Keeping an Eye on Hidden Markov Models in Gaze Data Classification

Malte Valentin Lueken[1]

[1] University of Amsterdam

Author Note

Add complete departmental affiliations for each author here. Each new line herein must be indented, like this line.

Enter author note here.

Abstract

One or two sentences providing a **basic introduction** to the field, comprehensible to a scientist in any discipline.

Two to three sentences of **more detailed background**, comprehensible to scientists in related disciplines.

One sentence clearly stating the **general problem** being addressed by this particular study.

One sentence summarizing the main result (with the words "**here we show**" or their equivalent).

Two or three sentences explaining what the **main result** reveals in direct comparison to what was thought to be the case previously, or how the main result adds to previous knowledge.

One or two sentences to put the results into a more **general context**.

Two or three sentences to provide a **broader perspective**, readily comprehensible to a scientist in any discipline.

Keywords: keywords

Word count: X

Keeping an Eye on Hidden Markov Models in Gaze Data Classification

Algorithm Development

As part of answering my research questions, I developed an algorithm named *gazeHMM* to classify gaze data into discrete eye movement events. The following section describes the development of gazeHMM and its final version that has been used to obtain results. Technical details can be found in the Technical Appendix.

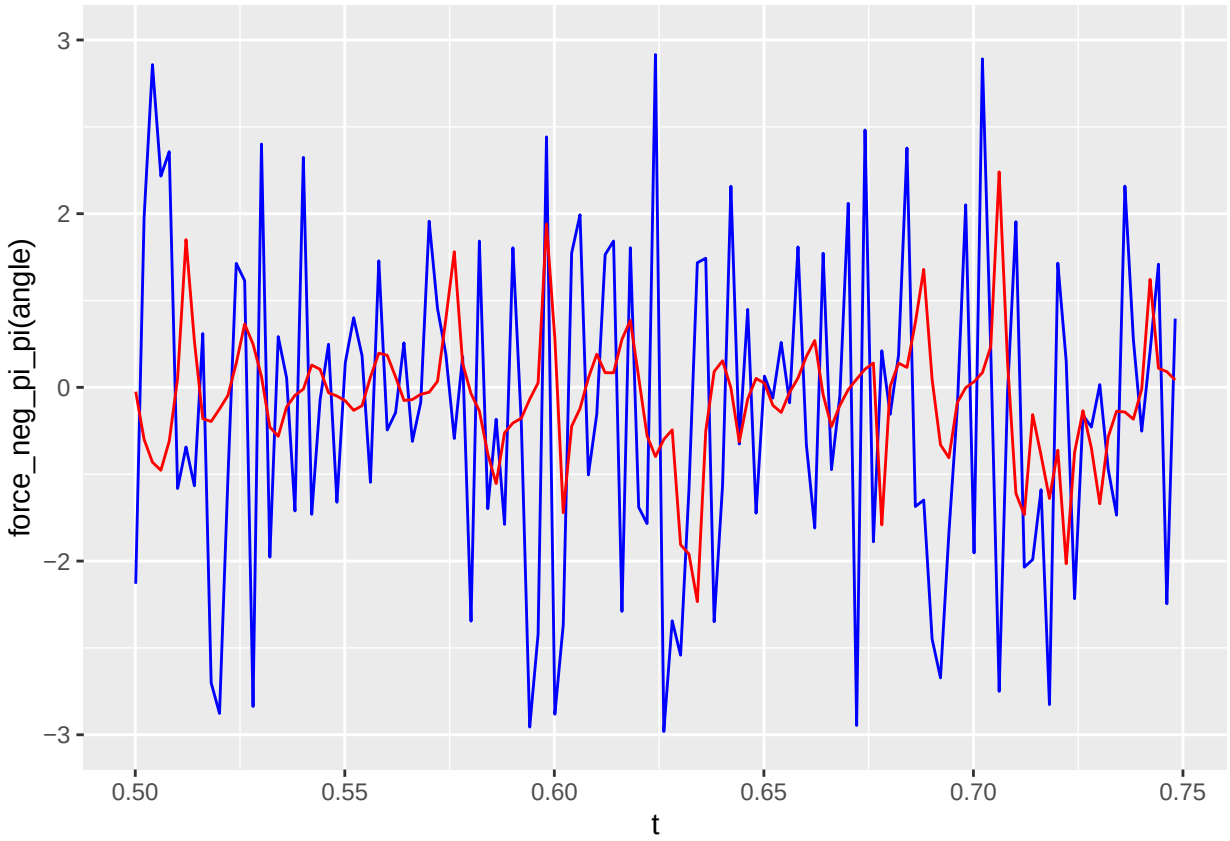
Eye movement metrics

Many different metrics can be used to describe gaze data and eye movement events (Zemblys et al.). For classifcation, the goal is to find those metrics that separate the gaze data samples belonging to different events the best. However, many metrics rely on thresholds or window ranges that have to be set by the user (Zemblys et al.). Since not relying on parameter settings by the user was one criterion for choosing metrics, I excluded them from the decision process. Finally, I chose to use three metrics that have been used in previous algorithms: velocity, acceleration, and sample-to-sample angle (e.g., Larsson et al., Pekkanen and Lappi). The reasons to chose these three are mainly theoretical. Figure X displays the theoretically expected distributions of samples belonging to different eye movement events. Fixations typically inherit samples with low velocity and acceleration (CITE). Due to tremor, the angle between samples should not follow any direction but a random walk (CITE). In contrast, saccade samples usually have a high velocity and acceleration and follow the same direction (CITE). PSO samples tend to have moderate velocity and high acceleration since they occur between high velocity saccades and low velocity fixations. They are specifically distinguished by their change in direction around 180° (CITE). Lastly, smooth pursuit samples have a moderate velocity but low acceleration (due to the smoothness) and like saccades they follow the same direction (CITE). In sum, velocity, acceleration, and sample-to-sample angle should theoretically separate eye

movement events well.

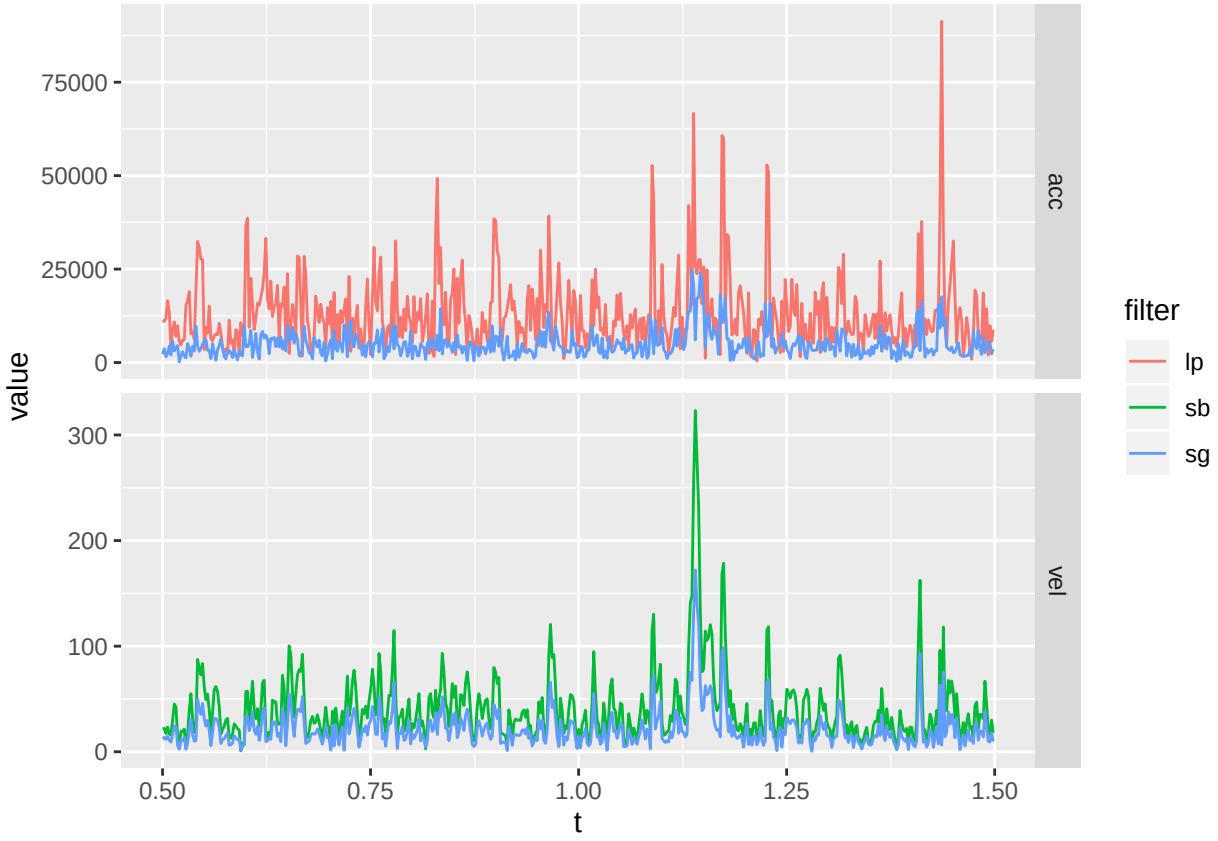
Fitering and smoothing

Like metrics, many methods can be used to filter or smooth gaze data before, while, or after computing eye movement metrics. Their purpose is to remove noise or artifacts from the gaze data that could distort the classification (CITE). Filtering (smoothing) methods that are applied before computing the eye movement metrics target the recorded gaze position. Unfortunately, this might also erase differences in the metrics between samples. For instance, tremor movements during fixations would be filtered out and in consequence, the angle of the samples would stay the same. This would make it harder for the algorithm to separate fixations from smooth pursuits. The same phenomenon would apply to PSOs and saccades (see Figure X).



Instead of filtering before computing the metrics, I decided to combine both in one

step. Previous algorithms have used two methods that both filter (smooth) and compute derivatives of the gaze position (i.e., velocity and acceleration). (CITE) compute the first and second discrete derivative (similar to a Sobel and Laplace filter, respectively, CITE) of the gaze position to be used by their algorithm. Similarly, (CITE) use a Savitzky-Golay (SG) filter (CITE) to estimate velocity and acceleration from a gaze signal. Figure X displays velocity and acceleration signals obtained by both methods. They follow a similar trend, but the SG velocity signal is less noisy and consequently, the SG acceleration signal is lower than the Laplace acceleration signal. I chose to implement the SG filter to compute velocity and acceleration signals, because it filters out more noise than the discrete derivatives but still preserves the edges in the signal to distinguish between events. To preserve motor noise in the change in angle signal, gazeHMM uses a Sobel filter to compute the first discrete derivative of sample angle.



The generative model

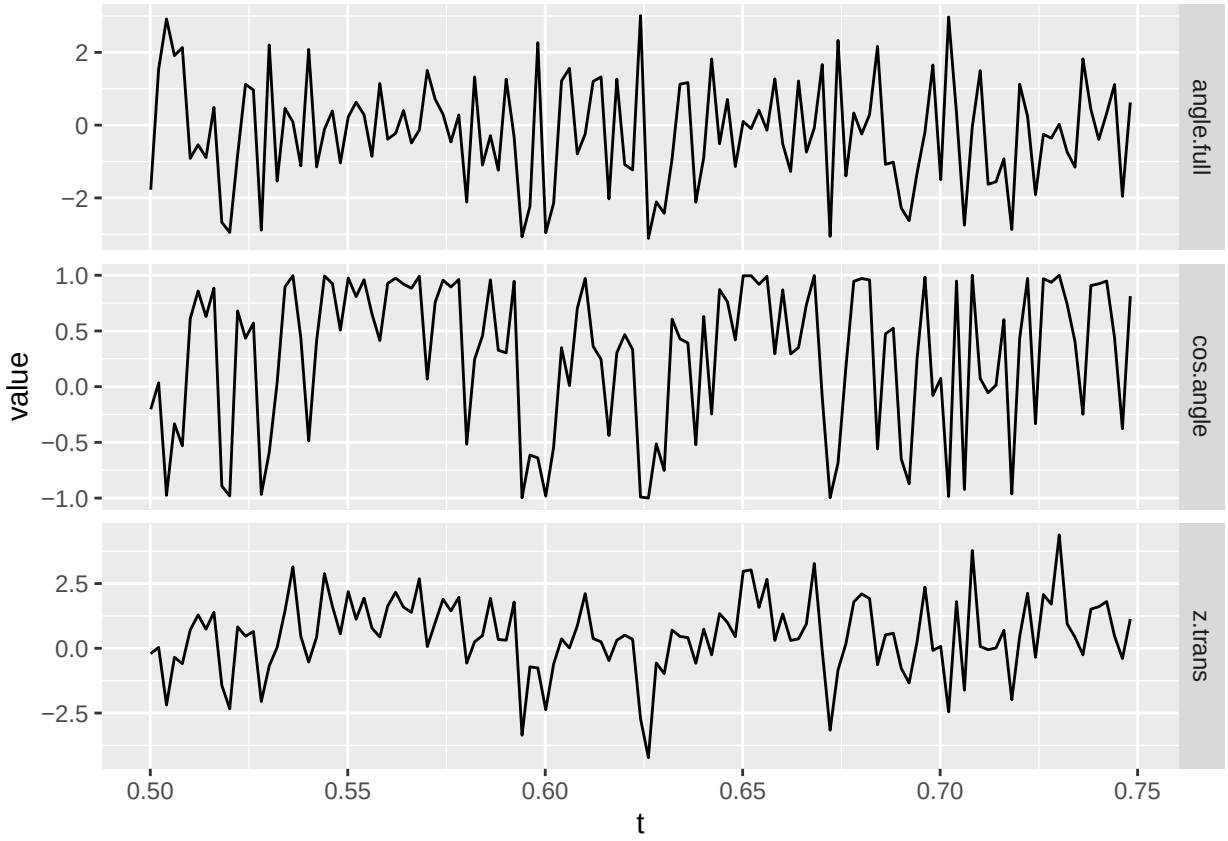
The generative model underlying gazeHMM is a multivariate hidden Markov model (HMM). It can have between two and four states that correspond to different eye movement events (see Table X). Thus, users can choose wether they would like to classify only fixations and saccades, or additionally PSOs and/or smooth puruits.

In general, HMMs consist of three submodels: An initial state model, a transition model, and a response model (CITE). In gazeHMM, the response model consists of three response variables which are the velocity and acceleration signals obtained by the SG filter and the change in angle signal obtained by the Sobel filter. The response variables are treated as independent conditional on the states. I acknowledge that conditional independence might not accurately resemble the relationship between velocity and acceleration (which are naturally correlated). This step was taken merely to keep the HMM simple and identifyable.

Previous algorithms using HMMs have used Gaussian distributions to describe velocity and acceleration signals (sometimes after log-transforming them). However, several reasons speak against choosing the Gaussian: First, both signals are usually positive (depending on the computation) and due to noise above zero. Second, the distributions of both signals appear to be positively skewed conditionally on the states and, third, have variances increasing with their mean. Thus, instead of using the Gaussian, it could be more appropriate to describe velocity and acceleration with a distribution that follows these three properties. In gazeHMM, I use gamma distributions with a shape and scale parametrization for this purpose. However, it has to be noted the gamma was choosen because of convenience and the best fitting distribution might be different between subjects and tasks.

To model the sample-to-sample angle, CITE described the Fisher-transformed cosine of sample angle with a Gaussian distribution. This approach can be problematic because

the cosine of sample angle only ranges between -1 and 1 (corresponding to angles of π and 0). Thus, it only operates on half of the unit circle and $\cos(\frac{1}{2}\pi) = \cos(\frac{3}{2}\pi)$. This property can be desirable if one is only interested in expression angles between 0 and π (which is in event classification). However, the sample-to-sample angle distributions for the different events will also be skewed having their modes at 0 (for saccades and smooth pursuits) or π (for PSOs). This consequence makes the Fisher-transformation and Gaussian not the best choices. Instead, I pursued a novel approach in gazeHMM: Using a mixture of von-Mises distributions (with a mean and concentration parameter) and a uniform distribution to model the derivative of sample angle. Both the distributions and the metric operate on the full unit circle (i.e., between 0 and 2π) which should lead to rather symmetric distributions. Moreover, the uniform distribution can distinguish fixations from the other events.



To describe the initial state model and the transition model I used multinomial distributions (with a category for each state). For all three submodels of the HMM, only

parameter intercepts were estimated.

Optimization and classification

The paramters of the HMM are estimated through maximum likelihood using an expectation-maximization (EM) algorithm (EM CITE). The EM is generally suitable to estimate likelihoods with missing variables. For HMMs, it imputes missing with expected values and iteratively maximizes the joint likelihood of parameters conditional on the observed data (velocity, acceleration, and sample-to-sample angle) and the expected hidden states (eye movement events; Visser & Speekenbrink). The sequence of hidden states is estimated through the Viterbi algorithm (CITE) by maximizing the posterior state probability (Visser & Speekenbrink). Parameters of the response distributions (except for the uniform distribution) were optimized on the log-scale (except for the mean parameter of the von-Mises distribution) using a spectral projected gradient method (CITE) and Barzilai-Borwein steplengths (CITE).

Postprocessing

Eye movement events usually obey certain theoretical (e.g., PSOs only follow saccades) and practical constraints (e.g., eye-tracking resolutions) (CITE). However, constraining the parameter estimation in gazeHMM led mostly to unreasonable results or did not improve the classification. For example, constraining the transition probabilities for PSOs to turn into non-saccade events to zero often caused PSOs not to appear in the classification output at all. Moreover, I could not explicitly control the duration of events in the HMM which ocassionally led to unreasonable short events. In consequence, I implemented a postprocessing routine in gazeHMM to heuristically compensate for such violations. This routine relabels one-sample fixations and smooth pursuits, saccades with a duration below a minimum threshold, and PSOs that follow non-saccade events as the previous event.

Implementation

The algorithm is implemented in R (version 3.6.3; CITE) and uses the packages signal (CITE) to compute velocity and acceleration signals, dplyr (CITE) to calculate sample-to-sample angles, depmixS4 (CITE) for the HMM, and BB (CITE) for Barzilai-Borwein spectral projected gradient optimization. The algorithm will be available on Github (LINK).

Simulation Study Design

Results

Simulation study

With a few exceptions, the simulation study revealed that the HMM recovered parameters well and accurately estimated the true state sequences. The most critical decrease in recovery occured when noise was added to the data generated by models including smooth pursuits. In the noise condition, scale parameters of gamma distributions were often badly recovered. Higher sample sizes slightly improved parameter recovery, but neither variation in starting values nor missing data affected it. In contrast, the accuracy of the model was linearly descreasing with more data missing but not influenced by manipulating parameters, noise, or starting value variation. Adding more states to the HMM generally decreased the parameter recovery and the accuracy. The full results are described in the following sections.

Parameter variation. In the first part of the simulation, I examined how varying the parameters in the HMM affects the deviation between the true and estimated parameters and the accuracy between the true and estimated state sequence. Figure X displays the RMdSPD between true and estimated parameters depending on which parameter has been manipulated in the HMM with two states. The RMdSPD was below

0.1 for all estimated and manipulated parameters (except the irrelevant initial state probabilities). Thus, no matter which relevant parameter changed in the HMM, the median deviation did not exceed 10% of the true parameter value. The regressions between manipulated true and estimated parameters for two states are shown in Figures X and X. With one outlier at parameter $\alpha_{acc;1}$, the estimated parameters matched the true parameters very closely: Changing the true parameters led to linear changes in the estimated parameters. However, the deviation between true and estimated response parameters seemed to increase slighlty with parameter magnitude. Considering accuracy, Figure X displays Cohen's kappa between true and estimated hidden state sequences for two states. With two exceptions, kappa values were almost one, indicating nearly perfect agreement. In sum, the HMM with two states recovered parameters and hidden states very well with outliers only occuring rarely.

For three states, the RMdSPD is shown in Figure X. When response parameters (other than $a_{i=j}$) were manipulated, the RMdSPDs for a_{12} and a_{31} were consistently below 0.5. Varying κ in states two and three led to RMdSPDs below 0.5 in the respective states. Otherwise, RMdSPDs were consistently lower than 0.1, indicating a median deviation between true and estimated parameters below 10% of the true parameter. Inspecting the regressions between manipulated true and estimated parameters (see Figures X and X) revealed a strong and unbiased linear relationship (intercepts close to zero and slopes close to one). Again, the deviation seemed to increase with true parameter magnitude (reverse for kappas). In contrast to the two-state HMM, larger deviatons and more outliers were observed. Cohen's kappa values for the three-state HMM are presented in Figure X. For most estimated models, the kappas between true and estimated state sequences were above 0.95, indicating almost perfect agreement. However, for some models I observed kappas clustered around zero or -0.33, suggesting that state labels switched (CITE). To summarize, the three-state HMM displayed a slightly worse parameter recover with more outliers than the two-state model but shows an almost equally good accuracy.

The RMdSPDs for the four-state HMM is illustrated in Figure X. For estimated transition probabilities and α_{vel} and β_{vel} parameters in states one and four, RMdSPDs were below 0.5. Estimated kappa parameters in smooth pursuits were also often below 0.5 when paramteres in states two, three, and four were varied. Otherwise, RMdSPDs were below 0.1, indicating a median deviation below 10% of the true parameter. Looking at the regressions between true and estimated paramters, Figure X shows strong and unbiased relationships but larger deviations and more outliers than in the previous models, especially for states one and four. Accuracy indicated by Cohen's kappa ranged between 0.6 and 0.9 for the majority of models, meaning moderate to almost perfect agreement between true and estimated state sequences. Here, some kappa values clustered around 0.25 and zero, which again can be interpreted as the results of label switching. Summarizing, the parameter recovery for the four-state HMM was slightly worse with even more outliers compared to the two- and three-state models. Especially the recovery of transition parameters in states corresponding to fixations and smooth pursuits decreased.

Overall, simulations in part one demonstrated that the HMM recovered parameters at least moderately well when parameters were manipulated (all RMdSPDs below 0.5). The HMM estimated state sequences very accurately. Adding states to the model decreased the accuracy and recovery slightly, leading to more outliers, especially when smooth pursuits were added.

Sample size and noise variation. In the second part, I varied the sample size of the HMM and added noise to the generated data. For the two-state HMM, the RMdSPDs were above 0.5 for β_{vel} and β_{acc} in both states (see Figure X). The other estimated parameters showed RMdSPDs close to or below 0.1. Increasing the sample size seemed to improve RMdSPDs for most parameters slightly. For β_{vel} and β_{acc} in both states, models with 2500 samples had the lowest RMdSPDs. Accuracy measured by Cohen's kappa was almost perfect with kappa values very close to one (see Figure X, left plot). To conclude, adding noise did not affect the parameter recover or accuracy of the two-state HMM,

whereas increasing the sample size improved the recovery slightly.

The RMdSPDs for the three-state HMM and β_{vel} and β_{acc} were above 0.5 in all three states. Again, the other estimated parameters were below or close to 0.1, only a_{12} and a_{31} with 500 samples were closer to 0.5. For most parameters in all three states, higher sample sizes had lower RMdSPDs. The accuracy of the estimated models was almost perfect with most kappa values above 0.95 (see Figure X, middle plot). Several outliers clustered around kappas of zero and -0.33, signaling label switching. For the three-state HMM, adding noise led to a slightly worse recovery and accuracy overall. However, scale parameters of gamma distributions in all three states were only badly recovered.

RMdSPDs regarding the four-state HMM are displayed in Figure X. For states one and four, values for most parameters (including all transition probabilites) were above 0.5. Similarly, RMdSPDs for β_{vel} and β_{acc} in states two and three were above 0.5. For states two and three, higher samples sizes showed slightly lower RMdSPDs. As in the previous part, most Cohen's kappa values ranged between 0.6 and 0.9, meaning substantial to almost perfect agreement between true and estimated states (Figure X, right plot). Multiple kappa values clustered around 0.25 or zero, which can be explained by label switching. In summary, adding noise to data generated by the four-state HMM heavily decreased the recovery for most parameters in states corresponding to fixations and smooth pursuits. For saccade and PSO states, only scale parameters of gamma distributions were badly recovered, but increasing the sample size slightly improved the recovery.

In general, the HMM recovered parameters well despite noise being added to the data. However, when smooth pursuits were added to the model, the parameter recovery for fixation and smooth pursuit states substantially decreased. When PSOs or smooth pursuits were included, scale parameters of gamma distributions were badly recovered. Increasing the samples in the HMM slightly improved the recovery of most parameters. The accuracy of the model was slightly lowered when including more states, but it was neither affected

by the range of noise τ_{noise} nor the sample size.

Variation of starting values. The third part of the simulation investigated how increasing the variation in generating starting values affected parameter recover and accuracy of the HMM. For the two-state HMM, all parameters displayed RMdSPDs lower than 0.1 (see Figure X). Cohen's kappa values were slightly lower than 1, indicating almost perfect accuracy (see Figure X). For the three-state HMM, RMdSPDs were lower than 0.1 except for a_{12} and a_{31}, which were below 0.5. As in previous parts, Cohen's kappa values were mostly above 0.95 (almost perfect accuracy) with exceptions clustering around zero and -0.33. Regarding the four state-HMM, RMdSPS of transition probabilities for states one and four were clustered above 0.1, other estimated parameters in the model showed values below 0.1. The HMM showed substantial to almost perfect accuracy, as Cohen's kappa values were mostly above 0.8 with a few values clustering around 0.6, 0.25, and zero. In conclusion, the parameter recovery of the model worked very well. State sequences were accurately estimated and more states being included in the model slightly decreased the accuracy. Increasing the variation in starting values for parameter estimation did neither affect the parameter recovery nor accuracy.

Since the manipulation in this part was not effective, I decided to explore post-hoc whether clusters of low Cohen's kappa values were due to label switching. According to Visser and Speekenbrink (CITE), label switching occurs when exchanging the order of the states leads to equally likely models. Thus, the model is accurate but has oppositely estimated states compared to the true states. In this case, an HMM with three states that is perfectly accurate but has one label swtiched would have a Cohen's kappa of 0. To test if label switching occured, I manually switched one or two labels post-hoc (see Figure X). It can be seen that this approach resolved low accuracy clusters for three and four states.

Missing data. In the last part, intervals of varying length in the data were set to be missing. Regarding parameter recovery, RMdSPDs were almost exactly mirroring those in the previous part for two, three, and four states. Looking at accuracies, Figure X

illustrates an interaction between the length of and amount of missing data intervals. Cohen's kappa values linearly decreased with the length of missing data intervals and the decrease became linearly steeper when more intervals were included. For two and three state models, kappas started at almost one and decreased to 0.6 for five missing intervals, indicating substantial to almost perfect accuracy. For the four-state model, kappas started around 0.85 and decreased to 0.5 for five missing intervals, suggesting moderate to substantial accuracy. As in previous parts, a few kappa values clustered around zero and -0.33 for the three-state model and around 0.6, 0.25, and zero for the four-state model. In total, missing data did not affect the parameter recovery but linearly decreased the accuracy of the model up to a moderate extent.

Technical Appendix