Keeping an Eye on Hidden Markov Models in Gaze Data Classification

Malte Lüken[1], Šimon Kucharský[1], & Ingmar Visser[1]

[1] Department of Psychology, University of Amsterdam. Amsterdam, The Netherlands

Author Note

Correspondence concerning this article should be addressed to Malte Lüken, Speelmanstraat 7 1063ZC Amsterdam, The Netherlands. E-mail: malte_lueken@arcor.de

Abstract

Eye-tracking allows researchers to infer cognitive processes from eye movements that are classified into distinct events. Parsing the events is typically done by algorithms. Previous algorithms have successfully used hidden Markov models (HMMs) for classification but still inhere several weaknesses. To adress these weaknesses, we developed gazeHMM, an algorithm that uses an HMM as a generative model, has no critical parameters to be set by users, and does not require human coded data as input. The algorithm classifies gaze data into fixations, saccades, and optionally PSOs and smooth pursuits. We evaluated the performance of gazeHMM in a simulation study and on two benchmark data sets. The simulation study showed that gazeHMM successfully recovered HMM parameters and hidden state sequences. Critical exceptions for good recovery were including the smooth pursuit state in the generative model and noisy data. The performance on benchmark data was assessed by model comparison and agreement of the classification to human coding compared to other algorithms. The model comparison yielded preferred generative models for the benchmark data with more events than we expected. Nevertheless, gazeHMM showed relatively high similarity to human coding and outperformed the other algorithms in some agreement criteria. For dynamic stimuli, our algorithm tended to switch rapidly between fixations and smooth pursuits. Thus, we recommend to use gazeHMM for classifiying smooth pursuits only in explorative settings. Future HMM algorithms could use covariates to better capture eye movement processes and explicitly model event durations to improve the classification of smooth pursuits.

Keywords: eye movements; eye-tracking; parameter recovery; dependent mixture models

Word count: 8861

Keeping an Eye on Hidden Markov Models in Gaze Data Classification

Introduction

Eye-tracking is typically used to study cognitive processes involving attention and information search based on recorded gaze position (Schulte-Mecklenbeck et al., 2017). Before these processes can be studied, the raw gaze data is classified into events that are distinct in their physiological patterns (e.g., duration), underlying neurological mechanisms, or cognitive functions (Leigh & Zee, 2015). Basic events are fixations, saccades, smooth pursuits, and post-saccadic oscillations (PSOs). Classifying raw eye-tracking data into these events reduces their complexity and is usually the first step towards cognitive interpretation (Salvucci & Goldberg, 2000). The classification is typically done by algorithms, which is considered faster, more objective, and reproducible compared to human coding (Andersson, Larsson, Holmqvist, Stridh, & Nyström, 2017). Hein and Zangemeister (2017) give a comprehensive overview of different classification algorithms (for a structured review on classifying saccades, see also Stuart et al., 2019).

Most of these classification algorithms use non-parametric methods to differentiate between eye movement events. For instance, the classic "Velocity-Threshold" algorithm classifies samples with a velocity above a fixed threshold as saccades (**???**). More recently, supervised statistical learning techniques have been successfully used to parse eye movement data. For example, Bellet, Bellet, Nienborg, Hafed, and Berens (2019) trained a convolutional neural network on eye-tracking data from humans and macaques and achieved classification results that were similarly accurate to those of human coders. However, these supervised methods have several disadvantages: First, they require labeled data (usually labeled by humans) as input. Second, they often rely on hyperparameters which are difficult to theoretically justify. Third, their parameters are usually difficult to interpret: Most supervised methods operate in a black-box fashion, which implies that they don't make any explicit assumptions about the data generating process. Oppositely, several

algorithms using unsupervised techniques have been developed. They typically don't need to be trained on labeled data and are thus more flexible in classifying data from different individuals, tasks, or eye-trackers.

Besides differentiating between supervised and unsupervised methods, algorithms can vary in whether they are explicitly modeling the data generating process and are thus able to simulate new data. To our knowledge, these *generative models* have been rarely used to classify eye movement data (cf. Mihali, Opheusden, & Ma, 2017, p. @Wadehn2019). However, they come with various advantages and we will summarize the two most important for eye movement classification here (for an overview, see **???**).

First, generative models allow to not only predict (classify) but also explain eye-tracking phenomena. For instance, they can inhere parameters that explain why oscillations only occur after but not before saccades. When the goal is to understand eye movements and improve their classification based on this understanding, this aspect is clearly an advantage over non-parametric or supervised methods. Second, generative models can challenge prevailing theoretical assumptions and bring up new research questions. For example, they might suggest that oscillations also occur before saccadic eye movements (as has been mentioned by Nyström & Holmqvist, 2010) or that the assumption of discrete eye movements does not hold [as has been discussed in Andersson et al. (2017); maybe others].

One class of generative models that are used in eye movement classification are hidden Markov models (HMMs). HMMs estimate a sequence of hidden states (i.e., a discrete variable that cannot be directly observed) that evolves parallel to the gaze signal. Each gaze sample depends on its corresponding state. Each state depends on the previous but not on earlier states of the sequence (Zucchini, MacDonald, & Langrock, 2016).

In the context of eye movements, HMMs are suitable models because the hidden states can be interpreted as eye movement events and gaze data are dependent time series (i.e., one gaze sample depends on the previous). Further, HMMs can be viewed as

unsupervised models that can learn the hidden states and parameters of the emission process from the observed data alone, and as such do not in principle need training labeled data as it would be the case for supervised methods.

On this basis, several classification algorithms using HMMs have been developed: One instance is described in Salvucci and Goldberg (2000) and combines the HMM with a fixed threshold approach (named "Identification by HMM" [I-HMM]). Samples are first labeled as fixations or saccades, depending on whether their velocity exceeds a threshold, and then reclassified by the HMM. Pekkanen and Lappi (2017) developed an algorithm that filters the position of gaze samples through naive segmented linear regression (NSLR). The algorithm uses an HMM to parse the resulting segments into fixations, saccades, smooth pursuits, and PSOs based on their velocity and relative angle (named NSLR-HMM). Another version by Mihali et al. (2017) uses a Bayesian HMM to separate microsaccades (short saccades during fixations) from motor noise based on sample velocity (named "Bayesian Microsaccade Detection" [BMD]). Moreover, Houpt, Frame, and Blaha (2018) developed a hierarchical approach that describes sample velocity and acceleration through an autoregression (AR) model, computes the regression weights through an HMM, and estimates the number of events with a beta-process (BP) from the data (named BP-AR-HMM).

Several studies have tested the performance of HMM algorithms against other classification methods: I-HMM has been deemed as robust against noise, behaviorally accurate, and showing a high sample-to-sample agreement to human coders (Andersson et al., 2017; Komogortsev, Gobert, Jayarathna, Koh, & Gowda, 2010; Salvucci & Goldberg, 2000). However, the agreement was lower when compared to an algorithm using a Bayesian mixture model (Kasneci, Kasneci, Kübler, & Rosenstiel, 2014; Tafaj, Kasneci, Rosenstiel, & Bogdan, 2012). NSLR-HMM showed even higher agreement to human coding than I-HMM but was outperformed by a recently developed algorithm using convolutional neural networks (Bellet et al., 2019; Pekkanen & Lappi, 2017).

In sum, HMMs seem to be a promising method for classifying eye movements. However, the existing HMM algorithms each have their weaknesses that could be improved upon.

First, I-HMM relies on setting an appropriate threshold to determine the initial classification, which can distort the results (Blignaut, 2009; Komogortsev et al., 2010; Shic, Scassellati, & Chawarska, 2008). Second, the current implementation of NSLR-HMM requires human-coded data, which limits its applicability. It also inherits fixed parameters that prevent the algorithm to adapt to the individual- or task-specific signals (contrary to the authors, we argue that this is a disadvantage). Third, BMD limits the classification to microsaccades which are irrelevant in many applications and sometimes even considered as noise (Duchowski, 2017). The opposite problem was observed for BP-AR-HMM: It tends to estimate an unreasonable number of events from the data of which many are considered as noise events. Therefore, the authors suggest using it as an exploratory tool followed by further event classification (Houpt et al., 2018).

For this article, we developed a novel algorithm for classifying gaze data, named gazeHMM, that relies on an HMM as a generative model and addresses the weaknesses of previous algorithms. The development was guided by four goals: (a) the algorithm should not require parameter settings through the user (e.g., thresholds) or (b) human-labeled data as input; (c) it should cover the most relevant eye movement events, namely fixations, saccades, smooth pursuit, and PSOs; (d) it should confirm rather than explore the presence of events in the data.

The Algorithm

Preprocessing

Algorithms require variables that describe gaze data (hereafter called *eye movement metrics*) to classify them into events. Many eye movement metrics have been proposed and

used in previous algorithms (for examples, see Zemblys, Niehorster, Komogortsev, & Holmqvist, 2018; Andersson et al., 2017). However, many of them rely on thresholds or window ranges that have to be set by the user (e.g., the distance between the mean position in a 100 ms window before and after each sample, see Olsson, 2007). This can be problematic because such parameters are often set without theoretical justification and differ substantially between metrics or heavily depend on the eye-tracker's sampling frequency (Andersson et al., 2017). In gazeHMM, we used velocity, acceleration, and sample-to-sample angle (synonymous to relative or change in angle; Larsson, Nystrom, & Stridh, 2013) because they belong to the most basic metrics which do not require parameter settings.

Theoretically, these three metrics should separate eye movement events clearly. Fixations typically inherit samples with low velocity and acceleration (Larsson et al., 2013). Due to tremor, we assume that the angle between samples should not follow any direction but a random walk (Duchowski, 2017; Larsson, Nyström, Andersson, & Stridh, 2015). In contrast, saccade samples usually have a high velocity and acceleration and roughly follow the same direction. PSO samples tend to have moderate velocity and high acceleration since they occur between saccades and low velocity events (Larsson et al., 2013). They can be specifically distinguished by their change in direction clustered around 180 degrees (Pekkanen & Lappi, 2017). Importantly, the oscillations depend on the resolution of the gaze recording: Eye-trackers with higher sampling frequency yield more changes in direction and more samples in between those changes. Those samples in between typically follow the same direction. Thus, with high sampling frequencies, PSO samples might also cluster around a sample-to-sample angle of zero with outliers around 180 degrees. Lastly, smooth pursuit samples have a moderate velocity but low acceleration (due to the smoothness) and like saccades they follow the same direction (Larsson et al., 2013; Leigh & Zee, 2015). Other algorithms focus exclusively on classifying microsaccades (e.g., Mihali et al., 2017), but as stated earlier, these events were not in the scope of gazeHMM.

The velocity and acceleration signals are computed from the raw gaze position by using a Savitzky-Golay filter (Savitzky & Golay, 1964; similar to Nyström & Holmqvist, 2010). The sample-to-sample angle is calculated as:

$$\alpha(t) = \arctan\left(\frac{y_{t+1} - y_t}{x_{t+1} - x_t}\right) - \arctan\left(\frac{y_t - y_{t-1}}{x_t - x_{t-1}}\right),$$

with $\alpha(t) := \alpha(t) + 2\pi$ for $\alpha(t) < 0$, and is therefore bound between 0 and 2π.

Most of the missing data in eye movement classification are due to blinks. Therefore, the user can indicate which samples should be labeled as blinks (other missing samples are treated as noise). Often, eye-trackers record a few samples with unreasonably high velocity and acceleration before losing the pupil signal when a blink occurs. Since these samples could distort the classification of saccades in the HMM, gazeHMM removes them heuristically. Before classifying the samples, it sets all samples within 50 ms before and after blink samples as missing. The window of 50 ms was rather motivated empirically than theoretically and can bet set to any value the user considers appropriate.

The Generative Model

We denote the three eye movement metrics by *X*, *Y*, and *Z*. Each metric was generated by a hidden state variable *S*. Given *S*, the HMM treats *X*, *Y*, and *Z* as conditionally independent. Conditional independence might not accurately resemble the relationship between velocity and acceleration (which are naturally correlated). This step was merely taken to keep the HMM simple and identifiable. In gazeHMM, *S* can take one of two, three, or four hidden states. The first state always represents fixations, the second saccades, the third PSOs, and the fourth smooth pursuits. Thus, users can choose whether they would like to classify only fixations and saccades, or additionally PSOs and/or smooth pursuits.

HMMs can be described by three submodels: An initial state model, a transition model, and a response model. The initial state model contains probabilites for the first

state of the hidden sequence $\rho_i = P(S_1 = i)$, with i denoting the hidden state. In gazeHMM, the initial states are modeled by a multinomial distribution. The evolution of the sequence is in turn described by the trantision model, which comprises the probabilities for transitioning between different states in the HMM. Typically, probabilities to transition from state i to j, $a_{ij} = P(S_{t+1} = j | S_t = i)$, are expressed in matrix form (Visser, 2011):

$$\mathbf{A} = \begin{bmatrix} a_{11} & ... & a_{1j} \\ \vdots & \ddots & \vdots \\ a_{i1} & ... & a_{ij} \end{bmatrix}.$$

Again, the transition probabilities for each state are modeled by multinomial distributions.

The response model encompasses distributions describing the response variables for every state in the model. Previous algorithms 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). Second, the distributions of both signals appear to be positively skewed conditionally on the states and third, to 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, we use gamma distributions with a shape and scale parametrization for this purpose:

$$X \sim Gamma(\alpha, \beta),$$

$$Y \sim Gamma(\alpha, \beta).$$

It has to be noted that the gamma was chosen out of convenience and the best fitting distribution might be different between eye-trackers, subjects, and tasks.

To model the sample-to-sample angle, we pursued a novel approach in gazeHMM: A mixture of von Mises distributions (with a mean and concentration parameter) and a

uniform distribution:

$$Z \mid fixation \sim U(0, 2\pi)$$

$$Z \mid saccade \sim von\ Mises(\mu, \kappa)$$

$$Z \mid PSO \sim von\ Mises(\mu, \kappa)$$

$$Z \mid smooth\ pursuit \sim von\ Mises(\mu, \kappa)$$

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. Because the fixations change their direction similar to a random walk, their sample-to-sample angle can be modeled by a uniform distribution. Thus, the uniform distribution should distinguish fixations from the other events.

All three submodels taken together, the joint likelihood of the observed data and hidden states can be expressed as:

$$L(X, Y, Z, S | \lambda) = \rho_{S_1} f_{S_1}(X_1) f_{S_1}(Y_1) f_{S_1}(Z_1) \prod_{t=1}^{T-1} a_{S_t S_{t+1}} f_{S_{t+1}}(X_{t+1}) f_{S_{t+1}}(Y_{t+1}) f_{S_{t+1}}(Z_{t+1}),$$

with λ denoting the vector containing the initial state and transtition probabilities as well as the response parameters. By summing over all possible state sequences, the likelihood of the data given the HMM parameters becomes (Visser, 2011):

$$L(X, Y, Z | \lambda) = \sum_{\text{all S}} \rho_{S_1} f_{S_1}(X_1) f_{S_1}(Y_1) f_{S_1}(Z_1) \prod_{t=1}^{T-1} a_{S_t S_{t+1}} f_{S_{t+1}}(X_{t+1}) f_{S_{t+1}}(Y_{t+1}) f_{S_{t+1}}(Z_{t+1}).$$

The parameters of the HMM are estimated through maximum likelihood using an expectation-maximization (EM) algorithm (Dempster, Laird, & Rubin, 1977; McLachlan & Krishnan, 1997). The EM algorithm 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 and the expected hidden states (eye movement events; Visser, 2011). When evaluating the likelihood for missing data, gazeHMM integrates over all possible values, which results in a probability density of one. The sequence of hidden states is estimated through the Viterbi algorithm (Forney Jr, 1973; Viterbi, 1967) by maximizing the posterior state probability.

Parameters of the response distributions (except for the uniform distribution) are optimized on the log-scale (except for the mean parameter of the von Mises distribution) using a spectral projected gradient method (Birgin, Martinez, & Raydan, 2000) and Barzilai-Borwein step lengths (Barzilai & Borwein, 1988). The implementation in depmixS4 allows to include time varying covariates for each parameter in the HMM. In gazeHMM, no such covariates were included and thus, only intercepts were estimated for each parameter.

Postprocessing

After classifying gaze samples into states, gazeHMM applies a postprocessing routine to the estimated state sequence. We implemented this routine because constraining the transition probabilities for PSOs to turn into non-saccade events to zero often caused PSOs not to appear in the state sequence at all. Moreover, gazeHMM does not explicitly control the duration of events in the HMM which occasionally led to unreasonably short events. Thus, the postprocessing routine heuristically compensates 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. Samples are relabeled as the state of the previous event. Finally, samples initially indicated as missing are labeled as noise (including blinks) and event descriptives are computed (e.g., fixation duration).

The algorithm is implemented in R (version: 3.6.3; R Core Team, 2020) and uses the packages `signal` (Ligges, Short, & Kienzle, 2015) to compute velocity and acceleration signals, `depmixS4` (Visser & Speekenbrink, 2010) for the HMM, `CircStats` (Lund & Agostinelli, 2018) for the von Mises distribution, and `BB` (Varadhan & Gilbert, 2009) for Barzilai-Borwein spectral projected gradient optimization. The algorithm is available on GitHub (www.github.com/maltelueken/gazeHMM).

Simulation Study

To assess how well the HMM recovers parameters and state sequences, we conducted a simulation study. The design and analysis of the study were preregistered on the Open Science Framework (https://doi.org/10.17605/OSF.IO/VDJGP). This section is majorly copied from the preregistration (with adapted tenses). The study was divided in four parts. Here, we only report the first two parts, the other two parts, which address starting values and missing data, can be found in the supplementary material (www.github.com/maltelueken/gazeHMM_validation).

The HMM repeatedly generated data with a set of parameters (true parameter values). The same model was applied to estimate the parameters from the generated data (estimated parameter values). We compared the true with the estimated parameter values to assess whether a parameter was recovered by the model. Additionally, we contrasted the true states of the HMM with the estimated states to judge how accurately the model recovered the states that generated the data.

Starting Values

The HMM always started with a uniform distribution for the initial state and state transition probabilities. Random starting values for the estimation of shape, scale, and concentration parameters were generated by gamma distributions with a shape parameter of $\alpha_{start} = 3$ and $\beta_{start;i} = \theta_i/2$, with θ_i being the true value of the parameter to be estimated in simulation $i \in (1, \ldots, I)$. This setup ensured that the starting values were positive, their distributions were moderately skewed, and the modes of their distributions equaled the true parameter values. Mean parameters of the von Mises distribution always started at their true values.

Design

In the first part, we varied the parameters of the HMM. For models with $k \in \{2, 3, 4\}$ states, $q \in \{10, 15, 20\}$ parameters were manipulated, respectively. For each parameter, the HMM generated 100 data sets with $N = 2500$ samples and the parameter varied in a specified interval in equidistant steps. This resulted in $100 \times (10 + 15 + 20) = 4500$ recoveries. Only one parameter alternated at once, the other parameters were set to their default values. All parameters of the HMM were estimated freely (i.e., there were no fixed parameters in the model). We did not manipulate the initial state probabilities because these are usually irrelevant in the context of eye movement classification. For the transition probabilities, we only simultaneously changed the probabilities for staying in the same state (diagonals of the transition matrix) to reduce the complexity of the simulation. The left over probability mass was split evenly between the probabilities for switching to a different state (per row of the transition matrix). Moreover, we did not modify the mean parameters of the von Mises distributions: As location parameters, they do not alter the shape of the distribution and they are necessary features for the HMM to distinguish between different states.

We defined approximate ranges for each response variable (see supplementary material) and chose true parameter intervals and default values so that they produced samples that roughly corresponded to these ranges. Tables 1 and 2 show the intervals and default values for each parameter in the simulation. Parameters were scaled down by factor 10 (compared to the reported ranges) to improve fitting of the gamma distributions. We set the intervals for shape parameters of the gamma distributions for all events to [1,5] to examine how skewness influenced the recovery (shape values above five approach a symmetric distribution). The scale parameters were set so that the respective distribution approximately matched the assumed ranges. Since the concentration parameters of the von Mises distribution are the inverse of standard deviations, they were varied on the inverse scale.

In the second part, we manipulated the sample size of the generated data and the amount of noise added to it. The model parameters were set to their default values. For models with $k \in \{2, 3, 4\}$ states and sample sizes of $N \in \{500, 2500, 10000\}$, we generated 100 data sets ($100 \times 3 \times 3 = 900$ recoveries). These samples sizes roughly corresponded to small, medium, and large eye-tracking data sets for a single participant and trial. To simulate noise, we replaced velocity and acceleration values y with draws from a gamma distribution with $\alpha_{noise} = 3$ and $\beta_{noise} = (y/2)\tau_{noise}$ with $\tau_{noise} \in [1, 5]$ varying between data sets. This procedure ensured that velocity and acceleration values remained positive and were taken from moderately skewed distributions with modes equal to the original values. To angle, we added white noise from a von Mises distribution with $\mu_{noise} = 0$ and $\kappa_{noise} \in 1/[0.1, 10]$ varying between data sets. τ_{noise} and κ_{noise} were increased simultaneously in equidistant steps in their intervals.

Data Analysis

For each parameter, we calculated the root median square proportion deviation (RMdSPD; analogous to root median square percentage errors, see Hyndman & Koehler, 2006) between the true and estimated parameter values:

$$\text{RMdSPD} = \sqrt{\text{Median}(\epsilon_1^2, \ldots, \epsilon_I^2)}$$

$$\epsilon_i^2 = \left(\frac{\hat{\theta}_i - \theta_i}{\theta_i}\right)^2,$$

where θ_i is the true parameter value and $\hat{\theta}_i$ is the estimated parameter value for simulation $i \in (1, \ldots, I)$, respectively. Even though it was not explicitly mentioned in the preregistration, this measure is only appropriate when $\theta_i \neq 0$. This was not the case for some mean parameters of the von Mises distributions. In those cases, we used $\theta_i = 2\pi$ instead. We treated RMdSPD < 0.1 as good, $0.1 \leq$ RMdSPD < 0.5 as moderate, and RMdSPD ≥ 0.5 as bad recovery of a parameter. By taking the median, we reduced the

Table 1

Intervals and Default Parameter Values for the Transition Model in the Simulation

	Initial state prob.	Trans. prob. same state	Trans. prob. other state
Interval	-	[.01,.99]	1-a(i=j)/(k-1)
Default	1/k	0.9	0.1/(k-1)

Note. The transition probability for staying in the same state is denoted by $a_{i=j}$ and the probability for switching to a different state by $a_{i \neq j}$. The number of states in the model is denoted by *k*.

influence of potential outliers in the estimation and using proportions enabled us to compare RMdSPD values across parameters and data sets.

Additionally, we applied a bivariate linear regression with the estimated parameter values as the dependent and the true parameter values as the independent variable to each parameter that has been varied on an interval in part one. Regression slopes closer to one indicated that the model better captured parameter change. Regression intercepts different from zero reflected a bias in parameter estimation.

To assess state recovery, we computed Cohen's kappa (for all events taken together, not for each event separately) as a measure of agreement between true and estimated states for each generated data set. Cohen's kappa estimates the agreement between two classifiers accounting for the agreement due to chance. Higher kappa values were interpreted as better model accuracy. We adopted the ranges proposed by Landis and Koch (1977) to interpret kappa values. Models that could not be fitted were excluded from the recovery.

Results

Parameter Variation.   In the first part of the simulation, we examined how varying the parameters in the HMM affected the deviation of estimated parameters and

Table 2

Intervals and Default Parameter Values for the Response Model in the Simulation

	Velocity		Acceleration		Rel. angle	
	Shape	Scale	Shape	Scale	Mean	Concentration
State 1						
Interval	[1,5]	[0.1,0.6]	[1,5]	[0.05,0.25]	-	-
Default	3	0.35	3	0.25	-	-
State 2						
Interval	[1,5]	[5,15]	[1,5]	[1,5]	-	1/[0.1,10]
Default	3	10	3	3	0	1
State 3						
Interval	[1,5]	[0.5,1.5]	[1,5]	[1,5]	-	1/[0.1,10]
Default	3	1	3	3	pi	1
State 4						
Interval	[1,5]	[0.5,1.5]	[1,5]	[0.05,0.25]	-	1/[0.1,10]
Default	3	1	3	0.15	0	1

Note. Shape parameters are denoted by α, scale parameters by β, mean parameters by μ, and concentration parameters by κ. The default values for the uniform distribution in state one were min $= 0$ and max $= 2\pi$.

accuracy of estimated state sequences. Figure 1 displays the RMdSPD between true and estimated parameters depending on which parameter has been manipulated in the HMM. The RMdSPD was below 0.1 for all estimated and manipulated parameters, indicating good recovery.[1] The regression lines between manipulated true and estimated parameters are shown in Figures 2 and 3. With one outlier at parameter $\alpha_{acc;1}$, the estimated parameters matched the true parameters very closely. Thus, parameter change was captured well and the estimation almost unbiased. Considering accuracy, Figure 4 displays Cohen's kappa between true and estimated hidden state sequences. With two exceptions, kappa values were almost one, suggesting nearly perfect agreement.

For the HMM with three states, the RMdSPD is shown in Figure 5. When response parameters (other than $a_{i=j}$) were manipulated, the RMdSPDs for a_{12} and a_{31} were consistently between 0.1 and 0.5. Varying κ in states two and three led to RMdSPDs between 0.1 and 0.5 in the respective states, which we interpreted as moderate recovery. Otherwise, RMdSPDs were consistently lower than 0.1, indicating good recovery. Inspecting the regression lines between true and estimated parameters (see Figures 6 and 7) revealed strong and unbiased linear relationships (intercepts close to zero and slopes close to one). In contrast to the two-state HMM, larger deviations and more outliers were observed. Cohen's kappa values are presented in Figure 8. For most estimated models, the kappa values between true and estimated state sequences were above 0.95, meaning almost perfect agreement. However, for some models, we observed kappas clustered around zero or -0.33, which suggests that state labels were switched (see supplementary material).

The RMdSPDs for the four-state HMM is shown in Figure 9. For estimated transition probabilities and α_{vel} and β_{vel} parameters in states one and four, RMdSPDs were between 0.1 and 0.5, suggesting moderate recovery. Also, estimated kappa parameters

---

[1] Note that the initial state probability ρ_i has RMdSPD = 1. Since the HMM only simulated one state sequence, this parameter is always either zero or one (leading to RMdSPD = 1). Therefore, we decided not to include it in the analysis.
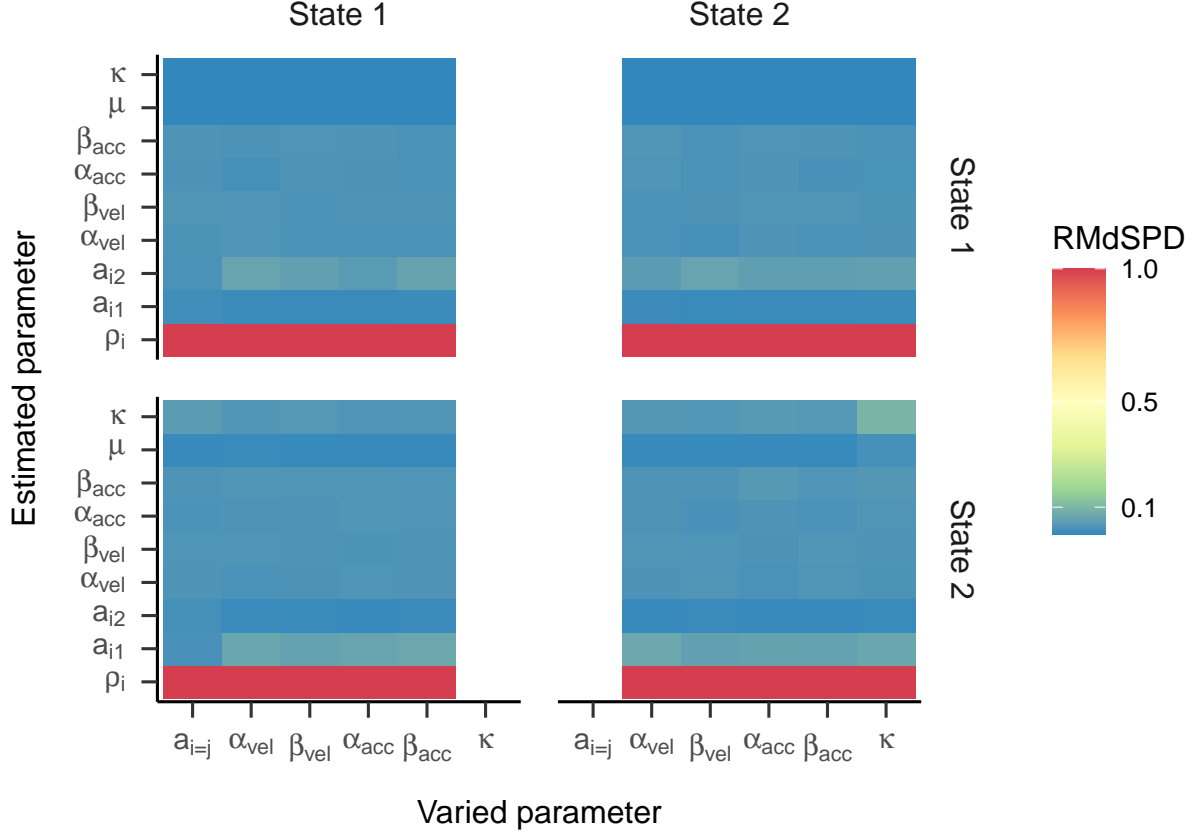
Figure 1. RMdSPD between true and estimated parameters of the two-state HMM in part one of the simulation. Labels on the x-axis indicate which true parameters have been manipulated and labels on the y-axis show for which estimated parameter the RMdSPD is displayed. Top facet labels specify in which state the parameters have been varied and right facet labels denote to which state estimated parameters belong. ρ_i is the initial probability for state i (indicated by the right facet label), a_{ij} is the probability to transition from state i to state j, α and β are the shape and scale parameters of the gamma distributions, and μ and κ are the mean and concentration parameter of the von Mises distribution.
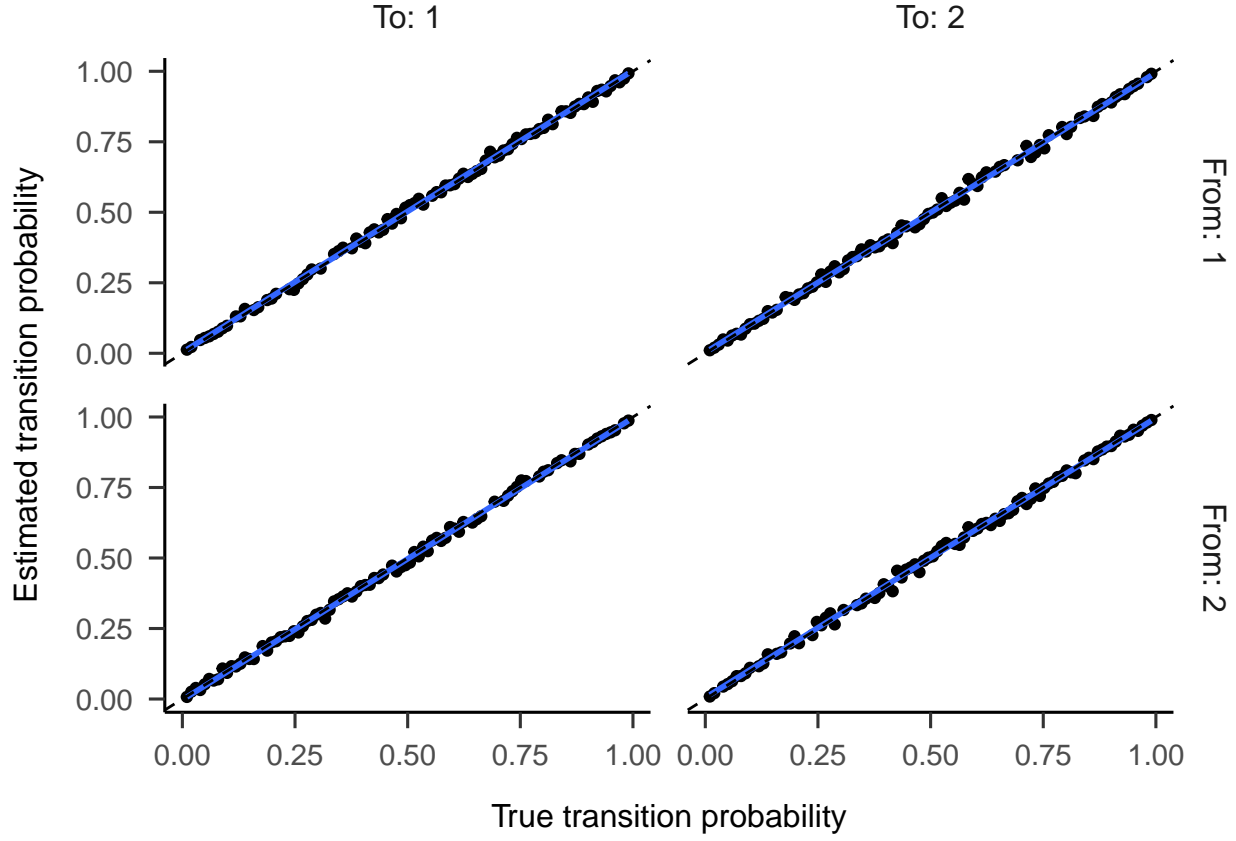
Figure 2. Regression lines between true and estimated transition probabilities for the two-state HMM in part one. Right facet labels show *from* and top facet labels show *to* which state the HMM is moving. Dashed lines refer to perfect recovery.

in state four were often moderately recovered when parameters in states two, three, and four were varied. Otherwise, RMdSPDs were below 0.1, indicating good recovery. Looking at Figures 10 and 11, the regression lines between true and estimated parameters exhibit strong and unbiased relationships. However, there were larger deviations and more outliers than in the previous models, especially for states one and four. Cohen's kappa ranged mostly between 0.6 and 0.9, meaning moderate to almost perfect agreement between true and estimated state sequences (see Figure 12). Here, some kappa values clustered around 0.25 and zero, which, again, suggests label switching.

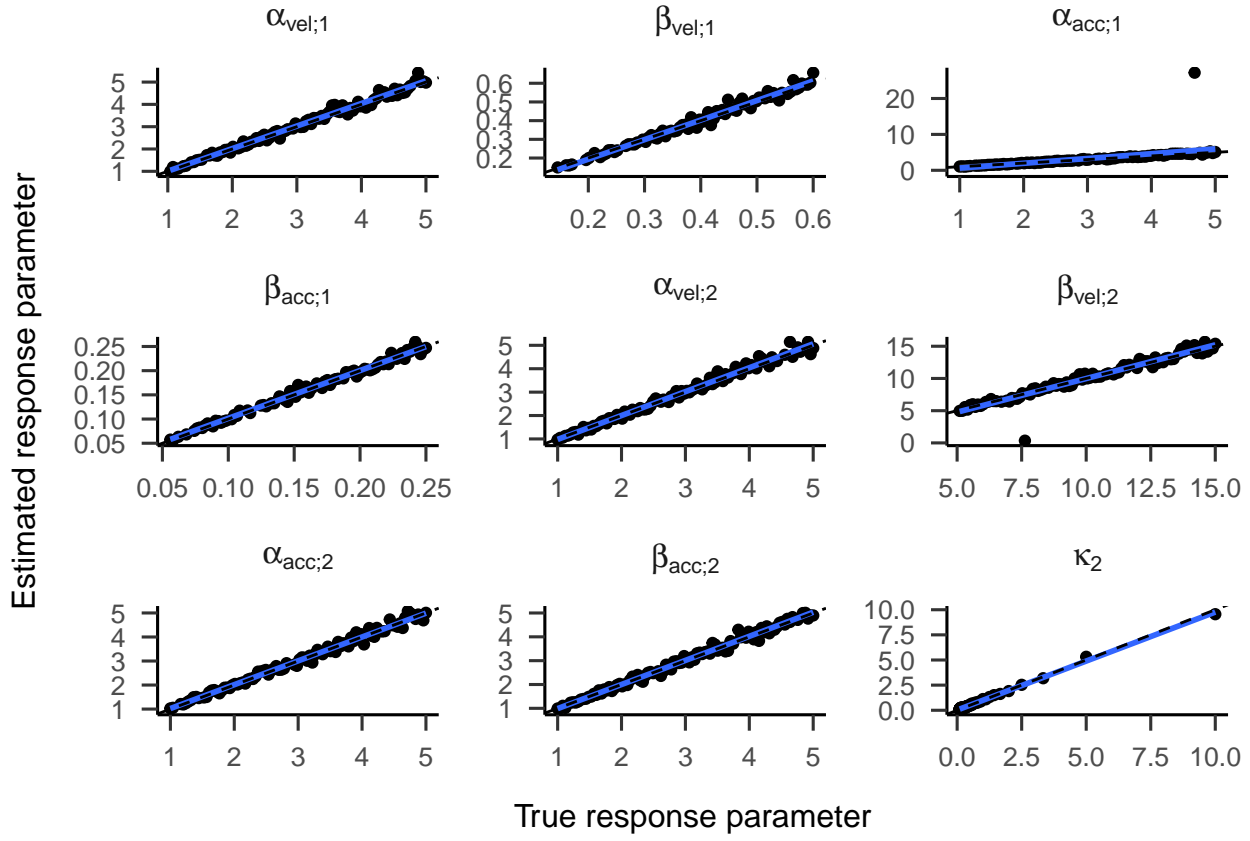Overall, simulations in part one demonstrated that the HMM recovered parameters

Figure 3. Regression lines between true and estimated response parameters of the two-state HMM in part one. Top facet labels indicate response parameters. Dashed lines refer to perfect recovery.

very well when they were manipulated. Deviations from true parameters were mostly small. In the four-state model, estimated transition probabilities for state one and four deviated moderately. Moreover, the HMM estimated state sequences very accurately with decreasing accuracy for the four-state model.

Sample Size and Noise Variation. In the second part, we 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 13), suggesting bad recovery. The other estimated parameters showed RMdSPDs close to or below 0.1, which means they were recovered well. Increasing the sample size seemed to improve RMdSPDs
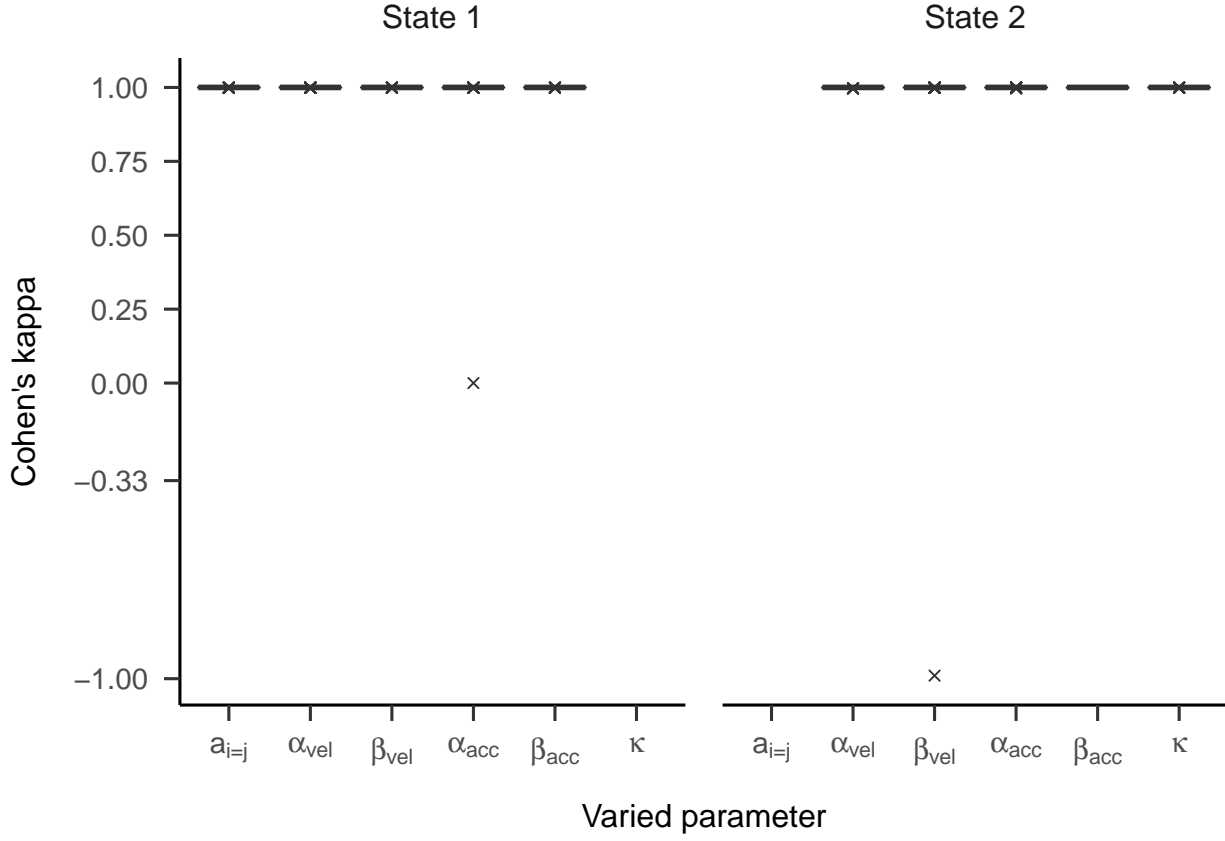
Figure 4. Boxplots displaying Cohen's kappa depending on which parameter of the two-state HMM has been manipulated in part one. Top facet labels indicate for which state parameters have been manipulated. Black solid lines symbolize medians. Crosses represent outliers (distance to first/third quartile higher than 1.5 times the inter-quartile range [IQR]).

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 14, left plot).

Regarding three states, the RMdSPDs for the β_{vel} and β_{acc} were above 0.5 in all three states (see Figure 15), indicating bad recovery. 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 across all three states, models with higher sample sizes had lower RMdSPDs. The state recovery of the estimated models was almost perfect with most kappa values
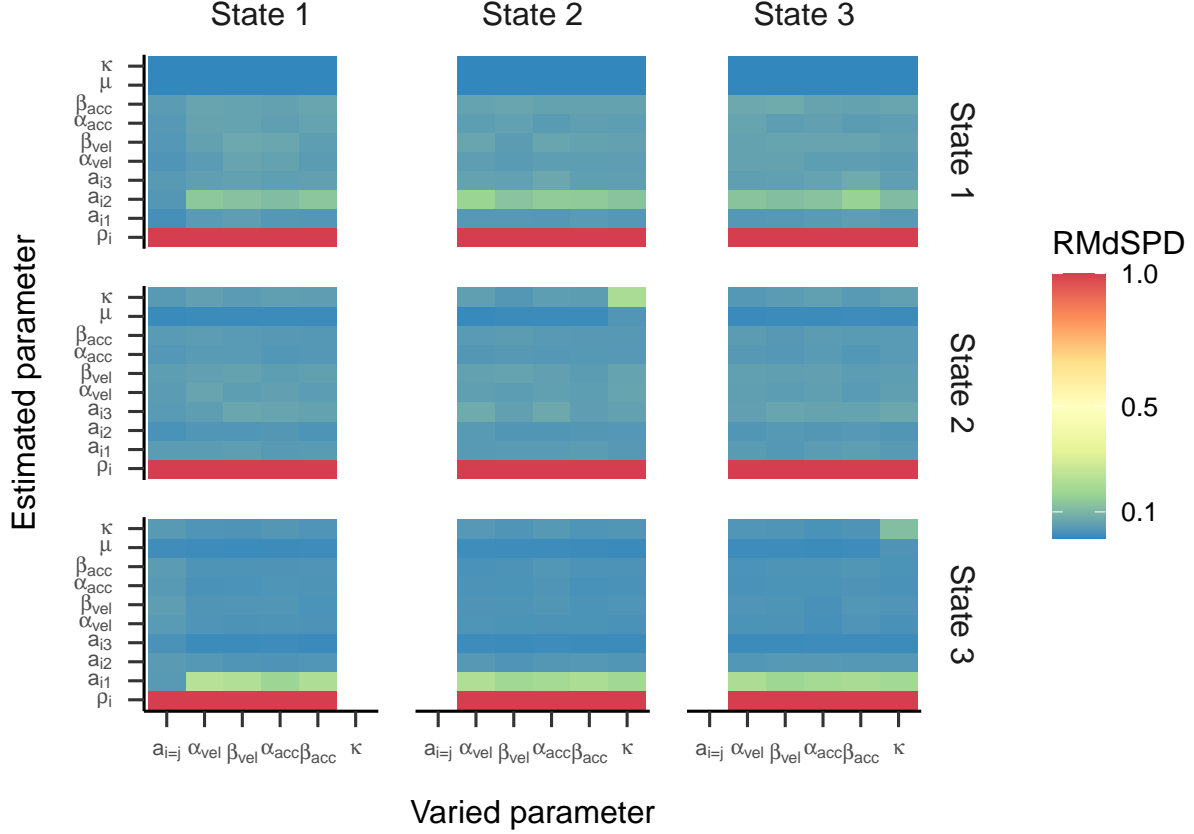
Figure 5. RMdSPD between true and estimated parameters of the three-state HMM in part one of the simulation. Labels on the x-axis indicate which true parameters have been manipulated and labels on the y-axis show for which estimated parameter the RMdSPD is displayed. Top facet labels specify in which state the parameters have been varied and right facet labels denote to which state estimated parameters belong. ρ_i is the initial probability for state i (indicated by the right facet label), a_{ij} is the probability to transition from state i to state j, α and β are the shape and scale parameters of the gamma distributions, and μ and κ are the mean and concentration parameter of the von Mises distribution.
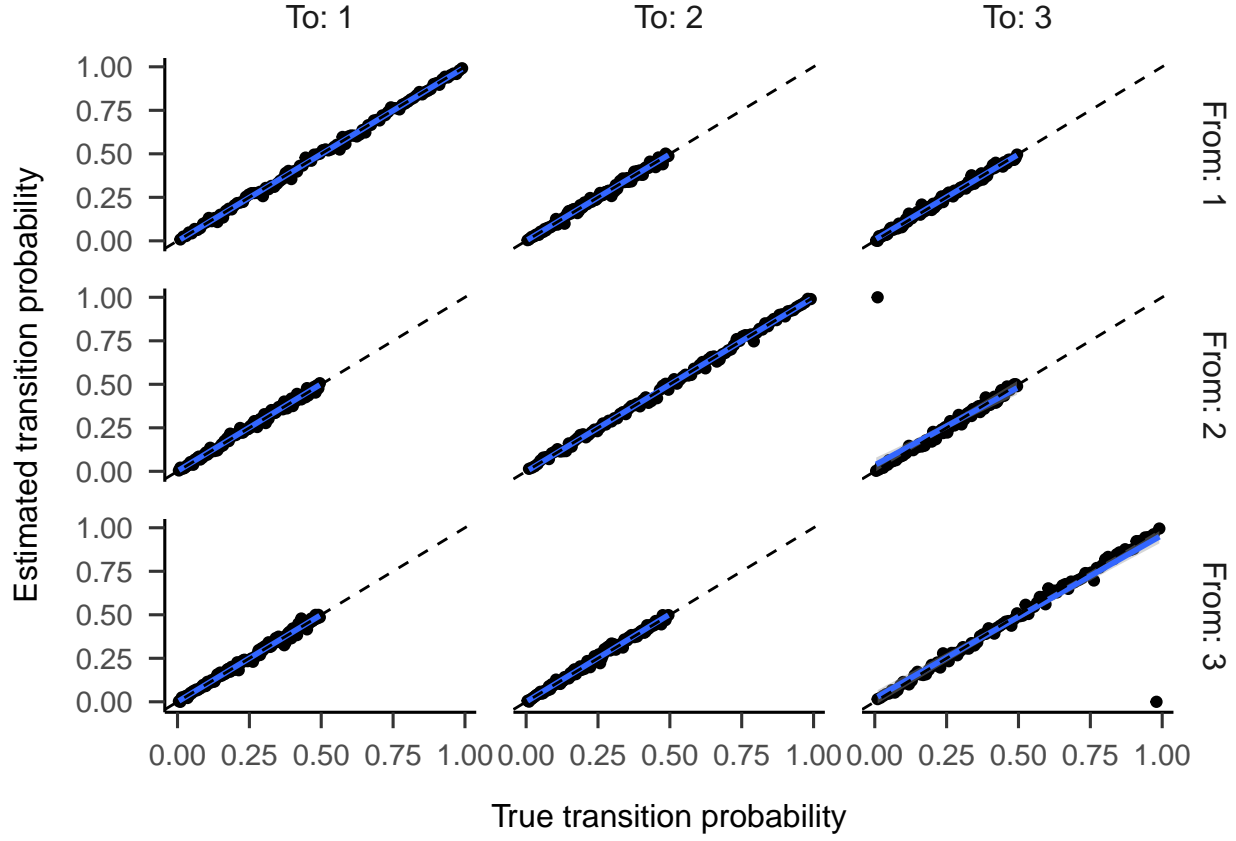
Figure 6. Regression lines between true and estimated transition probabilities for the three-state HMM in part one. Right facet labels show *from* and top facet labels show *to* which state the HMM is moving. Dashed lines refer to perfect recovery.

above 0.95 (see Figure 14, middle plot). Several outliers clustered around kappas of zero and -0.33, signaling label switching.

RMdSPDs regarding the four-state HMM are displayed in Figure 16. For states one and four, values for most parameters (including all transition probabilities) were above 0.5, suggesting bad recovery. Similarly, β_{vel} and β_{acc} in states two and three showed bad recovery. For states two and three, higher sample 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 14, right plot). Multiple kappa values clustered around 0.25 or zero, which can be explained by
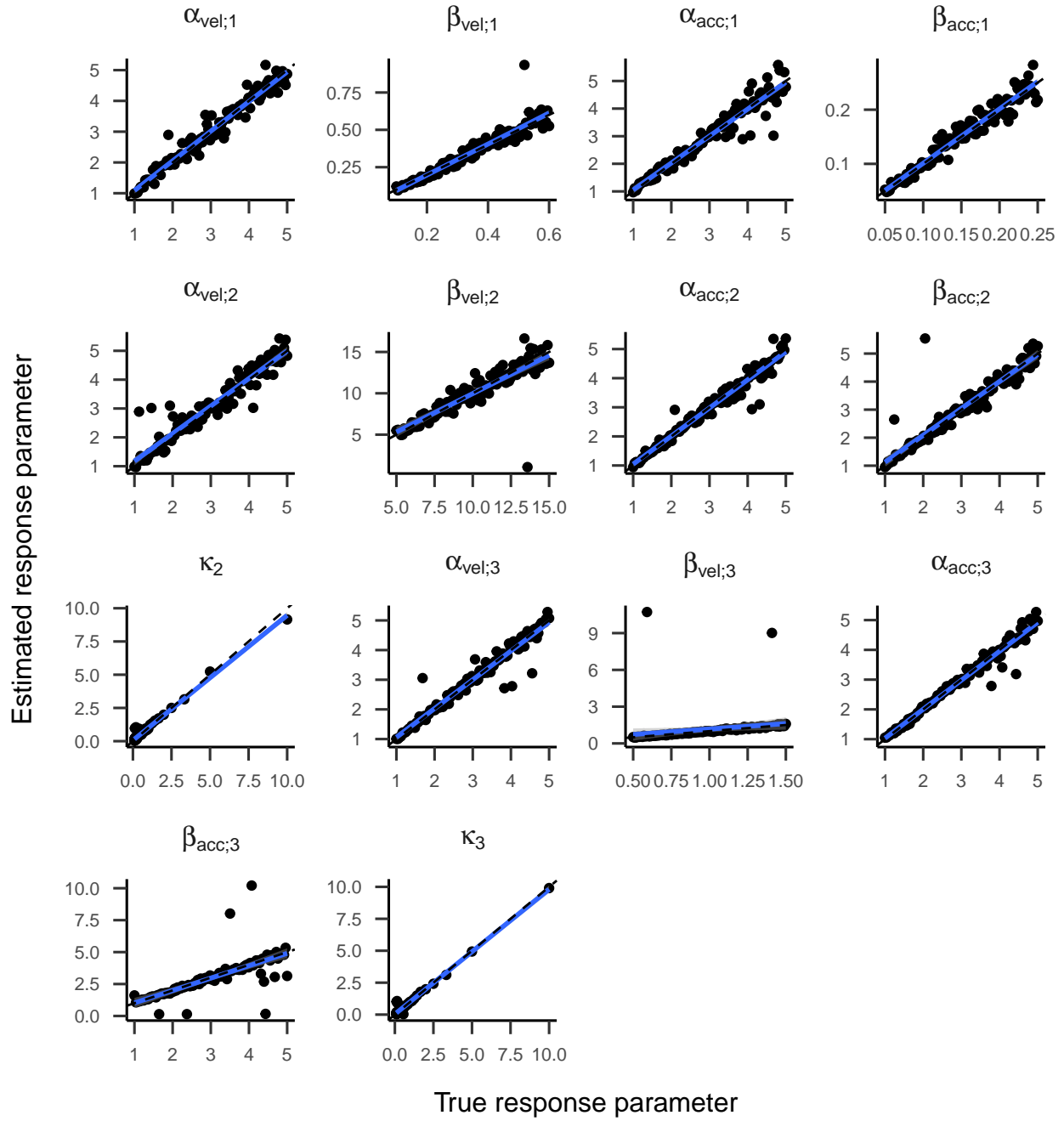
Figure 7. Regression lines between true and estimated response parameters of the three-state HMM in part one. Top facet labels indicate response parameters. Dashed lines refer to perfect recovery.
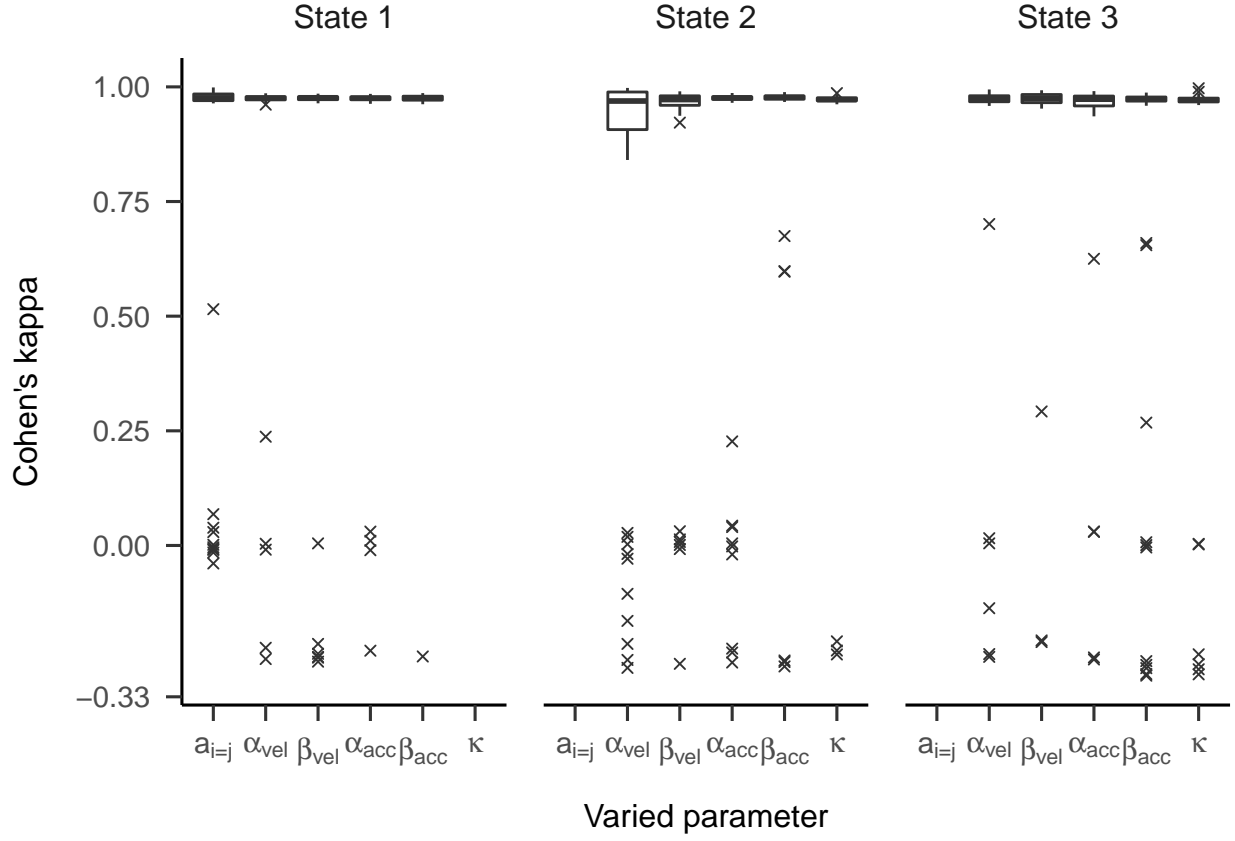
Figure 8. Boxplots displaying Cohen's kappa depending on which parameter of the three-state HMM has been manipulated in part one. Top facet labels indicate for which state parameters have been manipulated. Black solid lines symbolize medians and hinges the first and third quartile. Whiskers range from hinges to lowest/highest value within 1.5 times the IQR. Crosses represent outliers.

label switching.

In general, the HMM recovered parameters well despite noise being added to the data. However, in the four-state model, the parameter recovery for states one and four substantially decreased. In the three- and four-state models, scale parameters of gamma distributions were badly recovered. Increasing the sample size in the HMM slightly improved the recovery of most parameters. The state recovery of the model was slightly lowered when more states were included, but it was neither affected by the noise variability
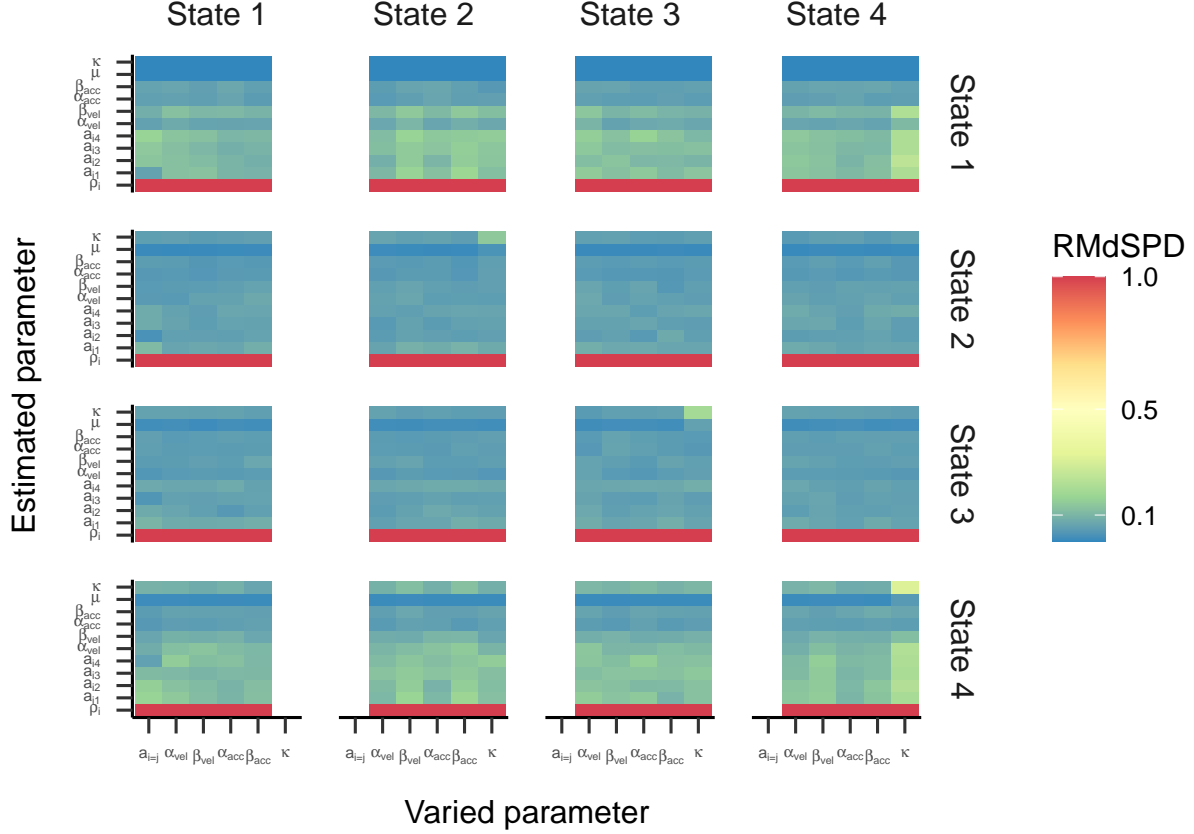
Figure 9. RMdSPD between true and estimated parameters of the four-state HMM in part one of the simulation. Labels on the x-axis indicate which true parameters have been manipulated and labels on the y-axis show for which estimated parameter the RMdSPD is displayed. Top facet labels specify in which state the parameters have been varied and right facet labels denote to which state estimated parameters belong. ρ_i is the initial probability for state i (indicated by the right facet label), a_{ij} is the probability to transition from state i to state j, α and β are the shape and scale parameters of the gamma distributions, and μ and κ are the mean and concentration parameter of the von Mises distribution.

Figure 10. Regression lines between true and estimated transition probabilities for the four-state HMM in part one. Right facet labels show *from* and top facet labels show *to* which state the HMM is moving. Dashed lines refer to perfect recovery.
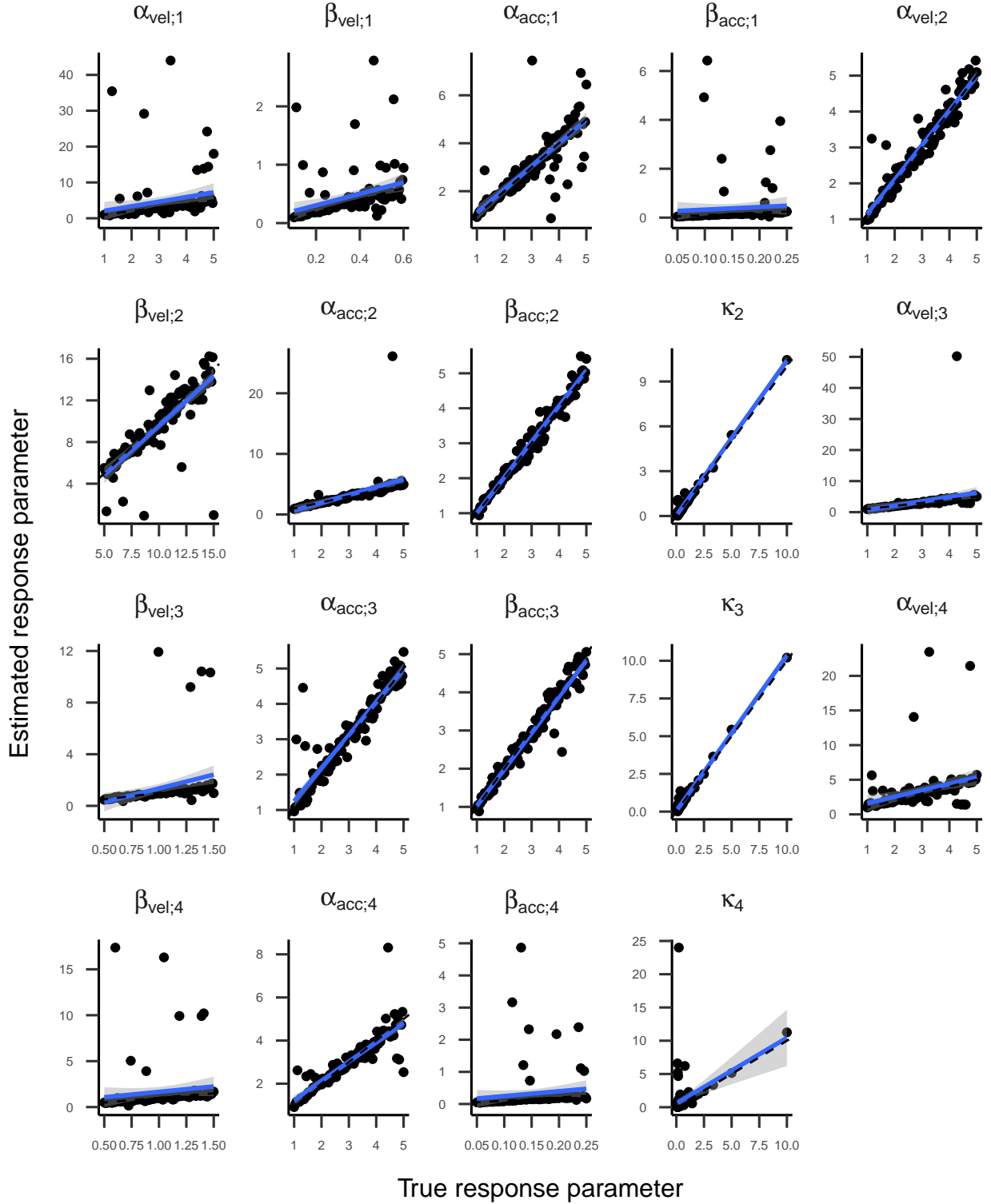
Figure 11. Regression lines between true and estimated response parameters of the four-state HMM in part one. Top facet labels indicate response parameters. Dashed lines refer to perfect recovery.
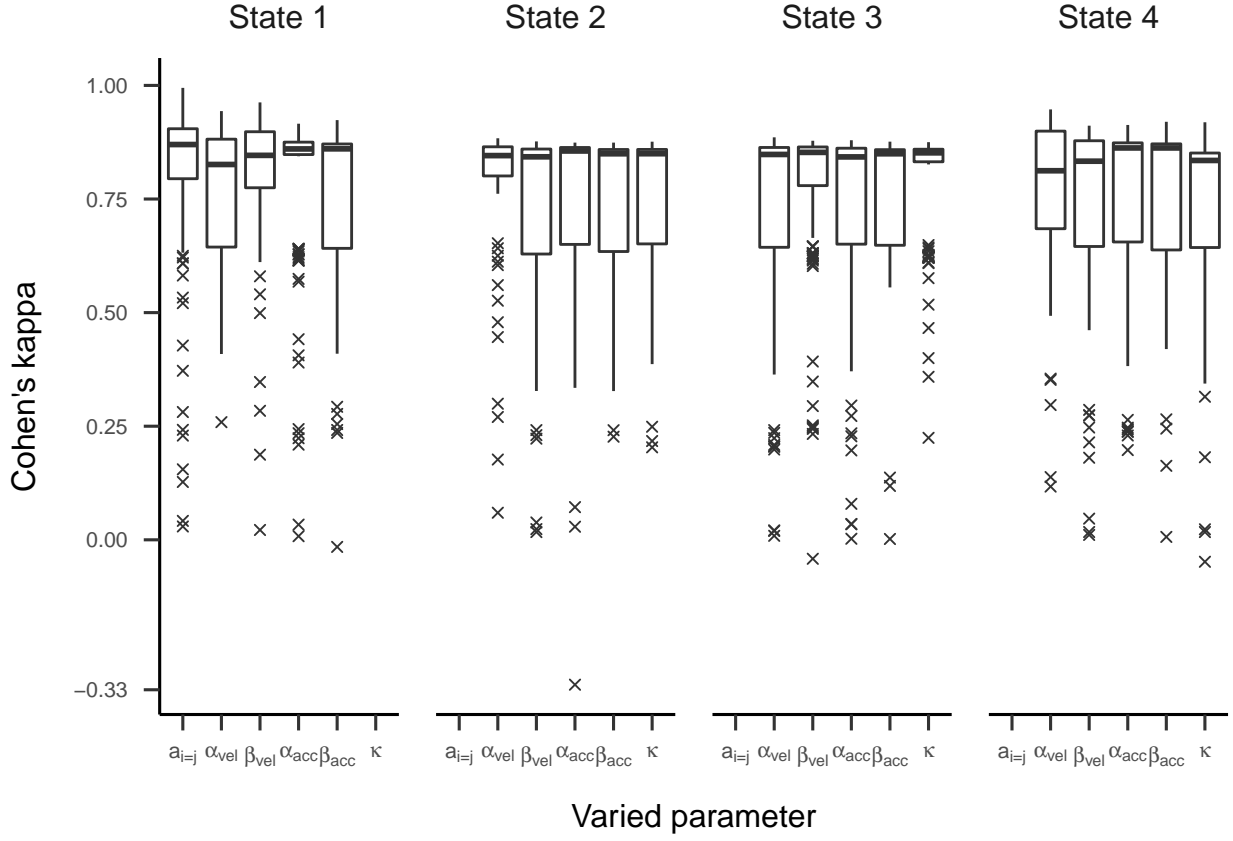
Figure 12. Boxplots displaying Cohen's kappa depending on which parameter of the four-state HMM has been manipulated in part one. Top facet labels indicate for which state parameters have been manipulated. Black solid lines symbolize medians and hinges the first and third quartile. Whiskers range from hinges to lowest/highest value within 1.5 times the IQR. Crosses represent outliers.

τ_{noise} nor the sample size.

Validation Study

To validate gazeHMM, we applied the algortihm on two benchmark data sets. As starting values, we used $\rho = 1/k$ for the initial state model as well as $a_{i=j} = 0.9$ and $a_{i \neq j} = 0.1/k$ for the transition model. The values for the response model are displayed in Table 3. For a fifth eye movement event, we chose starting values that would enable the
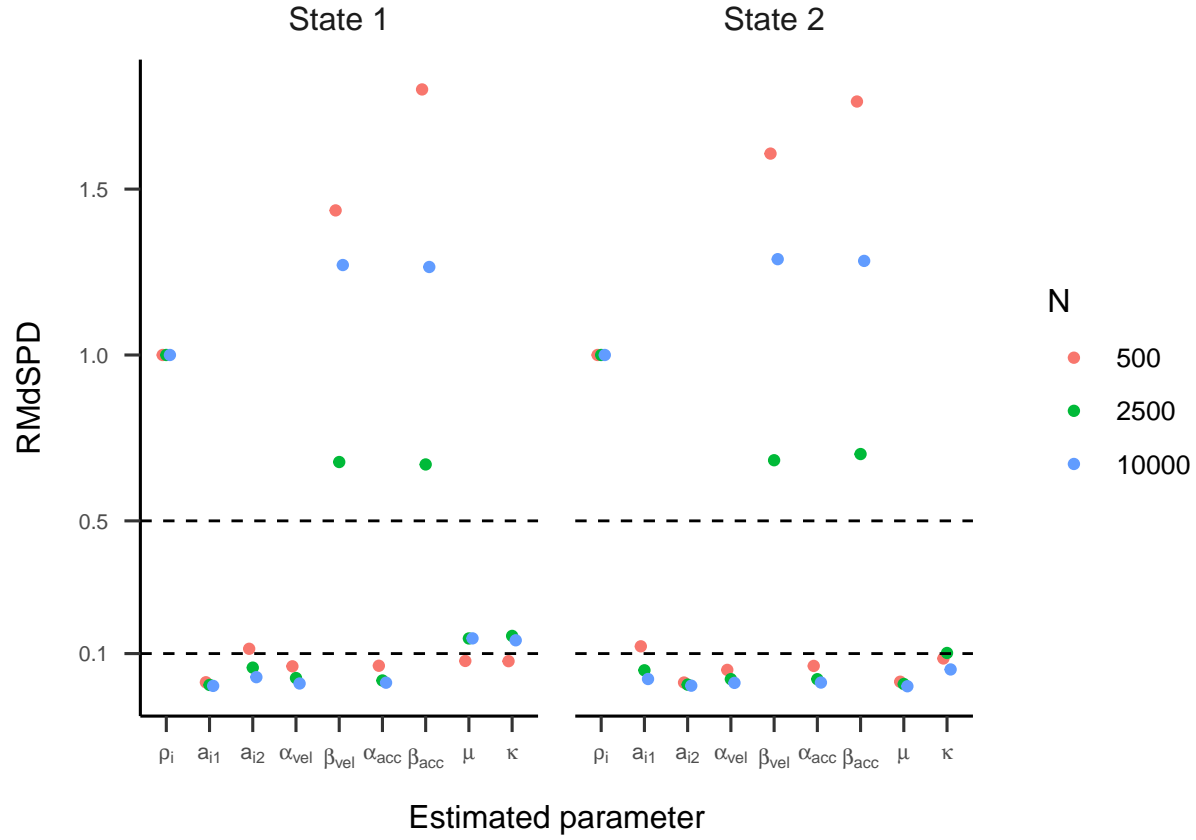
Figure 13. RMdSPD between true and estimated parameters of the two-state HMM in part two of the simulation. Colours indicate different sizes of generated data. Labels on the x-axis indicate for which estimated parameter the RMdSPD is displayed. Top facet labels denote to which state estimated parameters belong. ρ_i is the initial probability for state i (indicated by the top facet label), a_{ij} is the probability to transition from state i to state j, α and β are the shape and scale parameters of the gamma distributions, and μ and κ are the mean and concentration parameter of the von Mises distribution.
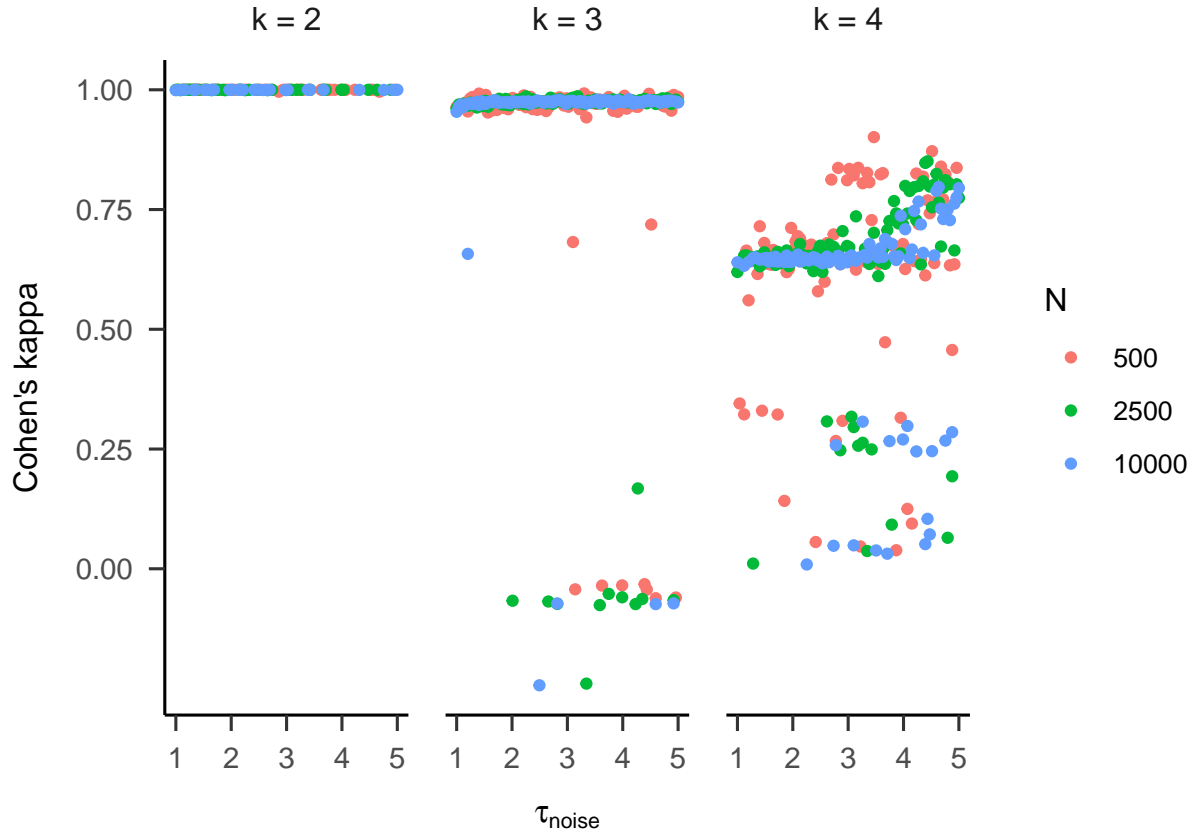
Figure 14. Cohen's kappa depending on the variation of noise added to the data generated by the HMM. Colours indicate different sizes of generated data. Top facet labels indicate the number of states in the HMM.

HMM to split any other event into two subevents (e.g., fixations into drift and microsaccades). In contrast to the simulation study, generating random starting values often led to bad model fits and label switching between states. To improve the fitting of the gamma distributions, velocity and acceleration signals were scaled down by factor 100^2 (so were the starting values for their gamma distributions).

---

[2] Scaling down by factor 100 differs from the simulation study (scaling down by 10). The algorithm allows the user to manually specify this factor, and in this case, factor 100 led to better model fits than factor 10.
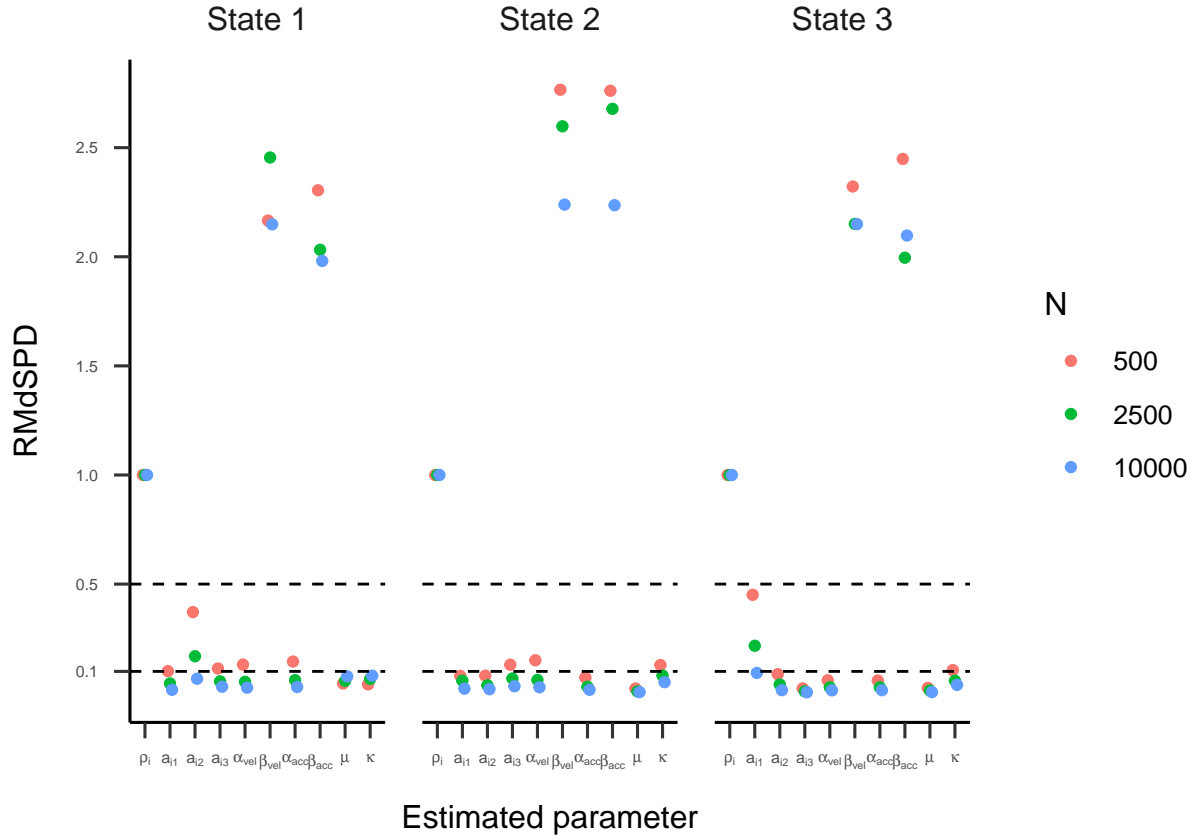
Figure 15. RMdSPD between true and estimated parameters of the three-state HMM in part two of the simulation. Labels on the x-axis indicate for which estimated parameter the RMdSPD is displayed. Top facet labels denote to which state estimated parameters belong. ρ_i is the initial probability for state i (indicated by the top facet label), a_{ij} is the probability to transition from state i to state j, α and β are the shape and scale parameters of the gamma distributions, and μ and κ are the mean and concentration parameter of the von Mises distribution.
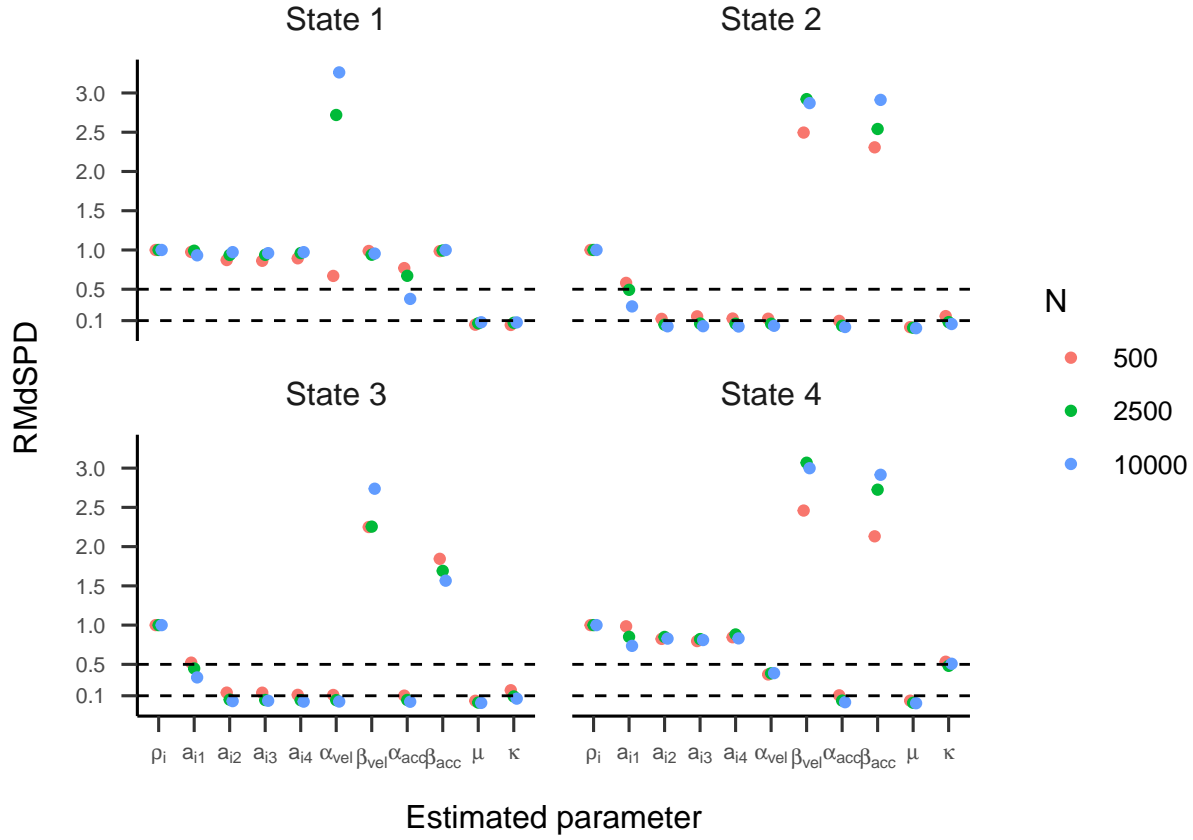
Figure 16. RMdSPD between true and estimated parameters of the three-state HMM in part two of the simulation. Labels on the x-axis indicate for which estimated parameter the RMdSPD is displayed. Top facet labels denote to which state estimated parameters belong. ρ_i is the initial probability for state i (indicated by the top facet label), a_{ij} is the probability to transition from state i to state j, α and β are the shape and scale parameters of the gamma distributions, and μ and κ are the mean and concentration parameter of the von Mises distribution.

Table 3

Starting Values for the Response Model for Fitting gazeHMM to
Benchmark Data

	Velocity		Acceleration		Rel. angle	
	Shape	Scale	Shape	Scale	Mean	Concentration
Fixation	10	10	10	10	-	-
Saccade	50	50	50	50	0	10
PSO	50	50	50	50	3.14	10
Pursuit	20	20	20	20	0	10
Event 5	20	20	50	50	0	10

Note. Starting values for velocity and acceleration signals are
shown before scaling down by factor 100.

Data Sets

We chose two data sets for validation: One was published in a study by Andersson et
al. (2017) and has been widely used for validation purposes (e.g., Pekkanen & Lappi,
2017). It contains eye-tracking data from three conditions: A static condition, where
subjects had to look freely at images, and two dynamic conditions, where they had to
follow a constantly moving dot or objects in a video. The data was sampled with 500 Hz
and two human coders (MN and RA) labeled them as belonging to six different eye
movement events: Fixation, saccade, PSO, smooth pursuit, blink, or other. Andersson et
al. (2017) used the data to compare 10 different classification algorithms. We adopted their
results to compare these 10 algorithms and the two human coders with gazeHMM. We used
the original data from the study but removed two recordings from the moving dots
condition because they majorly contained samples labeled as "other" or blinks. Moreover,

the recordings could not be matched to the results obtained by Andersson et al. (2017).

The second data set was published in Ehinger, Groß, Ibs, and Peter (2019) and has to our knowledge not yet been used for validation. Here, we only took tasks four and five out of 10 tasks because these are qualitatively different to the first data set. In task four, subjects were instructed to fixate a central target for 20 s. Task 5 was set up similarly, but subjects had to blink when they heard one out of seven beeps (with a beep duration of 100 ms and 1.5 s intervals in between). Eye movements were recorded with 250-500 Hz. For simplicity, we only used data obtained by the EyeLink (SR Research Ltd., Ontario, Canada) eye-tracker.

Data Analysis

A successful validation of gazeHMM was determined by two approaches: First, we applied gazeHMM with generative models containing 1-5 states to both data sets. The fits of the generative models were compared using Schwarz weights (Wagenmakers & Farrell, 2004), a conversion of the BIC (Schwarz, 1978) into model weights. They can be interpreted as the probability of a model having generated the data compared to the competing models. For the static condition in the Andersson et al. (2017) data set, we expected the generative model with three states (fixation, saccade, and PSO), and for the dynamic conditions the model with four states (incl. smooth pursuit) to display the highest Schwarz weight. Regarding the Ehinger et al. (2019) data set, we assumed that the one-state model (only fixation) would show the highest weights for both tasks.

The algorithm was applied separately to every subject, condition, or task. For the Andersson et al. (2017) data set, all generative models were successfully fitted, whereas for the Ehinger et al. (2019) data set, it was only 780 out of 900 models (87%, 60 models per task).

Second, we compared gazeHMM to other algorithms and human coders. We applied our algorithm with a three-state generative model to the static condition in the Andersson

et al. (2017) data set, and with a four-state model to the dynamic conditions. For

comparison criteria, we followed Andersson et al. (2017): We calculated the RMSD of

event durations and counts between all algorithms and the average of the two human

coders. Our results differ slightly from the original study because we excluded two

recordings (leading to less events) and calculated the event durations as

$Dur(e) = max(\mathbf{t}_e) - max(\mathbf{t}_{e-1})$, where $\mathbf{t_e}$ is the vector of sample time stamps for the event

e. Cohen's kappa was calculated for each event as the binary agreement between the

algorithms and the average of the human coders. Lastly, the overall disagreement indicated

which samples were classified differently by the algorithms compared to the average of the

human coders across all events. The human coders were compared directly to each other.

Results

Model Comparison.   Examining the Schwarz weights displayed in Figure 17, we

observed that the five-state generative model showed the highest weights in all three

conditions. Only in the moving dots condition, two subjects displayed the highest weights

for the four-, and one subject for the three-state model. In sum, we concluded that the

five-state generative model has most likely generated the Andersson et al. (2017) data,

opposing our expectations. Because the Ehinger et al. (2019) data set showed a similar

pattern, we included the results for this data in the supplementary material.

A recent model recovery study showed that the BIC tended to prefer overly complex

HMMs when they were misspecified (e.g., the conditional independence assumption was

violated; Pohle, Langrock, Beest, & Schmidt, 2017). Instead, the integrated completed

likelihood (ICL) criterion (Biernacki, Celeux, & Govaert, 2000) performed better at

choosing the correct data generating model. Therefore, we post hoc computed the weighted

ICL criterion (analogous to Schwarz weights) for the models fitted to the Andersson et al.

(2017) data set. Using the ICL as the model selection criterion yielded very similar results

to the BIC (see supplementary material). The preference for the five-state generative

model was even more consistent across conditions and subjects.
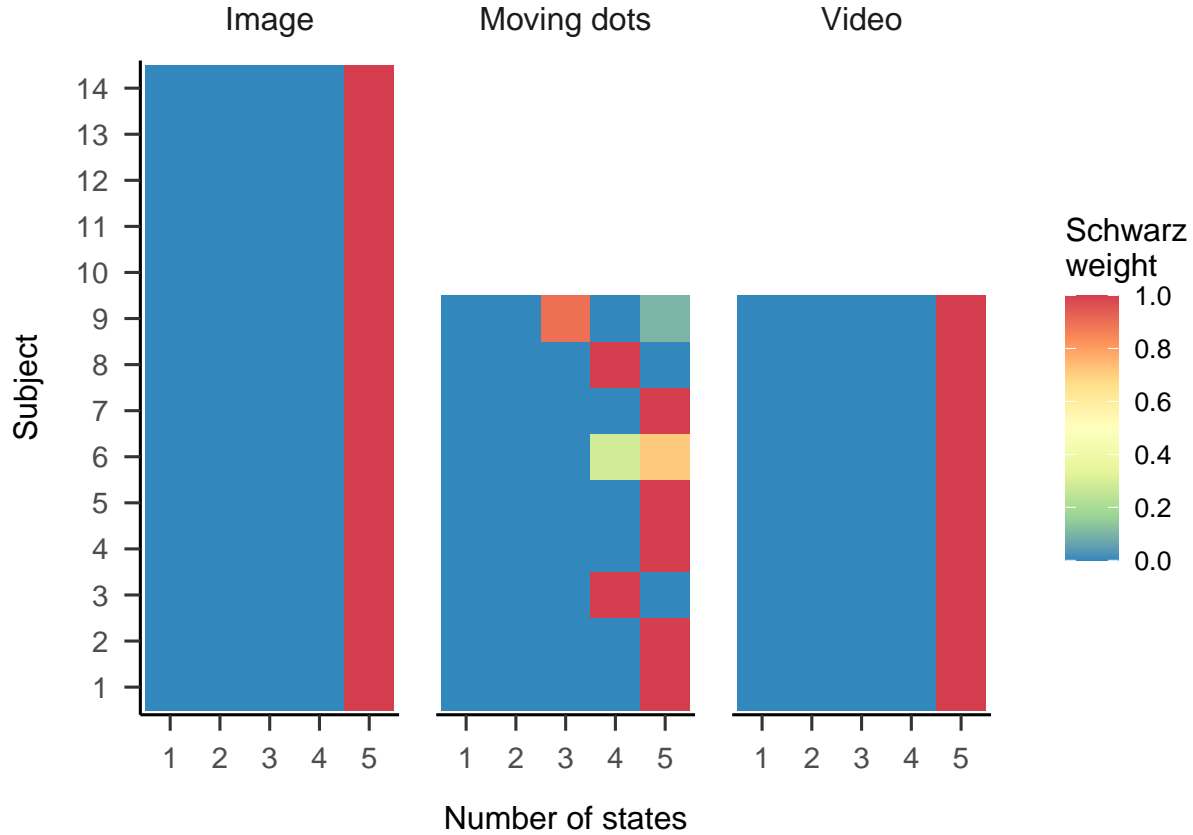
Figure 17. Schwarz weights displayed for each subject and generative models with different numbers of states. Top facet labels indicate the condition in the Andersson et al. (2017) data set. Higher weights indicate a better model fit.

Comparison to Other Algorithms. As displayed in Table 4, gazeHMM showed a relatively low RMSD for fixations in the static condition compared to the other algorithms that were applied to the Andersson et al. (2017) data set. The lower RMSD for fixations indicated more similar classification to the human coders in terms of their mean and SD duration as well as the number of classified fixations. Oppositely, for fixations in the dynamic conditions, the RMSD of gazeHMM was one of the highest among the compared algorithms, suggesting substantial differences to the human coders. It can be seen that gazeHMM classified a much larger number of fixations with very short durations. For

Table 4

Fixation Duration Descriptives and RMSD Between Algorithms and Human Coders

	Image				Moving dots				Video			
Algorithm	Mean	SD	Events	RMSD	Mean	SD	Events	RMSD	Mean	SD	Events	RMSD
coderMN	275	285	403	0.02	186	93	11	0.02	338	303	82	0.16
coderRA	271	287	391	0.02	174	94	12	0.02	255	185	81	0.16
gazeHMM-3	165	256	578	0.71	-	-	-	-	-	-	-	-
gazeHMM-4	-	-	-	-	16	17	381	1.12	22	25	1,243	1.18
CDT	465	643	276	1.44	60	115	177	0.44	244	354	226	0.18
IDTk	488	579	263	1.39	968	1,171	17	1.87	984	1,596	62	1.82
IKF	190	258	518	0.68	270	203	51	0.2	286	315	173	0.18
IMST	360	471	351	0.9	912	1,159	18	1.79	695	1,059	87	1.16
IHMM	148	236	717	0.72	316	298	47	0.34	259	348	207	0.18
IVT	127	223	843	0.73	292	298	51	0.31	225	334	240	0.18
NH	282	318	297	0.58	392	336	31	0.5	462	381	89	0.48
BIT	230	162	439	0.79	194	108	66	0.13	264	225	183	0.22

Note. Durations are displayed in milliseconds. gazeHMM-3 classified three and gazeHMM-4 classified four events. RMSD = root mean square deviation. Table design adapted from Andersson et al. (2017).

saccades, gazeHMM had a relatively high RMSD for the static condition, but it had the lowest RMSD for the moving dots condition, and a moderate value for the video condition (see Table 5). The deviation was mostly because gazeHMM classified a higher number of saccades than the human coders. Only two other algorithms classified PSOs (NH and LNS; Nyström & Holmqvist, 2010; Larsson et al., 2013). Here, gazeHMM showed a consistently higher RMSD than LNS and lower RMSD than NH (see Table 6). Our algorithm classified shorter and more PSOs than the human coders. No other algorithm parsed smooth pursuits, but the RMSD for gazeHMM was higher than among the human coders (see Table 7). Again, it classified a much larger number of smooth pursuits with short durations.

Table 8 contains the sample-to-sample agreement between the algorithms and human coders measured by Cohen's kappa. For fixations, gazeHMM showed one of the highest agreements for static, and *the* highest agreements for dynamic data. The absolute

Table 5

Saccade Duration Descriptives and RMSD Between Algorithms and Human

Coders

Algorithm	Image				Moving dots				Video			
	Mean	SD	Events	RMSD	Mean	SD	Events	RMSD	Mean	SD	Events	RMSD
coderMN	32	16	377	0.12	23	11	38	0.06	27	12	117	0.1
coderRA	34	14	374	0.12	22	12	38	0.06	27	11	127	0.1
gazeHMM-3	39	29	657	0.84	-	-	-	-	-	-	-	-
gazeHMM-4	-	-	-	-	27	12	46	0.22	30	20	153	0.5
EK	27	24	787	0.98	18	13	59	0.49	22	18	252	1.08
IDTk	28	19	258	0.49	34	11	6	0.58	26	19	53	0.63
IKF	70	40	356	1.52	62	29	21	1.45	62	25	107	1.22
IMST	19	12	336	0.77	13	5	13	1.1	20	10	76	0.77
IHMM	54	28	370	0.75	41	19	19	0.63	46	18	109	0.46
IVT	46	25	375	0.48	35	14	20	0.33	40	17	112	0.25
NH	56	21	344	0.59	44	14	33	0.39	47	17	104	0.46
LNS	33	13	390	0.36	27	11	42	0.28	30	10	122	0.4

Note. Durations are displayed in milliseconds. gazeHMM-3 classified three

and gazeHMM-4 classified four events. RMSD = root mean square deviation.

Table design adapted from Andersson et al. (2017).

Table 6

PSO Duration Descriptives and RMSD Between Algorithms and Human

Coders

Algorithm	Image				Moving dots				Video			
	Mean	SD	Events	RMSD	Mean	SD	Events	RMSD	Mean	SD	Events	RMSD
coderMN	25	14	313	0.68	14	5	24	0.63	23	13	97	0.43
coderRA	25	12	310	0.68	14	8	19	0.63	21	12	89	0.43
gazeHMM-3	14	15	518	1.48	-	-	-	-	-	-	-	-
gazeHMM-4	-	-	-	-	6	8	21	0.95	18	14	101	1.04
NH	31	15	237	1.54	23	13	11	1.25	31	20	78	1.19
LNS	30	15	319	1.19	20	8	21	0.54	30	19	87	0.82

Note. Durations are displayed in milliseconds. gazeHMM-3 classified three

and gazeHMM-4 classified four events. RMSD = root mean square deviation.

Table design adapted from Andersson et al. (2017).

Table 7

Smooth Pursuit Duration Descriptives and RMSD Between gazeHMM and Human Coders

	Image				Moving dots				Video			
Algorithm	Mean	SD	Events	RMSD	Mean	SD	Events	RMSD	Mean	SD	Events	RMSD
coderMN	363	187	3	0.23	370	238	36	0.4	559	391	51	0.14
coderRA	299	180	17	0.23	345	338	39	0.4	516	376	70	0.14
gazeHMM-4	-	-	-	-	23	22	400	1.97	21	23	1,281	1.95

Note. Durations are displayed in milliseconds. gazeHMM-3 classified three and gazeHMM-4 classified four events. RMSD = root mean square deviation. Table design adapted from Andersson et al. (2017).

agreement was substantial for the static and slight to fair for the dynamic conditions (Landis & Koch, 1977). For saccades, gazeHMM had the lowest agreement for the static condition and moderate agreement for the dynamic conditions. In absolute terms, the agreement was fair to moderate. Concerning PSOs, gazeHMM showed higher agreement than NH in the image and video conditions but consistently lower agreement compared to LNS. Absolutely, the agreement was slight (image) to moderate (video). Lastly, the agreement for smooth pursuit was lower compared to the human coders and fair in absolute values.

Concerning overall disagreement, Figure 18 shows that gazeHMM had less disagreement to the human coders across all events for all three conditions. For the static condition, the difference to most other algorithms was only slight (Med(Δ) = 2.65%), but for the dynamic conditions, the differences were substantial (video: Med(Δ) = 17.19%) and large (dots: Med(Δ) = 50.04%).

To explore which events gazeHMM classified differently than the average of human coder, we looked at the confusion matrix between the two (see Table 9). It can be seen that gazeHMM classified many fixations as smooth pursuits and vice versa. Moreover, it

Table 8

Cohen's Kappa Between Human Coders and Algorithms for Different Conditions and Events

Algorithm	Fixations			Saccades			PSOs			Smooth pursuits		
	Image	Dots	Video	Image	Dots	Video	Image	Dots	Video	Image	Dots	Video
coderMN	0.84	0.84	0.65	0.91	0.79	0.87	0.76	0.57	0.65	0.34	0.81	0.66
coderRA	0.84	0.84	0.65	0.91	0.79	0.87	0.76	0.57	0.65	0.34	0.81	0.66
gazeHMM-3	0.67	0	0	0.36	0	0	0.19	0	0	0	0	0
gazeHMM-4	0	0.16	0.17	0	0.62	0.51	0	0.24	0.46	0	0.28	0.2
CDT	0.38	0.07	0.11	0	0	0	0	0	0	0	0	0
EK	0	0	0	0.64	0.73	0.67	0	0	0	0	0	0
IDTk	0.36	0	0.03	0.45	0.25	0.38	0	0	0	0	0	0
IKF	0.63	0.04	0.14	0.58	0.43	0.59	0	0	0	0	0	0
IMST	0.38	0	0.03	0.54	0.31	0.52	0	0	0	0	0	0
IHMM	0.67	0.03	0.13	0.69	0.58	0.71	0	0	0	0	0	0
IVT	0.67	0.03	0.13	0.75	0.59	0.76	0	0	0	0	0	0
NH	0.52	-0.23	0.01	0.67	0.58	0.68	0.24	0.14	0.25	0	0	0
BIT	0.67	0.02	0.14	0	0	0	0	0	0	0	0	0
LNS	0	0	0	0.81	0.75	0.81	0.64	0.56	0.63	0	0	0

Note. gazeHMM-3 classified three and gazeHMM-4 classified four events.

Table design adapted from Andersson et al. (2017).

confused many PSOs with saccades. The heuristic to detect blinks seemed to work successfully since gazeHMM classified most blink samples in agreement with human coding and only a minor part was mistaken for saccades. Inspecting an example of gaze data classified by gazeHMM compared to human coding leads to a similar notion: Figure 19 illustrates that gazeHMM is rapidly switching between classifying fixations and smooth pursuits, whereas the human coder treats the same data as one large smooth pursuit event. In the example, the human coder also delayed the start of the PSO compared to gazeHMM.

Discussion

In this report, we presented gazeHMM, a novel algorithm for classifying gaze data into eye movement events. The algorithm models velocity, acceleration, and sample-to-sample angle signals with gamma distributions and a mixture of von Mises and a

Table 9

Confusion Matrix Between gazeHMM (Rows) and Human Coders (Columns) for Different Conditions

Event	Fixation	Saccade	PSO	Pursuit	Blink	Other
Image						
Fixations	0.88	0.08	0.08	0.72	0.07	0.03
Saccades	0.07	0.59	0.63	0.22	0.14	0.20
PSOs	0.04	0.08	0.23	0.03	0.05	0.16
Pursuits	0.00	0.00	0.00	0.00	0.00	0.00
Blinks	0.01	0.26	0.06	0.03	0.74	0.61
Moving dots						
Fixations	0.61	0.01	0.02	0.35	0.00	0.30
Saccades	0.01	0.78	0.60	0.03	0.00	0.48
PSOs	0.01	0.04	0.18	0.00	0.00	0.00
Pursuits	0.28	0.05	0.17	0.62	0.00	0.20
Blinks	0.10	0.12	0.04	0.00	0.00	0.02
Video						
Fixations	0.54	0.04	0.02	0.44	0.00	0.00
Saccades	0.02	0.57	0.35	0.02	0.14	0.82
PSOs	0.01	0.13	0.47	0.01	0.03	0.05
Pursuits	0.42	0.03	0.12	0.54	0.01	0.00
Blinks	0.00	0.23	0.04	0.00	0.82	0.14

Note. gazeHMM classified four events and blinks. Values indicate proportions of samples where gazeHMM and human coders agree divided by the total number of samples classified by the human coders for each event (i.e., columns sum to one).
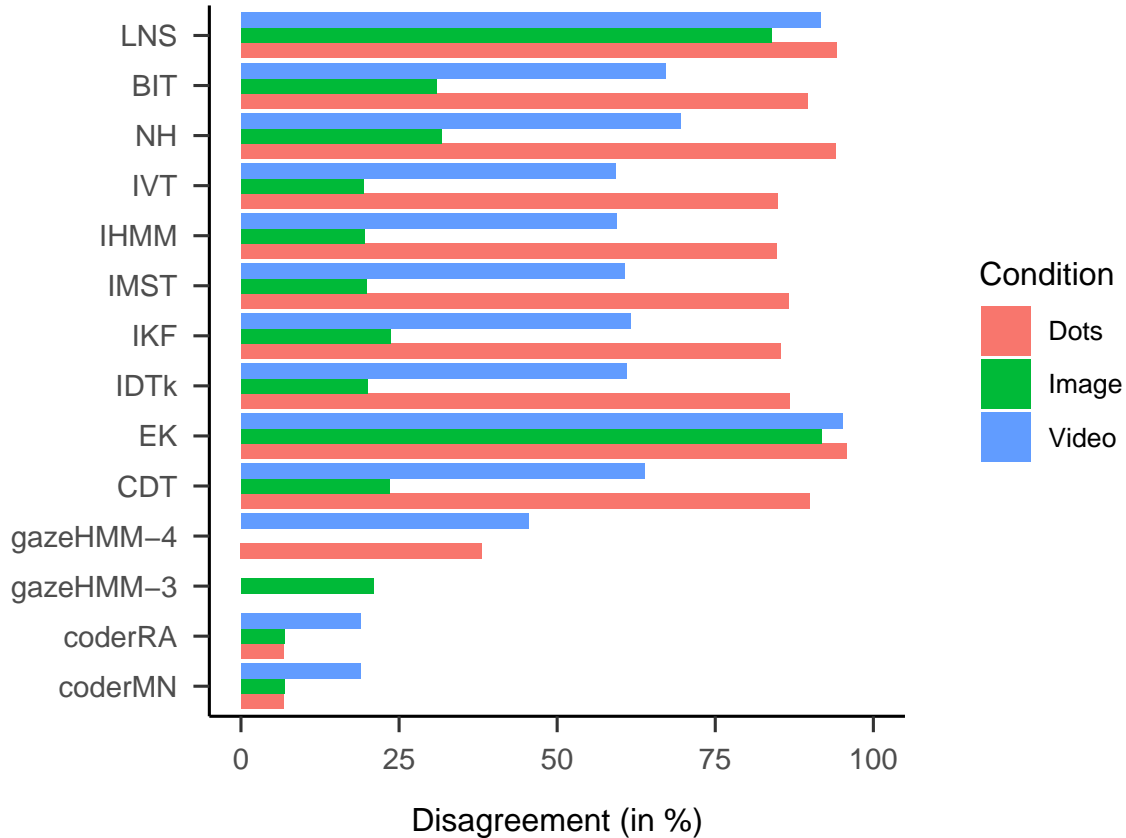
Figure 18. Disagreement between algorithms and human coders for different conditions (in %). gazeHMM-3 classified three and gazeHMM-4 classified four events.

uniform distribution. An HMM serves as the generative model of the algorithm and classifies the gaze samples into fixations, saccades, and optionally PSOs, and/or smooth pursuits. We showed in a simulation study that the generative model of gazeHMM recovered parameters and hidden state sequences well with a few exceptions. Adding a fourth event (i.e., smooth pursuit) to the model and noise to the generated data were the most critical factors for decreasing recovery. Furthermore, we applied gazeHMM with different numbers of states to benchmark data by Andersson et al. (2017) and compared the model fit. The model comparison revealed that a five-state HMM had consistently most likely generated the data. This result opposed our expectation that a three-state model would be preferred for static and a four-state model for dynamic data. When
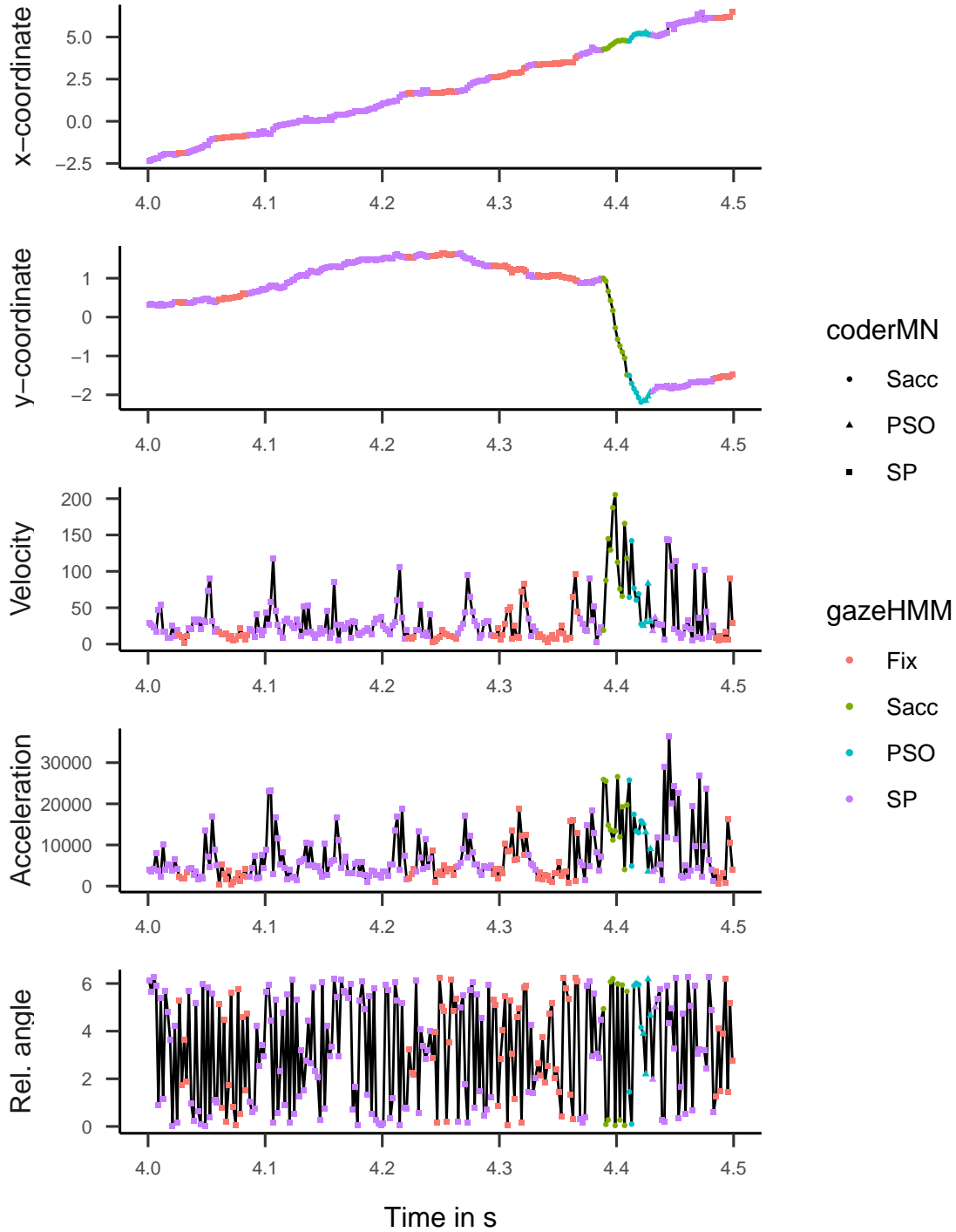
Figure 19. Example data from Andersson et al. (2017) displayed as x-, and y-coordinates, velocity (in deg/s), acceleration (in deg/s²), and sample-to-sample (relative) angle (in rad). Colors indicate classification by gazeHMM and symbols by human coder MN. Note that the human coder classified all fixations by gazeHMM as smooth pursuits and delayed the separation between the saccade and PSO.

comparing gazeHMM against other algorithms, gazeHMM showed mostly good agreement to human coding. On one hand, it outperformed the other algorithms in the overall disagreement to human coding. On the other hand, gazeHMM confused a lot of fixations with smooth pursuits, which led to a rapid switching between the two events. It also tended to mistake PSO samples as belonging to saccades.

Considering the results of the simulation study, it seems reasonable that adding the smooth pursuit state to the HMM decreased parameter and state recovery: It is the event that is overlapping most closely with another event (fixations) in terms of velocity, acceleration, and sample-to-sample angle. The overlap can cause the HMM to confuse parameters and hidden states. The decrease in parameter recovery (especially for scale parameters) due to noise shows that the overlap is enhanced by more dispersion in the data. The scale parameters might be particularly vulnerable to extreme data points. Despite these drawbacks, the recovery of the generative model in gazeHMM seems very promising. The simulation study gives also an approximate reference for the maximum recovery of hidden states that can be achieved by the HMM (Cohen's kappa values of ~1 for two, ~0.95 for three, and ~0.8 for four events).

The model comparison on the benchmark data suggested that the generative model in gazeHMM is not optimally specified for eye movement data, yet. There are several explanations for this result:

There were more eye movement events present in the data than we expected. Eye movement events can be divided into subevents. For example, fixations consist of drift and tremor movements (Duchowski, 2017) and PSOs encompass dynamic, static, and glissadic over- and undershoots (Larsson et al., 2013). A study on a recently developed HMM algorithm supports this explanation: Houpt et al. (2018) applied the unsupervised BP-AR-HMM algorithm to the Andersson et al. (2017) data set and classified more distinct states than the human coders. Some of the states classified by BP-AR-HMM matched the same event coded by humans. Since the subevents are usually not interesting

for users of classification algorithms, the ability of HMMs to classify might limit their ability to generate eye movements.

Model selection criteria are generally not appropriate for comparing HMMs with different numbers of states. This argument has been discussed in the field of ecology (see Li & Bolker, 2017), where studies found that selection criteria preferred models with more states than expected (similar to the result of this study; e.g., Langrock, Kneib, Sohn, & Deruiter, 2015). Li and Bolker (2017) explain this bias with the simplicity of the submodels in HMMs: Initial state, transition, and response models for each state are usually relatively simple. When they do not describe the processes in the respective states accurately, the selection criteria compensate for that by preferring a model with more states. Thus, there are not more latent states present in the data, but the submodels of the HMM are misspecified or too simple. Correcting for model misspecifications led to better model recovery in studies on animal movements (Langrock et al., 2015; Li & Bolker, 2017). However, Pohle et al. (2017) showed in simulations that the ICL identified the correct model despite several misspecifications. It has to be noted that the study by Pohle et al. (2017) only used data generating models with two states.

The submodels of gazeHMM were misspecified. Pohle et al. (2017) identified two scenarios in which model recovery using the ICL did not give optimal results: Outliers in the data and inadequate distributions in the response models. Both situations could apply to gazeHMM and eye movement data. Outliers occur frequently in eye-tracking data due to measurement error. Choosing adequate response distributions in HMMs is usually difficult and can depend on the individual and task from which the data are obtained from (Langrock et al., 2015). Moreover, gazeHMM only estimated intercepts for all parameters and thus, no time-varying covariates were included (cf. Li & Bolker, 2017). This aspect could indeed oversimplify the complex nature of eye movement data.

Comparing gazeHMM to other algorithms on benchmark data showed that gazeHMM showed good agreement with human coders. However, the evaluation criteria (RMSD of

event durations, sample-to-sample agreement, and overall disagreement) yielded different results. The fact that gazeHMM outperformed all other algorithms regarding the overall disagreement can be because it is the only algorithm classifying five events (incl. blinks). Nevertheless, Cohen's kappa values of 0.67 (fixations - image) or 0.62 (saccades - moving dots) indicate substantial agreement to human coders, especially in light of the maximum references from the simulation study. At this point it is important to mention that human coding should not be considered a gold standard in event classification: Hooge, Niehorster, Nyström, Andersson, and Hessels (2018) observed substantial differences between coders and within coders over time. Despite these differences, they recommend comparisons to human coding to demonstrate the performance of new algorithms and to find errors in their design.

Advantages of gazeHMM

In view of the four proposed goals that gazeHMM should fulfill, we can draw the following conclusions: Even though gazeHMM does require some parameter settings (in the pre- and postprocessing), they only have minor influence on the classification and are merely included to compensate for some drawbacks of the generative model. Their default values should be appropriate for most applications. A major advantage of gazeHMM is that it does not require human labeled data as input. Instead, it estimates all parameters and hidden states from the data. Since human coding is quite laborious, difficult to reproduce, and by times inconsistent (as noted earlier, Hooge et al., 2018), this property makes gazeHMM a good alternative to other recently developed algorithms that require human coded input (Bellet et al., 2019; Pekkanen & Lappi, 2017; Zemblys et al., 2018). This could also explain why the agreement to human coding is lower for gazeHMM than for algorithms that learn from human labeled data. Another advantage of gazeHMM is its ability to classify four eye movement events, namely fixations, saccades, PSOs, and smooth pursuit. Whereas most algorithms only parse fixations and saccades (Andersson et al.,

2017), few are able to classify PSOs (e.g., Zemblys et al., 2018), and even less categorize smooth pursuits (e.g., Pekkanen & Lappi, 2017). However, including smooth pursuits in gazeHMM led to some undesirable classifications on benchmark data, resulting in rapid switching between fixation and smooth pursuit events. Therefore, we recommend using gazeHMM with four events only for exploratory purposes. Without smooth pursuits, we consider gazeHMM's classification as appropriate for application. Lastly, its implementation in R using depmixS4 (Visser & Speekenbrink, 2010) should make gazeHMM a tool that is easy to use and customize for individual needs.

Future Directions

Despite its advantages, there are several aspects in which gazeHMM can be improved: First, a multivariate distribution could be used to account for the correlation between velocity and acceleration signals (for examples, see Balakrishnan & Lai, 2009). Potential problems of this approach might be choosing the right distribution and convergence issues (due to a large number of parameters). Another option to model the correlation could be to include one of the response variables as a covariate of the other.

Second, instead of the gamma being the generic (and potentially inappropriate) response distribution, a non-parametric approach could be used: Langrock et al. (2015) use a linear combination of standardized B-splines to approximate response densities, which led to HMMs with less states being preferred. This approach could potentially combat the problem of unexpectedly high-state HMMs being preferred for eye movement data.

Third, one solution to diverging results when comparing gazeHMM with different events could be model averaging: Instead of using the maximum posterior state probability of each sample from the preferred model, the probabilities could be weighted according to a model selection criterion (e.g., Schwarz weight) and averaged. Then, the maximum averaged probability could be used to classify the samples into events. This approach could lead to a more robust classification because it reduces the overconfidence of each competing

model and easily adapts to new data (analogous to Bayesian model averaging; Hinne, Gronau, Bergh, & Wagenmakers, 2020). However, the model comparison for gazeHMM often showed extreme weights for a five-state model, which would lead to a very limited influence of the other models in the averaged probabilities.

Fourth, including covariates of the transition probabilities and response parameters could improve the fit of gazeHMM on eye movement data. As pointed out earlier, just estimating intercepts of parameters could be too simple to model the complexity of eye movements. A candidate for such a covariate might be a periodic function of time (Li & Bolker, 2017) which could, for instance, capture the specific pattern of saccades. Whether covariates are improving the fit of submodels to eye movement data could in turn be assessed by inspecting pseudo-residuals and autocorrelation functions (see Zucchini et al., 2016).

Fifth, to avoid rapid switching between fixations and smooth pursuits as well as unreasonably short saccades, gazeHMM could explicitly model the duration of events. This can be achieved by setting the diagonal transition probabilities to zero and assign a distribution of state durations to each state (Bishop, 2006). Consequently, the duration distributions of fixations and smooth pursuits could differ from saccades and PSOs. This extension of the HMM is also called hidden semi-Markov model and has been successfully used by Mihali et al. (2017) to classify microsaccades. Drawbacks of this extension are higher computational costs and difficulties with including covariates (Zucchini et al., 2016).

Lastly, allowing constrained parameters in the HMM could replace some of the postprocessing steps in gazeHMM. This could potentially be achieved by using different response distributions or parameter optimization methods. Moreover, switching from the maximum likelihood to the Markov chain Monte Carlo (Bayesian) framework could help avoiding convergence problems with constrained parameters, but would also open new research questions about suitable priors for HMM parameters in the eye movement domain, efficient sampling plans, accounting for label switching, and computational efficiency,

naming only a few.

Conclusion

Both the simulation and validation studies showed that gazeHMM is a suitable algorithm for classifying eye movement events. For smooth pursuits, the classification is not optimal and thus not recommended, yet. On one side, the algorithm has some advantages over concurrent event classification algorithms, not relying on human-labeled input being the most important one. On the other side, it is not able to perform model comparisons as expected and still poses difficulties regarding good model specifications for different eye movement events. These difficulties highlight challenges for HMMs in general but also specifically for eye movements.

Open Practices and Data Availability Statement

The data and materials for the validation study (Andersson et al., 2017; Larsson et al., 2013, 2015) are available on GitHub at https://github.com/richardandersson/EyeMovementDetectorEvaluation. The code to obtain the results for the simulation and validation study is available at https://github.com/maltelueken/gazeHMM_validation and the algorithm can be accessed at https://github.com/maltelueken/gazeHMM. The simulation study was preregistered (https://doi.org/10.17605/OSF.IO/VDJGP), whereas the validation study was not.

References

Andersson, R., Larsson, L., Holmqvist, K., Stridh, M., & Nyström, M. (2017). One
    algorithm to rule them all? An evaluation and discussion of ten eye movement
    event-detection algorithms. *Behavior Research Methods*, *49*, 616–637.
    https://doi.org/10.3758/s13428-016-0738-9

Balakrishnan, N., & Lai, C.-D. (2009). *Continuous bivariate distributions* (2nd ed.). New
    York: Springer. https://doi.org/10.1007/b101765

Barzilai, J., & Borwein, J. M. (1988). Two-point step size gradient methods. *IMA Journal
    of Numerical Analysis*, *8*, 141–148. https://doi.org/10.1093/imanum/8.1.141

Bellet, M. E., Bellet, J., Nienborg, H., Hafed, Z. M., & Berens, P. (2019). Human-level
    saccade detection performance using deep neural networks. *Journal of
    Neurophysiology*, *121*, 646–661. https://doi.org/10.1152/jn.00601.2018

Biernacki, C., Celeux, G., & Govaert, G. (2000). Assessing a mixture model for clustering
    with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis
    and Machine Intelligence*, *22*(7), 719–726. https://doi.org/10.1109/34.865189

Birgin, E. G., Martinez, J. M., & Raydan, M. (2000). Nonmonotone spectral projected
    gradient methods on convex sets. *SIAM Journal of Optimization*, *10*, 1196–1211.
    https://doi.org/10.1137/S1052623497330963

Bishop, C. M. (2006). *Pattern recognition and machine learning.* New York: Springer.

Blignaut, P. (2009). Fixation identification: The optimum threshold for a dispersion
    algorithm. *Attention, Perception, & Psychophysics*, *71*(4), 881–895.
    https://doi.org/10.3758/APP.71.4.881

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from
    incomplete data via the em algorithm. *Journal of the Royal Statistical Society.
    Series B (Methodological)*, *19*(1), 1–38.

https://doi.org/10.1111/j.2517-6161.1977.tb01600.x

Duchowski, A. T. (2017). *Eye tracking methodology* (3rd ed.). London: Springer.
https://doi.org/10.1007/978-3-319-57883-5

Ehinger, B. V., Groß, K., Ibs, I., & Peter, K. (2019). A new comprehensive eye-tracking
test battery concurrently evaluating the Pupil Labs glasses and the EyeLink 1000.
bioRxiv. https://doi.org/10.1101/536243

Forney Jr, G. D. (1973). The viterbi algorithm. *Proceedings of the IEEE*, *61*(3), 268–278.
https://doi.org/10.1109/PROC.1973.9030

Hein, O., & Zangemeister, W. H. (2017). Topology for gaze analyses - Raw data
segmentation. *Journal of Eye Movement Research*, *10*(1), 1–25.
https://doi.org/10.16910/jemr.10.1.1

Hinne, M., Gronau, Q. F., Bergh, D. van den, & Wagenmakers, E.-J. (2020). A conceptual
introduction to Bayesian model averaging. *Advances in Methods and Practices in
Psychological Science*, *3*(2), 200–215. https://doi.org/10.1177/2515245919898657

Hooge, I. T. C., Niehorster, D. C., Nyström, M., Andersson, R., & Hessels, R. S. (2018). Is
human classification by experienced untrained observers a gold standard in fixation
detection? *Behavior Research Methods*, *50*, 1864–1881.
https://doi.org/10.3758/s13428-017-0955-x

Houpt, J. W., Frame, M. E., & Blaha, L. M. (2018). Unsupervised parsing of gaze data
with a beta-process vector auto-regressive hidden Markov model. *Behavior Research
Methods*, *50*, 2074–2096. https://doi.org/10.3758/s13428-017-0974-7

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy.
International Journal of Forecasting, *22*(4), 679–688.
https://doi.org/10.1016/j.ijforecast.2006.03.001

Kasneci, E., Kasneci, G., Kübler, T. C., & Rosenstiel, W. (2014). The applicability of

probabilistic methods to the online recognition of fixations and saccades in dynamic scenes. *Proceedings of the Symposium on Eye Tracking Research and Applications*, 323–326. https://doi.org/10.1145/2578153.2578213

Komogortsev, O. V., Gobert, D. V., Jayarathna, S., Koh, D. H., & Gowda, S. M. (2010). Standardization of automated analyses of oculomotor fixation and saccadic behaviors. *IEEE Transactions on Biomedical Engineering*, *57*(11), 2635–2645. https://doi.org/10.1109/TBME.2010.2057429

Landis, R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, *33*(1), 159–174.

Langrock, R., Kneib, T., Sohn, A., & Deruiter, S. L. (2015). Nonparametric inference in hidden Markov models using P-splines. *Biometrics*, *71*(2), 520–528. https://doi.org/10.1111/biom.12282

Larsson, L., Nystrom, M., & Stridh, M. (2013). Detection of saccades and postsaccadic oscillations in the presence of smooth pursuit. *IEEE Transactions on Biomedical Engineering*, *60*(9), 2484–2493. https://doi.org/10.1109/TBME.2013.2258918

Larsson, L., Nyström, M., Andersson, R., & Stridh, M. (2015). Detection of fixations and smooth pursuit movements in high-speed eye-tracking data. *Biomedical Signal Processing and Control*, *18*, 145–152. https://doi.org/10.1016/j.bspc.2014.12.008

Leigh, R. J., & Zee, D. S. (2015). *The neurology of eye movements* (5th ed.). Oxford University Press.

Li, M., & Bolker, B. M. (2017). Incorporating periodic variability in hidden Markov models for animal movement. *Movement Ecology*, *5*(1), 1–12. https://doi.org/10.1186/s40462-016-0093-6

Ligges, U., Short, T., & Kienzle, P. (2015). *Signal: Signal processing.* Retrieved from http://r-forge.r-project.org/projects/signal/

Lund, U., & Agostinelli, C. (2018). CircStats: Circular statistics, from "Topics in circular statistics" (2001). Retrieved from https://cran.r-project.org/web/packages/CircStats/

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

Mihali, A., Opheusden, B. van, & Ma, W. J. (2017). Bayesian microsaccade detection. *Journal of Vision*, *17*(1), 1–23. https://doi.org/10.1167/17.1.13

Nyström, M., & Holmqvist, K. (2010). An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data. *Behavior Research Methods*, *42*(1), 188–204. https://doi.org/10.3758/BRM.42.1.188

Olsson, P. (2007). *Real-time and offline filters for eye tracking* (Master's thesis). Royal Institute of Technology Stockholm. Retrieved from http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=pubmed&cmd=Retrieve&dopt=AbstractPlus&list_uids=12935495724606901081related:WUvHitMchLMJ

Pekkanen, J., & Lappi, O. (2017). A new and general approach to signal denoising and eye movement classification based on segmented linear regression. *Scientific Reports*, 1–13. https://doi.org/10.1038/s41598-017-17983-x

Pohle, J., Langrock, R., Beest, F. M. van, & Schmidt, N. M. (2017). Selecting the number of states in hidden Markov models: Pragmatic solutions illustrated using animal movement. *Journal of Agricultural, Biological, and Environmental Statistics*, *22*(3), 270–293. https://doi.org/10.1007/s13253-017-0283-8

R Core Team. (2020). *R: A language and environment for statistical computing.* Wien: R Foundation for Statistical Computing. Retrieved from https://www.r-project.org/

Salvucci, D. D., & Goldberg, J. H. (2000). Identifying fixations and saccades in eye-tracking protocols. *Proceedings of the Symposium on Eye Tracking Research and*

Applications, 71–78. https://doi.org/10.1145/355017.355028

Savitzky, A., & Golay, M. J. E. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, *36*(8), 1627–1639. https://doi.org/10.1021/ac60214a047

Schulte-Mecklenbeck, M., Johnson, J. G., Böckenholt, U., Goldstein, D. G., Russo, J. E., Sullivan, N. J., & Willemsen, M. C. (2017). Process-tracing methods in decision making: On growing up in the 70s. *Current Directions in Psychological Science*, *26*(5), 442–450. https://doi.org/10.1177/0963721417708229

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, *6*(2), 461–464.

Shic, F., Scassellati, B., & Chawarska, K. (2008). The incomplete fixation measure. *Proceedings of the Symposium on Eye Tracking Research and Applications*, 111–114. https://doi.org/10.1145/1344471.1344500

Stuart, S., Hickey, A., Vitorio, R., Welman, K., Foo, S., Keen, D., & Godfrey, A. (2019). Eye-tracker algorithms to detect saccades during static and dynamic tasks: A structured review. *Physiological Measurement*, *40*(2), 1–26. https://doi.org/10.1088/1361-6579/ab02ab

Tafaj, E., Kasneci, G., Rosenstiel, W., & Bogdan, M. (2012). Bayesian online clustering of eye movement data. *Proceedings of the Symposium on Eye Tracking Research and Applications*, 285–288. https://doi.org/10.1145/2168556.2168617

Varadhan, R., & Gilbert, P. (2009). BB: An R package for solving a large system of nonlinear equations and for optimizing a high-dimensional nonlinear objective function. *Journal of Statistical Software*, *32*(4), 1–26. Retrieved from http://www.jstatsoft.org/

Visser, I. (2011). Seven things to remember about hidden Markov models: A tutorial on

Markovian models for time series. *Journal of Mathematical Psychology*, *55*(6), 403–415. https://doi.org/10.1016/j.jmp.2011.08.002

Visser, I., & Speekenbrink, M. (2010). depmixS4: An R package for hidden Markov models. *Journal of Statistical Software*, *36*(7), 1–21. https://doi.org/10.18637/jss.v036.i07

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, *13*(2), 260–269. https://doi.org/10.1109/TIT.1967.1054010

Wagenmakers, E. J., & Farrell, S. (2004). AIC model selection using Akaike weights. *Psychonomic Bulletin and Review*, *11*(1), 192–196. https://doi.org/10.3758/BF03206482

Zemblys, R., Niehorster, D. C., Komogortsev, O., & Holmqvist, K. (2018). Using machine learning to detect events in eye-tracking data. *Behavior Research Methods*, 160–181. https://doi.org/10.3758/s13428-017-0860-3

Zucchini, W., MacDonald, I. L., & Langrock, R. (2016). *Hidden Markov models for time series - An introduction using R* (2nd ed.). Boca Raton, FL: Chapman & Hall. https://doi.org/10.1201/b20790