Gradient Ascent on POMDP Policy Graphs

Douglas Aberdeen
Research School of Information Science and Engineering
Australian National University

Jonathan Baxter
WhizBang! Labs

September 6, 2001
Outline

- Motivation
- Gradient ascent of stochastic finite state controllers
- Simulation based policy gradient
- Related Work
- Pitfalls of gradient ascent on FSCs
- The Heaven-Hell problem
- A better approach: expectations over I-state trajectories
- Model based policy gradient
- Experiments
A POMDP

Environment

Agent

\[ r_t = R(i_t) \]

\[ \nabla_t \eta \]

\[ \nu(y_t | i_t) \]

\[ \mu(u_t | \theta, y_t) \]

\[ P(u=0) \]

\[ P(u=1) \]
Bellman’s Equation
Richard Bellman (1957)

\[ J^* = r + \beta P J^*. \]

- Computes the value of each state \( J(s) \).
- Describes \( n_s \) equations with \( n_s \) unknowns (\( n_s = \) states).
- Model must be known.
- This formulation is for MDPs only.
- Intractable for more than a few tens of states.
Policy Iteration
Bellman (1957) and Howard (1960)

- Finds a solution to the Bellman equation via dynamic programming.
- Practical for much larger state spaces.
- Related method: value iteration.
- Function approximation for RL in use by 1965 (Waltz and Fu 1965).
Simulated Methods

- Do not require the environment model. They learn from experience.
- Q-learning (Watkin’s 1989).
- Eligibility traces: TD(\(\lambda\)) (Sutton 1988).
Historical perspective IV

Exact POMDP methods
Aström (1965), Sondik (1971)

- Re-introduces the environment model.
- Modified Bellman equation computes the value of belief states.
- At least PSpace-complete so approximate methods are needed.

Controlling POMDPs sans model, with infinite state and action spaces, is about as general as it gets.
Failings of current methods

The drawbacks of current approximate POMDP methods include:

- Assumption of a model of the environment.
- Only recalling events finitely far into the past.
- Use of an independent internal state model that does not aim to maximise the long term reward.
- Do not easily generalize to continuous observations and actions.
- Applications to toy problems only.
• Motivation

• Gradient ascent of stochastic finite state controllers

• Simulation based policy gradient

• Related Work

• Pitfalls of gradient ascent on FSCs

• The Heaven-Hell problem

• A better approach: expectations over I-state trajectories

• Model based policy gradient

• Experiments
Why we need internal state for POMDPs

Memoryless controllers are not optimal in partially observable environments:

\[ r = \begin{array}{cccccc}
U & 0 & 0 & 0 & 0 & 1 \\
\end{array} \]

(Peshkin, Meuleau, Kaelbling 1999)
Environment

\( r_t = R(i_t) \)

\( \nabla_t \eta \)

\( \nu(y_t | i_t) \)

\( y_t \)

\( i_t \)

\( \omega(b_{t+1} | \alpha, b_t, y_t) \)

\( b_{t+1} \)

\( \mu(u_t | \theta, b_{t+1}, y_t) \)

Agent
Figure 1: Stochastic I-state transition function.
Policy gradient methods

- Algorithms for of estimating the gradient of
  \[ \eta = \lim_{T \to \infty} \mathbb{E} \left[ \frac{1}{T} \sum_{t=1}^{T} r_t \right] \]
  with respect to the parameters of the policy.

- True gradient is \[ \nabla \eta = \pi' \nabla P[I - P + e\pi']^{-1} r \], where P is the MDP state transition matrix for the current policy.

- Learns the policy directly, i.e. no value functions.

- Works for POMDP environments if observations are belief states or if I-state is used.

- Variance in the gradient estimates is a problem.

  Hybrids: VAPS (Baird & Moore 1999).
Outline

- Motivation
- Gradient ascent of stochastic finite state controllers
- *Simulation based policy gradient*
- Related Work
- Pitfalls of gradient ascent on FSCs
- The Heaven-Hell problem
- A better approach: expectations over I-state trajectories
- Model based policy gradient
- Experiments
Simulation based policy gradient: GPOMDP

Baxter & Bartlett (1999)

• If \( P \) and \( \nu \) are not available we can approximate the gradient by introducing a discount factor \( \beta \).

• GPOMDP estimates the gradient from a single sampled environment trajectory, generating gradient contributions at each step.

• Provided \( \frac{1}{1-\beta} > \tau \), and \( T \) is sufficiently large, then the GOMDP estimate \( \tilde{\nabla}_T \eta \) is good.

• Unlike REINFORCE, GPOMDP does not require the identification of recurrent states.

• Computes the gradients for \( \omega(b|\alpha, a, y) \) and \( \mu(u|\theta, b, y) \) independently.
Policy graph learnt for the Load/Unload problem.
Outline

- Motivation
- Gradient ascent of stochastic finite state controllers
- Simulation based policy gradient
- Related Work
- Pitfalls of gradient ascent on FSCs
- The Heaven-Hell problem
- A better approach: expectations over I-state trajectories
- Model based policy gradient
- Experiments
Related work

- Use HMMs to learn a model (Chrisman 1992).
- Recurrent Neural Networks (Lin & Mitchell 1992).
- Differentiable approx. to piecewise function (Parr & Russell 1995).
- U–Tree’s: Dynamic finite history windows (McCallum 1996).
- External memory setting actions (Peshkin, Meuleau, Kaelbling 1999).
- Grad ascent on IOHMMs used as stochastic FSCs (Shelton 2001).
- Evolutionary approaches (Kwee 2001), (Glickman 2001).
• Motivation
• Gradient ascent of stochastic finite state controllers
• Simulation based policy gradient
• Related Work
  • Pitfalls of gradient ascent on FSCs
• The Heaven-Hell problem
• A better approach: expectations over I-state trajectories
• Model based policy gradient
• Experiments
Failings of policy gradient with I-state

1. GPOMDP has a large variance as $\beta \to 1$.

2. I-states increase the mixing time of the overall system.
   - Importance Sampling (Glynn 1996), (Shelton 2001);
   - replace $\mu$ with an MDP alg. that works on the I-states;
   - eligibility trace filtering to incorporate prior knowledge;
   - deterministic $\mu(u_t | b_{t+1}, y_t, a_t)$.

3. Sensible initial FSC transition probabilities result in very small gradients!
Zero gradient regions for FSCs

**Theorem 1.** If we choose $\theta$ and $\alpha$ such that $\omega(b|\alpha, a, y) = \omega(b|\alpha, y) \ \forall a$ and $\mu(u|\theta, b, y) = \mu(u|\theta, y) \ \forall b$ then $\nabla^\alpha \eta = [0]$.

- Applies to all FSC policy gradient approaches.
- The gradient degrades smoothly as the conditions are approached.
Avoiding zero gradient regions

Key idea: sparse finite state controllers.

\[ \omega(b = 0|\alpha, a, y = 0) = 0.7 \]
\[ \omega(b = 1|\alpha, a, y = 0) = 0.0 \]
\[ \omega(b = 2|\alpha, a, y = 0) = 0.3 \]

\[ \omega(b = 0|\alpha, a, y = 1) = 0.5 \]
\[ \omega(b = 1|\alpha, a, y = 1) = 0.5 \]
\[ \omega(b = 2|\alpha, a, y = 1) = 0.0 \]

Figure 2: Sparse stochastic I-state transition function.
Figure 3: Discrete Heaven-Hell problem. Agent must visit lower state to determine which way to move at the top of the T (Thrun 2000), (Geffner & Bonet 1998).
Outline

- Motivation
- Gradient ascent of stochastic finite state controllers
- Simulation based policy gradient
- Related Work
- Pitfalls of gradient ascent on FSCs
- The Heaven-Hell problem
- A better approach: expectations over I-state trajectories
- Model based policy gradient
- Experiments
A better approach to FSCs using GPOMDP

- We currently sample environment trajectories and I-states.
- We know $\omega$, the stochastic I-state transition function.
- Maintains a belief over I-states and computes expected action probabilities over the I-states.
- Computes the gradient estimate by taking the expectation over all possible I-state trajectories up to time $T$.
- Resembles IOHMM training (Bengio 1995).
- Works for continuous tasks.
Outline

- Motivation
- Gradient ascent of stochastic finite state controllers
- Simulation based policy gradient
- Related Work
- Pitfalls of gradient ascent on FSCs
- The Heaven-Hell problem
- A better approach: expectations over I-state trajectories
  - *Model based policy gradient*
- Experiments
Recall the equation for the true gradient:

$$\nabla \eta = \pi' \nabla P [I - P + e\pi']^{-1} r.$$
Model-based $\nabla_N \eta$

\[
\nabla \eta = \lim_{N \to \infty} \pi' \left[ \sum_{n=0}^{N} \nabla PP^n \right] r \\
\simeq \pi' \nabla P \left[ \sum_{n=0}^{N} P^n \right] r = \nabla_N \eta.
\]

• Worst case complexity $O(n_s^2 n_p n_o n_a)$.

• Load/Unload
  
  - $N = 6 \implies \angle(\nabla_N \eta - \nabla \eta) < 5^\circ$;
  
  - $N = 13 \implies \angle(\nabla_N \eta - \nabla \eta) < 1^\circ$.

• Robot nav $n_s = 208 \times 4$, $n_p = 896$, $n_o = 28$, $n_a = 4$:
  
  $P, \nabla \mu, \nabla \omega < 1s$, $\pi = 127s$, $P^{100} = 220s$, $\nabla P = 138s$. 
Motivation

Gradient ascent of stochastic finite state controllers

Simulation based policy gradient

Related Work

Pitfalls of gradient ascent on FSCs

The Heaven-Hell problem

A better approach: expectations over I-state trajectories

Model based policy gradient

Experiments
## Load/Unload time to convergence

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>time (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>known model</td>
<td>2.5</td>
</tr>
<tr>
<td>GPOMDP</td>
<td>28</td>
</tr>
<tr>
<td>GPOMDP sparse</td>
<td>13</td>
</tr>
<tr>
<td>GPOMDP sparse-exp</td>
<td>12</td>
</tr>
</tbody>
</table>
Robot navigation

Cassandra (1998)

- Noisy observations and actions.
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$\eta \times 10^{-2}$</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>sans I-state</td>
<td>1.37</td>
<td>model based gradient</td>
</tr>
<tr>
<td>GPOMDP sparse</td>
<td>2.32</td>
<td>20 I-states, connectivity=2</td>
</tr>
<tr>
<td>GPOMDP sparse-exp</td>
<td>2.20</td>
<td>”</td>
</tr>
<tr>
<td>belief GPOMDP</td>
<td>3.19</td>
<td>3 layer ANN, $y =$ belief state</td>
</tr>
<tr>
<td>MDP</td>
<td>5.23</td>
<td>fully observable</td>
</tr>
<tr>
<td>Noiseless MDP</td>
<td>5.88</td>
<td>theoretical</td>
</tr>
</tbody>
</table>
Key Conclusions

0. It is possible to perform a search for the optimal policy graph directly.

1. RL algorithms can be extended with I-states to perform this search.

2. A tough problem has been solved, using the sparse initialization trick to avoid the problem of low initial gradients.

3. We can take expectations over I-state trajectories instead of sampling them.
Future Work

- Larger problems from the literature.
- Speech processing.
- Bounds on policy error introduced by too few I-states.
- Automatic selection of $n_b$. 
Acknowledgments

- Drew Bagnell, Malcolm Strems
- Sebastian Thrun

Questions?

mailto:douglas aberdeen@anu.edu.au

So long and thanks for all the pizza!