Tutorial:

The Curse of Dimensionality and How to Mitigate it in Dynamic Programming Applications

Simon Haykin
McMaster University
Hamilton, Ontario, Canada
email: haykin@mcmaster.ca
Website: http://soma.mcmaster.ca

Dynamic Programming and The Curse-of-Dimensionality

- **Bellman's dynamic programming is a technique that deals with situations where decisions are made in stages, with the outcome of each decision being predictable to some extent before the next decision is made.**
- **For decision-making in a stochastic environment, long-term performance of the dynamic system (decision-maker) is improved by sacrificing short-term performance for the purpose of attaining a desired long-term goal.**

Practical Considerations

(i) **Bellman's dynamic programming (DP) requires knowledge of transition probabilities of the dynamic system from one state to the next:**

This requirement is not realistic in practice

(ii) **Curse of dimensionality**

Computational complexity of the DP algorithm increases exponentially with dimensionality of the state, which makes it impractical in large-scale applications.

Given these two realities, we are therefore compelled to seek approximate forms of Bellman's dynamic programming.

The Curse of Dimensionality according to Richard Bellman[1]

"In view of all that we have said in the foregoing sections, the many obstacles we appear to have surmounted. What casts the pall over our victory celebration? It is the curse of dimensionality, a malediction that has plagued the scientist from earliest days."

[1.] **R. Bellman, Adaptive control Processes, p.94, Princeton University Press, NJ, 1961.**

Approximate Dynamic Programming[2]

(i) Direct Approximation, based on Monte Carlo simulation for computing the required transition probabilities, exemplified by:

The *Q*-learning algorithm due to Watkins (1989)

This algorithm for control applications is restricted to dynamic systems with low to moderate state-space dimensionality

(ii) Indirect approximation, involving the combined use of projection theory and least-squares estimation, exemplified by:

The Least-squares Policy Iteration algorithm due to Bertsekas et al. (2004)

---

2. Dimitri Bertsekas, "Dynamic Programming and Optimal Control", Vol. 1 (2005); and vol. 2(2007), Athena Scientific.