Linear Programming approach to approximate Dynamic Programming

1 Main Ideas

In large-scale stochastic control problems, the curse of dimensionality is problematic. We are dealing with a state-action space of scale ∞^2. So, finding the value function (or state value function) is infeasible, since the tabular representation will fail. To address this problem, we try to represent the value function using a linear combination of some "basis functions". The algorithm proposed here uses linear programming in order to find the weights associated with each basis functions.

Another important contribution of this paper is the boundary that they proposed (equation 1) that limtis their approximation. Although the bound is not very promising, and does not guaranty a full stability, it's a good in a sense that the algorithm will, at least, find a solution.

$$\|J^* - \Phi\bar{r}\|_{1,c} \leq \frac{2}{1-\alpha} \min_r \|J^* - \Phi r\|_\infty \tag{1}$$

2 Questions

1. If the optimal value function is one of the features, will ALP compute the optimal value function?

Based on equation 1, the for error (the distance between the ALP solution and the real optimum of J) is always bound with the worst element in $\|J^* - \Phi r\|$. This is actually what an $\|.\|_\infty$ means. So, even in case of choosing the optimal value function as a feature, we still get a different solution.

2. How does the computational complexity of ALP compare to LSPI? Which one is better?

In general, ALP uses linear programming. So, the time complexity is polynomial of the problem size, which is $|\mathcal{S}||\mathcal{A}|$. However, the complexity of LSPI is $\mathcal{O}(k * m)$, in which m is the number of policies we consider to study and k is the number of samples we have.