Lecture 13: Heterogeneous Agent Model

Yasuyuki Sawada, Yaolang Zhong

University of Tokyo

yaolang.zhong@e.u-tokyo.ac.jp

January 21, 2026

Recap: Motivation for Machine Learning

- Dynamic programming faces the curse of dimensionality
  - Parameterization problem
    - Classical functional forms (linear, polynomials, splines)
    - Number of basis terms grows rapidly with state dimension
    - Strong restrictions and poor capture of interactions
  - Grid / data problem
    - Grid-based DP requires N^d state points
    - Computation and memory grow exponentially
    - Sparse grids mitigate but do not eliminate the problem

How Machine Learning Addresses the Curse of Dimensionality

- Machine learning relaxes both constraints
  - Neural networks for parameterization
    - Flexible, high-dimensional function approximation
    - Captures nonlinearities and interactions
  - Stochastic simulation for data collection
    - Learn from simulated trajectories, not full grids
    - Scales to high-dimensional state spaces

RL for Economic Models: Data Generation

- One observation: (s, a, r, s')
  - s: taken as given
  - a: determined by policy $\sigma_\theta(s)$
  - r, s': determined by the environment's reward function $R(s, a)$ and transition function $P(s' \mid s, a)$
- Policy Evalution (Update Q_ϕ^σ) — requires σ_θ from Policy Improvement:
  - Fixed-point iteration:

$$Q_\phi^\sigma(s, a) \leftarrow R(s, a) \; + \; \beta \sum_{s'} P(s' \mid s, a) \, Q_\phi^\sigma(s', \sigma_\theta(s'))$$

  - Gradient-based update:

$$\phi \leftarrow \phi - \alpha \nabla_\phi \underbrace{\left(R(s, a) \; + \; \beta \sum_{s'} P(s' \mid s, a) \, Q_\phi^\sigma(s', \sigma_\theta(s')) - Q_\phi^\sigma(s, a) \right)^2}_{\text{Loss}}$$

RL for Economic Models: Q-Learning / Value Iteration

- One observation: (s, a, r, s')
- Q-Learning (Update Q_ϕ) — self-contained, only Q_ϕ needed:
  - Fixed-point iteration:

$$Q_\phi(s, a) \leftarrow R(s, a) \; + \; \beta \sum_{s'} P(s' \mid s, a) \max_{a'} Q_\phi(s', a')$$

  - Gradient-based update:

$$\phi \leftarrow \phi - \alpha \nabla_\phi \underbrace{\left(R(s, a) + \beta \sum_{s'} P(s' \mid s, a) \max_{a'} Q_{\bar{\phi}}(s', a') - Q_\phi(s, a) \right)^2}_{\text{Loss}}$$

RL for Economic Models: Policy Improvement

- One observation: (s, a, r, s')
- Policy Improvement (Update σ_θ) — requires Q_ϕ from Policy Evaluation:
  - Fixed-point iteration:

$$\sigma_\theta(s) \leftarrow \arg\max_a Q_\phi(s, a)$$

  - Gradient-based update:

$$\theta \leftarrow \theta + \alpha \nabla_\theta \underbrace{\log \sigma_\theta(a \mid s) \cdot A(s, a)}_{\text{Policy Gradient}}$$

where $A(s, a) = Q_\phi(s, a) - \mathbb{E}_{a \sim \sigma_\theta}[Q_\phi(s, a)]$ is the advantage

RL for Economic Models: Euler Equation Update

- One observation: (s, a, r, s') where $a = c$ (consumption), $s = (k, z)$
- Euler Equation Residual (Update c_θ) — self-contained, only c_θ needed:
  - Fixed-point iteration:

$$u'(c_\theta(s)) = \beta \sum_{s'} P(s' \mid s, a) \left[1 + r(s') - \delta\right] u'(c_\theta(s'))$$

  - Gradient-based update:

$$\theta \leftarrow \theta - \alpha \nabla_\theta \underbrace{\left(u'(c_\theta(s)) - \beta \sum_{s'} P(s' \mid s) \, R'(s') \, u'(c_\theta(s')) \right)^2}_{\text{Euler Residual Loss}}$$

where $R'(s') = 1 + r(s') - \delta$ is the gross return

State Sampling Methods

- Updates require samples (s, a, r, s'); given s, the tuple is determined by (σ_θ, R, P)
- Three approaches to obtain state s:
  1. Grid-based (non-stochastic): Deterministic enumeration over $\mathcal{S}$
  $$s \in \{s^{(1)}, s^{(2)}, \ldots, s^{(N)}\}$$
  2. Ergodic sampling: Draw from stationary distribution
  $$s \sim \mu^* \quad \text{(long-run distribution under } \sigma^*\text{)}$$
  3. Trajectory sampling: Sequential states from simulation
  $$(s_0, a_0, r_0, s_1, a_1, r_1, s_2, \ldots) \quad \text{where } s_{t+1} \sim P(\cdot \mid s_t, a_t)$$

State Sampling Methods: Comparison

	Grid-Based	Ergodic	Trajectory
Coverage	Full $\mathcal{S}$	Likely states	Path-dependent
Correlation	None	None (i.i.d.)	High (sequential)
Scalability	$O(N^d)$	Good	Good
Requirements	Grid construction	Estimate μ^*	Simulation

- Grid-based: Classical DP approach; curse of dimensionality
- Ergodic: Concentrates on economically relevant states; requires μ^* estimation
- Trajectory: Standard RL approach; requires decorrelation via replay buffer

Experience Replay Buffer

- Problem: data generated by sequential interaction is highly correlated
  - Violates i.i.d. assumptions underlying gradient-based optimization
  - Leads to unstable and inefficient learning
- Solution: store past transitions in a replay buffer $\mathcal{D} = \{(s, a, r, s')\}_{t=1}^{N}$
- Training loop:
  1. Interact with the environment using current policy π_θ (eval mode)
  2. Store observed transitions in $\mathcal{D}$
  3. Sample a random mini-batch from $\mathcal{D}$
  4. Update actor and critic using the sampled batch
- Key benefits:
  - Breaks temporal correlation and stabilizes training
  - Improves sample efficiency by reusing past experience

Online (On-Policy) vs. Offline (Off-Policy) Learning

- On-policy: Data generated by the current policy π_θ
  - Learning uses state distribution μ^{π_θ}; policy and data evolve jointly
  - Advantages: Stable learning, no distribution mismatch
  - Limitations: Data discarded after each update; low sample efficiency
- Off-policy: Data collected by a different or past policy
  - Learning target decoupled from data collection
  - Advantages: High sample efficiency via replay; flexible data reuse
  - Limitations: Potential instability due to distribution mismatch
- Implications for economic models:
  - Simulation is cheap; training neural networks is expensive
  - Off-policy learning preferred to reuse simulated data efficiently

Parallel Simulation for Data Collection

- Vectorized environments: Simulate M agents in parallel

$$\{s_t^{(1)}, s_t^{(2)}, \ldots, s_t^{(M)}\} \xrightarrow{\pi_\theta} \{a_t^{(1)}, a_t^{(2)}, \ldots, a_t^{(M)}\}$$

- Each step generates M transitions simultaneously
- Benefits:
  - GPU parallelism for policy evaluation
  - Faster data collection
  - Better gradient estimates (more samples per update)

Heterogeneous Agent Models

- Representative agent models assume identical agents
  - Cannot capture wealth/income inequality
  - Miss redistributive effects of policy
  - Aggregate consumption = individual consumption (scaled)
- Heterogeneous agent models allow:
  - Idiosyncratic risk: individual-specific shocks (unemployment, health)
  - Incomplete markets: cannot fully insure against idiosyncratic risk
  - Endogenous wealth distribution: emerges from optimization
- Krusell & Smith (1998): First model combining HA with aggregate uncertainty

Krusell-Smith (1998): Key Innovation

- Challenge: With aggregate shocks, agents need to forecast future prices
- Future prices depend on future aggregate capital K', which in terms future wealth distribution μ'
- Future wealth distribution μ' depends on the current wealth distribution μ and the policy function σ
- The wealth distribution μ is infinite-dimensional
- Krusell-Smith insight:
  - Agents use bounded rationality
  - Approximate μ with a few moments (e.g., mean capital $\bar{K}$)
  - Empirically: first moment captures 99.9% of price variation

Model Environment: Preferences

- Continuum of agents indexed by $i \in [0, 1]$
- Time is discrete, infinite horizon: $t = 0, 1, 2, \ldots$
- Preferences over consumption streams:

$$\mathbb{E}_0 \left[\sum_{t=0}^{\infty} \beta^t u(c_{it}) \right]$$

- Utility function (CRRA):

$$u(c) = \frac{c^{1-\gamma} - 1}{1 - \gamma}, \quad \gamma > 0$$

- $\beta \in (0, 1)$: discount factor
- γ: coefficient of relative risk aversion

Model Environment: Labor Income Process

- Aggregate state $z \in \{z_g, z_b\}$ (good/bad times)
  - Follows Markov chain with transition matrix Π^z
  - Affects aggregate productivity
- Idiosyncratic state $\epsilon \in \{0, 1\}$ (unemployed/employed)
  - Transition probabilities depend on aggregate state z
  - $\pi(\epsilon' | \epsilon, z, z')$: probability of ϵ' given current (ϵ, z) and next z'
- Labor income:

$$y_{it} = \begin{cases} w_t \bar{\ell} & \text{if } \epsilon_{it} = 1 \text{ (employed)} \\ \mu^u & \text{if } \epsilon_{it} = 0 \text{ (unemployed, UI)} \end{cases}$$

where w_t is the wage and $\bar{\ell}$ is labor supply

Model Environment: Budget Constraint

- Agents can save in a single asset: capital $a_{it} \geqslant 0$
- Borrowing constraint: $a_{it} \geqslant \underline{a}$ (often $\underline{a} = 0$)
- Budget constraint:

$$c_{it} + a_{it+1} = (1 + r_t - \delta)a_{it} + y_{it}$$

  where:

  - r_t: rental rate of capital
  - δ: depreciation rate
  - y_{it}: labor income
- Incomplete markets: No insurance against ϵ shocks
- Agents self-insure through precautionary savings

Production Technology

- Representative firm with Cobb-Douglas technology:

$$Y_t = z_t K_t^{\alpha} L_t^{1-\alpha}$$

- $K_t = \int a_{it}\, di$: aggregate capital
- $L_t = \int \epsilon_{it} \bar{\ell}\, di$: aggregate effective labor
- Competitive factor markets imply:

$$r_t = \alpha z_t \left(\frac{K_t}{L_t} \right)^{\alpha - 1}$$

$$w_t = (1-\alpha) z_t \left(\frac{K_t}{L_t} \right)^{\alpha}$$

State Variables

- Individual state: (a, ϵ)
  - a: individual asset holdings
  - ϵ: employment status
- Aggregate state: (z, μ)
  - z: aggregate productivity shock
  - μ: distribution of agents over (a, ϵ)
- Key challenge: μ is infinite-dimensional
- Prices (r, w) are functions of (z, μ)

Household's Recursive Problem

- Value function $V(a, \epsilon; z, \mu)$ solves:

$$V(a, \epsilon; z, \mu) = \max_{c, a'} \left\{ u(c) + \beta \mathbb{E} \left[V(a', \epsilon'; z', \mu') \mid \epsilon, z \right] \right\}$$

- Subject to:

$$\begin{aligned} c + a' &= (1 + r(z, \mu) - \delta) a + y(\epsilon, w(z, \mu)) \\ a' &\geqslant \underline{a} \\ \mu' &= \Gamma(z, \mu, z') \end{aligned}$$

- Γ: law of motion for the distribution (unknown!)
- Policy function: $a' = g(a, \epsilon; z, \mu)$

Law of Motion for Distribution

- Given policy function g and transition π:

$$\mu'(A, \epsilon') = \int_{a \in A} \int_{\epsilon} \mathbf{1}\{g(a, \epsilon; z, \mu) \in A\} \cdot \pi(\epsilon' | \epsilon, z, z')\, d\mu(a, \epsilon)$$

- This is a functional equation in μ
- Aggregate capital evolves:

$$K' = \int a'\, d\mu' = \int g(a, \epsilon; z, \mu)\, d\mu(a, \epsilon)$$

Recursive Competitive Equilibrium: Definition

A recursive competitive equilibrium consists of:

1. Value function $V(a, \epsilon; z, \mu)$
2. Policy function $g(a, \epsilon; z, \mu)$
3. Price functions $r(z, \mu)$, $w(z, \mu)$
4. Law of motion $\Gamma(z, \mu, z')$

Such that:

(i) Given prices and Γ, V and g solve household's problem

(ii) Prices satisfy firm's FOCs:

$$r = \alpha z \left(\frac{K}{L}\right)^{\alpha-1}, \quad w = (1-\alpha) z \left(\frac{K}{L}\right)^{\alpha}$$

(iii) Markets clear: $K = \int a \, d\mu$

(iv) Γ is consistent with g and transition probabilities

The Curse of Dimensionality

- Problem: Distribution μ is infinite-dimensional
- Cannot store μ on a computer directly
- Standard approaches fail:
  - Discretize a into N_a grid points
  - With $N_\epsilon = 2$ employment states
  - Need to track $N_a \times N_\epsilon$ probabilities
  - Value function becomes $V(a, \epsilon; z, \mu_1, \ldots, \mu_{N_a \times N_\epsilon})$
- Even with $N_a = 100$: state space is $\mathbb{R}^{200}$
- Infeasible to solve by standard dynamic programming

Krusell-Smith's Bounded Rationality Approach

- Key insight: Agents don't know the true law of motion Γ
- Instead, agents use a perceived law of motion (PLM)
- Approximate μ with moments: $\vec{m} = (m_1, m_2, \ldots, m_J)$
- Simplest case: $\vec{m} = \bar{K}$ (mean capital only)
- PLM for log aggregate capital:

$$\log K' = \begin{cases} a_0 + a_1 \log K & \text{if } z = z_g \\ b_0 + b_1 \log K & \text{if } z = z_b \end{cases}$$

- Coefficients (a_0, a_1, b_0, b_1) to be determined in equilibrium

Approximate Equilibrium

An approximate recursive competitive equilibrium consists of:

1. Value function $\tilde{V}(a, \epsilon; z, K)$
2. Policy function $\tilde{g}(a, \epsilon; z, K)$
3. Price functions $\tilde{r}(z, K)$, $\tilde{w}(z, K)$
4. PLM coefficients $\Psi = (a_0, a_1, b_0, b_1)$

Such that:

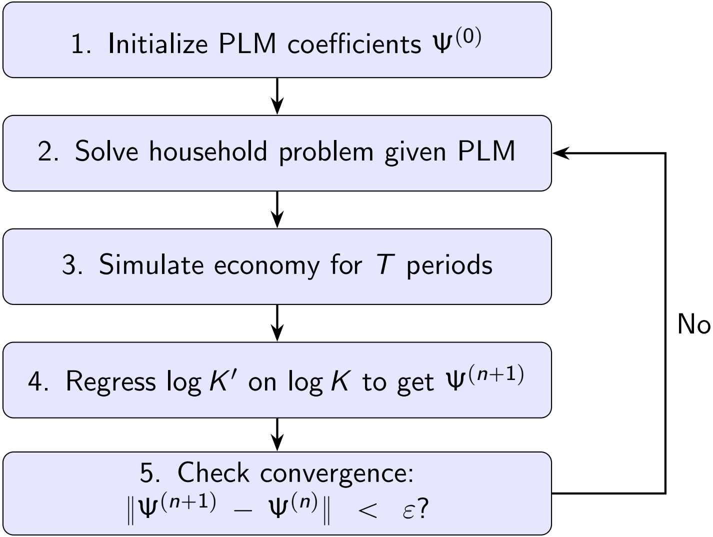
(i) Given prices and PLM, $\tilde{V}$ and $\tilde{g}$ solve household's problem

(ii) Prices from firm's FOCs with K as capital stock

(iii) PLM Ψ provides a good forecast of actual K'

Accuracy criterion: $R^2 > 0.9999$ for forecasting K'

Krusell-Smith Algorithm: Overview



Step 1: Initialize PLM

- Set initial guess for PLM coefficients:

$$\Psi^{(0)} = (a_0^{(0)}, a_1^{(0)}, b_0^{(0)}, b_1^{(0)})$$

- Common starting point:
  - $a_1^{(0)} = b_1^{(0)} \approx 0.99$ (high persistence)
  - $a_0^{(0)}, b_0^{(0)}$ chosen so steady-state K is consistent
- Or use steady-state Aiyagari model as initial guess

Step 2: Solve Household Problem

Given PLM Ψ, solve the Bellman equation:

$$\tilde{V}(a, \epsilon; z, K) = \max_{a' \geqslant \underline{a}} \left\{ u(c) + \beta \sum_{z', \epsilon'} \pi(z', \epsilon' | z, \epsilon) \tilde{V}(a', \epsilon'; z', K') \right\}$$

where:

$$\begin{aligned} c &= (1 + r(z, K) - \delta)a + y(\epsilon, w(z, K)) - a' \\ K' &= H(z, K; \Psi) \quad \text{(from PLM)} \end{aligned}$$

Solution methods:

- Value function iteration on grid
- Endogenous grid method (faster)
- Policy function iteration

Step 3: Simulate the Economy

- Track I agents for T periods (e.g., $I = 10,000$, $T = 11,000$)
- Initialize: draw (a_0^i, ϵ_0^i) from ergodic distribution
- For each period t:
  1. Compute aggregate capital: $K_t = \frac{1}{I}\sum_{i=1}^{I} a_t^i$
  2. Draw aggregate shock z_{t+1} from Markov chain
  3. For each agent i:
     - Draw ϵ_{t+1}^i from $\pi(\cdot|\epsilon_t^i, z_t, z_{t+1})$
     - Compute $a_{t+1}^i = \tilde{g}(a_t^i, \epsilon_t^i; z_t, K_t)$
- Discard first T_0 periods (burn-in, e.g., $T_0 = 1,000$)

Step 4: Update PLM via Regression

- From simulation, obtain time series $\{K_t, z_t\}_{t=T_0}^{T}$
- Run OLS regressions separately for each z:

$$\log K_{t+1} = a_0 + a_1 \log K_t + \varepsilon_t \quad \text{for } t : z_t = z_g$$

$$\log K_{t+1} = b_0 + b_1 \log K_t + \varepsilon_t \quad \text{for } t : z_t = z_b$$

- New PLM coefficients: $\Psi^{(n+1)} = (\hat{a}_0, \hat{a}_1, \hat{b}_0, \hat{b}_1)$
- Check R^2 for forecast quality (should be > 0.9999)

Step 5: Check Convergence

- Convergence criterion:
$$\|\Psi^{(n+1)} - \Psi^{(n)}\| < \varepsilon$$
- Typically $\varepsilon = 10^{-4}$ or smaller
- If not converged:
  - Update: $\Psi^{(n+1)} = \lambda\Psi^{(n+1)} + (1-\lambda)\Psi^{(n)}$
  - Dampening factor $\lambda \in (0.3, 0.5)$ for stability
  - Return to Step 2
- If converged:
  - Verify $R^2 > 0.9999$ for both regressions
  - This validates the approximate equilibrium

The Near-Aggregation Property

- Surprising result: First moment ($\bar{K}$) is nearly sufficient
- $R^2 > 0.99999$ in original KS paper
- Intuition:
  - Employed agents (high income) save similarly
  - Unemployed agents (low income) dissave similarly
  - Individual differences average out in aggregation
  - Distribution shape matters little for aggregate K'
- This is called approximate aggregation
- Does NOT always hold (e.g., with larger shocks or more nonlinearity)