Stochastic optimal control with linear Bellman equations

Emo Todorov

Applied Mathematics and Computer Science
University of Washington
**Summary**

Desirability function \( z(x) = \exp(-v(x)) \)

- Discrete time (Todorov 2006)
  \[ x' \sim p(\cdot|x) \]
- Continuous time (Kappen 2005)
  \[ dx = a(x) dt + B(x) \sigma d\omega \]

Running energy cost
\[ \text{KL}(\pi(\cdot|x), p(\cdot|x)) \quad \frac{1}{2\sigma^2} \|u\|^2 \]

Linear operator
\[ \mathcal{G}[z](x) = \int p(x'|x) z(x') \, dx' \quad \mathcal{L}[z] = a^T z_x + \frac{1}{2} \text{tr}(\Sigma z_{xx}) \]

Linear Bellman equation
\[ \exp(q(x)) z(x) = \mathcal{G}[z](x) \quad q(x) z(x) = \mathcal{L}[z](x) \]

Optimal control law
\[ \pi^*(x'|x) = \frac{p(x'|x) z(x')}{\mathcal{G}[z](x)} \quad u^*(x) = \sigma^2 B(x)^T \frac{z_x(x)}{z(x)} \]

Relation between the continuous and discrete formulations
\[ p_h(\cdot|x) = \mathcal{N}(x + ha(x); h\Sigma(x)) \]
\[ \pi_{h,u}(\cdot|x) = \mathcal{N}(x + ha(x) + hB(x)u; h\Sigma(x)) \]
\[ \text{KL}(\pi_{h,u}(\cdot|x), p_h(\cdot|x)) = \frac{h}{2\sigma^2} \|u\|^2 \]
\[ \mathcal{G}_h[z] = I + h\mathcal{L}[z] + o(h^2) \]
Comparison to dynamic programming

our framework: \[ Az = b \]

policy iteration:

evaluation: \[ A_{\pi(n)} v = b_{\pi(n)} \]

improvement: \[ \pi(n + 1) \]
Function approximation

We want to solve

$$
\lambda z (x) = \exp (-q (x)) \int p (x' | x) z (x') \, dx'
$$

Define the function approximator

$$
\tilde{z} (x; \theta, w) = \sum_i w_i f_i (x)
$$

$f_i$ are Gaussians with means and covariances contained in $\theta$. The dynamics $p$ are also Gaussian, thus the integral is computed analytically. Choose a set of collocation states $\{x_n\}$, and define the matrices $F (\theta), G (\theta)$ with elements $F_{ni} = f_i (x_n)$ and $G_{ni} = \exp (-q (x_n)) G [f_i] (x_n)$. Then the problem becomes

$$
\lambda F (\theta) w = G (\theta) w
$$

Solving for $\lambda$, $w$ is a linear eigen-problem. $\theta$ can be optimized via Gauss-Newton. The collocation set can span the entire state space, or just the region where good solutions are expected. Adaptive collocation is also possible.
Metronome example

- 40000 discrete states
- 40 adaptive bases

- $z(x)$
- $u(x)$

- $q(x)$
- $u(x)$
Basis function placement strategies

DDP/iLQG:

Spate-time optimization of limit cycles:

Approximate dynamic programming:
Consider a composite first-exit problem with final/boundary cost in the form

\[ b(x) = - \log \left( \sum_k w_k \exp(-b_k(x)) \right) \]

where \( b_k \) are the final costs for component problems whose solutions \( z_k(x) \) we already have. Then the solution to the composite problem is simply

\[ z(x) = \sum_k w_k z_k(x) \]

This yields closed-form solutions to LQG-like problems with arbitrary final costs:
Primitives from SVD of Green’s function

\[ z_I = G z_B \]

direct \hspace{1cm} R = 70 \hspace{1cm} R = 40

v(x)

\[
\begin{array}{ccc}
L & L & L \\
L & L & L \\
L & L & L \\
\end{array}
\]

z(x)

\[
\begin{array}{ccc}
L & L & L \\
L & L & L \\
L & L & L \\
\end{array}
\]

top 10 singular vectors:

\[
\begin{array}{ccc}
1 & 2 & \cdots & 10 \\
L & L & \cdots & L \\
L & L & \cdots & L \\
\end{array}
\]
Inverse optimal control

Suppose we are given a dataset of state transitions \( \mathcal{D} = \{x_n, x'_n\}_{n=1}^{N} \) generated by an optimally-controlled system. Our goal is to infer the cost function \( q(x) \) for which the system is optimal. The passive dynamics \( p(x'|x) \) are known.

This can be done by inferring \( v(x) \), computing \( z(x) = \exp(-v(x)) \), and substituting in the linear Bellman equation to obtain \( q(x) \).

It can be shown that the negative log-likelihood is

\[
L(v(\cdot)) = \sum_x a(x) v(x) + \sum_x b(x) \log \sum_{x'} p(x'|x) \exp(-v(x'))
\]

where \( b(\cdot) \) and \( a(\cdot) \) are the histograms of \( x_n \) and \( x'_n \) respectively.

\( L(v) \) is convex
Comparison to other IRL methods

<table>
<thead>
<tr>
<th></th>
<th>Error in cost function</th>
<th>CPU time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ng &amp; Russell</td>
<td>337.6</td>
<td>57.9</td>
</tr>
<tr>
<td>Abbeel &amp; Ng</td>
<td>1.0</td>
<td>3.7</td>
</tr>
<tr>
<td>Syed et al</td>
<td>???</td>
<td>0.4</td>
</tr>
<tr>
<td>Our method</td>
<td>0.000003</td>
<td>0.01</td>
</tr>
</tbody>
</table>

15-by-15 grid world
one “feature” per state

100-by-100 grid obtained by discretization

We handle 10,000 states/features in less than 1 min of CPU time
Duality with Bayesian inference

<table>
<thead>
<tr>
<th>Control</th>
<th>Continuous</th>
<th>Discrete</th>
</tr>
</thead>
<tbody>
<tr>
<td>dx = a(x) dt + B(x) (udt + dω)</td>
<td></td>
<td>xt+1 ~ u(·</td>
</tr>
<tr>
<td>ℓ(x, u, t) = q(x, t) + 1/2</td>
<td></td>
<td>u</td>
</tr>
<tr>
<td>is dual to</td>
<td></td>
<td>is dual to</td>
</tr>
<tr>
<td>dx = a(x) dt + B(x) dω</td>
<td></td>
<td>xt+1 ~ p(·</td>
</tr>
<tr>
<td>dy = h(x) dt + dv</td>
<td></td>
<td>yt ~ py(·</td>
</tr>
<tr>
<td>estimation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>q(x, t) = 1/2</td>
<td></td>
<td>h(x)</td>
</tr>
<tr>
<td>when</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Control</th>
<th>State distribution</th>
<th>Desirability function</th>
<th>Solution to transposed equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>estimation</td>
<td>posterior</td>
<td>backward filtering density</td>
<td>forward filtering density</td>
</tr>
</tbody>
</table>
Belief networks for estimation and control

Hierarchical generative models involve intermediate representations (features) which become part of the model of how sensory inputs depend on states of the world. Features are not needed to build generative models but presumably facilitate the inversion of such models (which is what Bayesian inference does).

Intermediate representations (synergies) can be incorporated in optimal control using cost functions of the form $q(h(x))$ where $h(x)$ are synergy states. By defining synergies that depend on only some aspects of the state but over extended time periods, we can achieve both spatial and temporal abstraction. The synergy states would have their own dynamics, which are learned from the plant dynamics yet have longer time scales. Unlike filtering which requires only the current and the previous time steps to be represented, control is about achieving goals that are removed in time and so it requires unfolding the time axis and representing multiple time steps. Limiting this unfolding to a fixed number of steps into the future is called receding horizon control. At the horizon $t+h$ we need some approximation of the desirability function $z(x_{t+h})$. 

(a) Hierarchical generative models involve intermediate representations (features) which become part of the model of how sensory inputs depend on states of the world.

(b) Intermediate representations (synergies) can be incorporated in optimal control using cost functions of the form $q(h(x))$ where $h(x)$ are synergy states.