My introduction to dynamic programming

I first heard the term “dynamic programming” in 1965 while an undergraduate math major at Cornell. I was sitting outside the student union building one warm Spring day, having survived a brutal Ithaca winter, chatting with an industrial engineering student I knew about operations research and its subfields linear programming, dynamic programming and simulation. Without even knowing what it was, I found the expression *dynamic programming* curiously engaging. Little did I know at that time that it would be the focus of the next 42 (and still counting) years of my career.

Shift forward to 1968, when I was an Operations Research PhD student at Stanford and taking a course in dynamic programming from Don Iglehart. I was intrigued by the wide applicability and elegant theory of dynamic programming (a term which I view as synonymous with Markov decision processes). I especially was drawn to the rich theory of value iteration and contraction mappings (Shapley [1953], Denardo [1967]), policy iteration (Howard [1960]) and sensitive optimality (Blackwell [1962], Veinott, Jr. [1969]). While walking across campus following our class, fellow PhD student Robert W. (Bob) Rosenthal (Radner and Ray [2003]) raised the question “Why does policy iteration work so well?” This intriguing question proved to be the main focus of my research on MDP theory and algorithms throughout the 70’s.

While at Stanford I also took courses by Chernoff and Siegmund on sequential analysis, by Karlin and Chung on stochastic processes in addition to math courses in real analysis, functional analysis and partial differential equations. This rigorous foundation would prove invaluable to my subsequent research on diffusion process control and MDPs.

When it was time to choose a dissertation topic, my advisor Arthur F. (Pete) Veinott Jr., suggested I look at the recent book on diffusion processes (Mandl [1968]). In particular he wanted me to investigate whether the sensitive optimality theory for Markov Decision Processes (Veinott [1969]) could be applied to one-dimensional diffusion processes. I found Mandl’s book extremely challenging and had to do considerable background reading in Markov processes, differential equations and functional analysis to understand it. Among other results in my thesis, I developed a theory of sensitive optimality for one-dimensional diffusion processes (Puterman [1974]) and established the convergence of policy iteration for controlled multi-dimensional diffusions (Puterman [1977]). The latter research combined methods from stochastic processes, partial differential equations and dynamic programming and would provide a lead in to my fundamental work on policy iteration.

Research on Policy Iteration

I completed my dissertation in 1972 while an assistant professor at the University of Massachusetts. But the lure of the west coast was too strong, my wife, another Stanford alum, and I moved in 1974 to British Columbia where I became an Assistant Professor in the Faculty of Commerce at the University of British Columbia (UBC). During my first year at UBC, I presented my dissertation research in a study group seminar in optimal control theory with Ulrich Haussman, Ray Rishel and Armand Makowski (the latter two were visiting UBC from the University of Kentucky). One of them noted that my work appeared related to that on quasilinearization (Kalaba [1959]).
Investigating this line of research led me to recognize the link between policy iteration and Newton’s method. This was formalized in Puterman and Brumelle [1979] in which we established in considerable generality that policy iteration was equivalent to Newton’s method applied to the discounted dynamic programming optimality equation (OE) expressed in the form

\[ Bv(s) \equiv \max_{a \in A} \left\{ r(s, a) + \lambda \sum_{j \in S} p(j \mid s, a)v(j) - v(s) \right\} = 0 \]  

(1)

(the notation follows Puterman [1994]). Prior to our research it was customary to express the OE as the fixed point equation

\[ Tv(s) \equiv \max_{a \in A} \left\{ r(s, a) + \lambda \sum_{j \in S} p(j \mid s, a)v(j) \right\} = v(s) \]

so that it was natural to solve it with value iteration.

Newton’s method in its simplest form, solves an equation \( f(x) = 0 \) by iteratively computing \( f'(x_n) \) and then solving the linearized version of \( f(x) = 0 \) given by

\[ f(x_n) - f'(x_n)(x - x_n) = 0 \]  

(2)

To establish the relationship between policy iteration and Newton’s method, we found it convenient to re-express the OE in the form \( Bv = 0 \) so that solving the OE became equivalent to finding the zero (or root) of an operator in a linear space. We then showed that determining the maximizing policy in policy iteration corresponded to the differentiation step in Newton’s method and evaluating a policy corresponded to solving an equation of the form (2).

Next we addressed Bob Rosenthal’s question of why policy iteration converged in so few iterations. It was known in great generality that value iteration converged linearly at rate \( \lambda \). The numerical analysis literature showed that under certain smoothness properties on \( f(x) \), Newton’s method converged quadratically. Our work built on this theory to establish conditions under which policy iteration converged quadratically and to provide error bounds. The culmination of this work was an analytic theory for policy iteration which showed its equivalence to Newton’s method and established that policy iteration converged in significantly fewer iterations than value iteration. Hence we had an answer to the question Bob Rosenthal raised many years earlier as well as a formal theory to build on.

In this research we expressed the policy iteration algorithm in terms of the recursion

\[ v_{n+1} = v_n + (I - \lambda P_n)^{-1}Bv_n \]

where \( B \) is the operator defined in (7) and \( P_n \) is the transition matrix corresponding to the maximizing decision rule at iteration \( n \). Of course, when evaluating a policy in policy iteration, it was not necessary to compute the above inverse but instead solve a linear equation which required \( O(N^3) \) operations where \( N \) is the number of states.

We noted that for \( \lambda < 1 \)

\[ (I - \lambda P_n)^{-1} = \sum_{m=0}^{\infty} (\lambda P_n)^m \]
And therefore we could obtain a class of algorithms by truncating the above series at $M_n$. The consequence of this was that when $M_n = 0$ for all $n$, the new algorithm was equivalent to value iteration, when $M_n = \infty$ for all $n$ the algorithm was equivalent to policy iteration and in between we had a wide range of corresponding to different choices of the sequence $M_n$. We referred to this algorithm as modified policy iteration (MPI) and analyzed its theory and computational properties in Puterman and Shin [1978, 1982]. In my opinion, MPI remains the most efficient algorithm for solving moderate size MDPs.

The Book

At UBC I taught a PhD course on MDPs. I received encouragement from several colleagues, most notably Nico van Dijk of the University of Amsterdam, to formally write up my course notes. The final impetus came from Dan Heyman and Matt Sobel whom I met with while attending an Operations Research Society Meeting in Atlanta in 1985. They were putting together the Stochastic Models (Heyman and Sobel [1990]) volume of the Handbook of Operations Research and Management Science series and asked me to write a chapter on MDPs. This was all of the encouragement I needed and began this work during my sabbatical in 1987. Unfortunately, my article was far too long for a handbook chapter and had to be condensed significantly. But I had a lot of written material and a framework to build on for my book Markov Decision Processes (Puterman [1994]). Writing this book became my passion over the next six years.

I had two main objectives for the book; to create a state of the art monograph that brought all the beautiful and deep research on MDPs together in one place in a common notation, and to provide an introduction to MDPs that was accessible to graduate students and researchers. In particular, I wanted to provide a complete treatment of discounted models, average reward models, sensitive optimality and continuous time models. I had hoped to also include a chapter on partially observed MDPs but due to other pressing commitments and my wife threatening to leave me if I didn’t finish the book, I chose to omit this material.

I feel that my book was timely, useful and filled a significant gap; it has motivated students and researchers, it has been used as a text at many universities and it has been a source for many favorable comments and reviews. I am deeply touched when I visit other universities and find it on my colleagues desks or bookshelves.

The book was published at a time when there was a lull in MDP research; perhaps because its theory was mature or perhaps because the main challenges at that point of time were computational. Shortly thereafter, several different groups, most notably computer scientists and electrical engineers, became actively interested in solving very large MDPs. In collaboration with operations researchers, these communities created a broad range of techniques under the headings reinforcement learning, neuron-dynamic programming and approximate dynamic programming (ADP) that perhaps have built on the foundations laid out in my book.

Concluding Remarks

Since completing the book, I have continued my research on MDP theory and application. With my extensive background in statistical modelling and application, I focused on problems which explored the relationship between optimization and estimation. Ding, Puterman, and Bisi [2002] investigated the trade-off between parameter estimation and control in a dynamic newsvendor setting while Carvalho and Puterman [2005] studied this trade-off in a dynamic pricing environment. Recently I have begun investigating using MDP methods to address health care
management problems. In Patrick, Puterman and Queyranne [2007], we use ADP to determine an approximately optimal policy for dynamic stochastic multi-priority patient scheduling. My current research focuses on extending this work as well as using MDPs methods to enhance cancer care delivery.

I anticipate that the MDP field will continue to expand, and that MDP models will become more broadly applied. Applications in telecommunications, information search and health care management as well as new theoretical challenges in approximate dynamic programming will move the field forward and attract many new researchers. I feel that I entered the field when many basic research challenges presented themselves. Good advice from my Stanford advisor Pete Veinott and our mutual love for mathematical elegance and rigor inspired my research career. I am grateful for this foundation as well as the opportunity to have worked on MDP problems with several outstanding PhD students.

Martin L. Puterman
Vancouver, BC
December 2007

References


