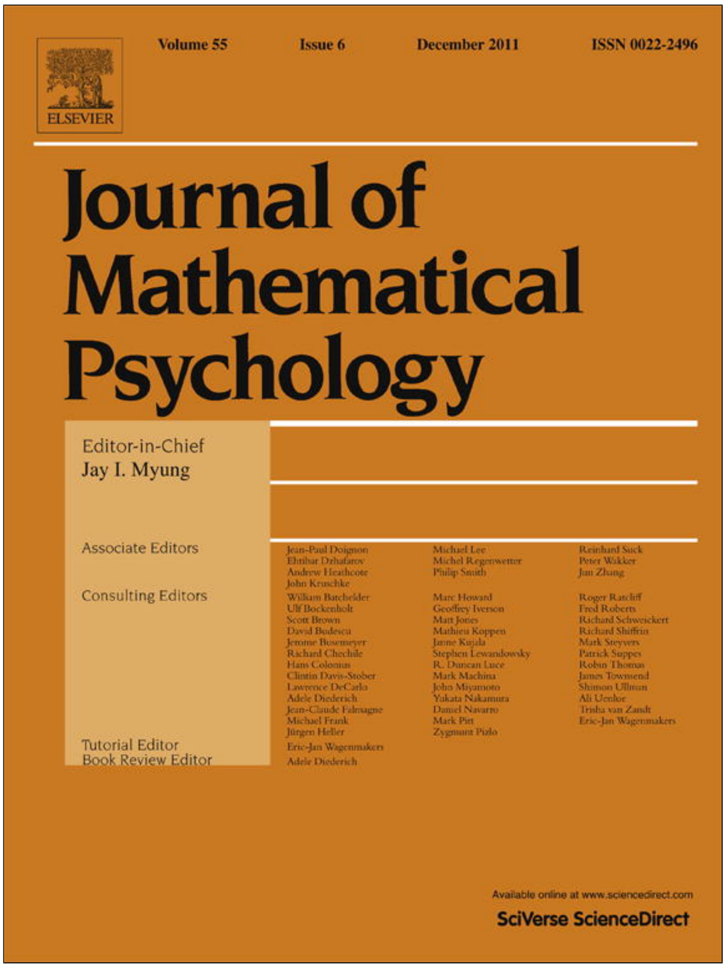
Seven things to remember about hidden Markov models: A tutorial on Markovian models for time series

Article *in* Journal of Mathematical Psychology · December 2011

DOI: 10.1016/j.jmp.2011.08.002

Ingmar Visser
University of Amsterdam



Volume 55 Issue 6 December 2011 ISSN 0022-2496

ELSEVIER

Journal of Mathematical Psychology

Journal of Mathematical Psychology 55 (2011) 403–415

Contents lists available at SciVerse ScienceDirect

Journal of Mathematical Psychology

journal homepage: www.elsevier.com/locate/jmp

Tutorial

Seven things to remember about hidden Markov models: A tutorial on Markovian models for time series

Ingmar Visser

Department of Psychology, University of Amsterdam, Roetersstraat 15, 1018 WB, Amsterdam, The Netherlands

ARTICLE INFO

Article history:
Received 26 December 2010
Received in revised form
27 July 2011
Available online 1 September 2011

Keywords:
Hidden Markov model
Dependent mixture model
Mixture model

ABSTRACT

This paper provides a tutorial on key issues in hidden Markov modeling. Hidden Markov models have become very popular models for time series and longitudinal data in recent years due to a combination of (relative) simplicity and flexibility in adapting the model to novel situations. The tutorial covers the conceptual description of the model, estimation of parameters through maximum likelihood, and ends with an application to real data illustrating the possibilities.

Contents

Psychology studies psychological processes and the usual way of gaining insight into those is by administering surveys or experiments to large numbers of participants. This however may not be the optimal way for studying such processes in the face of individual differences that come about through learning and development (Bower & Trabasso, 1964; Molenaar, 2004). An arguably better way for studying psychological processes that does not rely on (averaging) cross-sectional data is to gather and analyze time series data from individuals. Studying time series data has the advantage of witnessing the psychological processes through time. Hidden Markov models (HMMs) form a very flexible class of models for time series data. HMMs are especially suitable for studying psychological or cognitive processes in which qualitatively different cognitive states unfold over time.

In this paper, I address seven questions about hidden Markov models:

1. What are hidden Markov models?
2. What is Markov about hidden Markov models?
3. What is hidden about hidden Markov models?

E-mail address: i.visser@uva.nl.

0022-2496/$ – see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jmp.2011.08.002

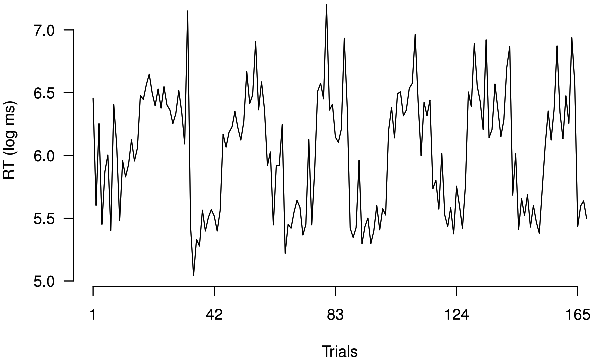


Fig. 1. Response times (RT) for 168 trials of a computerized experiment.

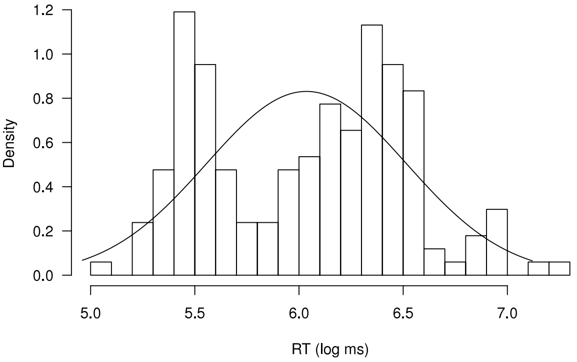


Fig. 2. Observed density of response times overlaid with the density of a Gaussian distribution, with mean equal to 6.04.

4. How to compute the likelihood of HMMs?
5. How to compute the hidden states of HMMs?
6. How to estimate the parameters of HMMs?
7. How to fit HMMs in practice?

The first three questions set the stage for the definition of HMMs and highlight some applications in psychology. The second set of three questions pertain to three classical problems in hidden Markov modeling, known as the evaluation problem, or how to compute the likelihood, the decoding problem, or how to compute the hidden states, and the learning problem, or how to estimate parameters. In the seventh and final section, I illustrate the use of HMMs for a real data set to indicate some of the possibilities. I also provide pointers to software that can be used for estimating HMMs, such that it is not necessary to implement or program the formulas in my answers to questions 1–6 yourself. Sections 1–3 are mostly meant for introducing notation and conceptual understanding of what HMMs are, whereas Sections 4–6 are more technical in nature. Note that it is not necessary to understand all the technical details in Sections 4–6 to appreciate the possibilities for applications and to understand the example models in Section 7. The paper ends with a discussion and some pointers to literature on hidden Markov and related models.

1. What are hidden Markov models?

To fix ideas on what hidden Markov models are, consider the data in Fig. 1; these are (log-transformed) response times from 168 trials of a computerized lexical decision experiment; more details on the experiment are provided in Section 7 below; the data are from Experiment 1 in Dutilh, Wagenmakers, Visser, and van der Maas (2011). The mean of these 168 consecutive response times is 6.04 log ms[1] and the standard deviation equals 0.48. An important question is whether it is appropriate to summarize these data using their mean and standard deviation. A first step to answering that question may be to look at the marginal distribution of the response times (i.e., collapsed over trials), which is provided in Fig. 2.

As can be seen in Fig. 2, the response times deviate considerably from the Gaussian distribution (and from any other uni-modal distribution), and it may be suspected that there are multiple modes in these data. This is the first defining characteristic of hidden Markov models: the marginal distribution of the data, i.e., collapsed over time, is a mixture distribution. The data are drawn from two or more distributions with different parameter values. Another way of saying this is that hidden Markov models have *discrete states* that generate the data. As is common in the literature on mixture models, the states of HMMs are also sometimes referred to as components.

To clarify this further, some notation is useful. Consider time series data denoted by $Y_{1:T} := (Y_1, Y_2, \dots, Y_T)$ for a time series of length T. The data can be response times as in the example, which follow a continuous distribution, but also data with discrete distributions are allowed in HMMs. An HMM consists of state variables $S_{1:T} := (S_1, \dots, S_T)$, and to say that HMMs are discrete implies that the state variables S_t are elements from a finite set $\mathcal{S} = \{1 \dots n\}$ such that we can write $S_t = i, i \in \mathcal{S}$. The state variables $S_{1:T}$ are thus random variables with discrete values. The set $\mathcal{S}$ is called the state-space of the HMM, and n is the number of states of the model. The observations Y_t are dependent on the state variables S_t such that the distribution of Y_t can be written as: $f_i(Y_t) := f(Y_t|S_t = i)$. Because the set $\mathcal{S}$ is finite, this means that the marginal distribution of the data is a mixture distribution with n components:

$$f(Y_t) = \sum_{i=1}^{n} p_i f_i(Y_t), \tag{1}$$

where p_i are the component proportions with the constraint that $\sum_{i=1}^{n} p_i = 1, p_i \geq 0$ and $f_i(\cdot)$ is the conditional distribution of the data in component i. See McLachlan and Peel (2000) for an overview of mixture distribution models.

Going back to the example response time data: if they stem from a hidden Markov model, the marginal distribution should follow a mixture distribution. To confirm this, a 2-component Gaussian mixture model was fitted to the response times. The 2-component model has a BIC of 200.5, compared to BIC= 239.5 for a 1-component model,[2] indicating that the 2-component model is superior. The parameters for the components of the 2-component mixture distribution are $\mu_1 = 5.48$ ($\sigma_1 = 0.126$), and $\mu_2 = 6.31$ ($\sigma_2 = 0.319$), respectively. The component proportions are 0.331 and 0.669, respectively. Hence, summarizing these data with their mean (6.04) and standard deviation (0.48) does not do justice to their nature (Fig. 3).

What then are these discrete states? In applications in psychology, the states of an HMM are assumed to be discrete (cognitive, emotional) states that produce typical behavior. In an application about sleep stages for example, the states of an HMM correspond to various stages such as REM sleep, deep sleep and wakefulness (Flexer, Sykacek, Rezek, & Dorffner, 2002). Hence,

[1] It is common to use log-transformed response times in analyses such that they better conform to the normal distribution.

[2] The BIC (Schwarz, 1978) is a model selection statistic that is frequently used in mixture and latent class modeling to select the number of components of a mixture distribution.

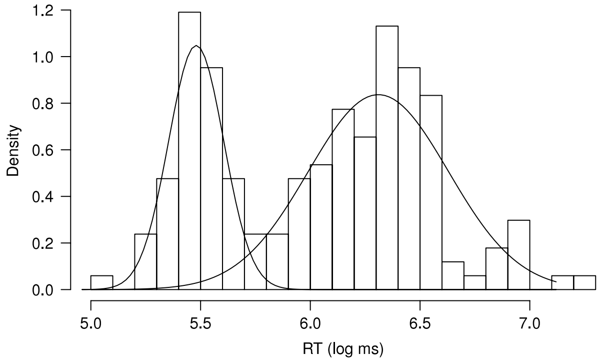


Fig. 3. Observed density of response times overlaid with the normal densities from the fitted 2-component Gaussian mixture distribution.

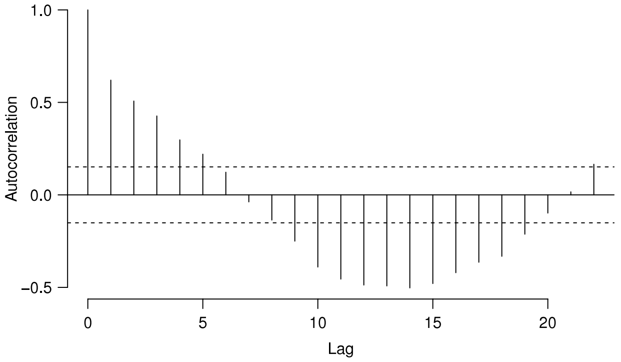


Fig. 4. Autocorrelation function of the response time data shown in Fig. 1; the dotted lines represent the 95% confidence limits.

in this application, the HMM had three states (REM, deep sleep and wakefulness), and each of these is associated with typical observations in the EEG measurements, which form the data $Y_{1:T}$.

In DNA sequence analysis, HMMs are used to align different observations from the same gene. A nucleotide sequence that forms a gene may vary in the population due to insertions, deletions or mutations. In such applications, the states of the HMM represent the nucleotides of a gene, and the observations are noisy versions of the same nucleotides, i.e, where the noise is caused by insertions, deletions or mutations (Krogh, 1998).

In applications in developmental psychology, the states of the HMM often correspond to particular strategies that participants use to solve problems. A famous task in developmental psychology is the conservation of liquid task in which participants have to indicate the expected level of a liquid if it would be poured into another glass with a different width. Participants typically use one of two strategies in indicating the level; the wrong strategy, that younger children often apply, is to indicate the same level as in the other glass, effectively ignoring that the second glass has a different width. Older children and adults typically apply the correct strategy of adjusting the height proportional to the change in width of the glass. The two strategies lead to different behavior, and can be represented by different states in an HMM (Schmittmann, Dolan, van der Maas, & Neale, 2005). See Kaplan (2008) for a review of applications of hidden Markov models in developmental psychology.

In applications in economy, the states of the HMMs can correspond to expansion and recession, and the interest is in studying the dynamics between these (Ghysels, 1994; Hamilton, 1989).

In sum, hidden Markov models are characterized by discrete (hidden) states, which can be interpreted as states in a (cognitive) process which each produce typical behavior. The dynamics of the process, or how the process evolves in time, is also modeled by the hidden Markov model, and it is the topic of the next section.

2. What is Markov about hidden Markov models?

The previous section discussed the discrete nature of hidden Markov models as their first defining characteristic. The second defining characteristic of hidden Markov models concerns the dependence of the data over time. Inspection of Fig. 1 reveals that the response times have serial dependence rather than being independent samples from a number of distributions. Computing the auto correlation function is especially revealing in this case: the lag-1 autocorrelation equals 0.62, whereas the lag-10 auto correlation equals −0.39, with both correlations being significantly different from zero. Fig. 4 displays the full autocorrelation function.

The data in HMMs are thus dependent rather than independent draws from the components of the mixture distribution. Another name for hidden Markov models (HMM) is hence the more directly informative 'Markov-dependent mixture model' (which was coined by Leroux & Puterman, 1992).

The Markov assumption in hidden Markov models pertains to the dependence between the consecutive states S_t, that is, they follow a Markov process, which means the dependence between the states can be expressed as:

$$P(S_t|S_1,\ldots,S_{t-1}) = P(S_t|S_{t-1}). \tag{2}$$

This means that the current state S_t only depends on the previous state S_{t-1}, and not on earlier states. This is a simplifying assumption that is often made. It is not, however, a necessary assumption: when it is loosened, so-called higher order Markov models result, in which the state S_t may depend on multiple previous states (see, e.g., Langeheine & Van de Pol, 2000).

The probabilities $P(S_{t+1}|S_t)$ are commonly referred to as transition probabilities as they govern the transitions between the states of the hidden Markov model. The transition probabilities are denoted by the matrix $\mathbf{A}(t)$ with entries:

$$a_{ij}(t) := P(S_{t+1} = j|S_t = i), \quad i, j = 1, \ldots, n. \tag{3}$$

Row one of the matrix $\mathbf{A}(t)$ hence contains the probabilities of moving from state $S_t = 1$ to state S_{t+1}. As a consequence, each row of $\mathbf{A}(t)$ sums to unity: $\sum_{j=1,\ldots,n} a_{ij}(t) = 1$, for each i. When the transition probabilities are independent of t, which is often the case, $\mathbf{A}$ is used instead of $\mathbf{A}(t)$, and the Markov process that governs the evolution of the hidden states is then said to be *homogeneous*.

An important property that can be derived from the transition matrix is the probability of the process being in a given state in the long run. This probability vector $\mathbf{p}$ is called the stationary distribution of a Markov chain and it can be computed by solving the following equation:

$$\mathbf{p} \cdot \mathbf{A} = \mathbf{p}. \tag{4}$$

The stationary distribution is the left eigenvector for eigenvalue 1 (Kemeny & Snell, 1960). When, for example, we have a transition matrix $\mathbf{A}$ with entries:

$$\mathbf{A} = \begin{pmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \end{pmatrix}, \tag{5}$$

the stationary distribution can be computed by solving the following set of equations:

$$\begin{aligned} 0.8p_1 + 0.4p_2 &= p_1, \\ 0.2p_1 + 0.6p_2 &= p_2, \\ p_1 + p_2 &= 1, \end{aligned}$$

which results in $\mathbf{p} = (2/3, 1/3)$. It can be easily verified that this is correct by inserting this value for $\mathbf{p}$ in Eq. (4); $2/3 * 0.8 + 1/3 * 0.4 = 16/30 + 4/30 = 20/30$, and $2/3 * 0.2 + 1/3 * 0.6 = 4/30 + 6/30 = 10/30$. Note that the stationary distribution only makes sense in homogeneous Markov models as in non-homogeneous models the solution to $\mathbf{p} \cdot \mathbf{A} = \mathbf{p}$ would depend on t.

Next to the transition parameters in $\mathbf{A}$, another set of parameters governs the probability distribution over the states of the HMM, and these are the initial state or prior probabilities $\boldsymbol{\pi}$ defined as:

$$\pi_i := P(S_1 = i), \quad i = 1, \ldots, n. \tag{6}$$

These probabilities determine in which state the process begins at time $t = 1$. Given $\boldsymbol{\pi}$ and $\mathbf{A}$, the probability vectors $P(S_t = 1, S_t = 2, \ldots, S_t = n)$ for $t > 1$ can be computed as:

$$P(S_t = 1, S_t = 2, \ldots, S_t = n) = \boldsymbol{\pi}\mathbf{A}^{t-1}, \tag{7}$$

where $\mathbf{A}^{t-1}$ is the $t-1$ power of $\mathbf{A}$. For example, the probability distribution over states at $t = 3$ is equal to $\boldsymbol{\pi}\mathbf{AA} = \boldsymbol{\pi}\mathbf{A}^2$. When the initial state probability vector $\boldsymbol{\pi}$ is equal to the stationary distribution $\mathbf{p}$, the HMM is said to be stationary. From above Eqs. (4) and (7), it follows that then $P(S_t = 1, S_t = 2, \ldots, S_t = n) = \mathbf{p}$ for every t.

So far, HMMs are defined as models with discrete states, each characterized by their distribution function, and the evolution of states over time is governed by a Markov process. To generate data from a hidden Markov model with Gaussian response distributions, take the following steps:

1. choose the initial state of the hidden Markov process by drawing from the initial state probability vector $\boldsymbol{\pi}$, which provides the value of the state variable $S_{t=1}$
2. draw an observation from the (Gaussian) state-dependent distribution $f_{S_t}(.)$
3. generate a transition from the appropriate row of the transition matrix $\mathbf{A}(t)$ which provides the next value of the state variable S_{t+1}
4. repeat steps 2 and 3 until $t = T - 1$.

This process of data generation illustrates the dependences between the different variables, states and observations, in hidden Markov models. These dependences are depicted in Fig. 5.

In many psychological applications that involve learning or development, stationarity is unlikely to hold. Rather, it is expected that participants in an experiment are less likely to return to an initial state of, say, guessing behavior, after learning has taken place. In fact, in some models it is even assumed that as soon as learning has taken place, no regression can occur to previous states. In an HMM, this means that the transition matrix has $a_{ii} = 1$ for some state S_i, i.e., the state that is associated with mastering the task (see, e.g., Schmittmann, Visser, & Raijmakers, 2006). Such a state is called an absorbing state. In other applications though, such as the example response time experiment, the participant is expected to switch back and forth between two modes of behavior, and stationarity could be a reasonable starting assumption.

3. What is hidden about hidden Markov models?

Using the earlier example of response times, and assuming that those are from a mixture distribution, Fig. 1 reveals that it is not entirely clear which response times belong to which component distribution. This is the key to the third defining characteristic of hidden Markov models: the underlying (discrete) states S_t of the model are *hidden*.

This characteristic distinguishes *Markov* models from *hidden Markov* models: the states are hidden and cannot be observed directly. More formally, the distribution function $f(Y_t|S_t = i)$, or $f_i(Y_t)$, is not a deterministic function but rather a probability density function. Whenever $f_i(Y_t)$ is a deterministic one-to-one function, mapping states S_t into observations Y_t, the process S_t becomes observed rather than hidden, and the model reduces from a hidden Markov model to a Markov model.

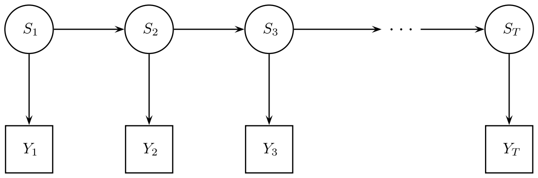


Fig. 5. Dependence graph for variables in a hidden Markov model.

What does this mean in actual cases? In speech recognition, in which HMMs are frequently applied (Rabiner, 1989), the observed variables Y_t are sound waves, and the hidden variables S_t are words (that need to be recognized). As there is ambiguity in the signal, e.g., confusion between /*p*/ and /*b*/, the relationship between the data (the speech signal), and the state (the word) is probabilistic. In the example mentioned earlier about sleep stages (Flexer et al., 2002), the observed variables are the EEG measurements and the hidden variables are the sleep stages. Again, there is a probabilistic relationship between EEG readings and sleep stages.

The application of Markovian models in psychology dates back at least to the 1950's when they were applied as models for language use (Miller, 1952). In those applications, words are the observed variables, and the hidden states and transitions between them can be interpreted as grammatical rules. HMMs were applied in a similar manner, i.e., as corresponding to a system of grammatical rules, in the analysis of implicit learning of artificial grammars (Visser, Raijmakers, & Molenaar, 2007).

Also in the domain of learning, Markovian models have been applied frequently (e.g., Batchelder, 1970, Kintsch & Morris, 1965, Wickens, 1982). In learning applications, accuracy of trials are the observed variables and underlying cognitive states are the hidden variables.

To summarize, HMMs are characterized by discrete, hidden states that follow a Markov process. The dependences between the variables in an HMM can be summarized in the dependency graph in Fig. 5. At each occasion of measurement, the process is in state S_t, and then emits an observation Y_t. In the literature on HMMs, the observations Y_t are often called observation symbols whenever they have discrete values. Note that so far the distribution of the Y_t was left unspecified. In the formulation of hidden Markov models below, this distribution is left open, and hence no limitations are placed on this distribution. Common special cases are where $Y_{1:T}$ is discrete, and the model is referred to as the HMM for discrete data or discrete HMM, and the case where Y_t conditional on S_t is Gaussian, which is referred to as the Gaussian Markov Model (GMM). The observations $Y_{1:T}$ can also be multivariate in which case typically local independence is assumed between them, as is common in latent variable models in general (Bollen, 2002).

Hidden Markov models are also called latent Markov models, although much of the literature on these topics is relatively separate. One of the reasons for this separation may be that the focus of HMMs has traditionally been on time series data (single participant, T large), whereas latent Markov models have mostly been applied in longitudinal data settings (n large, T small), alternatively called repeated measurements or panel data. This difference also leads to some differences in computing the likelihood and estimation of parameters of such models, which are rather non-trivial. Latent Markov models as applied to panel data date back at least to Wiggins (1973); Rolf Langeheine and colleagues have done much work in extending the basic model and applying it to different settings (Langeheine, 1988; Langeheine & Van de Pol, 1990; Vermunt, Langeheine, & Böckenholt, 1999). In the final section, this issue is discussed further.

4. How to compute the likelihood of HMMs?

The dependency graph in Fig. 5 is crucial in computing the likelihood of HMMs. Assuming that the hidden state variables S_t are known, the joint likelihood of the data and the hidden state sequence of the HMM is written as:

$$L(Y_{1:T}, S_{1:T}|\boldsymbol{\lambda}) = \pi_{S_1} f_{S_1}(Y_1) \prod_{t=1}^{T-1} a_{S_t S_{t+1}} f_{S_{t+1}}(Y_{t+1}), \tag{8}$$

where $\boldsymbol{\lambda}$ is the parameter vector of the HMM. The parameter vector $\boldsymbol{\lambda}$ of HMMs consists of three separate parts related to the three submodels of the hidden Markov model: the initial state probabilities $\boldsymbol{\pi}$, the transition model which determines the transition probabilities $\mathbf{A}$, and the response models, consisting of the state dependent distributions $f_i(Y_t)$. The parameter vector of the response models is customarily denoted by $\mathbf{B}$. In the case of a Gaussian HMM, for example, this vector contains the state dependent means and standard deviations. The parameter vector $\boldsymbol{\lambda}$ is defined as $\boldsymbol{\lambda} := (\boldsymbol{\pi}, \mathbf{A}, \mathbf{B})$.

As can be seen in Eq. (8), the computation of the likelihood follows the nodes in the dependency graph in Fig. 5. The first two terms are the initial state probability π_{S_1} and the density of the first response $f_{S_1}(Y_1)$. The next terms in the product are the transition probability to the next states, and the density of the response that is the result of that state.

Eq. (8) provides the joint likelihood of the data and the hidden state sequence, assuming that the latter is known. This means all the probabilities and densities depend on a specific state sequence $S_{1:T}$. To arrive at the likelihood of the data alone, independent of a specific state sequence, it is necessary to sum over possible state sequences:

$$L(Y_{1:T}|\boldsymbol{\lambda}) = \sum_{\text{all } S_{1:T}} \pi_{S_1} f_{S_1}(Y_1) \prod_{t=1}^{T-1} a_{S_t S_{t+1}} f_{S_{t+1}}(Y_{t+1}), \tag{9}$$

where the summation over 'all $S_{1:T}$' is an enumeration over possible state sequences $S_{1:T}$.

Computing the likelihood using Eq. (9) above is both impractical and unfeasible for two separate reasons. First, assuming a 2-state model for our response time data, which has 168 observations, the number of possible state sequences is 2^{168}. Consequently, the sum in Eq. (9) has 2^{168} summands, which is clearly infeasible. Second, each of the summands in Eq. (9) is the product of $2 * 168 = 336$ probabilities (the transition probabilities) and densities (the response densities), both of which are generally ≤ 1. As a consequence, this product is impractical to compute as it would easily cause underflow, i.e., generate numbers that are too small to represent in any reasonable computer system.

To solve the problem of computing the likelihood, an efficient way of combining state sequences is needed. To do so, the following so-called forward variables are defined as (Rabiner, 1989):

$$\alpha_t(i) := \pi_i f_i(Y_t), \quad i = 1, \ldots, n, t = 1, \tag{10}$$

$$\alpha_t(j) := \sum_{i=1}^{n} \alpha_{t-1}(i) a_{ij} f_j(Y_t), \quad t = 2, \ldots, T, \; j = 1, \ldots, n. \tag{11}$$

The variables $\alpha_1(i)$ are the probabilities of starting in state S_i, i.e., $S_1 = i$, multiplied by the conditional density of Y_1 in state S_i. Writing out, for example, $\alpha_t(j)$ for $t = 2$ and state 1 in a 2-state model gives:

$$\begin{aligned}\alpha_2(1) &= \alpha_1(1) a_{11} f_1(Y_2) + \alpha_1(2) a_{21} f_1(Y_2)\\ &= \pi_1 f_1(Y_1) a_{11} f_1(Y_2) + \pi_2 f_2(Y_1) a_{21} f_1(Y_2).\end{aligned}$$

This illustrates that $\alpha_2(1)$ combines the probabilities of different hidden state sequences that lead to state $S_{t=2} = 1$. For $t = T$, the vector of variables $\alpha_T(j)$ represents the likelihood of ending in state j at $t = T$, i.e $S_T = j$. The likelihood of the observation sequence $Y_{1:T}$ can thus be written as the sum over these variables:

$$L(Y_{1:T}|\boldsymbol{\lambda}) = \sum_{i=1}^{n} \alpha_T(i). \tag{12}$$

Computing the $\alpha_t(j)$ takes in the order of n^2 computations, and hence, computing the likelihood $L(Y_{1:T}|\boldsymbol{\lambda})$ takes $n^2 T$ computations ($4 * 168$ in our example, as against $336 * 2^{168}$ when using the naive method from Eq. (9)). Using the forward variables solves the infeasibility issue in computing the likelihood. The previously indicated impracticality still remains: the forward variables, as they are products of probabilities and densities, can diverge quickly as t increases resulting in underflow (or overflow).

Note that taking the logarithm in Eqs. (10) and (11) will not work because in (11) a sum over the states of the model is taken. There are various solutions to this problem, which all amount to ensuring that the forward variables remain in the order of magnitude of 1. Here, the approach by Rabiner (1989, p. 282–283) is discussed. The recursion for computing the forward variables in Eqs. (10) and (11) is replaced by the following recursion of scaled variables:

$$\hat{\alpha}_1(i) = \pi_i f_i(Y_1), \quad i = 1, \ldots, n \tag{13}$$

$$c_1 = \frac{1}{\sum_{i=1}^{n} \hat{\alpha}_1(i)} \tag{14}$$

$$\alpha_1^*(i) = c_1 \hat{\alpha}_1(i), \quad i = 1, \ldots, n \tag{15}$$

$$\hat{\alpha}_t(j) = \sum_{i=1}^{n} \alpha_{t-1}^*(i) a_{ij} f_j(Y_t), \quad t = 2, \ldots, T, \; j = 1, \ldots, n. \tag{16}$$

$$c_t = \frac{1}{\sum_{i=1}^{n} \hat{\alpha}_t(i)} \tag{17}$$

$$\alpha_t^*(i) = c_t \hat{\alpha}_t(i), \quad i = 1, \ldots, n. \tag{18}$$

Note that $\hat{\alpha}_1(i) = \alpha_1(i)$ and hence:

$$\alpha_1^*(i) = c_1 \alpha_1(i), \tag{19}$$

and by induction:

$$\alpha_t^*(i) = \left[\prod_{\tau=1}^{t} c_\tau\right] \alpha_t(i). \tag{20}$$

For $t = T$, and taking the sum over states gives:

$$\sum_{i=1}^{n} \alpha_T^*(i) = \left[\prod_{t=1}^{T} c_t\right] \sum_{i=1}^{n} \alpha_T(i), \tag{21}$$

and now we use that $\sum_{i=1}^{n} \alpha_T^*(i) = 1$ (by definition of the scaling factors), and that $\sum_{i=1}^{n} \alpha_T(i) = L(Y_{1:T}|\boldsymbol{\lambda})$ as seen in Eq. (12). As a result, the likelihood can now be written as:

$$L(Y_{1:T}|\boldsymbol{\lambda}) = \frac{1}{\prod_{t=1}^{T} c_t}, \tag{22}$$

so that the log-likelihood can be written as:

$$l(Y_{1:T}|\boldsymbol{\lambda}) = \log L(Y_{1:T}|\boldsymbol{\lambda}) = -\sum_{t=1}^{T} \log c_t. \tag{23}$$

In sum, computing the likelihood or log likelihood of HMMs is non-trivial, which is due to the nature of the model: the discreteness of the HMM states results in a combinatorial explosion of the number of possible state sequences. This same feature of HMMs makes computing the posterior state sequence non-trivial and this is discussed next.

5. How to compute the hidden states of HMMs?

Given an HMM for a time series, one may wish to know which sequence of states could have generated the particular time series, and at which time points switching between states has occurred. Suppose one uses an HMM to model a learning process. Among other states, this HMM has a state that corresponds to having mastered the task at hand, a so-called learned state. In such an application, it is interesting to know at which point the switch to this learned state has occurred for different individuals; such statistics can be used for further analysis, e.g., whether such switch points depend on person characteristics such as age. In the case of modeling sleep stages using HMMs, the interest is in classifying the EEG signal into the different sleep stages that may be used for further analysis, e.g., for diagnostic purposes.

The problem of computing the hidden states of HMMs is known as the decoding problem. The word *decoding* in this context stems from early applications of HMMs and related models in encryption, in information theory, and in the use of error correcting codes in storing information on computer hard drives. The terminology is also used in speech recognition applications.

To solve the decoding problem, it is necessary to compute the hidden state sequence $S_{1:T}$, conditional on the model and the data, such that the sequence has maximum probability $P(S_{1:T}|Y_{1:T}, \boldsymbol{\lambda})$. The resulting sequence $S_{1:T}$ is also referred to as the posterior or *a posteriori* state sequence, i.e., the state sequence after observing the data. As hinted to earlier, the problem once again is that the number of possible state sequences increases quickly as T grows. Hence, enumerating all possible state sequences and taking the one with the highest probability is infeasible.

One easy solution to this problem is simply by maximizing $P(S_t|Y_{1:T}, \boldsymbol{\lambda})$ instead, for each t separately. This is called *local* decoding as the states are chosen using local maximization for each t separately. An important drawback of local decoding is that it may lead to inadmissible state sequences: it could turn out that the transition probability a_{ij} between posterior chosen states $S_t = i$ and $S_{t+1} = j$ (for some t) is equal to zero. Of course this drawback only exists for models which have zero-entries in the transition matrix. The probabilities that are needed for local decoding, $P(S_t|Y_{1:T}, \boldsymbol{\lambda})$, are also used in the forward–backward algorithm that is used to optimize parameters of HMMs (this is further discussed in Section 6 below). In the remainder of this section, only global decoding is discussed, which guarantees that the hidden state sequence is a legal sequence according to the model.

The following algorithm, called the Viterbi algorithm (Forney Jr, 193; Viterbi, 1967) computes the globally optimal state sequence. Define the following δ and ψ variables (notation used here from Rabiner, 1989, p. 274):

$$\delta_1(i) := \pi_i f_i(Y_1), \quad i = 1, \ldots, n \tag{24}$$

$$\psi_1(i) := 0, \quad i = 1, \ldots, n \tag{25}$$

$$\delta_t(j) := \max_{1\leq i\leq n} \delta_{t-1}(i) a_{ij} f_j(Y_t), \quad t = 2, \ldots, T, j = 1, \ldots, n \tag{26}$$

$$\psi_t(j) := \operatorname*{argmax}_{1\leq i\leq n} \delta_{t-1}(i) a_{ij}, \quad t = 2, \ldots, T, j = 1, \ldots, n. \tag{27}$$

At each t, $\delta_t(i)$ represents the probability of the optimal path leading to state i, i.e., the path leading to state $S_t = i$ with maximal possible probability. Hence, at $t = T$, choosing i for which $\delta_T(i)$ is the maximum of $\boldsymbol{\delta}_T$, provides the final state in the optimal state sequence. The variable ψ is needed to track back the sequence of states that led to this final optimal state. The optimal state sequence q is determined in the following way:

$$q_T = \operatorname*{argmax}_{i} \delta_T(i), \tag{28}$$

$$q_t = \psi_{t+1}(q_{t+1}), \quad t = T-1, \ldots, 1. \tag{29}$$

That is, q_T is the maximum value of the δ-variables at time T, and subsequent states q_t are determined by following the path backwards which maximizes the probability along that path. Note that $\delta_t(i)$ is defined analogously to $\alpha_t(i)$, with the only difference that the *maximum* over possible paths is computed (in Eq. (26)) rather than the *sum* over possible paths as is done in the definition of $\alpha_t(i)$ (in Eq. (11)).

The Viterbi algorithm provides the globally optimal state sequence, which is an important property. Even so, the Viterbi algorithm has its own drawbacks. In particular, the algorithm provides only a single state sequence, and provides no information about the likelihood of similar sequences, which could be very close to the optimal one (see Guedon, 2007 for discussion of other possibilities). Also, under some circumstances, local decoding can lead to fewer overall errors in decoding than global decoding does, in particular when the states are not very well separated (see, e.g., Bulla, Mergner, Bulla, Sesboue, and Chesneau (in press)). Finally, it should be noted that in the example that is used here, and in many other applications where the states are well-separated and stable (i.e., high diagonal entries in the transition probability matrix $\mathbf{A}$), the differences between local and global decoding are expected to be minimal. Decoding of the hidden state sequence of HMMs of course assumes a model that is adequate for the data, which includes having optimized the parameters of the model, the topic of the next section.

6. How to estimate the parameters of HMMs?

Parameters of HMMs can be estimated using two well-known methods: the expectation–maximization (EM) algorithm or through direct maximization of the log-likelihood using general optimization methods. These methods are discussed in turn.

The EM algorithm (Dempster, Laird, & Rubin, 1978; McLachlan & Krishnan, 1997) centers around the idea that estimation of models is easy when the 'missing data' are known. The 'missing data' are latent or hidden variables, such as class memberships in mixture models. In the case of HMMs, assume for the moment that the hidden state sequence is known. Computing estimates for the transition probabilities would then proceed as follows:

$$a_{ij} = \frac{\text{the number of transitions from } S_i \text{ to } S_j}{\text{the number of transitions from } S_i},$$

which can be seen to be a simple counting exercise. Furthermore, assuming a Gaussian response model, for example, the mean parameter for each state of the HMM can be estimated by:

$$\mu_i = \frac{\sum\limits_{t \text{ for which } S_t = i} Y_t}{\text{the number of occurrences of state } S_i},$$

where the μ_i's are the state-dependent means of the Gaussian distributions.

How to solve the problem that the hidden state sequence is not in fact known? The next step in the EM algorithm is to replace knowledge of the hidden state sequence with reasonable guesswork, i.e., use the expected values. The EM algorithm for HMMs hence consists of the following steps:

1. Provide initial values of the parameters.
2. Compute the (expected) hidden state sequence based on the current parameter values.
3. Reestimate the parameters conditional on the current estimate of the hidden state sequence.
4. Repeat steps 2 & 3 until convergence.

Baum and Petrie (1966) prove that updating the parameters in this way results in convergence to a local maximum of the (log-) likelihood.

Step 2 in this algorithm could be done using the Viterbi algorithm discussed earlier in Section 5; that is, compute the hidden state sequence, update the parameters accordingly, repeat these steps until convergence. Note that this does *not* optimize the likelihood in Eq. (9) but the so-called state-optimized likelihood in Eq. (8), and the resulting algorithm is called the segmental K-means algorithm (Jelinek, 1976; Juang & Rabiner, 1990).[3] To optimize the likelihood, in step 2 in the algorithm, the *expected* values of state occupancy and state transitions (rather than the maximized values that the Viterbi algorithm computes) are to be used. The following notation is used for these quantities:

$$\gamma_t(i) := P(S_t = i|Y_{1:T}, \boldsymbol{\lambda}), \tag{30}$$

the probability of being in state S_i at time t, and:

$$\xi_t(i,j) := P(S_{t+1} = j, S_t = i|Y_{1:T}, \boldsymbol{\lambda}), \tag{31}$$

the probability of making a transition from S_i to S_j at time t.

These probabilities $\gamma_t(i)$ and $\xi_t(i,j)$ can be computed using the forward variables that are also used to compute the log-likelihood, and the so-called backward variables. The latter are, as the name suggests, similar to the forward variables but are computed starting at $t = T$ rather than at $t = 1$. Appendix A gives the full details of computing the backward variables, as well as computing $\xi_t(i,j)$ and $\gamma_t(i)$.

Given the values of $\gamma_t(i)$ and $\xi_t(i,\ j)$, the new estimates for the transition probabilities are (step 3 in the EM algorithm):

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,\ j)}{\sum_{t=1}^{T-1} \gamma_t(i)}. \tag{32}$$

The new estimates for the initial state probabilities are even simpler:

$$\pi_i = \gamma_1(i). \tag{33}$$

As for the response distribution parameters, the reestimation of course depends on the specific distributions and parameters. The case for the state dependent means of a Gaussian HMM is:

$$\mu_i = \frac{\sum_{t=1}^{T} \gamma_t(i) Y_t}{\sum_{t=1}^{T} \gamma_t(i)}. \tag{34}$$

From this equation, it can be seen that the new estimate of the state-dependent mean is a weighted sum of the observations Y_t, where the weighting factor for Y_t is the probability that that observation originated from state $S_t = i$. This is the core idea of the reestimation for the response model parameters: estimation is done using weights that express the certainty/uncertainty about data points originating from that distribution. This procedure generalizes easily to other parameters, such as the standard deviation of the Gaussian distribution, or to other measurement models, for example, logistic regression models.

The EM algorithm for HMMs is justified by the following argument. In Section 4, Eq. (8), the joint likelihood of data and hidden state sequence was defined, which can be rewritten as:

$$l(Y_{1:T}, S_{1:T}|\boldsymbol{\lambda}) = \log \pi_{S_1} + \sum_{t=2}^{T} \log a_{S_{t-1}S_t} + \sum_{t=1}^{T} f_{S_t}(Y_t),$$

by taking the logarithm and regrouping terms. In the context of the EM algorithm this (log-)likelihood is also called the complete data likelihood, as it assumes that the hidden states are known. This log-likelihood depends on the hidden state sequence, and to arrive at the proper log-likelihood, in the E-step of the EM algorithm, the expected values of the state sequences are used. The EM algorithm for HMMs consists in optimizing the following:

$$Q(\boldsymbol{\lambda}, \boldsymbol{\lambda}') = E_{\boldsymbol{\lambda}'}(l(Y_{1:T}, S_{1:T}|\boldsymbol{\lambda})), \tag{35}$$

the conditional expected log-likelihood function Q, where $\boldsymbol{\lambda}'$ is the current estimate of the parameters. Q can be written out as:

$$\begin{aligned} Q(\boldsymbol{\lambda}, \boldsymbol{\lambda}') = & \sum_{i=1}^{n} \gamma_1(i) \log P(S_1 = i|\boldsymbol{\pi}) \\ & + \sum_{t=2}^{T}\sum_{i=1}^{n}\sum_{j=1}^{n} \xi_t(i,j) \log P(S_t = j|S_{t-1} = i, \mathbf{A}) \\ & + \sum_{t=1}^{T}\sum_{j=1}^{n} \gamma_t(j) \log f(Y_t|S_t = j, \mathbf{B}), \end{aligned} \tag{36}$$

where the expected values $\xi_t(i,j)$ and $\gamma_t(i)$ are as defined above. Note that these expected values are computed using the previous value of the parameter vector $\boldsymbol{\lambda}'$. The M-step, or Maximization step, of the EM algorithm consists of the maximization of the right hand side of Eq. (36) for $\boldsymbol{\lambda} = (\boldsymbol{\pi}, \mathbf{A}, \mathbf{B})$. Because the right hand side of Eq. (36) consists of three separate parts, these can be maximized separately for $\boldsymbol{\pi}$, $\mathbf{A}$ and $\mathbf{B}$, assuming that there is no overlap in parameters between the initial probability, transition and response models. This property of the EM algorithm makes it fairly straightforward to consider more complex models.

Consider for example the response models, corresponding to the third part of this equation. Instead of only estimating a mean as was done earlier, it is possible to include direct effects of covariates. This would lead to a Markov mixture of regression models. Due to Eq. (36), estimation is straightforward as this is simply a weighted version of an ordinary regression model, with the weights being $\gamma_t(j)$. Similar arguments apply to the transition probabilities which may be parameterized using multinomial logistic regression such as to include the possible effects of (time-varying) covariates.

Next to the EM algorithm, direct maximization of the log-likelihood can also be used to arrive at parameter estimates. For example, in cases where one of the terms in Eq. (36) involves very complicated models for which no closed form (weighted) estimation methods exist, direct optimization is a good option (Zucchini, Raubenheimer, & MacDonald, 2008). Given proper optimization tools, direct optimization can also be considerably faster than the EM algorithm. See Turner (2008) for discussion of the optimal choice of maximization method when using direct optimization. On the downside, direct optimization typically requires good initial values of the parameters, whereas the EM algorithm is more robust in this respect (Bulla & Berzel, 2008).

When using direct optimization, the log-likelihood computation in Section 4 suffices in principle. However, to make convergence of the log-likelihood faster, it is advisable to also use the gradients, and possibly the Hessian, of the parameters (see for an algorithm that efficiently computes gradient and observed information for HMM parameters Lystig and Hughes (2002)). Having the Hessian available is of course also useful in computing standard errors of parameters. Visser, Raijmakers, and Molenaar (2000) compare several methods of computing standard errors of HMM parameters, among them bootstrapping and likelihood profiling. Another advantage of using direct optimization is that it is relatively easy to deal with missing values (Zucchini & MacDonald, 2009). Finally, direct optimization also allows straightforward fitting of equality or inequality constraints between parameters (Giudici, Ryden, & Vandekerkhove, 2000; Visser, Raijmakers, & Molenaar, 2002).

[3] The segmental K-means algorithms usually converges very quickly. Rabiner (1989) mentions the segmental K-means algorithm as an option to generate starting values for HMM parameters. See, e.g., Qin (2004) for a relatively recent application.

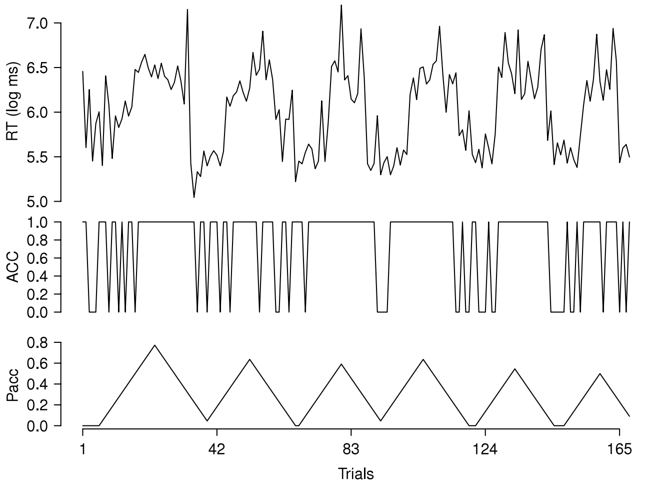


Fig. 6. Response times (RT), accuracy (ACC) and pay-off values (Pacc) for the first series of responses in data set speed from package **depmixS4** (Visser & Speekenbrink, 2010).

7. How to fit HMMs in practice?

To illustrate some of the possibilities of HMMs, in this Section I provide example models fitted to the response time data, as well as the accompanying accuracy data. Before discussing those models, it is useful to provide some more details on the background of the data.

7.1. Speed–accuracy data

The data are from consecutive lexical decision trials, in which the participant had to decide whether the presented sequence of letters formed a word or not. The data that are analyzed her are from participant A in Experiment 1 in Dutilh et al. (2011). At each trial, response time and accuracy were recorded. To manipulate the accuracy and response times, there was a dynamic pay-off for accuracy of responding. The pay-off for speed was determined as 1 minus the pay-off for accuracy. The pay-off for accuracy was varied continuously in a zigzag manner, and there were 6 maxima for the pay-off for accuracy during 168 trials. Two more blocks of trials (134 and 137 trials respectively) were recorded, and the data show similar effects as this first block; those data are not further analyzed here. The participant was shown the reward after each trial such that behavior could be adjusted accordingly. The response times (*RT*), accuracy (*ACC*), and the pay-off for accuracy (*Pacc*) are depicted in Fig. 6.

The most striking feature about these data, and in particular the response times, is their seemingly periodic nature, and the goal of modeling is to capture this aspect of the data. The periodicity of the data is clearly related to the pay-off for speed and accuracy, as shown by the correlations between the variables. The correlations between the variables are shown in Table 1. In the models below, the relationship between *Pacc* on the one hand, and *RT* and *ACC* on the other hand is explored.

Table 1
Correlations between variables from the speed–accuracy trade-off experiment; standard errors are between parentheses.

	RT	*ACC*
ACC	0.324 (0.069)	
Pacc	0.643 (0.046)	0.425 (0.063)

The most important question about these data is what the effect is of the (continuous) changes in pay-off for speed and accuracy. The first hypothesis states that *continuous* changes in pay-off should result in *continuous* changes in speed and accuracy (Pew, 1969; Wickelgren, 1977). Alternatively, the second hypothesis states that the changes in pay-off may result in switching between two modes of responding, one slow and accurate, and another with fast responses and accuracy at chance level. Hence, the second hypothesis states that *continuous* changes in pay-off should result in *discontinuous* changes in speed and accuracy (Dutilh et al., 2011). Dutilh et al. (2011) provide more data, as well as more theoretical background about the relevance of studying the speed–accuracy trade-off in experimental psychology.

7.2. Model specifications

In order to test above hypotheses, the appropriate models to compare are: (1) a 1-state model with the pay-off for accuracy as a predictor for response time and accuracy; (2) a 2-state model with switching between the states. The 2-state model can be further elaborated by making the transition probabilities depend on the pay-off for accuracy. Note that the 1-state model is not a proper hidden Markov model because it has only a single state, which is, as a consequence not hidden, and moreover there is no dynamic part in the model because the transition matrix consists only of a single element equal to 1. The 1-state model is in effect a multiple regression model for the response time and accuracy and it is specified by the following equations:

$$RT_t = \mu + \beta \cdot Pacc_t + \epsilon_t, \quad \text{with } \epsilon_t \sim \mathbf{N}(0, \sigma^2) \tag{37}$$

$$ACC_t \sim \mathbf{Bernoulli}(p_t), \quad \text{with } \mathrm{logit}(p_t) = b_0 + b_1\, Pacc_t, \tag{38}$$

where RT_t are the response times and ACC_t are the accuracy values; $Pacc_t$ represents the pay-off for accuracy, scaled from 0 to 1, with 1 representing the maximum reward for accuracy, and no reward for speed; conversely, $Pacc_t = 0$ means no reward for accuracy, and maximum reward for speed. For the *RT* variable, the parameters of the model are: the mean response time μ, the regression coefficient β relating the pay-off for accuracy to response time, and the standard deviation σ. For the *ACC* variable, the parameters of the model are the coefficients b_0 and b_1, where b_1 relates $Pacc_t$ to the logit of the probability $P(ACC_t = 1) = p_t$.

The second hypothesis states that there are two modes of behavior, and that the participant switches between these over the series of trials. Such a hypothesis is naturally captured by a 2-state HMM. The 2-state HMM is defined by the state dependent means and standard deviations of the *RT* variable and the state dependent probabilities for the *ACC* variable, i.e., $P(ACC_t = 1)$, as follows:

$$RT_t = \mu_i + \epsilon_t, \quad \text{with } \epsilon \sim \mathbf{N}(0, \sigma_i^2), i = 1, 2 \tag{39}$$

$$ACC_t \sim \mathbf{Bernoulli}(p_i), \quad i = 1, 2. \tag{40}$$

For modeling *RT* then, the parameters of this model are the state dependent means μ_i and standard deviations σ_i. For *ACC*, the parameters are the state dependent Bernoulli probabilities p_i. Next to these parameters, the transition matrix $\mathbf{A}$ and the initial state probability vector $\boldsymbol{\pi}$ are estimated.

Note that the variable *Pacc* has no role in the 2-state model described here. In the 1-state model, the assumption is expressed that *Pacc* influences response times and accuracy *directly*. In the 2-state model, the core assumption is that there are two modes of behavior, slow and accurate versus fast guessing behavior, and that switching between these modes of behavior is influenced by the pay-off for accuracy. Hence, *Pacc* is hypothesized to have an *indirect* influence on *RT* and *ACC* through the state variables. This hypothesis can also be made more within the framework of hidden Markov models by including *Pacc* as a covariate on the transition probabilities. That is, the transition probabilities are modeled as a function of *Pacc*. The third model that is fitted to the data incorporates this idea. Such models with time-varying covariates that influence the transition probabilities have been studied before in time series data (Hughes & Guttorp, 1994; Hughes, Guttorp, & Charles, 1999), and in the context of longitudinal data (Chung, Walls, & Park, 2007; Vermunt et al., 1999).

The measurement equations, i.e., the equations that relate the hidden states to the observed variables, of the third model are identical to those of the 2-state model (Eqs. (39) and (40)). The transition probabilities are modeled by the following equations:

$$\text{logit}(1 - a_{11}(t)) = \eta_0^1 + \eta_1^1 \cdot Pacc_t. \tag{41}$$

$$\text{logit}(a_{22}(t)) = \eta_0^2 + \eta_1^2 \cdot Pacc_t. \tag{42}$$

The transition probabilities are hence modeled by a logistic regression with *Pacc* as predictor. In HMMs with more than two states, multinomial logistic regression rather than just logistic regression is needed to model transition probabilities as function of covariates. See, e.g., Agresti (2002, chapter 7) for various possibilities. In this model, as in the first 2-state model, also the initial state probability vector $\boldsymbol{\pi}$ is estimated.

7.3. Results

The resulting parameters of the 1-state model are the following for the response times: $\mu = 5.56$, $\sigma = 0.367$, and $\beta = 1.58$. It can be seen that an increase in *Pacc* leads to an increase in *RT* as indicated by the positive value of $\beta = 1.58$. The parameters for the accuracy variable are $b_0 = -0.441$, and $b_1 = 6.64$. From the latter parameters, it can be seen that the intercept for the accuracy variable is the inverse logit of −0.441 which equals 0.39; this means that when the pay-off for accuracy is 0, the probability $P(ACC = 1) = 0.39$ according to the model. This is somewhat odd as the chance level in the task is 0.5. When the pay-off for accuracy is at its maximum, $P(ACC = 1) = 0.998$, almost indistinguishable from 1.

The 2-state model has the following parameters. The initial state probability vector $\boldsymbol{\pi}$ has values 1 and 0, indicating that the process begins in state 1. The state dependent mean response times are $\mu_1 = 6.43$ ($\sigma_1 = 0.254$), and $\mu_2 = 5.62$ ($\sigma_2 = 0.26$). The state dependent probabilities are $p_1 = 0.942$ and $p_2 = 0.569$. These values correspond well with the hypothesized interpretation of a 2-state model: state 1 represents highly accurate and relatively slow responding, whereas state 2 represents faster responding close to chance level of 0.5.

The dynamic part of the model should capture the switching between the states. The transition matrix **A** is estimated as:

$$\mathbf{A} = \begin{pmatrix} 0.907, 0.093 \\ 0.088, 0.912 \end{pmatrix}. \tag{43}$$

The probabilities on the diagonal of the matrix indicate that the states of the 2-state HMM are rather stable, i.e., there is a high probability of remaining in the states, indicating stability of the behavior.

The third and final model that was fitted to these data was another 2-state model with the addition of a covariate effect on the transition probabilities. The parameters of this model are the following. The state dependent means for *RT* are: $\mu_1 = 6.43$ ($\sigma_1 = 0.253$), and $\mu_2 = 5.63$ ($\sigma_2 = 0.274$). The state dependent parameters relating to *ACC* are $p_1 = 0.949$ and $p_2 = 0.571$. As can be seen, the parameters relating to the measurement part of the model are very similar to the parameters of the 2-state model without the covariate on the transition probabilities. The parameters of most interest for this model are the transition model parameters. These are $\eta_0^1 = 4.48$ and $\eta_1^1 = -20$ in state 1 of the model. Using Eqs. (41) and (42), this means that the transition probability a_{11} equals 0.0112 when $Pacc = 0$ and that $a_{11} = 1$ when $Pacc = 1$, which corresponds well with the interpretation of state 1 as the 'slow and accurate' state; i.e., whenever the pay-off for accuracy is low, the probability of remaining in the high-accuracy state 1 drops to almost zero. The corresponding parameters in state 2 are $\eta_0^2 = 7.8$ and $\eta_1^2 = -21.6$. This means that the transition probability a_{22} equals 1 when $Pacc = 0$ and that $a_{22} = 1.05e - 06$ when $Pacc = 1$. Again, these values correspond well with the interpretation of state 2 as a fast guessing state, i.e., when the pay-off for accuracy is zero, or conversely, when the pay-off for speed is maximal, the probability of remaining in the state is 1. Dutilh et al. (2011) discuss some further constraints to this model that correspond with hypotheses concerning the switching process between the two modes of behavior.

7.4. Model selection

Which of these models provides the best fit to the data? Several statistics are available for comparing non-nested models, of which AIC and BIC are best known (see Zucchini, 2000 for an introduction to model selection). The BIC is commonly used in applications of HMMs and it is used here too (but see for discussion of this practice, and an alternative Mackay (2002)). The BIC for the 1-state model is $BIC(1) = 315$, and its value in the 2-state model is $BIC(2) = 302.4$. The value of the BIC in the 2-state model with covariate is $BIC(2a) = 270.8$. This indicates that the 2-state models describe the data better than does the 1-state model. For purposes of completeness, a 3-state model was also fitted to the data which resulted in a BIC of 305.3. It is hence warranted to conclude that the 2-state models capture the data better than the 1- and 3-state models respectively. The BIC values also indicate that the 2-state model with covariate outperforms the 2-state model without the covariate. As these 2-state models are nested, it is also possible to test the difference in goodness-of-fit between these models by the log-likelihood ratio test (Giudici et al., 2000; Visser et al., 2002). The ratio statistic equals 41.8 which follows a χ^2-distribution with $df = 2$; the associated p-value is almost 0, indicating that the addition of the covariate parameters η_1^1 and η_1^2, significantly improves the goodness-of-fit of the 2-state model.

Next to model selection criteria for choosing between models, such as the BIC, other statistics for goodness-of-fit, should also be inspected; see Mackay Altman (2004) and Titman and Sharples (2008) for several options in this regard. Another important tool in inspecting the appropriateness of models is to make use of the hidden state sequence.

7.5. Posterior states

Using the Viterbi algorithm as described in Section 5, the hidden state sequence for the above fitted 2-state model (with covariate) can be computed. The posterior states can be used in several ways. Fig. 7 depicts the response times, and the model predicted response times. The model predicted response times are in this case simply the state-dependent means because there are no further predictors for the response times. Fig. 7 reveals at which trials the switches between the slow and fast states occur.

An interesting statistic that can be derived from the posterior state sequences is the residual correlation between *RT* and *ACC*. In the full data set, the correlation between *RT* and *ACC* is 0.324 ($p < 0.001$), meaning that a higher accuracy is associated with slower, i.e., higher, *RT*s. The correlation changes drastically when considered within each of the states separately. Within the slow-accurate state, the correlation equals −0.21 ($p = 0.058$), and within the fast-guessing state the correlation equals 0.08 ($p = 0.45$). Both correlations are non-significant (although the slow-state correlation trends toward significance), fulfilling an important assumption that was made in this modeling: responses should be independent conditional on the hidden state.

Another interesting statistic that was also considered in Section 1, is the autocorrelation. As was noted there, the lag-1

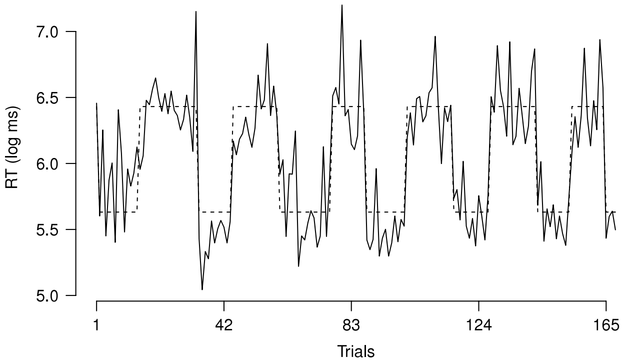


Fig. 7. Posterior predicted response times. The dashed line depicts the mean corresponding to the posterior estimated state at each trial.

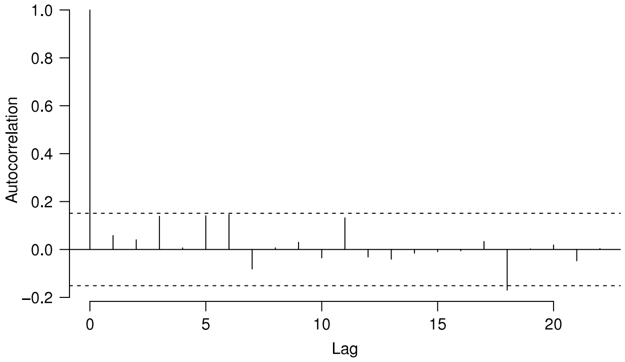


Fig. 8. Autocorrelation function of the residuals of the 2-state model (with covariate) of the response times; the dashed lines represent the 95% confidence limits.

autocorrelation was highly significant at 0.62. Conditional on the hidden states of the model, i.e., the lag-1 autocorrelation of the residual RTs has dropped to the non-significant value of 0.0583. The full autocorrelation function of the residuals is depicted in Fig. 8. This again fulfills an important assumption of the hidden Markov model: observations should be independent of each other conditional on the hidden states. Note that here independence refers to independence over time, whereas in the previous paragraph independence concerned independence of responses within multivariate measurements at each point in time.

7.6. *Software*

The models that are reported in this section are fitted using the **depmixS4** package (Visser & Speekenbrink, 2010) for the *R* program for statistical computing (R Development Core Team, 2010). The `speed` data set which is used here is part of that package. The *R*-code for fitting some of the models discussed here is included in Appendix B. An introduction to the capabilities of **depmixS4** with more elaborate examples is provided in Visser and Speekenbrink (2010), and in the online help pages of the package.

Within *R* (R Development Core Team, 2010), there are also several other packages that can fit hidden Markov models. Examples are package **HiddenMarkov** (Harte, 2010), which fits univariate time series with `glm` distributed responses, and package **RHmm** (Taramasco, 2009), which fits HMMs with either discrete, Gaussian, or multivariate normal data. Also worth mentioning is package **msm** (Jackson, 2010) which specializes in continuous time measurements. See Visser (2010), Section 2 for a brief overview of *R*-packages that may be used for hidden Markov models. Outside of *R*, Matlab has a number of HMM toolboxes.

8. Conclusions

This tutorial has provided an explanation and description of the key aspects of hidden Markov modeling. Hidden Markov models should be in the standard repertoire of (mathematical) psychologists working with time serial data as these models are very flexible and at the same time easy to understand and interpret. This tutorial together with an *R* package to fit HMMs should suffice to become more familiar with the possibilities of applying these models. However, if that proves insufficient, there is always the option of more reading, and in closing this tutorial I provide some pointers to further literature.

As mentioned earlier, hidden Markov models are equivalent to latent Markov models, and under the latter name there are many applications to longitudinal or panel data (Bartolucci, Lupparelli, & Montanari, 2009; Bartolucci, Pennoni, & Francis, 2007; Bijleveld & Mooijaart, 2003; Böckenholt, 1999, 2005; Chung et al., 2007). Bartolucci, Farcomeni, and Pennoni (2010) provide an overview of latent Markov models for categorical panel data.

Although there is no principled difference between latent and hidden Markov models, there are a number of pragmatic differences (which may be partially responsible for the fact that there is not much cross-fertilization between these literatures). First, estimation and inference in latent Markov models, i.e., when applied to data with small T, is hindered much less by issues related to scaling of the likelihood, and the combinatorial explosion of the number of possible state sequences as discussed in Sections 4 and 5. Second, in typical applications of latent Markov models, the default assumption is that the transition matrix depends on t nonparametrically, i.e $\mathbf{A}(t)$ is estimated separately for each t. It is possible to estimate such models because of using panel data, meaning many replications of the observed process. In HMMs, the default assumption is rather that the underlying Markov chain of the process is homogeneous over time, i.e., $\mathbf{A}(t) = \mathbf{A}$ for every t. In HMMs this is the natural assumption; certainly so when modeling single time series, in which case there is simply no information to estimate time-dependent transition matrices. A common ground between these models is to be found in cases where the transition matrix is made to depend on covariates or exogenous variables such as in Hughes and Guttorp (1994) and Vermunt et al. (1999), and in the example above.

Zucchini and MacDonald (2009) is, as far as I know, the only general introductory textbook about hidden Markov models. A recent literature review on HMMs is Ephraim and Merhav (2002), which provides an extensive review of inference, estimation, state estimation, forecasting and applications of HMMs, mostly in the technical sciences, with some mention of applications in economics. Other texts focus on special topics such as speech recognition (Rabiner, 1989) or biological sequence analysis (Krogh, 1998), and may hence be less appropriate for psychologists or social scientists. A much more advanced treatment of mixture models and hidden Markov mixture models can be found in Frühwirth-Schnatter (2006). Cappe, Moulines, and Ryden (2005) provide an in-depth treatment of various forms of inference in HMMs, e.g., treating smoothing, Monte-Carlo, ML and fully Bayesian estimation of HMM parameters. Ghahramani (2001) discusses HMMs in the context of the more general class of Bayesian network models.

Another possibility of getting more familiar with HMMs is by studying their use in the social sciences, which seems to be on the rise with quite a number of recent publications. Al-Ani (2004) uses HMMs as a diagnostic tool in sleep apnea. Bartolucci et al. (2007) apply HMMs to study patterns of criminal activity over time, as did earlier work by Bijleveld and Mooijaart (2003), which analyses recidivism in juvenile delinquency. Bartolucci and Solis-Trapala (2010) study the development of inhibitory control and attentional flexibility. Liechty, Pieters, and Wedel (2003) apply Bayesian HMMs to eye-movement data; an example of studying animal behavior also related to perceptual dynamics may be found in Otterpohl (2001). Another large field of application is in finance (Bulla & Bulla, 2006; Ghysels, 1994). Chung et al. (2007) study change in academic achievement, and provide an example of latent Markov models with the use of covariates on the transition probabilities. Extensions of HMMs in which random effects are added to transition probabilities or measurement model parameters can be found in Altman (2007). A special case of this, concerning mixtures of HMMs can be found in Schmittmann et al. (2006), in which two latent subpopulations each have their own learning strategy described by 2- or 3-state HMMs respectively.

Models for individual time series that are related to hidden Markov models are so-called regime-change models (Kim, 1994) and threshold autoregressive models (Hamaker, 2009), which typically study time series in which only 1 or a few changes occur. In the example used here, and in the typical HMM application, the observed behavior is expected to change back and forth between different modes many times, but see Frühwirth-Schnatter (2006, chapter 10) for discussion of the use of HMMs as change-point models. Hidden Markov models are the model of choice for analyzing cognitive processes based on time serial data. With this tutorial I hope the reader has sufficient starting points to explore the possibilities of applying these wonderful models to his or her own data.

Acknowledgments

I thank Darja Tutschkow, Brenda Jansen, three anonymous reviewers, and the editor for suggesting many improvements on previous drafts of this paper. I thank Han van der Maas for providing me with the data (which was published in Dutilh et al. (2011)) used in the example which required me to think about the inclusion of covariates for the transition probabilities.

Appendix A. Backward variables, ξ and γ

Here the definition of the backward variables and of $\gamma_t(i)$ and $\xi_t(i,j)$ are provided. The backward variables are defined analogously to the forward variables $\alpha_t(i)$. Remember that $\alpha_t(i) = P(Y_{1:t}, S_t = i|\boldsymbol{\lambda})$, the probability of the responses until time t, when the process is in state $S_t = i$. The backward variable are defined as $\beta_t(i) = P(Y_{t+1:T}|S_t = i, \boldsymbol{\lambda})$, the probability of the observations from $t+1$ to T, given state $S_t = i$.

$$\beta_T(i) = 1, \quad i = 1, \ldots, n \tag{A.1}$$

$$\beta_t(i) = \sum_{i=1}^{n} a_{ij} f_j(Y_{t+1})\beta_{t+1}(j),$$
$$t = T-1, T-2, \ldots, 1, i = 1, \ldots, n. \tag{A.2}$$

Combining the definitions of $\alpha_t(i)$ and $\beta_t(i)$, it should be clear that $\gamma_t(i)$ is:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{n} \alpha_t(i)\beta_t(i)}. \tag{A.3}$$

The product of $\alpha_t(i)$ and $\beta_t(i)$ gives $P(Y_{1:T}, S_t = i|\boldsymbol{\lambda})$, the probability of the full data and state $S_t = i$. Remember that $\gamma_t(i)$ was defined as: $\gamma_t(i) = P(S_t = i|Y_{1:T}, \boldsymbol{\lambda})$, and hence the product of $\alpha_t(i)$ and $\beta_t(i)$ needs to be normalized by dividing by the likelihood of the data $P(Y_{1:T}|\boldsymbol{\lambda})$, which can be written as $\sum_{i=1}^{n} \alpha_t(i)\beta_t(i)$.

The probabilities $\xi_t(i,j)$ can also be formed using the forward and backward variables. Remember that $\xi_t(i,j)$ is the joint probability of the data and making a transition from state S_i to S_j at time t, and hence it can be written as:

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} f_j(Y_{t+1})\beta_{t+1}(j)}{P(Y_{1:T}|\boldsymbol{\lambda})}. \tag{A.4}$$

The forward probability $\alpha_t(i)$ accounts for $Y_{1:t}$ and $S_t = i$, followed by the transition probability a_{ij}, and the probability of response Y_{t+1} in state $S_{t+1} = j$. The backward variable $\beta_{t+1}(j)$ accounts for the data $Y_{t+2:T}$. Finally also note the following relation between $\gamma_t(i)$ and $\xi_t(i,j)$:

$$\gamma_t(i) = \sum_{j=1}^{n} \xi_t(i,j). \tag{A.5}$$

To prevent underflow and other computational issues, the backward variables are scaled in the same way, and using the same scaling factors c_t, as the forward variables.

Appendix B. *R*-code for estimating the example models

The code below is used to fit models that are reported in the paper and serves as an illustration of how to fit such models using package **depmixS4** in *R*. The first four lines of code load the package into *R*, access the data set that is part of the package, and then selects the first series of trials. The full data set contains two more series of trials. For the illustrations in this tutorial, only the first of the three series is analyzed.

```
library(depmixS4)
data(speed)
sp1 <- data.frame(speed[1:168,])
names(sp1) <- c("RT", "ACC","Pacc")
```

In the introductory Section 1, two models are fitted on the response time variable (RT) of this data set, a 1- and a 2-component mixture model respectively. The 1-component mixture simply returns the mean and sd of the data but is included here for completeness and is fitted by the following code:

```
m1 <- mix(RT~1,nstates=1, data=sp1)
fm1 <- fit(m1)
bic1 <- BIC(fm1)
```

Fitting the model, rather than just computing the mean RT, gives access to log-likelihood which is subsequently used in computing the BIC and AIC for model selection purposes.

The 2-component mixture model is specified and fitted by the following code:

```
set.seed(1)
m2 <- mix(RT~1,nstates=2, data=sp1,
        respstart=c(rnorm(1,5),1,rnorm(1,6),1))
fm2 <- fit(m2,emcontrol=em.control(rand=F))
bic2 <- BIC(fm2)
```

The models from Section 7 can be fitted using the following blocks of code. First, a 1-state model with the variable *Pacc* as direct effect on speed and accuracy:

```
m1p <- depmix(list(ACC~Pacc,RT~Pacc), nstates=1,
    data=sp1, family=list(multinomial(),gaussian()))
fm1p <- fit(m1p)
```

Note that the model specification uses the ~-notation as common in *R* for specifying regression effects. Both accuracy and log response time depend on the pay-off for accuracy. The accuracy variable is modeled as a multinomial (with logistic link function), and the log response times are modeled as a Gaussian.

The model of most interest is the 2-state model of log response times and accuracy combined:

```
set.seed(1)
m2 <- depmix(list(RT~1,ACC~1),nstates=2, data=sp1,
   family=list(gaussian(),multinomial()))
fm2 <- fit(m2)
```

The following code models the transition probabilities as a function of the pay-off for accuracy:

```
set.seed(1)
m2a <- depmix(list(RT~1,ACC~1),nstates=2, data=sp1,
       family=list(gaussian(),multinomial()),
       transition=~Pacc)
fm2a <- fit(m2a)
```

The final model that was mentioned is a 3-state model which is specified and fitted using:

```
set.seed(1)
m3 <- depmix(list(RT~1,ACC~1),nstates=3, data=sp1,
      family=list(gaussian(),multinomial()))
fm3 <- fit(m3)
```

The probabilities of the posterior states, the $\delta_t(i)$ variables from Section 5, and the posterior state sequence itself can be accessed using the following call:

```
pst <- posterior(fm2)
```

The return value is a matrix with both the $\delta_t(i)$ probabilities as well as the posterior state sequence.

References

Agresti, A. (2002). Categorical data analysis. In *Wiley series in probability and mathematical statistics* (2 ed.). Hoboken, NJ: Wiley-Interscience.

Al-Ani, T. (2004). Using hidden Markov models for sleep disordered breathing identification. *Simulation Modelling Practice and Theory*, *12*, 117–128.

Altman, R. M. (2007). Mixed hidden Markov models. *Journal of the American Statistical Association*, *102*, 201–210.

Bartolucci, F., Farcomeni, A., & Pennoni, F. 2010. An overview of latent Markov models for longitudinal categorical data. Arxiv preprint http://arxiv.org/abs/1003.2804.

Bartolucci, F., Lupparelli, M., & Montanari, G. (2009). Latent Markov model for longitudinal binary data: an application to the performance evaluation of nursing homes. *The Annals of Applied Statistics*, *3*, 611–636.

Bartolucci, F., Pennoni, F., & Francis, B. (2007). A latent Markov model for detecting patterns of criminal activity. *Journal of the Royal Statistical Society: Series A Statistics in Society*, *170*, 115–132.

Bartolucci, F., & Solis-Trapala, I. L. (2010). Multidimensional latent Markov models in a developmental study of inhibitory control and attentional flexibility in early childhood. *Psychometrika*, *75*, 725–743.

Batchelder, W. (1970). An all-or-none theory for learning on both the paired-associate and concept levels. *Journal of Mathematical Psychology*, *7*, 97–117.

Baum, L. E., & Petrie, T. (1966). Statistical inference for probabilistic functions of finite state Markov chains. *Annals of Mathematical Statistics*, *67*, 1554–1563.

Bijleveld, C. C., & Mooijaart, A. (2003). Latent Markov modelling of recidivism data. *Statistica Neerlandica*, *57*, 305–320.

Böckenholt, U. (1999). Measuring change: mixed Markov models for ordinal panel data. *British Journal of Mathematical and Statistical Psychology*, *52*, 125–136.

Böckenholt, U. (2005). A latent Markov model for the analysis of longitudinal data collected in continuous time: states, durations, and transitions. *Psychological Methods*, *10*, 65–83.

Bollen, K. A. (2002). Latent variables in psychology and the social sciences. *Annual Review of Psychology*, *53*, 605–634.

Bower, G. H., & Trabasso, T. (1964). Concept identification. In R. C. Atkinson (Ed.), *Studies in mathematical psychology* (pp. 32–94). Stanford University Press.

Bulla, J., & Berzel, A. (2008). Computational issues in parameter estimation for stationary hidden Markov models. *Computational Statistics*, *23*, 1–18.

Bulla, J., & Bulla, I. (2006). Stylized facts of financial time series and hidden semi-Markov models. *Computational Statistics & Data Analysis*, *51*, 2192–2209.

Bulla, J., Mergner, S., Bulla, I., Sesboue, A., & Chesneau, C. 2011. Markov-switching asset allocation: do profitable strategies exist? *Journal of Asset Management* (in press).

Cappe, O., Moulines, E., & Ryden, T. (2005). Inference in hidden Markov models. In *Springer series in statistics*. New York: Springer.

Chung, H., Walls, T., & Park, Y. (2007). A latent transition model with logistic regression. *Psychometrika*, *72*, 413–435.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1978). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, *39*, 1–38. Series B Methodological.

Dutilh, G., Wagenmakers, E.-J., Visser, I., & van der Maas, H. L. J. (2011). A phase transition model for the speed–accuracy trade-off in response time experiments. *Cognitive Science*, *35*, 211–250.

Ephraim, Y., & Merhav, N. (2002). Hidden Markov processes. *IEEE Transactions on Information Theory*, *48*, 1518–1569.

Flexer, A., Sykacek, P., Rezek, I., & Dorffner, G. (2002). An automatic, continuous and probabilistic sleep stager based on a hidden Markov model. *Applied Artificial Intelligence*, *16*, 199–207.

Forney Jr, G. D. (1973). The viterbi algorithm. *Proceedings of the IEEE*, *61*, 268–278.

Frühwirth-Schnatter, S. (2006). Finite mixture and Markov switching models. In *Springer series in statistics*. Springer.

Ghahramani, Z. (2001). An introduction to hidden Markov models and Bayesian networks. *IJPRAI*, *15*, 9–42.

Ghysels, E. (1994). On the periodic structure of the business cycle. *Journal of Business and Economic Statistics*, *12*, 289–298.

Giudici, P., Ryden, T., & Vandekerkhove, P. (2000). Likelihood-ratio tests for hidden Markov models. *Biometrics*, *56*, 742–747.

Guedon, Y. (2007). Exploring the state sequence space for hidden Markov and semi-Markov chains. *Computational Statistics & Data Analysis*, *51*, 2379–2409.

Hamaker, E. (2009). Using information criteria to determine the number of regimes in threshold autoregressive models. *Journal of Mathematical Psychology*, *53*, 518–529.

Hamilton, J. (1989). A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica: Journal of the Econometric Society*, *57*, 357–384.

Harte, D. 2010. HiddenMarkov: Hidden Markov Models. *R* package version 1. 3–1.

Hughes, J., & Guttorp, P. (1994). A class of stochastic models for relating synoptic atmospheric patterns to regional hydrologic phenomena. *Water Resources Research*, *30*, 1535–1546.

Hughes, J., Guttorp, P., & Charles, S. (1999). A non-homogeneous hidden Markov model for precipitation occurrence. *Journal of the Royal Statistical Society: Series C Applied Statistics*, *48*, 15–30.

Jackson, C. 2010. msm: Multi-state Markov and hidden Markov models in continuous time. *R* package version 0.9.7.

Jelinek, F. (1976). Continuous speech recognition by statistical methods. *Proceedings of the IEEE*, *64*, 532–556.

Juang, B., & Rabiner, L. (1990). The segmental K-means algorithm for estimating parameters of hidden Markov models. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, *38*, 1639–1641.

Kaplan, D. (2008). An overview of Markov chain methods for the study of stage-sequential developmental processes. *Developmental Psychology*, *44*, 457–467.

Kemeny, J. G., & Snell, J. (1960). *Finite Markov chains*. Van Nostrand: Princeton.

Kim, C.-J. (1994). Dynamic linear models with Markov-switching. *Journal of Econometrics*, *60*, 1–22.

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

Krogh, A. (1998). An introduction to hidden Markov models for biological sequences. In S. L. Salzberg, D. B. Searls, & S. Kasif (Eds.), *Computational methods in molecular biology* (pp. 45–63). Amsterdam: Elsevier, (chapter 4).

Langeheine, R. (1988). Manifest and latent Markov chain models for categorical panel data. *Journal of Educational and Behavioral Statistics*, *13*, 299.

Langeheine, R., & Van de Pol, F. (1990). A unifying framework for Markov modeling in discrete space and discrete time. *Sociological Methods and Research*, *18*, 416–441.

Langeheine, R., & Van de Pol, F. (2000). Fitting higher order Markov chains. *Methods of Psychological Research Online*, *5*, 32–55.

Leroux, B. G., & Puterman, M. L. (1992). Maximum-penalized-likelihood estimation for independent and Markov-dependent mixture models. *Biometrics*, *48*, 545–548.

Liechty, J., Pieters, R., & Wedel, M. (2003). Global and local covert visual attention: evidence from a Bayesian hidden Markov model. *Psychometrika*, *68*, 519–541.

Lystig, T. C., & Hughes, J. P. (2002). Exact computation of the observed information matrix for hidden Markov models. *Journal of Computational and Graphical Statistics*,.

Mackay, R. J. (2002). Estimating the order of a hidden Markov model. *Canadian Journal of Statistics*, *30*, 573–589.

Mackay Altman, R. (2004). Assessing the goodness-of-fit of hidden Markov models. *Biometrics*, *60*, 444–450.

McLachlan, G. J., & Krishnan, T. (1997). *The EM algorithm and extensions*. New York: John Wiley & sons.

McLachlan, G. J., & Peel, D. (2000). Finite mixture models. In *Wiley series in probability and mathematical statistics*. New York etc: Wiley-Interscience.

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

Molenaar, P. (2004). A manifesto on psychology as ideographic science: bringing the person back into scientific psychology, this time forever. *Measurement*, *2*, 201–218.

Otterpohl, J. (2001). A constrained HMM-based approach to the estimation of perceptual switching dynamics in pigeons. *Neurocomputing*, *38–40*, 1495–1501.

Pew, R. (1969). The speed–accuracy operating characteristic. *Acta Psychologica*, *30*, 16–26.

Qin, F. (2004). Restoration of single-channel currents using the segmental k-means method based on hidden Markov modeling. *Biophysical Journal*, *86*, 1488–1501.

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

R Development Core Team 2010. *R*: A language and environment for statistical computing. *R* Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Schmittmann, V. D., Dolan, C. V., van der Maas, H. L. J., & Neale, M. C. (2005). Discrete latent Markov models for normally distributed response data. *Multivariate Behavioral Research*, *40*, 461–488.

Schmittmann, V. D., Visser, I., & Raijmakers, M. E. J. (2006). Multiple learning modes in the development of rule-based category-learning task performance. *Neuropsychologia*, *44*, 2079–2091.

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

Taramasco, O. 2009. RHmm: hidden Markov models simulations and estimations. *R* package version 1.3.1.

Titman, A. C., & Sharples, L. D. (2008). A general goodness-of-fit test for Markov and hidden Markov models. *Statistics in Medicine*, *27*, 2177–2195.

Turner, R. (2008). Direct maximization of the likelihood of a hidden Markov model. *Computational Statistics & Data Analysis*, *52*, 4147–4160.

Vermunt, J. K., Langeheine, R., & Böckenholt, U. (1999). Discrete-time discrete-state latent Markov modles with time-constant and time-varying covariates. *Journal of Educational and Behavioral Statistics*, *24*, 179–207.

Visser, I. (2010). Book review of Zucchini & MacDonald: hidden Markov models for time series: an introduction using *R*. *Journal of Mathematical Psychology*, *54*, 509–511.

Visser, I., Raijmakers, M. E. J., & Molenaar, P. C. M. (2000). Confidence intervals for hidden Markov model parameters. *British Journal of Mathematical and Statistical Psychology*, *53*, 317–327.

Visser, I., Raijmakers, M. E. J., & Molenaar, P. C. M. (2002). Fitting hidden Markov models to psychological data. *Scientific Programming*, *10*, 185–199.

Visser, I., Raijmakers, M. E. J., & Molenaar, P. C. M. (2007). Characterizing sequence knowledge using online measures and hidden Markov models. *Memory & Cognition*, *35*, 1502–1517.

Visser, I., & Speekenbrink, M. (2010). depmixs4: An *R*-package for hidden Markov models. *Journal of Statistical Software*, *36*, 1–21. *R* package, current version available from CRAN.

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, *13*, 260–269.

Wickelgren, W. (1977). Speed–accuracy tradeoff and information processing dynamics. *Acta Psychologica*, *41*, 67–85.

Wickens, T. D. (1982). *Models for behavior: stochastic processes in psychology*. San Francisco: W. H. Freeman and Company.

Wiggins, L. M. (1973). *Panel analysis*. Elsevier Scientific Publishing Company.

Zucchini, W. (2000). An introduction to model selection. *Journal of Mathematical Psychology*, *44*, 41–61.

Zucchini, W., & MacDonald, I. (2009). *Hidden Markov models for time series: An introduction using R. Number 110 in monographs on statistics and applied probability*. Boca Raton: CRC Press.

Zucchini, W., Raubenheimer, D., & MacDonald, I. L. (2008). Modeling time series of animal behavior by means of a latent-state model with feedback. *Biometrics*, *64*, 807–815.