Optimism in Sequential Decision-Making under Uncertainty

Peter Bartlett
Department of Statistics and Division of Computer Science
UC Berkeley

Joint work with
Ambuj Tewari.

slides at http://www.stat.berkeley.edu/~bartlett
Sequential Decision-Making under Uncertainty

Robot navigation

Chess
Sequential Decision-Making under Uncertainty

Portfolio optimization  Dynamic treatment regimes

Sequential. Statistical.
Controlling a Markov Decision Process

\( S, A = \) set of states, set of actions

\( p_s(a) = \) transition distribution (unknown)

\( r(s, a) = \) reward (unknown)

\[ V^\pi_T(s_0) = \mathbb{E} \sum_{t=0}^{T-1} r(s_t, a_t) \] total reward of policy \( \pi : S \to A \)

\[ R^\pi_T(s_0) = \sup_{\tilde{\pi}} V^\tilde{\pi}_T(s_0) - V^\pi_T(s_0) \] regret

Aim: Minimize regret
Controlling a Markov Decision Process

\[ S, A = \text{set of states, set of actions} \]
\[ p_s(a) = \text{transition distribution} \]
\[ r(s, a) = \text{reward} \]

\[ \lambda + h(s) = \max_a \left( r(s, a) + \langle p_s(a), h \rangle \right) \]

optimality equations
(linear program)
Minimizing regret: Exploration versus Exploitation

The Exploration/Exploitation Trade-off
How do we balance choosing actions that facilitate learning about the MDP (exploring, to gain more knowledge) with choosing actions that maximize average reward (exploiting knowledge we’ve gained)?
Approaches to Balancing Exploration and Exploitation

- Explicitly explore unknown territory (take the least tried actions) until we have an accurate model of the MDP.
  
  e.g., Kearns and Singh, 1998

- Implicitly explore: Be as optimistic about what we don’t know as the data allows.
  
  e.g., Burnetas and Katehakis, 1997; Auer, 2003
Algorithm: Optimistic Linear Programming

1. Model MDP (excluding “undersampled” actions).
2. Compute solution \((\hat{h}_t, \hat{\lambda}_t)\) to optimality equations
   \[
   \lambda + h(s) = \max_a \left( r(s, a) + \langle p_s(a), \hat{h} \rangle \right).
   \]
3. Choose action \(a_t\) to maximize the optimistic reward:
   \[
   U(s_t, a) = \sup \left\{ r(s_t, a) + \langle \hat{p}_{s_t}(a), \hat{h}_t \rangle : \|\hat{p}_{s_t}(a) - q\|_1 \leq \epsilon_{s_t,a,t} \right\},
   \]
   where \(\epsilon_{s,a,t}\) determines the size of a confidence set, which depends on
   how frequently \((s, a)\) has been visited.
Optimistic Linear Programming

- Optimistic about the outcomes of actions, but not unreasonably so.
- Computation involves solving linear programs.

\[ r(s, a_4) + \langle q_{opt}, \hat{h} \rangle, \quad \| \hat{p}_s(a_4) - q_{opt} \|_1 \leq \epsilon_{s,a} \]

\[ r(s, a_4) + \langle \hat{p}_s(a_4), \hat{h} \rangle \]
For the optimistic linear programming approach, the regret grows logarithmically with time $T$:

$$\limsup_{T \to \infty} \frac{R_T(s_0)}{\log T} \leq \frac{|S||A|\tau^2}{\Phi},$$

where

$\tau$ is a hitting time of the MDP under the optimal policy, and

$\Phi$ measures the gap between optimal and suboptimal actions.

The rate ($\log T$) is optimal.
Confessions and Open Problems

- Regret rate hides large transient terms.
- Dependence on $|S|$ is problematic in applications.
- Optimality relative to a restricted class of policies?
Other areas of interest

- Prediction with high-dimensional data.
  - Classification and regression with $\ell_1$ regularization.
  - Structured prediction: e.g., sequence classification, parsing.
  - Transfer learning.
- Prediction in adversarial settings.
  - Spam detection, portfolio optimization, web search.
  - Performance of statistical methods in these settings.