Rates of Convergence for Variable Resolution Schemes in Optimal Control

Rémi Munos MUNOS@CS.CMU.EDU
Andrew W. Moore AWM@CS.CMU.EDU
Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213 USA.
http://www.cs.cmu.edu/~*AUTON*/

Abstract

This paper presents a general method to derive tight rates of convergence for numerical approximations in optimal control when we consider variable resolution grids. We study the continuous-space, discrete-time, and discrete-controls case. Previous work described methods to obtain rates of convergence using general or linear approximators (Bertsekas & Tsitsiklis, 1996; Tsitsiklis & Van Roy, 1996; Gordon, 1999), multi-grids (Chow & Tsitsiklis, 1991), random or low-discrepancy grids (Rust, 1996). These results provide bounds on the error on the value function in terms of the representation power of the class of approximators considered, thus for uniform grids, in terms of the space discretization resolution (or the number of grid-points). Consequently, they do not explicitly consider the benefit of using non-uniform resolutions. However, empirical results (Munos & Moore, 1999b) have shown the importance of using variable resolution discretizations, especially for problems with high-dimensional state-spaces (in order to attack the "curse of dimensionality"). This paper provides some bounds on the approximation error of the value function in terms of the local interpolation error. Additionally, we are able to predict the effect that locally increasing the grid-resolution has on the quality of approximation of the value function, thus opening a way for designing efficient grid-refinement procedures. This analysis can be applied to stochastic or deterministic problems and can be used with many function approximators, including grid interpolators based on kd-trees, multi-grids, random, or low-discrepancy grids.

1. Introduction

This paper has two contributions. The first is a new technique for safely bounding the error caused by value function approximation. The second contribution exploits the first in order to decide where to increase the resources of the value function approximator.

This paper introduces new notational and formal frameworks for handling these issues, and a side-effect is unavoidable denseness. To help the reader through this development of concepts, we will now provide a road-map summary of the questions each part of the paper are answering:

- **What kinds of value function approximators is this applicable to?** (Section 2.1) This is applicable to Markov Decision Processes (MDPs) in which the state space is too large to be tabulated explicitly and is thus approximated with a discretized MDP. The classic example is continuous state space. The paper is applicable to any discretization in which some finite set of points $\{x_1, x_2, \ldots x_N\}$ represent the states of an MDP, and their transition probabilities are created by a combination of interpolation and discretization of the next state probability distribution. Examples include simple binning of a state space, averagers, piecewise linear or multi-linear interpolations, random or low-discrepancy grids, either in uniform resolution or variable resolution lattices.
- **How is the error of approximation defined?** (Section 2.2) There are two error expressions. The *local interpolation error* is relatively easy to estimate directly from the form of the value function and the specifics of the grid interpolation. It measures the extent to which the approximate backup operator is unable to represent the value function even if all next points were seeded with the correct values. The *approximation error* is the error we want to minimize, and is harder to estimate

directly. It measures the difference between the true and the approximate value functions.

- **What computation is used to bound this error?** (Section 2.3) We first show how a global bound on the interpolation error can provide a loose bound on the approximation error. But then we describe a procedure by which the loose bound can prove that certain actions in some states cannot possibly form part of the true optimal policy. Then a computation with the restricted set of possible actions can tighten the approximation error bounds. This process can be repeated to create tighter and tighter bounds until no further actions can be eliminated.
- **How does one state influence another?** (Section 3.2) We then briefly review earlier work defining the extent to which one state "contributes" to the value function of another state. This concept is needed for the next question.
- **How should we choose where to add resources if we want to improve the accuracy of the approximate value function at some area of the state space?** (Section 3.3) If we have a set of start states (possibly the whole state space) for which we want to increase the accuracy of the approximate value function, we can use the above analysis to decide where to increase the resolution. Unsurprisingly it turns out that the area most deserving of an increase is *not* necessarily near the start states.
- **What if we do not care about the value function, but only the policy?** Usually the policy, not the value function is the main goal of an optimal control problem. The above analysis can be extended to provide some bounds on the loss to occur if we follow an approximate (sub-optimal) policy. Moreover, we can deduce the parts of the state space where an increase of the resolution will significantly reduce this loss. Because of space limitations, this will be further developed in a future work.

2. Analysis of Error Bounds

2.1 Grid Approximations

We consider a Markov Decision Process (MDP) with a continuous state-space X and a discrete control-space U. The time is discrete and the probabilities of transition from state x, control u to next states y are $p(y|x,u)$. We consider the problem of finding the control that maximizes the expected sum of discounted rewards $r(x,u)$. For any function W, we introduce the operators:

$$\Gamma_u W(x) = \gamma \int_X p(y|x,u).W(y)dy$$

$\Gamma_u W(x)$ is the discounted expected next values (with a discount factor $\gamma < 1$) of the W function if we apply action u in state x.

$$\Gamma W(x) = \max_{u \in U} \Gamma_u W(x)$$

$\Gamma W(x)$ is the largest available discounted expected next values of W at x.

$$T_u W(x) = \Gamma_u W(x) + r(x,u)$$

$T_u W(x)$ is similar to $\Gamma_u W(x)$, except it incorporates the immediate reward.

$$TW(x) = \max_{u \in U} T_u W(x)$$

T is the *Bellman operator*. It is a contraction operator and its fixed point V is called the **value function**, and satisfies the *dynamic programming* (DP) *equation* $V(x) = TV(x)$. The **optimal control** $u^*(x)$ is the argument of $\max_{u \in U} T_u V(x)$.

In the deterministic case, when there is a unique successor $y(x,u)$ of (state x, control u), the operator $\Gamma_u W(x)$ is simply $\gamma W(y(x,u))$.

This paper concerns approximate MDPs built from a finite grid-representation X_N composed of N points $\{x_i\}_{i=1..N}$. Those representations include averagers (Gordon, 1999) such as kernel regression, piecewise linear or multi-linear interpolations (Davies, 1996; Munos & Moore, 1998) within triangular or hypercuboid cells, random and quasi-random grids (Rust, 1996) either in uniform or variable resolution. For a given grid, each of the above operators has an approximate equivalent, respectively Γ_u^N, Γ^N, T_u^N, and T^N:

$$\begin{aligned}
\Gamma_u^N W(x) &= \gamma \sum_{i=1}^{N} p_N(x_i|x,u).W(x_i) \\
\Gamma^N W(x) &= \max_{u \in U} \Gamma_u^N W(x) \\
T_u^N W(x) &= \Gamma_u^N W(x) + r_N(x,u) \\
T^N W(x) &= \max_{u \in U} T_u^N W(x)
\end{aligned}$$

that use some approximate transition probabilities p_N and reward function r_N. The **approximate value function** V^N is the solution to the DP equation $V^N(x) = T^N(V^N)$. The **approximate optimal control** $u_N^*(x)$ is the argument of $\max_{u \in U} T_u^N V^N(x)$.

2.2 Interpolation and Approximation Errors

This paper is about estimating the error in the value function caused by the use of finite grids. The first important point is that there are two kinds of errors, and we begin by defining and distinguishing them. The **local interpolation error** of the approximate operator on the value function is:

$$\begin{aligned} e_u^N(x) &= |T_u^N V(x) - T_u V(x)| \\ \text{and:} \quad e^N(x) &= |T^N V(x) - TV(x)| \end{aligned}$$

which measure the immediate local error in doing a backup of the true value function. Next, we introduce the **approximation error**:

$$\begin{aligned} \varepsilon_u^N(x) &= |T_u^N V^N(x) - T_u V(x)| \\ \text{and:} \quad \varepsilon^N(x) &= |T^N V^N(x) - TV(x)| = |V^N(x) - V(x)| \end{aligned}$$

which measures the difference between the approximately backed up approximate value function and the exactly backed up true value function.

Previous work (Chow & Tsitsiklis, 1991; Bertsekas & Tsitsiklis, 1996; Rust, 1996; Tsitsiklis & Van Roy, 1996; Gordon, 1999) have considered global bounds for general function approximators, that could be used for variable resolution (VR) grids only in a worst-case analysis in which we consider the lowest resolution of the grid. This paper proposes a method to estimate tight bounds on the error of approximation of the value function for VR grids.

In Section 2.3 we provide bounds on the approximation error ε^N in terms of the interpolation error e^N. Next, we estimate in Section 3 the non-local dependencies in the value function estimations. Then, we illustrate in Section 4 how the interpolation error can be derived from the local curvature of the value function and the specifics of the grid interpolation. Therefore, we are able to deduce the approximation error for a given VR grid.

Table 1 summarizes some useful notation that will be used in the following discussion.

2.3 Bounds on the Approximation Error of the Value Function

In this section, we give bounds on the approximation error ε^N in terms of the interpolation error e^N. Section 2.3.1 shows that this bound satisfies a DP equation (Equation (2) below) and Section 2.3.2 shows that the bound is the expected sum of the discounted interpolation errors obtained along a Markov process.

Table 1. Some useful notation

DP equation:	$V = TV$ (real)
	$V^N = T^N V^N$ (approximate)
Optimal control:	$u^* = \arg\max_{u\in U} T_u V$ (real)
	$u_N^* = \arg\max_{u\in U} T_u^N V^N$ (approx.)
	$v_N^* = \arg\max_{u\in U} T_u^N V$ (virtual)
	$u_\varepsilon^* = \arg\max_{u\in U_N} \Gamma_u^N \overline{\varepsilon^N}$
	$u_e^* = \arg\max_{u\in U'_N} e_u^N$
Error:	e^N (local interpolation)
	ε^N (approximation error)
	$\overline{\varepsilon^N}$ (bound on ε^N)
Global error:	$E_\Omega(X_N)$ for grid X_N
	$\overline{E}_\Omega^N$ (bound on $E_\Omega(X_N)$)

2.3.1 DP representation of error bounds

The error of approximation satisfies the following inequality:

$$\begin{aligned} \varepsilon^N(x) &= |T^N V^N(x) - TV(x)| \\ &\le |T^N V^N(x) - T^N V(x)| + |T^N V(x) - TV(x)| \\ &\le |T^N V^N(x) - T^N V(x)| + e^N(x) \end{aligned}$$

Thus, it is easy to derive *global rates of convergence*. Indeed, let e^N be a global bound for $e^N(x)$. Since Γ_u^N is a contraction operator of (max-) norm γ, we have: $\varepsilon^N(x) \le \max_u |T_u^N \left[V^N - V\right](x)| + e^N = \max_u |\Gamma_u^N \varepsilon^N(x)| + e^N \le \gamma||\varepsilon^N|| + e^N$. Thus ε^N is bounded by $\frac{1}{1-\gamma} e^N$.

However, we can do much better than that if we consider local instead of global bounds. Indeed we have (using the simplified notation u instead of $u(x)$ when there is no possible confusion):

$$\begin{aligned} |T^N V^N(x) - T^N V(x)| &= |T_{u_N^*}^N V^N(x) - T_{v_N^*}^N V(x)| \\ \text{and:} \quad e^N(x) &= |T_{v_N^*}^N V(x) - T_{u^*} V(x)| \end{aligned}$$

where u^* and u_N^* (defined above) are respectively the optimal control (derived from V) and the approximate optimal control (derived from V^N), and $v_N^*(x)$ is defined as the argument of $\max_u T_u^N V(x)$. This latter does not have any physical interpretation; instead, it represents the control that would optimize the next values based on the real value function V, but using the approximate operator T^N. We introduce this **virtual control** v_N^* for convenience. We thus obtain:

$$\begin{aligned} \varepsilon^N(x) &\le |T_{u_N^*}^N V^N(x) - T_{v_N^*}^N V(x)| + |T_{v_N^*}^N V(x) - T_{u^*} V(x)| \\ &\le \max_{u\in\{u_N^*, v_N^*\}} \left\{T_u^N [V^N - V](x)\right\} + \max_{u\in\{v_N^*, u^*\}} e_u^N(x) \end{aligned}$$

$$\varepsilon^N(x) \le \max_{u \in \{u^*_N, v^*_N\}} \{\Gamma^N_u \varepsilon^N(x)\} + \max_{u \in \{v^*_N, u^*\}} e^N_u(x)$$

This inequality can be used to derive several DP equations whose solutions give bounds on ε^N. Basically, we can derive two obvious bounds depending on the knowledge we have of the approximate optimal control:

- Let us assume we do not know anything about the optimal control u^* and the virtual control v^*_N. Then the best bound $\overline{\varepsilon^N}$ on ε^N (i.e. satisfying: $\varepsilon^N \le \overline{\varepsilon^N}$) is obtained by solving the DP equation:

$$\overline{\varepsilon^N}(x) = \max_{u \in U} \left\{\Gamma^N_u \overline{\varepsilon^N}(x)\right\} + \max_{u \in U} e^N_u(x)$$

- On the other hand, if we know that the controls u^*, u^*_N, and v^*_N are identical, then a tighter bound $\overline{\varepsilon^N}$ satisfies the Bellman equation:

$$\overline{\varepsilon^N}(x) = \Gamma^N_{u^*_N} \overline{\varepsilon^N}(x) + e^N_{u^*_N}(x) \qquad (1)$$

Now, in general, if we are uncertain about the virtual control v^*_N and the optimal control u^* but we know that there exists two subsets $U_N(x)$ and $U'_N(x)$ of U such that $v^*_N(x) \in U_N(x)$ and $u^*(x) \in U'_N(x)$, then a bound $\overline{\varepsilon^N}$ is obtained by solving the DP equation:

$$\overline{\varepsilon^N}(x) = \max_{u \in U_N(x)} \left\{\Gamma^N_u \overline{\varepsilon^N}(x)\right\} + \max_{u \in U'_N(x)} e^N_u(x) \qquad (2)$$

Of course, the smaller the subsets U_N and U'_N the tighter the bound obtained.

A way of choosing U_N and U'_N is to already have a first bound $\overline{\varepsilon^N}$ on the error of approximation. Indeed, we can choose $U_N(x)$ to be the set of controls u such that the difference between the expected values $T^N_{u^*_N} V^N(x)$ obtained by choosing the approximate optimal control $u^*_N(x)$ and the expected values $T^N_u V^N(x)$ obtained by choosing another control u is less than the sum $\Gamma^N_{u^*_N} \overline{\varepsilon^N}(x) + \Gamma^N_u \overline{\varepsilon^N}(x)$ of the expected errors of approximation (see figure 1):

$$U_N(x) = \Big\{u \in U \,\Big|\, T^N_{u^*_N} V^N(x) - \Gamma^N_{u^*_N} \overline{\varepsilon^N}(x) \le T^N_u V^N(x) + \Gamma^N_u \overline{\varepsilon^N}(x)\Big\} \qquad (3)$$

And similarly, we define:

$$U'_N(x) = \Big\{u \in U \Big| T^N_{u^*_N} V^N(x) - \Gamma^N_{u^*_N} \overline{\varepsilon^N}(x) - e^N_{u^*_N}(x) \le T^N_u V^N(x) + \Gamma^N_u \overline{\varepsilon^N}(x) + e^N_u(x)\Big\} \qquad (4)$$

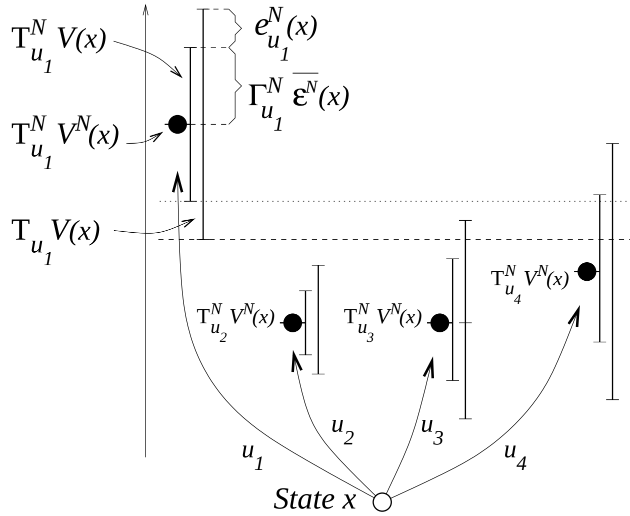

Figure 1. Illustration of the sets $U_N(x)$ and $U'_N(x)$ on a simple example. The black dots represent the values $T^N_{u_i} V^N(x)$ for several controls $u_1, ..., u_4$. Control u_1 provides the highest value thus $u_1 = u^*_N(x)$. On the right side of each dot, the first vertical line represents the possible value of $T^N_{u_i} V(x)$, knowing the bound on the error $\Gamma^N_{u_i} \overline{\varepsilon^N}(x)$. The second line represents the possible value of $T_{u_i} V(x)$, knowing the local interpolation error $e^N_{u_i}(x)$. In this example, the control u_2 is neither in $U_N(x)$ nor in $U'_N(x)$ (defined in (3) and (4)), which is illustrated graphically by the second vertical line being under the bottom horizontal dashed line. Thus, we are sure that u_2 is not the optimal control $u^*(x)$. Now, u_3 is not in $U_N(x)$ (the first vertical line is under the top horizontal dashed line) but is in $U'_N(x)$ (the second vertical line goes over the second horizontal dashed line). Thus u_3 might be the optimal control but is definitely not the virtual control $v^*_N(x)$. A similar reasoning shows that $u_4 \in U_N(x)$. Thus, in this example, we have: $U_N(x) = \{u_1, u_4\}$ and $U'_N(x) = \{u_1, u_3, u_4\}$.

It is easy to show that we have the property:

$$u^*_N(x) \in U_N(x) \subset U'_N(x) \subset U$$

Moreover, the theorem that follows states that for a given bound $\overline{\varepsilon^N}$ on the error of approximation, this is a safe way of choosing U_N and U'_N.

Theorem. *The virtual optimal control $v^*_N(x)$ is in $U_N(x)$ and the optimal control $u^*(x)$ is in $U'_N(x)$, where U_N and U'_N are defined in (3) and (4).*

Proof. Since $\overline{\varepsilon^N}(x)$ is an upper bound of $|V^N(x) - V(x)|$, we deduce that: $T^N_{u^*_N} V^N(x) - T^N_{u^*_N} V(x) \le \Gamma^N_{u^*_N} \overline{\varepsilon^N}(x)$ and $T^N_{v^*_N} V(x) - T^N_{v^*_N} V^N(x) \le \Gamma^N_{v^*_N} \overline{\varepsilon^N}(x)$. But, from the definition of v^*_N, we have: $T^N_{u^*_N} V(x) \le$

$T^N_{v^*_N(x)}V(x)$. Thus:

$$\begin{aligned} T^N_{u^*_N}V^N(x) - \Gamma^N_{u^*_N}\overline{\varepsilon^N}(x) &\leq T^N_{u^*_N}V(x) \\ \leq T^N_{v^*_N}V(x) &\leq T^N_{v^*_N}V^N(x) + \Gamma^N_{v^*_N}\overline{\varepsilon^N}(x) \end{aligned}$$

and $v^*_N(x) \in U_N(x)$.
Additionally, from the definition of e^N_u, we have: $T^N_{u^*_N}V(x) - T_{u^*_N}V(x) \leq e^N_{u^*_N}(x)$ and $T_{u^*}V(x) - T^N_{u^*}V^N(x) \leq e^N_{u^*}(x)$. Then, from the definition of $\overline{\varepsilon^N}(x)$, we have: $T^N_{u^*}V(x) - T^N_{u^*}V^N(x) \leq \Gamma^N_{u^*}\overline{\varepsilon^N}(x)$. Finally, from the definition of u^*, we have $T_{u^*_N}V(x) \leq T_{u^*}V(x)$ from which we deduce:

$$\begin{aligned} T^N_{u^*_N}V^N(x) - \Gamma^N_{u^*_N}\overline{\varepsilon^N}(x) - e^N_{u^*_N}(x) &\leq T^N_{u^*_N}V(x) - e^N_{u^*_N}(x) \\ \leq T_{u^*_N}V(x) \leq T_{u^*}V(x) &\leq T^N_{u^*}V(x) + e^N_{u^*}(x) \\ &\leq T^N_{u^*}V^N(x) + \Gamma^N_{u^*}\overline{\varepsilon^N}(x) + e^N_{u^*}(x) \end{aligned}$$

and $u^*(x) \in U'_N(x)$. □

So from an initial bound on the approximation error of the value function, we build the safe sets U_N and U'_N from which to choose the control from when we solve the DP equation (2). The solution to this equation provides us with a tighter bound $\overline{\varepsilon^N}$, which, in turn, can be used to define more restricted subsets U_N and U'_N. We can repeat this process until the newly created subsets U_N and U'_N do not change anymore. We conjecture that the error bound thus obtained is the best possible safe bound for a given grid approximation.

An efficient way to compute it without having to loop several times between estimations of bounds and of subsets U_N and U'_N is the following:

In the process of solving the DP equation (2) –let's say we use regular value iteration– we start with initial errors $\overline{\varepsilon^N_0} = \infty$ (thus $U_N = U'_N = U$) and apply the iterative rule $\overline{\varepsilon^N_{n+1}}(x) = \max_{U_N}\left\{\Gamma^N_u\overline{\varepsilon^N_n}(x)\right\} + \max_{U'_N} e^N_u(x)$ in which the subsets U_N and U'_N are updated (accordingly to $\overline{\varepsilon^N_n}$) at every iteration. We can prove that $\overline{\varepsilon^N_n}$ is a decreasing function of step n, thus once a control has been removed from the subsets U_N and U'_N it will never be used again. This algorithm is reminiscent of the action elimination procedure described in (Puterman, 1994).

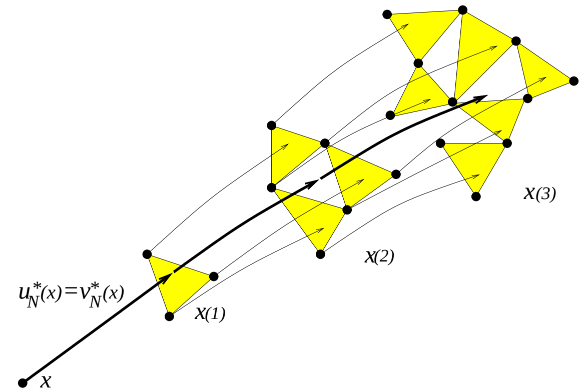

Figure 2. Example of a deterministic process. The optimal trajectory starting from x is shown in bold. The local error e^N made when interpolating the value $W(y)$ at a point y by a weighted linear combination $\sum_i p(x_i|x,u)W(x_i)$ of values at grid-points is graphically illustrated by the gray areas. In the case of $u^*_N = u^* = v^*_N$ the error of approximation $\overline{\varepsilon^N}(x)$ is the expected sum of the discounted interpolation errors $\gamma^k e^N(x(k))$ when $x(k)$ follows the Markov chain: $x(k) \to x(k+1)$ with probability $p_N(x(k+1)|x(k), u^*_N(x(k)))$.

2.3.2 Markov representation of the error bound

From the DP equation (2), we deduce the Markov representation:

$$\overline{\varepsilon^N}(x) = E\left[\sum_k \gamma^k e^N(x(k)) \,\middle|\, \begin{array}{l} x(0) = x \\ u(k) = u^*_\varepsilon(x(k)) \end{array}\right] \quad (5)$$

where $x(k)$ follows the Markov process: $x(k) \to x(k+1)$ with probability $p_N(x(k+1)|x(k), u^*_\varepsilon(x(k)))$, with $u^*_\varepsilon(x)$ being the argument of $\max_{u\in U_N(x)}\Gamma^N_u\overline{\varepsilon^N}(x)$.

Let us assume for now that the optimal control u^*, the virtual control v^*_N, and the approximate control u^*_N are equal. Figure 2 illustrates the error of approximation $\overline{\varepsilon^N}(x)$ for a deterministic process $\Gamma_u W(x) = \gamma W(y(x,u))$ with $y(x,u)$ being the successor state of state x, control u. The interpolation at $y(x,u)$ introduces a stochasticity in the approximate operator: $\Gamma^N_u W(x) = \gamma\sum_i p(x_i|x,u)W(x_i)$. Section 4.1 analyzes the interpolation error when using piecewise linear interpolation.

Thus, the quality of approximation at x depends on the interpolation error only at the areas of the state space that are reachable by the Markov process following the approximate optimal control.

Now, if u^*_N is not equal to v^*_N everywhere (see figure 3), the error of approximation still satisfies (5). However, the interesting parts of the state space are no

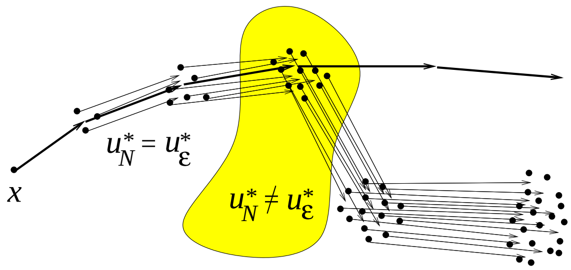

Figure 3. Same deterministic process. A trajectory following the approximate optimal control u_N^* is shown in bold. In the gray area, the control u_ε^* (that maximizes $\Gamma_u^N \overline{\varepsilon^N}$) differs from u_N^*, thus the error of approximation at x is the expected interpolation errors at the set of dots, and not around the trajectory.

longer the areas reachable by the Markov process following the approximate optimal control u_N^*, but the areas reachable by the Markov process following the control u_ε^* (that maximizes $\Gamma_u^N \overline{\varepsilon^N}$).

In both cases, the Markov representation (5) implies that we only need to consider the areas reachable by the Markov process $x(k)$ that follows the control u_ε^*.

Thus, instead of having a global rate of convergence that depends on the worst interpolation error over the whole state space, we obtain a much tighter rate that only considers the parts of the state space that have some *influence* on the area of interest. We define more precisely this notion of influence in the next section.

3. Error on the Value Function

3.1 Global Error

Here we want to analyze the quality of approximation of the value function for a given variable resolution grid. Depending on the optimization problem considered, we can define a criterion $E_\Omega(X_N)$ that gives a global estimation of how well a variable resolution grid X_N performs (Ω is a specific area of interest). For example, if we want to approximation the value function on a subset Ω (possibly the whole state space), the **global error** $E_\Omega(X_N)$ (to be minimized) would be the average ($\int_{x\in\Omega} \varepsilon^N(x)dx$) or the maximal ($\sup_{x\in\Omega} \varepsilon^N(x)$) error of approximation of V on Ω.

In Section 3.3, we describe how to estimate this error for any specific grid X_N. But more importantly, we also provide a method for computing the effect that an increase of the local interpolation error has on the global error, thus allowing us to design adaptive resolution refinement methods. This effect is expressed as the partial derivative of the global error $E_\Omega(X_N)$ with respect to a local decrease of the interpolation error e^N. In Section 4.1, we give an example of grid implementation and an estimation of the impact that locally adding new point to the grid has on the decrease in the interpolation error. By combining these results, we are able (Section 4.2) to predict the change in $E_\Omega(X_N)$ when we increase locally the resolution of the grid X_N.

But first, we describe a useful tool to measure non-local dependencies of a function satisfying a Bellman equation: the notion of *influence of a Markov chain.*

3.2 Influence of a Markov Chain

In (Munos & Moore, 1999a) we introduced the notion of influence of a Markov Chain as a way to measure non-local correlations between states. Briefly stated, if we have a set X_N of N values $v(i)$ satisfying some Bellman equation:

$$v(i) = \gamma \sum_j p(j|i)v(j) + r(i)$$

then, the influence $I_v(i|j)$ of a state i on another state j is the partial derivative of $v(j)$ with respect to $r(i)$:

$$I_v(i|j) = \frac{\partial v(j)}{\partial r(i)}$$

The influence measures the extent to which the state i "contributes" to the value of state j. We define the influence of a state i on a subset Ω as $I_v(i|\Omega) = \sum_{j\in\Omega} I_v(i|j)$.

In (Munos & Moore, 1999a), we showed that the influence satisfies the following property:

$$I_v(i|j) = \gamma \sum_k p(i|k).I_v(k|j) + \left\{ \begin{array}{lll} 1 & if & i=j \\ 0 & if & i \neq j \end{array} \right. \quad (6)$$

which is not a Bellman equation since the sum of the probabilities $\sum_k p(i|k)$ may be greater than 1. However, this is a fixed-point equation whose operator is contraction in 1−norm, thus has a unique solution −the influence− that can be computed by successive iterations.

For example, the influence $I_v(i|\Omega)$ on a subset Ω is obtained by taking the limit of the iterated:

$$I_v(i|\Omega) \leftarrow \gamma \sum_k p(i|k).I_v(k|\Omega) + \left\{ \begin{array}{lll} 1 & if & i\in\Omega \\ 0 & if & i \notin \Omega \end{array} \right.$$

Therefore, the computation of the influence $I_v(i|\Omega)$ is cheap and is equivalent to solving a Markov chain.

3.3 Approximation of the Value Function

Let us consider the problem of minimizing the global error $E_\Omega(X_N) = \int_{x\in\Omega} \varepsilon^N(x)$.

For a given grid X_N, the approximate value function V^N is computed by solving $V^N = T^N V^N$. Then the bound on the error of approximation $\overline{\varepsilon^N}$ is computed by solving (2) and using the procedure described at the end of section 2.3.1. Thus, we deduce that $\overline{E}_\Omega = \int_{x\in\Omega} \overline{\varepsilon^N}(x)$ is an upper bound for $E_\Omega(X_N)$ and $\overline{E}_\Omega^N = \sum_{x_i \in X_N \cap \Omega} \overline{\varepsilon^N}(x_i)$ approximates a proportion (depending on the number of grid point in Ω) of $\overline{E}_\Omega$.

Furthermore, now we estimate the partial contribution of each local interpolation error to the global error $E_\Omega(X_N)$, in the hope of deriving a adaptive refinement procedure that will increase locally the resolution of the grid at the most important areas of the state space for minimizing $E_\Omega(X_N)$.

First, if we are interested in decreasing the error at some state x_i, then we want to find for every other state x_j the extent to which a change in the local interpolation error at x_j will affect the error of approximation at state x_i. Thus, we want to compute the partial derivative of the bound $\overline{\varepsilon^N}(x_i)$ with respect to the local error $e^N(x_j)$. A convenient way to do this is to note from (2) that $\overline{\varepsilon^N}$ is the solution of a Markov chain whose Bellman equation is:

$$\overline{\varepsilon^N}(x_i) = \Gamma^N_{u^*_\varepsilon} \overline{\varepsilon^N}(x_i) + e^N_{u^*_e}(x_i) \tag{7}$$

with $u^*_\varepsilon(x_i)$ being the argument of $\max_{u\in U_N(x_i)} \Gamma^N_u \overline{\varepsilon^N}(x_i)$ and $u^*_e(x_i)$ the argument of $\max_{u \in U'_N(x_i)} e^N_u(x_i)$. From the previous section, the influence provides the desired solution:

$$I_{\overline{\varepsilon^N}}(x_j|x_i) = \frac{\partial \overline{\varepsilon^N}(x_i)}{\partial e^N(x_j)}$$

Now, if we want to minimize the bound $\overline{\varepsilon^N}$ over a subset Ω, we can derive the partial derivative of the bound $\overline{E}_\Omega^N$ with respect to $e^N(x_j)$ by computing the influence $I_{\overline{\varepsilon^N}}(x_j|\Omega)$:

$$\frac{\partial \overline{E}_\Omega^N}{\partial e^N(x_j)} = \frac{\partial \sum_{x_i\in\Omega} \overline{\varepsilon^N}(x_i)}{\partial e^N(x_j)} = I_{\overline{\varepsilon^N}}(x_j|\Omega) \tag{8}$$

Consequently, for a given grid x_N, the states x_j of highest influence $I_{\overline{\varepsilon^N}}(x_j|\Omega)$ show the areas in which decreasing the local interpolation error would decrease the most the global error $E_\Omega(X_N)$. Hence, a new grid can be built by locally increasing the resolution of the grid around those states x_j. A more precise formulation is given in section 4.2.

4. An Example of Grid Implementation

In section 4.1, we show a simple example of grid implementation for a deterministic problem and establish a first-order estimation of the decrease Δe^N of the local interpolation error when we add ΔN points to the grid. As a result, we can derive efficient VR refinement methods in section 4.2.

4.1 Piecewise Linear Interpolation

Let us consider the special case of a deterministic process. We denote $y = y(x,u)$ the successor state of (state x, control u). The operator is $T_u(x)W(x) = \gamma.W(y) + r(x,u)$. Let us use the approximate reward function $r_n = r$.

We consider a triangulation of the state space for which a function W is linearly interpolated inside each simplex: for $y \in Simplex\{x_0, ..., x_d\}$ (d is the dimension of the state space) we approximate $W(y)$ by $\sum_{i=0}^d \lambda_i(y) W(x_i)$ with the coefficients $\lambda_i(y)$ being the *x_i-barycentric coordinates* of y. These coefficients satisfy: $\lambda_i(y) \geq 0$, $\sum_{i=0}^d \lambda_i(y) = 1$, and $\sum_{i=0}^d \lambda_i(y).x_i = y$. For a given point y inside a simplex $\{x_0, ..., x_d\}$, the barycentric coordinates exist and are unique. Thus, the approximate operator is:

$$T_u^N W(x) = \gamma.\sum_{i=0}^d \lambda_i(y) W(x_i) + r(x,u)$$

Define $\delta(x,u)$ to mean the local discretization step around y, which means that the grid-points $\{x_i\}$ used for the interpolation at y are at a distance proportional to $\delta(x,u)$ from y. From Taylor's theorem, we have $W(x_i) = W(y) + (x_i - y)DW(y) + \frac{1}{2}(x_i - y)D^2W(y)(x_i - y)^T + o(\delta(x,u)^2)$ (where ^{T} means the transpose of the vector). If we assume that V is locally $\mathcal{C}^2$, we can deduce the interpolation error:

$$e^N(x) = \frac{1}{2}\gamma \sum_{i=0}^d \lambda_i(y)(x_i - y) D^2 V(y)(x_i - y)^T + o(\delta(x)^2)$$

with $y = y(x, u^*_e)$ being here the successor of state x when choosing the control u^*_e, and $\delta(x)$ being $\delta(x, u^*_e)$. Thus:

$$e^N(x) = \frac{1}{2}\gamma D^2 V(y).\delta(x)^2 + o(\delta(x)^2)$$

Now, if we increase the resolution of the grid by adding ΔN new points around y, the discretization step $\delta(x)$ will decrease as $-\Delta\delta(x) \sim \frac{\delta(x)}{d}.\frac{\Delta N}{N}$, because the number of points $N \sim \int_X \frac{ds}{\delta(s)^d}$. Thus, the local interpolation error will decrease as:

$$-\Delta e^N(x) \sim \gamma D^2 V(y) \frac{\delta^2(x)}{d} . \frac{\Delta N}{N} \quad (9)$$

Remark. In general, the local interpolation error $e^N(x)$ (and thus $\Delta e^N(x)$) depends on both *the local resolution of the grid* used for the back-up interpolation and *some measure of the curvature* of the value function V. In the previous implementation it depends on the second order differential of V. In random or low-discrepancy grids (Niederreiter, 1992; Rust, 1996) it depends on the variation of V. Since V is unknown, we need to estimate this curvature by using the approximate value function V^N at neighboring points. This important point will be developed in future work.

4.2 Algorithms for Variable Resolution

By combining the results of Sections 3.3 and 4.1 we are able to predict the effect that locally refining the grid has on the global error we are minimizing. Indeed, in the previous implementation, adding ΔN new points around y would decrease the local interpolation error of $-\Delta e^N(x)$ (according to (9)), which from Section 3.3, will increase the global error by $\Delta \overline{E}_\Omega^N = -\frac{\partial \overline{E}_\Omega^N}{\partial e^N(x_j)} . \Delta e^N(x)$ (according to (8)). An efficient variable resolution refinement procedure would build a new grid by adding new points around the areas of highest $\frac{\Delta \overline{E}_\Omega^N}{\Delta N}$ thus minimizing the bound on the expected global error.

Also we may consider removing points from the grid around the areas of lowest influence on the global error. Doing so can save us some computational resources, which can be used to increase the resolution of the grid somewhere else.

5. Conclusion

This paper has introduced new ideas for bounding the error of value function approximations. Equation (2) provides bounds on the approximation error in terms of the interpolation error. This latter mainly results from the specifics of the grid interpolation and the curvature of the value function. We also showed how to use a prior error bound to safely eliminate provably sub-optimal controls, thus tightening even more the bound on the approximation error. For a given VR grid, the Markov representation (5) expressed this bound as the expected sum of the discounted interpolation errors obtained in some specific area of influence, which can be easily computed. Thus, tight rates of convergence that only need to consider a restricted area of the state space can be derived.

Finally, the paper discussed the main consequence of obtaining these bounds: it showed us where to increase the resources of the approximator for the purpose of decreasing the error in the value function approximation.

In future work, we will extend this analysis for providing bounds on the loss to occur (based on the bounds in the value function) if we follow an approximate (sub-optimal) policy and deduce the parts of the state space where an increase of the resolution will reduce this loss (which are not necessarily the same parts as for reducing the value function error). Furthermore, we will investigate possible use in reinforcement learning, where the dynamics must be obtained from experience.

References

Bertsekas, D. P., & Tsitsiklis, J. (1996). *Neuro-dynamic programming.* Athena Scientific.

Chow, C., & Tsitsiklis, J. (1991). An optimal one-way multigrid algorithm for discrete-time stochastic control. *Proceedings of the IEEE Transactions on Automatic Control, 36-8*, 898–914.

Davies, S. (1996). Multidimensional triangulation and interpolation for reinforcement learning. *Advances in Neural Information Processing Systems, 8.*

Gordon, G. J. (1999). *Approximate solutions to markov decision processes.* Doctoral dissertation, CS department, Carnegie Mellon University, Pittsburgh, PA.

Munos, R., & Moore, A. (1998). Barycentric interpolators for continuous space and time reinforcement learning. *Advances in Neural Information Processing Systems.*

Munos, R., & Moore, A. (1999a). Influence and variance of a markov chain : Application to adaptive discretizations in optimal control. *Proceedings of the 38th IEEE Conference on Decision and Control.*

Munos, R., & Moore, A. (1999b). Variable resolution discretization for high-accuracy solutions of optimal control problems. *Proceedings of the International Joint Conference on Artificial Intelligence.*

Niederreiter, H. (1992). Random number generation and quasi-monte carlo methods. *SIAM CBMS-NSF Conference Series in Applied Mathematics, Philadelphia, 63.*

Puterman, M. L. (1994). *Markov decision processes, discrete stochastic dynamic programming.* A Wiley-Interscience Publication.

Rust, J. (1996). *Numerical dynamic programming in economics.* In Handbook of Computational Economics. Elsevier, North Holland.

Tsitsiklis, J., & Van Roy, B. (1996). An analysis of temporal difference learning with function approximation. *Technical report LIDS-P-2322, MIT.*