Monte Carlo HMMs and POMDPs

Sebastian Thrun, John Langford, and Dieter Fox

Problem: Learning models of unknown environments is a highly important problem in robotics and many other disciplines concerned with processing sensor data (e.g., speech). Optimal decision making using such learned models is also a key scientific problem of not lesser importance. In the most general (and important) case, environments are stochastic, dynamic, and only partially observable. They typically possess real-valued state and observation spaces, for which no parametric models are available. Decision making is often subject to close time bounds; the world never waits for the termination of a computer program. The problem attacked in this project is: How can we develop more capable learning and decision making tools for real-world tasks that meet these requirements?

Impact: This project will provide non-parametric methods for learning Hidden Markov models (HMMs) of partially observable stochastic environments. It will also provide improved any-time decision tools for partially observable Markov decision processes (POMDP) that guarantee decisions under arbitrary deadlines. In particular, our goal is to extend existing learning algorithms for HMMs/POMDPs to real-valued state and observation spaces, using non-parametric methods for learning models, and any-time methods for state estimation and decision making in POMDPs. These new methods will be better suited for a broad range of real-world problems, in that they yield improve modeling capabilities and provide control in a time-critical manner. The results should be applicable to a broad range of robotics problems (e.g., learning maps, exploration, active perception).

State of the Art: Most conventional HMM and POMDPs rely on discrete state spaces and, in many cases, discrete observation spaces. Existing learning algorithms are *parametric*, requiring significant a priori knowledge concerning the nature of the environment. The issue of *any-time* state estimation and decision making is still poorly understood, despite a significant (and insightful) attempts in the AI literature [1, 5].

Approach: We have recently extended HMMs, using Monte Carlo sampling methods. In our approach, which is called Monte Carlo HMMs (MCHMMs) [3, 4], all necessary probability densities are approximated using samples (Figure 1a), along with density trees generated from such samples (Figure 1b). A Monte Carlo version of Baum-Welch (EM) is employed to learn models from data, just as in regular HMM learning. Regularization during learning is achieved using an exponential shrinking technique, called shrinkage trees (Figure 1c). The shrinkage factor, which determines the effective capacity of the learning algorithm, is annealed down over multiple iterations of Baum-Welch, and early stopping is applied to select the right model. We have proven that under mild assumptions, Monte Carlo Hidden Markov Models converge to a local maximum in likelihood space, just like conventional HMMs. In addition, we have already obtained promising empirical results in a gesture recognition domain.

Future Work: We are currently in the process of refining this initial model, and extending it to POMDPs [2]. In POMDPs, state transitions are triggered by actions, and finding the right thing to do is a key concern in POMDPs. We believe the application of Monte Carlo sampling to POMDPs is straightforward. The decision making problem in POMDPs is typically solved using dynamic programming, which assigns values to belief states (a belief state is a probability distribution over states). Since belief states in POMDPs are then represented by entire sample sets, new function approximators are needed to represent the value function. We are currently extending nearest neighbor methods to deal with belief states represented by sample sets, using KL divergence as a distance function. Empirical results are forthcoming. We are also interested in the

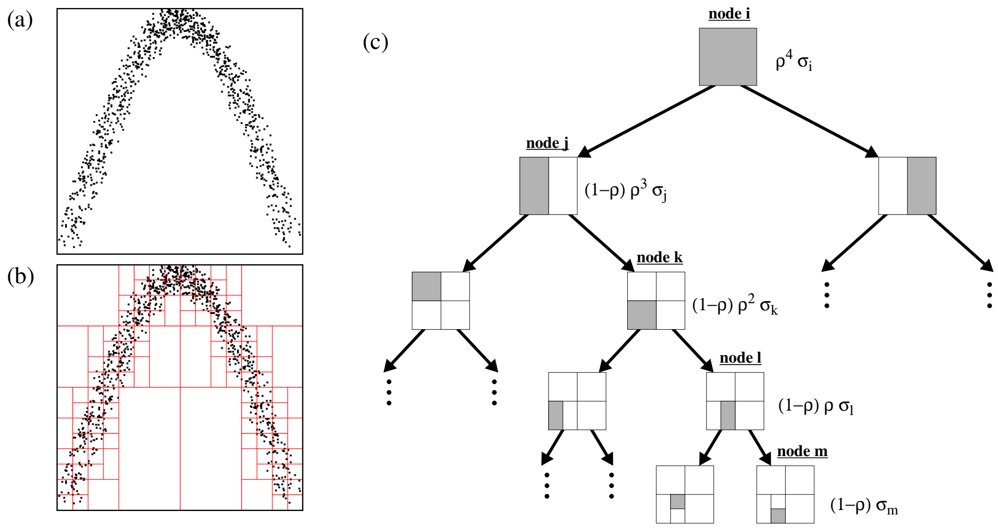

Figure 1: (a) Example data set (b) partitioned by a density tree. (c) Density tree with shrinkage for complexity control. Each branch cuts the area under a node into two equally-sized, rectangular halves, shown in grey. The density of each node is an exponentially weighted mixture of densities of the nodes along the path to the leaf. The shrinkage factor is annealed down over time, and early stopping is used for model selection (complexity control).

---

convergence properties of our new approach.

Selected References:

[1] T. L. Dean and M. Boddy. An analysis of time-dependent planning. In *Proceeding of Seventh National Conference on Artificial Intelligence AAAI-92*, pages 49–54, Menlo Park, CA, 1988. AAAI, AAAI Press/The MIT Press.

[2] M.L. Littman, A.R. Cassandra, and L.P. Kaelbling. Learning policies for partially observable environments: Scaling up. In A. Prieditis and S. Russell, editors, *Proceedings of the Twelfth International Conference on Machine Learning*, 1995.

[3] S. Thrun and J. Langford. Monte carlo hidden markov models. Technical Report CMU-CS-98-179, Carnegie Mellon University, Computer Science Department, Pittsburgh, PA, 1997.

[4] S. Thrun and J. Langford. Sample-based hidden markov models. Submitted for publication., 1999.

[5] S. Zilberstein and S. Russell. Approximate reasoning using anytime algorithms. In S. Natarajan, editor, *Imprecise and Approximate Computation*. Kluwer Academic Publishers, Dordrecht, 1995.