Dynamic Programming with Hermite Approximation[1]

Kenneth L. Judd, Hoover Institution

January 20, 2014

[1] Joint work with Yongyang Cai.

Envelope Theorem

- Envelope theorem: Let

$$\begin{aligned} V(x) &= \max_a \; f(x,a) \\ &\qquad \text{s.t. } g(x,a) = 0. \end{aligned}$$

Let $a^*(x)$ be the optimizer and $\lambda^*(x)$ be the shadow price.

$$\frac{\partial V}{\partial x} = \frac{\partial f}{\partial x}(x, a^*(x)) + \lambda^*(x)^\top \frac{\partial g}{\partial x}(x, a^*(x)).$$

Derivative of Value Functions in General Models

- For an optimization problem,

$$\begin{aligned} V(x) &= \max_a \; f(x,a) \\ &\quad \text{s.t. } g(x,a)=0, h(x,a)\geq 0, \end{aligned}$$

add a trivial control variable y and a trivial constraint $x-y=0$:

$$\begin{aligned} V(x) &= \max_{a,y} \; f(y,a) \\ &\quad \text{s.t. } g(y,a)=0, h(y,a)\geq 0, x-y=0. \end{aligned}$$

- Then by the envelope theorem, we get

$$V'(x)=\lambda,$$

where λ is the shadow price for the trivial constraint $x-y=0$.

Hermite Value Function Iteration (H-VFI)

Initialization. *Choose the approximation nodes,* $\mathbb{X}_t = \{x_t^i : 1 \le i \le N_t\} \subset \mathbb{R}^d$*, for every* $t < T$*, and choose a functional form for* $\hat{V}(x; \mathbf{b})$*. Let* $\hat{V}(x; \mathbf{b}^T) \equiv V_T(x)$*. Then for* $t = T-1, T-2, \ldots, 0$*, iterate through steps 1 and 2.*

Step 1. Maximization Step. For each $x^i \in \mathbb{X}_t$*,* $1 \le i \le N_t$*, compute*

$$
\begin{aligned}
v_i \quad = \quad & \max_{a \in \mathcal{D}(y,t), y} \quad u_t(y,a) + \beta \hat{V}(x^+; \mathbf{b}^{t+1}), \\
& \text{s.t.} \qquad x^+ = g_t(y,a), \\
& \qquad\qquad x_j^i - y_j = 0, , \quad j = 1, \ldots, d,
\end{aligned}
$$

and

$$
s_j^i = \tau_j^*(x^i),
$$

where $\tau_j^*(x^i)$ *is the shadow price of the constraint* $x_j^i - y_j = 0$*.*

Step 2. Hermite Fitting Step. Using an appropriate approximation method, compute the $\mathbf{b}^t$ *such that* $\hat{V}(x; \mathbf{b}^t)$ *approximates* (x^i, v_i, s^i) *data.*

Derivative of Value Functions in Optimal Growth Models

- For the optimal growth problem,

$$
\begin{aligned}
V_t(k) = \max_{k^+,c,l,y} \quad & u(c,l) + \beta V_{t+1}(k^+), \\
\text{s.t.} \quad & F(y,l) - c - k^+ = 0, \\
& k - y = 0,
\end{aligned}
$$

  with k^+, c and l as control variables, and y is the dummy variable.
- Formula for computing $V_t'(k)$:

$$V_t'(k) = \lambda,$$

  where λ is the shadow price for the dummy constraint $k - y = 0$, and given directly by optimization packages.

Chebyshev-Hermite Interpolation

- If we have Hermite data $\{(x_i, v_i, s_i) : i = 1, \ldots, m\}$ on $[x_{\min}, x_{\max}]$ with $x^i = (z_i + 1)(x_{\max} - x_{\min})/2 + x_{\min}$ (where $z_i = -\cos((2i-1)\pi/(2m)))$, then the following system of $2m$ linear equations produces coefficients for degree $2m-1$ Chebyshev polynomial interpolation on the Hermite data:

$$\begin{cases} \sum_{j=0}^{2m-1} b_j \mathcal{T}_j(z_i) = v_i, & i = 1, \ldots, m, \\ \frac{2}{x_{\max} - x_{\min}} \sum_{j=1}^{2m-1} b_j \mathcal{T}_j'(z_i) = s^i, & i = 1, \ldots, m, \end{cases} \tag{1}$$

where $T_j(z)$ are Chebyshev basis polynomials.

Multidimensional Hermite Approximation with Complete Chebyshev Polynomials

- Assume that we have Hermite data $\{(x^i, v_i, s^i) : i = 1, \ldots, N\}$ on $[-1, 1]^d$, where $x^i \in [-1, 1]^d$ are d-dimensional approximation nodes, $v_i = V(x^i)$, and $s^i = (s^i_1, \ldots, s^i_d)$ are the gradient of V at x^i, i.e., $s^i_j = \frac{\partial}{\partial x_j} V(x^i)$.
- Use least square method to compute coefficients for degree n complete Chebyshev polynomial approximation

$$\min_{\mathbf{b}} \left\{ \sum_{i=1}^{N} \left(v_i - \sum_{0 \le |\alpha| \le n} b_\alpha \mathcal{T}_\alpha \left(x^i\right) \right)^2 + \sum_{i=1}^{N} \sum_{j=1}^{d} \left(s^i_j - \sum_{0 \le |\alpha| \le n} b_\alpha \frac{\partial}{\partial x_j} \mathcal{T}_\alpha \left(x^i\right) \right)^2 \right\} \tag{2}$$

Derivative of Value Functions in Portfolio Optimization

- For the multi-stage portfolio optimization problem,

$$\begin{aligned} V_t(W) &= \max_{B,S} \; E\{V_{t+1}(R_f B + R^\top S)\}, \\ & \quad \text{s.t. } W - B - e^\top S = 0, \end{aligned}$$

  with the bond allocation B and the stock allocation S.
- Formula for computing $V_t'(W)$:

$$V_t'(W) = \lambda,$$

  where λ is the shadow price for the constraint $W - B - e^\top S = 0$.

Bounded Normal Random Variables

- Transform a standard normal random variable $\varsigma \sim \mathcal{N}(0,1)$ to a bounded random variable Ψ:

$$\Psi = \frac{1 - e^{-\kappa\varsigma}}{1 + e^{-\kappa\varsigma}} \Upsilon, \tag{3}$$

  where Υ and κ are two positive parameters. We see that Ψ has zero mean, and it is symmetric around the mean and bounded in $(-\Upsilon, +\Upsilon)$. Once we choose a number of Υ, we would like to choose a corresponding κ so that Ψ has a unit variance.

- Assume that the stocks have a bounded log-normal return $R = (R_1, \ldots, R_d)^\top$, i.e.,

$$\log(R_j) = \mu_i + \frac{1 - e^{-\kappa\varsigma_j}}{1 + e^{-\kappa\varsigma_j}} \Upsilon \sigma_j, \tag{4}$$

  where ς_j is a standard normal random variable for $j = 1, \ldots, d$, and the correlation matrix of $(\varsigma_1, \ldots, \varsigma_d)$ is Σ.

Nonlinear Change of Variable

- When the relative risk aversion coefficient γ in the power utility function is bigger than 1, the value function is steep and has a large magnitude at nearly 0 and also is very flat at a large wealth. So it will be hard to approximate the value function well on the state variable W if W has a small lower bound and a large upper bound.
- Use $w = \log(W)$ as our state variable.
- Approximate the value function with

$$\hat{V}(w; \mathbf{b}) = \sum_{j=0}^{n} b_j \mathcal{T}_j \left(\frac{2w - \overline{w} - \underline{w}}{\overline{w} - \underline{w}} \right),$$

  for any $w \in (\underline{w}, \overline{w})$, where $\underline{w} = \log(\underline{W})$ and $\overline{w} = \log(\overline{W})$,
- Approximation ranges are expanding just linearly. :

$$\begin{aligned} \underline{w}_{t+1} &= \min_{j=1,\ldots,d} \{\mu_j - \Upsilon \sigma_j\} + \underline{w}_t, \\ \overline{w}_{t+1} &= \max_{j=1,\ldots,d} \{\mu_j + \Upsilon \sigma_j\} + \overline{w}_t, \end{aligned}$$

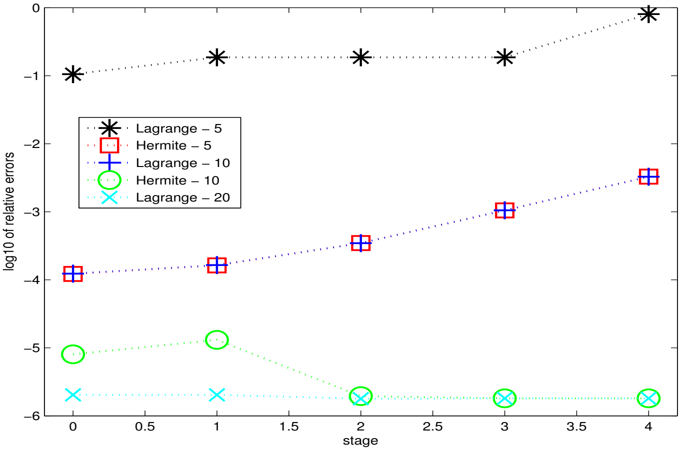
Example

- Portfolio with one bond and three stocks having the bounded and correlated log-normal returns
- Use product Gaussian-Hermite quadrature formula with 7 quadrature nodes in each dimension

Table: Relative Errors and Running Times of L-VFI or H-VFI for Dynamic Portfolio Optimization

m	L-VFI error	H-VFI error	L-VFI time	H-VFI time
5	0.8	0.00327	9 seconds	10 seconds
10	0.00328	1.3×10^{-5}	12 seconds	17 seconds
20	2.0×10^{-6}		33 seconds	

Relative Errors of H-VFI or L-VFI for Dynamic Portfolio Optimization



Note: The points labeled with "Lagrange - 5" are the errors of L-VFI with 5 approximation nodes, and "Hermite - 5" are for H-VFI with the same 5 approximation nodes.

Single-Country Stochastic Optimal Growth Problems

- Model:

$$V_0(k_0,\theta_0) = \max_{k_t,c_t,l_t} \mathbb{E}\left\{\sum_{t=0}^{T-1} \beta^t u(c_t,l_t) + \beta^T V_T(k_T,\theta_T)\right\}, \quad (5)$$
$$\text{s.t.} \quad k_{t+1} = F(k_t,l_t,\theta_t) - c_t, \quad 0 \le t < T,$$
$$\theta_{t+1} = h(\theta_t,\epsilon_t), \quad 0 \le t < T,$$

- DP Model for H-VFI:

$$\begin{aligned} V_t(k,\theta) \quad = \quad & \max_{k^+,c,l,y} \quad u(c,l) + \beta\mathbb{E}\left\{V_{t+1}(k^+,\theta^+)\right\}, \qquad (6)\\ & \text{s.t.} \quad k^+ = F(y,l,\theta) - c, \\ & \qquad \theta^+ = h(\theta,\epsilon), \\ & \qquad k - y = 0, \end{aligned}$$

Errors of optimal solutions of L-VFI or H-VFI for stochastic growth problems

			error of c_0^*		error of l_0^*	
γ	η	m	L-VFI	H-VFI	L-VFI	H-VFI
0.5	0.1	5	1.1(−1)	1.3(−2)	1.9(−1)	1.8(−2)
		10	5.4(−3)	2.7(−5)	7.8(−3)	3.7(−5)
		20	1.8(−5)	4.0(−6)	2.4(−5)	4.9(−6)
0.5	1	5	1.5(−1)	1.8(−2)	6.5(−2)	7.0(−3)
		10	7.2(−3)	3.4(−5)	2.9(−3)	1.5(−5)
		20	2.4(−5)	4.9(−6)	1.1(−5)	5.0(−6)
2	0.1	5	4.9(−2)	5.0(−3)	2.5(−1)	2.8(−2)
		10	2.5(−3)	1.6(−5)	1.5(−2)	8.0(−5)
		20	1.1(−5)	3.3(−6)	5.2(−5)	4.7(−6)
2	1	5	9.1(−2)	9.7(−3)	1.3(−1)	1.5(−2)
		10	4.2(−3)	2.7(−5)	6.7(−3)	4.7(−5)
		20	1.8(−5)	3.2(−6)	3.1(−5)	5.0(−6)
8	0.1	5	2.3(−2)	2.2(−3)	4.5(−1)	4.9(−2)
		10	9.5(−4)	1.2(−5)	2.2(−2)	2.6(−4)
		20	8.9(−6)	2.7(−6)	1.9(−4)	3.7(−6)
8	1	5	2.6(−1)	1.7(−2)	1.0(−0)	1.0(−1)
		10	8.4(−3)	3.8(−5)	5.2(−2)	2.4(−4)

Three-Country Optimal Growth Problems

- Model:

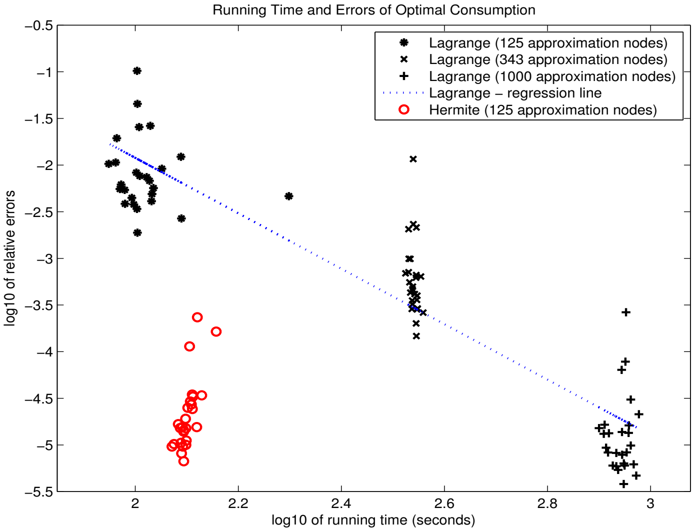
$$
\begin{aligned}
V_0(k_0) \quad = \quad & \max_{k_t, I_t, c_t, l_t} \quad \sum_{t=0}^{T-1} \beta^t u(c_t, l_t) + \beta^T V_T(k_T), && (7) \\
\text{s.t.} \quad & k_{t+1,j} = (1-\delta)k_{t,j} + I_{t,j}, \quad j = 1, \ldots, d, \\
& \Gamma_{t,j} = \frac{\zeta}{2} k_{t,j} \left(\frac{I_{t,j}}{k_{t,j}} - \delta \right)^2, \quad j = 1, \ldots, d, \\
& \sum_{j=1}^{d} (c_{t,j} + I_{t,j} - \delta k_{t,j}) = \sum_{j=1}^{d} \left(f(k_{t,j}, l_{t,j}) - \Gamma_{t,j} \right),
\end{aligned}
$$

Three-Country Optimal Growth Problems

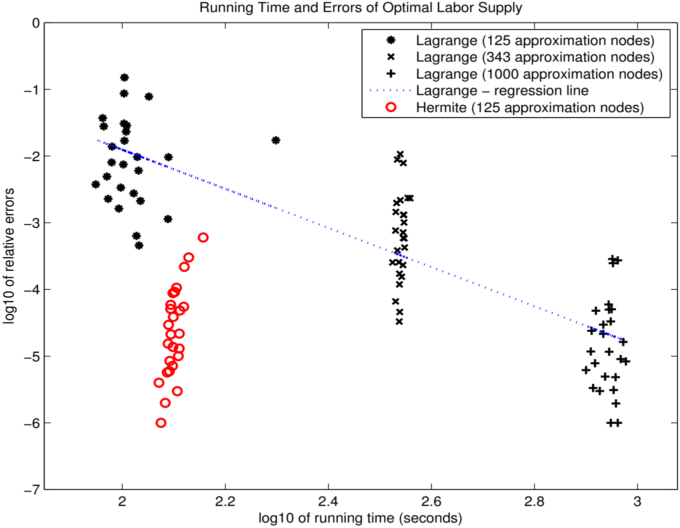
- DP Model for H-VFI:

$$
\begin{aligned}
V_t(k) \quad = \quad & \max_{k^+, I, c, l, y} \quad u(c,l) + \beta V_{t+1}(k^+), \qquad (8)\\
\text{s.t.} \quad & k_j^+ = (1-\delta) y_j + I_j, \quad j = 1, \ldots, d, \\
& \Gamma_j = \frac{\zeta}{2} y_j \left(\frac{I_j}{y_j} - \delta \right)^2, \quad j = 1, \ldots, d, \\
& \sum_{j=1}^{d} (c_j + I_j - \delta y_j) = \sum_{j=1}^{d} (f(y_j, l_j) - \Gamma_j), \\
& k_j - y_j = 0, \quad j = 1, \ldots, d,
\end{aligned}
$$

L-VFI vs H-VFI for Three-Country Optimal Growth Problems



L-VFI vs H-VFI for Three-Country Optimal Growth Problems



Six-Country Optimal Stochastic Growth Problems

- Model:

$$
\begin{aligned}
V_0(k_0,\theta_0) \quad = \quad & \max_{k_t, I_t, c_t, l_t} \quad \mathbb{E}\left\{\sum_{t=0}^{T-1}\beta^t u(c_t,l_t)+\beta^T V_T(k_T,\theta_T)\right\}, \qquad (9)\\
\text{s.t.} \quad & k_{t+1,j}=(1-\delta)k_{t,j}+I_{t,j}, \quad j=1,\ldots,d,\\
& \Gamma_{t,j}=\frac{\zeta}{2}k_{t,j}\left(\frac{I_{t,j}}{k_{t,j}}-\delta\right)^2, \quad j=1,\ldots,d,\\
& \sum_{j=1}^{d}\left(c_{t,j}+I_{t,j}-\delta k_{t,j}\right)=\sum_{j=1}^{d}\left(f(k_{t,j},l_{t,j},\theta_t)-\Gamma_{t,j}\right),\\
& \theta_{t+1}=g(\theta_t,\epsilon_t).
\end{aligned}
$$

H-VFI vs L-VFI for Six-Dimensional Stochastic Problems

	error of c_0^*		error of l_0^*		running times	
m	L-VFI	H-VFI	L-VFI	H-VFI	L-VFI	H-VFI
3	$3.8(-2)$	$3.6(-3)$	$5.4(-2)$	$5.2(-3)$	0.3	0.67
5	$5.5(-3)$		$8.2(-3)$		8.74	
6	$3.1(-3)$		$4.5(-3)$		36.6	

Note: $a(k)$ means $a \times 10^k$. Time unit is a hour.