# A Survey of Hidden Markov Models within the Bayesian Framework

**A. A. Salah[*], F. Köksal**
Department of Computer Engineering
Boğaziçi University
80815, Istanbul, Turkey
*salah,koksal@boun.edu.tr*

## Abstract

A much used tool in speech recognition and in other temporally sequential recognition tasks are Hidden Markov Models. Although developed as a stand-alone formalism, they are a special case of Bayesian belief networks. In this paper we define and describe HMMs from the standpoint of Bayesian decision theory. We devote a section to the important EM algorithm. We give examples and graphical representations to present the reader with a didactic introduction.

## 1 Introduction

Humans are often confronted with complicated decision situations in everyday life. Obviously, we do not process information the way a computer does, and our reasoning relies on heuristics rather than exhaustive probabilistic calculations, something that caused tumult in the rationality-believing world of philosophy [1, 2].

The most frequently used metaphor for human mental processing is the graph structure, where ideas are *connected* to, and sometimes *rooted* in other ideas [3]. Therefore it is only natural that scientists sought to depict relations visually, with -often directed- graphs.

What can be represented by a graph? Two frequently used models answer this question differently. The Finite State Machine (FSM) is a graphical representation, where each node represents a state of nature, and the arcs represent transitions between those states [4]. Thus, tracing a path on the FSM corresponds to looking at the transitions of the system represented by the FSM in time.

In Bayesian networks [5], nodes represent random variables, and arcs correspond to conditional dependencies. Note that the states of nature can be considered as random variables, this will help us later in reducing the Hidden Markov Model to a special case of Bayesian network.

The graphical models that use undirected graphs are called Markov Random Fields,

[*]This work is submitted as a term project for the IE 544 Decision Analysis course, June 8, 2001

those that use directed graphs are Bayesian networks. The direction in a directed graph implies the important notion of temporality. Implication relies heavily on appropriate temporal sequencing; if A is caused by B, then B must proceed A. It is not difficult to continue along this line and reach the concept of causality. But we will limit ourselves with the boundaries of probability and statistics, and introduce now the Markov Model. For a review of Bayesian networks, see [6, 7].

## 2 Hidden Markov Models

A Markov Model is a statistical model that relies on the assumptions that a signal can be characterized as a parametric random process, and that the parameters of this stochastic process can be determined [8]. What makes the model Markovian is that the data points that make up the signal are not independently and identically drawn from a probability distribution, but are linked to each other with a probabilistic dependence relation.

Consider a FSM where we have $N$ distinct states (nodes of the graph), and some transition arcs between those states. At a given time, the system characterized by this FSM will be in one, and only one of those states. Suppose that we have a discrete-time system, and we look at it at each time step $t$. The Markov assumption that lends its name to the model is that the probability of a state transition from state $S_t$ to state $S_{t+1}$ at time $t$ depends only on $S_t$, and not on the previous states. This means that once the system enters a state, it immediately *forgets* its history. We have just described a first-order Markov chain. A second-order Markov chain would be one in which the state transition probability at time $t$ depends on both $S_t$ and $S_{t-1}$.

Now, suppose that we have an observation sequence $O = O_1, O_2, ..., O_T$ that is generated by the system. This is actually the stochastic signal we have mentioned in the first paragraph of this section. If we know the state sequence $S = S_1, S_2, ..., S_T$ corresponding to this signal (where an observation symbol $O_i$ is generated while the system is in state $S_i$), we will have an Observable Markov Model (See Fig. 1 for a simple example). In this case, we can trace the state transitions and observation symbol generations in time.

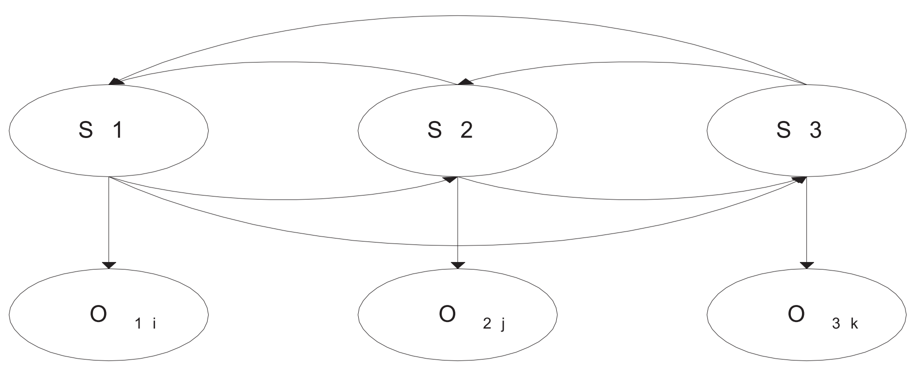


Figure 1: A Markov Model with three states

Unfortunately, most of the time we do not have access to the states, let alone the state transitions, of the system. We will have a set of observed signals, and maybe some rough idea about the complexity or nature of the system. In this case, we have a Hidden Markov Model (HMM): the states are hidden.

The joint probability distribution of a sequence of observations $O = O_1, O_2, ..., O_T$ can be calculated as

$$P(O_1, ..., O_T) = P(O_1) \prod_{t=2}^{T} P(O_t | O_1, ..., O_{t-1}), \tag{1}$$

regardless of the generating model [9]. By using the first-order Markov assumption and describing the probability of being in state $S_i$ at time $t$ as $P(q_t = S_i)$, the probability of an observation sequence with the associated state sequence in an Observable Markov Model is given by

$$P(O_1, ..., O_T, S_1, ..., S_T) = P(q_1 = S_1) P(O_1 | S_1) \prod_{t=2}^{T} P(S_t | S_{t-1}) P(O_t | S_t) \tag{2}$$

HMMs are useful to model sequential data, and they are applied to problems in speech recognition [8], bioinformatics [10, 11], dynamic handwritten digit recognition [12], and others. The discretization of analogue time, followed by a first-order Markovian assumption grossly simplifies the dependencies between the variables, and makes HMMs a good starting point in dealing with complex sequential data.

### 2.1 Elements of HMM

Following [13], we now formally specify an HMM by a set of parameters.

1. $N$: The number of states in the model
$$S = \{S_1, S_2 ..., S_N\}$$
2. $M$: The number of distinct observable symbols in the *alphabet*
$$V = \{v_1, v_2 ..., v_M\}$$
3. State transition probabilities
$$A = [a_{ij}] \;\; where \;\; a_{ij} \equiv P(q_{t+1} = S_j | q_t = S_i)$$
4. Observation probabilities
$$B = [b_j(m)] \;\; where \;\; b_j(m) \equiv P(O_t = v_m | q_t = S_j)$$
5. Initial state probabilities
$$\Pi = [\pi_i] \;\; where \;\; \pi_i \equiv P(q_1 = S_i)$$

Furthermore, we use $\lambda = (A, B, \Pi)$ to denote the parameter set that completely specifies the HMM. In the classical HMM literature, three main problems are defined:

1. Given a model $\lambda$, how do we find the probability $P(O|\lambda)$ of an observation sequence, $O = O_1, ..., O_T$?
2. What is the state sequence $Q = q_1, ..., q_T$ which best explains $O$ given $\lambda$?
3. How do we find $\lambda$ that maximizes the probability of generating $O$?
$$\lambda^* = arg \max_{\lambda} P(O|\lambda)$$

The last problem is the model learning, which we devote much of the current work. We will briefly mention the solutions to the first two problems. We have omitted the detailed equations for each problem, these are much repeated in the literature. For a good review, see [8], where the techniques we will mention in the next few paragraphs are described.

The way to calculate the probability of a given observation sequence is to marginalize over all possible paths in the model. This task seemingly requires an exponential number of calculations. However, the *forward-backward procedure* can be employed to compute probabilities of partial observation sequences and derive the desired probability by combining them in polynomial time.

To find the best state sequence, we use the *Viterbi algorithm* that formulates a maximization problem with an optimality criterion that maximizes the expected number of correct individual states. In each time step $t$, we select the maximum probability path to reach a given state in $t-1$ steps, and continue iterating until we reach step $T$.

The last problem is solved by the *Baum-Welch procedure* [14], which is a variant of the Expectation-Maximization (EM) algorithm. At each iteration, first we compute the probabilities of being in state $i$ at time $t$ and the marginal of these arc probabilities over all possible next states, given $\lambda = (A, B, \Pi)$. Then we update $\lambda$, given the probabilities computed in the first step. This procedure is guaranteed never to decrease $P(O|\lambda)$. Although it seems to solve all our problems, we will show why this is not the case in the discussion section.

## 3 Bayesian Network Representation of the HMM

We can think of the HMM as a special case of Bayesian network [15]. Assume that we observe the system output $O$ for time $1...T$, and represent the (hidden) states the system enters with $q_1, ..., q_T$. Our first-order Markov assumption implies that each state $q_i$ depends only on state $q_{i-1}$. We show how this structure can be represented as a Bayesian network in Fig 2.

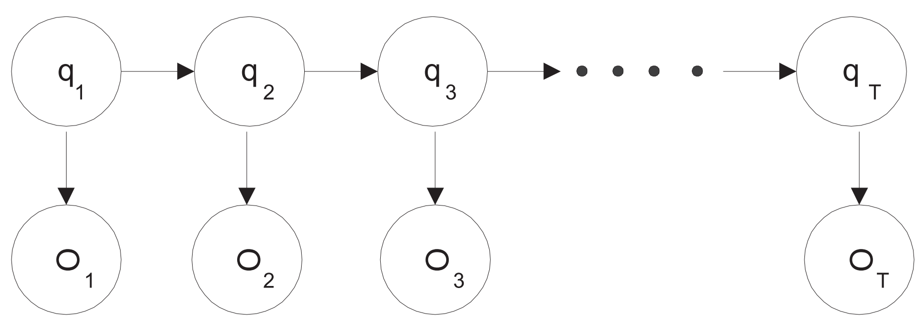


Figure 2: The Bayesian network depicting an HMM, unrolled for $T$ time slices.

Actually, the graphical representation of the Bayesian networks is so simple and powerful, almost all the probabilistic models can be represented as Bayesian networks in one way or other, as shown in Figure 3 [16].

Since the fully connected HMM is a cumbersome model, variants of HMMs have been developed over time. In Factorial HMMs, the states are not represented by

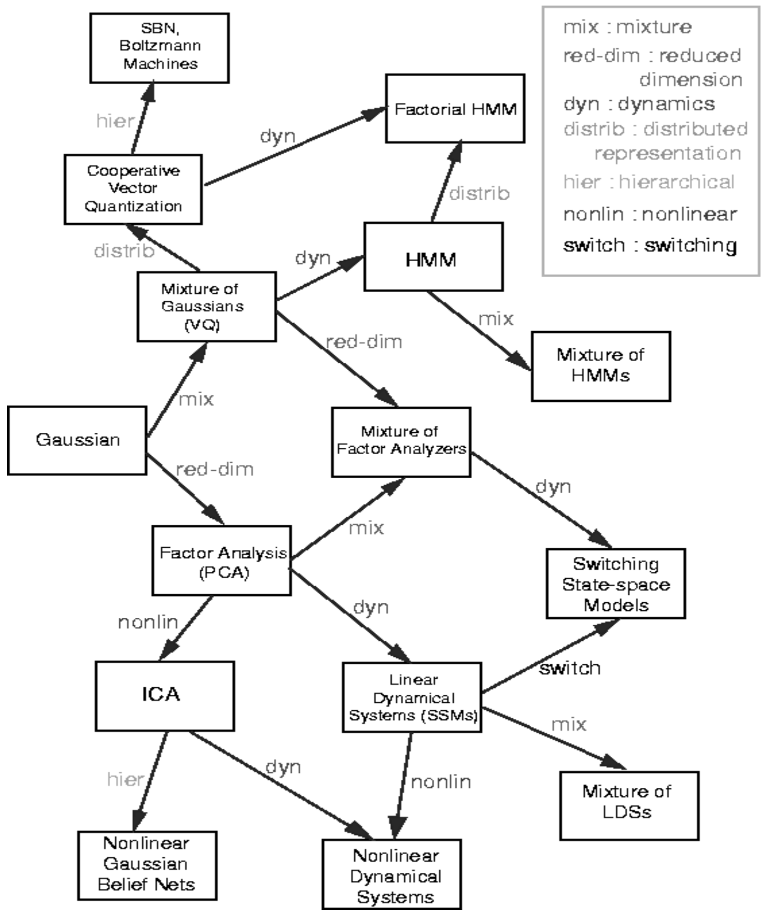


Figure 3: A summary of the relationships between some popular Bayesian networks, dubbed as a generative model for generative models by K. Murphy.

a single random variable, but a collection of state variables is used [17]. These variables are constrained in order not to increase the number of variables beyond computationally feasible bounds. The HMM can also be augmented to allow for input variables, such that it models the conditional distribution of sequences of output observations given sequences of inputs [18]. This type of input-output HMMs were also used for tree structured architectures [19].

## 4 Learning in Graphical Models and the EM Algorithm

In the learning of a complete graphical model from data, we need to consider two related learning tasks in parallel. The first one is learning the structure of the model, and the second is the learning of the parameters of the model, given the selected structure. In many learning tasks, especially in HMMs, some instance variables may be unobservable. For example in HMMs the exact state sequence is unknown. Furthermore, the training sample may contain missing values that need to be estimated.

We define in Table 1 four learning cases, depending on the structure of the model and on the observability of the variables.

Table 1: Learning in graphical models

| **Structure** | **Observability** | **Method** |
|---|---|---|
| Known | Full | Maximum Likelihood Estimation |
| Known | Partial | EM (or gradient ascent) |
| Unknown | Full | Search through model space |
| Unknown | Partial | EM + search through model space |

The EM algorithm [20] is the most frequently used method in computational statistics. It finds the maximum likelihood parameter estimates in problems where the structure of the model is known, but some variables cannot be observed. The true likelihood value of the model cannot be determined, but approximated in arbitrarily fine resolution by working on a tractable approximation problem. The name expectation-maximization comes from the fact that there exist two different estimation steps that need to be performed consecutively.

The first step is the estimation of the distribution for the unobserved variables, given the known values for the observed variables and the current estimate of the parameters (this is the (E)xpectation step). The second step is the re-estimation of the parameters to maximize the likelihood by assuming that the distribution found in the first step is correct. This is the (M)aximization step. It is shown that each iteration either improves the true likelihood value or does not change the current value, although does not necessarily find the global optimum. Neal and Hinton show that the convergence of the EM algorithm can be speeded up by an incremental variant that calculates only the distribution of one of the unobserved variables in each E step [21].

The EM algorithm is a generic framework and can be customized for different models [22, 23, 24]. For example the Baum-Welch re-estimation procedure is the EM algorithm applied to HMMs. However, it should be noted that the EM algorithm for HMMs is used for the prediction of both the structure and the parameters of the model [25].

In the most general setting of dynamic probabilistic networks, the EM algorithm manifests itself in a very simple form. The E-step of EM uses the currently estimated parameters to complete the data by computing the expected counts. The M-step then re-estimates the maximum likelihood parameter values as if the expected counts were true observed counts [26]. It is shown that Principal Components Analysis, Factor Analysis, mixture of Gaussian clusters, and Kalman filters can be unified as variations of unsupervised learning models [16]. The EM algorithm applies to all the mixture models under this categorization [27]. Due to this flexibility, EM has been one of the most researched areas of computer science (Stuart Russell remarks that every year, 200-300 research papers are published in which EM is the topic or the basic tool), and it has found application in diverse fields, such as efficient VLSI implementation of neural networks [28].

## 5 Discussion

In this section, we wish to discuss several issues that we think are more relevant than others. We have emphasized structure prediction in HMM, because this is a hot research issue. We also give a few remarks on the Dirichlet priors, that will help in giving an intuition about statistical modeling of data.

### 5.1 Structure Prediction

Much of the research around HMMs deal with structure prediction and topology optimization. Selecting an arbitrary number of states and deriving probabilities for a fully-connected topology does not use the sample information in structuring. Instead, different model structures can be evaluated against each other, and the model can be selected in a data-driven manner.

It should be obvious that our aim to maximize the posterior probability of data will not be an appropriate guide in selecting models, since increasing the model complexity will usually result in a better fit to the training data. Plainly, we need to have a balancing factor that keeps us from overlearning. Usually, this is done by associating a prior probability with the model. According to Bayes' rule, the posterior probability $P(M|x)$ of the model is proportional to the product of the model prior $P(M)$ and $P(x|M)$, the likelihood of the sample set given the model. Thus, although more complex models will have increased likelihoods, their model priors will compensate for this increase. Actually, they need to do more than that, since $P(M|x)$ needs to decrease asymptotically as the model complexity goes to infinity.

The use of priors is sometimes seen as a weakness, because there is no mathematical justification for a particular formula. Bayesians usually point out to the fact that the effects of priors diminish as the size of the sample is increased, because the likelihood $-log\, P(x|M)$ typically increases with the increased size of $x$, but $-log\, P(M)$ remains constant. See [29] (especially the chapter on Bayesian inference and the appendix on information theory) for a critique of the use of priors.

Buntine defines two contributing terms for the model prior [30]. The first term accounts for the model structure, and the second term concerns the number of adjustable parameters, given the structure. The structure prior relies on a mechanism similar to the *minimium description length* [31, 32], where structures requiring longer descriptions are penalized.

Stolcke and Omohundro use this method to determine HMM structures that have good generalization over the training data [33, 34]. Their method starts by generating large models that fit the data in the best possible way, and then the states are merged to maximize the posterior probability by increasing the model prior term (inevitably, the likelihood term decreases during the process). The other logical way to generate good models would be to start with minimal models, and grow the model to fit the data. This approach is taken in [35].

In Bayesian networks, 'Occam factors' were used to select models with fewer parameters [36]. The MDL principle was used by Lam and Bacchus to derive Bayesian networks from data [37]. For a survey on model selection in Bayesian networks, see for example [38].

### 5.2 Dirichlet Priors

Assuming a prior distribution on the parameters and finding the maximum likelihood under this assumption is beneficial in controlling the complexity of the resulting model, provided that the prior is selected appropriately. In the literature, the most frequently used prior is the Dirichlet distribution [39].

Remember that finding the best structure for a given data set is intractable for both HMMs and Bayesian networks [40]. Therefore iterative approximation techniques are used to optimize the structure. In order to work with the iterative techniques, we need to select the statistical distributions that characterize our system appropriately.

Consider the initial state distribution in the HMM. Cleary, the likelihood is multinomial. We reach the posterior by multiplying the prior with the likelihood. If the distribution of prior is conjugate to the multinomial distribution, as it is the case with the Dirichlet distribution, the posterior will be in the same family as the prior. Thus, the posterior may enter the next step of the iterative process as the prior. This is the main reason behind selecting the Dirichlet distribution [41].

The second important point is that the Dirichlet parameters can be interpreted as virtual observations. Thus, it has the effect similar to using a larger training set. For an appropriate selection of the parameters, the effect resembles adding one observation for each possible variable. This alleviates the undesired situation of having probabilities equal to zero in the resulting model (undesirable, because the probabilities are multiplied). It also provides a crude theoretical justification to the common regularization method for HMMs in which a small positive number is added to all the parameters. In fact we have used this regularization method in our previous work on Markov models [12], and noted an increase of several standard deviations in the classification accuracy of the Observable Markov Model.

## 6 Sources

Since this is a term project, we took the liberty to comment on some of the sources we have used, with the hope of increasing the usefulness of this work.

Most of the papers are downloaded from ResearchIndex Digital Library (http://citeseer.nj.nec.com/cs). It is uniquely the most useful resource on the Internet that allows tracking of citations, full postscript or pdf texts and instant access to abstracts.

The early paper by Pearl [3] is not available from his homepage, but can be reached from the UCLA Technical Report Repository (http://www.cs.ucla.edu/csd/pubs/pubs.html). It is very readable, and we recommend it as an introductory reading. The books by Pearl [5] and Hopcroft & Ullman [4] are both classics, and available from the library. The book by Baldi and Brunak [29] is recently made available in the library; it is an excellent text that goes way beyond the scope of bioinformatics.

In the first section of the second chapter, we have duplicated the elements of the HMMs exactly from the lecture notes of Alpaydın [13]. The representation is identical with Rabiner [8], and very concise.

A very useful tutorial was "A Brief Introduction to Graphical Models and Bayesian Networks" (http://www.cs.berkeley.edu/˜murphyk/Bayes/bayes.html) and the "HMM Toolbox" (/Bayes/hmm.html) from the same web site, both prepared by Kevin Murphy. These contain useful links to important readings and code as well.

Some of the papers were reviewed by our Machine Learning Group [7, 8, 27, 30, 41]. The last paper we have reviewed in our group is Ghahramani's survey paper [41]. The paper attempts to cover many things, but contains many important typos (in equations), and structural mistakes (like mentioning some things in the nonexistent appendix); not a very succesful paper in our opinion.

The papers of Cohen [1] and Gardner [2] were among the reading material from the Introduction to Cognitive Science course, given last year under the Linguistics program. The literature on rationality is quite large, but Cohen has a sound style (although we do not share his point of view in many things).

We have referenced some papers without reading them [14, 20, 31, 32]. These were the classical references that introduced their topics. Some of the other papers are important references that were slightly off topic [10, 11, 23], we have just skimmed through them.

## References


[1] Cohen, L.J., "Can human irrationality be experimentally demonstrated?", *Behavioral and Brain Sciences,* vol.4, pp.317-370, 1981.

[2] Gardner, H., "How rational a being," The Mind's New Science, 1985.

[3] Pearl, J., "Markov and Bayes networks: a comparison of two graphical representations of probabilistic knowledge," UCLA Computer Science Department Technical Report 860024 (R-46), Oct 1986.

[4] Hopcroft, J.E., Ullman, J.D., *Introduction to Automata Theory, Languages, and Computation,* Addison-Wesley, Reading, Mass., 1979.

[5] Pearl, J., *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference,* Morgan Kauffman, 1988.

[6] Charniak, E., "Bayesian networks without tears," *AI magazine,* 1991.

[7] Heckerman, D., "A tutorial on learning with Bayesian networks," Microsoft Research tech. report, MSR-TR-95-06, 1996.

[8] Rabiner, L.R., "A tutorial on Hidden Markov Models and selected applications in speech recognition," *Proceedings of the IEEE*, vol.17, no.2, Feb 1989.

[9] Bengio, Y., "Markovian models for sequential data," Technical Report 1049, Dept. IRO, Université de Montréal, 1996.

[10] Krogh, A., Brown, M., Mian, I.S., Sjölander, K., Haussler, D., "Hidden Markov models in computational biology: Applications to protein modeling," *Journal of Molecular Biology,* vol.235, pp.1501-1531, 1994.

[11] Baldi, P., Chauvin, Y., Hunkapiller, T., McClure, M.A., "Hidden Markov models of biological primary sequence information," *Proceedings of the National Academy of Sciences,* vol.91, no.3, pp.1059-1063, 1994.

[12] Salah, A.A., Alpaydın, E., Akarun, L., "A Selective Attention Based Method for Visual Pattern Recognition," to appear in the *Proceedings of the* $23^{rd}$ *Annual Conference of the Cognitive Science Society*, 2001.

[13] Alpaydın, E., CmpE 545 Artificial Neural Networks, lecture notes, 1999.

[14] Baum, L.E., Petrie, T., Soules, G., Weiss, N., "A maximization technique occuring in the statistical analysis of probabilistic functions of Markov chains," *Annals of Mathematical Statistics,* vol.41, no.1, pp.164-171, 1970.

[15] Smyth, P., Heckerman, D., and Jordan, M.I., "Probabilistic independence networks for hidden Markov probability models," *Neural Computation,* vol.9, pp.227-269, 1997.

[16] Roweis, S.T., Ghahramani, Z., "A unifying review of linear Gaussian models," *Neural Computation,* vol.11, no.2, pp.305-345, 1999.

[17] Ghahramani, Z., Jordan, M.I., "Factorial hidden Markov models," *Machine Learning,* vol.29, pp.245-275, 1997.

[18] Rimey, R.D., Brown, C.M., "Selective Attention as Sequential Behavior: Modeling Eye Movements with an Augmented Hidden Markov Model," Technical Report TR-327, Computer Science, University of Rochester, February 1990.

[19] Bengio, Y., Frasconi, P., "An input-output HMM architecture," in Tesauro, G., Touretzky, D.S., Leen, T.K. (eds.), *Advances in neural information processing systems 7*, pp.427-434, 1995.

[20] Dempster, A.P., Laird, N.M., and Rubin, D.B., "Maximum likelihood from incomplete data via the EM algorithm," *Journal of Royal Statistics Society,* (B), vol.39, pp.1-38, 1977.

[21] Neal, R.N., and Hinton, G.E., "A new view of the EM algorithm that justifies incremental, sparse and other variants," in M.I. Jordan (ed), *Learning in Graphical Models*, pp.355-368, 1998.

[22] Blimes, J.A., "A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models," Technical Report TR-97-021, International Computer Science Institute (ICSI) and Computer Science Division, Dept. of Electrical Engineering and Computer Science, U.C. Berkeley, 1998.

[23] Roweis, S.T., "EM algorithms for PCA and SPCA," *Advances in Neural Information Processing Systems 10*, 1998.

[24] Ghahramani, Z., Hinton, G.E., "The EM algorithm for mixtures of factor analyzers," Technical Report CRG-TR-96-1, University of Toronto, (revised), 1997.

[25] Ghahramani, Z., "Learning dynamic bayesian networks," in C.L. Giles and M. Gori (eds.), *Adaptive Processing of Temporal Information*, Lecture Notes in Artificial Intelligence. Springer Verlag, 1997.

[26] Friedman, N., Murphy, K., Russell, S.J., "Learning the structure of dynamic probabilistic networks," in *Proceedings of Uncertainty in Artificial Intelligence,* pp. 139-147, 1998.

[27] Jordan, M.I. and Jacobs, R.A., "Hierarchical mixtures of experts and the EM algorithm," *Neural Computation,* vol.6, pp.181-214, 1994.

[28] Köksal, F., Alpaydın, E., Dündar, G., "Weight Quantization for Multi-Layer Perceptrons," to appear in the *Proceedings of the International Conference on Artificial Neural Networks,* 2001.

[29] Baldi, P., Brunak, S., *Bioinformatics: The Machine Learning Approach,* MIT Press, 1998.

[30] Buntine, W.L., "A guide to the literature on learning probabilistic networks from data," *IEEE Transactions on Knowledge and Data Engineering*, yol.8, pp.195-210, 1996.

[31] Rissanen, J., "Modeling by shortest data description," *Automatica*, vol.14, pp.465-471, 1978.

[32] Rissanen, J., "A universal prior for integers and estimation by minimum description length," *The Annals of Statistics*, vol.11, no.2, pp.416-431, 1983.

[33] Stolcke, A., and Omohundro, S., "Hidden Markov Model induction by Bayesian model merging," in S.J. Hanson, J.D. Cowan, and C.L. Giles (eds.), *Advances in neural information processing systems,* vol.5, pp.11-18. San Francisco, CA: Morgan Kaufmann, 1993.

[34] Stolcke, A., and Omohundro, S., "Best-first model merging for Hidden Markov Model induction," Technical Report, TR-93-003, International Computer Science Institute, Berkeley, Ca., 1993.

[35] Ostendorf M., Singer, H., "HMM topology design using maximum likelihood successive state splitting," *Computer Speech and Language,* vol.11, no.1, pp.17-41, 1997.

[36] MacKay, D.J.C., "Bayesian interpolation," *Neural Computation,* vol.4, no.3, pp.415-447, May 1992.

[37] Lam, W., Bacchus, F., "Learning Bayesian belief networks: An approach based on the MDL principle," *Computational Intelligence,* vol.10, pp.269-293, 1994.

[38] Kearns, M., Mansour, Y., Ng, A.Y., Ron, D., "An Experimental and Theoretical Comparison of Model Selection Methods," in *Proceedings of the Seventh Workshop on Computational Learning Theory,* ACM Press, Morgan Kaufmann, 1995.

[39] Geiger, D., Heckerman, D., "A characterization of the Dirichlet distribution through global and local independence," Technical Report MSR-TR-94-16, Microsoft Research, (revised), 1996.

[40] Saul, L.K., Jordan, M.I., "Exploiting tractable substructure in intractable networks," in Touretzky, D.S., Mozer, M.C., Hasselmo, M.E. (eds.), *Advances in Neural Information Processing Systems 8*, MIT Press, pp. 486-492, 1996.

[41] Ghahramani, Z., "An introduction to Hidden Markov Models and Bayesian Networks," *International Journal of Pattern Recognition and Artificial Intelligence,* vol.15, no.1, pp.9-42, 2001.