The Partially Observable Hidden Markov Model with Application to Keystroke Biometrics

John V. Monaco and Charles C. Tappert

Abstract—This work introduces the partially observable hidden Markov model (POHMM), a generalization of the hidden Markov model (HMM) in which the underlying system state is partially observable through event metadata at each time step. Whereas in a HMM, the hidden state is inferred through the observed values, the hidden state in a POHMM is inferred through both the observed values and a partially observed state. Despite an explosion in the number of model parameters, parameter estimation can still be performed in time that scales linearly with the number of observations as demonstrated by the derived parameter estimation algorithm. Marginal distributions of the model act as a fallback mechanism when novel sequences are encountered in likelihood estimation. A parameter smoothing technique is used to account for missing data during parameter estimation and simultaneously reduce the degrees of freedom of the model to avoid overfitting, especially for a small number of observations. The POHMM is applied to keystroke time intervals, in which the user can be in either an active or passive hidden state, and the keyboard key names partially reveal the underlying system state. The proposed model is shown to consistently outperform the standard HMM in keystroke biometric identification and verification tasks for various types of keystroke input and is generally preferred over the HMM in a Monte Carlo goodness of fit test.

Index Terms—hidden Markov model, time intervals, behavioral biometrics, keystroke dynamics

I. INTRODUCTION

Time intervals from human actions are typically bursty with many short intervals separated by few long ones [3]. Events are often non-uniformly distributed in time due to a number of complex underlying processes, including various mechanisms for temporal processing [27] and representations of time [28]. Humans maintain both implicit and explicit representations of time for different tasks. It is believed that movement initiation is explicitly timed, while the duration of movement is implicitly timed [28]. Additionally, humans perform actions across a wide range of time scales separated by orders of magnitude, as exemplified by Newell’s time scale of human action [17].

Time interval biometrics utilize the timestamps from a sequence of timed events for the purpose of identification and verification. The time intervals, or the time between events, is of interest. Let \( t_n \) be the time of the \( n^{th} \) event. The series of time intervals is given by

\[
\tau_n = t_n - t_{n-1}.
\]

The original event times can be reconstructed from the time intervals and the time of the first event, \( t_0 \). In the case an event has duration, i.e., it is not instantaneous, then \( t_n \) marks the time of onset. Events may also contain additional information, such as intensity and location, which can be incorporated into a temporal model to capture spatiotemporal behavior.

The goal of this work is to develop a model of temporal behavior. In the simplest realization of nonhomogeneous behavior, a user can be in one of two behavioral states corresponding to active and passive periods of activity, and the resulting time intervals are dependent on the underlying system state. Often in human time interval data, there is also some incomplete information about the underlying system state. This usually comes in the form of spatial information or the type of event that took place, such as the key that was pressed on a keyboard. In a two-state model of human behavior, the probability of being in either an active or passive state may be greater depending on which key was pressed. Certain keys, such as punctuation and the Space key, indicate a greater probability of being in a passive state as the typist often pauses between words and sentences as opposed to between letters in a word [22]. This reasoning extends to other activities, such as email, where a user might be more likely to pause after sending an email instead of receiving an email, and programming, where a user may fix bugs quicker than making feature additions.

This work introduces the partially observable hidden Markov model (POHMM), a generalization of the hidden Markov model (HMM) in which the underlying system state is partially observable through event metadata at each time step. Whereas in a HMM, the hidden state is inferred through the observed values, the hidden state in a POHMM is inferred through both the observed values and a partially observed state. Despite an explosion in the number of model parameters, parameter estimation can still be performed in time that scales linearly with the number of observations as demonstrated by the derived parameter estimation algorithm. Marginal distributions act as a fallback mechanism when novel sequences are encountered in likelihood estimation, and a proposed parameter smoothing technique is used to account for missing data during parameter estimation and simultaneously reduce the degrees of freedom of the model to avoid overfitting.

The rest of this article is organized as follows. Section II reviews previous modeling efforts for latent processes with partial observability. Section III reviews the standard HMM and defines a two-state HMM for time intervals. Section IV describes the POHMM, followed by a case study of the POHMM applied to keystroke time intervals in Section V.
Finally, Section VI concludes the article. The POHMM is implemented in the pohmm Python package\(^1\).

II. MODELING PARTIALLY OBSERVABLE STATES

There have been various generalizations of the standard HMM (Figure 1) to deal with hidden states that are partially observable in some way. These models are referred to as partly-HMM [12], partially-HMM [18], and context-HMM [8]. For clarity, these models have been redrawn in the figure using the notation of this paper.

The partly-HMM (Figure 1a) is a second order model in which the first state is hidden and the second state is observable [12]. In the partly-HMM, both the hidden state and observation at time \( t \) are dependent on the observation at time \( t-1 \). The partly-HMM can be applied to problems that have a transient underlying process, such as gesture and speech recognition, as opposed to a piecewise stationary process that the HMM assumes [9]. Parameter estimation can be performed by expectation maximization (EM), similar to the HMM.

Partially observable states can also come in the form of partial and uncertain ground truth regarding the hidden state at each time step. The partially-HMM (Figure 1b) deals with this scenario, in which an uncertain hidden state label may be observed at each time step [18]. The probability of observing the uncertain label and the probability of the label being correct, were the true hidden state known, are controlled by parameters \( p_{obs} \) and \( p_{true} \), respectively. Thus, the probability of observing a correct label is \( p_{obs} \times p_{true} \). This model is motivated by language modeling applications in which manually labeling data is expensive and time consuming. Ground truth state labels may help in parameter estimation although they may be incorrect or missing due to human error [15]. Similar to the HMM, the EM algorithm can be used for estimating the parameters of the partially-HMM [18].

Past observations can also provide context for the transition and emission probabilities in a HMM. In [8], Forchhammer proposed the context-HMM, in which the transition and emission probabilities at time \( t \) are conditioned on an observed context function. The context functions are defined as \( A_V (x_t) = v_t \) and \( B_V (x_{t-1}) = w_t \) for the transition and emission probabilities, respectively, where the context sequences \( v_t \) and \( w_t \) are functions of the observation sequence. At each time step, the hidden state and observation are dependent on a context from the previous time step, shown in Figure 1c. The context-HMM has information theoretic motivations, with applications such as image compression [7]. Used in this way, the neighboring pixels in an image can provide context for the emission and transition probabilities.

There are two scenarios in which the current models fall short. The first is when there is missing data during parameter estimation, such as a missing observations or context, and the second is when there is missing or novel data during likelihood calculation. An elegant solution in both these scenarios calls for explicit marginal emission and transition distributions, where the context or uncertain label is marginalized out. The model proposed in this work, described in Section IV, has explicit marginal distributions that act as a fallback mechanism when observations with missing partial states are encountered. This leads directly to a parameter smoothing technique that restricts overfitting and allows the model to be applied to sequences with very few observations that would otherwise lead to degenerate distributions.

III. HIDDEN MARKOV MODEL

The HMM was developed over several decades, dating back to 1960 when the forward-backward procedure was first introduced [24]. The theory was later made accessible to non-statisticians in the well-cited tutorial work, [19]. The HMM soon became popular for modeling behavioral data, such as

\(^1\)Source code, examples, and installation instructions are available at https://github.com/vmonaco/pohmm
speech, handwriting, gesture, and linguistics. More recently, it has been used to model temporal behavior, such as keystroke dynamics and terrorist activity [21], [20].

The HMM is a finite-state model in which observed values at time \( t \) depend on an underlying hidden state. The hidden state may or may not correspond to some physical state of the system, depending on the application of interest. In speech, the hidden states generally have no physical correspondence, whereas the hidden states in a model of temporal behavior might correspond to the activity levels of the system [13], [20].

The system advances in discrete steps, typically with a first-order dependency between the hidden states. At the \( n^{th} \) time step \( t_n \), a feature vector \( x_n \) is observed and the system can be in any one of \( M \) hidden states. Let \( x_{1:N} \) be the complete observation sequence from times 0 to \( T \), where \( N \) is the total number of observations. The HMM is unsupervised since ground truth about the hidden state at each time step is generally not available. The latent variable \( z_n \) is introduced to represent the hidden state at time \( t_n \). The structure of the HMM is shown in Figure 2.

The model starts in state \( j \) at time 0 with probability \( \pi_j \) and transitions from state \( i \) to state \( j \) with probability \( a_{ij} \). The transition matrix is denoted by \( A = [a_{ij}] \) and starting probability vector by \( \pi = [\pi_j] \). The stationary probability of being in state \( j \) is given by \( \Pi_j \), where

\[
\Pi_j = \sum_{1 \leq i \leq M} \Pi_i a_{ij} .
\]  

(2)

The stationary probability vector \( \Pi \) can be determined by taking any row from repeated powers of the transition matrix,

\[
\lim_{N \to \infty} A
\]  

as the values in column \( j \) converge to the stationary probability for state \( j \).

While in state \( j \) at time \( t_n \), the system emits an observation vector \( x_n \) distributed according to some density function \( f(\cdot; b_j) \) parametrized by vector \( b_j \). The emission distribution can be either continuous, discrete, or a mix of both. The HMM is completely described by the number of states \( M \), starting probabilities \( \pi \), transition matrix \( A \), and emission distribution parameters \( b \). The model parameters are given by \( \theta = \{ \pi, A, b \} \).

There are generally three problems associated with the HMM [19].

1) Determine \( P(x_{1:N}^T | \theta) \), the likelihood of an observation sequence, given model parameters \( \theta \).
2) Determine \( z_{1:N}^T \), the maximum likelihood sequence of hidden states, given model parameters \( \theta \) and an observation sequence.
3) Determine \( \arg \max_{\theta \in \Theta} P(x_{1:T}^T | \theta) \), the maximum likelihood parameters \( \theta \), given an observation sequence.

The first and third problems are necessary for identifying and verifying users in biometric applications, while the second problem is useful in understanding user behavior. The rest of this section reviews the solutions to each of these problems and defines a two-state HMM for time intervals.

### A. Model likelihood

Calculating \( P(x_{1:T}^T | \theta) \), the likelihood of an observation sequence \( x_{1:T}^T \) for a given model, is necessary for user identification and verification. Let the forward variable \( \alpha_j(n) \) be the probability of the partial observation sequence \( x_{1:n}^T \) and state \( j \) at time \( t_n \), given model parameters \( \theta \). This can be computed inductively by Algorithm 1.

#### Algorithm 1 HMM forward algorithm.

1) Initialization: \( \alpha_j(1) = f(x_1; b_j)\pi_j \)
2) Induction: \( \alpha_j(n + 1) = \left( \sum_{i=1}^{M} \alpha_i(n) a_{ij} \right) f(x_{n+1}; b_j) \)
3) Termination: \( P(x_{1:T}^T | \theta) = \sum_{j=1}^{M} \alpha_j(N) \)

There are several ways of handling the underflow errors that will eventually occur as \( N \) increases in Algorithm 1 due to the floating point values becoming infinitesimally small. Intermediary values may be scaled or calculated in log-space. For long observation sequences, typically the loglikelihood is used. The order of computations required for the forward algorithm is \( O(M^2 N) \), since it requires \( M^2 \) calculations for each observation vector.

### B. Hidden state prediction

It is generally not necessary to know the sequence of hidden states to perform user identification and verification, although this function is useful in other applications. This requires estimating \( z_{1:N}^T \), the most likely sequence of hidden states, given the observation sequence \( x_{1:T}^T \) and parameters \( \theta \). This is accomplished by the Viterbi algorithm, a dynamic programming algorithm that determines the most likely hidden state at each time step \( t_n \).

Similar to the forward variable in the previous section, the backward variable \( \beta \) is introduced, where \( \beta_j(n) \) is the probability of the partial observation sequence \( x_{n+1:T}^T \) and state \( j \) at time \( t_n \), given the model parameters \( \theta \). Like the forward algorithm, the backward algorithm is \( O(M^2 N) \), shown in Algorithm 2.

With both the forward and backward variables, it is straightforward to calculate the posterior probability of being in state \( j \) at time \( t_n \), given the observation sequence and model parameters. This is given by the forward-backward variable \( \gamma_j(n) \).
Algorithm 2 HMM backward algorithm.

1) **Initialization**: \( \beta_j(N) = 1 \)
2) **Induction**: \( \beta_j(n) = \sum_{j=1}^{M} a_{ij} f(x_{n+1}; b_j) \beta_j(n+1) \)
3) **Termination**: \( P(x_1^N|\theta) = \sum_{j=1}^{M} \beta_j(1) \pi_j \)

\[
\gamma_j(n) = \frac{\alpha_j(n) \beta_j(n)}{P(x_1^N|\theta)} = \frac{\alpha_j(n) \beta_j(n)}{\sum_{i=1}^{M} \alpha_i(n) \beta_i(n)}
\]

where \( 1 \leq n \leq N \). The most likely hidden state at time \( t_n \) can then be determined by

\[
z_n = \arg \max_{1 \leq j \leq M} \gamma_j(n).
\]

### C. Parameter estimation

Parameter estimation is one of the most important problems associated with the HMM. The goal is to determine \( \arg \max_{\theta \in \Theta} P(x_1^N|\theta) \), the maximum likelihood (ML) parameters, given observed data \( x_1^N \). This requires estimating the starting probabilities, transition probabilities, and emission distribution parameters. The starting and transition parameters have closed-form solutions, while the formula for the emission distribution parameters. The starting and transition parameters can be determined directly from \( \gamma_\cdot(n) \), while summing \( \xi_{ij}(n) \) over \( n \) gives the expected number of transitions from state \( i \) to state \( j \). The variables \( \gamma_j(n) \) and \( \xi_{ij}(n) \) are used to update the model parameters. The re-estimated starting probabilities are determined directly from \( \gamma_j(1) \),

\[
\hat{\pi}_j = \gamma_j(1).
\]

The transition matrix is updated by the formula

\[
\hat{\beta}_{ij} = \frac{\sum_{n=1}^{N-1} \xi_{ij}(n)}{\sum_{n=1}^{N-1} \gamma_i(n)}
\]

and the updated stationary probabilities are given by

\[
\hat{\Pi}_j = \frac{\sum_{n=1}^{N} \gamma_j(n)}{\sum_{i=1}^{M} \sum_{n=1}^{N} \gamma_i(n)}.
\]

The re-estimates for parameter vectors \( b_j \), \( 1 \leq j \leq M \), depend on the density function \( f(\cdot) \), where

\[
\hat{b}_j = \arg \max_{b \in \mathcal{B}} \sum_{n=1}^{N} \gamma_j(n) \ln f(x_a; b)
\]

and \( \mathcal{B} \) is the parameter space of \( f(\cdot) \). The complete parameter estimation procedure, commonly referred to as the Baum-Welch (BW) algorithm, is shown in Algorithm 3.

Algorithm 3 HMM Baum-Welch algorithm for parameter estimation.

1) **Initialization**
   Choose initial parameters \( \theta^0 \) and let \( \hat{\theta} \leftarrow \theta^0 \)
2) **Expectation**
   Use \( \hat{\theta} \) and \( x_1^N \) to compute \( \alpha_j(n), \beta_j(n), \gamma_j(n), \xi_{ij}(n) \), and let \( \hat{P} \leftarrow P(x_1^N|\theta) \)
3) **Maximization**
   Update \( \pi, A, \) and \( b \) using the re-estimation formulae and let \( \hat{\theta} \leftarrow \{ \hat{\pi}, \hat{A}, \hat{b} \} \)
4) **Termination**
   If \( P(x_1^N|\theta) - \hat{P} < \epsilon \) then terminate and let \( \hat{\theta} \leftarrow \hat{\theta} \), otherwise go to step 2

### D. Hidden Markov model for time intervals

Since the simplest realization of nonhomogeneous human behavior assumes two discrete states, a two-state HMM is defined in which hidden states correspond to the active and passive states of the user. This two-state model is introduced here for the purpose of motivating and providing a benchmark against which the model proposed in Section IV will be compared to. The resulting model is shown in Figure 3.

![Fig. 3. Two-state hidden Markov model.](image)

Human-generated time intervals generally follow a heavy-tailed distribution and are well described by a log-normal [23], [3], [2]. This serves as motivation for choosing the log-normal as the density function for time intervals \( \tau \), given by
where \( \eta \) is the log-mean and \( \rho \) is the log-standard deviation. Note that a mixture of log-normals also gives rise to a log-normal. The log-normal parameter re-estimation formulae are

\[
f(x; \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(\frac{-(x - \mu)^2}{2\sigma^2}\right)
\]

respectively. A normal distribution can be used for spatial features, where the normal density is defined as

\[
f(x; \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(\frac{-(x - \mu)^2}{2\sigma^2}\right)
\]

Parameters \( \mu \) and \( \sigma \) are re-estimated similarly by

\[
\hat{\mu}_j = \frac{\sum_{n=1}^{N} \gamma_j(n) x_n}{\sum_{n=1}^{N} \gamma_j(n)}
\]

and

\[
\hat{\sigma}_j^2 = \frac{\sum_{n=1}^{N} \gamma_j(n)(x_n - \hat{\mu}_j)^2}{\sum_{n=1}^{N} \gamma_j(n)}
\]

where \( x_n \) is a scalar component of \( x_n \). Note that this definition assumes independence between features since covariance terms are not considered.

**IV. Partially observable hidden Markov model**

In time interval data, there is often some additional information accompanying each event that gives an indication as to the hidden state. This could be the name of a key that was pressed while typing, the type of commit made to a source code repository, or some other symbol that indicates the type of action that occurred. Certain types of actions may indicate a greater probability of being in a particular hidden state. For example in keystroke dynamics, the probability of being in a passive state should be greater when the Space key is pressed than any letter key since the typist is more likely to pause between words than between letters. The keyboard key name is a partially observed state, which partially reveals the hidden state of the system. This phenomenon has been observed in transcription typists [22], however the HMM defined in the previous section lacks this dependence on the type of event that occurred.

Let \( \omega \) be a partially observed state symbol that comes from a finite alphabet of size \( m \). In the partially observable hidden Markov model, the true system state is a latent variable and depends on both the partially observed state \( \omega \) and the previous hidden state. The POHMM dependency structure is shown in Figure 4.

The partially observed state introduces a dependence into the HMM parameters defined in the previous section. Specifically, model parameters include \( \pi_{j|\omega} \), the probability of starting in state \( j \), given partially observed state \( \omega \), and \( b_{j|i|\omega} \) for the parametrized emission distribution \( f(\cdot; b_{j|i|\omega}) \) that depends on partially observed state \( \omega \). Similarly, the probability of transitioning between states \( i \) and \( j \), given partially observed states \( \omega \) and \( \psi \), is denoted by \( a_{ij|\omega,\psi} \). A summary of POHMM parameters and variables is shown in Table I.

![Figure 4. Partially observable hidden Markov model structure.](image-url)

**TABLE I**

| Parameter or Variable | Description |
|-----------------------|-------------|
| \( x_n^N \) | Sequence of observed values, where \( x_n \) may be a scalar or a vector |
| \( \Omega_n^N \) | Sequence of partially observable states, where the length of \( \Omega_n^N \) is \( N \) and \( \Omega_n \) is the partially observed state at time \( t_n \) |
| \( m \) | The number of unique partially observed states in \( \Omega_n^N \) |
| \( \omega \) or \( \psi \) | Partially observed state |
| \( z_n \) | Hidden state of the system at time \( t_n \) |
| \( M \) | The number of hidden states |
| \( a_{ij|\omega,\psi} \) | Probability of transitioning from state \( i \) to state \( j \), given observed partially observed state \( \omega \) while in state \( i \) and \( \psi \) in state \( j \) |
| \( \pi_{j|\omega} \) | Probability of being in state \( j \) at time \( t \), given observed partially observed state \( \omega \) |
| \( \Pi_{j|\omega} \) | Stationary probability of being in state \( j \), given partially observed state \( \omega \) |
| \( b_{j|i|\omega} \) | Observation distribution parameters in state \( j \), given partially observed state \( \omega \) |
| \( \gamma_{j|\omega}(n) \) | Posterior probability of being in state \( j \), given observations and partially observed state \( \omega \) at time \( t_n \) |
| \( \xi_{ij|\omega,\psi}(n) \) | Posterior probability of transitioning from state \( i \) at time \( t_n \) to state \( j \) at time \( t_{n+1} \), given partially observed state \( \omega \) and \( \psi \) at times \( t_n \) and \( t_{n+1} \), respectively |

Marginal distribution parameters can also be defined, where the partially observed state is marginalized out. Let \( \pi_{j|\omega} \) and \( f(\cdot; b_{j|i|\omega}) \) be the marginalized starting probability and emission probability, respectively. Similarly, the parameters \( a_{ij|\omega,\psi} \),
Proceeds similar to the HMM, using the partially observed B. Hidden state prediction

Computation of POHMM marginal distributions is covered in Section IV-E.

While the total number of parameters in the HMM is $M + M^2 + MK$, where $K$ is the number of free parameters in the emission distribution $f(\cdot)$, the POHMM contains $m \times (M + mM^2 + MK)$ parameters. After accounting for normalization constraints, the number of degrees of freedom (dof) is $m \times (M - 1 + mM(M - 1) + MK)$, corresponding to the number of free parameters. This will reduce further after parameter smoothing strategies are applied, as discussed in Section IV-F.

A. Model likelihood

The POHMM likelihood is computed by a modified forward procedure, similar to the HMM. At each time step $t_n$, the system emits a partially observable state $\Omega_n$ and is in 1 out of $M$ hidden states. The model likelihood is given by $P(x_1^n|\theta, \Omega_1^n)$, which is the probability of the observation sequence, given both the model parameters $\theta$ and the sequence of partially observed states $\Omega_1^n$. This can be calculated using the conditional model parameters $\pi_{j|\omega}$, $a_{ij|\omega, \psi}$, and $b_j|\psi$ for hidden states $1 \leq j \leq M$ and partially observed states $\omega, \psi \in \Omega_1^n$. The modified forward algorithm and backward algorithm for the POHMM are shown in Algorithms 4 and 5, respectively.

The model likelihood may be obtained by either algorithm upon termination. Note that like the HMM, the modified forward and backward algorithms both take $O(M^2N)$ time to compute. The forward variable $\alpha_{j|\Omega_n}(n)$ is the probability of observing the sequence $x_1^n$ and being in state $j$ at time $t_n$, given the model parameters $\theta$ and the partially observed state sequence $\Omega_1^n$ with partially observed state $\Omega_n$ at time $t_n$. Similarly, the backward variable $\beta_{j|\Omega_n}(n)$ is the the probability of the observing the sequence $x_1^n$ and being in state $j$ at time $t_n$, given the model parameters $\theta$ and the partially observed state sequence $\Omega_1^n$ with partially observed state $\Omega_n$ at time $t_n$.

\textbf{Algorithm 4} POHMM forward algorithm.

1) \textbf{Initialization:} $\alpha_{j|\Omega_1}(1) = f(x_1; b_j|\Omega_1) \pi_{j|\Omega_1}$

2) \textbf{Induction:} 

$$\alpha_{j|\Omega_{n+1}}(n + 1) = \left( \sum_{i=1}^{M} a_{ij|\Omega_n(n)} a_{ij|\Omega_{n+1}} f(x_{n+1}; b_j|\Omega_{n+1}) \right) \frac{\alpha_{j|\Omega_n(n)}(n)}{f(x_{n+1}; b_j|\Omega_{n+1})}$$

3) \textbf{Termination:} $P(x_1^n|\theta, \Omega_1^n) = \sum_{j=1}^{M} \alpha_{j|\Omega_1}(N)$

B. Hidden state prediction

Determination of the most likely sequence of hidden states proceeds similar to the HMM, using the partially observed state-dependent parameters. First, the posterior probability of

$$P(\Omega_1^n|x_1^n; \theta) = \frac{P(x_1^n|\Omega_1^n, \theta) P(\Omega_1^n)}{P(x_1^n)}$$

being in state $j$ at time $t_n$, given partially observed state $\Omega_n$, is defined using the POHMM forward and backward variables

$$\gamma_{j|\Omega_n}(n) = \frac{\alpha_{j|\Omega_n}(n) \beta_{j|\Omega_n}(n)}{P(x_1^n|\theta, \Omega_1^n)}$$

Hidden states are then taken as the maximum likelihood states at each time step.

$$z_n = \arg \max_{1 \leq j \leq M} \gamma_{j|\Omega_n}(n) \quad (19)$$

C. Parameter estimation

Parameter estimation is similar to the HMM, where the POHMM uses a modified Baum-Welch algorithm. Using the modified forward-backward variable given by Equation 18, the POHMM starting probabilities are

$$\hat{\pi}_{j|\omega} = \gamma_{j|\Omega_1}(1) \quad (20)$$

where $\omega = \Omega_1$.

Generally, it may not be possible to estimate $\hat{\pi}_{j|\omega}$ for many $\omega$ due to there only being one $\Omega_1$ (or several $\Omega_1$ for multiple observation sequences). To deal with this, parameter smoothing is introduced in Section IV-F.

To complete the equations for updating parameters in the modified Baum-Welch algorithm, the POHMM analogue of $\xi_{ij}(t)$ defined for the HMM in Equation 6 is needed. Let $\xi_{ij|\Omega_n, \Omega_{n+1}}(n)$ be the probability of transitioning from state $i$ at time $t_n$ to state $j$ at time $t_{n+1}$, given the partially observed states $\Omega_n$ at time $t_n$ and $\Omega_{n+1}$ at time $t_{n+1}$ as well as the observation sequence and model parameters $\theta$. Using the POHMM forward and backward variables, this is given by

$$\xi_{ij|\Omega_n, \Omega_{n+1}}(n) = \frac{\alpha_{ij|\Omega_n}(n) a_{ij|\Omega_{n+1}} f(x_{n+1}; b_j|\Omega_{n+1}) \beta_{j|\Omega_{n+1}}(n + 1)}{P(x_1^n|\theta, \Omega_1^n)}$$

$$1 \leq n \leq N - 1.$$ 

Note that computing $\xi_{ij|\omega, \psi}(n)$ for the POHMM can be performed in linear time in the number of observations, $O(M^2N)$, since the partially observed states are not enumerated at each time step.
Next, the transition probabilities are estimated. In contrast to the HMM, which has $M^2$ transition probabilities, there are $m^2M^2$ transition probabilities in the POHMM. Computing the updated transition probabilities can be performed in $O(M^2N)$ time. The equation is

$$
\hat{a}_{ij|\omega,\psi} = \frac{\sum_{n \in \mathcal{T}_{\omega,\psi}} \xi_{ij|\omega_n,\omega_{n+1}}(n)}{\sum_{n \in \mathcal{T}_{\omega,\psi}} \gamma_{i|\omega_n}(n)},
$$

where $\mathcal{T}_{\omega,\psi} = \{n|\omega_n = \omega, \omega_{n+1} = \psi\}$ is the set of indexes where the partially observed state $\omega$ is observed at time $t_n$ and $\psi$ is observed at time $t_{n+1}$. Note that $\hat{a}_{ij|\omega,\psi}$ requires only the transitions between partially observed states $\omega$ and $\psi$.

Estimating the emission distribution parameters depends on the density function $f(\cdot)$, where the new estimates are given by

$$
b_{j|\omega} = \arg\max_{b \in B} \sum_{n \in \mathcal{T}_\omega} \gamma_{j|\omega_n}(n) \ln f(x_n; b),
$$

where $\mathcal{T}_\omega = \{n|\omega_n = \omega\}$

As an example, the normal emission distribution parameters are estimated for the POHMM, conditioned on the partially observed state. Using the forward-backward variable from Equation 18.

$$
\hat{\mu}_{j|\omega} = \frac{\sum_{n \in \mathcal{T}_\omega} \gamma_{j|\omega_n}(n)x_n}{\sum_{n \in \mathcal{T}_\omega} \gamma_{j|\omega_n}(n)},
$$

$$
\mathcal{T}_\omega = \{n|\omega_n = \omega\}
$$

and

$$
\hat{\sigma}_{j|\omega}^2 = \frac{\sum_{n \in \mathcal{T}_\omega} \gamma_{j|\omega_n}(n)(x_n - \hat{\mu}_{j|\omega})^2}{\sum_{n \in \mathcal{T}_\omega} \gamma_{j|\omega_n}(n)},
$$

$$
\mathcal{T}_\omega = \{n|\omega_n = \omega\}
$$

are the updated mean and variance estimates for hidden state $j$, given partially observed state $\omega$. Note that the estimates depend only on the elements of $\gamma_{\omega_n}(n)$ where $\omega_n = \omega$. The log-normal POHMM emission parameters are re-estimated similarly.

The modified Baum-Welch algorithm, which uses Equations 20, 22, and 23 to update model parameters in each iteration, is shown in Algorithm 6. The convergence criterion is a threshold $\epsilon$ on the loglikelihood reduction. The rest of this section deals with other aspects of parameter estimation, including initialization, marginal distributions, and parameter smoothing.

### D. Parameter initialization

Parameter initialization is an important step in the Baum-Welch algorithm and may ultimately determine the quality of the estimated parameters. Initial parameters may either be chosen randomly or derived from a set of observed values.

| Algorithm 6: POHMM modified Baum-Welch algorithm. |
|--------------------------------------------------|
| 1) **Initialization**                             |
| Choose initial parameters $\theta^0$ and let $\hat{\theta} \leftarrow \theta^0$ |
| 2) **Expectation**                               |
| Use $\hat{\theta}$, $x_1^N$, $\Omega_1^N$ to compute $\alpha_{ij|\Omega_n}(n)$, $\beta_{ij|\Omega_{n+1}}(t)$, $\gamma_{j\Omega_n}(n)$, $\xi_{ij\Omega_n,\Omega_{n+1}}(n)$, let $\hat{P} \leftarrow P(x_1^N|\theta, \Omega_1^N)$ |
| 3) **Maximization**                              |
| Update $\pi$, $A$, and $B$ using the re-estimation formulae and let $\hat{\theta} \leftarrow \{\hat{\pi}, \hat{A}, \hat{B}\}$ |
| 4) **Termination**                              |
| If $P(x_1^N|\hat{\theta}, \Omega_1^N) - \hat{P} < \epsilon$ then terminate and let $\hat{\theta} \leftarrow \hat{\theta}$, otherwise go to step 2 |

In this work a fixed parameter initialization strategy is used, which guarantees reproducible parameter estimates.

The starting and transition probabilities are simply initialized as

$$
\pi_{j|\omega} = \frac{1}{M}
$$

and

$$
a_{ij|\omega,\psi} = \frac{1}{M}
$$

for all $i$, $j$, $\omega$, and $\psi$. This reflects the belief of equal probabilities in the absence of any particular $\omega$ in $\Omega_1^N$.

Next, the emission density parameters are initialized. The strategy proposed here is to initialize parameters in such a way that there is a correspondence between hidden states from two different models. That is, for any two models A and B, hidden state $j = 1$ corresponds to the active state and $j = 2$ corresponds to the passive state. Using a log-normal emission distribution for time intervals, this is accomplished by spreading the log-mean initial parameters. Let

$$
\eta_\omega = \frac{\sum_{n \in \mathcal{T}_\omega} \ln x_n}{|\mathcal{T}_\omega|}, \quad \mathcal{T}_\omega = \{n|\omega_n = \omega\}
$$

and

$$
\rho_\omega^2 = \frac{\sum_{n \in \mathcal{T}_\omega} (\ln x_n - \eta_\omega)^2}{|\mathcal{T}_\omega|}, \quad \mathcal{T}_\omega = \{n|\omega_n = \omega\}
$$

be the log-mean and log-variance of observations $x_1^N$ conditioned on the partially observed state $\omega$. The model parameters are then initialized as

$$
n_{j|\omega} = \eta_\omega + \left(\frac{2h(j-1)}{M-1} - \frac{1}{h}\right) \times \rho_\omega
$$

and

$$
\rho_{j|\omega}^2 = \rho_\omega^2
$$

for $1 \leq j \leq M$, where $h$ is a bandwidth parameter that determines the spread. In a two-state model, this ensures that state $j = 1$ corresponds to the active state, i.e., the state with the smaller log-mean time interval.
E. Marginal distributions

In order for the POHMM to handle missing or novel partially observed states during likelihood calculation, the marginal distributions are used. When computing the likelihood of a novel sequence, it is possible that some partially observed states in the novel sequence were not observed during parameter estimation. For example, this situation can occur when partially observed states correspond to key names of freely-typed text and novel keys are observed during testing. A fallback mechanism (sometimes referred to as a “backoff” model) is typically employed to handle missing data during training and novel data during testing, such as those used in keystroke [16], [25] and linguistics [11]. The POHMM marginal distributions, in which the partially observed state is marginalized out, are analogous to a two-level fallback hierarchy where missing or novel partially observed states fall back to the marginals.

Let the probability of observing partially observed state $\omega$ at time $t_1$ be $\pi_\omega$, and the probability of transitioning from partially observed state $\omega$ to $\psi$ be denoted by $a_{\omega,\psi}$. Both of these can be computed directly from the partially observed state sequence $\Omega_N^t$. The marginal probability $\pi_j$, is the probability of starting in hidden state $j$, in which the partially observed state has been marginalized out, given by

$$
\pi_j = \sum_{\omega \in \Omega} \pi_{j|\omega} \pi_\omega
$$

(32)

where $\Omega$ is the set of unique partially observed states in $\Omega_N^t$. Marginal transition probabilities can also be calculated. Let $a_{ij|\omega}$ be the probability of transitioning from hidden state $i$ to hidden state $j$, given the observed partially observed state $\omega$ while in hidden state $i$. The second partially observed state for hidden state $j$ has been marginalized out. This probability is given by

$$
a_{ij|\omega} = \sum_{\psi \in \Omega} a_{ij|\omega,\psi} a_{\omega,\psi}
$$

(33)

The marginal probability $a_{ij|\omega}$ is defined similarly by

$$
a_{ij|\omega} = \sum_{\omega \in \Omega} a_{ij|\omega,\psi} a_{\omega,\psi}
$$

(34)

Finally, the marginal $a_{ij..}$ is the probability of transitioning from hidden state $i$ to $j$,

$$
a_{ij..} = \frac{1}{m} \sum_{\omega \in \Omega} \sum_{\psi \in \Omega} a_{ij|\omega,\psi} a_{\omega,\psi}
$$

(35)

No denominator is needed in Equation 33 since the normalization constraints of both transition matrices carry over to the left-hand side. Equation 35 is normalized by $\frac{1}{m}$ since $\sum_{\omega \in \Omega} \sum_{\psi \in \Omega} a_{\omega,\psi} = m$ where $m$ is the number of unique partially observed states in $\Omega$.

The marginal emission distribution is a mixture of the emissions of the partially observed states. For a normal or log-normal emission, the marginal emission is simply a mixture of normals or log-normals, respectively. Let $\mu_{ij}$, and $\sigma_{ij}^2$ be the mean and variance of the marginal distribution for hidden state $j$. The marginal mean is a weighted sum of the partially observed state distributions, given by

$$
\mu_{ij} = \sum_{\omega \in \Omega} \Pi_\omega \mu_{ij|\omega}
$$

(36)

where $\Pi_\omega$ is the stationary probability of observing partially observed state $\omega$. This can be calculated directly from the partially observed state sequence $\Omega_N^t$, where

$$
\Pi_\omega = \frac{1}{N} \sum_{n=1}^{N} I(\Omega_n = \omega)
$$

(37)

and $I(\cdot)$ is the indicator function. Similarly, the marginal variance is given by

$$
\sigma_{ij}^2 = \sum_{\omega \in \Omega} \Pi_\omega \left( (\mu_{ij|\omega} - \mu_{ij..})^2 + \sigma_{ij|\omega}^2 \right).
$$

(38)

Calculation of the marginalized log-normal distribution parameters is exactly the same.

F. Parameter smoothing

While the marginal distributions can be used to handle missing or novel data during likelihood calculation, parameter smoothing handles missing or infrequent data during parameter estimation. The purpose of parameter smoothing is twofold. First, it reduces the dof of the model to avoid overfitting, a problem often encountered when there is a large number of parameters and small amount of data. Second, parameter smoothing provides superior estimates in the case of missing or infrequent data. For motivation, consider a keystroke letter sequence of length $N$. There are at most 27 unique keys that can be observed, including the space key, and $27 \times 27$ unique digrams (subsequences of length 2). Most of these will rarely, or never, be observed in a typing sample of English text. Additionally, it is possible that some sequences encountered in the testing phase were not observed during parameter estimation. The POHMM handles this sparsity by mixing the conditional and marginal distributions for each set of parameters.

After parameter smoothing, each parameter in the POHMM becomes a weighted average with the corresponding marginal parameters. There are several different weighting strategies that can be used. An inverse frequency strategy uses the partially observed state frequencies to define the weights as

$$
w_\omega = 1 - \frac{1}{1 + f(\omega)}
$$

(39)

where $f(\omega) = \sum_{t=1}^{N} I(\Omega_n = \omega)$, the frequency of partially observed state $\omega$ in the observed sequence $\Omega_N^t$. This ensures that parameters dependent on infrequent partially observed states rely more heavily on the marginal distribution, whereas the parameters for frequently occurring partially observed states remain independent in expectation. Alternatively, the stationary probabilities can be used as the weights, where

$$
w_\omega = \Pi_\omega.
$$

(40)
Other weighting strategies are possible, such as using fixed weights. The POHMM starting probabilities are smoothed by

$$\pi_j|\omega = w_\omega \pi_j|\omega + (1-w_\omega)\pi_j,$$

for each $\omega \in \Omega$. This ensures that the starting probability conditioned on infrequent or missing partially observed states is estimated using some knowledge from the marginalized starting probability. Similarly, emission parameters are smoothed estimated using some knowledge from the marginalized start-

$$\Omega$$

ditioned on infrequent or missing partially observed states is

$$w_\omega, \psi = \frac{1}{f(\omega, \psi) + f(\psi)}$$

Weights for the conditional and marginal transition probabilities are defined as

$$w_{\omega, \psi} = \frac{1}{f(\omega, \psi) + f(\omega)}$$

The smooth parameters transition weights for transition probabilities follow similar formulae. Let $f(\omega, \psi)$ be the frequency of partially observed state $\omega$ followed by $\psi$ in the observed sequence $\Omega^T$. Weights for the conditional and marginal transition probabilities are defined as

$$w_{\omega, \psi} = 1 - \left(\frac{1}{f(\omega, \psi) + f(\omega)} + \frac{1}{f(\omega, \psi) + f(\psi)}\right)$$

$$w_\omega = 0$$

where $w_{\omega, \psi} + w_{\omega, \psi} + w_{\omega, \psi} + w_\omega = 1$. The smoothed parameter transition matrix is given by

$$a_{ij|\omega, \psi} = w_{\omega, \psi} a_{ij|\omega, \psi} + w_\omega a_{ij|\omega} + w_{\omega, \psi} a_{ij|\psi} + w_{\omega, \psi} a_{ij}.$$ \hspace{3pt} (41)

In this strategy, the weight for the marginal $a_{ij|\omega, \psi}$ is 0, although in other strategies this could be non-zero. The POHMM parameter estimation procedure, including marginal calculations and parameter smoothing, is given by Algorithm 7.

\begin{algorithm}
\begin{enumerate}
\item Initialization
Choose initial parameters $\theta^0$ and let $\theta \leftarrow \theta^0$
\item Expectation
Use $\theta$, $x^N$, $\Omega^N$ to compute $a_{ij|\Omega^N(n)}$, $\beta_{j|\Omega^N(n)}$, $\gamma_{j|\Omega^N(n)}$, $\xi_{ij|\Omega^N,\Omega^N+1(n)}$, let $\hat{P} \leftarrow P(x^N|\theta, \Omega^N)$
\item Maximization
Update $\pi$, $A$, and $b$ using the re-estimation formulae and let $\hat{\theta} \leftarrow \{\hat{\pi}, \hat{A}, \hat{b}\}$
\item Marginal distributions
Calculate marginal distributions
\item Parameter smoothing
Calculate smoothing weights and smooth the parameters with marginals
\item Termination
If $P(x^N|\hat{\theta}, \Omega^N) - \hat{P} < \epsilon$ then terminate and let $\hat{\theta} \leftarrow \hat{\theta}$, otherwise go to step 2
\end{enumerate}
\end{algorithm}

\section{V. Case study}

Four keystroke datasets are analyzed in this work, summarized in Table II. They consist of long free-text, long fixed-

\begin{table}[ht]
\centering
\begin{tabular}{|c|c|c|c|c|c|}
\hline
Dataset & Ref & Users & S/U & K/S & $\tau$ (ms) \\
\hline
Free-text & [26] & 56 & 6 & 748±98 & 310 ms \\
Fixed-text & [16] & 60 & 4 & 100 & 263 ms \\
Keypad & [1] & 30 & 20 & 11 & 376 ms \\
Mobile & [10] & 52 & 20 & 11 & 366 ms \\
\hline
\end{tabular}
\caption{Keystroke data summary. S/U=Samples per User, K/S=Keystrokes per Sample.}
\end{table}

text and short fixed-text input collected on desktop and laptop keyboards, as well as short fixed-text input collected on a mobile device. The long free-text (free-text) dataset contains input from users who answered essay-style questions as part of a class exercise [26]. A subset of the original dataset was randomly selected to obtain 56 users who answered 10 questions each. Of this dataset, each sample contained $748\pm98$ keystrokes. The long fixed-text dataset (fixed-text) contains 60 users who each copied 4 different nursery rhymes or fables [16], [26]. Each sample is truncated so that it contains exactly 100 keystrokes. The short fixed-text dataset (kepad) contains keystrokes recorded using only the keypad on a desktop keyboard [1]. This dataset contains 30 users who entered the phone number 9141937761 followed by the Enter key. Each user provided 20 correct entries, and each sample contains 11 keystrokes (10 digits + Enter). Incorrect entries were discarded. The mobile short fixed-text dataset (mobile) contains keystrokes recorded on a mobile device from 52 users entering the same digit sequence, 9141937761, followed by the Enter key [10]. Data was recorded on 5 identical LG- 

D820 Nexus 5 devices which have 4.95 inch touch screens and 1080x1920 pixel resolution. Each keystroke is described by the key that was pressed, the press time, and the duration. The mobile dataset additionally contains a number of sensor features for each keystroke, such as acceleration, pressure, and touchscreen location.

The two-state model assumes that the user can be in either a passive or active state while typing, and the key names partially reveal the underlying state of the user. The POHMM can take advantage of this by using the key names as partially observed states. For example, a typist is more likely to pause between words after the space key is pressed, or between sentences after the period key is pressed. Therefore, the probability of being in a passive state should be higher when the space or period keys are press compared to when letter keys are pressed.

There are several ways to define the partially observed states for keystroke data. For short fixed-text, a partially observable state can be created for each key on the keyboard and for long text, the keys can be grouped into several categories. This avoids an unnecessarily large number of partially observed states for long text. The two types of partially observable states are summarized as follows.

\textbf{Key type} Keys are grouped as: left letters, right letters,
Space, Shift, Period, Comma, and other. Left letters include “qwertasdfgzxcvb” and right letters include “yuiopjklmn”. Each key group is a partially observed state. This strategy is used for free-text and long fixed-text.

**Key name** Each individual key is a partially observed state.

This strategy is used for short fixed-text.

An example, the log-normal time interval distributions of the fitted POHMM are shown in Figure 5 for both fixed-text and free-text keystroke samples. The passive state in the free-text example has a heavier tail than the fixed-text, while the active state distributions in both examples are comparable.

![Fig. 5. Keystroke time interval distribution examples showing the model time interval distribution in milliseconds of each state and the empirical densities of the classified events as determined by the modified Viterbi algorithm.](image)

The POHMM is fit to each sample in the long fixed-text and long free-text datasets, and the mean model parameters are shown in Table III. The mean time intervals $\tau$ correspond to the press-press latencies, and the mean duration (key hold time) is given by $d$. The mean proportion of events in each state as determined by the modified Viterbi algorithm is also shown. On average, 60% of keystrokes in fixed text and 75% of keystrokes in free text are in an active state. In both datasets, left-hand keys take longer to press and are held down longer than right-hand keys. Longer time intervals for the Shift, Period, and Comma keys are observed in both states. In both states and both datasets, the duration of the Shift key is about twice as long as other keys since it is typically used as a modifier (i.e., for upper case letters and special symbols).

### A. Goodness of fit

To determine whether the proposed model is consistent with observed data, a goodness of fit test is performed through Monte Carlo hypothesis testing. The test proceeds as follows. For each keystroke sample (using the press-press latency time intervals only), the model parameters $\theta_m$ are determined. The area test statistic between the model and empirical distribution is then taken. The area test statistic is a compromise between the Kolmogorov-Smirnov (KS) test and Cramér-von Mises test [14], given by

$$A = \int |P_D(\tau) - P_M(\tau; \theta_m)|d\tau$$

where $P_D$ is the empirical cumulative distribution and $P_M$ is the model cumulative distribution.

The marginal density of the model is given by

$$g(x; \theta) = \sum_{j=1}^{2} \Pi_j f(x; b_j)$$ (42)

where $\Pi_j$ is the stationary probability of being in state $j$ and $f$ is the marginal emission probability. Using the fitted model with parameters $\theta_m$, a surrogate data sample the same size as the empirical data is generated. The surrogate data is then treated similarly to the empirical data, and estimated parameters $\theta_s$ are determined using the surrogate observations. The area test statistic between the surrogate-data-trained model and surrogate data is computed, given by $A_s$. This process repeats until enough surrogate statistics have accumulated to reliably determine $Pr(|A_s - \langle A_s \rangle| > |A - \langle A_s \rangle|)$. The biased $P$ value is given by

$$I(|A_s - \langle A_s \rangle| > |A_m - \langle A_s \rangle|) + 1 \over m + 1$$ (43)

where $I(\cdot)$ is the indicator function. The null hypothesis (that the model is consistent with the data) is tested for each sample in each dataset. Each test requires fitting $S + 1$ models (1 empirical and $S$ surrogate samples).

A Monte Carlo goodness of fit test is performed for time intervals only. The results indicate that the POHMM is more consistent with the data compared to a HMM, in which key names are ignored. The proportion of samples in each dataset for which the model is rejected is shown in Table IV. The p-value distributions indicate that the POHMM is better suited for fixed-text, keypad, and mobile input, while both the HMM and POHMM are largely rejected for long free-text input.

### TABLE IV

**Keystroke goodness of fit test results showing the proportion of users that reject the null hypothesis that the model is consistent with the data.**

|       | HMM  | POHMM |
|-------|------|-------|
| Keystroke (fixed) | 0.32 | 0.23 |
| Keystroke (free)  | 0.98 | 0.89 |
| Keypad            | 0.77 | 0.40 |
| Mobile            | 0.56 | 0.13 |

### B. Identification and verification

Identification accuracy (ACC) is measured by the proportion of correctly classified samples. The user of the model with the highest likelihood is taken as the label for an unknown sample. Verification accuracy is measured by the equal error rate (EER) and area under curve (AUC) of the receiver operating characteristic (ROC) derived from the scores of all users. A verification decision using the POHMM proceeds as follows. The likelihood of a query sample is obtained under the model corresponding to the identity of the claimed user. This score is then normalized by the minimum and maximum likelihoods under every other model in the system, i.e., the models of every other user. The resulting normalized score is compared to a decision threshold, which varies between 0 and 1. If the score is less than the threshold, then the
TABLE III
KEYSTROKE AVERAGE ESTIMATED EMISSION PARAMETERS FOR EACH PARTIALLY OBSERVED STATE (KEY TYPE) IN MILLISECONDS.

|                  | Fixed-text | Free-text |
|------------------|------------|-----------|
|                  | Active (60%) | Active (75%) | Passive (40%) | Passive (25%) |
|                  | [Left] | [Right] | Space | Shift | Period | Comma | [Other] | [Left] | [Right] | Space | Shift | Period | Comma | [Other] |
| \( r \)          | 134  | 131   | 131   | 143   | 236    | 263   | 209    | 152   | 147   | 142   | 183   | 284    | 260   | 187    |
| \( d \)          | 100   | 90    | 98    | 185   | 92     | 94    | 86     | 100   | 90    | 100   | 213   | 86     | 92    | 80     |

![Cumulative distribution for p-value](image)

![Cumulative distribution for p-value](image)

![Cumulative distribution for p-value](image)

![Cumulative distribution for p-value](image)

When dealing with behavioral biometrics, it usually makes more sense to consider the continuous verification performance of a system [5]. In this work, continuous verification is enforced through a penalty function in which each event incurs a non-negative penalty within a rolling window. The user is rejected from the system if the cumulative penalty incurred within the sliding window exceeds a given threshold. The penalty at any given time can be thought of as the inverse of trust. The sliding window ensures that the maximum achievable penalty is bounded, and may actually decrease as behavior becomes more consistent with a model. An alternative to the penalty function is the penalty-and-reward function [4]. The approach introduced here avoids having to determine the distance threshold and reward parameters in [4]. The sliding window replaces the reward, since penalties outside the window do not contribute towards the cumulative penalty.

The penalty of each event is determined as follows. The marginal probability of each new event, given the preceding events, can be obtained from the forward lattice, \( \alpha \), given by

\[
P(x_{t+1} | x_1^t) = P(x_{t+1}^t) - P(x_1^t) \tag{44}
\]

When a new event is observed, the likelihood is obtained under each model in the system. The likelihoods are ranked, with the highest model given a rank of 0, and the lowest a rank of \( U - 1 \), with \( U \) models in the system. The rank of the claimed user’s model is the incurred penalty. Thus, if a single event is correctly matched to the user’s model, a penalty of 0 is incurred. If it scores the second highest likelihood, a penalty of 1 is incurred, and so on. The rank penalty is added to the cumulative penalty in the sliding window, while penalties outside the window are discarded.

The continuous verification performance is reported as the number of events that can occur before an impostor is detected. This is determined by adjusting the penalty threshold so that the genuine user is never locked out of the system. Since the genuine user’s penalty is always below the threshold, this is the maximum number of events that an impostor can execute before being rejected by the system while the genuine user is never rejected.

An example of the penalty function for genuine and impostor users is shown in Figure 7. The decision threshold is set to the maximum penalty incurred by the genuine user so that a false rejection does not occur. The average penalty for impostor users, with 95% CI, is shown. In this particular example, the impostor penalties exceed the decision threshold after about 20 keystrokes.

---

Some authors use the true positive rate (TPR) instead of the FRR. In this case, a perfect classifier would have an AUC of 1 and a random classifier an AUC of 0.5.
For each sample, the maximum rejection time (MRT) is determined for each impostor. This procedure is summarized as follows.

1) Determine the maximum penalty threshold such that the genuine user is not rejected.
2) Determine the number of keystrokes it takes each impostor to be rejected.
3) Repeat Steps 1-2 to obtain the MRT for each sample. The average MRT (AMRT) is the average time it takes an impostor to be rejected. Since the MRT is obtained for each sample, a confidence interval for the MRT can be obtained.

Identification and verification results are obtained for each dataset. For each evaluation metric, a stratified cross-fold validation is performed with the number of folds equal to the number of samples per user in each dataset. Thus, there is one sample from each user in each fold. Figure 8 shows the ROC curves obtained for each dataset using the HMM and POHMM. Table V summarizes identification and verification performance results for each keystroke dataset using both the HMM and POHMM. Results for a subset of the fixed-text keystroke dataset are included here, in which users had to copy a nursery rhyme containing 100-150 characters. The POHMM significantly (p < 0.05) outperforms the HMM in almost every comparison. For fixed-text tasks, impostors are locked out of the system after 25 keystrokes on average. For free-text input, it takes about 4 times as many keystrokes for an impostor to be detected. On mobile devices an impostor can typically be detected after only 1 keystroke, which corresponds to 0 lockout time since the single keystroke results in a penalty score above the lockout threshold.

**VI. Discussion**

This work introduced the POHMM, an extension of the HMM in which hidden states are partially observable through event types. A subclass of the POHMM is the HMM in which the event types are all the same or possibly unknown. The POHMM marginal distributions can account for missing data during testing, while parameter smoothing helps to avoid overfitting and accounts for missing data during training. Computational complexities of the POHMM parameter estimation and likelihood calculation algorithms are comparable to that of the HMM, which are linear in the number of observations.

The POHMM is different from the partly-HMM [12], being a first order model, and different from the partially-HMM [18], since it doesn’t assume a partial labeling. In particular, the POHMM makes the following assumptions.

1) There is some additional information associated with each event, such as the event type. This represents the partially observed state.
2) There is a shared underlying stochastic process between events with different types. This represents the hidden state.

**TABLE V**

| Model | Fixed-text | Free-text | Keypad | Mobile with sensors |
|-------|------------|-----------|--------|---------------------|
| M=Model, H=HMM, P=POHMM. Datasets: A=nursery rhymes, B=Fixed-text keystroke, C=Free-text keystroke, D=Keypad, E=Mobile without sensors, F=Mobile with sensors. |
| | | | | |
| M | ACC | EER | AUC | AMRT |
| A | H | 0.88 (0.05) | 0.03 (0.02) | 0.01 (0.00) | 34.79 (19.91) |
| | P | 0.99 (0.02) | 0.00 (0.01) | 0.00 (0.00) | 25.27 (13.04) |
| B | H | 0.56 (0.04) | 0.10 (0.03) | 0.05 (0.02) | 30.41 (14.88) |
| | P | 0.75 (0.07) | 0.08 (0.04) | 0.03 (0.02) | 24.55 (13.54) |
| C | H | 0.53 (0.07) | 0.13 (0.03) | 0.05 (0.02) | 162.0 (128.5) |
| | P | 0.82 (0.05) | 0.06 (0.01) | 0.02 (0.01) | 103.2 (122.5) |
| D | H | 0.58 (0.08) | 0.12 (0.02) | 0.05 (0.01) | 4.24 (1.83) |
| | P | 0.77 (0.06) | 0.05 (0.02) | 0.02 (0.01) | 3.21 (1.98) |
| E | H | 0.42 (0.05) | 0.15 (0.02) | 0.07 (0.01) | 5.23 (2.02) |
| | P | 0.59 (0.05) | 0.10 (0.02) | 0.04 (0.01) | 4.16 (2.15) |
| F | H | 0.93 (0.04) | 0.02 (0.01) | 0.00 (0.00) | 0.16 (0.68) |
| | P | 0.94 (0.02) | 0.01 (0.01) | 0.00 (0.00) | 0.15 (0.64) |
3) Some events have missing or unknown partially observed states.

4) Novel partial states are observed during testing.

The POHMM is most similar to the context-HMM [8] in the sense that emission and transition probabilities are conditioned on some observed values. There are several important differences between the POHMM and context-HMM. In the POHMM, the hidden state at time $t$ is dependent on the partially observed state at times $t$ and $t-1$, whereas the hidden state of the context-HMM at time $t$ is dependent only on the context at time $t-1$. Additionally, the context sequences are defined as functions of the observed values, which differ from the partially observed states of the POHMM; the POHMM partially observed states are composed of metadata that generally capture the event types. For missing partially observed states, the marginal distributions of the POHMM also act as a fallback mechanism, whereas the context-HMM does not account for missing or undefined context. This is an important distinction since novel partially observed state sequences may be observed during testing. The marginal distributions of the POHMM are also used in parameter smoothing, which reduces the dof of the model and provides superior parameter estimates with small amounts of data.

The application of the POHMM extends beyond keystroke biometrics to time intervals that may be obtained from other sources. There are a number of reasons to utilize time intervals as a biometric, motivating the development of the temporal behavioral model in this work. Timestamped events from human behavior are truly ubiquitous. In the Information Age, most human-computer interactions generate timestamped events in some way. From the keys pressed on a keyboard and the transmission of an email message, to the submission of a research article through an online submission system, a timestamp is generated and stored for each event. In many scenarios, timestamps can be observed without user cooperation or knowledge, further increasing the ubiquity of human temporal behavior.

ACKNOWLEDGMENT

This research was supported in part by an appointment to the Postgraduate Research Participation Program at the U.S. Army Research Laboratory administered by the Oak Ridge Institute for Science and Education through an inter agency agreement between the U.S. Department of Energy and USAERL.

REFERENCES

1. Ned Bakelman, John V Monaco, Sung-Hyuk Cha, and Charles C Tappert. Keystroke biometric studies on password and numeric keypad input. In European Intelligence and Security Informatics Conference (EISIC), pages 204–207. IEEE, 2013.

2. A-L. Barabási, K-I Goh, and A Vazquez. Reply to comment on "the origin of bursts and heavy tails in human dynamics". arXiv preprint physics/0511186, 2005.

3. Albert-Laszlo Barabási. The origin of bursts and heavy tails in human dynamics. Nature, 435(7039):207–211, 2005.

4. Patrick Bours. Continuous keystroke dynamics: A different perspective towards biometric evaluation. Information Security Technical Report, 17(1):36–43, 2012.

5. Patrick Bours and Soumik Mondal. Performance evaluation of continuous authentication systems. IET Biometrics, 2015.

6. Christophe Couvreur. Hidden Markov models and their mixtures. DEA Report, Dep. of Mathematics, Université catholique de Louvain, 1996.

7. Søren Forchhammer and Tage S Rasmussen. Adaptive partially hidden markov models with application to bilevel image coding. Image Processing, IEEE Transactions on, 8(11):1516–1526, 1999.

8. Søren Otto Forchhammer and Jorma Rissanen. Partially hidden markov models. IEEE Transactions on Information Theory, 42(4):1253–1256, 1996.

9. T Ijobayashi, Junko Furumaya, and Ken Mas. Partly hidden markov model and its application to speech recognition. In Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on, volume 1, pages 121–124. IEEE, 1999.

10. L Jain, JV Monaco, MJ Coakley, and CC Tappert. Passcode keystroke biometric performance on smartphone touchscreens is superior to that on hardware keyboards. International Journal of Research in Computer Applications & Information Technology, 2(4):29–33, 2014.

11. Dan Jurafsky, James H Martin, Andrew Kehler, Keith Vander Linden, and Nigel Ward. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition, volume 2. MIT Press, 2000.

12. Tetsunori Kobayashi and Satoshi Haruyama. Partly-hidden markov model and its application to gesture recognition. In Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on, volume 4, pages 3081–3084. IEEE, 1997.

13. Yi Lu and Leilei Zeng. A nonhomogeneous poisson hidden markov model for claim counts. Astin Bulletin, 42(01):181–202, 2012.

14. R Dean Malmgren, Daniel B Stouffer, Adilson E Motter, and Luís AN Amaral. A poissonian explanation for heavy tails in e-mail communica- tion. Proceedings of the National Academy of Sciences, 105(47):18153–18158, 2008.

15. Bernard Merialdo. Tagging english text with a probabilistic model. Computational linguistics, 20(2):155–171, 1994.

16. John V Monaco, Ned Bakelman, Sung-Hyuk Cha, and Charles C Tappert. Recent advances in the development of a long-text-input keystroke biometric authentication system for arbitrary text input. In European Intelligence and Security Informatics Conference (EISIC), pages 60–66. IEEE, 2013.

17. Allen Newell. Unified theories of cognition. Harvard University Press, 1994.

18. Huseyin Ozkan, Arda Akman, and Suleyman S Kozat. A novel and robust parameter training approach for hmms under noisy and partial access to states. Signal Processing, 94:490–497, 2014.

19. Lawrence Rubiner. A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2):257–286, 1989.

20. Vasanthan Raghavan, Aram Galstyan, Alexander G Tartakovsky, et al. Hidden markov models for the activity profile of terrorist groups. The Annals of Applied Statistics, 7(4):2402–2430, 2013.

21. Ricardo N Rodrigues, Glauco FG Yared, Carlos R do N Costa, João BT Yabu-Uti, Fábio Violaro, and Lee Luan Ling. Biometric access control through numerical keyboards based on keystroke dynamics. In Advances in Biometrics, pages 640–646. Springer, 2005.

22. Timothy A Salthouse. Perceptual, cognitive, and motoric aspects of transcription typing. Psychological bulletin, 90(3):303, 1986.

23. DB Stouffer, RD Malmgren, and LAN Amaral. Comments on "the origin of bursts and heavy tails in human dynamics". arXiv preprint physics/0510216, 2005.

24. RL Stratonovich. Conditional markov processes. Theory of Probability & Its Applications, 5(2):156–178, 1960.

25. Charles C Tappert, Sung-Hyuk Cha, and Robert S Zack. A keystroke biometric system for long-text input. International Journal of Information Security and Privacy (IJISP), 4(1):32–60, 2010.

26. Mary Villani, Charles Tappert, Gian N, Justin Simone, H St Fort, and Sung-Hyuk Cha. Keystroke biometric recognition studies on long-text input under ideal and application-oriented conditions. In Computer Vision and Pattern Recognition Workshop, 2006. CVPRW’06. Conference on, pages 39–39. IEEE, 2006.

27. Martin Wiener, Matthew S Matell, and H Branch Coslett. Multiple mechanisms for temporal processing. Frontiers in integrative neuroscience, 5, 2011.

28. Howard N Zelaznik, Rebecca Spencer, and Richard B Ivry. Dissociation of explicit and implicit timing in repetitive tapping and drawing movements. Journal of Experimental Psychology: Human Perception and Performance, 28(3):575, 2002.