Engineering Applications of Artificial Intelligence ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Predictive models of human supervisory control behavioral patterns using hidden semi-Markov models

Yves Boussemart [a,*], Mary L. Cummings [b]

[a] Engineering Systems Division, Massachusetts Institute of Technology, 77 Massachusetts Avenue 33-407, Cambridge, MA 02139, United States
[b] Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 77 Massachusetts Avenue 33-311, Cambridge, MA 02139, United States

ARTICLE INFO

Article history:
Received 3 September 2010
Received in revised form
31 March 2011
Accepted 21 April 2011

Keywords:
Hidden semi-Markov models
Human supervisory control
Unmanned vehicles
Operator model
Pattern recognition
Human behavioral patterns

ABSTRACT

Behavioral models of human operators engaged in complex, time-critical high-risk domains, such as those typical in Human Supervisory Control (HSC) settings, are of great value because of the high cost of operator failure. We propose that Hidden Semi-Markov Models (HSMMs) can be employed to model behaviors of operators in HSC settings where there is some intermittent human interaction with a system via a set of external controls. While regular Hidden Markov Models (HMMs) can be used to model operator behavior, HSMMs are particularly suited to time-critical supervisory control domains due to their explicit representation of state duration. Using HSMMs, we demonstrate in an unmanned vehicle supervisory control environment that such models can accurately predict future operator behavior both in terms of states and durations.

1. Introduction

Formally, Human Supervisory Control, or HSC, is the process by which one or more human operators intermittently interact with a computer, receiving feedback from and providing commands to a controlled process or task environment, which is connected to that computer (Sheridan, 1992). Because the computer allows operators and tasks to be decoupled both in time and space, operators in HSC settings often work under time-pressure and in high risk environments. Furthermore, this work is primarily cognitive and procedural, i.e., other than the occasional button press or lever engagement, most work happens via internal information processing that follows a set of pre-defined steps. Typical HSC domains include military command and control, air traffic control, railway systems and process control including the operation of nuclear power plants. Because HSC systems are often mission and/or life-critical, operator failure could lead to disastrous outcomes (Leveson, 1986).

Unmanned Vehicles Systems (UVSs) form a representative application of time-pressured and mission-critical human supervisory control. Within this domain, a significant amount of resources and research has been devoted to leveraging automation in order to shift the current operating paradigm in which multiple operators control a single unmanned vehicle to one in which a single operator could control multiple vehicles (Dixon and Wickens, 2003; Mitchell and Cummings, 2005). This radical change in paradigm represents a challenge both for operators and systems designers (Ollero and Maza, 2007). The control of multiple UVSs compounds the potential for operator cognitive overload while simultaneously increasing the potential consequences of operator failure. Thus, constant monitoring of the operator behavior is critical for proper system behavior. That task is usually assigned to supervisors who typically rely on expert knowledge and experience to detect anomalous conditions. Because continuously monitoring the behavior of multiple operators while providing high-level guidance is a demanding task for the supervisor, the ability to automatically recognize the likely onset of an operator's off-nominal behavior, as defined by a deviation from an expected behavioral pattern determined by procedures, has immense practical value as potential serious accidents could be avoided.

By leveraging recent advances in processing power and machine learning algorithms, models of operator behaviors can be learned from data and are thus capable of monitoring operators controlling one or more UVSs. Thus, automation can support the performance monitoring task by generating an alert to a supervisor if the deviation from the expected behavior of the operators under his or her supervision exceeds a given threshold. This paper describes a stochastic modeling technique developed to detect and predict such deviations from expected behavior, validated through human-in-the-loop experimental data. In contrast with previously published research, we show that the proposed modeling approach can accurately predict not only future types of operator behaviors, but also their length of

* Corresponding author. Tel.: +1 617 942 1684; fax: +1 617 253 4196.
E-mail address: yves@mit.edu (Y. Boussemart).

0952-1976/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.engappai.2011.04.008

duration, as an important characteristic in time-critical HSC contexts.

2. Background

In general, given a training set of known behavioral patterns, there are two alternatives to detect anomalous behaviors: (1) show that the observed pattern is similar to a known adversary pattern or (2) show that the observed pattern is dissimilar to a known normal pattern (Singh et al., 1996). The first option is impractical because it is, in general, difficult to generate an exhaustive list of adversary patterns in applications characterized by a large number of degrees of freedom such as HSC settings. In contrast, because HSC environments are procedure-based, predictive models of human behavior embody the known normal patterns and can therefore be used to detect and predict anomalous behaviors. In addition to providing anomaly detection capability, most predictive models also comprise a descriptive component. Thus, within the context of HSC behaviors, the real-time use of predictive models can support the performance monitoring task of a team supervisor by generating alerts when anomalous situations are predicted. These same models can also be analyzed off-line and provide a better understanding in the typical behavior of operators.

There exists a wide range of computational modeling techniques, which can be divided into three main categories: symbolic models, architecture-based models and statistical models. The first two classes of model tend to be deductive (i.e., use of a top-down methodology relying on predefined theories) whereas statistical models tend to be inductive (i.e., a bottom-up approach and data driven).

Symbolic modeling techniques represent different mental objects using variables and rules. The most commonly used decision support tools relying on such methods are expert systems (Endsley, 1987). This methodology, however, suffers from its strict reliance on rules that must be correctly elicited from a Subject Matter Expert (SME) *a priori*. This problem of knowledge elicitation is both time consuming and may introduce the bias of a given SME in the system. In contrast, architecture-based models make use of theoretical frameworks aimed at replicating cognitive processes, and therefore serve as blueprints for intelligent agents. GOMS (Wayne et al., 1992) and ACT-R (Anderson, 1993), are two such cognitive frameworks. The practical use of ACT-R is limited because of sophisticated cognitive task modeling required to fit the framework. GOMS is similarly limited because it assumes that all users behave deterministically and follow the same human processor model, which narrowly limits use to expert behaviors. Thus, the main shortcoming of both symbolic and architecture-based methods lies in their use of *a priori* definition of rules or cognitive processes. Eliciting such rules or cognitive processes is problematic in HSC settings because of the complexity of the decisions, as well as uncertainty in the environment. Moreover, such approaches are inherently brittle as they cannot be used to describe or predict anomalous, never-before-seen events. In contrast, statistical models make use of an inductive, data-driven approach in the sense that they rely on the exploitation of the statistical patterns exhibited in the human behavior data stream in order to describe and predict possible future actions.

The alternative approach to symbolic and architecture-based models is applying statistical learning techniques to human behaviors, relying on the idea that human actions can be appropriately modeled by serial processes because humans can solve only one complex problem at a time (Welford, 1952; Broadbent, 1958). Therefore, pattern recognition techniques have been used to model human behaviors ranging from large-scale populations patterns (Pentland, 2008) to detailed small-scale cognitive processes (Griffiths et al., 2008). Such applications have included computer system intrusion detection (Terran, 1999), ship navigation (Gardenier, 1981) or car driving (Pentland and Liu, 1999). Yet, little work employs such techniques in the HSC context, even though the correctness of operator behavior in such settings is often mission and life-critical. Statistical learning techniques can be beneficial in the HSC domain because, in contrast with qualitative models, they provide a formal, quantitative basis for describing human behavior patterns and for predicting future actions. This is especially true for HSC application because the procedures provide structure to the behavior of an operator thereby facilitating the emergence of behavioral patterns that can be exploited by the statistical models.

One popular statistical learning tool, hidden Markov models, has been used to investigate human cognition through behavioral patterns in rule-based continuous control tasks such as driving (Pentland and Liu, 1999) or space shuttle landing (Hayashi et al., 2005). More recently, HMMs have also been shown capable of modeling and predicting knowledge-based tasks such as supervisory control behaviors of Unmanned Vehicle (UVs) operators (Boussemart and Cummings, 2008). The structure of the HMM is particularly suitable for inferring underlying, hidden cognitive processes from the patterns of visible events extracted from human behavior, especially in unsupervised training contexts (Boussemart et al., 2009). These behavioral patterns correspond to statistically linked clusters of observable events, which we call operator states. For example, the set of actions required to change the path of a vehicle – such as vehicle selection and waypoint manipulation – tend to occur concomitantly and could be clustered in an operator state linked with replanning. The learned model thus synthesizes operator behavioral patterns through the definition of a number of such operator states along with the probabilities of going from one operator state to another.

Classical HMMs, however, have a structural shortcoming in that they cannot explicitly take the timing of state transitions into account. In HSC settings, timing is often a critical consideration, since correct actions taken even 1 s late can cause problems, e.g., and air traffic controller waving off an aircraft 1 s too late can be disastrous. Thus, the lack of timing information can lead to flawed, less precise models. It is precisely this shortcoming that this paper explores through the use of hidden semi-Markov models, a version of HMMs capable of explicitly modeling the timing of state transitions for HSC behaviors. The next section outlines the computational aspects of HMMs and their structural shortcoming is addressed by presenting the HSMMs methodology.

3. Methodology

3.1. Hidden Markov models

Hidden Markov models were popularized by a seminal paper by Rabiner and Juang (1986). They consist of stochastic Markov chains based around a set of hidden states that cannot be directly observed. Each hidden state generates an observable symbol according to a specific emission function. Although the sequence of hidden states cannot be observed directly, the probability of being in a specific state can be inferred from the sequence of observed symbols. Transition functions describe the dynamics of the hidden state space. There are two types of probability parameters in HMMs: state transition probabilities and observable symbol output probabilities. Given a finite sequence of

Please cite this article as: Boussemart, Y., Cummings, M.L., Predictive models of human supervisory control behavioral patterns using hidden semi-Markov models. Engineering Applications of Artificial Intelligence (2011), doi:10.1016/j.engappai.2011.04.008

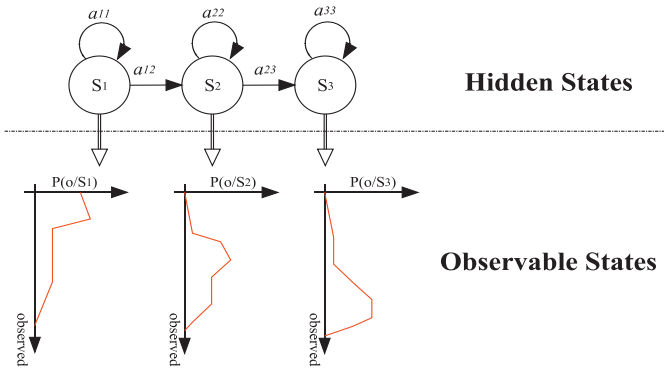

Fig. 1. A 3-state hidden Markov model.

hidden states, all the possible transition probabilities and symbol output probabilities can be multiplied at each transition to calculate the overall likelihood of all the output symbols produced in the transition path up to that point. Summing all such transition paths, one can then compute the likelihood that the sequence was generated by the HMM.

Adopting the classic notation from Rabiner and Juang (1986), let N be the number of states $S=\{S_1,S_2,\ldots,S_N\}$ in the HMM and M be the number of observation symbols $V=\{V_1,V_2,\ldots,V_M\}$ (i.e. the dictionary size). Let S_i^t denote the property of being in state i at time t. The state transition probability from state i to state j is $A=\{a_{ij}\}$ where $a_{ij}=P\left(S_j^{t+1},S_i^t\right); i,j=1,\ldots,N$. The symbol output probability function in state i is $B=\{b_i(c)\}$, where $b_i(c)=P(V_c|S_i)$. The initial probability of being in state i at time $t=0$ is $\pi=\{\pi_i\}$, where $\pi_i=P(S_i^0)$.

The model parameters must be valid probabilities and thus satisfy the constraints

$$\sum_j^N a_{ij}=1,\sum_c^M b_j(c)=1,\sum_j^N \pi_j=1,\forall j$$

$$a_{ij}\geq 0,b_j(c)\geq 0,\pi_j\geq 0 \quad (1)$$

Thus, an HMM is formally defined as the tuple, $H=\{S,V,A,B,\pi\}$. Fig. 1 illustrates the HMM concept by showing a graphical representation of a 3-state model, where the set of hidden states' $\{S_1,S_2,S_3\}$ transition probabilities are defined as a set of a_{ij}'s. Each state has a probability density function of emitting a specific observation.

An HMM respects the first order Markov assumption if the transition from the current state to the next state only depends on the current state.

3.1.1. HMM algorithms

HMM algorithms embed three functions: model evaluation, most likely state path and model learning. The first function is evaluation, i.e. the probability that a given sequence is produced by the model. This probability of a given sequence of data given the model is useful because, according to the Bayes' rule, it is a proxy for the probability of the model given the data presented. We can thus compare different models and choose the most likely one by solving the evaluation problem with the forward/backward dynamic programming algorithm. Finally, the HMM parameters must be learned given a fixed data set, and the most commonly used algorithm is a form of Expectation–Maximization (EM) called the Baum–Welch algorithm. The goal of the Baum–Welch algorithm is to maximize the posterior likelihood of the observed sequence O^s for a given HMM. More formally,

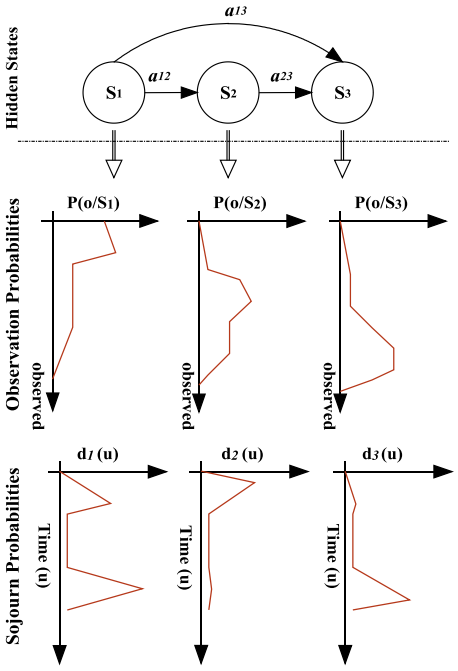

Fig. 2. A 3-state hidden semi-Markov model.

Baum–Welch computes the optimal model H^* such that

$$H^*=argmax_H\left(\prod_s P(O^s;H)\right) \quad (2)$$

Expectation maximization hypothesizes an initial, arbitrary set of model parameters. These model parameters are then used to estimate a possible state sequence $S_{O^s}=\{s^1,\ldots,s^{l_s}\}$ via the Viterbi algorithm. This is the expectation or E-step of the EM algorithm. The model parameters are then re-estimated given the state labels S_{O^s}. This constitutes the maximization step (or M-step) of the learning algorithm. Through this iterative EM procedure, it can be proven that the Baum–Welch algorithm converges to a local optimum (Baum and Petrie, 1966).

3.2. Hidden semi-Markov models

One of the shortcomings of classical HMMs is that they do not provide a way to explicitly deal with state durations. In classical HMMs, the probability of staying in a given state has to be geometrically distributed according to the state self-transition probability. Indeed, the probability of staying in state i for j iterations is $(a_{ii})^j$. Assuming a geometric state duration distribution may not be valid in all contexts, and could be problematic in HSC domains which often dictate that operators perform actions in time-pressured scenarios. Hidden Semi-Markov Models (HSMMs, also known as explicit duration hidden Markov models) address this specific issue (Rabiner, 1989; Guedon, 2003).

Structurally, a HSMM is similar to an HMM in that it is composed of an embedded Markov chain (usually first order) that represents the transitions between the hidden states $\{S_t\}$. In addition, an HSMM incorporates a discrete state occupancy distribution representing the time spent in non-absorbing states. The set of such distributions is noted $D=\{d_j(u)\}$ and represents the probability of staying u units of time in state j.

Fig. 2 shows a 3-state hidden semi-Markov model including the duration distributions $d_j(u)$ for all states. Formally, the

Please cite this article as: Boussemart, Y., Cummings, M.L., Predictive models of human supervisory control behavioral patterns using hidden semi-Markov models. Engineering Applications of Artificial Intelligence (2011), doi:10.1016/j.engappai.2011.04.008

duration distribution probability is defined as follows:

$$d_j(u) = P\left(S_{k\neq j}^{t+u+1}, S_j^{t+u-v}, v=0,\ldots,u-2 \mid S_j^{t+1}, S_{h\neq j}^{t}\right), \quad u=1,\ldots,M_j \tag{3}$$

Where M_j is an upper bound to the time spent in state j. In contrast to the HMM, there can be no transition of the form a_{ii} in an HSMM as demonstrated in Fig. 2. Furthermore, the conditional independence between the past and the future in HSMMs only holds when the process evolves from one state to another, while this property holds at each time step for HMMs. This distinction denotes the relaxation of the Markov assumption to a semi-Markov regime.

3.2.1. HSMM algorithms

Similarly to HMMs, the forward/backward algorithm is a central estimation mechanism for HSMMs. However, the addition of the duration probability makes the algorithm more complex than for HMMs. Guedon (2003) proposes a possible derivation of the forward/backward algorithm adapted to HSMMs.

The ability of HSMMs to extract information from timing data comes at the expense of model complexity since HSMMs typically need a significantly higher number of parameters than regular HMMs. This has implications both in terms of the amount of data needed to train models, as well as for model generalizability. As can be expected, learning HSMMs is significantly more difficult than learning HMMs. The first issue is generalizability (i.e. regularization) in that HSMMs contain significantly more parameters than HMMs with identical numbers of hidden states, and are therefore more prone to overfit the training data (Guedon, 2003). An n-state HMM with a d-sized dictionary has $n+n^2+dn$ number of parameters ($d \gg n$ usually). A similar HSMM with a maximum state duration M_t has $n+n^2+dn+nM_t$ parameters ($M_t \gg d \gg n$ usually). For practical models (i.e. with relatively small n), the dominant factors become the size of the dictionary d and the maximum state duration M_t.

In contrast with the size of the dictionary, the maximum state duration can be traded off against time resolution granularity because it is computed in terms of time-steps. Obtaining fine grained time-resolution (i.e. multiple time-steps per second) can become expensive if some states could have long durations. The higher number of parameters means that achieving a parsimonious and generalizable model is difficult and requires more training data, often a problem in small sample HSC settings. Additionally, from a purely computational perspective, learning the model can be impractical. Looking at the cost of a forward/backward pass, a n-state HMM with a d-sized dictionary will typically have a run-time of $O(n^2d)$. In contrast, the same run-time for an HSMM with a maximum state duration M_t will be $O(n^2dM_t^2)$ (Mitchell et al., 1999).

The solution to both of these problems, i.e. model generalizability and computational complexity, lies in reducing the number of parameters that need to be learned. As shown earlier, a significant proportion of the number of parameters in an HSMM is devoted to defining a set of fully non-parametric duration distributions. One way to reduce the number of parameters in the model is to use parameterized distributions, such as Gaussian mixture models, in order to describe the duration probabilities (Marin et al., 2005). This reduction promotes model generalizability and reduces the computation load at the cost of imposing additional constraints on the expression of the duration probability distribution function.

3.2.2. Duration distributions as Gaussian mixture models

Gaussian Mixture Models (GMMs) are defined as a weighted sum of independent normal distributions. The GMM definition of the state duration is as follows:

$$d_j(u) = \sum_{k=1}^{M_m} \phi_{jk} * \frac{e^{(-(u-\mu_{jk})^2)/2\sigma^2}}{\sqrt{2\pi\sigma_{jk}^2}} \tag{4}$$

Where M_m is the number of modes in the GMM and ϕ_{jk} represents the weighting parameter of the kth Gaussian in state j, which has a mean μ_{jk} and a standard deviation σ_{jk}. The GMMs parameters, i.e. ϕ_{jk},μ_{jk} and σ_{jk}, can be learned by a process of expectation maximization identical to the one used for the other parameters of the HSMM. For a GMM with a single mode, the solution can be computed by taking the partial derivative of the following function and setting it to 0 (Marin et al., 2005). For GMMs with more than one mode, the derivation of the re-estimation equations remains similar to the single mode case, with the additional requirement of computing the appropriate weight parameter ϕ_{jk}. The full derivation of the GMM re-estimation equations can be found in (Boussemart, 2011).

While requiring fewer parameters, parametric functions such as GMMs impose additional constraints on the possible distribution of the duration probability functions. In HSC settings, this assumption is reasonable because the time it takes for a human to perform a specific task can generally be accurately fit by specific distributions, such as Gaussian, Gamma or Beta (Law and Kelton, 2000). Using parameterized duration distributions may not be warranted in all settings however, so the number of parameters used should be calibrated to the quantity and quality of the available training data so as to minimize the risk of overfitting.

In summary, the structure of HMMs constrains the expression of the state duration to a geometric distribution, which may not reflect the time pressure inherent in many supervisory control settings. In contrast, HSMMs are capable of explicitly modeling state durations, which allow for more flexible models. The state duration distributions may be expressed non-parametrically at the cost of significantly increasing the number of parameters that need to be learned, which may impair the generalizability of the model. A solution is to use parameterized state duration distributions. In this work, we use mixtures of Gaussian models to address this issue. Thus, HSMMs are more complex than HMMs but are capable of providing information (i.e., state duration) that HMMs cannot. Because HSC contexts are often inherently time critical, state duration information is important because it provides a basis for estimating not only what the next most likely actions are, but also when they are most likely to take place. In the following section, we present experimental data used to compare HMMs to non-parametric HSMMs, as well as parametric HSMMs in an HSC setting.

4. Experimental data

As a representative case, we focus on Unmanned Vehicle (UV) systems where a single operator controls multiple heterogeneous unmanned vehicles. This context is characteristic of expected future military operations (DoD, 2007), and highlights the system-level importance of monitoring the behaviors of an operator in charge of a larger number of highly automated resources. This representation is generalizable to any supervisory control task where a human operator is supervising high levels of automation distributed across a number of complex tasks.

Data used to develop the HMM and HSMM of a UV human supervisory control system was obtained from a previous experiment (Nehme et al., 2008). While the goal of the original experiment was to validate a discrete event simulation model of an operator controlling multiple heterogeneous unmanned vehicles, the recorded user interface interactions represent a rich

Please cite this article as: Boussemart, Y., Cummings, M.L., Predictive models of human supervisory control behavioral patterns using hidden semi-Markov models. Engineering Applications of Artificial Intelligence (2011), doi:10.1016/j.engappai.2011.04.008

corpus of supervisory control behavior which can be used as a training set for comparing HMMs, non-parametric HSMMs and parametric HSMMs.

In the experiment, a single human operator controlled a team of UVs composed of unmanned air and underwater vehicles (UAVs and UUVs). The user interface is shown in Fig. 3.

In this interface, the UVs perform surveillance tasks with the ultimate goal of locating specific objects of interest in urban coastal and inland settings. UAVs can be of two types: one that provides high level sensor coverage (High Altitude Long Endurance or HALE), while the other provides more low-level target surveillance and video gathering (Medium Altitude Long Endurance or MALE). In contrast, all UUVs are all of the same type, with a similar goal of searching for targets of interest. Thus, the single operator controls a heterogeneous team of UVs which may consist of up to three different types of platforms.

In this simulation, the HALE performs a target designation task (simulating some off-board identification process). Once designated, operators use either the MALEs or UUVs to perform a visual target acquisition task, which consists of looking for a particular item in an image by panning and zooming the camera view. Once a target is visually identified, an automated planner chooses the next target assignment, creating possibly sub-optimal target assignments that the human operator can correct. Furthermore, threat areas appear dynamically on the map, and entering such an area could damage the UV, so the operator can optimize the path of the UVs by assigning a different goal to a UV or by adding waypoints to a UV path in order to avoid threat areas.

Participants maximized their score by (1) avoiding dynamic threat areas, (2) completing as many of the visual tasks correctly, (3) taking advantage of re-planning when possible to minimize vehicle travel times between targets and (4) ensuring a vehicle was always assigned to a target whenever possible. Training was done through an interactive tutorial and an open-ended practice session. Final scores corresponded to the total number of targets correctly identified. All data were recorded to an online database. The data of interest for this paper consisted of user interactions with the interface in the manner of clicks, such as operator UV selections on the map or on the left sidebar, waypoint operations (add, move, delete), goal changes and the start and end of visual tasks, as seen in Fig. 3.

4.1. Applying hidden Markov models to human supervisory controller behavior

Determining the user behavior parameters of HMMs and HSMMs requires training the model on the observed behavioral data. The raw behavioral data, which consists of the logged user interface events described above, cannot be used directly by the learning algorithms, and must be pre-processed. Fig. 4 shows the overall process, which consists of a grammatical and a statistical phase. The grammatical phase translates the low level observed user interactions into abstract events, which then form the basis of the observable state space for the statistical phase. In this phase, the model learning algorithms are used in order to obtain the model parameters.

The first step of the process consists of parsing low-level input information (such as mouse clicks on a screen) into abstract events according to a set of grammatical rules. The role of the grammar is thus to abstract low level user interface interactions into a set of meaningful tasks that can both be learned by the algorithm, as well as interpreted by a human modeler. Thus, the grammar represents feature extraction which reduces the size of the state space. It also defines the scope of the observable space usable by the machine learning process (Eads et al., 2005).

For application to HSC settings, we propose that the grammar should take the form of a 2D space where the rows defines a set of operands (i.e. entities that are acted on) while the columns delineates a set of operations (i.e. what is being performed). A set of operations can be established through a general task analysis (Kirwan and Ainsworth, 1992) or a more specialized Cognitive Task Analysis (CTA) (Schraagen et al., 2000). This orthogonal representation of the state space is essentially a generic ontology that represents type of objects and their relations. In the application, the user interactions were first categorized by operands, i.e. the type

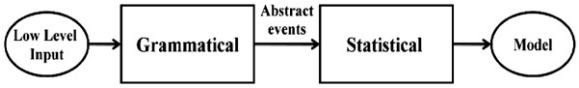

Fig. 4. Combined grammatical and statistical approach infers future behavior from a stream of current behavior.

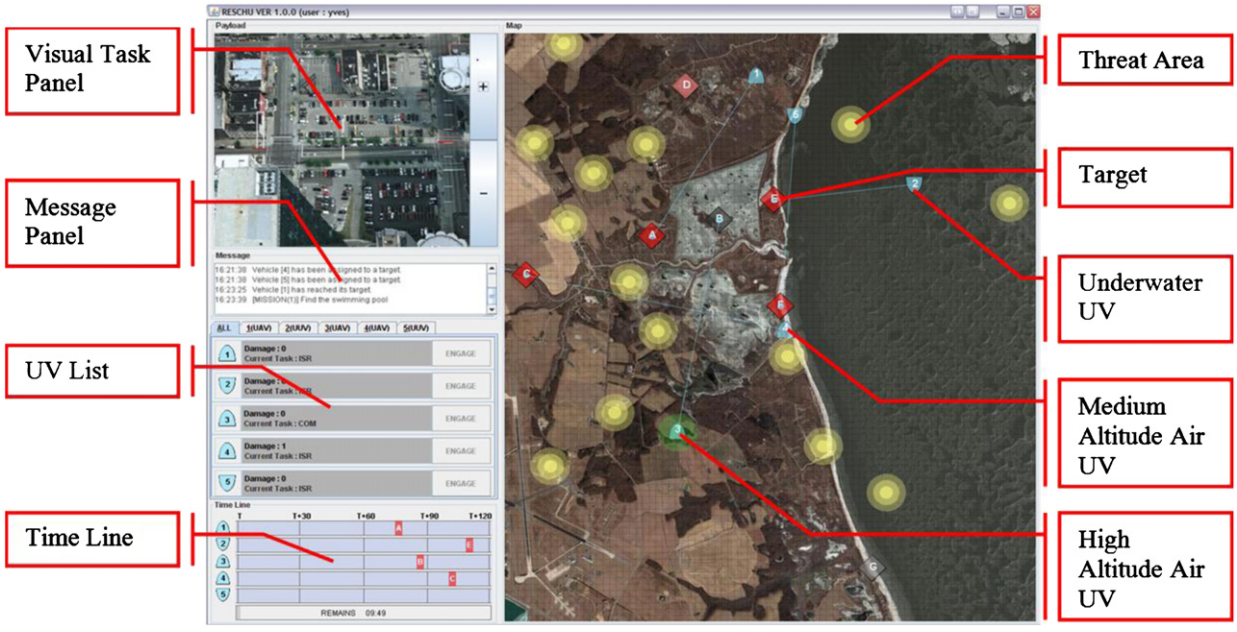

Fig. 3. The experimental user interface.

Please cite this article as: Boussemart, Y., Cummings, M.L., Predictive models of human supervisory control behavioral patterns using hidden semi-Markov models. Engineering Applications of Artificial Intelligence (2011), doi:10.1016/j.engappai.2011.04.008

Table 1
The HMM/HSMM grammar.

UV type/**mode**	**Select sidebar**	**Select map**	**Waypoint edit**	**Waypoint add/del**	**Goal**	**Visual task/engage**
All UVs	–	–	–	–	–	–
Underwater UV	–	–	–	–	–	–
MALE	–	–	–	–	–	–
HALE	–	–	–	–	–	–

of UV under control (All UVs, UUVs, MALEs or HALEs) and define the rows of Table 1. Then, the interactions with each of the UV types were separated into different operations in Table 1 while selection on either the sidebar or on the map, waypoint manipulation (addition, deletion and modification), goal changes and finally, the visual task engagement. These different operations define the columns in Table 1. The table of operands and operations represent all possible user interactions with the system.

The second step of the process is the statistical learning phase which attempted to model either the state (HMM) or the state and time-sequenced (HSMM) data. During this phase, the maximum likelihood estimates of the HMM and HSMM parameters are computed. In this experiment, we used 49 data traces of an individual user's 10 min long trials, resulting in a total of 3420 data points for the HMM model, and 54,368 data points for the HSMM models. Since no more than two events ever happened in the same second, we used a time resolution of 500 ms per time step in the HSMM algorithm.

The maximum duration for the longest task, i.e. the visual search, was 45 s, so$M_t=90$. We used a 3-fold cross-validation by training the algorithm on 46 subjects, keeping 3 subjects as a test set. The training was stopped when the likelihood of the cross-validated sequences started to decrease, which indicated that the model overfit the training data. Furthermore, because the EM algorithm is akin to a gradient search, it is susceptible to local optima (MacKay, 2004). The procedure outlined above must be repeated a large number of times with different initial values parameters of the HSMM in order to avoid such local optima, while keeping track of the most-likely model obtained in the process. To this end, we performed 10,000 iterations of this procedure for each of the model sizes (i.e. number of hidden states). In the next section, we present the results of the comparison between the HMMs, non-parametric HSMMs and parametric HSMMs.

5. Results and discussion

The HSMM model learning methodology described earlier was applied to the RESCHU data set, and both non-parametric and parametric models were learned. The parametric models used Gaussian mixture models with up to 3 modes[1] in order to express the state duration probability function. Similar to HMMs, several HSMMs need to be learned and the best one can be chosen through the process of model selection.

One common method to compare models of different sizes is the Bayesian Information Criterion (BIC) (Burnham and Anderson, 2002), which regularizes the model by balancing model fit and model complexity. The BIC allows the comparison of different models, in particular with different number of hidden states, trained on the same underlying data. More specifically, the BIC penalizes the likelihood $\mathcal{L}(H)$of the model H by a complexity factor proportional to the number of parameters P in the model and the number of training observations K. Thus, a lower BIC score is better.

$$BIC = -2\log(\mathcal{L}(H)) + P\log(K) \quad (5)$$

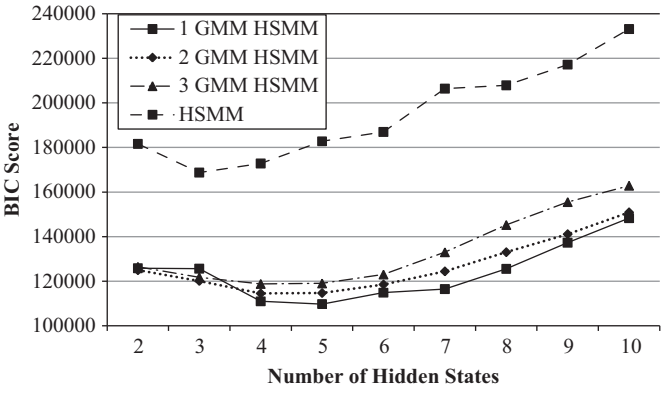

Fig. 5. BIC scores (lower is better) for the HSMMs and GMM–HSMM of different sizes.

The results in Fig. 5 show that the BIC scores of the non-parametric HSMMs (from 2 to 10 hidden states) are higher than that of any GMM–HSMM, regardless of the model size. The poorer scores of the non-parametric HSMM are due to the large number of parameters needed to specify every point in the distribution of the durations.

In contrast, the GMM–HSMMs, regardless of the number of modes used to specify their duration distribution, have fewer parameters and their BIC scores indicate that they are likely to generalize better than their non-parametric counterparts. Within the group of GMM–HSMMs, we see a similar trend where the simpler models tend to have a better BIC score. Thus for the RESCHU data, the BIC metric indicates that a 5-state HSMM with a 1-mode GMM used to define the duration distribution that provides the best HSMM model for this particular UV application, and that requiring full specification of all the parameters of the duration time distribution can be detrimental to model generalizability. However, using a parametric function to specify this distribution also imposes an additional assumption with regards to the form of the duration time expression. This may not be appropriate in applications that require highly specific time distributions, i.e., very tight tolerances for user interactions.

While the BIC of the 5-state 1-mode GMM–HSMM is the lowest of all HSMMs with a score of BIC=109,775 (Fig. 5), the best HMM model trained on the same data set, built around 8 hidden states (see Boussemart (2011) for more details), is an order of magnitude lower (BIC=13,420). Although the BICs cannot be compared directly due to the rescaling of the training data with the HSMM time resolution, the results suggest that the less complex HMMs are likely to generalize better to unseen data than HSMMs. HMMs, however, are not capable of using and providing timing information data, which are often critical in HSC settings. Thus, whether to use HMMs or HSMMs presents a trade space between external validity and model fit.

[1] A maximum of 3 modes for the GMM–HSMM was chosen because hidden states in HMMs and HSMMs typically represent less than 3 different modes of operation.

Please cite this article as: Boussemart, Y., Cummings, M.L., Predictive models of human supervisory control behavioral patterns using hidden semi-Markov models. Engineering Applications of Artificial Intelligence (2011), doi:10.1016/j.engappai.2011.04.008

5.1. Description of the 5-state 1-mode GMM–HSMM model

Fig. 6 presents an overview of the selected model, highlighting the different hidden states while graphically showing the state transition probability matrix. The 5 hidden states of the selected GMM–HSMM along with the state transition probabilities $A=\{a_{ij}\}$ are presented. All the transitions with less than 5% probability have been removed for legibility purposes; all states are otherwise fully connected. Note that because HSMMs explicitly model state durations, there are no self-transitions for the hidden states. The hidden states are also labeled according to their emission functions over the set of observable events. For example, Table 2 shows the distribution function $B=\{b_i(c)\}$ for the 2nd state. In particular, the distribution shows that state 2 correlates with planning behaviors for the MALEs (i.e. selecting the map, adding/deleting a waypoint or changing the goal of a MALE UV).

The transition probabilities between the hidden states, as highlighted in Fig. 6, can provide valuable insight into operator behavior. The model suggests that the planning and visual task states are heavily linked both for UUV and MALE types and expresses the idea that operators alternate regularly between these two activities. For both types of UVs, there is a high likelihood of engaging in planning behavior with a vehicle of the similar type after a given visual task (0.79 for the MALEs and 0.62 for the UUVs). This demonstrates that the first action an operator does after finishing a visual task is to send the vehicle towards another target, a typical replanning strategy for RESCHU. While the transition between the planning and visual tasks for the MALEs is strong (0.72), the transition between UUV planning and the visual task is comparatively weaker (0.33). This result is not surprising as UUVs are slower vehicles in RESCHU. Thus, an operator is less likely to perform a visual task right after retasking such a vehicle because the UUV will have to take longer to reach the assigned goal.

Fig. 7 shows the duration distribution functions $D=\{d_j(u)\}$ for the different states of the 5-state model in Fig. 6. The x-axis is labeled in 0.5 s intervals because this is the time resolution needed to parse out all the events in distinct discrete time steps. In other words, at most 2 events happened in the same second in the training data set, and therefore a time resolution of 0.5 s is needed to put them in different time intervals. The y-axis shows the probability of staying in a given state for a duration x. Fig. 7 demonstrates that the planning tasks (states 0, 2 and 4) require, on average, much less time to accomplish that the visual tasks of states 1 and 3, which agrees with observed data (as well as real world UAV operations). The mean duration of a planning state is 8.03 s whereas the mean duration of the visual task states is 25.43 s. These duration times thus present distinctly separate modes of operator behavior.

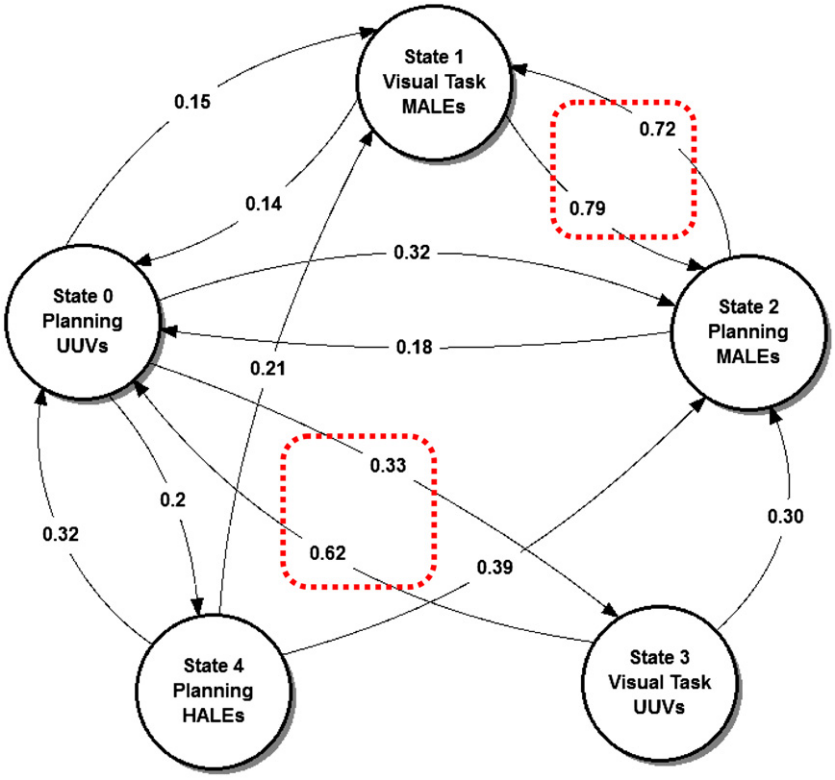

Fig. 6. Transition probabilities in the 5-state 1-mode GMM–HSMM (transitions with less than 0.1 weights have been removed for legibility purposes).

Table 2
State 2 observation probability function: interacting with MALEs.

All UVs	0.00	–	–	–	–	–
Underwater UV	0.00	0.00	0.00	0.00	0.00	0.00
MALE	0.00	0.30	0.00	0.28	0.41	0.01
HALE	0.00	0.00	0.00	0.00	0.00	0.00
UV type/**mode**	**Select sidebar**	**Select map**	**Waypoint edit**	**Waypoint add/del**	**Goal**	**Visual task/engage**

Please cite this article as: Boussemart, Y., Cummings, M.L., Predictive models of human supervisory control behavioral patterns using hidden semi-Markov models. Engineering Applications of Artificial Intelligence (2011), doi:10.1016/j.engappai.2011.04.008

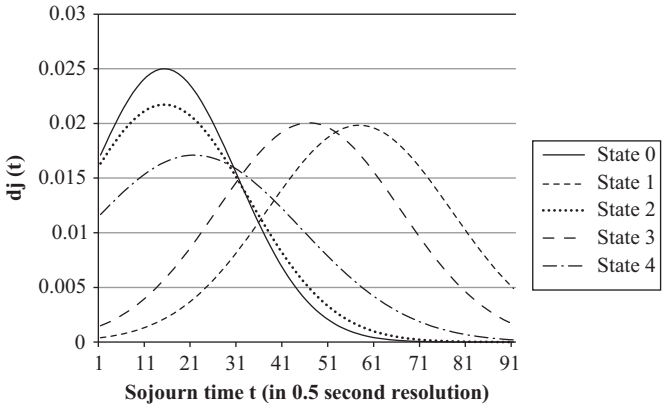

Fig. 7. Hidden state duration probabilities.

In addition, one of the most interesting features of the model is that 3 of 5 the hidden states in Fig. 6 represent operator planning and replanning operator behavior with each of the 3 types of UVs (HALEs, MALEs and UUVs). The last 2 hidden states represent the visual tasks for both MALEs and UUVs (recall HALEs do not perform a visual task). The expected durations of the visual task states for MALEs and UUVs are comparatively longer (28.5 and 22.5 s, respectively) than that of the interaction states (around 8 s). The fact that the learning algorithm was able to segregate the visual task states as different from the planning states highlights the insights that can be obtained from patterns contained in such a data set.

Overall, the qualitative interpretation of the model selected as the most likely is consistent with the RESCHU task and suggests that the learning algorithm was capable of extracting coherent and valuable information from the sequences of behavioral data used in model training. Given that this model can accurately describe operator behaviors, the next section will detail how such models can be used in a predictive manner.

5.2. Prediction metric

Additional value of using such models lies in their predictive abilities. Models capable of accurately predicting future operator behavior could be of great value in HSC settings which often can be life or mission critical. In order to measure model predictive capability, we introduce the Model Accuracy Score, a metric that weighs the quality and timing of the predictions according to a weighting parameter α.

5.2.1. MAS metric

Measuring the predictive capabilities of a regular HMM is a relatively simple process. The next n actions can be predicted from the model parameters and verifying if the predictions are correct is straightforward. However, measuring the predictive capabilities of HSMMs is more complex than for HMMs because the predictions are made on two independent dimensions: the first measures if the predicted event is correct (or at least of high probability), and the second dimension measures the timing of the prediction, i.e., did the prediction timing coincide with the occurrence of next event? The Model Accuracy Score (Huang, 2009), or MAS, is an aggregate metric that considers both dimensions, i.e. quality and timing of the predictions. The MAS assesses the predictions capability of a model according to the following:

$$MAS(t) = \frac{\sum_{i=t,\ldots,t-n} \alpha \times Quality(i) + (1-\alpha) \times Timing(i)}{n} \quad (6)$$

The MAS is a running average of n subscores, where the α parameter is the weighting factor used to balance the respective importance of quality and timing of the predictions. In the current application, we determined that $n=10$ provided a good balance between smoothing and sensitivity. The effects of the α parameter on the results will be discussed in the following section. The range of the MAS is [50 and 100], and each MAS sub-score is computed every time a user event is logged. The range of values of the MAS was chosen to promote a human operator's understanding by mimicking a prediction accuracy percentage where a score of 50 would mean no better than chance while a score of 100 would represent perfect predictions. For example, a MAS of 90 indicates that the model's prediction of the human's next action in controlling the unmanned vehicles is well within the set of expected actions (both in actual state transition and in timing of action). Conversely, a MAS of 50 predicts that the next action is outside the expected set of states or required time window for action. However, it is unlikely that a single MAS prediction is useful, as it is a running average and decision-makers will likely require further context and a temporal representation of the MAS to make an informed decision.

The MAS comprises two sub-scores that represent quality and timing. The quality of the prediction is computed by determining if the current event is within the top 5[2] predicted events at the previous iteration and scaled according to the ranking of the prediction. For example, if the current event was the top ranked in the predictions, the maximum score of 50 is assigned. In contrast, if the event is the 5th in the ranking, a score of 10 is assigned and any event out of the top five is given a score of 0. Thus, the quality sub-score of the prediction is exactly equivalent to the prediction performances metric used to evaluate classical HMMs. The timing of the prediction is evaluated by measuring the difference between the predicted and the actual state duration. Specifically, that difference is measured in terms of number of standard deviations away from the predicted mean state duration, both of which can be computed from the $D=\{d_j(u)\}$ distributions. The timing score is not penalized if the event happens within one standard deviation before or after the prediction. The 1 standard deviation standard was chosen because given the means and standard deviations of the durations of the planning and visual task operator states (states 0, 2, 4 and states 1 and 3, respectively), the chances of type I and II errors were 8% and 9%, respectively, for the most distant states (states 0 and 1), low by human modeling standards. Any timing deviation further than 1 standard deviation is penalized according to the Gaussian cumulative tail probability. For example, if an event arrives within 1 standard deviation of the predicted, the assigned score for the timing of the prediction is 50. In contrast, should a state duration be between 1 and 2 standard deviations away from the predicted, the timing score received will be 27.2, which is computed based on the area under the Gaussian curve between 1 and 2 standard deviations. Deviations larger than 3 standards deviations from the predicted mean receive a 0 score for the timing metric. This process is summarized in Fig. 8.

5.2.2. MAS sensitivity

Fig. 9 explores the sensitivity of the MAS to the weighting of the quality and timing of the predictions sub-scores (in particular the 1 standard deviation rule for the timing sub-score), both of which are essentially subjective components. Specifically, Fig. 9 shows the MAS obtained for the 5-state 1-mode GMM HSMM given different values of α (0.0, 0.5 and 1.0) and different time

[2] Following Huang's (2009) work, the top 5 events are considered in the metric in order to balance the penalty incurred for inaccurate predictions.

Please cite this article as: Boussemart, Y., Cummings, M.L., Predictive models of human supervisory control behavioral patterns using hidden semi-Markov models. Engineering Applications of Artificial Intelligence (2011), doi:10.1016/j.engappai.2011.04.008

resolutions ranging from 0.003 to 1 standard deviation. With an α value of 1.0, the MAS only considers how well the model is able to predict the next events with no consideration of timing. In this case, the value of the MAS is unaffected by the change in time resolution as shown by the constant MAS of 78.31. In contrast, with an α value of 0.0, the MAS only measures how well the states durations are predicted, and is more sensitive to the changes in time resolution. Finally, an α value of 0.5 weighs both quality and timing of the prediction equally.

As expected, the MAS scores decrease monotonically with finer-grained time resolution due to the increased penalty for falling outside of the full-score interval. For all values of $\alpha < 1.0$, the maximum MAS is obtained when the time resolution is 1 standard deviation (97.31 for $\alpha=0.0$ and 88.10 for $\alpha=0.5$). The MAS values then decrease and plateau for time resolutions finer than 0.03 standard deviations with MAS ranging from 55 to 60 for $\alpha=0.0$ and from 69 to 67 for $\alpha=0.5$. The MAS curves obtained for different values of α intersect the ordinate at the constant MAS score obtained for $\alpha=1.0$ (78.31) and the abscissa at a time resolution of 0.1 standard deviations. This intersection marks the time resolution setting at which the timing part of the metric stops contributing to the MAS. Finer-grained resolutions lead to a decreased MAS due to more stringent penalties for inaccurate predictions. In other words, at a time resolution smaller than 0.1 standard deviation, the timing of the prediction becomes inaccurate compared to the quality of the prediction and the overall MAS score is decreased.

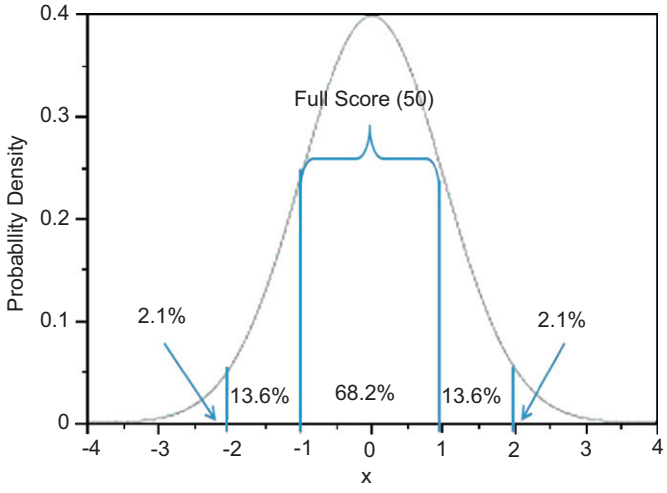

Fig. 8. Timing score scaling with a resolution of 1 standard deviation (Huang, 2009).S sensitivity.

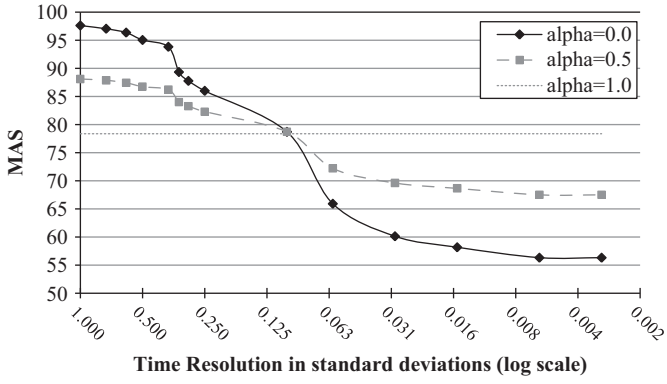

Fig. 9. MAS for the 5-state 1-mode HSMM given different time resolutions and α values.

A similar analysis can be carried out for models of different sizes. Fig. 10 (a) and (b) shows the same results as Fig. 9 but were obtained given a 4-state 1-mode GMM HSMM and a 6-state 1-mode GMM HSMM, the 2 closest models to their 5-state counterpart that exhibited the highest BIC score. Fig. 10 (a) shows that the 4-state 1-mode GMM HSMM can provide accurate timing predictions for time resolutions ranging from 1 to 0.25 standard deviations. Similarly, the 6-state 1-mode GMM HSMM provides accurate timing predictions for resolution up to 0.33 standard deviations (Fig. 10 (b)).

For reference, Fig. 11 compares the MAS scores of regular HMMs, 1-mode GMM HSMMs and non-parametric HSMMs trained on the same data set when $\alpha=1.0$. The comparison is only valid for this specific value of α, because HMMs cannot provide timing information and therefore the HMM MAS can only consider the quality of the prediction.

Fig. 11 shows that the MASs of the HSMMs are generally lower than that of the HMMs, which means that the additional number of parameters that need to be learned to specify $D=\{d_j(u)\}$ hinders the HSMM ability to accurately predict state durations. If looking at the highest MAS in Fig. 11, the simpler HMM models provide marginally higher MAS (MAS=83.0 for an 8-state HMM vs. MAS=79.5 for the 5-state 1-mode GMM–HSMM) at the expense of not providing timing information. In addition, the best HMM produces a more detailed 8-state model than the 5-state HSMM. Thus, comparing the predictive capability of the HMM and the selected GMM–HSMM provides insight into the practical consequences of using a larger number of parameters to define the model.

The MAS scores of the HMM were higher than any of the GMM–HSMMs. This performance delta highlights the trade-off between simple models which focus solely on quality of the prediction and more complex HSMM models which incorporate timing information. Our results suggest that simpler HMMs may be preferable in non-time critical applications. HMMs are simpler and can perform better than HSMMs at predicting next states, while

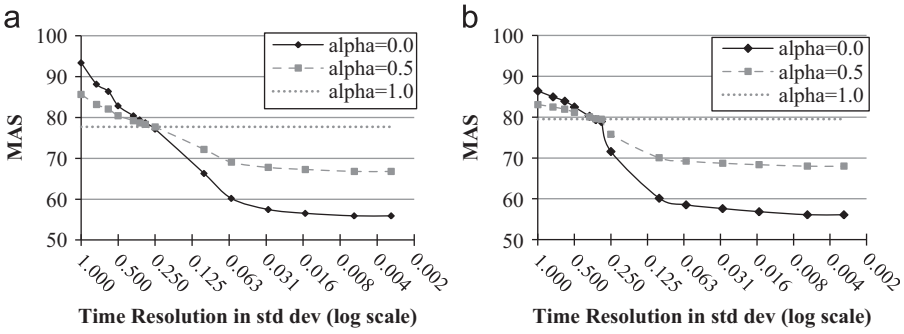

Fig. 10. MAS for the 4- and 6-state 1-mode HSMM given different time resolutions and α values.

Please cite this article as: Boussemart, Y., Cummings, M.L., Predictive models of human supervisory control behavioral patterns using hidden semi-Markov models. Engineering Applications of Artificial Intelligence (2011), doi:10.1016/j.engappai.2011.04.008

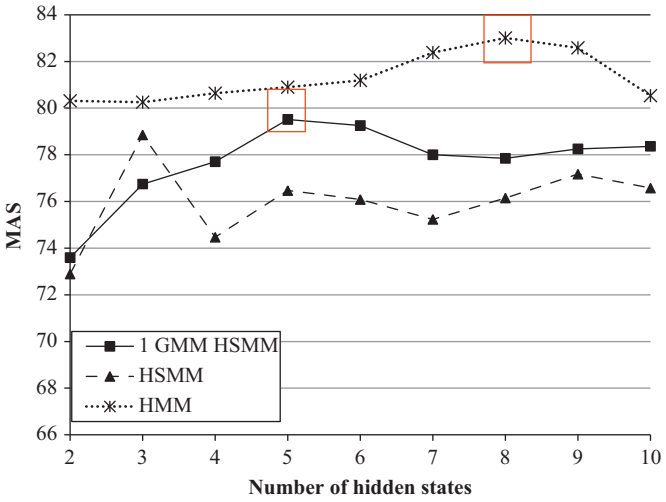

Fig. 11. MAS with $\alpha=1.0$ for 1-mode GMM HSMMs, HSMMs and HMM or different sizes.

being more computationally manageable and better capable of generalizing from a smaller training data set. However, HSMMs are capable of providing valuable information for time-sensitive applications that HMMs cannot, and for our UV application, the impact of the increased complexity on state predictions was relatively minor.

6. Conclusion

This paper presented a methodology capable of learning a set of hidden semi-Markov models in human supervisory control settings, which led to the selection of the optimal model using information theoretic measures. The selected model not only captured the underlying distribution of events in the training data, but also segregated qualitatively different behaviors (such as a differentiating a visual task from a planning action). We propose that such models can also be used to predict behaviors using the Model Accuracy Score, which is a flexible aggregate metric that weighs both quality and timing of the predictions. The advantage of the HSMM over the HMM approach lies in the HSMM's capability of explicitly modeling state durations. This, however, comes at a cost as HSMMs are significantly more complex than HMMs. HSMMs are thus more challenging from a computational standpoint, while also less likely to generalize well to unseen data. Thus, if the explicit expression of the state duration is not required, using regular HMMs should be favored. If the state duration is important, however as in the case of many HSC applications, HSMMs can provide a powerful framework to capture such information.

It should be noted that one limitation of the HMM/HSMM approach is the lack of generalizability beyond supervisory control domains with human–computer interaction in the form of control manipulation. This means that such models are inherently limited in settings where operators interact with a system primarily through visual monitoring, such as a nuclear power plant operator observing a system for long periods of time without activating any type of control device. Another limitation is that the models are data-driven and thus, inherently descriptive. Such models cannot prescribe what the optimal behavior should be without additional *a priori* performance knowledge. In order to ensure proper detection and prediction, the parameters of the HMMs and HSMMs must be learned from training data. However, the models we present can prescribe what the "typical" response of a person should be, given prior data about such typical behaviors. Such models are useful in highly proceduralized environments, commonly found in supervisory control settings. Finally, an additional limitation of the current results is the use of offline learning, in that the model parameters are learnt once based on a given data set. In contrast, the model parameters could conceivably be constantly updated as the system's exposure to real situations is increased. Such online learning algorithms represent a great avenue for future research.

While we have shown that HSMMs can predict both quality and timing of state transitions, the larger question remains as to how to leverage such predictions in an applied setting. As previously mentioned, the ability of a model to predict a single next observable event is not intrinsically useful in isolation. Because human behavior is rarely deterministic, a single low MAS score does not necessarily signal an anomalous situation. Instead, a consistent sequence of low MAS scores means that operator behavior either is different from what the model expects or within a range of the solution space not properly captured by the model. For example, the training data may not contain scenarios where an operator must deal with multiple simultaneous system failures. Should such an event happen in actual operations, the resulting MAS scores would likely be low, even if the behavior of the operator is perfectly appropriate in such an emergency situations.

This discrepancy raises the question of the choice of the appropriate threshold separating a marginal from a problematic MAS score. Different operational conditions and even different operators may have different thresholds, and there is no formal analytical way as of yet to determine what that threshold should be. Ultimately, it is our assertion that a human supervisor should use the output of such models to decide if the different behavior is qualitatively good or bad, and subsequently take the necessary remedial actions. A related research effort that investigated how HSMMs can be translated into real-time decision support displays for supervisors of HSC systems demonstrated that such models can be useful for anomalous behavior detection, but that the complexity of the models may ultimately limit their use in real-time settings (Castonia, 2010).

Acknowledgements

This research was sponsored by the Boeing Research and Technology and the Office of Naval Research.

References

Anderson, J.R., 1993. Rules of the Mind. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Baum, L.W.,, Petrie, T., 1966. Statistical Inference for Probabilistic Functions of Finite State Markov Chains. The Annals of Mathematical Statistics 37 (6), 1554–1563.

Boussemart, Y., 2011. Predictive models of procedural human supervisory control behaviors. Engineering Systems Division. Cambridge, Massachusetts Institute of Technology. PhD: 150.

Boussemart, Y., M.L. Cummings, 2008. Behavioral recognition and prediction of an operator supervising multiple heterogeneous unmanned vehicles. Humans operating unmanned systems, HUMOUS'08, Brest, France.

Boussemart, Y., J. Las Fargeas, M.L. Cummings, N. Roy, 2009. Comparing learning techniques for hidden Markov Models of human supervisory control behavior. AIAA Infotech@Aerospace'09 Conference, Seattle, Washington.

Broadbent, D.E., 1958. Perception and Communication. Pergamon, Oxford.

Burnham, K.P., Anderson, D.R., 2002. Model Selection and Multimodel Inference, a Practical Information Theoretic Approach. Springer, New York.

Castonia, R.W. (2010). The Design of a HSMM-based Operator State Modeling Display. AIAA Infotech@Aerospace 2010, Atlanta, GA.

Dixon, S.R., C.D. Wickens (2003). Control of Multiple-UAVs: A Workload Analysis. 12th International Symposium on Aviation Psychology, Dayton, OH.

DoD, 2007. Unmanned Systems Roadmap (2007–2032). Washington DC.

Please cite this article as: Boussemart, Y., Cummings, M.L., Predictive models of human supervisory control behavioral patterns using hidden semi-Markov models. Engineering Applications of Artificial Intelligence (2011), doi:10.1016/j.engappai.2011.04.008

Eads, D., K. Glocer, S. Perkins, J. Theiler, 2005. Grammar-guided feature extraction for time series classification. Neural Information Processing Systems (NIPS '05). Vancouver, BC.

Endsley, M., 1987. The application of human factors to the development of expert systems for advanced cockpits. Human Factors Society 31st Annual Meeting, Santa Monica, CA.

Gardenier, J.S., 1981. Ship navigational failure detection and diagnosis. In: Rasmussen, J., Roude, W.B. (Eds.), Human Detection and Diagnosis of System Failure. Plenum, Boston, pp. 49–74.

Griffiths, T.L., Kemp, C., Tenenbaum, J.B., 2008. Bayesian models of cognition. Cambridge Handbook of Computational Cognitive Modeling. Cambridge University Press, R. Sun.

Guedon, Y., 2003. Estimating Hidden Semi-Markov Chains From Discrete Sequences. Journal of Computational & Graphical Statistics 12 (3), 604–639.

Hayashi, M., B.Beutter, McCann, R.S., 2005. Hidden Markov Model analysis for space shuttle crewmember's scanning behavior. IEEE International Conference on Systems, Man and Cybernetics. Waikoloa, Hawaii, 1615–1622.

Huang, H., 2009. Developing an Abstraction Layer for the Visualization of HSMM-Based Predictive Decision Support Electrical Engineering and Computer Science, 118. Cambridge Massachusetts Institute of Technology, Masters.

Kirwan, B., Ainsworth, L.K., 1992. A Guide to Task Analysis: The Task Analysis Working Group. Taylor & Francis, Bristol, PA.

Law, A.M., Kelton, D., 2000. Simulation Modeling and Analysis. McGraw-Hill.

Leveson, N.G., 1986. Software Safety: Why, What, and How Computing Surveys 18 (2), 125–163.

MacKay, D.J.C., 2004. Information Theory, Inference, and Learning Algorithms. Cambridge University Press, Cambridge, UK; New York.

Marin, J.-M., Mengersen, K., Robert, C.P., Dey, D.K., Rao, C.R., 2005. Bayesian Modelling and Inference on Mixtures of Distributions. Handbook of Statistics, 25. Elsevier, North-Holland, Amsterdam 459–507.

Marin, M., Mengerson, K., Robert, C.P., 2005. Bayesian Modelling and Inference on Mixtures of Distributions. In: Dey, D., Rao, C.R. (Eds.), Handbook of Statistics, 25. Elsevier, North-Holland, Amsterdam, pp. 15840–15845.

Mitchell, C., Harper, M., Jamieson, L., 1999. On the complexity of explicit duration HMMs. Speech and Audio Processing, IEEE Transactions on 3 (3), 213–217.

Mitchell, P.J., Cummings, M.L. 2005. Management of multiple dynamic human supervisory control tasks. The 10th International Command and Control Research and Technology Symposium (ICCRTS), McLean, VA.

Nehme, C.E., Crandall, J. Cummings, M.L., 2008. Using discrete-event simulation to model situational awareness of unmanned-vehicle operators. 2008 Capstone Conference. Norfolk, VA.

Ollero, A.,, Maza, I., 2007. Multiple Heterogeneous Unmanned Aerial Vehicles. Springer, Berlin Heidelberg.

Pentland, A., 2008. Honest Signals: How they Shape Our World. MIT Press, Cambridge, Mass.

Pentland, A.,, Liu, A., 1999. Modeling and prediction of human behavior. Neural Computations 11 (1), 229–242.

Rabiner, L., Juang, B., 1986. An introduction to hidden Markov models. ASSP Magazine, IEEE [see also IEEE Signal Processing Magazine] 3 (1), 4–16.

Rabiner, L.R., 1989. A tutorial on Hidden Markov Models and selected applications in speech recognition. Proceedings of the IEEE 77 (2), 257–286.

Schraagen, J.M., Chipman,, S., Shalin, V.E., 2000. Cognitive Task Analysis. Erlbaum, Mahwah, NJ.

Sheridan, T.B., 1992. Telerobotics, Automation and Human Supervisory Control. The MIT Press, Cambridge, MA.

Singh, S., Tu, H., Donat, W., Pattipati,, K., Willet, P., 1996. Anomaly detection via feature-aided tracking and hidden Markov models. IEEE Transactions on Systems, Man, and Cybernetics.

Terran, L., 1999. Hidden Markov Models for Human Computer Interface Modeling International Joint Conferences on Artificial Intelligence, Workshop on Learning About Users, Stockholm, Sweden.

Wayne, D.G., Bonnie,, E.J., Michael, E.A., 1992. The precis of Project Ernestine or an overview of a validation of GOMS. SIGCHI Conference on Human Factors in Computing Systems. ACM, Monterey, California, United States.

Welford, A.T., 1952. The psychological refractory period and the timing of high-speed performance—a review and a theory. British Journal of Psychology 43, 2–19.

Please cite this article as: Boussemart, Y., Cummings, M.L., Predictive models of human supervisory control behavioral patterns using hidden semi-Markov models. Engineering Applications of Artificial Intelligence (2011), doi:10.1016/j.engappai.2011.04.008