INSE6300	Jia Yuan Yu

4: Dynamic programming

Concordia	February 16, 2016

First, a visual shortest path example: `http://web.mit.edu/15.053/www/AMP-Chapter-11.pdf`.

1 Examples of backward induction

The backward induction algorithm for MDPs proceeds as follows.

1. Set $j = N$, and $V_N(s) = \max_{a \in A} r_N(s,a) = g(s)$ for all $s \in S$;
2. For $j = N-1, N-2, \ldots, 1$:
   (a) For $s \in S$:
      i. Compute
$$V_j(s) = \max_{a \in A} \left\{ r_j(s,a) + \sum_{s' \in S} P(s' \mid s,a) V_{j+1}(s) \right\};$$
      ii. Output $\sigma_j(s) \in \arg\max_{a \in A} \left\{ r_j(s,a) + \sum_{s' \in S} P(s' \mid s,a) V_{j+1}(s) \right\}$.

The output of this algorithm is a sequence of *policies* $\sigma_1, \ldots, \sigma_N$ that are optimal (cf. Puterman, Section 4.3).

1.1 Intuition

We need to make inventory decision $a_1, a_2, \ldots, a_{N-1}$ for time steps $1, \ldots, N-1$. Why does backward induction work? Consider the time step $N-1$: you observe the value of the inventory level (state) s_{N-1}, which takes possible values $\{0, 1, \ldots, C\}$, and you take the last decision a_{N-1} according to the actual value of s_{N-1}:

$$a_{N-1}(0) \in \arg \max_{a=0,\ldots,C} \overbrace{\underbrace{r(0,a)}_{\text{immediate reward at time } N-1} + \sum_{j=0}^{C} \mathbb{P}(s_N = j \mid s_{N-1} = 0, a_N = a) \underbrace{g(j)}_{\text{salvage at time } N}}^{V_{N-1}(0)},$$

$$\ldots$$

$$a_{N-1}(C) \in \arg \max_{a=0,\ldots,C} \overbrace{\underbrace{r(C,a)}_{\text{immediate reward at time } N-1} + \underbrace{\sum_{j=0}^{C} \mathbb{P}(s_N = j \mid s_{N-1} = C, a_N = a) g(j)}_{\text{Expected salvage } \mathbb{E}g(s_N)}}^{V_{N-1}(C)}.$$

Consider time step $N-2$: you observe s_{N-2}, and take decision a_{N-2}, then observe s_{N-1} at time step $N-1$ and take action a_{N-1}. The total future reward is

$$r(s_{N-2}, a_{N-2}) + r(s_{N-1}, a_{N-1}) + g(s_N).$$

Recall that

- we can optimize the expected value of $r(s_{N-1}, a_{N-1}) + g(s_N)$ by selecting a_{N-1} as a function of s_{N-1};
- having observed $s_{N-2} = i$, we known the distribution of s_{N-1}, and s_N;
- having observed $s_{N-2} = i$, we can optimize the expected future reward through the function a_{N-1} above and:

$$a_{N-2}(i) \in \arg\max_{a=0,\dots,C} \quad r(i,a) + \underbrace{\mathbb{E}\Big[r(s_{N-1}, a_{N-1}(s_{N-1})) + g(s_N)\Big]}_{\sum_{j=0}^{C} \mathbb{P}(s_{N-1}=j|s_{N-2}=i, a_{N-2}=a) V_{N-1}(j)},$$

  so that a_{N-2} is only a function of i and $\mathbb{P}$ and r and g.

1.2 Yield management example

Airline with a single flight. The time horizon is $1, \dots, T$. The state represents the number of seats remaining on the flight. At each time step t, a customer appears with probability λ. The decision of the airline is the price a_t, which takes values $v_1, \dots, v_n$. The probability that the customer t purchases a ticket is a function of a_t.

What is the expected revenue at each time step? What are the state transition probabilities?

What would happen if customers are allowed to cancel their purchases?

1.3 Portfolio management

Two types of assets: a liquid asset with a fixed interest rate, which may be sold at every time step, and a non-liquid asset that may only be sold after a maturity of N time steps. The state is a vector in R^{N+1}, the fraction of investment in the liquid asset, and in non-liquid assets with maturity $1, \dots, N$ steps away. The decision maker can choose to move a fixed fraction α of liquid asset into non-liquid assets.

2 References

- Pricing Substitutable Flights in Airline Revenue Management, D. Zhang and W. L. Cooper, 2006.