Coupled hidden Markov models for complex action recognition

Matthew Brand, Nuria Oliver, and Alex Pentland
brand,nuria,sandy@media.mit.edu
Vision and Modeling Group, MIT Media Lab Cambridge, MA 02139

Abstract

We present algorithms for coupling and training hidden Markov models (HMMs) to model interacting processes, and demonstrate their superiority to conventional HMMs in a vision task classifying two-handed actions. HMMs are perhaps the most successful framework in perceptual computing for modeling and classifying dynamic behaviors, popular because they offer dynamic time warping, a training algorithm, and a clear Bayesian semantics. However, the Markovian framework makes strong restrictive assumptions about the system generating the signal—that it is a single process having a small number of states and an extremely limited state memory. The single-process model is often inappropriate for vision (and speech) applications, resulting in low ceilings on model performance. Coupled HMMs provide an efficient way to resolve many of these problems, and offer superior training speeds, model likelihoods, and robustness to initial conditions.

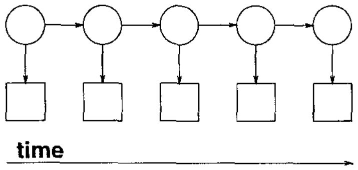
1. Introduction

Computer vision is turning to problems of perceiving and interpreting action, sparking interest in models of dynamical behavior used elsewhere in perceptual computing, particularly hidden Markov models (HMMs). HMMs are presently the most favored model in speech and vision, mainly because they can be learned from data and they implicitly handle time-varying signals. Their clear Bayesian semantics also makes them well-suited for computing with uncertainties.

An HMM is a quantization of a system's configuration space into a small number of discrete states. A single finite discrete variable s indexes the current state of the system. State changes, approximating the dynamics of the system, are described by a table of transition probabilities $P_{i|j} \doteq P_{s(t)=i|s(t-1)=j}$. This representation succeeds to the degree that the system fits the Markov condition: Any information about the history of the process needed for future inferences must be reflected in the current state. Consequently, HMMs are ill-suited to systems that have compositional state, e.g., multiple interacting processes that have structure in both time and space. For example, in video signals one might want to model the behavior of players in a sport, or, more generally, of multi-participant actions normally described by natural language verbs (e.g., "A gave B the C."). We present algorithms for coupling and training HMMs to model interactions between processes that may have different state structures and degrees of influence on each other. These problems often occur in vision, speech, or both—coupled HMMs are well suited to applications requiring sensor fusion across modalities.

2. HMMs and the Markov condition

A hidden Markov model consists of a set of discrete states $S = \{s_1, s_2, s_3, \ldots, s_N\}$, a state variable $s(t) \in S$, state-to-state transition probabilities $P_{i|j} \doteq P_{s(t)=i|s(t-1)=j}$, $1 < i, j < N$, prior probabilities for the first state $P_{s(1)=i}$, and output probabilities for each state $P_i(o) \doteq P_{s(t)=i}(o(t))$. Graphically, Markov models are often depicted "rolled out in time" as probabilistic inference graphs:



Square nodes represent the observations $o(t)$; circular nodes represent the hidden state variable $s(t)$; horizontal arcs represent the transition matrix $P_{s(t)|s(t-1)}$; and parameters associated with the vertical arcs determine the probability of an observation given the current state $P_{s(t)}(o(t))$, e.g., these parameters may be means and covariances (μ_i, Σ_i) of multivariate Gaussians. The state variable and the output vary over time, and at any any time t, memory is limited to the value of state variable $s(t-1)$.

Conventional extensions to the basic Markov model are generally limited to increasing the memory of the system (durational modeling), which give the system compositional state in time. We are interested in systems that have compositional state in *space*, e.g., more than one simultaneous state variable. Recently, Jordan, Saul, and Ghahramani have developed a variety of higher-order HMMs, including factorial HMMs [5] for independent processes; linked HMMs

1063-6919/97 $10.00 © 1997 IEEE

[8] that model noncausal (contemporaneous) symmetrical influences; and hidden Markov decision trees [7] that feature a cascade of noncausal influences from master to slave HMMs. The training algorithms are based on an equivalence between HMMs and a class of Boltzmann machine architectures with tied weights [9, 10]. The linked HMM excepted, these algorithms use mean-field approximations from statistical mechanics.

We present an algorithm for coupling two HMMs with causal (temporal), possibly asymmetric influences. Theoretical and empirical arguments for this architecture's advantages can be found in [2]. To illustrate the difference between causal and noncausal couplings, imagine modeling opponents in a tennis match: The noncausal HMM couplings can represent the fact that it is unlikely to see both players playing net simultaneously; the causal HMM coupling can represent the fact that one player rushing to the net will drive the other back and restrict the kinds of returns he attempts.

The coupling algorithm is based on projections between component HMMs and a joint HMM; in principle it is also possible to derive an approximation algorithm in the mean field framework or an exact algorithm using junction-tree representations [6]. Our experiences with these methods have led to somewhat inferior models and extremely long computations, respectively. We sketch our coupling algorithm here; a detailed exposition including convergence properties and performance analysis can be found in [2].

3. Coupling and Factoring HMMs

Two HMMs are coupled by introducing conditional probabilities between their hidden state variables. The resulting distribution does not satisfy the Markov property. Therefore there is no simple decomposition of the prior probability that might lead to simple parameter estimation procedures. The traditional work-around for modeling a system with two state variables forms a super-HMM from the Cartesian product of all possible states. This is unsatisfactory because the number of states is now squared and training data becomes very sparse on a per-state basis. However, with a very large number of parameters it is very easy to raise the posterior probability of the model, but the result is gross over-fitting of the data and consequently poor generalization. Our algorithm takes this oversized parameter space and embeds within it a subspace manifold which represents all possible parameterizations of a much smaller system of coupled HMMs. Forward-backward analysis obtains posterior state probabilities in the larger space; we calculate the closest point on the manifold and reestimate so that the posterior probability of the model increases but the parameters stay on the manifold.

We obtain a joint HMM C from two component HMMs A, B by taking the Cartesian product of their states a_i, b_j and transition parameters $P_{a_i|a_j} P_{b_k|b_l}$. This results in a quadratic state table with joint states $c_{ij} = \{a_i, b_j\}$. We obtain transition and output probabilities as follows:

$$\begin{aligned} P_{c_{ik}|c_{jl}} &= \Psi(P_{a_i|a_j} P_{b_k|b_l}, P_{a_i|b_l} P_{b_k|a_j}) & (1) \\ P_{c_{ik}}(o) &= P_{a_i}(o) P_{b_k}(o) & (2) \end{aligned}$$

Note that we have introduced coupling parameters $P_{a_i|b_l} P_{b_k|a_j}$. If the combining function Ψ is linear and respects sum-to-one constraints, linear projections will factor the joint HMM back into its components.

$$\begin{aligned} P_{a_i|a_j} &= \sum_l P_{b_l} \sum_k P_{c_{ik}|c_{jl}} \\ P_{b_k|b_l} &= \sum_j P_{a_j} \sum_i P_{c_{ik}|c_{jl}} & (3) \end{aligned}$$

where $P_{b_l} = 1/|\{B\}|$ and $P_{a_j} = 1/|\{A\}|$ in the absence of any posterior probabilities.

This projections factors the $(|\{A\}| \cdot |\{B\}|)^2$-dimensional transition table of the joint HMM into $|\{A\}|^2$– and $|\{B\}|^2$-dimensional transition tables which parameterize two component HMMs. Note that we may just as easily define a projection which factors out the interaction between the component HMMs:

$$\begin{aligned} P_{a_i|b_l} &= \sum_j P_{a_j} \sum_k P_{c_{ik}|c_{jl}} \\ P_{b_k|a_j} &= \sum_l P_{b_l} \sum_i P_{c_{ik}|c_{jl}} & (4) \end{aligned}$$

This is the basis of an algorithm in which a joint HMM is trained via standard HMM methods but constrained to factor consistently along both projections. As propels it up through likelihood space, we factor and reconstitute it, thus simultaneously training the component HMMs. Here we formulate the algorithm with factoring after reestimation of the joint HMM; factoring can also be done after forward-backward analysis, so that reestimation can occur in the component HMMs, e.g.:

$$\begin{aligned} & P_{a(t)=i, a(t-1)=j|O} = \\ & \qquad \frac{\sum_k \sum_l C_{jl,t-1} \cdot P_{ik|jl} \cdot P_{c(t)=ik}(o(t)) \cdot C'_{ik,t}}{P(O)} & (5) \\ \sim & \quad \frac{P_{a(t)=i}(o(t))}{P(O)} \sum_k C'_{ik,t} \sum_l \left(C_{jl,t-1} \cdot P_{ik|jl} \right) & (6) \\ \sim & \quad \frac{P_{a(t)=i}(o(t)) \cdot P_{i|j}}{P(O)} \left(\sum_k C'_{ik,t} \right) \left(\sum_l C_{jl,t-1} \right) & (7) \end{aligned}$$

where C and C' are the forward and backward variables for the joint HMM. Eqns. 6,7 are approximations that allow substantial speed-ups but sacrifice some information.

Note that we do not take the Cartesian product of the output parameters. They are reestimated directly in the component HMMs using posterior component state probabilities. This has three advantages: (1) $O(2N)$ output parameters are reestimated instead of $O(N^2)$; (2) the statistics are more robust; (3) forward-backward analysis and run-time Viterbi analysis are considerably faster, since the bulk of computation is in computing multivariate Gaussians and this is reduced by $O(N)$. E.g., recognition with a CHMM can be considerably faster than with a conventional HMM with the equivalent number of joint states.

In principle, factoring and reconstitution can violate the conditions under which convergence is guaranteed, because the factorings into transition and coupling probabilities may not be consistent. To restore the convergence property, we introduce a post-factoring conditioning step which guarantees that the model parameters have moved to a point on the manifold with higher posterior probability. A simple gradient descent will find a point on the manifold closest to the maximum posterior probability parameterization. If U, V are two factorings of the transition probabilities out of any joint state, the gradient toward the closest manifold point is

$$\begin{aligned}\frac{\partial E}{\partial u_i} &= \sum_j 2u_i v_j^2 - 2v_j w_{ij} \\ &= 2\sum_j v_j(u_i v_j - w_{ij}) \qquad (8)\end{aligned}$$

In practice, conditioning results in a very small improvement, and the unconditioned algorithm has always converged. The results reported below were obtained without conditioning.

4. Experiments

T'ai Chi Ch'uan is a Chinese martial art and meditative exercise, consisting of stylized full-body and upper-body gestures. Like most signals generated by human activity, these gestures are the result of multiple interacting processes. A simple way to decompose upper-body gestures is to treat each arm as a process. The arms are neither independent nor wholly mutually determined; some form of interactional modeling is appropriate.

4.1. Data collection and preprocessing

Using a self-calibrating stereo blob tracker [1], we obtained 3D hand tracking data for three T'ai Chi gestures involving arm-motions: the left[1] single whip, the left cobra, and the left brush knee. Figure 4.1 illustrates the gestures, the blob-tracking, and the feature vectors.

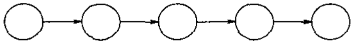
We collected 52 sequences, roughly 17 of each gesture. The extracted feature vector consisted of the 3D (x, y, z) centroid (mean position) of each of the blobs that characterize the hands. All the gestures were performed by the second author, seated in a swivel chair and moving her upper body and hands. Each gesture began with both hands in a rest or neutral position and ended with the hands in a gesture-specific final position or returning to neutral position. The main sources of noise were blob instabilities, variations in the performance of each gesture, and variations in initial body rotation and position from sequence to sequence. The extracted feature vector, being simple (x, y, z) positions, reflects this noise directly.

The frame rate of the vision system varied from 10-20 Hz. We resampled the data using time-stamped frames and cubic spline interpolation to produce a 30Hz signal, then low-pass filtered with a 3Hz cutoff. Similar preprocessing is used by Campbell *et al.* [4], who go on to convert the feature vector to head-centered cylindrical coordinates velocities $(dr, d\theta, dz)$ for rotation and shift invariance; we remain with raw 3D (x, y, z) coordinates. The resulting six-dimensional time series data $(x_r, y_r, z_r, x_l, y_l, z_l)$ was used for training.

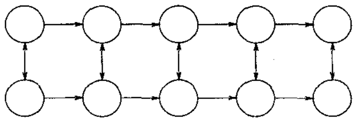
4.2. Results of training different architectures

Three HMM architectures, reflecting different independence structures between hidden states, were trained and tested to find the optimal number of states to model each gesture.

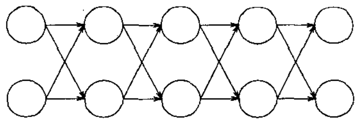
1. Conventional HMMs: 3-state models for cobra and single whip; a 5-state model for brush knee.



2. Linked HMMs (a simplification of CHMMs with symmetric noncausal joint probabilities between chains): 2+2-state models for the cobra and single whip, and a 3+3-state model for the brush knee.



3. Coupled HMMs: 3+3-state models for cobra and brush knee, and a 3+2-state model (a 3-state chain coupled with a 2-state chain) for the single whip gesture. This accords with our intuitions about the single whip, in which one hand does most of the work.



Once appropriate state counts were found, 50 instantiations of each model were trained on 5 randomly selected instances of gesture, and the best (highest-likelihood) models were kept for comparison. We did this because HMMs

[1] Many T'ai Chi forms have mirror-image counterparts.

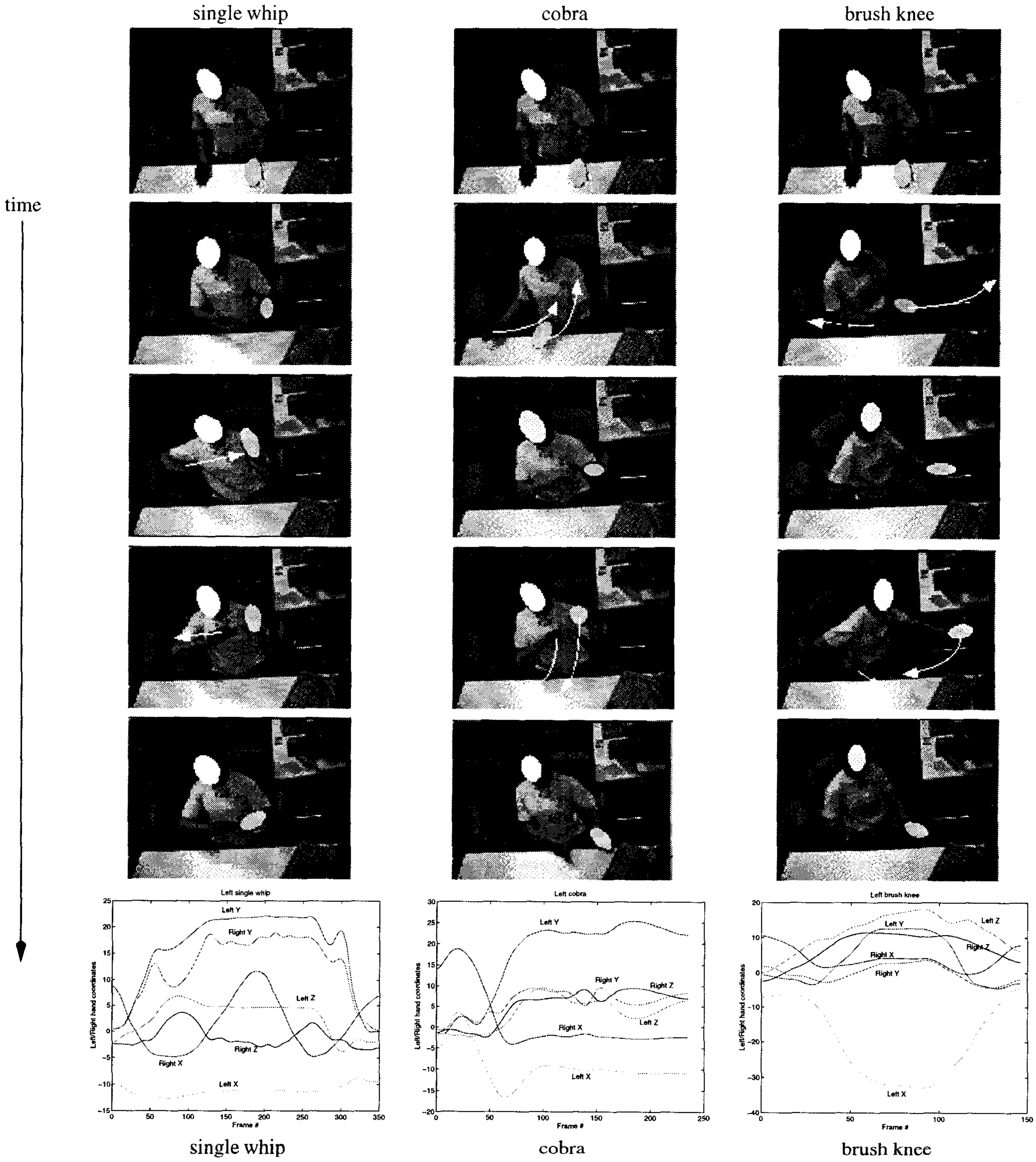


Figure 1. Selected frames from the gestures overlaid with hand blobs from vision. Graphs in the bottom row show the evolution of the feature vector over time. Sequences may be viewed at `http://vismod.www.media.mit.edu/archive`

are known to produce models of varying quality, even when trained repeatedly with the same data. In all cases, the models were set up with a full state-to-state connection topology, so that the training algorithm was responsible for determining an appropriate state structure and sequence for the training data. LHMMs and CHMMs have two output variables, allowing us to split the right-hand and left-hand data streams.

4.3. Results of classification test

To compare the performance of the three previously described architectures in a classification task, we used the trained models to classify the full data set of 52 gestures. The Viterbi algorithm was used to find the maximum likelihood model for HMMs, and a modified Viterbi procedure was used on the joint forms of the LHMMs and CHMMs. Despite use of the joint forms, there were negligible differences in compute times between the three architectures. Two-thirds of the testing data was not been seen in training, including gestures performed at varying speeds and from slightly different views. Figure 2 shows the per-sequence likelihoods for each of the models.

Summing up figure 2, the classification accuracies are:

	Single HMMs	Linked HMMs	Coupled HMMs
accuracy	69.2%	36.5%*	**94.2%**
# params	25+30+180	27+18+54	36+18+54

The bottom row shows the number of degrees of freedom in the largest best-scoring model: state-to-state probabilities + output means + output covariances. The conventional HMM has a large number of covariance parameters because it has a 6-D output variable; the other architectures have two 3-D output variables.

We were surprised by the low accuracy (*) of the LHMM in classifying all the sequences. This is because the LHMM model of the cobra did not not correctly model its temporal structure; having a very low discrimination power, it claimed nearly all sequences. In fact, the LHMM performed significantly better than the HMM on the other two gestures.

We note that Campbell et al. [4] were able train conventional HMMs with $(x_l, y_l, z_l, x_r, y_r, z_r)$ feature vectors to classify 18 different T'ai Chi gestures with accuracies as high as 94%. The HMMs had carefully tuned transition topologies and were each trained on 18 examples of gestures constrained not to have rotational or transitional variation (with variation, rates fell to 34%). Similar circumstances would certainly raise the we obtained.

4.4. Sensitivity analysis

HMMs are notoriously sensitive to the random values assigned to parameters at initialization of training. To test the sensitivity of final model likelihoods to initial conditions, we randomly initialized each architecture and trained it on five examples of a gesture. This was repeated fifty times per gesture and architecture. After training, each model was tested on all examples of its gesture, and we calculated the mean and variance of the resulting posterior probabilities. Fitting Gaussians to these statistics, we obtained the distributions depicted in figure 3, which shows the probability distribution of the per gesture likelihood for coupled, linked and single HMMs.

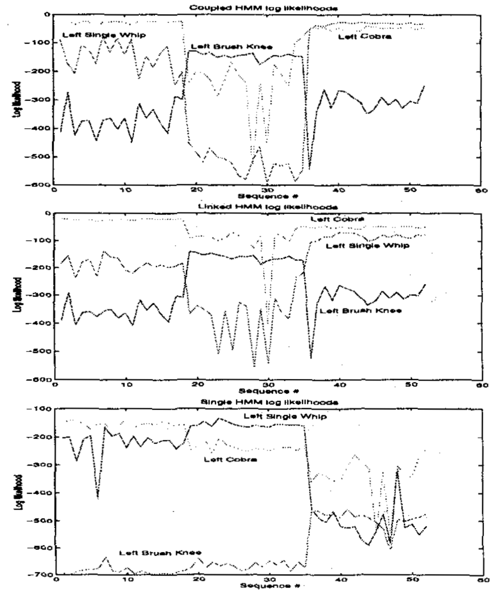


Figure 2. Classification by the CHMM, LHMM, and HMM, showing per-sequence normalized log likelihood. The left third of each graph represent single whip gestures; the middle third represent brush knees; and the right third represent cobras. The curves show the probability of each model on each sampled gesture. As the top graph illustrates, only the CHMM models correctly discriminate the appropriate gestures.

Conventional HMMs were quite sensitive to the initial values of the parameters. LHMMs were generally more robust, depending on the structure of the gesture. Finally, CHMMs were least sensitive to initial conditions and produced the highest likelihood models.

These results also show why the HMMs performed as well as they did in the classification test. In choosing the best-of-50, we took models from the right (optimal) end of the distribution. Had we picked typical models (the mean), the HMMs would have done quite a bit worse than their already mediocre performance.

4.5. Discussion

The CHMMs outperform the other models because the two hands are separate but coordinated processes. To the HMM and LHMM, variations in their coordination can only be modeled as noise; to a CHMM, these variations contain information that can be modeled by the coupling probabil-

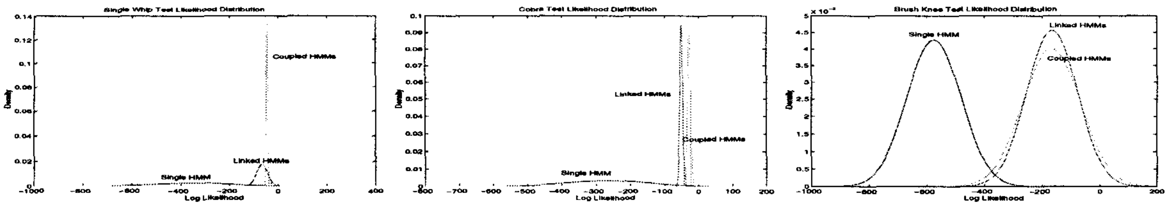


Figure 3. Likelihood probability distribution for each HMM type, learning single whip, cobra, and brush knee gestures, respectively. The CHMM produces the most likely models with a high consistency, indicated by the rightmost distributions.

ities. The dynamic programming algorithms (Viterbi and forward-backward analysis) used in conventional HMMs automatically handle variations in tempo; in CHMMs this extends to variations in process synchronization (up to some point where the system is so out of sync that it is no longer recognizable). It is important to point out that issue here is degree of synchronization of the *underlying states* of the two processes, not of their output signals. Thus CHMMs can work with enormous variation in the signals. This accounts for some of the robustness the CHMMs achieve despite unusually small training samples; it is demonstrated more emphatically in another project where CHMMs are used to identify complex actions in video from the varying spatial relations between hands, tools, and objects [3].

5. Conclusion

Hidden Markov models (HMMs) are used widely in perceptual computing as trainable, time-flexible classifiers of signals that originate from processes like speech and gesture. We believe that a conventional HMM is *not* a good model because most interesting signals fail to satisfy the restrictive Markov condition. Speech recognition researchers have grown increasingly frustrated with the performance of HMMs for this very reason, and vision researchers will run into it even faster. We have presented a mathematical framework for coupled hidden Markov models (CHMMs) which offers a way to model multiple interacting processes without running afoul of the Markov condition. CHMMs couple HMMs with temporal, asymmetric conditional probabilities. To demonstrate their superiority to conventional HMMs, we used a variety of HMM-based architectures to do visual classification of two-handed gestures from T'ai Chi, a martial art. CHMMs yield higher likelihood models with better discriminatory power in fewer epochs *and* these models often runh faster than comparable HMMs in a modified Viterbi algorithm. In addition, CHMMs are far less sensitive to initial conditions than conventional HMMs, e.g., they are more reliable. We also compared CHMMs with linked HMMs (LHMMs), which have atemporal, symmetric joint probabilities between chains. LHMM architectures have been proposed as a desirable higher-order HMM architecture, but experiments show that CHMMs offer a significantly more appropriate model of the conditional independence structure of human gesture.

6. Acknowledgements

Thanks to Andy Wilson for basic Matlab HMM code; Ali Azarbayejani for the self-calibrating stereo hand tracker; Dave Becker for T'ai Chi guidance; and Mike Jordan for illuminating conversations about HMMs.

References

[1] A. Azarbayejani and A. Pentland. Real-time self-calibrating stereo person-tracker using 3-D shape estimation from blob features. In *Proceedings, International Conference on Pattern Recognition*, Vienna, August 1996. IEEE.

[2] M. Brand. Coupled hidden markov models for modeling interacting processes. *Forthcoming (under review)*, November 1996. Also available as MIT Media Lab Vision and Modeling TR #405.

[3] M. Brand. The "Inverse Hollywood Problem": From video to scripts and storyboards via causal analysis. In *Proceedings, AAAI97*, 1997.

[4] L. W. Campbell, D. A. Becker, A. Azarbayjani, A. F. Bobick, and A. Pentland. Invariant features for 3-D gesture recognition. In *Proceedings, International Conference on Automatic Face and Gesture Recognition*, pages 157–162, Killington, VT, 1996. IEEE.

[5] Z. Ghahramani and M. I. Jordan. Factorial hidden Markov models. In D. S. Touretzky, M. C. Mozer, and M. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, Cambridge, MA, 1996. MIT Press.

[6] F. V. Jensen, S. L. Lauritzen, and K. G. Olesen. Bayesian updating in recursive graphical models by local computations. *Computational Statistical Quarterly*, 4:269–282, 190.

[7] M. I. Jordan, Z. Ghahramani, and L. K. Saul. Hidden Markov decision trees. In D. S. Touretzky, M. C. Mozer, and M. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, Cambridge, MA, 1996. MIT Press.

[8] L. K. Saul and M. I. Jordan. Boltzmann chains and hidden Markov models. In G. Tesauro, D. S. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, Cambridge, MA, 1995. MIT Press.

[9] P. Smyth, D. Heckerman, and M. Jordan. Probabilistic independence networks for hidden Markov probability models. AI memo 1565, MIT, Cambridge, MA, Feb. 1996.

[10] C. Williams and G. E. Hinton. Mean field networks that learn to discriminate temporally distorted strings. In *Proceedings, Connectionist models summer school*, pages 18–22, San Mateo, CA, 1990. Morgan Kaufmann.