

Smooth Approximation Methods

- Smooth Approximation methods differ from discrete approximation methods by approximating the value function $V = (V_0, \dots, V_T)$ or decision rule $\alpha = (\alpha_0, \dots, \alpha_T)$ by smooth functions of the state variable s rather than by a finite grid of points.
- For example many methods parameterize the value function or policy function in terms of a vector $\theta \in R^k$ of unknown parameters, choosing $\hat{\theta}$ so that the resulting estimates of V_θ or α_θ are as close as possible to the true solutions of V and α .
- Smooth approximation methods can be viewed from an abstract perspective as selecting estimates of \hat{V} , and $\hat{\alpha}$ from a finite-dimensional submanifold of the infinite dimensional space of value or policy functions.
- Differences in various approximation methods are a result of different ways of parameterizing these submanifolds and of different metrics for locating particular elements \hat{V} and $\hat{\alpha}$ that are “closest” to the true solution V and α .
- Think of our basic recursion relation that determines the value function V_t at time t . Assume that we have already computed a smoothed version of the time $t + 1$ value function so that we can quickly evaluate $\hat{V}_{t+1}(s)$ at any point $s \in S$.
- Using \hat{V}_{t+1} we can compute estimates of the time t value function \hat{V}_t at any finite collection of points (s_1, \dots, s_N) using the standard backward induction formula for dynamic programming

$$\hat{V}_t(s_i) = \hat{\Gamma}(\hat{V}_{t+1})(s_i) = \max_{a \in A(s_i)} \left[u(s_i, a) + \beta \int \hat{V}_{t+1}(s') p(ds' | s_i, a) \right] \quad (1)$$

for $i = 1, \dots, N$. We use the notation $\hat{\Gamma}(\hat{V}_{t+1})(s_i)$ to denote an approximation to the value of the true Bellman operator $\Gamma(\hat{V}_{t+1})(s_i)$ at s_i , because we will typically not be able to find analytic solutions to the integration and maximization problems in (1), so we must rely on approximate numerical solutions to these problems.

- Since it is costly to compute approximate solutions to the RHS of (1) at each of the N points (s_1, \dots, s_N) we would like to choose N as small as possible.

- However, in order to increase the accuracy of solutions to the numerical optimization and integration problems involved in calculating the time $t - 1$ value function $\hat{V}_{t-1} = \hat{\Gamma}(\hat{V}_t)$ we need to evaluate \hat{V}_t at a large number of points $s \in S$.
- Rather than doing so, we will consider approximation methods that only require us to evaluate \hat{V}_t at a relatively small number of points, $(\hat{V}_t(s_1), \dots, \hat{V}_t(s_N))$, and use these points as data in order to generate accurate “predicted” values of $\hat{V}_t(s)$ at other points $s \in S$ at much lower cost than evaluating (1) at each $s \in S$.
- A general approach is to approximate the value function by nonlinear least squares using a parametric family $V_\theta = (V_{0,\theta}, \dots, V_{T,\theta})$ that are smooth functions of $s \in S$ and a $(T + 1)k \times 1$ vector of unknown parameters $\theta = (\theta_0, \dots, \theta_T)$.
- Then our estimate of the time t value function is $\hat{V}_t = V_{t,\hat{\theta}}$ where $\hat{\theta}_t$ is defined recursively by

$$\hat{\theta}_t = \arg \min_{\theta \in R^k} \sigma_N(\theta) \sqrt{\sum_{i=1}^N |V_{\theta_t}(s_i) - \hat{\Gamma}(V_{\hat{\theta}_{t+1}})(s_i)|^2}, \quad (2)$$

where $V_{\hat{\theta}_{t+1}}$ is the least square estimate of the $t + 1$ value function V_{t+1} .

- The degree to which the smooth approximation approach succeeds in providing accurate approximations with a small number N depends on the degree of smoothness in the true solution. N will have to be quite large if we are not sure about the monotonicity and concavity properties of the value function.
- Notice that smooth approximation is similar to discrete approximation in that both methods require a specification of a grid (s_1, \dots, s_N) . The issue of optimal choice of grids will therefore appear also in this context.
- A number of other decisions have to be made to implement a smooth approximation method:
 1. Which object should we parameterize: V , α , or something else?
 2. How should this object be parameterized: via a linear in parameters specification such as a polynomial series approximation (Judd), or a non-linear-in-parameters approximation such a neural network, or by piecewise polynomial approximations such as spline or Hermite polynomials?

3. How should $\hat{\theta}$ be determined: via nonlinear least squares fit, or via a projection method such as Chebyshev interpolation or Galerkin's method, or some other possibly randomized procedure?
4. Which numerical optimization and integration methods should be used to compute the RHS of (1) at each of the grid points (s_1, \dots, s_N) ?

The answers to these questions lead to many different possible algorithms. We will mention some, and leave other for you to work on.