Scientific Programming 10 (2002) 185–199
IOS Press

Fitting hidden Markov models to psychological data

Ingmar Visser*, Maartje E.J. Raijmakers and Peter C.M. Molenaar
Department of Psychology, Developmental processes research group, University of Amsterdam

Abstract. Markov models have been used extensively in psychology of learning. Applications of hidden Markov models are rare however. This is partially due to the fact that comprehensive statistics for model selection and model assessment are lacking in the psychological literature. We present model selection and model assessment statistics that are particularly useful in applying hidden Markov models in psychology. These statistics are presented and evaluated by simulation studies for a toy example. We compare AIC, BIC and related criteria and introduce a prediction error measure for assessing goodness-of-fit. In a simulation study, two methods of fitting equality constraints are compared. In two illustrative examples with experimental data we apply selection criteria, fit models with constraints and assess goodness-of-fit. First, data from a concept identification task is analyzed. Hidden Markov models provide a flexible approach to analyzing such data when compared to other modeling methods. Second, a novel application of hidden Markov models in implicit learning is presented. Hidden Markov models are used in this context to quantify knowledge that subjects express in an implicit learning task. This method of analyzing implicit learning data provides a comprehensive approach for addressing important theoretical issues in the field.

1. Introduction

Markov models have been used in psychology at least since the 1950's [27,28]. They have been applied mostly in the areas of learning and memory [3,21,30]. In the area of learning, Markov models have proven to be very flexible models in describing and formalizing the development of knowledge. Although hidden or latent Markov models have been around for a while in psychology (see e.g. [45]), there have been relatively few applications. This is possibly due to inherent problems in estimating latent variable models. Estimation of parameters was usually based on the method of moments, which is hard to adapt to different kinds of data to be modeled. Using method of moments estimation, it is not feasible to model long sequences of trials or many different sequences of trials, such as those gathered in implicit learning experiments. Hidden Markov models are very flexible and can be used to model any set of sequences of trials, whether these are fixed length sequences, single sequences, or multiple sequences of different lengths. There is a great advantage of using discrete HMMs in the context of implicit learning and concept identification which is that the raw data from these experiments are modelled instead of derived measures which is necessary when using continuous models. New applications are available due to the flexibility in parameter estimation. The maximum likelihood framework provides methods for comparing models with different constraints imposed on their parameters. Adopting the framework of hidden Markov model (HMM) parameter estimation in applications of Markov models has many advantages.

In spite of the improvement of model estimation, for applications in psychology to be feasible, some important statistical features are lacking in the HMM framework. First, model selection criteria are needed to compare models and to decide which model best describes the data. In the present paper we compare and evaluate several candidate criteria for example data sets. Second, absolute measures for goodness-of-fit, called model assessment criteria, are needed to decide whether a model is adequate for the data at hand. We propose

*Corresponding author: Ingmar Visser, Department of Psychology, Roetersstraat 15, 1018 WB Amsterdam, The Netherlands. Tel.: +31 20 525 6735; Fax: +31 20 639 0279; E-mail: op_visser@macmail.psy.uva.nl.

ISSN 1058-9244/02/$8.00

a prediction error measure for this purpose, which is applicable to a wide range of Markov models. Other measures for testing goodness of fit, and for testing specific hypotheses, are considered as well. Third, in some applications it may be necessary to impose equality or other linear constraints on parameters for theoretical reasons. As far as we are aware, equality constraints in HMMs have received little attention. We compare three methods for fitting equality constraints, and, in specific cases, general linear constraints. There is similar work on equality constraints in latent class analysis [29], which points to difficulties in finding general solutions for fitting equality constraints. We developed a program for fitting HMMs, based on the EM algorithm [33], which incorporates all these features. Many goodness-of-fit statistics are standard output, others are available on request [40].

1.0.0.1. Hidden versus latent Markov models In the psychological literature on the subject hidden Markov models are usually referred to as latent Markov models (LMM [45]). In fact, the literature on latent Markov models, which are used in sociological and psychological applications, is largely separate from the literature on hidden Markov models which are mainly used in speech recognition and biological sequence analysis. In this paper we use the term hidden Markov model for two main reasons. The main difference in the literature between HMMs and LMMs is the kind of data they applied to. Although this may not seem a principled difference, the consequences for parameter estimation are profound. Fitting HMMs to timeseries data asks for different parameter estimation techniques than those that are usually applied in the context of LMMs. Hence, we refer to the models as HMMs while being aware that they are identical to LMMs.

1.0.0.2. Overview of the paper In the present paper only discrete hidden Markov models are considered, that is, HMMs with a discrete hidden state space and discrete observation symbols. In Section 2, we first present the definitions and notation that we use throughout the paper. Next, we describe a toy model and data set. We discuss model fitting with and without equality constraints, using this data set. In Section 3, we compare and evaluate model selection criteria and criteria for assessing goodness-of-fit. In Sections 4 and 5, we present two applications of fitting HMMs to experimental data. The first data set is from a concept identification experiment. Models are fitted in both exploratory and confirmatory analyses. We discuss fitting an HMM with a linear constraint between parameters, which is based on theoretical considerations. We show how the likelihood ratio statistic can be used to test the tenability of such constraints. The second data set is from an implicit learning experiment. In this experiment subjects unconsciously learn finite state languages by reproducing them. In order to gain insight into the knowledge that subjects acquire in such an experiment, we fit HMMs to the data. In addition, we discuss the connection between HMMs, finite state automata and regular languages in this section.

2. Fitting hidden Markov models

2.1. Definitions and notation

A discrete HMM may be represented as a five-tuple $< S, O, \mathbf{A}, \mathbf{B}, \pi >$. S represents a set of states $S_i, i = 1 \ldots n$. O is a set of observation symbols $O_j, j = 1 \ldots m$. Observation symbols are alternately called observations, symbols or responses. $\mathbf{A}$ represents a transition matrix with conditional probabilities $a_{ij}, i, j = 1 \ldots n$ of moving from state S_i to state S_j, i.e. $a_{ij} = P(S_i|S_j)$. $\mathbf{B}$ is the matrix of conditional observation probabilities b_{ij} of observing symbol (or category) O_j in state S_i, i.e. $b_{ij} = P(O_j|S_i)$. π is a vector of initial, unconditional probabilities π_i of starting in state S_i, i.e. $\pi_i = P(S_i)$. All parameters are probabilities and are subject to constraints of the form $\sum_{j=1}^{n} a_{ij} = 1$, and similarly for the observation probabilities ($\sum_{j=1}^{n} b_{ij} = 1$) and initial state probabilities ($\sum_{i=1}^{n} \pi_i = 1$). The parameters together are denoted $\lambda = (\mathbf{A}, \mathbf{B}, \pi)$. This notation is taken from [33], and is used throughout this paper.

2.2. Toy model and data

In this section, we consider a two-state HMM with three observation symbols. The two states are called S_1 and S_2 and the observation symbols 1, 2, and 3. The parameter values are:

$$\mathbf{A} = \begin{pmatrix} 0.9 & 0.1 \\ 0.3 & 0.7 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} 0.7 & 0.0 & 0.3 \\ 0.0 & 0.4 & 0.6 \end{pmatrix}$$

$$\pi = \begin{pmatrix} 0.5 & 0.5 \end{pmatrix}$$

From this model, we generated a data set consisting of a single sequence of 1000 symbols. This data set is used below to illustrate parameter estimation, estimation with equality constraints, model selection, and model assessment.

2.3. Parameter estimation

Throughout this paper, parameter estimates are obtained by maximizing the likelihood using the Baum-Welch or EM algorithm for HMM parameters [33]. The EM algorithm is an iterative procedure for finding maximum likelihood (ML) parameter estimates of a given model and a data set. The likelihood of a data set is denoted $P(\mathbf{O}|\lambda)$. The general expression for the likelihood is [33, p. 272]:

$$P(\mathbf{O}|\lambda) = \sum_{q \in Q} \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \ldots a_{q_{T-1} q_T} b_{q_T}(O_T) \tag{1}$$

where $\mathbf{O}$ is a sequence of T observations $O_1 \ldots O_T$, $b_{q_t}(O_t)$ is the observation probability b_{ij} with $i = q_t$ and $j = O_t$. The sum runs over all possible sequences of the hidden states. The EM algorithm finds the parameter values λ that maximize the likelihood. In our implementation of the EM algorithm we maximize the logarithm of the likelihood or the loglikelihood, rather than the likelihood itself. This is necessary because for long sequences, computing the likelihood leads to problems with underflow, i.e., the likelihood becomes too small to compute as can be seen easily from the expression for the likelihood above.

The loglikelihood may have many local maxima. Hence, in order to retrieve the model parameters and to find the global maximum of the loglikelihood, it is necessary to fit a model repeatedly with different starting values for the parameters. On the data set specified above, we fitted 100 two-state models, using random starting values for the model parameters λ. All the models converged to the same solution, up to a reshuffling of states. Note that in specifying an HMM, states are assigned arbitrary designations, and hence the identity of a given state is only determined by the transition probabilities to other states, the observation symbol probabilities, and the initial state probabilities. The loglikelihood of the model equals -883.068. The parameter estimates of the fitted model is

$$\mathbf{A} = \begin{pmatrix} 0.888 & 0.112 \\ 0.316 & 0.684 \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} 0.708 & 0 & 0.292 \\ 0 & 0.384 & 0.616 \end{pmatrix}$$

$$\pi = \begin{pmatrix} 1 & 0 \end{pmatrix}$$

Note that the parameter estimates are very close to their true values, except for the initial state probabilities which are estimated as one for state S_1 and zero for state S_2. The reason for this is that the sequence of symbols that was generated actually starts with the symbol 1 which can only be produced from state S_1. As a result, the value for that initial state probability has to be one. In order to be able to estimate initial state probabilities, multiple sequences have to be available because these probabilities can only be estimated on the basis of the first symbols of a number of sequences.

When fitting HMMs to single sequences it is (almost) always the case that one of the initial state probabilities is estimated at a value of one and the others at zero. For long sequences of observations, the contribution of this parameter to the loglikelihood is negligible. This is not the case for non-ergodic models, i.e. models with absorbing states, but in general, non-ergodic models can not be estimated on the basis of single sequences. In Section 4 a non-ergodic model is fitted on multiple sequences.

2.4. Equality constraints

In some applications it may be desirable to estimate parameters subject to equality constraints. Some models of concept identification and paired associate learning, contain free parameters that theoretically should be equal. In these applications, the hidden states of an HMM are interpreted as knowledge states. Theory may dictate that in some situations, or at specific trials, two knowledge states should lead to identical responses. As a consequence, the associated observation probabilities should be estimated at equal values. In the EM algorithm for HMMs, it is not obvious how to implement equality constraints within the steps of the algorithm itself. At each iteration of the algorithm, the model parameters are re-estimated independently of each other. Re-estimation is only subject to row sum constraints that ensure that the rows of the matrices $\mathbf{A}$ and $\mathbf{B}$ and the vector of initial state probabilities π sum to one. These constraints are defined explicitly above in Section 2.1. As far as we are aware, no solutions exist for imposing equality constraints in HMMs using the EM algorithm.

This problem is similar to the situation in latent class analysis. In latent class analysis, solutions have been found for some special cases that are of particular interest for psychologists [29]. In particular, in re-estimation of response probabilities in latent class analysis, these probabilities are weighted with the class proportions, i.e., the proportion of subjects that are member of that class. Without changing the optimization

routine, specification of equality constraints in HMMs may be achieved by setting the parameters of interest to be equal after each iteration of the EM algorithm, i.e., after the M-step of the algorithm and before the next E-step. This is done by calculating a weighted average of these parameters. Weighting is necessary because different parameters have different contributions to the loglikelihood. When weighting is not applied, the maximization does not result in ML estimates of the parameters.

The weight factor for parameters that we used is the long term expected proportion of passages through the state with which the parameter is associated. The expected proportion of passage times through each state of the model is different for each state because the transition probabilities from each state to itself are different. For example, there are more passages through state S_1 in our toy model. In general, the expected proportions are computed by solving $\mathbf{p}$ from the following equation:

$$\mathbf{p}\,\mathbf{A} = \mathbf{p},$$

where $\mathbf{p}$ is a probability vector of length n, the number of states in the model, and $\mathbf{A}$ is the transition matrix of the model [**?**,]]Kem60. The vector $\mathbf{p}$ contains the long term probabilities of the process being in each state. For a regular and ergodic Markov chain (not for non-ergodic or cyclic chains), $\mathbf{p}$ is found easily by computing increasing powers of $\mathbf{A}$. The convergence of the series $\mathbf{A}^n$ is fast. For the transition matrix from this model:

$$\mathbf{A} = \begin{pmatrix} 0.888\,0.112 \\ 0.316\,0.684 \end{pmatrix},$$

$$\mathbf{A}^{15} = \begin{pmatrix} 0.73838\,0.26162 \\ 0.73815\,0.26185 \end{pmatrix}$$

As can be seen from this example, the probabilities in A^{15} are converged up to the third decimals. This procedure can be repeated until the desired degree of accuracy is reached.

The justification for using this weighting scheme is best illustrated with an example. Suppose a model has three states. Suppose further that 95% of the observations results from only one of these states. This means that observation parameters belonging to this state contribute more to the loglikelihood than the observation parameters from other states do. Conversely, changing one of these latter observation parameters is unlikely to result in a large change in the loglikelihood since the contribution of these parameters is small.

For the toy model, we imposed an equality constraint on two parameters of the observation matrix, b_{11} and b_{23}. Weighting was done by the expected proportions of passage times. In order to check whether in fact the ML estimates of the parameters were found, instead of weighting, a search algorithm within the EM algorithm was used to optimize the loglikelihood of this model with the equality constraint in place. After each iteration of the EM algorithm, i.e., after the M-step of the algorithm, the maximum likelihood for a range of values of b_{11} and b_{23} is found by the secant method [12]. In this search, the other parameters of the model, i.e. the transition parameters and initial state probabilities, remain fixed. For each step in this search, the likelihood has to be evaluated so this may seem a time-consuming procedure. However, in comparison with an iteration of the EM algorithm, computing the loglikelihood for a fixed set of parameter values is relatively fast. In Table 1 we present the resulting parameter estimates and the corresponding loglikelihood. For comparison, we also present the parameter estimates of the model without the weighting. In Table 1, only the parameter estimates of the observation matrix are provided, because the other parameters hardly differ between the models.

The parameter estimates that result from the weighting method and the search method are identical to the third decimal, but differ slightly thereafter. The loglikelihoods were identical to the fifth decimal number. As can be seen, when weighting is not applied during optimization of the model, both the parameter estimates and the loglikelihood are very different from the other results. An important disadvantage of the search method in fitting equality constraints is that it can not be implemented for general models within the EM algorithm, whereas the weighting method can. Unless parameter estimates need to be more precise than in this example, i.e., up to and including the third decimal, it is therefore easier to use the weighting method for fitting equality constraints. When analyzing relatively small data sets, as will generally be the case in applications in psychology, the differences in parameter estimates are far from significant.

The method of computing weighted averages of parameters within the EM algorithm is, however, not without problems. In latent class analysis, it is known that using this method for particular types of (complex) equality constraints leads to bad estimates [29]. We suspect that similar problems may arise in fitting HMMs. By computing likelihood profiles (see Section 3.4), it is always possible to find out whether the

Table 1
Parameter estimates for model with equality constraint

method	observation matrix parameters	$\log L$		
weight	0.689712	0	0.310288	−886.732
	0	0.310288	0.689712	
search	0.689961	0	0.310039	−886.732
	0	0.310039	0.689961	
no weights	0.639599	0	0.360401	−889.662
	0	0.360401	0.639599	

difference between the maximum likelihood and the likelihood of the fitted model is significant. That is, when the parameter estimates are not the ML estimates, the likelihood profile has negative values in the neighborhood of the estimated value (cf. Section 3.4).

2.5. Likelihood ratio statistic

In both the comparison of models with an identical number of states, and in testing the tenability of specified equality constraints, the likelihood ratio statistic can be used [14]. It is defined as follows [43]:

$$R_c = -2 \log \left[\frac{P(\mathbf{O}|\lambda_c)}{P(\mathbf{O}|\lambda)} \right], \tag{2}$$

where R_c is likelihood ratio statistic, $\log P(\mathbf{O}|\lambda_c)$ is the loglikelihood of the constrained model, e.g. a model with an equality constraint, and $\log P(\mathbf{O}|\lambda)$ is the loglikelihood of the unconstrained model. If the model specified by λ_c is correctly specified, i.e., it is the true model, and it is nested under λ, R_c follows a χ^2 distribution with the df degrees of freedom, where df equals the difference in freely estimated parameters in $\log P(\mathbf{O}|\lambda_c)$ and $\log P(\mathbf{O}|\lambda)$.

For the above fitted model with an equality constraint between b_{11} and b_{23}, df equals 1 because the equality involves two parameters, i.e., one of them is not freely estimated in the constrained model. In this case $R_c = 7.328$, $p < 0.01$. Hence, if $\alpha = 0.05$, the equality constraint results in a model that is worse than the unconstrained model and the constraint should be dropped. However, this decision should ultimately also depend on the power, i.e., on the number of datapoints used to estimate the parameters. The likelihood ratio statistic is also used in computing likelihood profiles which are discussed in Section 3.4.

3. Model selection and model assessment

The EM algorithm estimates parameters of a fully specified model, that is, a model with a fixed number of states and a fixed number of observation symbols. We only fitted a two-state model to the data set from the toy model. In general however, the optimal number of states for a given data set may not be known beforehand. In such a case, it is necessary to fit a number of models with an increasing number of states, to find the model that best describes the data. Of these, the best model is selected by some criterion that weights the model fit, i.e., the loglikelihood, and the economy, the number of parameters of the model. The latter restriction is needed, because for a sequence of observations of length $T = 100$, it is possible to specify a model with 100 states which will have a likelihood of one. Such a model does not reduce the data in any useful sense.

Statistics for model selection are, for example, the Minimum Description Length principle [13], Akaike's Information Criterion [1], the Bayesian Information Criterion [34], and many variants of these [2]. Because these statistics are defined for all kinds of models, simulation studies are necessary to gain insight into their applicability to specific models. In this section, the AIC, the BIC, and a variant of the BIC are compared in selecting fitted HMMs in a simulation study.

3.1. Definitions

When comparing HMMs with a fixed number of states, the likelihood ratio statistic can be used as a selection criterion. In fact, it is implicitly used when selecting the model with the best likelihood from a number of fitted models with equal numbers of states. When comparing HMMs with different numbers of states the situation is different. In particular, HMMs with different numbers of (hidden) states are not nested. The reason for this is the following. In order to arrive at, say a two-state model from a three state model, a number of parameters have to be set to zero, in particular one initial state probability and the transitions to and from that same state. As a consequence the observation probabilities of that particular state are not identified anymore. Moreover, when constraining a three-state model to a two-state model, the functional roles of the

remaining two states may be very different from the states in the three-state model (see [7], for the similar case in latent class analysis). As a consequence of the models not being nested, the likelihood ratio test for comparing model fits can not be used, because the distribution of the likelihood ratio is unknown. Criteria for comparison of non-nested models include Akaike's Information Criterion (AIC) and Bayesian Information Criterion (BIC) criterion. They are defined as follows:

$$\begin{aligned} AIC &= -2\log L + 2n_p \\ BIC &= -2\log L + n_p \log(T), \end{aligned} \tag{3}$$

where L is the likelihood of the fitted model, n_p the number of parameters of the model and T the number of observations used in fitting the model (see [2], for definitions and theoretical foundation of these measures and [24] for an application in latent class analysis). Both criteria consist of two terms: one for the model fit, i.e. the loglikelihood, and a second term for parsimony. The second term is the penalty term, as it increases with the number of parameters used in fitting the model. Usually n_p is taken to be the number of freely estimated parameters. In the present example $n_p = 2 \times (2-1) + 2 \times (3-1) + (2-1) = 7$. The first contribution is from the transition matrix, the second is from the observation matrix and the last is from the initial state vector (because each row of parameters sums to one, one of those parameters is not freely estimated). In fitting HMMs, we also use two variants of these measures that we denote the adjusted AIC, (A-AIC) and adjusted BIC (A-BIC). The adjustment of these measures is in the number of parameters that are counted as freely estimated parameters. In fitting large HMMs, with more than 5 states say, often a large number of both the transition parameters and observations parameters is found to be zero. This is certainly the case when fitting data from finite state automata, as we do in Section 5. In the A-AIC and A-BIC, instead of using n_p as above, we first determine the number of freely estimated parameters and then subtract the number of parameters estimated at zero. Next we add the number of parameters that is estimated at a value of one. The justification for this is that parameters that are estimated at zero do not provide information about the data, i.e. they do not occur in computing the loglikelihood. The parameters that are estimated at one are added because otherwise there would be no difference in the numbers of parameters between, say a four- and a five-state model, with all zeroes and ones in the parameter matrices. In particular, in such models the number of parameters would be zero. A similar procedure for (dis)counting parameters is used in latent class analysis [39].

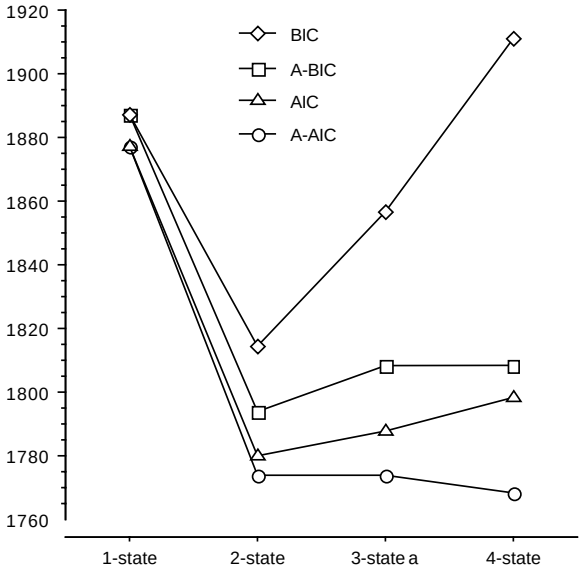


Fig. 1. AIC, BIC, A-AIC and A-BIC selection criteria for models with a different numbers of hidden states.

3.2. Simulation

A 3-state model was optimized for the above described data set from the toy model. Of 100 sets of random starting values, only two converged within the preset limit of 500 iterations. In comparison, all two state models converged within 100 iterations. We should note here, that this is not generally the case. Especially for larger models, the EM algorithm is quite sensitive to different starting values and there can be many local maxima in the likelihood. Therefore we always generate many sets of random starting values and choose the model with the maximum loglikelihood from the (converged) models. In this case, setting the maximum to 2000 iterations improved the situation somewhat, 4 of 100 3-state models converged. Similarly, 100 4-state models were fitted on the same data set (the maximum number of iterations was again set to 2000) of which 4 converged. In Fig. 1 the resulting values of AIC, BIC, A-AIC and A-BIC are plotted for models including one to four states.

According to the AIC, BIC and A-BIC criteria, the two-state model is the best model. The A-AIC is lower for the three-state model and even lower for the four-state model, though the two-state model is the correct model. Also in examples with larger models, we noted that the A-AIC is too liberal in selecting the correct model. With larger models and many datapoints, we have observed that the A-BIC performs very well. When fitting large models, the AIC tends to select even

larger models, whereas the BIC (especially with many data points) tends to select models that have less states than the true model has. More importantly, the A-BIC can also be used for choosing between models with equal numbers of states. Although in general the likelihood suffices in selecting models with equal numbers of states, in cases where many parameters are estimated at zero, it could be argued that resulting models are not nested. For example, when, in fitting two seven-state models, different parameters are estimated at zero in the resulting models, these models are no longer properly nested. As a consequence, the likelihood ratio statistic can not be used. In such a case the BIC does not discriminate between models, because when both models are seven-state models their theoretical numbers of freely estimated parameters are equal. Hence, model selection using the BIC in this case reduces to choosing the model with the largest loglikelihood.

3.3. Assessing model fit: Prediction errors

The AIC, BIC and A-BIC measures work well for selecting models. They do not however, provide information about the absolute fit of the model. That is, they can not be used to assess whether a fitted model is an adequate probability model for the data. A measure that can be used for this purpose is a prediction error statistic based on occurrences of n-tuples in the data. For example, the fitted two-state model predicts a certain (relative) frequency for the occurrence of the triples 123, 322 etc., which can be compared with the data. The prediction error measure P_ϵ used here is a Pearson χ^2 and is defined as follows:

$$P_\epsilon = \sum \frac{(F_O - F_E)^2}{F_E},$$

where F_O are observed frequencies, F_E are expected frequencies based on the model and the sum runs over all n-tuples of interest. See for example [46] for χ^2 measures of goodness-of-fit for contingency tables. See [8] for a discussion of similar measures in latent class analysis. For $n = 5$ the sum runs over $3^5 = 243$ cells since there are three observation symbols in these data. Theoretically, for a correctly specified true model, P_ϵ follows a χ^2-distribution with $df = k - n_p - 1$ degrees of freedom, where k is the number of frequencies over which the sum is computed, and n_p is the number of free parameters in the model. If any of these assumptions is false, P_ϵ follows a non-central χ^2-distribution. In Table 2, the values of P_ϵ, df and p-values arte given for the 2-state model fitted on the data set from the toy model. P_ϵ is given for 2-tuples to 6-tuples of observed sequences.

Table 2
Prediction errors

tuple	P_ϵ	df	p-value	p-value (b)
2	3.43	2	0.178	0.26
3	17.64	20	0.612	0.44
4	78.81	74	0.329	0.18
5	208.07	236	0.905	0.62
6	625.72	722	0.996	0.66

The third column has the df's, which are the theoretical values of df. As can be seen from the table, the p-values in column four of the table, are very high, indicating that the model captures the frequency information in the data very well. For two reasons, there may be an error in the number of degrees of freedom. First, there are dependencies between the parameters. Rabiner [33] shows that for a particular two-state model, with constrained parameter matrices, there is a structural dependence between the parameters. For larger models, with many more parameters, this is certainly also the case. Second, there are dependencies between the cells from which P_ϵ is formed. In general, the frequencies of different sequences of symbols are dependent on each other. For example, if the frequency of 12 is very low, the frequencies of 122 and 123 tend to be low as well, when compared with other triplets of symbols. For these reasons we also computed bootstrapped p values [22]. For an introduction to the principles of bootstrapping see [10]. We generated 100 data sets and fitted a two-state model on each data set. The bootstrapped p-values are now computed as the proportion of models for which P_ϵ of the originally fitted model is smaller than the bootstrapped models. As can be seen in the fifth column of Table 2, these p-values, $p(b)$, are substantially lower than the p-values based on the theoretical values of df except for the 2-tuples.

3.4. Confidence intervals/standard errors

Computing confidence intervals or standard errors is an important part of establishing that a model is adequate. Large confidence intervals may be indicative of identification problems of the model. The standard way of computing confidence intervals is by calculation of the Hessian, the matrix of second partial derivatives of the loglikelihood to the parameters. The diagonal of the inverted Hessian provides the variances of the parameters. Visser et al. [41] have shown that in fitting timeseries with HMMs, with the length T of the series over 100, it becomes computationally unfeasi-

Table 3
Bootstrapped standard errors

parameter	value	mean	std error	t-ratio
a_{11}	0.8882	0.8884	0.0172	51.65
a_{21}	0.3155	0.3237	0.0358	8.81
b_{11}	0.7085	0.7071	0.0278	25.50
b_{22}	0.3841	0.3876	0.0453	8.48

ble to compute the Hessian. Therefore, they compare three other methods of computing confidence intervals for HMM parameters: the bootstrap, likelihood profile and the finite differences approximation to the Hessian. The latter method is very sensitive to round-off errors, to differences in machine precision, and should therefore be used with care.

In likelihood profiling [38,26], a parameter is fixed at a series of values just off its ML estimate and the other parameters are re-estimated. The plot of the fixed parameter values against the likelihood ratio of the refitted models is the likelihood profile. Using likelihood ratio testing, the confidence interval can be determined. That is, the fixed values of the parameter, below and above its ML estimate value, that result in a likelihood ratio of 3.84, provide the 95% confidence interval.

In bootstrapping, a data set is generated using the fitted model and a new model, with the same number of states as the fitted model, is optimized for these data, resulting in a new set of parameter values. The standard deviations of the resulting distributions of the parameters are the approximate standard errors of the parameters [9,10]. In Table 3 the bootstrapped standard errors are given along with the mean of the bootstrapped distribution of the parameter. Note that only standard errors of four parameters are given. Some parameters are estimated at zero or one, and hence have no standard errors. The standard-errors of the other non-zero parameters are identical to those parameters of which they are the complement. For example, the standard error of a_{12} is identical to the standard error of a_{11} because they sum to one.

As can be seen from the standard errors of the parameters, the parameters are all significantly different from zero, i.e., their confidence intervals do not include zero. The t-ratio, which is computed as the ratio of the standard error and the parameter value, is also provided. It is used to check whether parameters are significantly different from zero using the rule of thumb that it should be larger than two, when α equals 0.05. For a more precise statistic to test whether parameters are different from zero, the likelihood ratio statistic R can be used (cf. Section 2.5), which in this case is defined as $R = -2 \times (\log L_0 - \log L_f)$, where L_0 is the likelihood of the model that results from setting the parameter of interest to zero and L_f is the likelihood of the fitted model. This statistic has a χ^2 distribution with $df = 1$, if the parameter is zero in the true model. Note that for computing this likelihood ratio statistic, the parameter of interest has to be set to zero, and the model has to be re-estimated. Hence, for large models this can be a time consuming procedure. For the parameters in this model, the likelihood ratio statistics are infinite for the transition matrix parameters and for parameters b_{11} and b_{22}. That is, setting any of these parameters to zero, leads to an inadmissible model with a likelihood of zero and a loglikelihood of $-\infty$. Hence, this provides an extra confirmation that these parameters are significantly different from zero. Parameters b_{13} and b_{23}, have associated likelihood ratios of 60.99 and 80.22 respectively. Using an α equal to 0.05, it follows that these parameters are also significantly different from zero.

4. Concept identification

Markov models were used frequently in psychology in the 1970's and 1980's [45]. Applications of Markov models included paired-associate learning [4] concept-identification [32], forgetting [20], and conservation learning [3]. Judging by the large number of specialized programs for different applications – Markov-forget for models of forgetting [20], Markov-count for 2-stage learning [19], an SAS module for testing homogeneity in Markov response sequences [11] – a comprehensive framework for parameter estimation seemed to be lacking. In many applications moment-estimators of the parameters are used [45]. For example, parameters are estimated on the basis of the mean number of errors until a certain criterion is reached or the number of errors on the first trial, the second trial et cetera. In fitting simple models, such as the all-or-none model of paired-associate learning, this method works well. For more intricate models, however, maximum likelihood (ML) estimation of parameters is more efficient and powerful. Brainerd [5] used ML estimation for obtaining parameter estimates of a three-state Markov model for memory development. Adopting the framework of hidden Markov models for parameter estimation for these models has many advantages. Maximum likelihood parameter estimates are easily obtained, model selection statistics are available and likelihood ratio statistics are available for testing whether constraints on parameters are in agreement with the data. More-

over, the HMM framework provides the possibility of explorative model fitting. We illustrate these advantages using data from a concept identification experiment. In particular, we discuss model selection, fitting and testing linear constraints between parameters, exploratory and confirmatory approaches to model fitting and goodness-of-fit tests.

4.1. Experimental data

In concept identification tasks, subjects are typically presented with two stimuli, that differ in various dimensions, from which they have to choose. After a choice is made, feedback is given on their correctness of the choice. From a series of such trials, subjects have to deduce the features of the stimuli that identify the concept. The stimuli that were used in this experiment are shown in Fig. 2.

In the experiment, two of the four stimuli shown in Fig. 2 were presented on a computer screen. Subjects' task was to choose one of them and identify the intended concept. Subjects were required to respond quickly. They received a financial reward for responding within the preset time. When subjects responded outside the time limit, feedback was withheld at that trial, and one Dutch guilder was subtracted from their financial rewards. At each trial, subjects were told whether the chosen stimulus was correct or incorrect. To succeed, subjects needed to find out which concept defined the correct stimuli. In this case, the set of concepts to choose from is 'small', 'big', 'circle' or 'triangle'. Sixty two subjects participated in the experiment, producing 62 sequences of responses scored as correct or incorrect. Subjects were said to have identified the concept when the last ten responses were correct. Once this condition was satisfied, the experiment was stopped. On average, subjects needed 22.3 trials before they learned the concept. Usually, in such an easy task, less trials are needed to identify the correct concept. However, because of the time pressure applied, subjects did not have much time to consider their best choice. We compare two candidate models of concept identification data, the all-or-none model, and the win-stay/lose-shift model or concept identification model. To check whether other models may describe the data better, we also fitted unconstrained models and included those in the analysis.

4.2. Model fits and constraints

The all-or-none model has two states, whereas the concept identification model has three states. Both models have a learned state, in which subjects are when they have mastered the task [45]. In this state, the probability of producing a correct answer is one or close to one. In order to accomodate for the possibility that subjects make an error, say due to a lapse of concentration, even though they have mastered the task, the probability of making an error in the learned state is estimated instead of being set to zero. The all-or-none model supposes that learning is an all-or-none process: either subjects have mastered the concept or they have not. If they have not, the probability of producing a correct answer is hypothesized to be 0.5, because there are two possible alternatives. This is the guessing probability and the knowledge state that underlies this is called the guessing state. In the fitted model, the guessing probability was estimated instead of being fixed at 0.5. In Fig. 3(a) this model is represented graphically. Once subjects enter the learned state, they can not leave it, and hence the transition probability a_{ll} of remaining in the learned state is one. The probability of learning, α, of moving from the guessing state to the learned state, was estimated. In Markov modeling this parameter is usually called the learning rate.

In the win-stay/lose-shift model or concept identification model ('ci model' henceforth), an extra assumption is introduced into the model. This assumption is that learning is an hypothesis testing process and that learning only occurs after an error has been made [45]. Just as the all-or-none model, the ci model has a learned state, which represents the knowledge state of subjects who have mastered the concept. In the learned state, the probability of making an error is zero or close to zero. In addition, there are a guessing state and an error state. Subjects remain in the guessing state as long as they are producing correct answers, but fail to identify the concept. They move to the error state, when they have made an error. When in the error state, subjects choose a new hypothesis about the concept to be learned, which is informed by their last error. The ci model is depicted in Fig. 3.

The parameters of the ci model are constrained in the following way. First, the probability of guessing the correct hypothesis before the first trial is 0.5 times the learning rate. Learning only occurs after an error. An error implies that only half of the possible hypotheses remain to be chosen from, since the other half is inconsistent with the error just made. As a result, when the

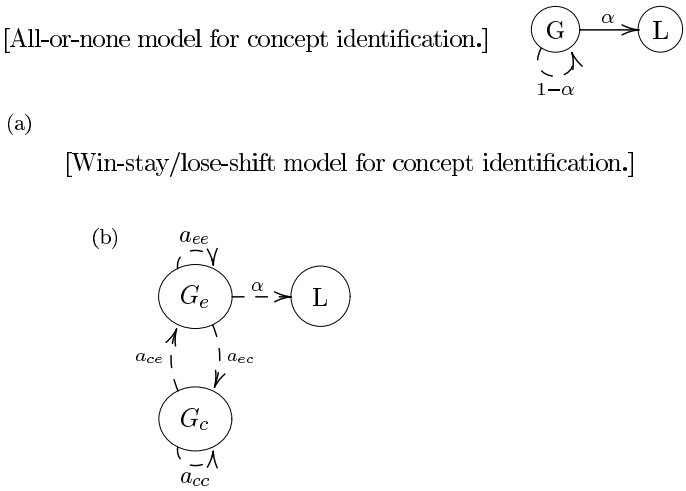


Fig. 3. The all-or-none model and the win-stay/lose-shift model for concept identification

learning rate is α, the initial probability of the learned state is $\alpha/2$. This constraint is fitted in the following way. After each iteration in the EM algorithm, the optimal likelihood is found by varying α and the parameters that systematically depend on it. The value of α that maximizes the likelihood is then entered into the next iteration of the EM algorithm. It is important to note here that in this search, only α and the parameters that depend on it vary. The parameters that depend on α are in the same row of the transition matrix and the initial state probabilities. The other parameters in the model remain fixed, which makes the search very fast to compute. By computing the likelihood profile of α, it is easily verified that indeed the ML estimate is found (cf. Section 3.4).

To be able to better evaluate the confirmative model fits, several unconstrained models were fitted as well. A number of models with 2, 3, and 4 states were fitted. Fitting was done using 100 sets of starting values for each n-state model. In Table 4, parameter estimates for all the models are presented, that is, the all or none model (all), the ci model (ci) and exploratory two- and three-state models, denoted exp 2 and exp 3, respectively. In columns two to four the estimated parameters are listed, $\mathbf{A}$ gives the transition matrix, p_c is the probability for a correct answer in each state and π are the initial state probabilities.

We have left out the four-state explorative model because it does not fit any better than the exp 3 model. Table 4 shows that the all-or-none model and the unconstrained model (exp 2 in the table) are very similar. The exp 2 model clearly has a learned state in which the probability of a correct answer is very high (0.948) and a guessing state in which this probability is close to 0.5 (0.569). The exp 2 model has only one additional free parameter which the all-or-none model does not have which is a_{12}, i.e., the transition from the learned state back to the guessing state. In fact, setting this parameter to zero and reestimating the other parameters results in the all-or-none model. The likelihood ratio statistic for this model constraint is equal to $R = -2\times(\log L_{all} - \log L_{exp2}) = -2\times(-669.574 + 667.685) = 3.778,\ df = 1,\ p = 0.0519$. It follows that the parameter is not significant. As a consequence the exp 2 model reduces to the all-or-none model.

The exp 3 model, which is the result from exploratory fitting of three-state models, has clearly identifiable states. State 1 is the learned state with $p_c = 0.940$. The other two states can both be interpreted as guessing states, but with a different learning rate, i.e., a different transition probability to the learned state. States 2 and 3 are best interpreted as guessing states for two types of persons, slow learners and fast learners. State 2 has a learning rate of 0.233, which is comparable to the ci model, whereas state 3 has a learning rate of 0.034. It might be suspected that this latter parameter is non-significant. The likelihood ratio statistic assoociated with setting the parameter to zero is 12.83 which makes the parameter highly significant. Also, the interpretation for the resulting would be very different from the one given above: instead of having a group of slow learners we would have a group of non-learners represented by the disconnected state. Discussion of the ci-model is deferred until after presentation of the goodness-fo-fit measures.

In Table 5, the goodness-of-fit measures of these models are provided, except for the exp 2 model, since

Table 4
Models for concept identification: parameter estimates

model		**A**			p_c	π	$\log L$
exp 2	S_1	0.985	0.015		0.948	0	−667.685
	S_2	0.114	0.886		0.569	1	
all	S_1	1	0		0.957	0	−669.574
	S_2	0.088	0.912		0.603	1	
exp 3	S_1	1	0	0	0.940	0	
	S_2	0.233	0.767	0	0.467	0.753	−660.918
	S_3	0.034	0	0.966	0.626	0.247	
ci	S_1	1	0	0	0.945	0.102	
	S_2	0	0.5	0.5	1	0.405	−671.305
	S_3	0.204	0.398	0.398	0	0.493	

it reduces to the all-or-none model. In Table 5, first the loglikelihood $\log L$ is given, then the BIC and the number of free parameters df that is used in computing the BIC, the (bootstrapped) prediction errors P_ϵ for 5-tuples, i.e., P_ϵ is the χ^2 statistic for observed and expected frequencies of sequences of five symbols, $df(P_\epsilon)$ is the theoretical number of df's for P_ϵ, p is the p-value associated with P_ϵ and $p(b)$ is the bootstrapped p-value associated with P_ϵ.

As can be seen from the p-values associated with P_ϵ, all three models are adequate descriptions of the data. The current data set may not be large enough to bring out differences between these models. The bootstrapped p-values are close to one, which may indicate that the models are overfitted. To alleviate this, the models can be further constrained. According to the BIC criterion, the ci model is the best model for these data. As expected, the states of the model are clearly interpretable. The probability correct in the learned state is comparable to the other models, $p_c = 0.945$. The fact that it is not 1 can be explained by imprecision of the subjects. The learning rate $\alpha = 0.204$ is somewhat smaller than expected. Given a total of four possible hypotheses, after having made an error, the probability of choosing the correct hypothesis should be 0.5. Alternatively, when the position of the stimulus is also considered as a possible hypothesis, the total number of hypotheses becomes six and correspondingly α is hypothesized to be 0.33. The 95%- confidence interval of α is $0.165 < \alpha < 0.267$ and 0.33 is not in the interval. A possible interpretation for this is that the hypothesis sampling procedure that subjects engage in is not very efficient. This inefficiency may be due to the time pressure imposed on subjects while performing the task.

We tested the constraint that we imposed on α and the initial state probability for the first state for significance with the likelihood ratio statistic. Fitting the model again without the constraint results in a loglikelihood of -670.569. Using the likelihood ratio statistic R, which equals 1.277 ($df = 1$, $p = 0.258$), we conclude that the constraint is warranted.

4.3. Conclusion

It can be seen from this example that using the HMM framework for estimating parameters for well-known Markov models provides much flexibility in comparing models and in testing the significance of parameter values and constraints. It is therefore much easier to test model assumptions that are provided by theoretical considerations than it is when using standard estimation procedures such as the method of moments. Moreover, a single estimation program can be used for all these applications instead of the specialized programs such as those of [19,20].

5. Implicit learning

HMMs are equivalent to regular grammars. As such, they can be used for describing data of several kinds of experiments, such as inductive learning, systems control and implicit learning experiments. In this section we focus on this latter kind of experiment. In both systems control and implicit learning experiments, often regular grammars are used to generate stimuli (see [15], for an introduction to formal languages). The canonical representations of regular grammars are finite state automata (FSA). In this section we will consider HMMs and FSAs as representations of regular grammars.

HMMs can be represented as FSAs, by shifting from an edge representation to a vertex representation, that is, instead of having labeled states, FSAs have labeled arcs (see e.g. [25], for different representations of FSAs). The example data that we will present are from an implicit learning experiment in which an FSA is used to generate stimuli. In Fig. **??** both an HMM and an FSA representation of the same grammar are depicted.

Table 5
Models for concept identification: goodness-of-fit measures

model	$\log L$	BIC	df	$P_\epsilon(5)$	$df(P_\epsilon)$	p	$p(b)$
all	−669.574	1375.30	5	25.43	27	0.55	0.97
exp 3	−660.918	1401.36	11	27.81	21	0.15	0.97
ci	−671.207	1371.33	4	24.02	28	0.68	0.99

In Fig. **??** the grammar is represented as HMM. The generation of strings in the HMM is very similar to the procedure with the FSA: starting from a particular state, one of the arcs leaving that state is chosen and the letter in that state becomes a symbol in the sequence. Again it can be seen that ADBC is a legal sequence as it is in the FSA, but BAC is not; none of the nodes with labels B, A and C are connected in that particular order. This FSA is depicted in Fig. 4. Sequences are generated using this FSA by moving from state to state, starting in state # 1/7 until state # 1/7 is reached again. For example, the sequence ADBD is a grammatical sequence that passes through states 1, 3, 4, 6 and 7.

5.1. Experimental data

Sequence learning has become the paradigm of choice in studying implicit learning as witnessed many recent papers using that paradigm [17,16,36,35,37]. Typically, in sequence learning, subjects are presented with a sequence of stimuli that, unbeknownst to them, is manipulated such that the order is not random. Subjects are required to reproduce the stimuli by typing a key that corresponds to the presented stimulus. Subjects' performance is measured by reaction times (RT). The typical result is a larger decrease in RTs for subjects in the experimental condition than for subjects in the control condition where the sequence of stimuli is random. From this result, we can conclude that subjects use knowledge of the sequential structure to respond faster than subjects in the random condition. When asked, subjects are usually not aware of the fact that the stimulus sequence is manipulated. Hence, the knowledge that subjects have is said to be implicit.

As an illustration, we use data from an experiment by [42]. In their experiment, there were four different stimuli and the sequence of stimuli that we used was generated by the FSA in Fig. 4. There were a total of 12000 trials that were divided into 24 blocks of 500 trials each. Each block of 500 trials was divided into runs of trials that subjects had to reproduce. Alternated with the standard RT trials, subjects were presented with free generation trials. At those trials, instead of reproducing the stimulus, subjects were required to guess where they thought the next stimulus would appear. In each block of 500 trials, there was an average of 95 such trials, in runs from 3 to 7 trials. These are the data that we are interested in here. Blocks 6, 12, 18 and 24 were blocks with randomly ordered stimuli to provide a baseline for the RT analysis. Those blocks are not analyzed here.

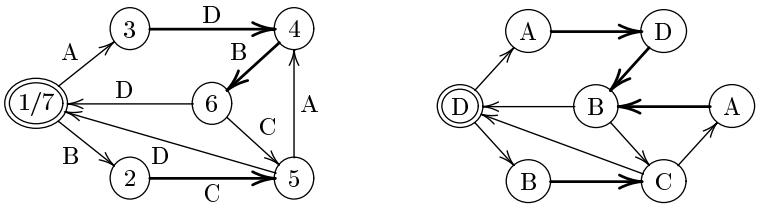
The generation trials are seen as a direct expression of knowledge from the grammar that subjects gained during the sequence learning experiment. It is therefore interesting to compare this knowledge with the grammar in such a way that it is possible to quantify how much subjects have learned. This is accomplished by fitting HMMs to the data generated by subjects and comparing these with the HMM representation of the grammar. In this case, the similarity or dissimilarity between models is of interest, which is quantified by a distance measure.

5.2. Distance between models

Comparing fitted HMMs in general can be done by the model selection criteria that were described in Section 2. Model selection criteria can be very helpful in selecting the best model. However, in the case of implicit learning data, it would be interesting to compare a fixed model, the grammar, with the fitted model and model criteria are not very helpful in this case. For comparing models in this sense, i.e., for determining the similarity or dissimilarity between models, Rabiner [33] provides a distance measure. In the implicit learning experiment, we expect free generation data to become more like the grammar as learning continues. Hence, we expect the distance between fitted models and the grammar to decrease. The distance measure for comparing models for this purpose, is computed as follows [33, p. 281]:

$$\begin{aligned} &D(\lambda_f, \lambda_t) \\ &= \frac{-\log P(O_T|\lambda_f) + \log P(O_T|\lambda_t)}{T}, \end{aligned} \tag{4}$$

where $O_t, t = 1 \ldots T$ is a sequence generated by the true model, that is, the grammar, T is the length of the sequence, $\log P(O_t|\lambda_t)$ is loglikelihood of the sequence O_t given the parameter values of the true model λ_t (i.e. the HMM representation of the grammar) and



(a) Finite state automaton used to generate strings for the experiment.

(b) Hidden Markov representation of the same FSA.

Fig. 4. Finite state automaton and hidden Markov model for the same grammar. In both figures the thickness of the arrows represents the probability of going from one state to the next. The probabilities of arcs leaving a state sum to one. See the text for further details.

$\log P(O_t|\lambda_f)$ is loglikelihood of the sequence O_t given the parameter values of the fitted model λ_f. Note that we use the real values of the loglikelihood here which are all negative.

This distance measure expresses how well the fitted model can describe data that are generated from the grammar in comparison with how well the grammar itself does so. The distance measure can be interpreted as the cross-entropy between the models (see e.g. [44], for an introduction to entropy and information distance measures). Cross-entropy or entropy in turn can be interpreted as a measure of coding-efficiency. When, in comparison with another model, a model has a lower entropy, this means that the latter model represents a more efficient coding of the data than the first model. Hence, positive values of D indicate that the fitted model λ_f is less efficient in coding data from the true model λ_t than the true model itself is [25,23]. Conversely, negative values of D indicate that the fitted model represents a more efficient encoding of the data than does the original model.

Note that this distance is not symmetric. It measures the distance from the true model to the fitted model but not vice-versa. In general, it can be made to be symmetric by computing [33]:

$$D_s(\lambda_1, \lambda_2) = \frac{D(\lambda_1, \lambda_2) + D(\lambda_2, \lambda_1)}{2}$$

This can only be done however, when the values $\log P(O_i|\lambda_j)$for all combinations of $i, j = 1, 2$ exist. This is only the case when data produced by the one model can be fitted by the other and vice versa. If for example, the model given by parameter values λ_2 allows a transition AB but λ_1 does not, then $\log P(O_2|\lambda_1)$ becomes $-\infty$. Computing a distance with this likelihood then is not possible.

5.3. Fitting hidden Markov models to generation data

In implicit learning, generation data are often analyzed by counting numbers of n-tuples that subjects generate. For example, triples of generated sequences are counted and compared with a control group or control condition in which subjects were presented with random sequences of stimuli [36,31]. Alternatively, n-tuples may be compared with chance levels of generating them [6]. Using HMMs, analysis of n-tuples is done simultaneously. That is, instead of analyzing pairs, triples, quadruples, et cetera, separately, HMMs provide a way of finding a model that best describes all these n-tuples simultaneously.

The goal of fitting HMMs on implicit learning data is to compare them with the grammar that was used in the experiment. Hence, it is important that the distance defined in Eq. (5) is computable. In computing this distance, we use a data set generated by the grammar. For the distance to be computable, the likelihood of this data set under the fitted model should exist. This is achieved by using the HMM representation of the grammar as a starting point. When the fitted models contain all the state transitions that also occur in the grammar, the distance is surely computable. Because subjects do not only generate grammatical sequences, this model has to be adapted such that it can accommodate arbitrary sequences. Starting with the transition probabilities from the grammar, this is achieved by setting all other transitions to random values different from zero. The observation probabilities of the model remain fixed at the values from the grammar. In this way, we ensure that the fitted models can generate grammatical sequences, while at the same time allow for arbitrary sequences of subjects to be modeled.

The implicit learning data consist of sequences of trials generated by the subjects. The generation data from the subjects were pooled for each grammatical experimental block, resulting in a total of 20 data sets to be analyzed. Each of these data sets consists of 152 sequences of trials, varying in length from 3 to 7 symbols resulting in data sets of approximately 800

data points.[1] HMMs were fitted to these sets in the following way. We generated 300 sets of starting values for each data set, according to the scheme described above. These models were optimized and the A-BIC criterion was used to select the best model. In this case, the A-BIC is an appropriate selection criterion because all the models have equal numbers of states and hence equal numbers of freely estimated parameters. However, fitted models may differ in the numbers of parameters in the transition matrix that are estimated at zero. Hence these models are not nested and the likelihood ratio can not be used to select the best model. Using either AIC or BIC results in identical models as using the likelihood ratio because all fitted models have equal numbers of freely estimated parameters. Hence, we used the A-BIC criterion for model selection. For the resulting 20 models we computed their distance to the grammar.

The distances decrease from 0.619 in the first experimental block to 0.371 in the last block. A regression analysis of the distances with the block numbers as predictor shows that the distances decrease significantly due to training ($R^2 = 0.636, p < 0.0001$). Our goal here was to devise a method of quantifying how much knowledge is expressed in a generation task of a sequence learning experiment. The distance measure provided here does exactly that. For a complete description of this experiment, more elaborate analyses of the generation data, analyses of the RT data and a discussion of the theoretical implications, see [42]. The distance measure provided in Eq. (5) can also be used more generally for comparing similarity or dissimilarity between models.

6. Summary and conclusion

Markov and latent or hidden Markov models have been popular in psychology for a long time. They have proven to be very flexible models in describing all kinds of data, mostly in the area of memory and learning [45]. In the 1980s, however, there is a drop in the number of articles published on the subject. This may be in part due to the lack of a comprehensive framework for estimating parameters of such models. Hidden Markov models, and the associated maximum likelihood estimation procedure, provide such a framework. However, latent or hidden variable models present extra challenges. Notably, model selection and the assessment of goodness-of-fit, are notorious problems in fitting latent variable models.

In Section 3, we proposed and compared several candidate measures for model selection. We found both AIC and BIC to work well in a simulation study. However, for comparing larger models, we use an adjusted BIC, because in such models many parameters tend to be estimated at zero. Hence, using the number of freely estimated parameters in computing the BIC overestimates the number of parameters that is used in computing the likelihood of a given data set. We also introduced and compared two methods for fitting HMMs with equality constraints imposed on the observation parameters which to the best of our knowledge is novel in the context of fitting HMMs. We showed that the weighted average method is adequate and can be used for general equality constraints, whereas the more laborious search method can be used for fitting HMMs with general linear constraints imposed on the parameters.

In two illustrative examples, we applied model selection criteria and likelihood ratio testing of constraints. Moreover, we used the prediction error measure that we introduced, in the concept identification example. In both examples we showed new ways of analyzing data which provide insight into theoretical issues in the respective areas of investigation.

[1] In our experience these data sets are about the size needed to get reliable parameter estimates, certainly in the case of explorative model fitting. With smaller data sets there is a real danger of selecting models that have less states than the true model. In the setting described here, involving confirmatory model fits, data sets may be as small as 100 data points, however only at the cost of large standard errors.

References

[1] H. Akaike, Information theory and an extension of the maximum likelihood principle, in: *Second international symposium on information theory*, B.N. Petrov and F. Csaki, eds, Academiai Kiado, Budapest, 1973, pp. 267–281.

[2] H. Bozdogan, Akaike's information criterion and recent developments in informational complexity, *Journal of Mathematical Psychology* **44**(1) (2000), 62–91.

[3] C.J. Brainerd, Markovian interpretations of conservation learning, *Psychological Review* **86**(3) (1979), 181–213.

[4] C.J. Brainerd, Developmental invariance in a mathematical model of associative learning, *Child Development* **51**(2) (1980), 349–363.

[5] C.J. Brainerd, Three-state models of memory development: A review of advances in statistical methodology, *Journal of Experimental Child Psychology* **40** (1985), 375–394.

[6] A. Cleeremans and J. L. McClelland. Learning the structure of event sequences. *Journal of Experimental Psychology: General*, 120:235–253, 1991.

[7] C.C. Clogg, Latent class models, in: *Handbook of statistical modeling for the social and behavioral sciences*, G. Arminger, C.C. Clogg and M.E. Sobel, eds, Plenum press, New York, 1995.

[8] L.M. Collins, P.L. Fidler, S.E. Wugalter and J.D. Long, Goodness-of-fit testing for latent class models, *Multivariate Behavioral Research* **28**(3) (1993), 375–389.

[9] B. Efron and R.J. Tibshirani, Bootstrap methods for standard errors, confidence intervals and other measures of statistical accuracy, *Statistical Science* **1** (1986), 54–77.

[10] B. Efron and R.J. Tibshirani, *An introduction to the bootstrap*, Monographs on statistics and applied probability 57, Chapman & Hall, New York, 1993.

[11] S.V. Faraone, A Statistical Analysis System (SAS) computer program for markov chain analysis, *Journal of Psychopathology and Behavioral Assessment* **8**(4) (1986), 367–379.

[12] P.E. Gill, W. Murray and M.H. Wright, *Practical optimization*, Academic Press, New York, 1981.

[13] P. Grünwald, Model selection based on minimum description length, *Journal of Mathematical Psychology* **44** (2001), 133–152.

[14] J.A. Holt and G.B. Macready, A simulation study of the difference chi-square statistic for comparing latent class models under violation of regularity conditions, *Applied Psychological Measurement* **13**(3) (1989), 221–231.

[15] J.E. Hopcroft, R. Motwani and J.D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, (2 ed.), Addison Wesley, Boston, 2001.

[16] L. Jimenez and C. Mendez, Which attention is needed for implicit sequence learning? *Journal of Experimental Psychology: Learning, Memory and Cognition* **25**(1) (1999), 236–259.

[17] L. Jimenez and C. Mendez, Implicit sequence learning with competing implicit cues, *The Quarterly Journal of Experimental Psychology* **55**(4) (2001), 345–369.

[18] J.G. Kemeny and J. Snell, *Finite Markov chains*, Van Nostrand, Princeton, 1960.

[19] J. Kingma and J. Reuvekamp, Markov count: a program for computing the learning statistics of two-stage markov learning experiments, *Educational and Psychological measurement* **47**(1) (1987), 89–98.

[20] J. Kingma and J. Reuvekamp, Markov-forget: A package for parameter estimation and hypothesis testing of 5, 6, 7, 8, 9 and 10-parameter two-state forgetting models, *Educational and Psychological measurement* **47**(3) (1987), 673–678.

[21] W. Kintsch and C.J. Morris, Application of a Markov model to free recall and recognition. *Journal of Experimental Psychology* **69**(2) (1965), 200–206.

[22] R. Langeheine, J. Pannekoek and F. VandePol, Bootstrapping goodness-of-fit measures in categorical data analysis. *Sociological methods and research* **24**(4) (1996), 492–516.

[23] M. Li and P. Vitányi, *An introduction to Kolmogorov complexity and its applications*, (2 ed.), Graduate texts in computer science, Springer, New York etc., 1997.

[24] T.H. Lin and C.M. Dayton, Model selection information criteria for non-nested latent class models. *Journal of Educational and Behavioral Statistics* **22**(3) (1997), 249–264.

[25] D. Lind and B. Marcus, *Symbolic dynamics and coding*, Cambridge University Press, 1995.

[26] W.Q. Meeker and L.A. Escobar, Teaching about approximate confidence regions based on maximum likelihood estimation, *The American Statistician* **49**(1) (1995), 48–53.

[27] G.A. Miller, Finite Markov processes in psychology, *Psychometrika* **17** (1952), 149–167.

[28] G.A. Miller and N. Chomsky, Finitary models of language users, in: *Handbook of mathematical psychology*, (Chapter 13), R. Luce, R.R. Bush and E. Galanter, eds, Wiley, New York, 1963.

[29] A. Mooijaart and P. van der Heijden, The EM algorithm for latent class analysis with equality constraints, *Psychometrika* **57**(2) (1992), 261–269.

[30] R.I. Nicolson, Shades of all-or-none learning: A stimulus sampling model, *British Journal of Mathematical and Statistical Psychology* **35**(2) (1982), 162–170.

[31] P. Perruchet and M.-A. Amorim, Conscious knowledge and changes in performance in sequence learning: Evidence against dissociation, *Journal of Experimental Psychology: Learning, Memory and Cognition* **18**(4) (1992), 785–800.

[32] J.G.W. Raaijmakers, A general framework for the analysis of concept identification tasks. *Acta Psychologica* **49** (1981), 233–261.

[33] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, *Proceedings of IEEE* **77**(2) (1989), 267–295.

[34] G. Schwarz, Estimating the dimension of a model, *Annals of Statistics* **6** (1978), 461–464.

[35] C.A. Seger, Two forms of sequential implicit learning, *Consciousness and Cognition: An International Journal* **6**(1) (March 1997), 108–131.

[36] D.R. Shanks and T. Johnstone, Evaluating the relationship between explicit and implicit knowledge in a sequential reaction time task, *Journal of Experimental Psychology: Learning, Memory and Cognition* **25**(6) (1999), 1435–1451.

[37] D.R. Shanks and P. Perruchet, Dissociation between priming and recognition in the expression of sequential knowledge, *Psychonomic Bulletin & Review*, In press.

[38] D.J. Venzon and S.H. Moolgavkar, A method for computing profile-likelihood-based confidence intervals. *Applied Statistics* **37**(1) (1988), 87–94.

[39] J. Vermunt, *IEM: A general program for the analysis of categorical data [Program manual]*, 1997.

[40] I. Visser, *Markovfit 1.6 [Computer program manual]. Program for fitting hidden Markov models for categorical data*, 2001, Copies of manual and source codes available from ingmar@dds.nl.

[41] I. Visser, M.E.J. Raijmakers and P.C.M. Molenaar, Confidence intervals for hidden Markov model parameters, *British journal of mathematical and statistical psychology* **53** (2000), 317–327.

[42] I. Visser, M.E.J. Raijmakers and P.C.M. Molenaar, Associations and dissociations between direct and indirect measures of sequence knowledge, *Submitted for publication*, 2002, Copies available from first author, ingmar@dds.nl.

[43] A. Wald, Test of statistical hypotheses concerning several parameters when the number of observations is large, *Transactions of the American Mathematical Society* **54** (1943), 426–482.

[44] J. Whittaker, *Information Divergence*, (Chapter 4), Wiley series in probability and mathematical statistics, John Wiley, Chichester, 1990.

[45] T.D. Wickens, *Models for Behavior: Stochastic processes in psychology*, W.H. Freeman and Company, San Francisco, 1982.

[46] T.D. Wickens, *Multiway contingency tables analysis for the social sciences*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1989.