Inverse Reinforcement Learning with Graphical Models

Saurabh Daptardar
Department of ECE
Rice University
Houston, TX 77005
svd3@rice.edu

Abstract

The goal of this project is to better understand the underlying mechanisms and decision making process of the brain by characterizing the predictive latent variables and their relationship to recorded neural responses. Behavioral experiments show that our brain uses latent variables to make decisions, but this process remains poorly understood. An agent in control system can only partially observe any environment and works with an estimated model (what it believes) of the environment. We will use graphical models and reinforcement learning as a means for identifying state and latent variables of the system and to find the belief model (infer the latent dynamics) assumed by the (agent).

1 Introduction

Our brains have been developed to be intelligent enough to understand the physical world and and interact with it. Behavioral experiments have shown that we don't act randomly [1], but to achieve a certain goal, and there needs to be certain rational reasoning behind the actions we take . Capturing purposeful, sequential decision-making behavior can be quite difficult for general-purpose statistical machine learning algorithms; in such problems, algorithms must often reason about consequences of actions far into the future. To consider the consequences of actions into the computation, any agent needs an prediction model. This model we refer to as the belief model. A very simple and brief pipeline for computation would be receives a stimulus (observe), update belief state, derive an action and act, state of the world updates, and all this process obviously is stochastic. The causal dependencies in this can be modeled as graphical models. We won't study how this belief model is learned by the agent, but only try to figure out what this belief model is.

We formulate the problem in a general framework of Markov Decision Processes (MDP). It assumes that agents act to optimize some reward function. We hypothesize that the brain is (near)-optimal and must be solving a MDP, optimizing some reward function, assuming (believing) belief dynamics to plan its actions.

The Reinforcement Learning (RL) framework solves the forward problem of finding the optimal policy given the dynamics and rewards. But the agent has already performed the task with its believed optimal policy, our goal is to solve the inverse problem (IRL) to find the dynamics the agent used to derive the its optimal policy with the observations it made and actions it took (experimental data). We propose an algorithm with Graphical model inference to estimate the parameters of the belief dynamics. In this report we only study the simulated experiments (not apply it to actual data).

2 Background

We won't go into the Reinforcement Learning topics in this report as it is not relevant to this course. But just touch upon and define only the concept and nomenclature necessary to understand the work presented.

In RL setting, the objective is to find a policy π to act optimally (maximize total rewards) given reward function $r(z_t, a_t)$ and dynamics $\Pr(z_{t+1}|z_t, a_t)$. A policy is a mapping from the set of sets to set of actions $\pi : \mathcal{Z} \to \mathcal{A}$. A stochastic policy is the probability of choosing an action given the state $\pi(a_t|z_t) = p(a_t|z_t)$.

The inverse RL problem became studied mainly for apprenticeship learning [2, 3]. Most of the literature in IRL deals with trying to compute the reward function under the assumption that dynamics are known. The IRL problem is ill posed in many cases, and needs big assumptions to solve. There is a degeneracy inherent in the MDP formulation, which makes IRL particularly difficult. There exists many pairs of dynamics and reward functions which give the exact same policy which makes finding the true dynamics and rewards together difficult. To relax the problem, assumption that one of the two quantities is known is made. Hence [4, 5] explore this in probabilistic inference and bayesian frameworks, choosing a solution with maximum entropy. Unlike other papers we try to find belief dynamics assuming the reward function is the similar to the external rewards given to the agent.

3 Graphical model inference

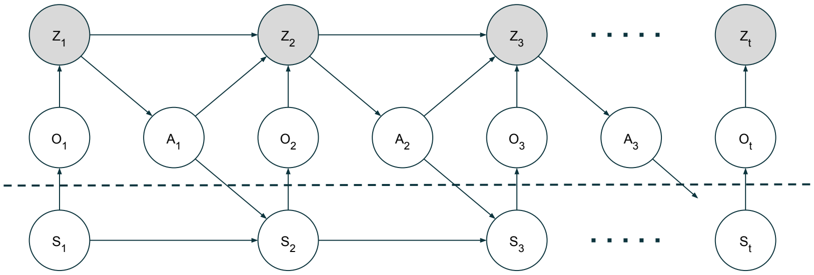


Figure 1: Full model

We model the experimental task as a temporal directed graphical model with repetition of similar blocks as shown in the above figure, the edges are assumed to be known from a general control systems perspective and so we don't have to estimate the edges which edges exist, but the conditional distributions they represent. We assume the model admits to *Recursive Factorization* and hence the joint distribution can be written as product over factors of the blocks. The dashed line in the figure separates the computations that the agent might be doing from the real world dynamics.

The variables s_t, o_t, z_t, a_t represent the real physical world states, the partial observations we make of the states, the latent variable which will be belief states in our case that the brain uses, and the actions taken which affect the world states. We have certain factor distributions already known to us, such as world dynamics given by the design or construction of the experiment. By dynamics we mean conditional probabilities $\Pr(s_{t+1}|s_t, a_t)$ and $\Pr(o_t|s_t)$. In the experiments we get to observe all the actions taken and observations made by the agent, i.e. $a_{1:T}$ and $o_{1:T}$ are known. We also know all the world state information but that shouldn't directly affect the latent dynamics directly. It is decoupled by the observations o_t as shown by dashed line in the figure which is why we won't use it into the computations but use it only as a generative model for simulation studies.

The model describes the following computational framework. The state s_t generates some observation o_t with known conditional distribution given above, the latent variable z_t in the brain has its own dynamics $\Pr(z_{t+1}|z_t, a_t)$ by which the latent state Markov chain evolves. The latent state gets corrected after the observation o_t given a belief emission model $p(o_t|z_t)$. Note here that the latent variable is not causing the observations, but that it is a model belief of the brain. The posterior distribution over z_t is just weighted by emission model and the belief dynamics acts as a prior over latent states. Given stationary dynamics of latent states, we have an stochastic optimal policy. The action thus taken according to the policy π affect the world state and the next latent state.

The optimal policy distribution is calculated according to the RL framework but it implicitly depends on the dynamics. Let Θ be the belief dynamics model which completely describes the transitions and emission probabilities then policy is $\pi(a_t|z_t;\Theta)$, which makes the problem even more difficult. Note here that even though we have data of actions taken, we don't know the actual policy as we can't observe the latent states and just that the policy must be optimal w.r.t. some dynamics. We need to jointly find the dynamics and the policy such that the policy is optimal w.r.t. the dynamics and they also maximize the likelihood of the actions and observations (the data).

To solve this problem we have proposed a novel iterative algorithm for the joint estimation which is as follows:

- Step 1 Initialize the belief dynamics and policy randomly
- Step 2 Repeat the following until convergence:
  - Step 2.1 Find or update (incrementally) policy $\pi(a|z)$ using any method which improves policy. Improved policy means agent gets more total reward following this policy than previous.
  - Step 2.2 Fix the policy and estimate/update the parameterized dynamics $p(z_{t+1}|z_t, a_t)$ and $p(o_t|z_t)$ using Maximum Likelihood Estimate.

The convergence criterion is when the policy stabilizes (no further changes) and log likelihood is maximized. We usually use EM [6] for computing the MLE, so the algorithm converges once the policy stabilizes and EM converges. Generally policy stabilizes much sooner than the dynamics. Also, to speed up the computation we don't have to wait for EM to converge in each iteration. We just need to move in the direction of the MLE.

3.1 Maximum Likelihood Estimator

The update in policy is done according to the optimality principles in RL framework. To estimate the parameters for the other part we maximize the partial data log likelihood of actions and observations generated. We already stated the assumption that the model admits to recursive factorization and so

$$\Pr(a_{1:T}, z_{1:T}, o_{1:T}|s_{1:T};\theta) = \prod_{t=1}^{T-1} p(z_{t+1}|z_t, a_t, o_{t+1})\pi(a_t|z_t)p(o_t|s_t)$$

$$\text{where,} \quad p(z_{t+1}|z_t, a_t, o_{t+1}) \propto p(z_{t+1}|z_t, a_t)p(o_{t+1}|z_{t+1})$$

We get the partial data log likelihood by marginalizing on z_t $\forall t = 1, 2 \dots T$ but since this computation would be intractable we optimize the tightest lower bound derived using Jensen's inequality. We skip some derivations in order to make it in 4 page limit.

If we choose to parameterize the distributions as every element in the matrices (transition and emission) then

$$\text{MLE:} \quad \widehat{\theta} = \operatorname*{argmax}_{\theta\in\Theta} \sum_{z_1, z_2, \dots z_t} Q(z_{1:T}) \sum_{t=1}^{T-1} \log P_{ij}^k \mathbb{1}_{ijk}(z_t, z_{t+1}, a_t) + \log \xi_{il} \mathbb{1}_{il}(z_t, o_t)$$

$$\text{s.t.} \quad \sum_{j\in\mathcal{Z}} P_{ij}^k = 1 \quad \forall i \in \mathcal{Z}, k \in \mathcal{A} \quad \text{and,} \quad \sum_{l\in\mathcal{O}} \xi_{il} = 1 \quad \forall i \in \mathcal{Z}$$

This case is similar to Hidden Markov models (HMM) [7] and hence we will derive the EM updates similar to that of HMMs by forward and backward algorithms [8]. We will redefine the forward and backward procedure variables as follows:

$$\alpha_t(i,k) = \Pr(o_{1:t}, a_{1:t-1}, z_t = i, a_t = k)$$
$$\beta_t(i,k) = \Pr(o_{t+1:T}, a_{t+1:T}|z_t = i, a_t = k)$$

We derive recursive updates for the forward and backward algorithms. It is easy to show, choosing $Q(z_{1:T}) = p(z_{1:T}|a_{1:T}) = \frac{p(z_{1:T}, a_{1:T};\theta)}{p(a_{1:T})}$, and with our redefined variables α and β and the updates will be

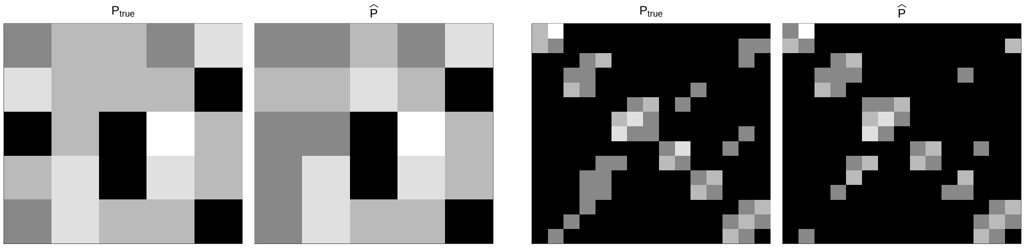
$$P_{ij}^{k,(m+1)} = \frac{\sum\limits_{t=1}^{T-1} \pi(a_{t+1}|j) P_{ij}^{k,(m)} \alpha_t(i,k)\beta_t(j,a_{t+1})}{\sum\limits_{t=1}^{T-1} \sum\limits_{j\in\mathcal{Z}} \pi(a_{t+1}|j) P_{ij}^{k,(m)} \alpha_t(i,k)\beta_t(j,a_{t+1})}$$

$$\xi_{il}^{(m+1)} = \frac{\sum\limits_{t=1}^{T-1} \mathbb{1}_l(o_t)\alpha_t(i,a_t)\beta_t(i,a_t)}{\sum\limits_{t=1}^{T-1} \alpha_t(i,a_t)\beta_t(i,a_t)}$$

4 Results and Discussion

We simulated the data using the whole model as shown in Figure 1. We randomly generate dynamics and emissions for bot latent and world states produce the world states and observations by sampling. For sampling the latent states we compute the posterior given the observation o_t(and *not the action* a_t as this is the generative model) by just weighting prior $p(z_t|z_{t-1}, a_{t-1})$ with believed emissions $p(o_t|z_t)$ and sampling then. The optimal policy distribution is pre computed as we know the true dynamics and actions are sampled from the policy after the latent states.

The results for dynamics with size of state space $n = 5$ and 10 are shown in Figure 2 below. On the left of each is the true dynamics, and on the right is the estimated dynamics. We simulated the experiments both with random matrices and random but structured matrices. As can be seen in the Figure 2b the matrix is more structured and sparse. The errors reported below are mean squared errors averaged over whole matrix and also over multiple simulations with similar state space size. For structured matrices the errors have been reported separately in the table below Many time with smaller evidence space (observations and actions) the algorithm doesn't recover true parameters and gives huge error in terms of MSE. It is important to note that MSE as a measure of error may not be appropriate for this problem. I'm currently studying to find a better measure along the lines of Fisher information, but that is work in progress.



(a) number of latent states = 5

(b) number of latent states = 15

Figure 2: comparison between true and estimated dynamics matrices with our algorithm

Table 1: Mean squared error averaged over trials

	1	2	3	4	5	6	7	8	9	10	11	12	13
$\|\mathcal{Z}\|$	0.078	0.05	0.017	0.09	0.063	0.029	0.096	0.071	0.035	0.112	0.088	0.067	0.041
$\|\mathcal{O}\| \times \|\mathcal{A}\|$	5	5	5	10	10	10	15	15	15	30	30	30	30
error	3x2	6x2	6x2	6x3	6x3	6x3	12x3	12x3	12x3	15x3	20x3	20x3	20x3

For future work, I want to re-parameterize the distribution instead of using matrix elements, so that it gives structure and generality to the distribution. Re-parameterization also would be important to reduce dimensionality a lot as the number of parameters current grow $\mathcal{O}(n^2)$ to the number of states.

References

[1] Alexandre Pouget, Jeffrey M Beck, Wei Ji Ma, and Peter E Latham. Probabilistic brains: knowns and unknowns. *Nature neuroscience*, 16(9):1170–1178, 2013.

[2] Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, pages 663–670, 2000.

[3] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM, 2004.

[4] Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. *Urbana*, 51(61801):1–4, 2007.

[5] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

[6] Todd K Moon. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60, 1996.

[7] Lawrence Rabiner and B Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986.

[8] Lloyd R Welch. Hidden markov models and the baum-welch algorithm. *IEEE Information Theory Society Newsletter*, 53(4):10–13, 2003.

[9] Steffen L Lauritzen. *Graphical models*, volume 17. Clarendon Press, 1996.

[10] Martin J Wainwright, Michael I Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

[11] Michail G Lagoudakis and Ronald Parr. Least-squares policy iteration. *Journal of machine learning research*, 4(Dec):1107–1149, 2003.