# A Variational Learning Algorithm for the Abstract Hidden Markov Model

**Jeff Johns*** and **Sridhar Mahadevan**
Computer Science Department
University of Massachusetts
Amherst, MA 01003
{johns, mahadeva}@cs.umass.edu

**Abstract**

We present a fast algorithm for learning the parameters of the abstract hidden Markov model, a type of hierarchical activity recognition model. Learning using exact inference scales poorly as the number of levels in the hierarchy increases; therefore, an approximation is required for large models. We demonstrate that variational inference is well suited to solve this problem. Not only does this technique scale, but it also offers a natural way to leverage the context specific independence properties inherent in the model via the fixed point equations. Experiments confirm that the variational approximation significantly reduces the time necessary for learning while estimating parameter values that can be used to make reliable predictions.

## Introduction

Interest in activity modeling has increased dramatically in recent years. Current applications in this area include visual scene annotation (Torralba *et al.* 2003), motion prediction for assisting the cognitively impaired (Liao, Fox, & Kautz 2004), and behavior classification for robots (Drumwright, Jenkins, & Mataric 2004). The underlying theme in this line of research is learning how to aggregate sequential data in a manner that can be used for making accurate predictions.

Several models have been proposed to perform activity recognition. Coupled hidden Markov models (CHMM) (Brand, Oliver, & Pentland 1996) factorize the state and observation space beyond that of a traditional hidden Markov model (HMM). CHMMs consist of sets of HMMs where each state is dependent on all other states at the previous timestep. Recently, Xiang, Gong, and Parkinson (2003) proposed dynamically multi-linked hidden Markov models (DML-HMM) as an extension to the CHMM. The state space in these models is factorized by using causal relationships to learn the state dependencies. A third activity recognition model is the abstract hidden Markov model (AHMM) (Bui, Venkatesh, & West 2002), which consists of an HMM where the state variables depend on a hierarchy of action variables. Actions become more abstract at higher levels of the hierarchy.

*Supported by the National Science Foundation under contract number REC-0411776.
Copyright © 2005, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

In this paper, we investigate the parameter estimation problem for the AHMM. As with any hierarchical model, exact inference in the AHMM quickly becomes intractable when the hierarchy grows; therefore, an approximate solution is desirable. Patterson *et al.* (2003) addressed this problem by using a Monte Carlo expectation maximization algorithm. There are several issues to address with sampling methods: how to determine convergence, the sample size required to adequately learn the distribution, and how sample size scales with the number of levels in the hierarchy. As an alternative, we present a deterministic, approximate inference algorithm based on variational principles (Jordan *et al.* 1999). Approximate, deterministic methods are often preferable to sampling techniques for large models (Murphy 2002). We show that variational inference can readily exploit the AHMM's context specific independence properties. The variational algorithm is compared with exact inference in terms of time spent per EM iteration and likelihood values achieved upon completion of learning.

The remainder of this paper is organized as follows. The next two sections discuss the general AHMM architecture and why the learning problem becomes more difficult as the hierarchy depth increases. Then, we explain the variational approximation to the AHMM. We finish with a presentation of the experimental results and conclusions.

## AHMM Architecture

The AHMM is a probabilistic model used to explain the interaction between behaviors, or policies, at different levels of abstraction. A policy represents either a mapping from states to a probability distribution over actions or from states to a probability distribution over lower level policies. Policies become more abstract at higher levels of the model hierarchy. From a modeling standpoint, this type of high-level abstraction helps an observer infer intent from low-level observations. Likewise, from a learning standpoint, useful abstractions can significantly reduce the complexity of planning.

We illustrate the AHMM using a dynamic Bayesian network as shown in Figure 1. The bottom portion of the model consists of observations $o_t$ and states $s_t$ in the typical HMM format. Abstract policy variables $\pi_t^m$ along with a corresponding flag variable $e_t^m$ are placed in a hierarchy above the state. The flag indicates whether the current policy is to

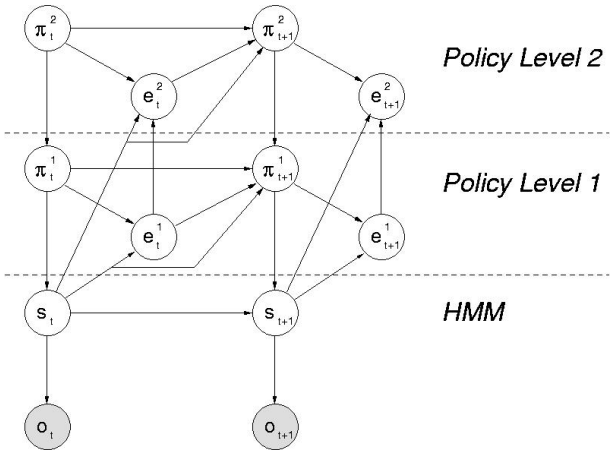


Figure 1: A 2-level AHMM.

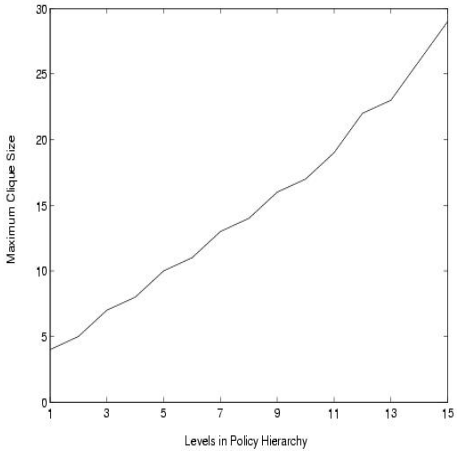


Figure 2: Size of the largest clique during exact inference versus depth of the AHMM hierarchy.

continue or terminate in the next timestep. The joint probability of the AHMM with M policy levels is described in Equation 1 †.

$$P(\{\pi_t^m, s_t, e_t^m, o_t\}) = P(s_1|\pi_1^1)P(o_1|s_1)\prod_{m=1}^{M} P(\pi_1^m|\pi_1^{m+1})P(e_1^m|\pi_1^m, s_1, e_1^{m-1}) \prod_{t=2}^{T} P(s_t|s_{t-1}, \pi_t^1)P(o_t|s_t) \prod_{m=1}^{M} P(\pi_t^m|\pi_{t-1}^m, \pi_t^{m+1}, e_{t-1}^m, s_{t-1})P(e_t^m|\pi_t^m, s_t, e_t^{m-1}) \quad (1)$$

Two context specific independence (CSI) properties (Boutilier *et al.* 1996) exist in the AHMM. First, a policy variable either depends deterministically on the policy at the previous timestep (flag = continue) or it depends on a higher level policy and the previous state (flag = terminate). This relationship is described in Equation 2 where $P_{cont}^{\pi^m}$ is simply an identity matrix of size $|\pi^m| \times |\pi^m|$. Second, a flag variable is constrained to continue if the flag at the level beneath it has not yet terminated (Equation 3). The CSI properties reduce the number of model parameters, a fact that we exploit to accelerate the variational approximation.

$$P(\pi_t^m|\pi_{t-1}^m, \pi_t^{m+1}, e_{t-1}^m, s_{t-1}) = \begin{cases} P(\pi_t^m|\pi_{t-1}^m) \equiv P_{cont}^{\pi^m} & \text{if } e_{t-1}^m = \text{continue} \\ P(\pi_t^m|\pi_t^{m+1}, s_{t-1}) \equiv P_{term}^{\pi^m} & \text{if } e_{t-1}^m = \text{terminate} \end{cases} \quad (2)$$

$$P(e_t^m|\pi_t^m, s_t, e_t^{m-1}) = \begin{cases} P(e_t^m = \text{continue}) = 1 & \text{if } e_t^{m-1} = \text{continue} \\ P(e_t^m|\pi_t^m, s_t) \equiv P_{term}^{e^m} & \text{if } e_t^{m-1} = \text{terminate} \end{cases} \quad (3)$$

†Ignore $\pi_t^{m+1}$ when $m = M$ and ignore $e_t^{m-1}$ when $m = 1$.

## Difficulty of Learning

Learning the parameters of a Bayesian network with hidden variables is typically carried out using the iterative expectation maximization (EM) algorithm (Dempster, Laird, & Rubin 1977). The expected value of the log likelihood is calculated in the E-Step. To do so, one needs to calculate the posterior distribution of the hidden variables given the observations. It is well known that the computational complexity of an exact E-Step using the junction tree algorithm is proportional to the size of the largest clique formed during triangulation of the corresponding Bayesian network. Unfortunately, the AHMM has a treewidth that increases linearly with the number of policy levels in the hierarchy. This results in large cliques which cause the E-Step to become computationally intractable.

To illustrate this problem, consider a generic AHMM where the depth of the hierarchy is varied from 1 level to 15 levels. We assume the number of values a policy node can take on decreases exponentially as you move up the policy hierarchy (i.e. 100 potential policy values at level 1, 10 policy values at level 2, etc . . . ). We then employ the heuristic-based triangulation methodology described in Huang and Darwiche (1994) to determine the maximum clique size for each AHMM. Figure 2 shows empirically that the maximum clique size increases linearly with the number of levels in the hierarchy. If the maximum clique size is 10 and the average number of policy values is 4, there would be greater than one million entries for the potentials in the maximal clique. Thus, exact inference using the junction tree algorithm is not computationally feasible for any nontrivial hierarchies.

To speed up learning, an approximate inference algorithm must be used in the E-Step. The next section discusses the mean field approximation. The concept behind the approach is very simple. Removing edges from the full AHMM avoids the problem of forming large cliques.

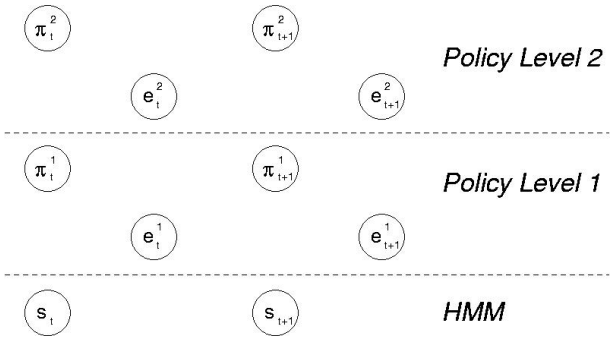


Figure 3: Mean field variational approximation of the 2-level AHMM.

## Variational Approximation

### Model

We consider a completely factorized variational approximation to the AHMM (Figure 3). The joint distribution for this model is shown in Equation 4. This mean field approximation is inspired by Ghahramani and Jordan's (1997) approximation of the factorial HMM.

$$Q(\{\pi_t^m, s_t, e_t^m\}|\{h_t^{\pi^m}, h_t^{e^m}, h_t^s\}) = \prod_{t=1}^{T} Q(s_t|h_t^s) \prod_{m=1}^{M} Q(\pi_t^m|h_t^{\pi^m})\, Q(e_t^m|h_t^{e^m}) \tag{4}$$

By assuming all variables are independent, the only parameters for this distribution are the variational parameters. The variational parameters $\{h_t^{\pi^m}\}$, $\{h_t^{e^m}\}$, and $\{h_t^s\}$ are the means of the policy, flag, and state variables respectively. As such, each parameter determines the probability of the variable taking on each of its possible values.

### Inference

The variational approximation is used to form a lower bound on the log likelihood of the data under the full AHMM. This is proven using Jensen's inequality.

$$\log P(\{o_t\}) = \log \sum_{\{\pi_t^m, e_t^m, s_t\}} P(\{\pi_t^m, e_t^m, s_t, o_t\})$$

$$\geq \sum_{\{\pi_t^m, e_t^m, s_t\}} Q(\{\pi_t^m, e_t^m, s_t\}) \log \frac{P(\{\pi_t^m, e_t^m, s_t\}|\{o_t\})}{Q(\{\pi_t^m, e_t^m, s_t\})}$$

The Kullback-Liebler divergence (Cover & Thomas 1991) is the difference between the two sides of the above equation.

$$KL(Q||P) = \sum_{\{\pi_t^m, e_t^m, s_t\}} Q(\{\pi_t^m, e_t^m, s_t\}) \log \frac{Q(\{\pi_t^m, e_t^m, s_t\})}{P(\{\pi_t^m, e_t^m, s_t\}|\{o_t\})}$$

By updating the variational parameters, we can minimize the KL divergence between the full posterior distribution and the variational distribution in order to attain the tightest lower bound on the log likelihood of the data. The updates are done by taking the derivative of the KL divergence with respect to the variational parameters, setting the result to zero, and solving for the variational parameters. This yields the fixed point equations.

As an example, we show the fixed point equation for $h_t^{\pi^m}$ for $1 < t < T$ and $1 < m < M$ in Equation 5. We abuse the notation in this equation for the sake of readability. The variational parameters are vectors and the transition models are multi-dimensional matrices. Rather than using summations to index into each vector, we leave the equations in a vectorized format with the understanding that the vectors need to be handled appropriately. Also, the symbol $\varphi$ represents the softmax operator which ensures the variational parameter vector sums to one.

$$h_t^{\pi^m} = \varphi \left\{ \begin{array}{l} h_{t,term}^{e^{m-1}} \cdot h_t^s \cdot h_t^{e^m} \cdot \log(P_{term}^{e^m}) \\ + \; h_{t-1,cont}^{e^m} \cdot h_{t-1}^{\pi^m} \cdot \log(P_{cont}^{\pi^m}) \\ + \; h_{t,cont}^{e^m} \cdot h_{t+1}^{\pi^m} \cdot \log(P_{cont}^{\pi^m}) \\ + \; h_{t-1,term}^{e^m} \cdot h_t^{\pi^{m+1}} \cdot h_{t-1}^s \cdot \log(P_{term}^{\pi^m}) \\ + \; h_{t-1,term}^{e^{m-1}} \cdot h_t^{\pi^{m-1}} \cdot h_{t-1}^s \cdot \log(P_{term}^{\pi^{m-1}}) \end{array} \right\} \tag{5}$$

Inspection of the fixed point equations provides intuition as to the role of the variational parameters. The terms involved in each equation are contained in the Markov blanket of the variable being updated. Thus, each equation provides approximate sufficient statistics that are used to determine the mean of the variable.

The fixed point equations are simplified by exploiting the AHMM's CSI properties. For example, had the CSI properties not been taken into account, there would be roughly double the number of terms in Equation 5. These simplifications exploit the nice properties of the AHMM and result in even faster approximate inference. Our work shows that CSI properties can be exploited in a very natural and general way by variational methods.

The fixed point equations are updated in an iterative fashion during the E-Step of the EM algorithm. The E-Step is completed when the KL divergence converges to a minimum, which is theoretically guaranteed. Each iteration takes time $O(TMNP)$ where T is the number of timesteps, M is the number of policy levels, N is the number of hidden states, and P is the maximum number of policy values. We found that the number of iterations until the KL divergence converged depends on the length of the sequences but typically took less than 15 iterations. Therefore, in contrast to exact methods, learning using variational inference techniques scales well with the number of levels in the hierarchy.

We summarize these equations in pseudocode (Algorithm 1) for a variational E-Step. The expectations from the E-Step are then used to maximize the complete data likelihood in the M-Step. An exact M-Step for the AHMM is tractable.

**Algorithm 1** Variational E-Step

**repeat**
  *// Update Variational Parameters*
  **for** $t = 1$ to $T$ **do**
    **for** $m = 1$ to $M$ **do**
      $h_t^{\pi^m} \Leftarrow$ result of Equation 5
      *Update* $h_t^{e^m}$ *via fixed point equation*
    **end for**
    *Update* $h_t^s$ *via fixed point equation*
  **end for**
  *// Calculate KL Divergence*
**until** KL Divergence converges

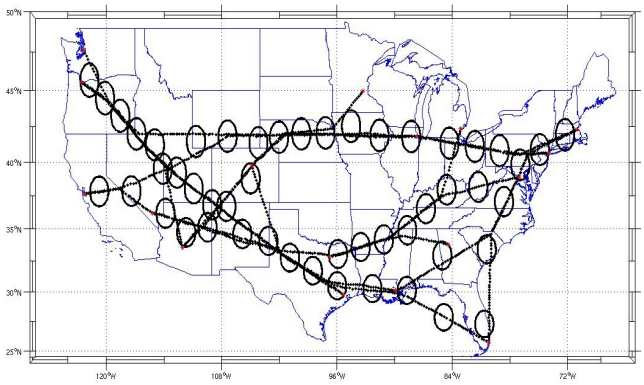


Figure 4: Airline domain with 15 different routes. The ovals correspond to Gaussian covariance matrices for the state variables.

## Experiments

Experiments were run on three domains to compare learning using the variational approximation with learning using exact inference under the junction tree algorithm. Two activity recognition datasets were used (Osentoski, Manfredi, & Mahadevan 2004). In these experiments, a robot converted its raw laser readings into x-y positions of a person walking. The experiments were conducted in two separate domains: a small laboratory/cubicle environment and an entryway inside the University of Massachusetts Computer Science Department. There were six different trajectories in the laboratory environment and eight in the entryway. The length of a typical sequence in these domains was approximately 20 timesteps. To test the variational technique in a larger domain, a third experiment was conducted using synthetic data. The dataset contained sequences of latitude-longitude readings intended to represent airline routes. There were fifteen different routes in this domain where the longest sequence was approximately 250 timesteps. A picture of the state space with datapoints for all fifteen routes is shown in Figure 4. As the figure indicates, the routes were constrained to follow corridors through the United States. This constraint ensured different trajectories would have overlapping regions of state space. For all experiments, observations are Gaussian with a tied covariance matrix while states, policies, and flags are multinomial variables.

Each experiment was run using a 1-level and a 2-level AHMM. An experiment was further divided into the case where the policy variable at the top of the hierarchy was either observed or unobserved. Note that all other variables besides the observations are hidden.

The two metrics used to compare the variational approximation with exact inference are the log likelihood (both for training and test sets) and the time per EM iteration. Furthermore, in the experiments where the top level policy variable is observed during learning, we present prediction accuracies for the test data. Note that the log likelihood was calculated using exact techniques to ensure a consistent basis for comparing results.

### Unobserved Case for Top Level Policy Variables

When all variables other than the observation are hidden, the learning algorithm clusters sequences using the policy variables. The effectiveness of the clustering can be measured by the likelihood of the data under the learned model and by a visual inspection of the clusters. We will discuss both of these measures in this section.

The log likelihood values for the three domains are presented in Table 1. All values reported in the table are averages for 20 experiments. As expected, the log likelihoods achieved using exact inference are better than those attained using the variational approximation. This improvement is statistically significant. However, the usefulness of the model trained using approximate methods is not solely a function of the log likelihood. The AHMM's function is to model activity and distinguish different sequences. Therefore, we examined the marginal probabilities on the top-level policy variables to inspect what the model learns.

There are three interesting points to make about the marginals. First, the marginal probabilities on the top-level policy variable tend to remain constant through time until the sequence overlaps in state space with a different sequence. Second, when sequences overlap, they tend to use the same policy. This ability for different sequences to reuse the same policy in similar states is an important quality of the AHMM. In effect, a reused policy is a macro that any sequence can invoke when in the appropriate state. Third, when approximate inference was used during learning, the marginals typically assign more weight to one policy value. In constrast, when using exact inference during learning, the marginals tend to be more uniformally distributed over all possible policy values.

Figure 5 highlights the first two of these points regarding the marginals. This figure from the airline domain shows the last half of two different sequences: *Chicago-Portland* and *Houston-Seattle*. The maximum policy value for $\pi_t^1$ is shown for a 1-level AHMM model when trained using the variational algorithm. The two sequences follow separate policies, $\pi_t^1 = 1$ and $\pi_t^1 = 2$ respectively, until they converge. Then, both sequences use the same policy, $\pi_t^1 = 3$, until reaching the final destinations. In this example, almost all marginals had a belief greater than $0.9$ on the maximum policy value.

The running time for the variational learning algorithm is significantly better than the exact algorithm's running time.

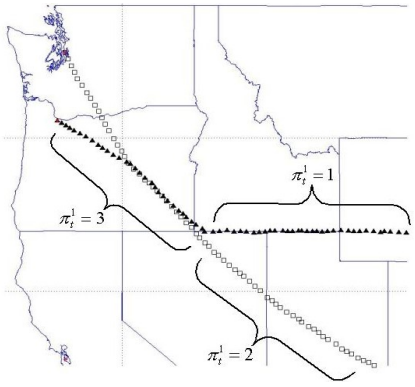


Figure 5: Two different trajectories for the airline domain for a 1-level AHMM trained using the variational algorithm. The most likely policy value is shown. Policy 3 is used by both sequences when the two routes converge.

The last column in Table 1 shows the time between EM iterations. Averaging across the three domains, the variational algorithm proved to be 17% faster than the exact algorithm for 1-level AHMMs. For 2-level AHMMs, the improvement jumped to 64%. The exponential growth in complexity using exact inference quickly becomes apparent when going from a 1-level to a 2-level AHMM. We note that the percent improvement in running time is based on a relatively naïve implementation of the variational EM algorithm. A more optimized version would achieve larger gains.

### Observed Case for Top Level Policy Variables

The same set of experiments was conducted but with the top-level policy variable ($\pi_t^1$ for 1-level AHMMs and $\pi_t^2$ for 2-level AHMMs) observed while learning the model parameters. The purpose of these experiments was to test the variational learning algorithm on a classification task as opposed to an unsupervised clustering problem. After training the model, the test dataset was used to make predictions about the top-level policy variable. This prediction was compared to the known label to generate a classification accuracy. The test accuracy and log likelihoods are shown in Table 2.

The classification accuracy was nearly 100% in all cases. One sequence in the entryway dataset was misclassified for the 2-level AHMM trained using the variational algorithm. This error is somewhat mitigated in that the mistaken label corresponded to a similar sequence. In the airline domain, the classification accuracy was 95% for all sets of experiments. The classification error in this domain occurred for two trajectories (*Houston-Seattle* and *Houston-Portland*) that consisted of identical state sequences. Notice that the Gaussian state in the upper-left portion of Figure 4 is unable to capture the difference in the final two destinations; therefore, the two sequences are indistinguishable from one another. These two trajectories were purposefully chosen to verify this effect given our choice of 50 hidden states to model the data. Overall, the classification accuracies indicate the variational algorithm can effectively learn the model parameters in a classification task.

(a) Lab Dataset

| Inference Method | Policy Levels | Training Log Lik. | Testing Log Lik. | Time (s) |
|---|---|---|---|---|
| Exact | 1 | -58 | -103 | 8.7 |
| Approx. | 1 | -103 | -145 | 6.8 |
| Exact | 2 | -43 | -84 | 18 |
| Approx. | 2 | -75 | -113 | 8.2 |

(b) Entryway Dataset

| Inference Method | Policy Levels | Training Log Lik. | Testing Log Lik. | Time (s) |
|---|---|---|---|---|
| Exact | 1 | -2090 | -2038 | 65 |
| Approx. | 1 | -2377 | -2342 | 52 |
| Exact | 2 | -1650 | -1616 | 284 |
| Approx. | 2 | -2014 | -1935 | 71 |

(c) Airline Dataset

| Inference Method | Policy Levels | Training Log Lik. | Testing Log Lik. | Time (s) |
|---|---|---|---|---|
| Exact | 1 | -21598 | -21685 | 745 |
| Approx. | 1 | -22392 | -22487 | 680 |
| Exact | 2 | -18392 | -18546 | 2443 |
| Approx. | 2 | -19344 | -19634 | 934 |

Table 1: Log likelihood on the training and testing datasets and time in seconds per EM iteration for exact inference and variational inference. All variables other than observations are hidden. Results are averaged over 20 trials.

The analysis of the log likelihood values is similar to the completely unobserved case. Training using exact inference produces statistically better likelihoods compared to training using the variational approximation. However, it is interesting that the 2-level AHMMs trained using approximate inference yield better likelihoods for the unobserved case than for the observed case. One possible reason for this anomaly is that more useful intermediate levels of abstraction are formed based on bottom-up learning (i.e. from the states in the AHMM to sequentially higher policy levels) than from a mixture of bottom-up and top-down (when the top-level policy is observed) learning. Evidence for this phenomena exists for the hidden Markov decision tree (HMDT) (Jordan, Ghahramani, & Saul 1997). The learning curves for training the HMDT exhibit separate ramps in the log likelihood value corresponding to learning different levels of abstraction in the decision tree.

## Conclusions

We presented a deterministic, approximate inference algorithm used to learn the parameters of an abstract hidden Markov model. A completely factorized distribution was used to form a lower bound on the likelihood of the full AHMM. The Kullback-Liebler divergence was minimized in an iterative E-Step to achieve the tightest possible lower bound given our choice of approximating distribution. In this paper, we showed that the algorithm significantly speeds up parameter estimation as the number of levels in the

(a) Lab Dataset

| Inference Method | Policy Levels | Training Log Lik. | Testing Log Lik. | Testing Accuracy |
|---|---|---|---|---|
| Exact | 1 | -50 | -100 | 100% |
| Approx. | 1 | -56 | -101 | 100% |
| Exact | 2 | -30 | -78 | 100% |
| Approx. | 2 | -93 | -148 | 100% |

(b) Entryway Dataset

| Inference Method | Policy Levels | Training Log Lik. | Testing Log Lik. | Testing Accuracy |
|---|---|---|---|---|
| Exact | 1 | -2047 | -2008 | 100% |
| Approx. | 1 | -2252 | -2200 | 100% |
| Exact | 2 | -1760 | -1692 | 100% |
| Approx. | 2 | -2170 | -1956 | 97% |

(c) Airline Dataset

| Inference Method | Policy Levels | Training Log Lik. | Testing Log Lik. | Testing Accuracy |
|---|---|---|---|---|
| Exact | 1 | -21766 | -21812 | 95% |
| Approx. | 1 | -21750 | -21783 | 95% |
| Exact | 2 | -19134 | -19023 | 95% |
| Approx. | 2 | -21455 | -21655 | 95% |

Table 2: Log likelihood on the training and testing datasets and accuracy on predicting the top-level policy value on the testing dataset. The top-level policy is observed during training. Results are averaged over 20 trials.

AHMM's hierarchy increases. More specifically, the running time scales linearly with the number of levels as opposed to exponentially using exact inference. Furthermore, we showed that the variational algorithm easily exploits the context specific independence properties of the AHMM. The CSI properties simplify the model and result in computationally efficient fixed point equations.

Learning using the variational algorithm produced models that were effective for performing activity recognition. The algorithm was shown to work for both unsupervised and supervised problems. In the unsupervised task, the algorithm clustered sequences using the policy variables. Interestingly, different sequences learned to use the same policy when in the same state. This type of clustering allows the model to make accurate predictions on new, unseen sequences. The prediction accuracy on test datasets was nearly identical whether using exact inference or approximate inference during parameter estimation.

Exact inference methods produced better likelihoods than the variational approximation. This improvement is a result of the mean field assumption. Completely factorizing the model makes it more challenging to capture all the dependencies among the variables. In the future, we plan to investigate how much a structured approximation improves the extent of learning. A tighter bound on the likelihood of the data under the full AHMM can be formed by keeping some of the temporal links in the model while leaving inference tractable. An interesting question is what type of policies can be learned using different approximations. We also plan to test the variational algorithm on larger, multidimensional datasets and compare the algorithm with a sampling method.

## References


Boutilier, C.; Friedman, N.; Goldszmidt, M.; and Koller, D. 1996. Context-specific Independence in Bayesian Networks. *In Proceedings of Twelfth Conference on Uncertainty in Artificial Intelligence* 115–123.

Brand, M.; Oliver, N.; and Pentland, A. 1996. Coupled Hidden Markov Models for Complex Action Recognition. *In IEEE CVPR97* 994–999.

Bui, H.; Venkatesh, S.; and West, G. 2002. Policy Recognition in the Abstract Hidden Markov Model. *Journal of Artificial Intelligence Research* 17:451–499.

Cover, T., and Thomas, J. 1991. *Elements of Information Theory*. New York, USA: John Wiley and Sons.

Dempster, A.; Laird, N.; and Rubin, D. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of Royal Statistical Society Series B* 39:1–38.

Drumwright, E.; Jenkins, O.; and Mataric, M. 2004. Exemplar-based Primitives for Humanoid Movement Classification and Control. *IEEE Conference on Robotics and Automation* 140–145.

Ghahramani, Z., and Jordan, M. 1997. Factorial Hidden Markov Models. *Machine Learning* 29:245–275.

Huang, C., and Darwiche, A. 1994. Inference in Belief Networks: A Procedural Guide. *International Journal of Approximate Reasoning* 11(1):393–405.

Jordan, M. I.; Ghahramani, Z.; Jaakkola, T.; and Saul, L. K. 1999. An Introduction to Variational Methods for Graphical Models. *Machine Learning* 37(2):183–233.

Jordan, M.; Ghahramani, Z.; and Saul, L. 1997. Hidden Markov Decision Trees. *In Proceedings of Advances in Neural Information Processing Systems* 9.

Liao, L.; Fox, D.; and Kautz, H. 2004. Learning and Inferring Transportation Routines. *In Proceedings of AAAI-04*.

Murphy, K. 2002. *Dynamic Bayesian Networks: Representation, Inference and Learning*. Ph.D. Dissertation, University of California, Berkeley.

Osentoski, S.; Manfredi, V.; and Mahadevan, S. 2004. Learning Hierarchical Models of Activity. *In Proceedings of the International Conference on Intelligent Robots and Systems*.

Patterson, D.; Liao, L.; Fox, D.; and Kautz, H. 2003. Inferring High-Level Behavior from Low-Level Sensors. *International Conference on Ubiquitous Computing* 73–89.

Torralba, A.; Murphy, K.; Freeman, W.; and Rubin, M. 2003. Context-based Vision System for Place and Object Recognition. *International Conference on Computer Vision* 1:273–280.

Xiang, T.; Gong, S.; and Parkinson, D. 2003. Outdoor Activity Recognition using Multi-Linked Temporal Processes. *In Proceedings of British Machine Vision Conference* 619–628.