Dynamic Programming

Part 1: Exercise 3, 4 & 5 (week 8)

Mikkel Reich

University of Copenhagen, Department of Economics

Plan for today

- Continuous states and choices
- 1D Interpolation
- Models with shocks
- Numerical integration
- Exercise 3, 4 & 5

Exercise 3: Cake-eating with continuous choice (+ states)

You can split a slice of cake! You can eat, say, 0.3 of a cake slice.

$V_t(W_t)$ is then defined for $W_t \geq 0$ and not "only" $W_t \in \mathbb{N}_0$.

$\rightarrow$ Implement using **grids**, e.g: $W_t \in \{0, 0.2, 0.4, \ldots, \} \; \forall t$

Similarly, we can do grid search on "share of total cake to eat", i.e.

when we reach cake size W_t, we search over:

$c_t \in \{0 * W_t, 0.01 * W_t, 0.02 * W_t, \ldots, 1 * W_t\}$

Small modification of Bellman equation:

$$V_t(W_t) = \max_{0 \leq c_t \leq W_t} \{\sqrt{c_t} + \beta V_{t+1}(W_{t+1})\} \quad \text{s.t. } W_{t+1} = W_t - c_t$$

(implement in infinite horizon in ex. 3 and finite horizon in ex. 4)

Exercise 3: Interpolation

$V_{t+1}(W_{t+1})$ is only known on the grid for W_{t+1} but $0 \leq c_t \leq W_t$

$\rightarrow$ Next period state $W_{t+1} = W_t - c_t$ can "fall" outside its grid.

$\rightarrow$ Use interpolation / function approximation for $V_{t+1}(W_{t+1})$

(Simple + useful) **linear interpolation algorithm** (`np.interp`):

1. Ensure that W_{t+1} is within two points in the grid (otherwise "extrapolation")
2. Find the two "neighbouring points" to W_{t+1} in the grid
3. Compute "weighted average" of $V_{t+1}(W_{t+1})$ in these two points (see [link])

Many alternatives: polynomial, neural nets, ... (next week)

Exercise 4: Cake-eating with uncertainty

What if your cake could (stochasticly) grow over-night with ϵ?

$\rightarrow$ Make state transition stochastic: $W_{t+1} = W_t - c_t + \epsilon$

$\rightarrow$ Form expectation over future value in Bellman equation:

$$V_t(W_t) = \max_{0 \leq c_t \leq W_t} \{\sqrt{c_t} + \beta \underbrace{\mathbb{E}[V_{t+1}(W_t - c_t + \epsilon)]}_{=\sum_{i=1}^{K} \pi_i V_{t+1}(W_t - c_t + \epsilon_i)}\}$$

, when ϵ is discretly uniform on $\{0, 1, 2, \ldots, K-1\}$ (ex. 4).

ϵ continuously distributed with PDF $p(\epsilon)$ $\rightarrow$ evaluate integral:

$$\mathbb{E}[V_{t+1}(W_t - c_t + \epsilon)] = \int_{\mathbb{R}} p(\epsilon) V_{t+1}(W_t - c_t + \epsilon) d\epsilon$$

Analytical expressions are hopeless $\rightarrow$ **Numerical integration**

Exercise 5: Numerical integration

Compute $\mathbb{E}[f(X)] = \int_{\mathbb{R}} p(x)f(x)dx$, where $p(x)$ is PDF of r.v. X

Monte Carlo: Draw N (large) number of X: $\{x_i\}_{i=1}^{N}$ and compute

$$\int_{\mathbb{R}} p(x)f(x)dx \approx \frac{1}{N}\sum_{i=1}^{n} f(x_i)$$

Works across distributions, no curse of dimensionality but slow

If $X \sim \mathcal{N}(\mu, \sigma^2)$, we can do much better!

Gauss-Hermite Quadrature: Deterministic method for approximating integrals like

$$\int_{\mathbb{R}} \exp(-x^2)f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i)$$

, where w_i is computed using Hermite polynomials and x_i are the n roots of the Hermite polynomial H_n.

Exercise 5: Intuition for Gauss-Hermite Quadrature

In practice: "import" raw pairs $\{(w_i, x_i)\}_{i=1}^{n}$ by using `gauss_hermite(n)`from `tools.py` (uses Golub-Welsch algorithm)

Exact integration if $deg(f) \leq 2n - 1$

$\rightarrow$ Polynomials are universal function approximators

$\rightarrow$ $f(x)$ as approx. to non-polynomial value/policy function

"Under the hood" **(Advanced)**:

1. Polynomial division of $f(x)$ with $H_n(x)$
2. Choose nodes $\{x_i\}_{i=1}^{n}$ as roots of $H_n(x)$
3. Use orthogonality of Hermite polynomials to reduce to "weighted integration" wrt. $r(x)$ with $deg(r) \leq n - 1$
4. Interpolate $r(x)$ exactly using n-degree Lagrange polynomial

Exercise 5: Gauss-Hermite for normally distributed r.v.

Consider $X \sim \mathcal{N}(\mu, \sigma^2)$, function $f(\cdot)$ and compute expectation:

$$\mathbb{E}[f(X)] = \int f(x) \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} \int f(x) \exp\left(-\left(\frac{x-\mu}{\sqrt{2\sigma^2}}\right)^2\right) dx$$

Change variables $z = \frac{x-\mu}{\sqrt{2\sigma^2}} \Rightarrow dz = \frac{1}{\sqrt{2\sigma^2}} dx \Leftrightarrow dx = \sqrt{2\sigma^2} dz$

$$\Rightarrow \mathbb{E}[f(x)] = \frac{1}{\sqrt{\pi}} \int f\left(\mu + \sqrt{2}\sigma z\right) \exp(-z^2) dz$$

$$\approx \frac{1}{\sqrt{\pi}} \sum_{i=1}^{n} w_i f\left(\mu + \sqrt{2}\sigma z_i\right)$$

$\rightarrow$ Can "integrate out" normal r.v. X by using Gauss-Hermite

$\rightarrow$ Choose weights as $\left\{\frac{w_i}{\sqrt{\pi}}\right\}_{i=1}^{n}$

$\rightarrow$ Choose nodes as $\left\{\mu + \sqrt{2}\sigma z_i\right\}_{i=1}^{n}$

Your time to shine!