Using Hierarchical HMMs in Dynamic Behavior Modeling

Shih-Yang Chiao
Department of Communication Systems
Lancaster University, Lancaster,
LA1 4YR
U.K.
s.chiao@lancaster.ac.uk

Costas S. Xydeas
Department of Communication Systems
Lancaster University, Lancaster,
LA1 4YR
U.K.
c.xydeas@lancaster.ac.uk

Abstract - *This paper is concerned with the modeling of the behaviors of players operating in competitive environments that are characterized by interactions amongst players or group of players. Thus the development of a new, on-line, hierarchical, probabilistic behavior modeling system architecture that is based on Discrete Hidden Markov Models (HMM-D) is presented. For the purpose of integrating multiple object features in on-line behavior modeling, a novel Multi-HMM discrete densities approach is implemented that (i) takes into account the statistical dependency that may exist between different discrete input features and (ii) is accompanied with a probabilistic decision-making tree. The resulting system efficiently segments the evolution-with-time trajectories of player behaviors and yields improved behavior recognition rates. Furthermore, the hierarchical nature of the system allows individual player classification results to be used towards the modeling and classification of higher-level tactical behaviors of groups of players, as defined within an application envelope. The proposed system is applied in a relatively simple 3-D "air-patrol" scenario and system simulation performance results are provided in terms of certain metrics.*

Keywords: Situation and threat assessment, Hierarchical Behavior Modeling System Architecture, Dependent Training, Behavior Recognition and Prediction.

1 Introduction

This paper addresses the issue of modeling the behavior of players operating in competitive environments. Thus a general modeling methodology is proposed that employs Discrete Hidden Markov Models (HMM-D) [1] in the hierarchical structure shown in Figure 1 and that also takes into account the statistical dependency that may exist between different discrete input features. HMM is a rigorous probabilistic classification framework that has been successfully applied in several applications in general [2] and speech recognition in particular [3]. Furthermore, its natural capability of dealing with time varying patterns of arbitrary lengths is attractive from a behavior-modeling point of view, due to the expected variability in the time lengths of player behaviors. A brief introduction to HMMs and their application to modeling a 3-D air-patrol scenario are presented in the following section.

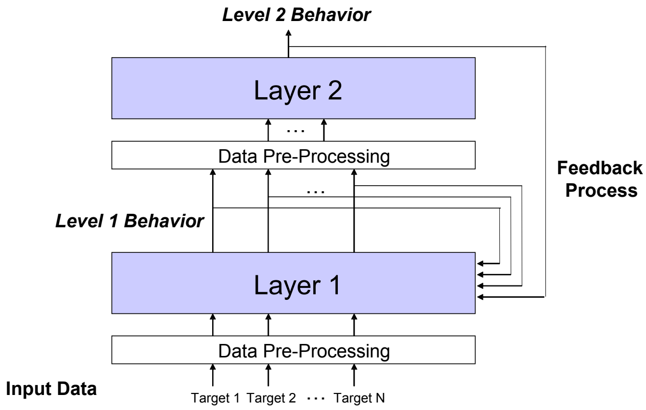

Figure 1. A hierarchical behavior classification structure based on hidden markov models.

2 Single Input Feature HMM-D

A one-feature discrete observation HMM network is described in [3] and [4]. Briefly there are N states $\{S_1,S_2,\dots,S_N\}$ in the model and M possible observations can be generated by the model. At every time step one of the states, say S_j, is entered based on the state transition probability $\{a_{ij}\}$ which depends on the previous state S_i.

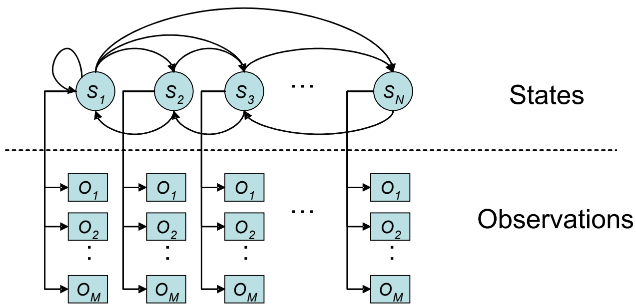

Figure 2. A general HMM structure with N Hidden States and M possible observations per state.

After each transition is made, an observation, say the m-th observation O_m, is produced from S_j with corresponding observation probabilities $\{b_j(O_m)\}$, note that the initial state probabilities are defined as $\{\pi_i\}$. A compact notation

$\lambda^p=\{\{a_{ij}\},\{b_j(O_m)\},\{\pi_i\}\}$ is set to indicate the pth model parameters, $p=1,...,P$. Therefore given an observation sequence $O=\{o_1,o_2,...,o_T\}$ obtained over a period of time T, the likelihood probability $P(O|\ \lambda^p)$ can be calculated by tracing the Viterbi paths [3]. Figure 2 shows an example of such a HMM network.

2.1 3-D Air-Patrol Application

In most real-time HMM systems, observations O_m are not single values but vectors of feature values observed at every time step m [5][6]. Different features can be obtained and used from different sensors and techniques which select/combine these features are of importance and have a direct effect on system performance.

In a 3-D air-patrol type of environment, multiple features are extracted at the same time step, on a per player basis. Observations can be represented statistically by two types of probability densities, i.e. discrete or continuous. Figure 3 shows object (observer)/player situations that illustrate the need for using multiple-features in a 3-D air-patrol application. The distance measured between a player and object (observer) can be one of the features, say Feature 1. Notice that the values of Feature 1 observed at a given instant in time are similar in Figures 3(a) and (b) and also in Figures 3(c) and 3(d). However the players in Figures 3(a) and 3(c) exhibit the same type of behavior or (i.e. Civil Air Patrol (CAP)), whereas the behavior of the players in Figures 3(b) and 3(d) is that of a "Straight Line". Therefore the use of multiple features is necessary since single features are unlikely to be able to offer effective discrimination, i.e. between the "CAP" and" Line" behaviors in this example. Thus sequences of observation feature vectors can be modeled and classified via HMMs employing discrete (HMM-D) or continuous (HMM-C) probability densities, as outlined in the next section.

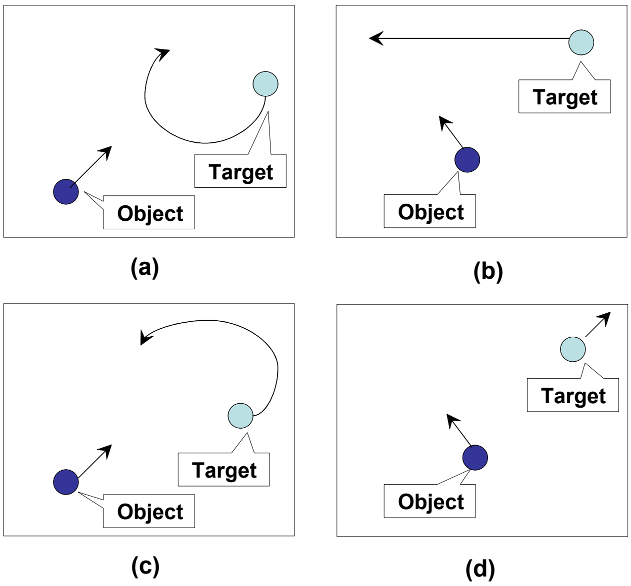

Figure 3. Example of 3-D air-patrol instance situations between object and player.

Furthermore, it is important at this point to consider the "on-line" nature of the required behavior classification and the fact that observation sequences represent a succession of single (Level one) behaviors like "CAP" or "Line". This implies the need of a segmentation process to be applied across the "evolution-with-time" sequence of observations. Segmentation will provide the observation points in time where the behavior of a player changes. An accurate segmentation process is of course important in obtaining accurate classification performance for the different behaviors, since it will allow only the appropriate part of an observation sequence to be presented to the classification process.

Another interesting issue which can be addressed within the context of modeling the behavior of players is that of predicting future behavior(s) of individual players or groups of players. Thus, higher-level behaviors (i.e. maneuvers) can be modeled by adding additional layers to hierarchical system architecture and allowing a feedback process to operate between layers. This enables lower level decisions to be revisited and inconsistencies to be removed at lower levels, in a process that includes further refinement of the segmentation process. Figure 1 shows such a two level classification system.

3 HMMs with Feature Vectors

When multiple input features are obtained at the same time step, vector-format observations can be modeled by two types of probability densities, namely discrete or continuous.

3.1 HMM-D

In the case of HMMs with discrete densities, a vector quantizer (VQ) is employed to represent an input vector with one of the vectors that populate a finite size codebook [7]. Sequences of quantized vectors are then modeled by a single HMM-D structure. This approach is computationally efficient. However, since the VQ process involves designing (training) a codebook, HMM parameters must be re-estimated when a new feature is added to the input vector. Moreover, an inaccurate VQ design can degrade significantly system classification performance.

An alternative method can be formulated when features (elements) in the input vector are assumed to be independent [8]. In this case a single HMM network is designed and employed for each input discrete (quantized) feature (element). Thus when C features $\{o_t^{(1)},o_t^{(2)},...,o_t^{(C)}\}$ (vector elements) are defined at a given time t; the system employs C HMM classifiers, i.e. there are C models per class, and the total likelihood probability $P(O|\ \lambda^p)$ is given as:

$$P(O|\lambda^p)=\prod_{i=1}^{C}P(O^{(i)}\mid\lambda_i^p)\text{, }p=1,2,...,P \quad (1)$$

This "multi-HMM-D" process is based on the assumption that input features are statistically independent (noted hereafter as IM-HMM-D) and may work well in certain type of applications but not in others.

3.2 HMM-C

HMMs with continuous mixture densities assume that the distributions of observations can be represented as a mixture of continuous densities [9] (Gaussian density is frequently used). Thus observation transition probabilities are considered as mixtures of the form:

$$b_j(O_t) = \sum_{k=1}^{K} \omega_{jk} f_{jk}(O_t) \qquad (2)$$

where $f_{jk}(O_t)$ is a Gaussian density with mean μ_{jk} and a covariance matrix U_{jk}. Note that the summation of mixture weight w_{jk} through all K mixtures is equal to one. A general Gaussian density [10] is given as:

$$f_{jk}(O_t) = \frac{1}{\sqrt{(2\pi)^L \left|U_{jk}\right|}} e^{-\frac{1}{2}(O_t-\mu_{jk})' \cdot U_{jk}^{-1} \cdot (O_t-\mu_{jk})} \qquad (3)$$

In many applications, observation vectors are “continuous” vectors (or signals) and HMM-C offer high classification performance. In other applications; however, vector-format input observations are not continuous, correlation coefficients of continuous densities are uncorrelated or nearly close to zero, the matrices U_{jk} are nearly singular [11] and $f_{jk}(O_t)$ estimates can be inaccurate.

The next session presents a new Multi-HMM-D modeling approach which recognizes the existence of statistical dependencies among the elements (features) of input observation vectors.

4 Dependent Multi-HMM-D

Consider that C features are extracted from the same object at a given time and that each feature produces a sequence of observations $O^{(i)}=\{o_1^{(i)},o_2^{(i)},\ldots,o_T^{(i)}\}$, $i=1,\ldots,C$, over a time period T. In the conventional IM-HMM-D system of Figure 4(a), sequences of observations are produced from sequences of states (Viterbi traces) $Q^{(i)}=\{q_1^{(i)},q_2^{(i)},\ldots,q_T^{(i)}\}$, which are “independent” between features. However, dependency between features can be introduced as shown in the DM-HMM-D system of Figure 4(b) where states that belong to different features (paths) are now “linked”.

In this system the resulting likelihood probability is defined as:

$$P^*(O|\lambda^p) = \log(P(O|\lambda^p)) = \sum_{i=1}^{C} \log(w_i(O)\cdot P(O^{(i)}|\lambda_i^p)) \qquad (4)$$

$$p=1,2,\ldots,P$$

where the weight $w_i(O)$ is a function of all observations and is attached to the likelihood probability of the i-th feature. Furthermore, $w_i(O)$ effectively represent the importance or otherwise of individual input features within the HMM classification framework.

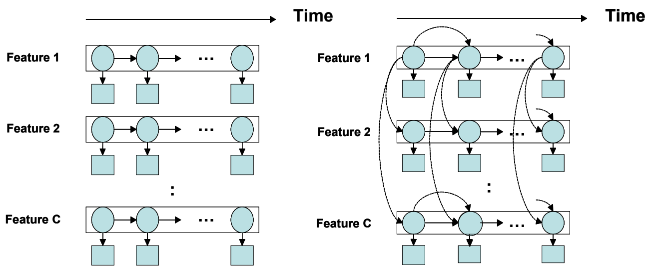

Figure 4. (a) Conventional state transitions across C independent HMM paths; (b) Dependency is introduced by linking states across feature paths, DM-HMM-D.

Note that it is difficult to calculate directly these weights; however, they can be estimated via a “training” process which calculates a $C\times C$ “dependency” matrix D.

$$D = \begin{bmatrix} D(N_1,N_1) & D(N_1,N_2) & \cdots & D(N_1,N_C) \\ D(N_2,N_1) & D(N_2,N_2) & \cdots & D(N_2,N_C) \\ \vdots & \vdots & \ddots & \vdots \\ D(N_C,N_1) & D(N_C,N_2) & \cdots & D(N_C,N_C) \end{bmatrix} \qquad (5)$$

N_i is the number of states in the i-th HMM and $D(N_i,N_j)$ is a $N_i\times N_j$ matrix representing the relation of connections between the states of the i-th and the j-th HMMs. The element of matrix $D(N_i,N_j)$, $1\leq i,j \leq C$, is described as:

$$D(N_i,N_j) = \begin{bmatrix} d(S_1^{(i)},S_1^{(j)}) & d(S_1^{(i)},S_2^{(j)}) & \cdots & d(S_1^{(i)},S_{N_j}^{(j)}) \\ d(S_2^{(i)},S_1^{(j)}) & \cdots & \cdots & d(S_2^{(i)},S_{N_j}^{(j)}) \\ \vdots & \ddots & & \vdots \\ d(S_{N_i}^{(i)},S_1^{(j)}) & d(S_{N_i}^{(i)},S_2^{(j)}) & \cdots & d(S_{N_i}^{(i)},S_{N_j}^{(j)}) \end{bmatrix} \qquad (6)$$

The joint probability of the u-th state in the i-th model and the v-th state in the j-th model, is noted as $d(S_u^{(i)},S_v^{(j)})$, and was calculated by counting Viterbi traces [12] through all K training data streams. Note that $1\leq u \leq N_i$ and $1\leq v \leq N_j$. The joint probability is calculated using Equations (7) and (8), where $q_{k,t}^{(i)}$ indicates the state of the i-th observation in the k-th data stream at time t. T_k is the length of the kth data stream.

$$d(S_u^{(i)},S_v^{(j)}) = \frac{\sum_{k=1}^{K}\sum_{t=1}^{T_k} h(q_{k,t}^{(i)},q_{k,t}^{(j)},S_u^{(i)},S_v^{(j)})}{\sum_{k=1}^{K} T_k} \quad ;i \neq j \qquad (7)$$

$$d(S_u^{(i)},S_v^{(j)}) = \frac{\sum_{k=1}^{K}\sum_{t=1}^{T_k-1} h(q_{k,t}^{(i)},q_{k,t+1}^{(j)},S_u^{(i)},S_v^{(j)})}{\sum_{k=1}^{K} T_k - 1} \quad ;i = j \qquad (8)$$

Furthermore the counting function $h(a,b,c,d)$ is equal to one if and only if the conditions $\{a=c\}$ and $\{b=d\}$ are satisfied, otherwise it is zero.

During classification Equation (4) can be approximated as:

$$
\begin{aligned}
P^*(O|\lambda^p) &= \sum_{i=1}^{C}\log(P(O^{(i)}|\lambda_i^p)) + \sum_{i=1}^{C}\log(w_i(O)) \\
&\approx P^*_{indep} + \log(\prod_{i=1}^{C} p(Q^{(i)})) + \log((\prod_{i,j=1,i\neq j}^{C} p(Q^{(i)}|Q^{(j)}))^{\frac{1}{2}}) \\
&= P^*_{indep} + \log(\prod_{i=1}^{C}\prod_{t=1}^{T-1} d(q_t^{(i)}, q_{t+1}^{(i)})) + \log((\prod_{i,j=1,i\neq j}^{C}\prod_{t=1}^{T} d(q_t^{(i)}, q_t^{(j)}))^{\frac{1}{2}}) \\
&= P^*_{indep} + P^*_{dep} \qquad (9)
\end{aligned}
$$

where joint probabilities $d(q_t^{(i)},q_t^{(j)})$ are obtained from the predefined matrix *D*. Note that Equations (1) and (9) are the same when the weights w_i are equal to one, i.e. under the assumption of feature independency.

5 Decision Making Tree

A decision making tree based on the maximum a posterior (MAP) rule is employed in order to decide player behaviors and to detect behavior changes, using HMM classification probabilities. Figure 5 shows the structure of this segmentation and decision-making tree. Notice that the system is also allowed to "look ahead", i.e. to gather input data from the first *K* time points, before on-line classification is initiated. In this case a delay time T_{delay} is specified.

A different HMM model is available (obtained during training) for each type of behavior. Thus, when the *i*-th HMM model produces the highest likelihood value, given a input set of unknown type data, this indicates that the input data belongs to the *i*-th type of behavior. In Figure 5, the behavior P_B indicates the behavior type with the highest probability and *M* indicates the Level-2 behavior type with the highest probability.

The decision tree also employs dynamic maximum and minimum likelihood probabilities $\{P'_{max}, P'_{min}\}$ for all Level-1 behavior types, in order for the system to confirm the "continuation" of the same behavior with time. This mechanism of dynamic thresholds is employed since maximum likelihood probability values are changing relatively slowly with time unless the observed input behavior had been changed. Notice that Level-2 behavior information is also employed within the decision making tree process. This is represented in Figure 1 as feedback information from the Level 2 to the Level 1 parts of the system.

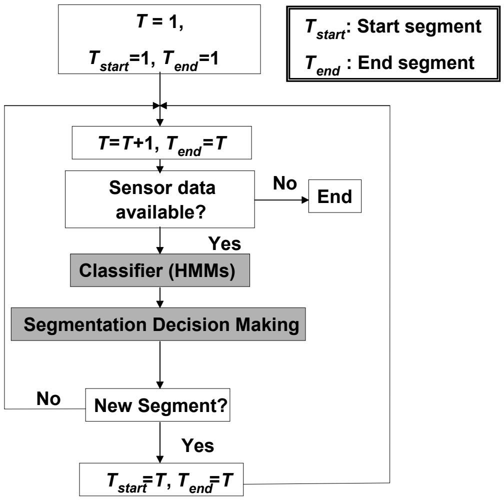

Figure 6. Integrated HMM classification and segmentation process.

The whole integrated segmentation/classification procedure is outlined in Figure 6. Given a segment starting point T_{start}, segment size is incremented and the resulting observation sequences are presented to the HMM classifiers.

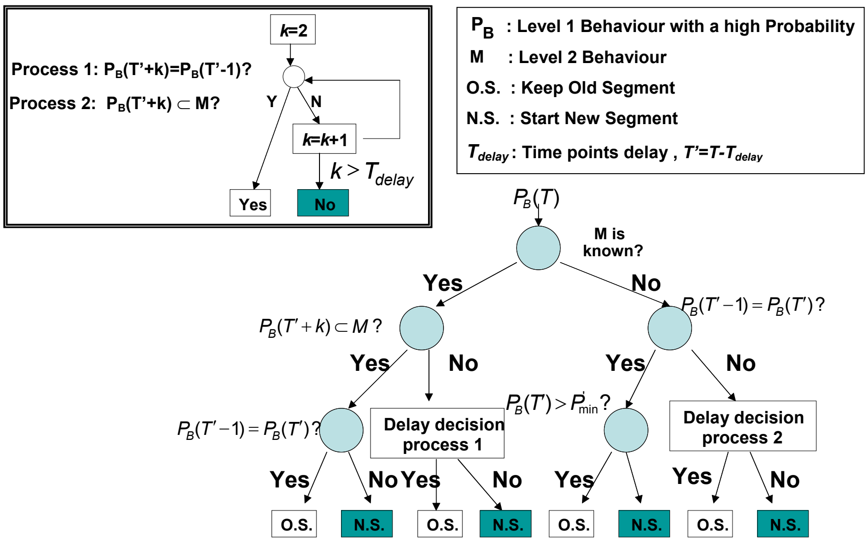

Figure 5. Segmentation and decision-making tree mechanism

6 Experimental Results

A simple 3-D air-patrol environment was simulated [13][14] where players were allowed to perform seven different Level-1 types of behaviors [15], i.e. Line, Clockwise CAP (Circle Air Patrol), Counterclockwise CAP, Intercept, Take-off, Landing and unknown (not recognized as one of the first six behaviors). The six features listed in Table 1 were used as inputs to the Level-1 segmentation/classification part of the system. Examples of 3-D air-patrol Level-1 behaviors are shown in Figure 7 where several red "targets" fly within the operational envelope of the green observer.

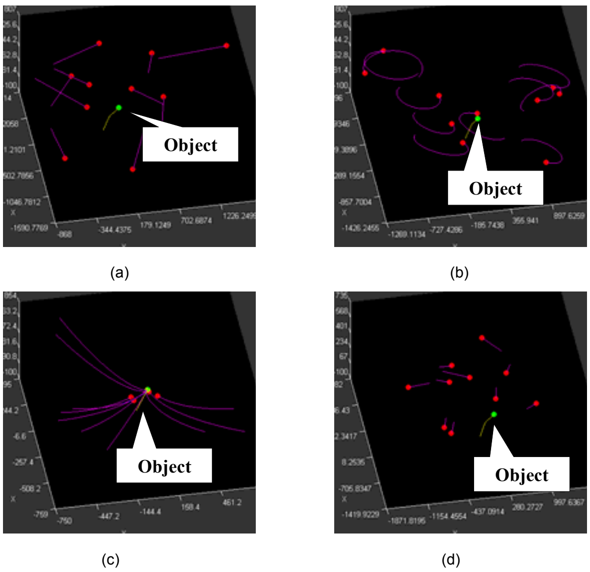

Figure 7. Examples of 3-D behaviors of targets doing (a) Line, (b) CAP (Clockwise and counterclockwise), (c) Intercept, and (d) Take-off and Landing.

1000 different versions of each Level-1 behavior were produced and used to train the HMM models. The System performance was evaluated using data generated from 100 different occurrences of each behavior.

Table 1. Level 1 Features

No. of Feature	Feature Description
1	The distance between object and target
2	The distance between a fixed point and target
3	Speed of target
4	Flight direction of target
5	Altitude of target
6	X-Y plain distance between object and target

Furthermore four Level-2 behaviors were also simulated, that is (i) two patrol aircraft in a single CAP and once the observer (green) is detected within a specified distance, one of the aircrafts commences an intercept action, (ii) two pairs of aircraft in two CAPs and upon detection, within a specified distance, an aircraft from the other pair commences an intercept, (iii) two pairs of aircraft in CAP and once the observer is detected, another aircraft that is on the ground initiates take-off and intercept and (iv) other (none of the first three maneuvers).

The second layer in the system operates with the six input features listed in Table 2.

Table 2 . Level 2 features

No. of Feature	Feature Description
1	The number of targets exhibiting Line behavior
2	The number of targets exhibiting CAP behavior
3	The number of targets exhibiting Intercept behavior
4	The number of targets exhibiting Take-off behavior
5	The number of targets exhibiting Landing behavior
6	X-Y distance between object and target

A Level-2 behavior example is shown in Figure 8, where aircrafts 1, 2, 3 and 4 perform CAP and once aircraft 0 is detected within a specify distance aircraft 2 commences an intercept action.

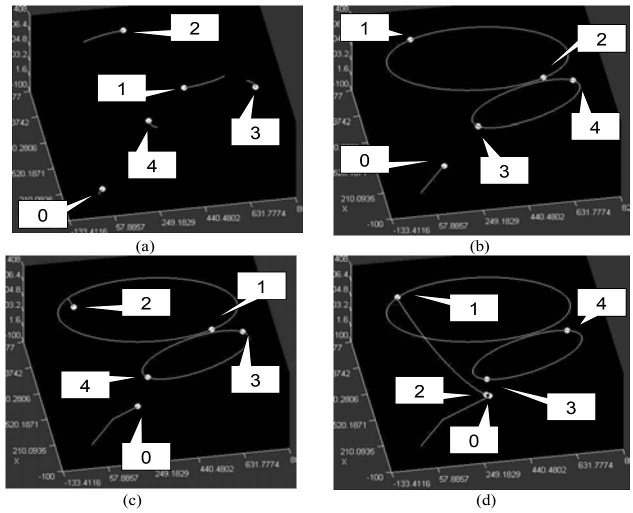

Figure 8. An example of Level-2, Type-2 behavior.

6.1 Data Pre-Processing

The data preprocessing block, see Figure 1, that accepts input sensor and other application related data, performs the processes of normalization and compression often associated with feature extraction and provides observation sequences to HMMs. Furthermore, instead of applying vector quantization on the resulting input data, each signal is scalar quantized to m levels. This produces C observation sequences $O^{(i)}$ per player, $i=1,\ldots,C$.

The procedure of data preprocessing is performed in two stages: (i) each feature, say the i-th feature, is adaptively scalar quantized (SQ) to an level index M_i, by first calculating at every instant in time the maximum and minimum feature values encountered within the current segment $\{o_1^{(i)},\ldots,o_t^{(i)}\}$ and using these values to define the range of the quantizer. That is:

$$M_i = SQ^m{}_{\{o^{(i)}_{max}, o^{(i)}_{min}\}}(o_t^{(i)}) \quad (10)$$

where $\{o^{(i)}_{max}, o^{(i)}_{min}\}$ specify the quantisation range values, as defined from the sequence $\{o_1^{(i)}, \ldots, o_t^{(i)}\}$, to be used at time t. (ii) Repetitions of the same observations are removed from the sequence.

Table 3 shows the effect of the pre-processing procedure. That is, variability in the original input data is drastically reduced together with the length of observation sequences.

Table 3. Effect of the Data Pre-Processing block

	Number of different patterns (Ave.) for Six Feature					Maximum Lengths of Observation Sequences				
Behavior Types	Line	CAP	Int.	T.O.	Land.	Line	CAP	Int.	T.O.	Land.
Before Data Pre-Processing	774	908	840	837	975	130	150	89	132	190
After Data Pre-Processing	19	237	1	19	19	19	28	10	19	19

Table 4 compares the on-line (Level-1) classification performance of six behavior types for the IM-HMM-D and DM-HMM-D methods. It is shown that the extension of the independent HMM networks into the proposed DM-HMM-D scheme offers a significant advantage, i.e. the accuracy rate of recognition increases from 84.667% to 98.168%. Note that systems were trained and tested under the same conditions.

Table 4. Recognition accuracy using IM-HMM-D and DM-HMM-D

Model	IM-HMM-D	DM-HMM-D
Description	6 classes, 6 features and 8-state models	
Accuracy Rate	84.667%	98.168%

System performance was also evaluated in terms of behavior classification error rates with respect to behavior time duration (i.e. behavior segment length) that is available at the input of the classification process.

Figure 9 indicates that errors occur mainly at the very beginning of a new segment, following a behavior changes.

In addition, after a new behavior is adopted by a player, DM-HMM-D provides significantly smaller error rates, within a relatively shorter period of time, as compared to the results obtained from IM-HMM-D. Note that this fast response is of particular importance to critical on-line applications.

6.2 Error Rate vs. Decision Delay Time Points

Experiments were also performed in order to evaluate the on-line behavior classification performance, as a function of the time delay (observation time points needed prior to decision) that may be used in the decision-tree process. Results shown in Figure 10, for the six types of Level-1 behaviors, are obtained while the two systems were trained with 1000 different scenarios per type of behavior in a total of 9000 experiments.

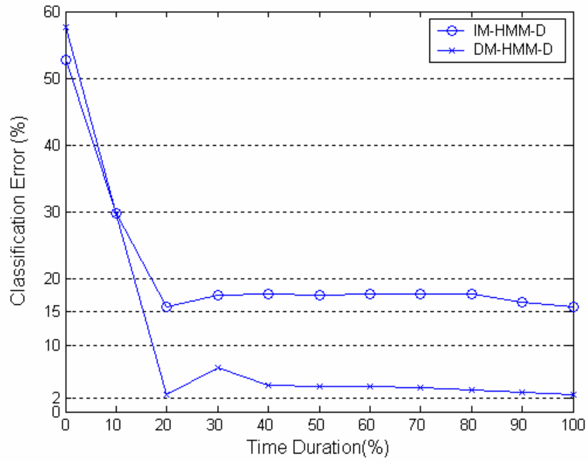

Figure 9 Level-1 behavior classification error rate vs. behavior duration time. Delay is equal to zero points

Furthermore system classification accuracy was tested in 3400 different experiments (100 scenarios per type Level-1 and Level-2 behaviors, excluding impossible combinations of behaviors) for different number of delay time points, see Figure 10. DM-HMM-D error rates are less than one percent when allowing for a fifteen time point decision delay. IM-HMM-D, on the other hand, exhibits more than a fifteen percent error rate with a twenty-five time point's delay.

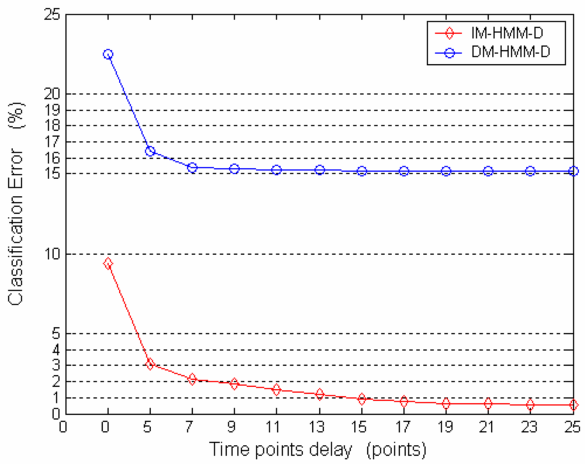

Figure 10 Level-1 behavior classification error rates vs. Delay introduced in the decision tree.

7 Conclusions

An on-line, system architecture is presented in this paper that allows the hierarchical modeling and classification of the behavior of different players. Thus a new DM-HMM-D methodology is proposed that provides significant gains in classification performance, as compared to the conventional IM-HMM-D modeling approach.

The proposed system employs a novel, integrated segmentation/classification process that offers an effective on-line classification. Furthermore, this hierarchical behavior modeling approach is generic and can be expended to include as many tactical and strategic behavior layers as required in a given application domain. Thus a two-layer system has been implemented and its performance characteristics investigated in the case of a simple 3-D air-patrol type of application.

Note that the system can be used as part of and/or extended to form a sophisticated hierarchical situation awareness/risk assessment/prediction process.

References

[1] LE. Baum and T. Petrie. Statistical Inference for probabilistic Functions of Finite State Markov Chains. *Ann Math. Stat.,* vol.37, pp. 1554-1563, 1966.

[2] Yoshua Bengio. Markovian Models for Sequential Data. *Neural Computing Surveys*, vol.2, pp. 129-162, 1999.

[3] Lawrence R. Rabiner. A Tutorial on Hidden Markov Models and Selected Application in Speech Recognition. *Proceedings of IEEE*, vol.77, no.2 pp. 257-285, 1989.

[4] Zoubin Ghahramani. An Introduction to Hidden Markov Models and Bayesian Networks. *International Journal of Pattern Recognition and Artificial Intelligence*, vol.15, no.1, pp. 9-42, 2001.

[5] Kenneth N. Ross and Ronald D. Chaney. Hidden Markov Models for Threat Prediction Fusion. *Proceedings of SPIE,* vol.4051, pp. 300-311, 2000.

[6] Eric Horvitz, Corinne Ruokangas and Sampath Srinivas. A Decision-Theoretic Approach to the Display of Information for Time-Critical Decisions: The Vista Project" *Proceedings of SOAR-92, NASA/Johnson Space Center,* Houston, TX, 1992.

[7] RM. Gray. Vector Quantization. *IEEE ASSP Magazine,* pp. 4-29, 1984.

[8] Kai-Fu Lee. *Automatic Speech Recognition: the Development of the SPHINX System, Kluwer Academic Publications,* 1989.

[9] Xuedong Huang, Kai-Fu Lee, and Hsiao-Wuen Hon. On semi-continuous hidden Markov modeling. *International Conference on Acoustics, Speech and Signal Processing,* pp. 689-692*,* 1990.

[10] Magdi A. Mohamed and Paul Gader. Generalized Hidden Markov Models-Part I: Theoretical Frameworks. *IEEE Transactions on Fuzzy system*, vol.8, no.1, pp. 67-81, 2000.

[11] Todd K. Moon and Wynn C. Stirling. *Mathematical Methods and Algorithms for Signal Processing,* Prentice-Hall, Englewood Cliffs, N.J, 2000.

[12]A. J. Viterbi. Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm. *IEEE Transactions on Information Theory,* Vol.IT-13 pp. 260-269, 1967.

[13]Shih-Yang Chiao and Costas. S. Xydeas. Modelling Competitive Behaviors by Hierarchical HMM Architecture. *Annual PGNet Proceedings Symposium on Telecommunications, Networking and Broadcasting,* pp. 285-287, 2003.

[14]Shih-Yang Chiao and Costas S. Xydeas. Modelling Behaviors of Players in Competitive Environment. *IEEE/WIC International Conference on Intelligent Agent Technology,* pp. 566-569, 2003.

[15]Robert L. Shaw. *Fighter Combat: Tactics and Maneuvering*, Airlife Publishing Ltd, England. 1985.