Incremental learning of full body motions via adaptive Factorial Hidden Markov Models

Dana Kulić, Wataru Takano and Yoshihiko Nakamura
Department of Mechano-Informatics, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656 Japan
{dana,takano,nakamura}@ynl.t.u-tokyo.ac.jp

Abstract

This paper describes a novel approach for incremental learning of motion pattern primitives through long-term observation of human motion. Human motion patterns are abstracted into a factorial hidden Markov model representation, which can be used for both subsequent motion recognition and generation. The model size is adaptable based on the discrimination requirements in the associated region of the current knowledge base. As new motion patterns are observed, they are incrementally grouped together based on their relative distance in the model space. A new algorithm for sequentially training the Markov chains is developed, to reduce the computation cost during model adaptation. The resulting representation of the knowledge domain is a tree structure, with specialized motions at the tree leaves, and generalized motions closer to the root. Tests with motion capture data for a variety of motion primitives demonstrate the efficacy of the algorithm.

1. Introduction

Learning from observation is an attractive proposition for humanoid robots, as the similar body structure to humans can be utilized to bootstrap learning. A variety of algorithms have been proposed for learning human motions through demonstration and imitation (Breazeal and Scassellati, 2002, Schaal et al., 2003). As noted by Breazeal and Scasellati, the majority of algorithms discussed in the literature assume that the motions to be learned are segmented a-priori, and that the model training takes place off-line. In this case, there is no need to segment or classify the observed motions, as the required information is provided by the designer. However, a robot which is an inhabitant of the human environment should be capable of continuous learning over its' entire lifespan. The robot should be able to observe, segment and classify demonstrated actions on-line during co-location and interaction with the (human) teacher. During this type of learning, the number of motion primitives is not known in advance and may be continuously growing, and must be determined autonomously by the robot, as it is observing the motions. In addition, as the number of observed motions and learned motion primitives increases, the robot must be able to organize the acquired knowledge in an efficient and easily searchable manner.

Due to the fact that motions are being sorted incrementally, before all the motions are known, there will be a tradeoff between classification accuracy and the number of training examples. As more examples become available, a more accurate clustering can be achieved. However, if the robot is able to extract appropriate motion primitives quickly, after only a few examples are shown, the learned motion primitives can immediately be used for motion recognition and generation. The learned motion can then be further refined through other learning modalities, such as practice (Bentivegna et al., 2006) and feedback from the teacher (Nicolescu and Matarić, 2005), which may be more effective than repeated observation alone. Therefore, we seek an algorithm which will accurately cluster with few examples. The ability to cluster quickly and accurately will depend on the similarity between the motions to be clustered, and the accuracy of the model used to represent each motion. For dissimilar motions, even a simple model can be successfully used, while for similar motions, a higher accuracy model is required to distinguish between motions. However, a higher accuracy model requires more memory and time resources. Therefore, it would be preferable if the model choice could be adaptable to the current level of similarity of the motions in the relevant region of the knowledge base.

In order to extract motion primitives during online observation, several key issues must be addressed by the learning system: automated motion segmentation, recognition of previously learned motions, automatic clustering and learning of new motions, and the organization of the learned data into a storage system which allows for easy data retrieval. In this paper, our focus is on the last three items.

1.1 *Related Work*

(Breazeal and Scassellati, 2002) and (Schaal et al., 2003) review recent works on motion learning by imitation. For example, (Billard et al., 2006) use HMM models for motion recognition and generation. The Bayesian Information Criterion (BIC) is used to select the optimal number of states for the HMM. However, all the exemplar motion patterns are acquired and grouped before the training begins, and the number of motions to be learned is specified a priori.

(Ogata et al., 2005) develop a connectionist architecture suitable for long term, incremental learning. In their work, a neural network is used to learn a navigation task during cooperative task execution with a human partner. However, in their implementation, the robot learns only one task, and no hierarchical organization of knowledge takes place.

(Kadone and Nakamura, 2006) describe a system for automated segmentation, recognition and abstraction of human motions based on associative neural networks with non-monotonic sigmoid functions. However, the abstracted motion representation can only be used for subsequent motion recognition, and cannot be used for motion generation.

(Takano and Nakamura, 2006) develop a system for automated segmentation, recognition and generation of human motions based on Hidden Markov Models. In their approach, a set of HMMs is trained incrementally, based on automatically segmented data. Each new motion sequence is added to the HMM which has the highest likelihood of generating the motion. However, no mechanism is proposed for the emergence of a hierarchy among different levels of abstraction.

Another important issue during model learning is the selection of the mode size. The use of the BIC (Billard et al., 2006) or Akaike criterion (Kulić et al., 2007a) have been proposed, however, both these criteria are based on a tradeoff between model performance at characterizing the observations, and the number of parameters, and both require a time-consuming search of the model space to find the best matching model. In the motion recognition domain, the model size required depends not only on the model performance for the current observation, but also on the structure of the knowledge database itself. If there are many similar motions in the database, a more accurate model is required, so that they can be easier discriminated. On the other hand, if motions are dissimilar, a very simple model can easily discriminate between them.

In this paper, a variable structure Hidden Markov Model based representation is used to abstract motion patterns as they are perceived. Individual motion patterns are then clustered in an incremental fashion, based on intra model distances. The resulting clusters are then used to form a group model, which can be used for motion generation. The model size is adjusted automatically on-line, based on the accuracy requirements in the given region of the knowledge space. A new training algorithm for Factorial Hidden Markov Models is proposed, which allows the Markov chains to be trained incrementally, and therefore to take advantage of the existing information in the simpler model when upgrading the model.

1.2 *Connections to Biological Models*

Our motion model is inspired by the mirror neuron system, found in humans and other primates (Rizzolatti et al., 2001, Rizzolatti and Craighero, 2004). The mirror neuron system is believed to be a neural mechanism which links action observation and execution. The same neuronal representation is used for both motion recognition and motion generation. In monkeys, mirror neurons activate only when goal directed actions are observed (for example, grasping an object, or biting into food), but not when the demonstrator mimics the action without the object being present. However, mirror neurons also activate when the monkey cannot observe the action visually, but other means of inferring the action are available (eg. sound or previous knowledge), indicating that, in monkeys, the mirror neurons are used for action understanding, and not primarily for imitation learning. For humans, on the other hand, mirror neurons fire for both goal directed actions and for non-goal directed movements (Rizzolatti and Craighero, 2004). In addition, brain imaging studies indicate that human mirror-neurons code for both the action *and* for the movements forming an action. These two important differences seem to indicate that in humans, mirror neurons are used both for action understanding and imitation learning. In animal and human studies, two important classes of mirror neurons have been found: "strictly congruent" and "broadly congruent" mirror neurons, depending on the specificity of the action being encoded (Gallese et al., 1996, Rizzolatti and Craighero, 2004). In our approach, the variable structure HMM emulates the mirror neuron function, and the location of the model in the hierarchy emulates the congruence properties of mirror neurons. Leaf nodes (most specific models) correspond to strictly congruent mirror neurons, while upper level nodes correspond to broadly congruent mirror neurons.

A second key question in biology and cognitive science is the model of learning, i.e., how do the mirror neurons acquire their mirror-like properties. Heyes and colleagues (Heyes and Ray, 2000, Heyes, 2001) propose the Associative Sequence Learning (ASL) theory of imitation. They postulate that learning is based on *action units*, which are the basic units of the majority of actions being observed. For any sequence being observed, action units may be smaller (i.e., body positions) or larger (sequences of actions). Learning proceeds via two sets of associative processes, resulting in horizontal and vertical links. The horizontal process forms sequence associations between sensory representations of the action units forming the demonstrated action. The vertical process forms associations between the sensory representation for each action unit and the associated motor representation of the same component. This learning can occur either directly, when the action unit is contiguously observed and executed, or indirectly, when a second stimulus, such as a spoken word, is consistently paired on some occasions with sensory input from the action unit and on other occasions with performance of that unit. The ASL theory postulates that the development of the imitation mechanism is highly experience-dependent, consisting of correlation links between sensory and motor data which are formed over time. This theory has recently been verified in human experiments, showing modulation of the mirror effect with counter-intuitive training (Catmur et al., 2007). As also noted by Calinon and Billard (Calinon and Billard, 2007), our approach of modeling the sensory data flow as a set of state vectors linked temporally with a stochastic model (the HMM) and obtained incrementally over time, based on the experiences observed by the robot, conforms with the ASL theory. The HMM model learning corresponds to the horizontal process of the ASL model, which forms sequence associations between sensory representations of the action units.

2. Incremental Behavior Learning

In the proposed approach, a hierarchical tree structure is incrementally formed representing the motions learned by the robot. Each node in the tree represents a motion primitive, which can be used to recognize a similar motion, and also to generate the corresponding motion for the robot. Rather than using a fixed size model for the motions, the model accuracy is adjusted based on the recognition requirements in the given region of the knowledge database. Initially, each motion and motion group is encoded as a simple Hidden Markov model, with few states. As the required model complexity increases, additional dynamic chains are added to the model, to form Factorial Hidden Markov Models (Ghahramani and Jordan, 1997).

2.1 Encoding with HMMs and FHMMs

A Hidden Markov Model (HMM) abstracts the modeled data as a stochastic dynamic process. The dynamics of the process are modeled by a hidden discrete state variable, which varies according to a stochastic state transition model $A[N, N]$, where N is the number of states in the model. Each state value is associated with a continuous output distribution model $B[N, K]$, where K is the number of outputs. Typically, for continuous data, a Gaussian or a mixture of Gaussians output observation model is used. Efficient algorithms have been developed for model training (the Baum-Welch algorithm), pattern recognition (the forward algorithm) and hidden state sequence estimation (the Viterbi algorithm) (Rabiner, 1989). Once trained, the HMM can also be used to generate a representative output sequence by sampling the state transition model to generate a state sequence, and then sampling the output distribution model of the current state at each time step to generate the output time series sequence.

A Factorial Hidden Markov Model (FHMM) (Ghahramani and Jordan, 1997) is a generalization of the HMM model, where multiple dynamic processes interact to generate a single output. In an FHMM, multiple independent dynamic chains contribute to the observed output. Each dynamic chain m is represented by its own state transition model $A_m[N_m, N_m]$ and output model $B_m[N_m, K]$, where M is the number of dynamic chains, N_m is the number of states in dynamic chain m, and K is the number of outputs. At each time step, the outputs from all the dynamic chains are summed, and output through an expectation function to produce the observed output. The expectation function is a multivariate Gaussian function with the chain output as the means, and a covariance matrix representing the signal noise. A comparison of the HMM and FHMM structure is shown in Fig 1.

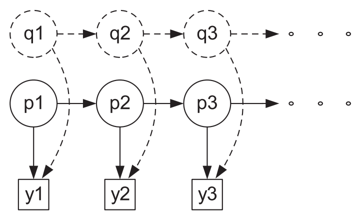
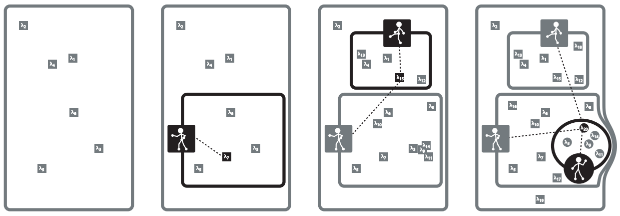


Figure 1: The evolution of a Hidden Markov Model. In a regular HMM (solid lines), at each time step, the hidden state variable is in a state p_t, governed by the state transition model, and outputs observation y_t, governed by the output distribution model. In a Factorial HMM (dashed lines added), multiple hidden state variables p_t and q_t evolve, each according to its own state transition model. At each time step, the observation vectors from all states are summed to form the observation y_t.

FHMMs can provide a more accurate representation of the motion, and enable better discrimination ability in areas of the knowledge base where there are many similar motions, without being prone to overfitting (Ghahramani and Jordan, 1997, Kulić et al., 2007b). In addition, it has been shown that the FHMM and single chain HMM models of the same motion remain sufficiently similar, so that FHMM models may be used simultaneously with HMM models, by using FHMM models only in dense areas of the motion model space where better discriminative ability is required.

2.2 Incremental Learning Algorithm

The algorithm pseudocode is shown in Fig. 3, while a schematic of the incremental memory structure formation is shown in Fig. 2.



(a) (b) (c) (d)

Figure 2: Schematic Illustration of the Segmenting Algorithm. (a) initial state, when only one group is present; (b) a child group forms when enough similar examples are observed; (c) new observations are located into the closest group based on the distance between the new observation and the group model; (d) a higher order model is used in dense areas of the motion space

This algorithm allows the robot to incrementally learn and classify behaviors observed during continuous observation of a human demonstrator. The generation of a hierarchical structure of the learned behaviors allows for easier retrieval, and the automatic generation of the relationships between behaviors based on their similarity and inheritance. In addition, the robot's knowledge is organized based on the type of training received, so that the robot's knowledge will be most specialized in those areas of the behavior space where the most data has been observed.

Each newly acquired observation sequence is encoded into a Hidden Markov Model. In order to train the model, the HMM parameters, such as the number of states and the number of gaussian mixtures must be selected. In previous work, we have used the Akaike Information Criterion (AIC) to select the best fitting model (Kulić et al., 2007a). However, this approach can be time consuming, as it requires training each model candidate, and performing an exhaustive search of the model space. Using the AIC also does not consider the need for a better model when many similar motions need to be distinguished. Instead of using a fixed size model determined by the AIC, in this paper we propose using a variable size model, where the number of dynamics chains in an FHMM model are increased based on the density of the motion exemplars in the relevant region of the motion space. With this approach, each motion is initially represented by a simple, single-chain, front-to-back HMM. If a better model is required, additional chain(s) are added as described below.

procedure INCREMENTALCLUSTER
    Step1 Encode observation sequence O_i into an HMM λ_i
    Step2 Search the behavior tree for the closest group λ_{Gj} to the current observation model λ_i, based on the inter-model distance
    Step3 Place λ_i into the closest group G_c
    Step4 Perform clustering on all the exemplar motions within G_c
    Step5 If a sufficiently similar subgroup of motions is found, form a new group G_n, as a child of G_c, containing the observation sequences of the subgroup
    Step6 Using the observations sequences of the new subgroup, form the group model λ_{Gn}
end procedure

Figure 3: Segmenting Algorithm Pseudocode

Once the newly observed behavior is encoded as an HMM, it is compared to existing groups (if any). Here, the distance between two models can be calculated (Rabiner, 1989) by Equation 1.

$$D(\lambda_1, \lambda_2) = \frac{1}{T}[logP(O^{(2)}|\lambda_1) - logP(O^{(2)}|\lambda_2)] \quad (1)$$

where λ_1, λ_2 are two models, $O^{(2)}$ is an observation sequence *generated* by λ_2 and T is the length of the observation sequence. Since this measure is not symmetric, the average of the two intra distances is used to form a symmetric measure. This distance measure is based on the relative log likelihood that a generated sequence is generated by one model, as compared to a second model. It represents a Kullback-Leibler distance between the two models. The likelihood based distance measure means that it can similarly be applied to Factorial HMM models, by using the modified forward procedure (Ghahramani and Jordan, 1997) to calculate the log likelihood, as well as used to compare FHMM and HMM models (Kulić et al., 2007a).

The repository of known groups is organized in a tree structure, so that the new observation sequence does not need to be compared to all known behaviors. The comparison procedure is implemented as a tree search. If the distance between the new observation and the cluster is larger than a threshold based on the maximum intra observation distance D^G_{max}, this cluster will not be considered as a possible match for the new observation sequence. If there are multiple candidate clusters, the new sequence is placed in the closest cluster. If there are no candidates, the new sequence is placed in the parent cluster. In the case of a new motion pattern which is completely dissimilar to any existing motion patterns, the motion pattern will be placed into the root node.

The maximum intra observation distance for the placement node D^G_{max} is also the criterion used to decide the level of model complexity required for the motion sequence. If the new motion is most similar to a node which D^G_{max} falls below a certain threshold, the FHMM model is generated by adding additional chain(s) to the current representation, to increase the discriminative ability of the model.

When a new observation sequence is added to a group, a clustering procedure is invoked on that group, to determine if a subgroup may be formed. The complete link hierarchical clustering algorithm is used to generate the hierarchical tree structure within a group (Jain et al., 1999). Clusters are formed based on two criteria: number of leaves in the subgroup, and the maximum proximity measure of the potential subgroup.

If a new subgroup is generated in Step 5, a new group model is trained using the raw observation sequences from all the group elements. The structure of the new group model is determined based on the maximum intra observation distance for group, D^G_{max}. The generated model is subsequently used by the robot to generate behaviors. The group model replaces the individual observations in the parent node.

If one of the group elements allocated to the new cluster is already a group model, the generated motion sequence based on that model is used for the training. In this case, a modified form of the re-estimation formulas for multiple observation sequences (Rabiner, 1989) is used. The algorithm is modified by over-weighting the group models, in order to account for the fact that there are multiple observation sequences stored in the generated model, and therefore more weight should be given to the group model, as compared to the individual observation sequences.

3. Sequential Training

Factorial HMMs require a modification to the very efficient Baum-Welch training algorithm. Using either the Gharmani and Jordan approximate training algorithm (Ghahramani and Jordan, 1997), or the generalized-backfitting algorithm (Jacobs et al., 2002), the chains are trained simultaneously, which significantly increases training time. Similarly, due to the more complex structure of the FHMM model, the recognition algorithm (the forward procedure) is more complex and time consuming as compared to a single chain HMM. Therefore, we would like to use a compact representation (single HMM chain) when patterns are easy to distinguish, and a more detailed representation (multiple chains) during generation and when motions are very similar to each other. For this approach, a modified training algorithm is developed, training the chains sequentially instead of simultaneously.

Analyzing the parallel-trained FHMMs (Kulić et al., 2007b), it appears that FHMM chains trained in parallel each approximate a (scaled) version of the motion data. To approximate this result with the sequential training algorithm, first, a standard HMM chain is trained on the motion data, using the Baum-Welch training algorithm. This chain alone can then be used for the initial recognition (during the tree search). Next, additional chains are trained on the error between the true data and the motion generated by the scaled sum of the preceding chains. The training data for each subsequent chain is calculated based on the observation data and the existing contribution from the previously trained chains.

$$e^n_t = \frac{1}{W}(y^n_t - \sum_{i=0}^{m-1} WC_i) \tag{2}$$

where e^n_t is the residual error (a set of N time series data), y^n_t is the original data (a set of N time series data), $W = 1/M$ is the weight applied to each chain, M is the new number of chains, and C_i is the contribution of each previously trained chain i.

The contributions from the previously trained chains can be calculated based on : (a) the $\gamma_i(t)$ values for each chain, i.e.,

$$C_i = \sum_{j=0}^{J} \mu^{(i)}_j \gamma^{(i)}_j(t) \tag{3}$$

where $\mu^{(i)}_j$ is the (already trained) mean values vector for chain i and state j, and $\gamma^{(i)}_j(t)$ is the probability that state j in chain i is active at sample time t,

(b) the Viterbi sequence for each chain, i.e.,

$$C_i = \mu^{(i)}_{j*} \tag{4}$$

where $\mu^{(i)}_{j*}$ is the mean value vector at the current state $j*$, as determined by the Viterbi algorithm for each time series sequence, or

(c) the generated sequence for each chain:

$$C_i = \hat{y^i_t} \tag{5}$$

where $\hat{y^i_t}$ is the output vector generated by chain i at time step t.

Once the training data data for the new chain are generated, the new chain is trained with the Baum-Welch algorithm. Following training, for the forward (recognition algorithm), the variance at each state combination is computed as follows:

$$\Sigma = \sum_{i=0}^{M} W^2 \Sigma_j^{(i)} \tag{6}$$

where Σ is the resulting covariance, and $\Sigma_j^{(i)}$ is the covariance at the corresponding state j of chain i.

The pseudo-code for the algorithm is outlined in Figure 4.

procedure SequentialFHMMTrain
    Initialize first chain
    $A_0[N_0, N_0]$, $B_0[N_0, K]$, $\pi_0(i)$
    Baum-Welch algorithm: train chain on time series y_t^n
    for $m \leftarrow 1, M$ **do**
        Initialize next chain
        $A_m[N_m, N_m]$, $B_m[N_m, K]$, $\pi_m(i)$
        Calculate Residual Error $e_t^n = \frac{1}{W}(y_t^n - \sum_{i=0}^{m-1} WC_i)$
        Baum-Welch algorithm: train chain on time series e_t^n
    end for
end procedure

Figure 4: Sequential FHMM Training Algorithm Pseudocode

The developed algorithm approximates the actual FHMM distribution with a tractable distribution assuming that the Markov chains are independent given the data. This is equivalent to a single pass of the generalized-backfitting algorithm (Jacobs et al., 2002), or the structured variational inference approximation (Ghahramani and Jordan, 1997). In addition, the sequential approach allows the on-line segmenting algorithm to re-use the training information of the simpler model (i.e., the previously trained chains).

4. Simulations

The proposed approach was tested on a data set containing a series of 9 different human movement observation sequences obtained through a motion capture system (Kadone and Nakamura, 2005). The data set contains joint angle data for a 20 degree of freedom humanoid model from multiple observations of walking (WA - 28 observations), cheering (CH - 15 observations), dancing (DA - 7 observations), kicking (KI - 19 observations), punching (PU - 14 observations), sumo leg raise motion (SL - 13 observations), squatting (SQ - 13 observations), throwing (TH - 13 observations) and bowing (BO - 15 observations).

4.1 Sequential Training Verification

In the first set of simulations, the performance of the sequentially trained Factorial HMMs is compared to FHMMs trained with the exact algorithm (Ghahramani and Jordan, 1997). For each type of motion, four training procedures were applied: exact training (Ghahramani and Jordan, 1997), sequential training using Gamma-based residual error (equation 3), sequential training using Viterbi-based residual error (equation 4), and sequential training using Generation-based residual error (equation 5). Each FHMM consisted of a two chains of front-to-back (Bakis type) hidden Markov chains of 15 states each. For both training methods, the covariance matrix was constrained to be diagonal during training, and the minimum covariance was constrained to 0.001, to avoid numerical underflow/overlow during training.

Table 1: FHMM Recognition Results, Exact Training. The training column indicates the average log likelihood and standard deviation over the 100 test cases for a randomly selected exemplar out of the training set, the novel column indicates the log likelihood distribution for an example of the same motion, but outside of the training set, and the different column indicates the log likelihood distribution for an example of a different motion.

	Training	Novel	Different
WA	2134 ± 165	1616 ± 205	-66589 ± 15158
CH	2468 ± 284	1220 ± 724	-60908 ± 19899
DA	3016 ± 288	1675 ± 913	-60922 ± 15016
KI	1344 ± 170	733 ± 651	-48060 ± 17208
PU	2613 ± 295	1281 ± 868	-52786 ± 23188
SL	2818 ± 350	791 ± 1418	-49262 ± 20734
SQ	2617 ± 229	2025 ± 302	-58060 ± 17283
TH	3223 ± 254	1893 ± 641	-48517 ± 21690
BO	2599 ± 295	1821 ± 681	-54118 ± 18877

Each model was trained on 5 randomly selected exemplars of a motion type. The learning algorithm for each model was run for a maximum of 100 iterations, or until convergence. Each model was then tested by computing the log-likelihood of observing the following observation sequences: an example from the training set, a randomly drawn example of the same motion outside the training set, and an example of a different motion. This testing procedure was repeated for 100 trials. At the start of each trial, the state transition parameters were initialized to random values. The mean and variance parameters in the output distribution model were initialized by calculating the mean values and covariance of the output vector over all the training data. The means were initialized by sampling from a Gaussian distribution with the data based means and covariance. The tests were carried out on a 2.66GHz Intel Xeon CPU. The average time per training cycle for the exactly trained FHMMs was 2.4 seconds, compared to 0.1 seconds for the sequentially trained models. The results are shown in Tables 1 and 2, for the FHMM models trained via the exact method, and the sequentially trained FHMM models, respectively.

Among the three sequential training approaches, the sequential training using Gamma-based residual error (equation 3) achieved the best results, however, the difference was not significant. As can be seen from Tables 1 and 2, the sequentially trained models achieve comparable and sometimes improved results over the Ghahramani and Jordan (Ghahramani and Jordan, 1997) approach, while requiring significantly less training time. The improvement in performance is likely due to the fact that Ghahramani and Jordan assume a single, constant covariance matrix across all states, whereas in the proposed approach, separate covariance matrices at each state are estimated, providing a more accurate estimate of the variance.

Table 2: FHMM Recognition Results, Sequential Training (using Gamma-based residual error).

	Training	Novel	Different
WA	2271 ± 90	1930 ± 154	-58235 ± 19017
CH	2632 ± 151	1731 ± 526	-44197 ± 23454
DA	3290 ± 327	2414 ± 709	-52712 ± 19468
KI	1564 ± 146	1081 ± 427	-27053 ± 14655
PU	2735 ± 262	1736 ± 732	-41404 ± 17690
SL	2936 ± 317	1505 ± 1617	-23740 ± 15093
SQ	2863 ± 158	2453 ± 171	-38365 ± 13790
TH	3331 ± 177	2384 ± 469	-27690 ± 17980
BO	2851 ± 268	2260 ± 513	-26450 ± 15528

4.2 Learning Algorithm Verification

In the second set of experiments, the incremental motion learning algorithm is validated. In this set of experiments, motion sequences are presented to the algorithm in random order. Motion sequences are presented one at a time, simulating on-line, sequential acquisition. After each motion is presented, the algorithm is executed, performing incremental clustering. In each simulation performed, the algorithm correctly segments the behaviors such that the resulting leaf nodes represent the grouping that would be obtained with an off-line method. Out of 100 simulation runs performed, there was no cases of misclassification at the leaf nodes, showing that the final segmentation is robust to presentation order. A comparison of the clustering performance between using only HMM models, and using adaptable models is shown in Fig. 5. The adaptable model can distinguish between similar motions TH and PU, whereas those motions cannot be distinguished given the same number of examples, when only single chain HMM models are used. Note that the actual order of node formation will vary depending on the motion presentation order.

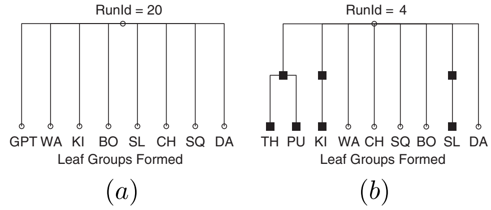


Figure 5: Sample Segmentation Results: (a) using only single-chain model, most motions are segmented correctly, but punch and throw cannot be distinguished and are grouped together in group GPT; (b) using adaptable models, all motions can be correctly distinguished (black squares indicate branches where higher accuracy models have been applied by the algorithm)

Figure 6 shows a comparison between motion encoded by the two chain FHMM and the single chain HMM before any trajectory post processing has been applied, for the walk subgroup. As can be seen in Figure 6, due to the higher number of states available to represent the motion, FHMMs achieve better spacial accuracy compared to a single chain HMM model.

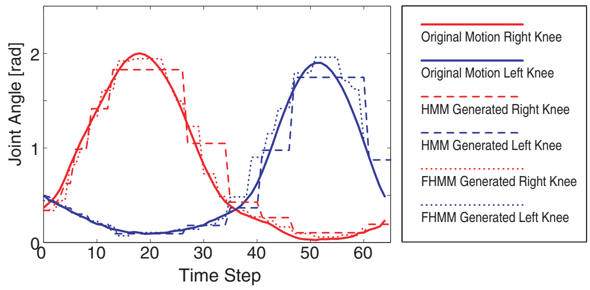


Figure 6: Comparison of Generation Results of the HMM and FHMM (two chains) for the Knee Joints during a Walking Motion, prior to applying any post processing

5. Conclusions

This paper develops a novel approach towards on-line, long term incremental learning and hierarchical organization of whole body motion primitives. The learned motions are aggregates of the observed motions, which have been autonomously clustered during observation. The appropriate level of accuracy required for each motion model is determined based on the similarity of the motions to be distinguished, such that a larger model is only used in dense regions of the knowledge base. A sequential training approach is introduced, to allow incremental training of the Markov chains, as the model accuracy requirements increase. The proposed algorithm achieves comparable results to the exact training algorithm, while significantly reducing the training time, and allowing existing model knowledge to be reused.

The clustered motions are organized into a hierarchical tree structure, where nodes closer to the root represent broad motion descriptors, and leaf nodes represent more specific motion patterns. The tree structure and level of specialization will be based on the history of motions observed by the robot. The resulting knowledge structure is easily searchable for recognition tasks, and can also be utilized to generate the learned robot motions.

Acknowledgment

This work is supported by the Japanese Society for the Promotion of Science grant 18.06754 and Category S Grant-in-Aid for Scientific Research 15100002.

References

Bentivegna, D. C., Atkeson, C. G., and Cheng, G. (2006). Learning similar tasks from observation and practice. In *IROS06*, pages 2677–2683.

Billard, A., Calinon, S., and Guenter, F. (2006). Discriminative and adaptive imitation in uni-manual and bi-manual tasks. *Robotics and Autonomous Systems*, 54:370–384.

Breazeal, C. and Scassellati, B. (2002). Robots that imitate humans. *Trends in Cognitive Sciences*, 6(11):481–487.

Calinon, S. and Billard, A. (2007). Learning of gestures by imitation in a humanoid robot. In Nehaniv, C. L. and Dautenhahn, K., (Eds.), *Imitation and Social Learning in Robots, Humans and Animals*, pages 153–177. Cambridge University Press.

Catmur, C., Walsh, V., and Heyes, C. (2007). Sensorimotor learning configures the human mirror system. *Current Biology*, 17:1527–1531.

Gallese, V., Fadiga, L., Rogassi, L., and Rizzolatti, G. (1996). Action recognition in the premotor cortex. *Brain*, 119:593–609.

Ghahramani, Z. and Jordan, M. I. (1997). Factorial hidden markov models. *Machine Learning*, 29:245–273.

Heyes, C. (2001). Causes and consequences of imitation. *Trends in Cognitive Sciences*, 5(6):253–261.

Heyes, C. and Ray, E. (2000). What is the significance of imitation in animals? *Advances in the Study of Behavior*, 29:215–245.

Jacobs, R. A., Jiang, W., and Tanner, M. A. (2002). Factorial hidden markov models and the generalized backfitting algorithm. *Neural Computation*, 14:2415–2437.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323.

Kadone, H. and Nakamura, Y. (2005). Symbolic memory for humanoid robots using hierarchical bifurcations of attractors in nonmonotonic neural networks. In *IROS05*, pages 2900–2905.

Kadone, H. and Nakamura, Y. (2006). Segmentation, memorization, recognition and abstraction of humanoid motions based on correlations and associative memory. In *HUMANOIDS06*, pages 1–6.

Kulić, D., Takano, W., and Nakamura, Y. (2007a). Incremental on-line hierarchical clustering of whole body motion patterns. In *RO-MAN07*, pages 1016–1021.

Kulić, D., Takano, W., and Nakamura, Y. (2007b). Representability of human motions by factorial hidden markov models. In *IROS07*. To Appear.

Nicolescu, M. N. and Matarić, M. J. (2005). Task learning through imitation and human-robot interaction. In Dautenhahn, K. and Nehaniv, C., (Eds.), *Imitation and social learning in robots, humans and animals: behavioral, social and communicative dimensions*. Cambridge University Press.

Ogata, T., Sugano, S., and Tani, J. (2005). Open-end human-robot interaction from the dynamical systems perspective: mutual adaptation and incremental learning. *Advanced Robotics*, 19:651–670.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77(2):257–286.

Rizzolatti, G. and Craighero, L. (2004). The mirror-neuron system. *Annual Reviews of Neuroscience*, 27:169–192.

Rizzolatti, G., Fogassi, L., and Gallese, V. (2001). Neurophysical mechanisms underlying the understanding and imitation of action. *Nature Reviews: Neuroscience*, 2:661–670.

Schaal, S., Ijspeert, A., and Billard, A. (2003). Computational approaches to motor learning by imitation. *Phil. Trans. of the Royal Society of London B: Biological Sciences*, 358:537 – 547.

Takano, W. and Nakamura, Y. (2006). Humanoid robot's autonomous acquisition of proto-symbols through motion segmentation. In *HUMANOIDS06*, pages 425–431.