Fitted Value Iteration and Fitted Q-Iteration

Michael Noseworthy

April 7, 2017

Online vs. Offline Methods

- Online Methods
  - Agent learns as it interacts with the environment
  - Can update control policy at each time-step
- Offline Methods (Batch Learning)
  - Agent does not directly interact with the system
  - Input: A set of four-tuples $(s_t, a_t, r_{t+1}, s_{t+1})_i$
  - Output: Approximation to the optimal policy $\hat{\pi}^*$

Value Iteration

- Bellman Optimality Equation:

$$v^*(s) = \max_a \sum_{s',r} p(s', r|s, a) \left[r + \gamma v^*(s') \right]$$

- Recall the value iteration algorithm:

$$v_{k+1}(s) = \max_a \sum_{s',r} p(s', r|s, a) \left[r + \gamma v_k(s') \right]$$

- We will refer to this backup with the operator T

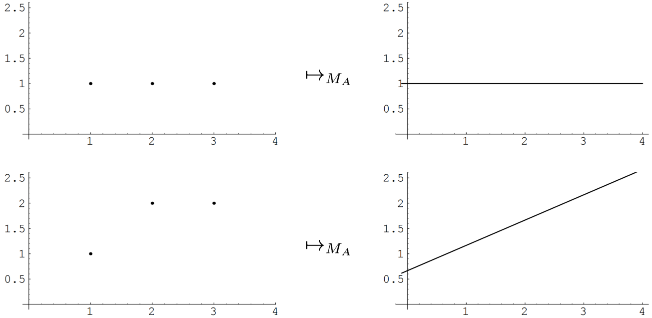
$$v_{k+1}(s) = T(v_k(s)) \tag{1}$$

Fitted Value Iteration

- As we know, DP methods like value iteration do not scale
  - We need function approximation!
- If we represent v by some function approximator, we can alternate between fitting this function, and a step of value iteration

Convergence

- We run into problems when the function approximator "exaggerates"
  - This means there is a large difference between fitted functions and only a small difference between target functions



- Neural nets and linear regression can exaggerate like this

Convergence

- More formally, an approximator exaggerates if the fitted functions $\hat{f}$ and $\hat{g}$ for input data f and g are farther apart in max-norm than f and g are
$$||\hat{g} - \hat{f}||_\infty > ||g - f||_\infty$$

Theorem (Gordon, 1999)

Let T be the parallel value backup operator for some Markov decision process M with discount $\gamma < 1$. Let A be a function approximator with mapping M. Suppose M is a nonexpansion in max norm. Then $M \circ T$ has contraction factor γ; so the fitted value iteration algorithm based on A converges in max norm at the rate γ when applied to M.

Averagers

- Approximators with the following properties are called *averagers*
  - Linearity: Each $\hat{f}(x)$ must be a linear function of the target values
  - Monotonicity: Increases a training value cannot decrease a fitted value
  - Nonexpansivity: The approximator does not exaggerate
- We can then write the fitted value at each state as:

$$k_i + \sum_{j=1}^{n} \beta_{ij} f_j \ s.t. \ \sum_{j=1}^{n} \beta_{ij} \leq 1 \ and \beta_{ij} > 0$$

- Fitted value iteration will converge with an averager for any discounted MDP (Gordon, 1995)

Fitted Q-Iteration

- We can also look at the Bellman Optimality equation for q^*:

$$Q^*(s,a) = \mathbb{E}\left[R_t + \gamma \max_{a'} Q^*(s',a') | S_t = s, A_t = a\right]$$

- Let our dataset be a collection of experience: (s_t, a_t, r_t, s_{t+1})
  - Input (x): s_t, a_t
  - Output (y): $r_t + \gamma \max_a \hat{Q}_{k-1}(s_{t+1}, a)$
- Now fit $\hat{Q}_k$ using x, y and repeat until convergence.

Optimization Horizon

- This algorithm iteratively extends the optimization horizon.
- At the first iteration, we are solving a 1-step optimization problem:

$$\hat{Q}_1 = \mathbb{E}\left[r_t | s_t = s, a_t = a\right]$$

- At the N'th timestep, we are solving an N-step optimization problem:

$$\hat{Q}_N = \mathbb{E}\left[r_t + \gamma \max_{a'} \hat{Q}_{N-1}(s_{t+1}, a') | s_t = s, a_t = a\right]$$

Approximators for Fitted-Q Iteration

- Kernel-Based Approximators
  - Ormoneit and Sen (2002)
- Tree-Based Approximators
  - Ernst, Geurts, and Wehenkel (2005)
- Neural Fitted-Q
  Riedmiller (2005)