Continuous-Time Methods in Macroeconomics

Jesús Fernández-Villaverde[1] and Galo Nuño[2]

October 15, 2021

[1]University of Pennsylvania

[2]Banco de España

Motivation

- Many interesting questions in macroeconomics require:

  1. Nonlinear techniques. Examples: How do financial crises arise? Why do countries or firms default? When do firms invest in large, lumpy projects? Why do individuals decide to migrate?

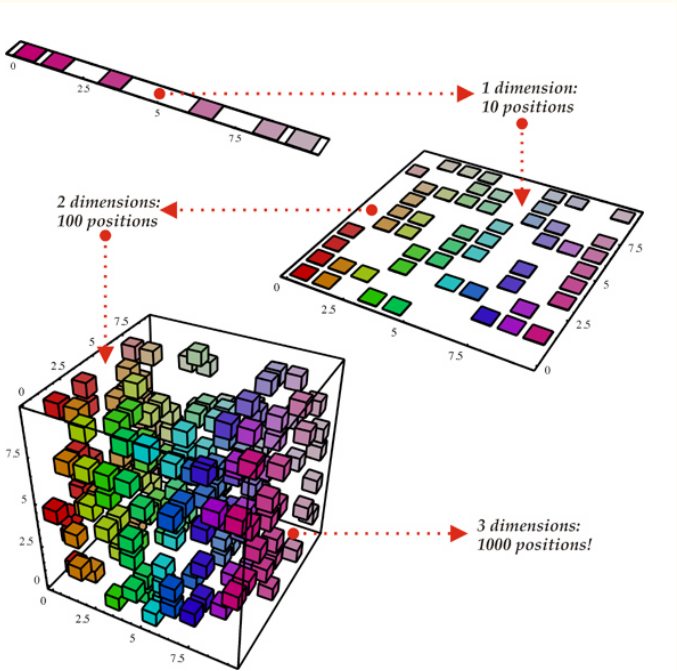
  2. Heterogeneous agents. Examples: What mechanisms account for changes in income and wealth inequality? Is there a trade-off between inequality and economic growth? How does inequality affect monetary and fiscal policy? What are the consequences of entry-exit in models of industry dynamics?

  3. Many state variables. Examples: Discrete node models, corporate finance models, rich life-cycle models, models where parameters are quasi-states.

- Often, all three elements come together. Example: heterogeneous agents models with nominal frictions and many assets.

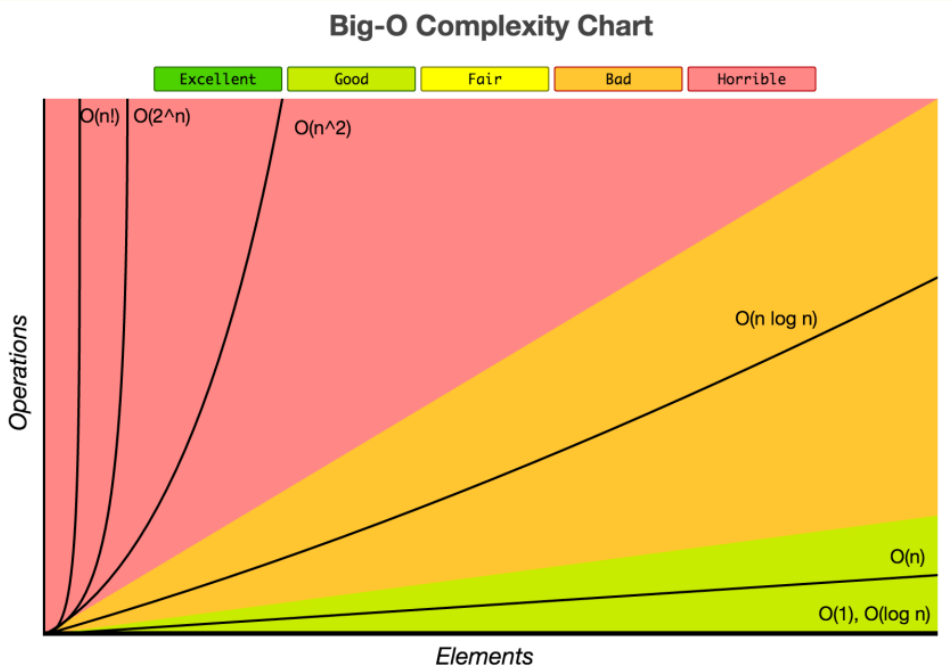
The challenge

- Modeling this class of problems rarely leads to analytic solutions. Thus, we must resort to numerical techniques.
- We want accurate and fast solution methods that can handle these models.
- Fast includes both coding and running time.
- While standard dynamic programming techniques can tackle, in theory, most environments, we would need to struggle with the "curse of dimensionality."
- Similar concern for projection methods.
- Challenges for perturbation approaches.


1 dimension:
10 positions
2 dimensions:
100 positions
3 dimensions:
1000 positions!

Our goal

- Move to the “feasible” region of the Big-O complexity chart.
- This is relevant both for time and memory complexity.
- In particular, we want to find ways to keep the “curse of dimensionality” under control.

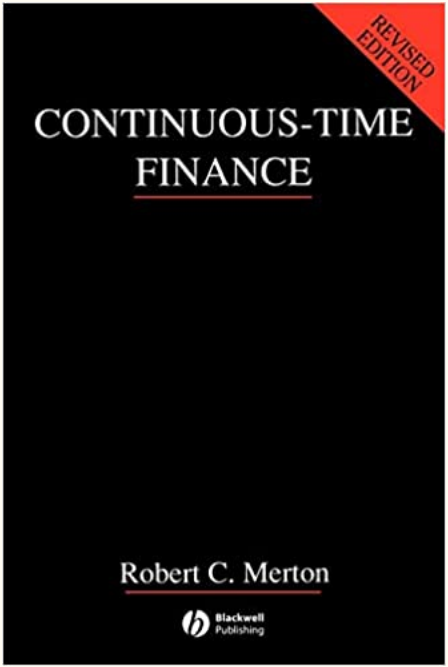

Big-O Complexity Chart
Excellent
Good
Fair
Bad
Horrible
O(n!)
O(2^n)
O(n^2)
O(n log n)
O(n)
O(1), O(log n)
Operations
Elements

Taming the "curse of dimensionality"

- Three strategies:
  1. Better numerical algorithms (i.e., continuous-time methods, deep learning).
  2. Better software implementations (i.e., robust OS, modern programming languages, functional programming, flexible data structures, advances in massive parallelization).
  3. Better hardware designs (i.e., GPUs, AI accelerators, FPGAs).
- Some of these techniques are relatively new in economics or, at least, less familiar to many researchers.
- A complete treatment of the material would require at least a whole semester.
- In this class, we will focus on better numerical algorithms: continuous-time methods and deep learning.

Why continuous time? I

- Long and illustrious tradition in finance: classical results by Merton and others.
- However, less used in macroeconomics (except in growth and neoclassical investment theories).
- Why?
  1. Economic data comes in discrete intervals: most time-series is in discrete time.
  2. Arrival of dynamic programming in the early 1970s.
  3. Stochastic calculus has some entry cost (notice: in growth theory, you can often skip stochastic calculus because you deal with deterministic models).
- Recent "boom" of continuous-time methods in business cycle research and related areas: Stokey (2009), Brunnermeier and Sannikov (2014), Ahn et al. (2017), ...


REVISED EDITION
CONTINUOUS-TIME FINANCE
Robert C. Merton
Blackwell Publishing

Why continuous time? II

- Itô's Lemma allow us to substitute the integrals of discrete time for derivatives in continuous time).

  Bellman equation:

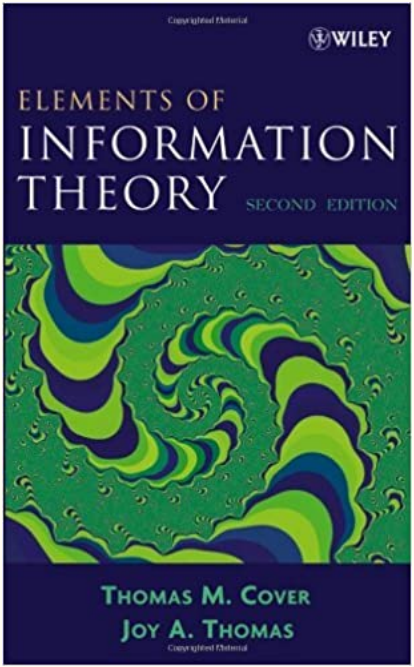
$$V(x) = \max_{\alpha} \left\{ u(\alpha, x) + \beta \int V(x') \, p(dx|\alpha, x) \right\}$$

  vs. Hamilton-Jacobi-Bellman equation:

$$\rho V_t(x) = \frac{\partial V}{\partial t} + \max_{\alpha} \left\{ u(\alpha, x) + \sum_{n=1}^{N} \mu_{nt}(x, \alpha) \frac{\partial V}{\partial x_n} + \frac{1}{2} \sum_{n_1, n_2 = 1}^{N} \left(\sigma_t^2(x, \alpha)\right)_{n_1, n_2} \frac{\partial^2 V}{\partial x_{n_1} \partial x_{n_2}} \right\}$$

- Why is this so important? Integrals depend on typical sets and typical sets are hard to characterize: the average member of a population with many dimensions (the "Asimov data set") is an outlier.

- Check: `https://mc-stan.org/users/documentation/case-studies/curse-dims.html`.


Copyrighted Material
WILEY
ELEMENTS OF
INFORMATION
THEORY
SECOND EDITION
THOMAS M. COVER
JOY A. THOMAS
Copyrighted Material

Why continuous time? III

- A few other mathematical advantages:
  1. Elegant and powerful math.
  2. Sparsity of transitions matrices.
  3. Easier to write complex FOCs, ...
- Related: much more work on PDEs than on stochastic difference equations.
- However: there are many occasions where discrete-time methods are still quite useful.

Why deep learning?

- Neural networks are compositional while traditional functional approximation methods are additive.

  Compare:

$$y = f(x) \cong g^{NN}(\mathbf{x};\theta) = \theta_0 + \sum_{m=1}^{M} \theta_m \phi \left(\theta_{0,m} + \sum_{n=1}^{N} \theta_{n,m} x_n \right)$$

  with a standard projection:

$$y = f(x) \cong g^{CP}(\mathbf{x};\theta) = \theta_0 + \sum_{m=1}^{M} \theta_m \phi_m(\mathbf{x})$$

  where ϕ_m is, for example, a Chebyshev polynomial.

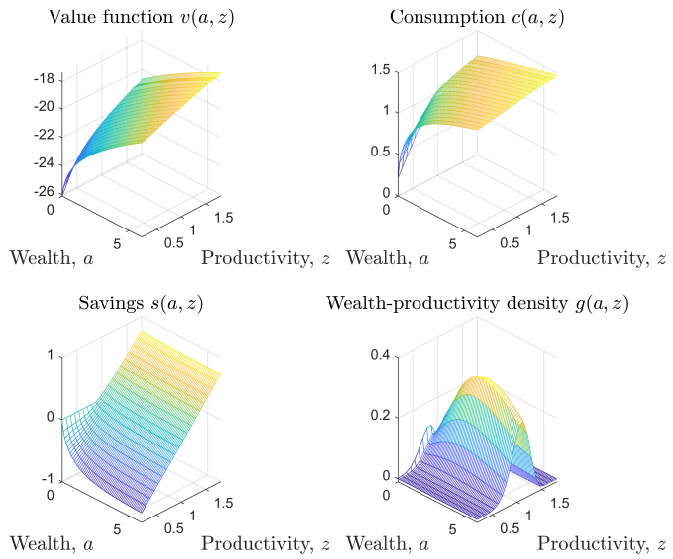
- This crucial difference allows neural networks to break the "curse of dimensionality."

- Furthermore, better hardware and software.

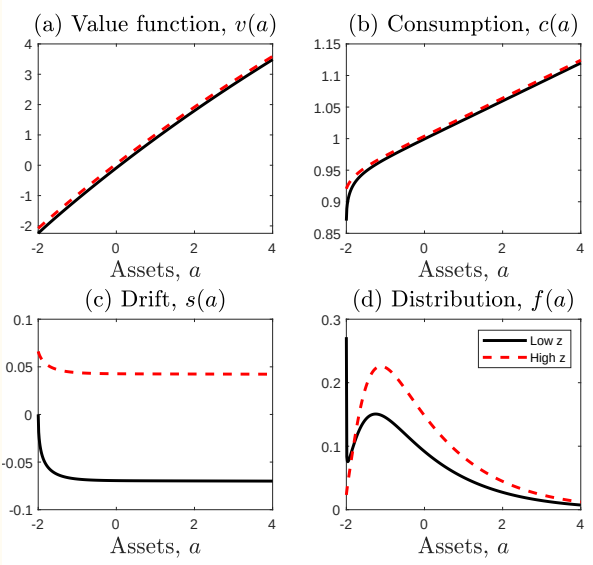
Course outline

1. Dynamic programming in continuous time.
2. Deep learning and reinforcement learning.
3. Heterogeneous agent models.
4. Optimal policy with heterogeneous agent models.

At the end of this course you will be able to do this...



... or this


(a) Value function, $v(a)$
4
3
2
1
0
-1
-2
-2
0
2
4
Assets, a
(b) Consumption, $c(a)$
1.15
1.1
1.05
1
0.95
0.9
0.85
-2
0
2
4
Assets, a
(c) Drift, $s(a)$
0.1
0.05
0
-0.05
-0.1
-2
0
2
4
Assets, a
(d) Distribution, $f(a)$
0.3
0.2
0.1
0
Low z
High z
-2
0
2
4
Assets, a