Robust Approximate Optimization for Large Scale Planning Problems

Marek Petrik
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
petrik@cs.umass.edu

1 Overview

I address a general class of optimization problems known as *sequential decision making*, which involves maximizing the expected utility of a sequence of actions taken over some finite or infinite period of time. Rich formal models for sequential decision making already exist. The Markov Decision Process (MDP), in particular, has been shown to provide a very effective framework with several advantages. The framework is extremely general in terms of its expressive power – almost every problem can be mapped easily into an MDP (or one of its variants). MDPs thus offer a tradeoff between the complexity of modeling and the computational time required to solve the problem.

Solving sequential decision problems is crucial in a wide range of domains. Within AI, examples of such problems include mobile robot control for space exploration, helicopter control, visual tracking and gesture recognition, and recommender systems. Within operations research, relevant problems include machine maintenance, scheduling aircraft landings, nuclear power-plant management, and managing blood inventories [Powell, 2007]. While these problems are easy to formulate, they can be extremely hard to solve.

There are two significant obstacles in solving practical problems formulated as MDPs. First, the size of many real-world problems tends to be very large. This is generally attributed to the "curse of dimensionality" – the exponential growth of the size of the state space as new features are added to the model. Second, a precise model of the problem is rarely available at the design time, or the problem may change frequently after the initial model is created. Both issues can be addressed by solving the problems approximately, which is often sufficient in practice. As a result the recent research focuses on solving MDPs approximately by **approximate dynamic programming**.

Approximate dynamic programming (ADP) approximates the value function by constraining it to a small subspace generated by a **basis**. The value function is then calculated based on samples of the domain [Powell, 2007]. This value function approximation not only simplifies the problem but also increases the robustness of the policy with regard to missing data. Therefore, an important issue in choosing the basis is the tradeoff between the precision and certainty. This can be seen as the bias-variance tradeoff studied in machine learning.

The main reasons that prevents a wide-spread use of ADP in practice is that applying the existing methods often requires: 1) deep understanding of the domain and 2) thorough tweaking of the parameters [Powell, 2007]. These issues are typically due to a large approximation error. Therefore, to develop robust algorithms, it will be necessary to understand the cause of the large approximation error in common settings, and to develop automatic methods for reducing it.

The approach I take in my thesis is based on **approximate linear programming** (ALP) [de Farias, 2002], in which the MDP is formulated as a linear program. This linear program is then solved approximately by restricting the value function to a small subspace and sampling only a subset of all the constraints. I chose approximate linear programming because it offers better theoretical convergence properties than other ADP algorithms [de Farias, 2002]. Despite the good theoretical properties, ALP often does not perform well in practical problem. However, the current understanding of the method is limited, and therefore I believe there is potential for a significant improvement. Even if ALP does not outperform other ADP algorithms, it is an important algorithm since it can be used to calculate an admissible heuristic functions [Petrik and Zilberstein, 2008].

The specific goals I want to achieve in my thesis are the following:

1. Identify the main challenges faced in practical applications of ALP. These results can guide the research in improving the important properties of ALP.
2. Develop well-motivated constraint sampling methods for ALP and reinforcement learning problems in general. The new methods will be based on sampling bounds, unlike the current methods, which are quite arbitrary [de Farias, 2002].
3. Develop methods for generating the basis in some general contexts. While it is impossible to develop a good method that works in all contexts, I will concentrate on specific problem classes, such as resource management problems.

Accomplishment of these goals will bring a widespread practical use of ADP methods closer to reality. For example, given a robust method, researchers and practitioners outside of AI will be able to leverage ADP to solve their problems.

2 Results So Far

In practice, it is important that algorithms do not only return good solutions, but also that they can assess the quality of the solution. In ADP, the solution quality can be determined by calculating the error bound on the value function. The existing error bounds, as I showed, can be inadequate in many settings [Petrik and Scherrer, 2009]. This in particular occurs in MDPs with a discount factor close to 1. I have derived new tighter bounds, based on the results on aggregation in linear programming.

I have studied an application of ALP to a practical blood inventory management problem [Powell, 2007] and identified the key issues with the approximation error [Petrik and Zilberstein, 2009a]. To better structure the analysis, I divide the approximation error into three components based on its cause. The first part is the **representational error**, which is the fundamental error caused by limiting the value function to the approximation subspace. It is the error between the optimal value function and its closest possible approximation in the basis. The second part is the **transitional error**, which results from the ALP formulation. ALP is not guaranteed to find the best possible approximation of the optimal value function. The third part is the **sampling error**, caused by considering only a subset of the constraints in the linear program.

One of the main reasons for the poor empirical performance of ALP is a very large **transitional error**. In particular, this happens when the approximation creates a virtual loop; that is a transition between states that have very similar approximation features. A possible approach to reducing this type of error is to prevent such transitions by requiring specific structures in the approximation basis [de Farias, 2002; Petrik and Zilberstein, 2008]. These structures often require deep analysis of the domain, which is impractical in many settings. I have therefore recently developed a method that can reduce this type of error by automatically relaxing certain constraints. A demonstration of these methods' effect for a simple chain problem is shown in Figure 1. Here v^* is the optimal solution, v_{alp} is the solution of ALP, and v_1 and v_2 are solutions of ALP with two proposed types of constraint relaxations [Petrik and Zilberstein, 2009b]. Notice that the approximate value function is an upper bound on the true value function, and therefore needs to be minimized. The constraint relaxation method leads to significant reduction in the approximation error in other domains as well, including blood inventory management, and other reinforcement learning benchmark problems.

Finally, to reduce the **representational error**, it is necessary to design a good approximation basis. In [Petrik, 2007], I have proposed a new method for automatically creating the approximation used basis with ADP, which constitutes the basis selection problem. The method outperforms some other procedures on simple benchmark problems, but cannot guarantee that the basis improves in every iteration. Therefore, using the convergent properties of approximate linear programming, I developed a basis selection method that is guaranteed to eventually converge to the optimal solution. This method still requires that the model is completely specified, and thus needs to be extended to the general sampled problems.

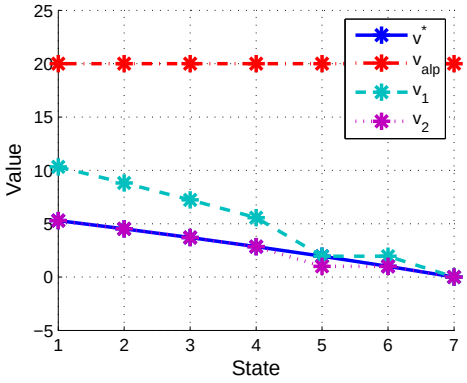

Figure 1: Value functions on a simple chain problem.

3 Work Plan

I will continue to extend the analysis of the approximation error in ALP and propose methods for reducing the error. In particular, I will concentrate on the following issues:

1. Further analyze the performance of approximate dynamic programming methods in large-scale problem settings, such as the blood inventory management problem.
2. The application of ALP to the blood inventory management pointed to a large sampling error, which stressed the importance of good constraint sampling methods. To develop the sampling methods, I will derive tighter sampling error bounds.
3. The representational error in ALP applications may be also very large, since the solution is limited to a linear combination of the approximation features. I will extend ALP beyond linear combinations of features using regularization.

[de Farias, 2002] Daniela P. de Farias. *The Linear Programming Approach to Approximate Dynamic Programming: Theory and Application*. PhD thesis, Stanford University, 2002.

[Petrik and Scherrer, 2009] Marek Petrik and Bruno Scherrer. Biasing approximate dynamic programming with a lower discount factor. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

[Petrik and Zilberstein, 2008] Marek Petrik and Shlomo Zilberstein. Learning heuristic functions through approximate linear programming. In *International Conference on Automated Planning and Scheduling (ICAPS)*, 2008.

[Petrik and Zilberstein, 2009a] Marek Petrik and Shlomo Zilberstein. Blood management using approximate linear programming. Presented at INFORMS Computing Society Meeting, January 2009.

[Petrik and Zilberstein, 2009b] Marek Petrik and Shlomo Zilberstein. Constraint relaxation in approximate linear programs. In *International Conference on Machine Learning*, 2009.

[Petrik, 2007] Marek Petrik. An analysis of Laplacian methods for value function approximation in MDPs. In *International Joint Conference on Artificial Intelligence*, pages 2574–2579, 2007.

[Powell, 2007] Warren B. Powell. *Approximate Dynamic Programming*. Wiley-Interscience, 2007.