Inducing hybrid models of task learning from visualmotor data

Devika Subramanian
devika@cs.rice.edu
Department of Computer Science, Rice University,
6100 Main St MS 132, Houston TX 77005

Abstract

We develop a new hybrid model of human learning on the NRL Navigation Task (Gordon et. al. 1994). Unlike our previous efforts (Gordon & Subramanian, 1997) in which our model was crafted from verbal protocols and eyetracker data, we demonstrate the feasibility of using visualmotor data (time series of sensor-action pairs) gathered during training to construct models of a subject's strategy. The goal of our cognitive modeling is to provide a sufficiently detailed description of the subject's strategic misconceptions in real-time, in order to tailor a personalized, task training protocol. Using a small-parameter hybrid model that can be estimated directly and efficiently from the visualmotor data, we study the deviation of the subject's action choices from that dictated by a near-optimal policy for the task. This model gives us a clear description of the subject's current strategy relative to the near-optimal policy, thus directly suggesting performance hints to the subject. We also provide evidence that our model parameters are sufficient to account for individual differences in learning performance.

Introduction

Our goal is to build computational models of humans learning to perform complex visualmotor tasks. By a model of human learning, we mean an explicit representation of the human's action policies (mapping from the perceptual inputs to motor actions) and its evolution over time. The models will be used in designing personalized training protocols to help humans achieve high levels of competence on these tasks. This intended use places constraints on the class of models we can consider and the methods for evaluating them. In particular, the models need to be detailed enough to pinpoint problems in a subject's learning; yet be coarse enough to be unambiguously built from the available visualmotor learning data. Our criterion for evaluating models is empirical: (i) they must accurately identify incorrect aspects of the subject's strategy, and (ii) when used in place of the human, they must yield comparable performance.

A major challenge in this endeavour is the fact that the visualmotor data are at an extremely low level. One approach to modeling in such a situation is to start with a cognitive architecture, and then to find parameter settings for that architecture which recreate the available low-level data. This tactic is adopted by Newell in UTC, Anderson in ACT* and in EPIC by Kieras and Meyer. We take an alternative approach here based on behavioral cloning (Sammut et. al., 1998). In our approach, the low-level visualmotor data is taken as the ground truth, and using ideas from machine learning and data mining we "compress" the data in the form of a policy which maps sensors to actions. If there are high level regularities at the policy level in the learning data, they will be reliably extracted by our learning algorithms. This approach has the advantage that cognitive modeling constructs arise endogenously from the data, rather than being stipulated *a priori.*

Our task domain is the NRL Navigation task (Gordon, et al., 1994) developed by Alan Schultz at the Naval Research Laboratory (NRL). It requires piloting an underwater vehicle through a field of mines guided by a small suite of sonar, range, bearing and fuel sensors. Sensor information is presented via an instrument panel that is updated in real-time. The sensors are noisy. Decisions about motion of the vehicle (speed and turn) are communicated via a joystick interface. The task objective is to rendezvous with a stationary target before exhausting fuel and without hitting the mines. The mines may be stationary or drifting. A trial or episode begins with the vehicle being randomly placed on one side of a mine field and ends with one of three possible outcomes: the vehicle reaches the target, hits a mine, or exhausts its fuel. Reinforcement, in the form of a scalar reward dependent on the outcome, is received at the end of each episode. Since the mine configurations vary from episode to episode, it is fruitless for subjects to memorize a sequence of actions that will get the vehicle to the target. To solve the task, subjects must learn a policy for choosing actions based on the sensor values presented to them.

The Navigation task belongs to the family of partially observable Markov decision processes. With the addition of the last action taken, we can transform it into a fully observable Markov decision process (MDP). This transformation lends theoretical tractability because deterministic optimal decision procedures exist for MDPs. However, the size of the state space is about 10^{18} and there are 153 choices of action at each time step, which make the Navigation task extremely challenging both for humans as well as for present-day learning algorithms like reinforcement learning (Sutton, 1988).

There are four major sources of complexity in the Navigation task from a cognitive perspective: (1) the need for rapid decision making with incomplete information, (2) the sheer number (10^{18}) of distinct sensor configurations for which an action choice has to be computed, and the need to learn a partition in the sensor space while acquiring a policy, (3) limited binary feedback at the end of each episode, and, (4) a tightly coupled action space in which the different components (turn and speed) cannot be learned independently. Together, these make the task difficult for our human subjects; one out of every three never acquires the task with our current training protocols.

Our data was gathered as follows. Five subjects ran the Navigation task with a configuration of 60 mines, small mine drift, and low sensor noise.[1] Subjects trained for five days, spending an hour each day running consecutive episodes. The number of episodes per hour varied from around 60 to 160. Each episode varied from 40 to 200 time steps. At the beginning of the first session, subjects were told they had to navigate through a minefield to get to a target location. They were allowed to interact with the task to get comfortable with the use of the joystick. We collected the time series of sensor action pairs as well eyetracker data for the entire training period. We also videotaped the subject and recorded all their verbal utterances. In this paper, we focus on the time series of sensor action pairs to determine the strategy used by the subject.

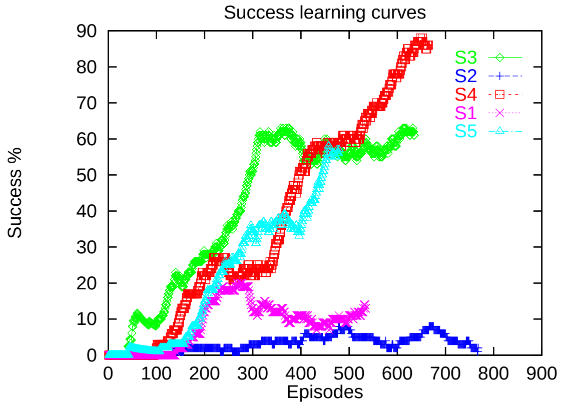


Figure 1: The evolution of success percentages on the Navigation task as a function of training for five subjects.

In Figure 1 we show the learning curves of the five subjects. Note that the success learning curves are remarkably similar for the three subjects who eventually acquired the task. Subjects go through periods of relatively stable performance, punctuated by substantial improvements. The success curves for the subjects who fail to learn the task are also very similar. This raises hope for building a common computational model for all subjects, with a few parameters to account for individual variations.

The visualmotor performance data for the task is a time series in which each element is of the form: *(episode,timestep,range,bearing,s1,s2,s3,s4,s5,s6,s7, last_turn, last_speed turn,speed).* We have over thirty megabytes of visualmotor data for each subject. Extracting the policy used by the subject from this data is difficult for several reasons: (1) the high dimensionality of the data, and the need to find a small number of partitions in the sensor space that meaningfully cluster action choices, (2) noise in the motor data because of joystick hysteresis, (3) data is non-stationary, since the policy adopted by a subject changes with training.

The approach: comparison against the optimal policy

The key to interpreting visualmotor data is a partitioning of the sensor space into a small number of equivalence classes, each of which is associated with an action choice policy. In this paper, we use the discretization of the sensor space adopted by a near-optimal policy to analyze the distribution of

[1] Five undergraduates at San Diego State University participated in this experiment. The data collection was performed by our collaborators Diana Gordon and Sandra Marshall.

1. *Part 1: Seek goal: (Sonar in direction of goal is clear)* Follow that sonar at half speed, unless it is the straight ahead sonar, then travel at full speed.
2. *Part 2: Avoid mine/gap finder: (Sonar not in direction of goal is clear)* Turn in place in the direction of the first clear sonar counted from the middle outward.
3. *Part 3: Avoid mine/gap finder: (No clear sonar)* If the last turn was nonzero, turn again in the same direction by that amount, else initiate a turn by summing the sonars to the left and right, and turning in the direction of the lower sum.

Table 1: The three-part near-optimal policy for the NRL Navigation Task. The italicised conditions for each part represent the equivalence class of sensor values that define the part.

actions chosen by our subjects. This approach allows us to determine the deviations of the subject's strategy from that of the near-optimal policy, which can then be the basis of directed training. A potential disadvantage of the approach is that if there are other near-optimal policies that adopt very different discretizations, a subject using them would be misdiagnosed as making strategic errors[2]. We now describe the near-optimal policy that we discovered, and then present results of modeling the subject's strategy viewed through its sensor space discretization.

A near-optimal policy for the Navigation task

A near-optimal policy for the task is deterministic and is shown in Table 1. It must be emphasized that *discovering this solution was not easy!*. It took several months of work with a machine learning algorithm to arrive at this policy.

The near-optimal policy in Table 1 succeeds at least 99.7% of the time; its performance has not been matched by our best human subjects. There are three key properties of the near-optimal policy.

1. *task decomposition*: the policy decomposes the overall goal into the subgoals of avoid-mine and seek-goal, a decomposition which appears universal among our human subjects. However, the solutions to the sub-goals are tightly coupled and this is difficult for humans to learn.
2. *dependence between turn and speed choices*: Turning at zero (or close to zero) speeds is essential for success on this task. In addition, turning consistently in one direction while trying to find gaps in the minefield, is crucial.
3. *appropriate discretizations*: the near-optimal policy discretizes the sonar values that range from 0 to 220 into a binary distinction of clear/blocked with the threshold set at 50. The bearing sensor with 12 values is discretized into six, and the range sensor is ignored. The action space is discretized too: the turn action with 17 values is discretized into nine values, and speed with 9 values is discretized into three (zero, half speed, full speed).

The near-optimal policy partitions the state space into three mutually exclusive and collectively exhaustive components. The effective number of states considered by Parts 1 and 2 of the policy is $2^7 * 6$ which is 768. This is because both parts consider the values of seven sonars, each of which is discretized into clear and blocked, and six values for bearing. The 768 states are really equivalence classes over $\approx 10^{14}$ base states in the original sensor space. Part 3 examines the previous turn, and thus deals with an effective state space of size $9 * 27$ which is 243.

Model extraction algorithm

For ease of presentation, we first describe the model extraction method under the assumption that the visualmotor sequence data represents a stationary process. This assumption will be relaxed at the end of this subsection. Using the discretizations and definitions of three parts of the near-optimal policy, we classify each sensor-action pair in the visualmotor sequence as belonging to Part 1, Part 2 or Part 3 equivalence classes. For example, if the sonar in the direction of the goal is clear in the sensor vector, the sensor action pair is classified as a Part 1 pair.

Since the action decisions in Part 1 (resp. Part 2) of the near-optimal policy depend only on the current values of the discretized bearing, we estimate the conditional probability that the subject

[2] However, we were unable to determine other near-optimal policies for the NRL task after months of computation and investigation.

chooses a particular discretized[3] action (turn and speed) given the value of the discretized bearing. For discretized action a in the set A, and discretized bearing b let n_{ab} be the number of times a is taken by the subject in a Part 1 sensor action pair with bearing b.

$$P(a|b) = \frac{n_{ab}}{\sum_{c \in A} n_{cb}}$$

The action selection scheme adopted by the near optimal policy for Part 3 sensor equivalence class is inherently sequential. Therefore, to fit Part 3 behavior, we use hidden Markov models (HMMs) (Rabiner, 1989). We identify sequences of sensor-action pairs that belong to Part 3 and train a three state left-to-right HMM on the data[4].

The parametric hybrid model that we construct from the subject data is shown in Figure 2. Note that the model reflects the task structure. In particular, we use conditional action probability distributions to extract subject behavior on the seek-target subgoal of the task, and a combination of a conditional action probability distribution and an HMM to describe the solution of the coupled subgoal of avoid-mine. This model has few relatively few parameters and can be easily estimated online. It describes the subject's policy viewed through the equivalence class filter imposed by the near-optimal policy. By comparing the subject's model for the three parts against that of the near-optimal policy, we can read off strategic errors in the subject's policy. Examples of such comparisons are offered in the next section.

To accommodate the non-stationary visualmotor data sequence, we identify stationary subsequences from which the conditional probabilities are estimated and the HMMs are trained. We estimate conditional probability distributions for Part 1 and Part 2 and HMMs for Part 3 over small contiguous blocks[5] of episodes in the data sequence. We then use a standard measure of distance between distributions (KL-divergence[6]) to determine when a significant shift in the Part 1 and Part 2 distributions have occurred. For the HMMs for Part 3, we use KL divergence between both the transition probabilities and the output probabilities to determine when a significant shift has occurred. This procedure identifies points in the sequence that correspond to significant differences in the action selection distributions. These shift points are supported by verbal protocol data as well as eyetracker data. The sequences between shift points are taken as stationary, and the model extraction procedure described above is applied to them.

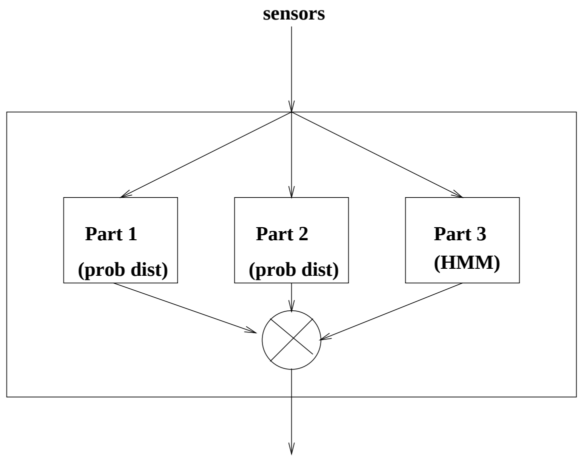


Figure 2: The structure of our hybrid model for the Navigation task.

We now turn to the presentation of experimental results from the use of our model extraction technique on the visualmotor data corpus for the NRL Navigation task.

Modeling Results

Examination of the conditional probability distributions of part 1 and 2 and HMMs of part 3 from the three successful subjects reveals that they learn the following.

1. to follow the as-the-crow-flies strategy in the direction of the goal in states in Part 1.
2. to slow down significantly when turning in Parts 1, 2 and 3.
3. to turn minimally to avoid mines in states in Part 2.

[3] The original action set has cardinality 153; the discretized set has nine turns and three speeds making a total of 27 actions.

[4] We experimented with a number of hidden states ranging from 2 to 10, and using log-likelihoods on a left-out test set, we determined that three was the best choice for number of hidden states.

[5] The size of the blocks is determined empirically, and we respect day boundaries in the construction of the blocks.

[6] $KLdiv(p,q) = \sum_{s \in S} p * log(p/q)$, where p and q are discrete distributions defined over a set S.

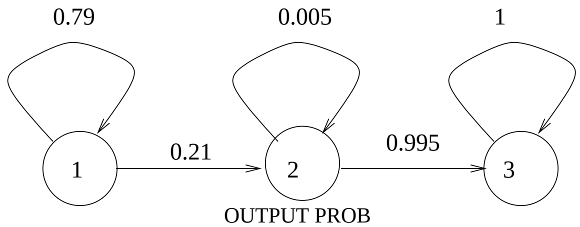
action	day 1	day 2	day 3	day 4	day 5
$t = 0, s = 0$	0.334	0.370	0.192	0.090	0.078
$t < 0, s = 0$	0.104	0.083	0.106	0.052	0.031
$t > 0, s = 0$	0.083	0.075	0.081	0.021	0.035
$t = 0, s > 0$	0.408	0.454	0.552	0.695	0.646
$t < 0, s > 0$	0.042	0.005	0.015	0.052	0.081
$t > 0, s > 0$	0.028	0.014	0.053	0.090	0.129
KLdiv	3.528	4.220	2.894	2.369	2.011

Table 2: The evolution of the conditional action probability distribution for Subject 4 in Part 1 when bearing = 11 o'clock. The turn t and speed s choices are discretized into six categories for reading ease. Turns greater than zero are left turns, and turns less than zero are right turns For a full explanation of this table, please see the text below.

4. to turn in place consistently to find gaps in the minefield in Part 3.

We demonstrate the first point above with data from Part 1 for Subject 4. For this subject, shifts in Part 1 distributions correspond to day boundaries, so we present the evolution of his action selection policy for each day of training. Table 2 presents the conditional probability of Subject 4 taking an action a, given that the bearing (goal direction) is 11 o'clock. That is, the target lies slightly to the left of the current heading of the vehicle. The near-optimal policy dictates a mild turn to the left. The KL divergence between the subject's policy and the near-optimal policy is shown in the last row of the table. Note that the subject's policy initially diverges and then approaches the near-optimal policy between day 2 and day 3. Also note the rapid decline in the probability of pausing (turn and speed both equal to zero) as training proceeds, with the most dramatic reductions occurring between day 2 and day 3 and day 3 and day 4. The probability that the subject chooses a left turn goes down from day 1 to day 2, but then steadily increases from day 3 forward. All action probabilities except for straight ahead ($t = 0, s > 0$) and left turn ($t > 0, s > 0$) rapidly decay to zero, indicating that the subject is learning to follow bearing well in the Part 1 equivalence class.

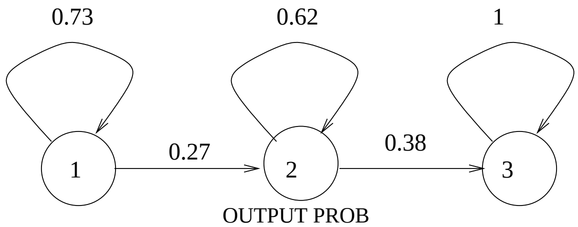
It should be emphasized that while Part 1, Part 2 and Part 3 models for each subject co-evolve, they do not evolve at the same rate, and rarely do significant shifts in these probability models coincide. While Part 1 distributions evolve rather slowly and shifts in them occur aligned with day boundaries; Part 3 HMMs evolve much more quickly. For example, for Subject 5, the Part 3 HMM we acquired on data from episodes 45-67 of day 2, differs significantly from the one learned from episodes 68-90 of day 2. These two HMMs are shown in Figures 3 and 4. The first HMM in Figure 3 is a mathematical description of the following strategy: pause (speed = 0 and turn = 0) for a while, and then make an average of two moves with non-zero speed and turn, and finally settle into oscillating back and forth between pauses, left and right turns at zero speed until time runs out. Note that the probability of left and right turns in the terminal hidden state 3 are about the same. In Figure 4, the HMM encodes the following very different strategy: pause for a while, make a left turn at zero speed, and then settle into an action pattern with a consistent preference for turning to the right at zero speed. That is, the subject no longer oscillates back and forth when hemmed in by mines, she sweeps them from left to right trying to find a gap between the mines. This behavior is fairly close to the near-optimal policy for Part 3. In fact, with practice we can get her to spend less time in the state labeled 1, completely eliminate state 2, and in state 3, we can zero out her tendency to pause and increase her probability to turn right. This analysis forms the basis for designing lessons to help the subject acquire greater competence at the task.



ACTIONS:	OUTPUT PROB		
0 0	0.84	0.00	0.35
right 0	0.03	0.003	0.33
left 0	0.05	0.00	0.31
other	0.08	0.997	0.00

Figure 3: A hidden Markov model that generates and explains the behavior of Subject 5 in states where all sonars are blocked, day 2, episodes 45-67.

How good a fit to performance does the model in Figure 2 provide? The results on Subject 5 for day 2, for episodes 45-67 and episodes 68-90 are shown in Table 3. Note that although the magnitudes pro-



ACTIONS:		OUTPUT PROB	
0 0	0.82	0.05	0.37
right 0	0.0024	0.00	0.553
left 0	0.025	0.95	0.07
other	0.131	0.00	0.00

Figure 4: A hidden Markov model that generates and explains the behavior of Subject 5 in states where all sonars are blocked, day 2, episodes 68-90.

D 2, ep 45-67	Succ	Exp	Timeouts	Total
Subject 5	0	12	11	23
Model	0	17	6	23
D 2, ep 68-90	Succ	Exp	Timeouts	Total
Subject 5	0	2	13	15
Model	0	4	11	15

Table 3: The behavioural fit of the new hybrid model to Subject 5, day 2, episodes 45-90.

duced by the model only coarsely approximate those produced by the subject, the trends are captured. For example, both model and subject increase the number of timeouts and reduce the number of their explosions. To get better fits to the performance data, we are currently experimenting with distributions for Parts 1 and 2 conditioned additionally on the previous action.

Conclusions and Related Work

Our work builds on several distinct pieces of work in the cognitive science as well as the machine learning community. The use of probabilistic models in generating hints for performance improvement is considered by (VanLehn, et. al., 1998). Our work uses a mixture of probabilistic models (conditional action distributions and HMMs) instead of Bayesian networks, and our models are automatically learned from visualmotor data. While the structure of the model is obtained from task analysis (Fredericksen and White, 1989), the parameters are learned by sampling the visualmotor data corpus. The idea of behavior cloning introduced by (Sammut et. al., 1998) underlies our approach, however the specific techniques for partitioning and learning from nonstationary data are different and novel.

In sum, we have developed a new hybrid model for the NRL Navigation task and presented methods for automatically learning it from low level visualmotor data. The model succinctly represents the deviation of the subject's policy from a near optimal policy, and allows directed design of new training instances. The model is expressive enough to capture individual differences in strategy. Our current work is to provide closer behavioral fits to the visualmotor data by using richer probabilistic representations.

Acknowledgements

We sincerely thank our collaborators Diana Gordon of the Naval Research Lab and Sandy Marshall of San Diego State University, and our sponsors Helen Gigley and Susan Chipman of the Office of Naval Research for making this research possible.

References

Fredericksen, J. and White, B. (1989). An Approach to Training Based on Principled Task Decomposition. *Acta Psychologica*, 71:89-146.

Gordon, D., Schultz, A., Grefenstette, J., Ballas, J., & Perez, M. (1994). *User's guide to the navigation and collision avoidance task* (AIC-94-013). Washington, D.C.: Naval Research Laboratory.

Gordon, D., & Subramanian, D. (1997). A cognitive model of learning to navigate. *Proceedings of the 19th Annual Conference of the Cognitive Science Society* (pp. 271-276). Lawrence Erlbaum Associates.

Rabiner, L. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *proc. IEEE*, 37(2):257-286.

Sammut, C. and Harries, M. B. (1998). Extracting hidden context. *Machine Learning*, 32:101-126.

Sutton, R. (1988). Learning to predict by the method of temporal differences. *Machine Learning*, 3(1):9-44.

Gertner, A. S, Conati, C. and VanLehn K. (1998). Procedural help in Andes: generating hints using a Bayesian network student model. *Proceedings of the AAAI-98*, AAAI Press.