMM with $L = 3$ conditional transition classes.

To accomplish this for generated sequences we extend the transition Matrix $A = (a_{i,j})$ to a set of matrices, cf. Fig2:

$$A \rightarrow (A_1, \ldots, A_L).$$

Suppose the model is in state i at time t and we already observed the partial sequence $(O_1, \ldots, O_t)$. The current transition matrix A_l is determined by the function

$$l = f(O_1, \ldots, O_t).$$

As a simple example we used the following step-function

$$f(O_1, \ldots, O_t) = \lceil L \sum_{\tau=1}^{t} O_\tau \rceil, \text{where} 0 \leq \sum_{\tau=1}^{t} O_\tau \leq 1.$$

Further extensions of f can be found in [22]. In [12] it is demonstrated how to modify the usual Baum-Welch reestimation formulas to be applicable to this model extension.

6 Experimental Results

We tested our training algorithm with data sets containing up to 50,000 time series from savings and loan bank data. In a first step we restricted our model to the saving period. Several different model topologies with varying number of states were examined, as well as variations on the number

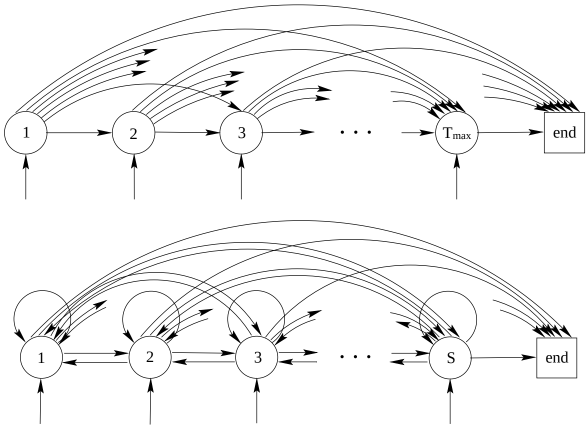

Fig. 3. Two different model types for the savings period: a simple left-right-model and a fully connected model topology. Note that the top one can only generate sequences up to length $T_{\max}$, while the length distribution for sequences generated by the bottom model is a mixture of exponentials.

of HMMs and transition matrices (not shown). Optimal and stable results were achieved with a simple left-right model with $N = 13$ states and self-transitions, cf. Fig. 3. The number of HMM clusters was $K = 9$ and the number of transition matrices was $L = 6$. This model parameters are used for the results in this section.

The first quantity to be examined was the sum of the relative savings amount (SRSA) per sequence. Fig. 4 displays the SRSA of the real data and the prediction of three different models: the currently used k-means model [6], the naive HMM approach and our extended model of section 5. The SRSA of the real data is 0.0 until approximately 37 %, reaches a sharp maximum at 39 % and has a long tail until 80 %. Note that the contracts require a savings amount of at least 40 % including interest.

Observe, that the k-means prediction has a much too sharp maximum and consequently a much too small variance due to the fixed time lengths of the k-means prototypes. The naive HMM approach achieves the maximum with high accuracy, but results in a much too broad length distribution. This can be avoided when using our extended model, where both the maximum and the variance are met.

A more complex model: further events in loan banking

Fig. 5 shows one of the model topologies we investigated with regard to its capability in modeling the complete course of loan bank contracts. The three periods saving, allocation, and repayment correspond to three distinct groups of model states. The states displayed as squares represent certain

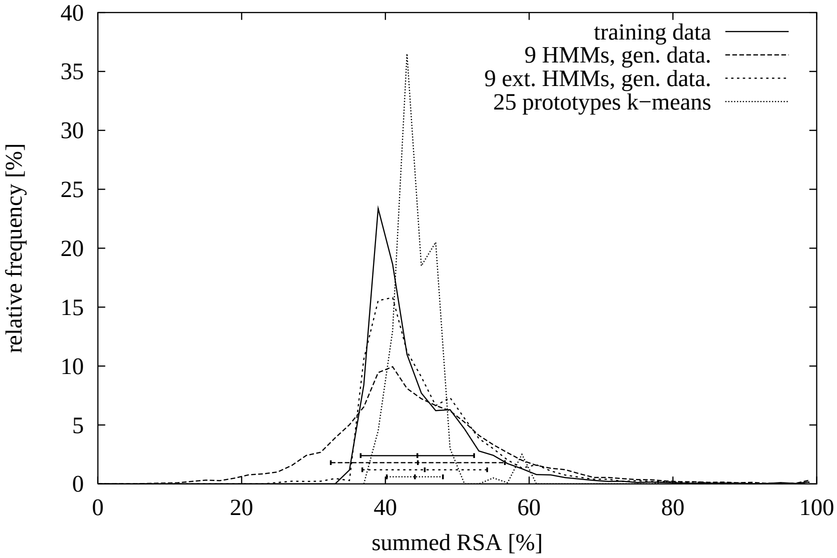

Fig. 4. Sum of relative savings amount (SRSA) for real data, generated data (HMMs and extended HMMs) and weighted k-means prototypes. The horizontal error bars show mean and standard deviation for the observed data.

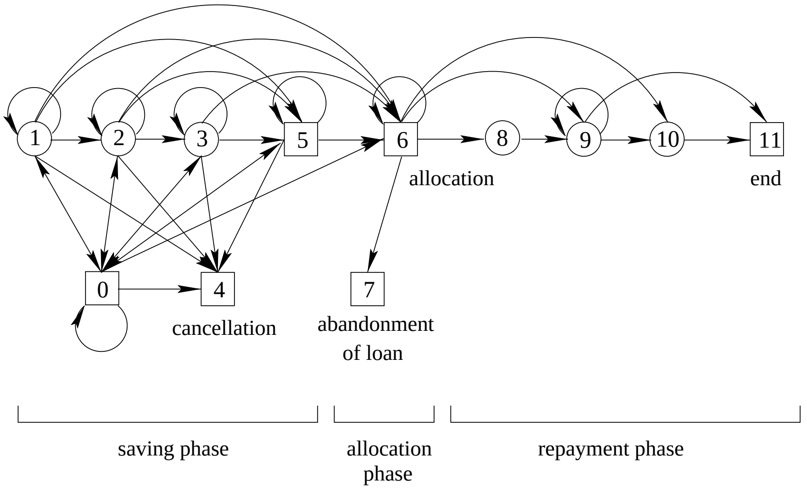

Fig. 5. Graph of an HMM for modeling the three phases of a loan banking contract.

important discrete events such as canceling the contract. The emission probability function of these states is characterized by a very small initial variance and an initial mean value which is used to represent the corresponding event. Furthermore, the emission parameters of these special states are never changed during training.

Another view is given in Fig. 6. Here the capability of extending truncated real sequences is displayed. The predicted data were generated by two different models (of same size and topology) which were trained on two different sets of sequences. The training sets are: *pred1* containing all contracts for the year 1985 and *pred2* containing contracts regardless of the contract year.

The truncated set consists of all contracts for the year 1986 and the sequences were truncated in 1992 and extended by the above mentioned models until a an end-state (e. g. a state with no outgoing transitions) was reached.

Fig. 6 shows the yearly savings amount (YSA) and the yearly amortizations (YAM) summed over all sequences of the two generated sets (*pred1*, *pred2*) und of the real data (*real*, not truncated here). The YSA data is closely approximated by both predictions. For the YAM graph the prediction using the training set *pred1* is more accurate.

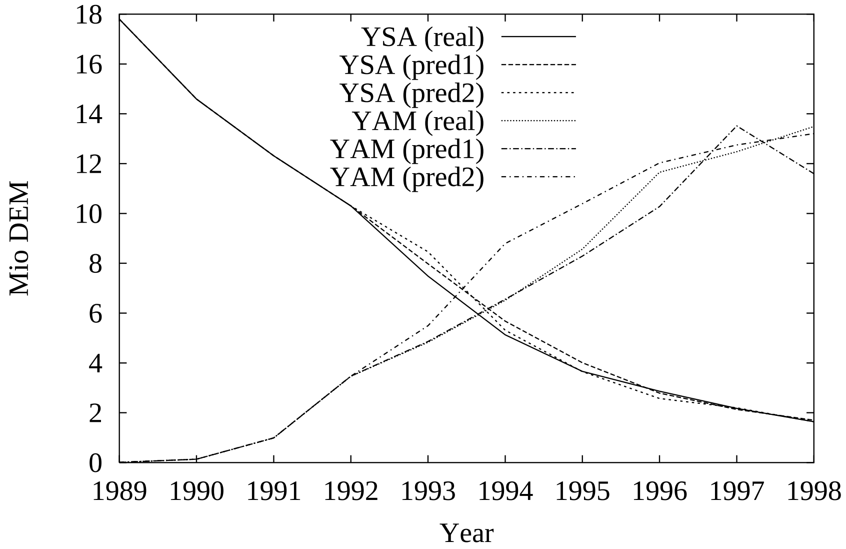

Fig. 6. Real and predicted yearly savings amount (YSA) and amortizations (YAM) for two different training scenarios. Prediction starts 1993.

7 Conclusion and Outlook

We presented a new algorithm for clustering data, which performed well for the task of generating statistical models for prediction of loan bank customer collectives. The generated clusters represent groups of customers with similar behavior. The prediction quality exceeds the previously used k-means based approach.

HMMs lend themselves to various extensions [8,21]. Therefore, we were able to incorporate many other relevant loan-bank parameters into our current model. These can then be estimated with *one* homogeneous statistical training algorithm, instead of using a collection of individual heuristics. We expect an even higher overall prediction accuracy and a further reduction of human intervention when applied to this and other application problems. These results will be described elsewhere [12,22]. The clustering approach is general in its applicability: An analysis of gene expression data from experimental genetics is forthcoming [18].

On a more pragmatic note: HMMs can be easily visualized, cf. Fig. 5. This visualization is a crucial aid in effectively communicating peculiarities of models to experts from the problem domain, who otherwise might not be able to fully participate in the modeling process.

8 Acknowledgments

The research was conducted at the Center for Applied Computer Science at the University of Cologne (ZAIK) and partially (BK, BW) funded by the German Landesbausparkassen. The authors would like to thank Prof. Dr. R. Schrader (ZAIK) for his support.

References

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

2.L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Statist.*, 41:164–171, 1970.

3.C. Becchetti and L. Prina Ricott. *Speech Recognition: Theory and C++ Implementation.* John Wiley & Sons, New York, 1999.

4.C. J. Burke and M. Rosenblatt. A markovian function of a markov chain. *Ann. math. stat.*, 29:1112–1120, 1958.

5.Robert J. Elliott, Lakhdar Aggoun, and John B. Moore. *Hidden Markov models.* Springer-Verlag, New York, 1995. Estimation and control.

6.Bachem A. et al. Analyse großer Datenmengen und Clusteralgorithmen im Bausparwesen. In C. Hipp, W. Eichhorn, W.-R., and W.-R. Heilmann, editors, *Beiträge zum 7. Symposium Geld, Finanzwirtschaft, Banken und Versicherungen, Dezember 1996*, number 257, pages 955–961, 1997.

7.B.S. Everitt. *Cluster Analysis.* Edward Arnold, London, 1993.

8.Z. Ghahramani and M. I. Jordan. Factorial hidden Markov models. *Machine Learning*, 29:245–273, 1997.

9.I. Holmes and W. J. Bruno. Finding regulatory elements using joint likelihoods for sequence and expression profile data. *Proc Int Conf Intell Syst Mol Biol*, 8:202–10, 2000.

10.X. D. Huang, Y. Ariki, and M. A. Jack. *Hidden Markov Models for Speech Recognition*. Edinburgh University Press, 1990.

11.B. Knab, R. Schrader, I. Weber, K. Weinbrecht, and B. Wichern. Mesoskopisches Simulationsmodell zur Kollektivfortschreibung. Technical Report ZPR97-295, Mathematisches Institut, Universität zu Köln, 1997.

12.Bernhard Knab. *Erweiterungen von Hidden-Markov-Modellen zur Analyse ökonomischer Zeitreihen*. PhD thesis, July 2000.

13.Bernhard Knab, Alexander Schliep, Barthel Steckemetz, Bernd Wichern, Achim Gädke, and Disa Thoransdottir. The GNU Hidden Markov Model Library. Technical report. Available from http://www.zpr.uni-koeln.de/ hmm.

14.Iain L. MacDonald and Walter Zucchini. *Hidden Markov and other models for discrete-valued time series*. Chapman & Hall, London, 1997.

15.G.J. McLachlan and K.E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, Inc., New York, Basel, 1988.

16.T. Petrie. Probabilistic functions of finite state Markov chains. *Ann. Math. Statist*, 40:97–115, 1969.

17.L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, February 1989.

18.Alexander Schliep, Alexander Schönhuth, Tobias Müller, and Christine Steinhoff. Probabilistic clustering of gene expression time series using HMM. Technical report. In preparation.

19.K. Sjolander, K. Karplus, M. Brown, R. Hughey, A. Krogh, I. S. Mian, and D. Haussler. Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology. *Comput Appl Biosci*, 12(4):327–45, Aug 1996.

20.P. Smyth. A general probabilistic framework for clustering individuals. Technical Report TR-00-09, University of California, Irvine, 2000.

21.S. Thrun and J. Langford. Monte carlo hidden Markov models. Technical Report CMU-CS-98-179, Carnegie Mellon University, Pittsburgh, PA, 1998.

22.Bernd Wichern. *Hidden-Markov-Modelle zur Analyse und Simulation von Finanzzeitreihen*. PhD thesis, November 2001.