Issues in Hidden Markov Modelling and Task Driven Perception

Adrian Hartley and Jeremy Wyatt
School of Computer Science
University of Birmingham
Birmingham, UK, B15 2TT
{arh,jlw}@cs.bham.ac.uk

Abstract

The starting point of this position paper is the observation that robot learning of tasks, when done autonomously, can be conveniently divided into three learning problems. In the first we must derive a controller for a task given a process model. In the second we must derive such a process model, perhaps in the face of hidden state. In the third we search for perceptual processing functions that find natural regularities in the robot's sensory sequence, and which have utility in so far as they ease the construction of process models. We sketch an algorithm which takes a stochastic process approach to modelling, and combines methods for solving each of these three problems.

Introduction

One of the great challenges of intelligent robotics is to find a class of algorithms which can quickly learn near-optimal behaviours for reasonably complex tasks, given the typical problems of hidden state, and unreliable sensing and action. We are working on this problem within a stochastic process framework; tasks are viewed as they are in reinforcement learning (Sutton & Barto 1998); and learning is model-based. We are concerned here with speeding up learning on processes that have hidden state. Our approach is to combine data driven learning of perceptual categories (Oates, Schmill, & Cohen. 1999) with task driven hidden Markov modelling (McCallum 1996). We hypothesise that this will lead to faster and improved process modelling. Once process modelling is complete a variety of methods can be applied to the model to find a policy which optimises a reward function used to measure performance on the task. In the subsequent sections we review the problems involved in process modelling; sketch the algorithm we are currently implementing, and discuss some of the problems and issues that we expect to arise from our approach.

Issues in stochastic process modelling

In reinforcement learning (RL) the problem of learning a policy for a given task is defined as learning a policy which maximises the expected return for each state that the system can occupy. If the process can be modelled as a finite state Markov decision process (MDP) then the trivial solution is to learn an estimated first order Markov model and compute the value function given the reward function. It is also possible to learn a compact Markov model using a graphical model (Andre, Friedman, & Parr 1998). This sort of approach is useful where there are a number of qualitative variates, and table look-up approaches are inadequate. There are also a variety of techniques, both model-based and model-free for learning policies for Markov processes defined on continuous state and action spaces(Moore 1991; Reynolds 2000; Munos & Moore 1999). Thus while improvements to these techniques are still being made, the problem of finding an optimal policy given a reward function for the Markov case can be regarded as essentially solved.

If the process contains hidden state then the problem of learning a process model is rather harder. Techniques based on the Baum-Welch algorithm can be applied to sequences of observations drawn either from discrete or continuous spaces. Basic HMM techniques require that the number of states in the model be known in advance. In addition they are only guaranteed to converge to a local maximum likelihood model, though techniques that employ a set of randomly generated initial models can give improved results. However, methods such as Lonnie Chrisman's perceptual distinctions algorithm (Chrisman 1992) and Andrew McCallum's Utile Distinctions (UDM) algorithm allow the refinement of the number of model states. From a reinforcement learning perspective McCallum's (McCallum 1996) approach is particularly appealing since it recognises that the model required need only provide accurate predictions of the utility of state-action pairs. We refer to such techniques as task driven HMM algorithms. However, such algorithms have weaknesses. They require long sequences of data and may not detect n^{th} order hidden state unless there is an improvement in predictive accuracy for each intermediate order over the previous one. Directly searching the space of possible models with hidden state up to order n for a good starting model is clearly computationally intractable.

Combining model learning and perceptual learning

Our approach is to tackle this problem by learning perceptual functions of sensory time series that capture their nat-

ural regularities. Our intention is to learn such categories and use class membership as a observation token representing the sensory sequence to feed into a task driven hidden Markov modelling process. To illustrate the approach we are using Dynamic Time Warping (DTW), together with agglomerative clustering, in the manner of Cohen and Oates (Oates, Schmill, & Cohen. 1999), to generate the classes and tokens. The HMM algorithm we use is McCallum's UDM algorithm and the two techniques are combined as shown in Figure 1.

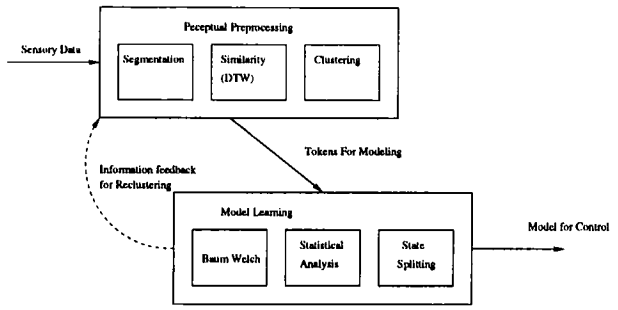

Figure 1: The structure of the algorithm

The upper row of processes represent the sensory learning algorithm and consist of the following phases:

- Segmentation: Firstly the time series of experiences or raw observation values needs to be partitioned into useable sub-sequences. The reward and action sequences associated with the sensory sequence also needs to be segmented. We discuss the issues at length in the next section.
- Similarity: the similarity for each pair of sensory segments is calculated. Following Cohen and Oates we intend to use DTW to generate these distances.
- Clustering: Using the similarity measurements from DTW clustering can be performed. The clusters are then the observations employed by the HMM algorithm. Cluster prototypes would be used for recognition of cluster membership.

The lower row of processes are essentially those of McCallum's UDM algorithm.

Segmentation of time series

The major issue we are currently considering is the problem of how to segment the time series. If we choose a short time period then we may well end up with a few clusters with many instances. Each observation token will then contain little state, and so will be too general to speed our HMM. If we choose a long time period then we will end up with more clusters each of which is more specific (assuming that we have enough sensory data). In the extreme case this will cause us to have a model with many more states than necessary to satisfy the Markov assumption. One sensible definition of an ideal model on this view is that it should satisfy the Markov criterion with as few states as possible.

The selection of the initial segmentation of the time series will therefore be important in determining the efficiency of the modelling process. If we do not know the initial time scale on which to segment then we must essentially either guess or segment on many time scales at once. We speculate on the likely consequences of this choice below.

If we segment on a single scale that is too short then we will end up with a few clusters, and thus with a model with few states. We will then be required to create additional states in the model according to McCallum's state splitting criterion. This is equivalent to sequencing together the time series for two clusters (i.e. sequencing two prototypes). One consequence of this is that we must rerun DTW on the re-segmented time series and then recluster. This is one form of feedback from the model learning algorithm to the perceptual learning algorithm.

If we segment on a scale that is too long then providing the clustering is appropriate we should be able to generate a hidden Markov model directly without creating new model states. However we may find that we can then discard model states and still retain a Markov model. To do this we need a criterion for merging states. We do not currently have a clearly defined criterion but believe that the sort of scheme we should try involves merging states that have very similar transition probabilities (Dean, Givan, & Leach 1997). In merging them we can then seek a re-segmentation of those sections of time series on a finer time scale.

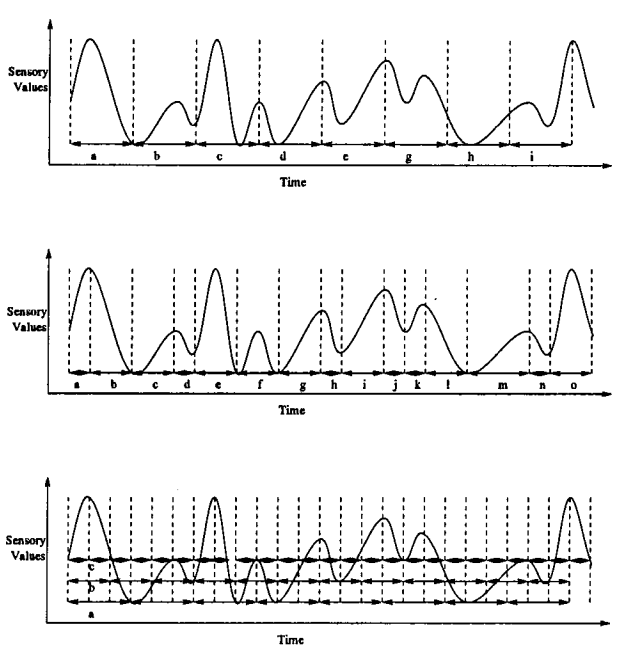

Figure 2: Different schemes for segmentation. (Top) A single segmentation on a single time scale. (Middle) A single segmentation on many time scales. (Bottom) Three segmentations on three time scales, labelled a, b and c.

The process of revising the number of model states; leading to resegmentation, rewarping, reclustering; and finally to reoptimisation of the HMM; is clearly a complex process.

Since we are creating states by finding the natural regularities in the input time series there are no guarantees that we will obtain a better model at each stage. This issue is further complicated by the fact that we are only guaranteed to find a model which is locally a maximum likelihood model. We may decide to reject the results of a resegmentation on these grounds. In addition the results of many resegmentations will be that we end up with a time series segmented on different time scales at different places.

Of course there may be a number of useful heuristics which we can apply to obtain an initial segmentation. We suggest three:

- Firoiu and Cohen's (Firoiu & Cohen 1999) best linear piece-wise fit criterion (which segments on second order changes).
- Segmentation when there is a reward event.
- Segmentation when there is an action or behaviour change.

Each of these would lead to a segmentation of the form shown in the middle panel of Figure 2. A more radical suggestion is that we should actually produce several segmentations at once, each on a different timescale (bottom panel Figure 2). Each time scale could be warped and clustered separately or together. We might then need a very different algorithm for selecting hidden Markov models, and possibly for optimising them, which is capable of locally choosing the best scale and prototype with which to model at each level. We currently have done no work on what the computational issues for such an approach might be. The interactions between multiple time scales for time series segmentation and the notion of multiple time scales in the semi-Markov models used by Sutton et al. (Sutton, Precup, & Singh 1999) to model options is particularly interesting.

Conclusion

We have outlined a potential scheme for boosting the speed and power of task driven HMM algorithms. In the process we have suggested a criterion for perceptual learning which is itself task driven. Perceptual categories are influenced by feedback from the modelling process. Perceptual categories which support reliable predictions with respect to some task are retained, while non reliable categories are modified. The algorithm we have sketched is currently being refined and implemented, and we hope to be able to present initial results, together with a complete specification of the algorithm at the symposium itself.

References

Andre, D.; Friedman, N.; and Parr, R. 1998. Generalized prioritized sweeping. In Jordan, M. I.; Kearns, M. J.; and Solla, S. A., eds., *Advances in Neural Information Processing Systems*, volume 10. The MIT Press.

Chrisman, L. 1992. Reinforcement learning with perceptual aliasing: The perceptual distinctions approach. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, 183–188.

Dean, T.; Givan, R.; and Leach, S. 1997. Model reduction techniques for computing approximately optimal solutions for Markov decision processes. In Geiger, D., and Shenoy, P. P., eds., *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence (UAI-97)*, 124–131. San Francisco: Morgan Kaufmann Publishers.

Firoiu, L., and Cohen, P. R. 1999. Abstracting from robot sensor data using hidden Markov models. In *Proceedings of the Sixteenth International Conference on Machine Learning*, 106–114. Morgan Kaufmann.

McCallum, A. K. 1996. *Reinforcement Learning with Selective Perception and Hidden State*. Ph.D. Dissertation, University of Rochester.

Moore, A. 1991. Variable resolution dynamic programming: Efficiently learning action maps in multivariate real-valued state-spaces. In Birnbaum, L., and Collins, G., eds., *Machine Learning: Proceedings of the Eighth International Workshop*, 333–337. Morgan Kaufmann.

Munos, R., and Moore, A. 1999. Variable resolution discretization in optimal control. *Machine Learning Journal*.

Oates, T.; Schmill, M. D.; and Cohen., P. R. 1999. Identifying qualitatively different outcomes of actions: Experiments with a mobile robot. In *Proceedings of The Sixteenth International Joint Conference on Artificial Intelligence , Workshop on Robot Action Planning*.

Reynolds, S. I. 2000. Decision boundary partitioning: Variable resolution model-free reinforcement learning. In *Proceedings of the 17th International Conference on Machine Learning (ICML-2000)*. San Fransisco: Morgan Kaufmann.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement Learning: An Introduction*. MIT Press/Bradford Books.

Sutton, R. S.; Precup, D.; and Singh, S. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence* 112:181–211.