Nonlinear Programming Method for Dynamic Programming

Yongyang Cai, Kenneth Judd, Valentina Michelangeli, and Che-Lin Su

March 24, 2011

1 Introduction

2 Parametric Dynamic Programming

An infinite horizon optimal decision-making problem has the following general form:

$$V(x_0) = \max_{a_t \in \mathcal{D}(x_t)} E\left\{ \sum_{t=0}^{\infty} \beta^t E\{u(x_t, a_t)\} \right\},$$

where x_t is the state process with initial state x_0, and $0 \leq \beta < 1$.

The dynamic programming (DP) model for the infinite horizon problems is

$$V(x) = \max_{a \in \mathcal{D}(x)} \quad u(x, a) + \beta E\{V(x^+) \mid x, a\},$$

where x is called the state variable (or vector), a is called the action variable (or vector), x^+ is the next-stage state conditional on the current-stage state x and the action a, and $V(x)$ is called the value function.

In DP problems, if state variables and control variables are continuous such that value functions are also continuous, then we have to use some approximation for the value functions, since computers cannot model the entire space of continuous functions. We focus on using a finitely parameterizable collection of functions to approximate value functions, $V(x) \approx \hat{V}(x; \mathbf{b})$, where $\mathbf{b}$ is a vector of parameters. The functional form $\hat{V}$ may be a linear combination of polynomials, or it may represent a rational function or neural network representation, or it may be some other parameterization specially designed for the problem. After the functional form is fixed, we focus on finding the vector of parameters, $\mathbf{b}$, such that $\hat{V}(x; \mathbf{b})$ approximately satisfies the Bellman equation.

3 Approximation

An approximation scheme consists of two parts: basis functions and approximation nodes. Approximation nodes can be chosen as uniformly spaced nodes, Chebyshev nodes, or some other specified nodes. From the viewpoint of basis functions, approximation methods can be classified as either spectral methods or finite element methods. A spectral method uses globally nonzero basis functions $\phi_j(x)$ such that $\hat{V}(x) = \sum_{j=0}^{n} c_j \phi_j(x)$ is the degree-n approximation. Here we present Chebyshev polynomial approximation as an example of spectral methods. In contrast, a finite element method uses locally basis functions $\phi_j(x)$ that are nonzero over sub-domains of the approximation domain. See Judd [2] and Cai [1] for detailed discussion of approximation methods.

3.1 Chebyshev Polynomial Approximation

Chebyshev polynomials on $[-1, 1]$ are defined as $T_j(x) = \cos(j \cos^{-1}(x))$, while general Chebyshev polynomials on $[a, b]$ are defined as $T_j((2x - a - b)/(b - a))$ for $j = 0, 1, 2, \ldots$. These polynomials are orthogonal under the weighted inner product: $\langle f, g\rangle = \int_a^b f(x)g(x)w(x)dx$ with the weighting function $w(x) = \left(1 - \left(\frac{2x-a-b}{b-a}\right)^2\right)^{-1/2}$. The polynomials $T_j(x)$ on $[-1, 1]$ can be recursively evaluated:

$$
\begin{aligned}
T_0(x) &= 1, \\
T_1(x) &= x, \\
T_{j+1}(x) &= 2xT_j(x) - T_{j-1}(x), \qquad j = 1, 2, \ldots.
\end{aligned}
$$

Using the above orthogonal polynomials, we have the least-squares polynomial approximation of V with respect to the weighting function

$$
w(x) = \left(1 - \left(\frac{2x - a - b}{b - a}\right)^2\right)^{-1/2},
$$

i.e., a degree-n polynomial $\hat{V}_n(x)$, such that $\hat{V}_n(x)$ solves

$$
\min_{\deg(\hat{V}) \le n} \int_a^b (V(x) - \hat{V}_n(x))^2 w(x)dx.
$$

Thus, we know that the least-squares degree-n polynomial approximation $\hat{V}_n(x)$ on $[-1, 1]$ has the form

$$
\hat{V}_n(x) = \frac{1}{2}c_0 + \sum_{j=1}^{n} c_j T_j(x),
$$

where

$$c_j = \frac{2}{\pi} \int_{-1}^{1} \frac{V(x)T_j(x)}{\sqrt{1-x^2}} dx, \qquad j = 0, 1, \ldots, n,$$

are the Chebyshev least-squares coefficients. It is difficult to compute the coefficients because the above integral generally does not have an analytic solutions, even if we know the explicit form of V.

4 Optimal Growth Problems

There are plenty of applications of dynamic programming (DP) method in economics. In this section we present the application to optimal growth models.

4.1 Deterministic Optimal Growth Problems

An infinite-horizon economic problem is the discrete-time optimal growth model with one good and one capital stock, which is a deterministic model. The aim is to find the optimal consumption function such that the total utility over the infinite-horizon time is maximal, i.e.,

$$\begin{aligned} V(k_0) = \max_{c} \quad & \sum_{t=0}^{\infty} \beta^t u(c_t) \\ \text{s.t.} \quad & k_{t+1} = F(k_t) - c_t, \quad t \geq 0, \end{aligned}$$

where k_t is the capital stock at time t with k_0 given, c_t is the consumption, β is the discount factor, $F(k) = k + f(k)$ with $f(k_t)$ the aggregate net production function, and $u(c_t)$ is the utility function. This objective function is time-separable.

If we add a new control variable l as the labor supply in the above model, it becomes

$$\begin{aligned} V(k_0) = \max_{c,l} \quad & \sum_{t=0}^{\infty} \beta^t u(c_t, l_t) \\ \text{s.t.} \quad & k_{t+1} = F(k_t, l_t) - c_t, \quad t \geq 0, \end{aligned}$$

where $F(k, l) = k + f(k, l)$ with $f(k_t, l_t)$ the aggregate net production function, and $u(c_t, l_t)$ is the utility function. This objective function is still time-separable.

4.2 Stochastic Optimal Growth Problems

We consider the stochastic optimal growth model now. Let θ denote the current productivity level and $f(k, l, \theta)$ denote net income. Define $F(k, l, \theta) =$

$k + f(k, l, \theta)$, and assume that θ follows $\theta_{t+1} = g(\theta_t, \varepsilon_t)$, where ε_t are i.i.d. disturbances. Then the infinite-horizon discrete-time optimization problem becomes

$$
\begin{aligned}
V(k_0, \theta_0) = \max_{k,c,l} \quad & E\left\{\sum_{t=0}^{\infty} \beta^t u(c_t, l_t)\right\} \\
\text{s.t.} \quad & k_{t+1} = F(k_t, l_t, \theta_t) - c_t, \\
& \theta_{t+1} = g(\theta_t, \varepsilon_t), \quad t \geq 0,
\end{aligned}
$$

where x_0 and θ_0 are given. The parameter θ has many economic interpretations. In the life-cycle interpretation, θ is a state variable that may affect either asset income, labor income, or both. In the monopolist interpretation, θ may reflect shocks to costs, demand, or both.

Its DP formulation is

$$
V(k, \theta) = \max_{c,l} \quad u(c, l) + \beta E\{V(F(k, l, \theta) - c, \theta^+) \mid \theta\},
$$

where θ^+ is next period's θ realization.

In the above model, k_{t+1} is a deterministic variable that is fully dependent on k_t, l_t, θ_t and c_t. But we can extend it to a stochastic capital stock case:

$$
\begin{aligned}
V(k_0, \theta_0) = \max_{k,c,l} \quad & E\left\{\sum_{t=0}^{\infty} \beta^t u(c_t, l_t)\right\} \\
\text{s.t.} \quad & k_{t+1} = F(k_t, l_t, \theta_t) - c_t + \epsilon_t, \\
& \theta_{t+1} = g(\theta_t, \varepsilon_t), \quad t \geq 0,
\end{aligned}
$$

where ϵ_t are i.i.d. disturbances, and independent of ε_t. Its DP formulation is

$$
\begin{aligned}
V(k, \theta) = \max_{c,l} \quad & u(c, l) + \beta E\{V(k^+, \theta^+) \mid \theta\} \\
\text{s.t.} \quad & k^+ = F(k, l, \theta) - c + \epsilon, \\
& \theta^+ = g(\theta, \varepsilon).
\end{aligned}
$$

5 Nonlinear Programming Method to Solve Bellman Equations

There are many ways to solve Bellman equations, such as value function iteration and policy iteration methods. Here we give one nonlinear programming method to solve Bellman equations for optimal growth problems. We call this method as DPNLP in the following. For a given set of capital points $\{k_i : \ i = 1, \dots, m\}$, and a given approximation form $\hat{V}(k; \mathbf{b})$, we come to solve the following model:

$$
\begin{aligned}
\max_{c,l,\mathbf{b}} \quad & \sum_{i=1}^{m} v_i, \\
\text{s.t.} \quad & v_i \leq u(c_i, l_i, k_i^+) + \beta v_i^+, \quad i = 1, \ldots, m, \\
& k_i^+ = F(k_i, l_i) - c_i, \quad i = 1, \ldots, m, \\
& v_i = \hat{V}(k_i; \mathbf{b}), \quad i = 1, \ldots, m, \\
& v_i^+ = \hat{V}(k_i^+; \mathbf{b}), \quad i = 1, \ldots, m, \\
& c_i > 0, \; l_i > 0, \; \underline{k} \leq k_i^+ \leq \bar{k}, \quad i = 1, \ldots, m,
\end{aligned}
$$

where $[\underline{k}, \bar{k}]$ is the range of capital for approximation.

We often need to add shape-preservation in the model, as the value function is increasing and concave:

$$
\begin{aligned}
\max_{c,l,\mathbf{b}} \quad & \sum_{i=1}^{m} v_i, \\
\text{s.t.} \quad & v_i \leq u(c_i, l_i, k_i^+) + \beta v_i^+, \quad i = 1, \ldots, m, \\
& k_i^+ = F(k_i, l_i) - c_i, \quad i = 1, \ldots, m, \\
& v_i = \hat{V}(k_i; \mathbf{b}), \quad i = 1, \ldots, m, \\
& v_i^+ = \hat{V}(k_i^+; \mathbf{b}), \quad i = 1, \ldots, m, \\
& \hat{V}'(k_i'; \mathbf{b}) \geq 0, \quad i = 1, \ldots, m', \\
& \hat{V}''(k_i'; \mathbf{b}) \leq 0, \quad i = 1, \ldots, m', \\
& c_i > 0, \; l_i > 0, \; \underline{k} \leq k_i^+ \leq \bar{k}, \quad i = 1, \ldots, m.
\end{aligned}
$$

where $\{k_i' : \; i = 1, \ldots, m'\}$ are the set of capital points for shape constraints in $[\underline{k}, \bar{k}]$. Usually the number of points for shape preservation is more than the number of points for approximation.

Since there are so many constraints, an optimization solver is often hard to find a feasible solution, so we want to have a higher degree approximation method while giving penalty in the extra degree of freedom. For example, if we use degree-n Chebyshev polynomial approximation, then

$$
\begin{aligned}
\max_{c,l,\mathbf{b}} \quad & \sum_{i=1}^{m} v_i - \sum_{j=0}^{n} \frac{1}{(j+1)^2} |b_j|, \\
\text{s.t.} \quad & v_i \leq u(c_i, l_i, k_i^+) + \beta v_i^+, \quad i = 1, \ldots, m, \\
& k_i^+ = F(k_i, l_i) - c_i, \quad i = 1, \ldots, m, \\
& v_i = \hat{V}(k_i; \mathbf{b}), \quad i = 1, \ldots, m, \\
& v_i^+ = \hat{V}(k_i^+; \mathbf{b}), \quad i = 1, \ldots, m, \\
& \hat{V}'(k_i'; \mathbf{b}) \geq 0, \quad i = 1, \ldots, m', \\
& \hat{V}''(k_i'; \mathbf{b}) \leq 0, \quad i = 1, \ldots, m', \\
& c_i > 0, \ l_i > 0, \ \underline{k} \leq k_i^+ \leq \bar{k}, \quad i = 1, \ldots, m.
\end{aligned}
$$

To cancel the absolute operator in the objective function, we let $b_j = b_j^+ - b_j^-$ with $b_j^+, b_j^- \geq 0$, so $|b_j| = b_j^+ + b_j^-$. That is, the model becomes

$$
\begin{aligned}
\max_{c,l,\mathbf{b}} \quad & \sum_{i=1}^{m} v_i - \sum_{j=0}^{n} \frac{1}{(j+1)^2} (b_j^+ + b_j^-), \\
\text{s.t.} \quad & v_i \leq u(c_i, l_i, k_i^+) + \beta v_i^+, \quad i = 1, \ldots, m, \\
& k_i^+ = F(k_i, l_i) - c_i, \quad i = 1, \ldots, m, \\
& v_i = \hat{V}(k_i; \mathbf{b}), \quad i = 1, \ldots, m, \\
& v_i^+ = \hat{V}(k_i^+; \mathbf{b}), \quad i = 1, \ldots, m, \\
& \hat{V}'(k_i'; \mathbf{b}) \geq 0, \quad i = 1, \ldots, m', \\
& \hat{V}''(k_i'; \mathbf{b}) \leq 0, \quad i = 1, \ldots, m', \\
& c_i > 0, \ l_i > 0, \ \underline{k} \leq k_i^+ \leq \bar{k}, \quad i = 1, \ldots, m, \\
& \mathbf{b} = \mathbf{b}^+ - \mathbf{b}^-.
\end{aligned}
$$

For the stochastic problems, the above model becomes

$$
\begin{aligned}
\max_{c,l,\mathbf{b}} \quad & \sum_{j=1}^{J}\left\{\sum_{i=1}^{m} v_{i,j} - \sum_{i=0}^{n} \frac{1}{(i+1)^2}(b_{i,j}^{+} + b_{i,j}^{-})\right\}, \\
\text{s.t.} \quad & v_{i,j} \leq u(c_{i,j}, l_{i,j}, k_{i,j}^{+}) + \beta \sum_{j'}^{J} P_{j,j'} v_{i,j,j'}^{+}, \quad i = 1, \ldots, m;\ j = 1, \ldots, J, \\
& k_{i,j}^{+} = F(k_i, l_{i,j}, \theta_j) - c_{i,j}, \quad i = 1, \ldots, m;\ j = 1, \ldots, J, \\
& v_{i,j} = \hat{V}(k_i, \theta_j; \mathbf{b}), \quad i = 1, \ldots, m;\ j = 1, \ldots, J, \\
& v_{i,j,j'}^{+} = \hat{V}(k_{i,j}^{+}, \theta_{j'}; \mathbf{b}), \quad i = 1, \ldots, m;\ j, j' = 1, \ldots, J, \\
& \hat{V}'(k_i', \theta_j; \mathbf{b}) \geq 0, \quad i = 1, \ldots, m';\ j = 1, \ldots, J, \\
& \hat{V}''(k_i', \theta_j; \mathbf{b}) \leq 0, \quad i = 1, \ldots, m';\ j = 1, \ldots, J, \\
& c_i > 0,\ l_i > 0,\ \underline{k} \leq k_i^{+} \leq \bar{k}, \quad i = 1, \ldots, m, \\
& \mathbf{b} = \mathbf{b}^{+} - \mathbf{b}^{-}.
\end{aligned}
$$

where $P_{j,j'}$ is the conditional probability of $\theta^{+} = \theta_{j'}$ given $\theta = \theta_j$, for any $j, j' = 1, \ldots, J$.

When the utility magnitude is large, the magnitude of value function will be large, so that it may be a challenge for optimization solvers. In this case, it will be important to scale the utility function appropriately.

6 Examples

At first, we try to solve deterministic optimal growth problems, in which we choose

$$F(k, l) = k + Ak^{\alpha} l^{1-\alpha}$$

with $\alpha = 0.25$ and $A = (1 - \beta)/(\alpha\beta)$. The utility function is set as

$$u(c, l, k^{+}) = \frac{c^{1-\gamma}}{1-\gamma} - B\frac{l^{1+\eta}}{1+\eta}$$

with $B = (1 - \alpha)A^{1-\gamma}$. Thus, we know that the steady state is $k_{ss} = 1$ with optimal labor supply $l_{ss} = 1$. When γ is large and c is small, the utility has a large magnitude, so we should scale it appropriately. One good choice is to divide it by $A^{1-\gamma}$ and then subtract it by $1/(1 - \gamma) - (1 - \alpha)/(1 + \eta)$. That is, the utility function is changed as

$$u(c, l, k^{+}) = \frac{(c/A)^{1-\gamma} - 1}{1-\gamma} - (1 - \alpha)\frac{l^{1+\eta} - 1}{1+\eta}$$

This constant scaling will keep the same optimal controls.

In the DPNLP model, we choose $m = 15$ Chebyshev nodes in $[0.2, 1.8]$, and the approximation method is Chebyshev polynomial with degree $n = 21$. The number of nodes for shape constraints is chosen as 60. The optimization solver is CONOPT. We tried $\beta = 0.8$, 0.9, 0.95 0.99, 0.995, 0.999, $\gamma = 0.5, 2, 8$, and $\eta = 0.1, 1$, all these examples give us good solutions.

We also tried to use SNOPT to solve the above optimal growth problems, and got almost the same results with CONOPT.

Now we come to solve stochastic optimal growth problems. We will keep the same utility function as the deterministic one, but the production function is changed as

$$F(k, l, \theta) = k + \theta A k^{\alpha} l^{1-\alpha}$$

where θ is the stochastic state. In the examples, the possible values of θ and θ^+ are

$$\theta_1 = 0.95, \ \theta_2 = 1.0, \ \theta_3 = 1.05,$$

and the probability transition matrix from θ to θ^+ is

$$P = \begin{bmatrix} 0.75 & 0.25 & 0 \\ 0.25 & 0.5 & 0.25 \\ 0 & 0.25 & 0.75 \end{bmatrix}.$$

We keep all parameters the same with the deterministic examples, and our results show that all the examples with stochastic states have good solutions.

References

[1] Yongyang Cai. *Dynamic Programming and Its Application in Economics and Finance.* PhD thesis, Stanford University, 2009.

[2] Kenneth Judd. *Numerical Methods in Economics.* The MIT Press, 1998.