Structural Reinforcement Learning for Heterogeneous Agent Macroeconomics

Yucheng Yang*
Zurich

Chiyuan Wang*
Peking University

Andreas Schaab
Berkeley

Benjamin Moll
LSE

*equal contribution

Zurich Workshop on the Frontier of Quantitative Macroeconomics

Heterogeneous agent models with aggregate risk

- Classic papers by Krusell-Smith and Den Haan from late 90s...
- ... huge literature since then
- Key problem: rational expectations + general equilibrium
  $\Rightarrow$ distribution = state variable in Bellman equation ("Master equation")
  - true even though households/firms only care about prices
  - intuition: equilibrium prices are not Markov, only the distribution is
    $\Rightarrow$ forecast prices by forecasting distributions
- Despite recent impressive advances, still no general, efficient global solution method for HA models with aggregate risk
- This still really holds back HA literature, e.g. ~~non-linearities, crises~~

Today: sidestep Master eqn using reinforcement learning

RL = learning value & policy functions in incompletely-known Markov decision processes from Monte Carlo simulation (a.k.a. "approximate DP")

Here: RL about equilibrium prices but not individual states $\Rightarrow$ "Structural RL"

- for clarity: RL by the computational economist, not the model agents

Outcome: efficient & flexible global solution method for HA models with agg risk

- solves problems traditional methods struggle with:

1. non-trivial market clearing (Huggett with agg. risk) $\approx$ 1 min on Google Colab
2. portfolio choice $\approx$ 1 min
3. HANK with forward-looking price/wage Phillips curve $\approx$ 4 min

How does it work?

- in contrast to dynamic programming, RL can handle non-Markov states
- replace distribution with low-dim. prices in state space, grid-based not DNNs
- efficient market clearing using policy functions (= demand curves)

Literature and contribution

Global solution of full RE equilibrium (Master equation)
Han-Yang-E, Schaab, Gu-Lauriere-Merkel-Payne, Gopalakrishna-Gu-Payne, Huang, ...

- sidestep Master equation rather than "taming curse of dimensionality"
- Han-Yang-E DeepHAM = also RL-inspired

Global solution with low-dimensional state space Krusell-Smith, Den Haan, ...

- difference: no perceived law of motion

Self-confirming equilibrium & **restricted perceptions equilibrium**

- agents form price exp. from data generated by economy in which they live
- but do so using restricted state space

"Sequence space" Auclert-Bardóczy-Rognlie-Straub, AzinovicYang-Žemlička

- global solution in sequence space (low-dim. prices) via Monte Carlo

Adaptive (least squares) learning Bray, Marcet-Sargent, Evans-Honkapohja, Jacobson, Giusto, ...

- similarity: stochastic approximation; difference: ours $\neq$ theory of learning

Companion paper and codes

Recurrent Structural Policy Gradient
for Partially Observable Mean Field Games

Clarisse Wibault [1 2] **Johannes Forkel** [1] **Sebastian Towers** [1] **Tiphaine Wibault** [3] **Juan Duque** [4] **George Whittle** [2]
Andreas Schaab [5] **Yucheng Yang** [6] **Chiyuan Wang** [7] **Michael Osborne** [2] **Benjamin Moll** [† 8] **Jakob Foerster** [† 1]

[†]Equal supervision [1]FLAIR, Foerster Lab for AI Research, University of Oxford [2]MLRG, Machine Learning Research Group, University of Oxford [3]iFo Insitute, Ludwig-Maximilians-Universität Munich [4]MILA, Québec AI Institute [5]UC Berkeley [6]Swiss Finance Institute, University of Zurich [7]Peking University [8]London School of Economics. Correspondence to: Clarisse Wibault <clarisse.wibault@magd.ox.ac.uk>.

https://arxiv.org/abs/2602.20141

⚡ **MFAX: Mean-Field Games in JAX** ⚡

https://clarisse-wibault.github.io/rspg/

Google Colab notebook for Krusell-Smith model

Plan

1. Textbook HA model with aggregate risk – why prices aren't Markov
2. Brief primer on reinforcement learning (RL)
3. Structural RL for HA macro: sidestepping the Master equation

Textbook HA model with aggregate risk

Textbook HA model: Huggett (1993) with aggregate risk

- Continuum of agents i, heterog. in $(b_{i,t}, y_{i,t})$, $y_{i,t}$ = idios. risk, agg. shock z_t
- State of the economy: distribution $G_t(b, y)$ and agg. shock z_t
- Households choose consumption $c_{i,t}$ to maximize

$$v_{i,0} = \max_{\{c_{i,t}\}} \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t u(c_{i,t}) \quad \text{subject to}$$

$$c_{i,t} + q_t b_{i,t+1} = b_{i,t} + y_{i,t} z_t, \quad y_{i,t+1} \sim \mathcal{T}_y(\cdot|y_{i,t}), \quad b_{i,t+1} \geq \underline{b}$$

- Market clearing: bond price q_t such that

$$\int b'_t(b, y, z_t)\, dG_t(b, y) = 0, \quad \text{all } t$$

Note: agent problem depends on G_t only through low-dimensional price q_t

More compact notation: individual states s, prices p

- Continuum of agents i, heterogeneous in $s = (b, y)$
- State of the economy: distribution $G_t(s)$ and agg. shock z_t
- Price vector p_t, here only one price $p_t = q_t$
- Households choose consumption $c_{i,t}$ to maximize

$$v_{i,0} = \max_{\{c_{i,t}\}} \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t u(c_{i,t}) \quad \text{subject to}$$

$$s_{i,t+1} \sim \mathcal{T}_s(\cdot | s_{i,t}, c_{i,t}, z_t, p_t) = \text{budget constraint + income process}$$

- Market clearing: price p_t (bond price) such that

$$\int b'_t(s, z_t)\, dG_t(s) = 0, \quad \text{all } t$$

Note: agent problem depends on G_t only through low-dimensional price p_t

Even more compact notation: equilibrium price functional

- Continuum of agents i, heterogeneous in $s = (b, y)$
- State of the economy: distribution $G_t(s)$ and agg. shock z_t
- Households choose consumption $c_{i,t}$ to maximize

$$v_{i,0} = \max_{\{c_{i,t}\}} \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t u(c_{i,t}) \quad \text{subject to}$$

$$s_{i,t+1} \sim \mathcal{T}_s(\cdot | s_{i,t}, c_{i,t}, z_t, p_t)$$

- Low-dimensional equilibrium price functional

$$p_t = P^*(G_t, z_t), \qquad z_{t+1} \sim \mathcal{T}_z(\cdot | z_t)$$

Note: agent problem depends on G_t only through low-dim. price functionals

Even more compact notation: equilibrium price functional

- Continuum of agents i, heterogeneous in $s = (b, y)$
- State of the economy: distribution $G_t(s)$ and agg. shock z_t
- Households choose consumption $c_{i,t}$ to maximize

$$v_{i,0} = \max_{\{c_{i,t}\}} \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t u(c_{i,t}) \quad \text{subject to}$$

$$s_{i,t+1} \sim \mathcal{T}_s(\cdot | s_{i,t}, c_{i,t}, z_t, p_t)$$

- Low-dimensional equilibrium price functional

$$p_t = P^*(G_t, z_t), \qquad z_{t+1} \sim \mathcal{T}_z(\cdot | z_t)$$

Generalizes to $s \in \mathbb{R}^{n_s}$, $z \in \mathbb{R}^{n_z}$, $p \in \mathbb{R}^{n_p}$, reward function $R(s, a, z, p)$, a = actions

Discretizing the individual state space (important slide)

- Discretize individual state $s \in \{s_1, ..., s_J\}$ with $J = J_1 \times ... \times J_n$
- Value function, distribution, etc are J-dimensional vectors

$$\mathbf{v}_t = \begin{bmatrix} v_t(s_1) \\ \vdots \\ v_t(s_J) \end{bmatrix}, \qquad \mathbf{g}_t = \begin{bmatrix} g_t(s_1) \\ \vdots \\ g_t(s_J) \end{bmatrix}$$

- Consumption policy $c = \pi(s, \cdot) \Rightarrow J \times J$ transition matrix for s

$$\mathbf{A}_{\pi, z_t} \quad \text{with entries} \quad \Pr(s_{i,t+1} = s_{j'} | s_{i,t} = s_j) = \mathcal{T}_s(s_{j'} | s_j, \pi(s_j, \cdot), z_t, p_t)$$

- Law of motion for distribution $\mathbf{g}_t$ (Chapman-Kolmogorov equation)

$$\mathbf{g}_{t+1} = \mathbf{A}_{\pi, z_t}^{\top} \mathbf{g}_t, \quad z_{t+1} \sim \mathcal{T}_z(\cdot | z_t)$$

- Note: high-dimensional state $(\mathbf{g}_t, z_t)$ is Markov

Key difficulty: equilibrium prices are not Markov

- Equilibrium prices satisfy

$$
\begin{aligned}
p_t &= P^*(\mathbf{g}_t, z_t) \\
\mathbf{g}_{t+1} &= \mathbf{A}_{\pi,z_t}^{\top}\mathbf{g}_t \\
z_{t+1} &\sim \mathcal{T}_z(\cdot|z_t)
\end{aligned}
$$

- Difficulty: while $(\mathbf{g}_t, z_t)$ is Markov, low-dimensional p_t is not Markov!

- Dynamic programming can only handle Markov states $\Rightarrow$ Master equation

$$
V(s, \mathbf{g}, z) = \max_c u(c) + \beta \mathbb{E}\left[V(s', \mathbf{g}', z')|s, \mathbf{g}, z\right] \quad \text{s.t.} \quad s' \sim \mathcal{T}_s(\cdot|s, c, z, P^*(\mathbf{g}, z))
$$

- Without Markov transition prob's: cannot even write Bellman equation!

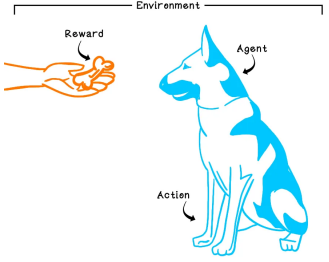
- But what if there was a way to approximate value and policy functions with p_t process for which there are **no Markov transition probabilities?**

Brief primer on reinforcement learning

RL: learning value & policy functions in incompletely-known Markov decision processes from Monte Carlo simulation (a.k.a. “approximate DP”)

Playing Atari with Deep Reinforcement Learning

Volodymyr Mnih Koray Kavukcuoglu David Silver Alex Graves Ioannis Antonoglou
Daan Wierstra Martin Riedmiller
DeepMind Technologies
{vlad,koray,david,alex.graves,ioannis,daan,martin.riedmiller} @ deepmind.com



Computing an expected value

Random variable x

How compute expected value $\mathbb{E}[x]$? Two approaches:

1. **Exact:** know probability distribution $f(x) \Rightarrow$ calculate

$$\mathbb{E}[x] = \int x f(x) dx$$

2. **Monte Carlo:** don't know f but can sample $\{x_1, x_2, \ldots, x_N\}$

$$\mathbb{E}[x] \approx \bar{x} = \frac{1}{N} \sum_{n=1}^{N} x_n$$

Or update incrementally (stochastic approximation method):

$$\bar{x}_k = \frac{1}{k} \sum_{n=1}^{k} x_n \quad \text{satisfies} \quad \bar{x}_k = \bar{x}_{k-1} + \frac{1}{k} (x_k - \bar{x}_{k-1}), \quad \frac{1}{k} = \text{“learning rate”}$$

Computing a value function

For now: eliminate actions and individual states

$$v(p_0) = \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t u(p_t)\right], \quad p_t = \text{exogenous stochastic process}$$

Two approaches:

1. **Dynamic programming:** p_t Markov and know $f(p'|p)$

$$v(p) = u(p) + \beta \int v(p') f(p'|p) dp'$$

Computing a value function

For now: eliminate actions and individual states

$$v(p_0) = \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t u(p_t)\right], \quad p_t = \text{exogenous stochastic process}$$

Two approaches:

1. **Dynamic programming:** p_t Markov and know $f(p'|p)$

$$v(p) = u(p) + \beta \int v(p') f(p'|p) dp'$$

2. **Monte Carlo:** don't know f but sample N trajectories $\{p_t^n\}_{t=0}^T$

$$v(p_0) \approx \widehat{v}(p_0) = \frac{1}{N} \sum_{n=1}^{N} \sum_{t=0}^{T} \beta^t u(p_t^n)$$

Computing a value function

For now: eliminate actions and individual states

$$v(p_0) = \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t u(p_t)\right], \quad p_t = \text{exogenous stochastic process}$$

Two approaches:

1. **Dynamic programming:** p_t Markov and know $f(p'|p)$

$$v(p) = u(p) + \beta \int v(p') f(p'|p) dp'$$

2. **Monte Carlo:** don't know f but sample N trajectories $\{p_t^n\}_{t=0}^T$

$$v(p_0) \approx \widehat{v}(p_0) = \frac{1}{N} \sum_{n=1}^{N} \sum_{t=0}^{T} \beta^t u(p_t^n)$$

Can also extend to compute optimal policy (Howard): policy gradient method

Sidestepping the Master Equation via RL

Sidestepping the Master Equation via RL

Recall: states $s = (b, y)$, price $p = q$, agents choose $c_{i,t}$ to maximize

$$v_{i,0} = \max_{\{c_{i,t}\}} \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t u(c_{i,t})\right] \text{ s.t. } s_{i,t+1} \sim \mathcal{T}_s(\cdot|s_{i,t}, c_{i,t}, z_t, p_t), \quad p_t = P^*(G_t, z_t)$$

Assumption 1: agents observe prices p_t but not distribution $G_t(s)$ ▸ Wold representation

Assumption 2: consumption policy π does not condition on price histories

$$c_{i,t} = \pi(s_{i,t}, z_t, p_t)$$

Extension: keep track of price histories (h lags or RNN) $\Rightarrow$ similar results

Similarity to standard RL: don't know transition probabilities of p_t but can sample

Difference to standard RL: agents know individual dynamics

- know $u, \mathcal{T}_s$: utility function & budget constraint, RE about income process
- want hybrid method that takes advantage of this structural knowledge

Restricted perceptions equilibrium

A pair of mappings (π^*, P^*) constitutes a restricted perceptions equilibrium if:

1. **Optimality:** For any price sequence $\{p_t\}$ generated by $p_t = P^*(\mathbf{g}_t, z_t)$ and exogenous sequence $\{z_t\}$, agents choose $\pi^*(s, z, p)$ to solve:

$$\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t u(\pi(s_t, z_t, p_t))\right],$$

   subject to the individual budget constraint and state transition equations.

2. **Market clearing:** For every period t, all markets clear.

3. **Consistency:** The prices that agents use to form expectations coincide with the prices in the simulated economy when all agents follow π^*:

$$p_t = P^*(\mathbf{g}_t, z_t), \qquad \mathbf{g}_{t+1} = \mathbf{A}^{\top}_{\pi^*(p_t, z_t)} \mathbf{g}_t$$

Simulating the economy given policy $\pi(s, z, p)$

For given (suboptimal) policy $\pi(s, z, p)$, can simulate economy forward in time

- RL parlance: "rollout" of policy π = agents interacting with environment
- important: this is very cheap computationally

Recall: discrete $s \Rightarrow$ vectors $\boldsymbol{\pi}(z, p), \mathbf{g}_t$, sparse transition matrix $\mathbf{A}_{\pi(z,p)}$

For given policy $\boldsymbol{\pi}(z, p)$ and $(\mathbf{g}_0, z_0)$, economy evolves as:

$$
\begin{aligned}
p_t &= P^*(\mathbf{g}_t, z_t) \\
\mathbf{g}_{t+1} &= \mathbf{A}^\top_{\pi(z_t,p_t)}\mathbf{g}_t \\
z_{t+1} &\sim \mathcal{T}_z(\cdot|z_t)
\end{aligned}
$$

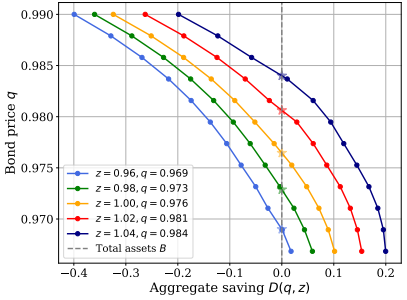
Efficient handling of non-trivial market clearing

$$D_t(p, z) = 0, \qquad D_t(p, z) := \int b'(s, z, p) dG_t(s) = \text{agg. demand for bonds}$$

Key: policies $b'(s, z, p)$ double as individual demand functions

When rolling out $\boldsymbol{\pi}(z, p) = \{\mathbf{c}(z, p), \mathbf{b}'(z, p)\}$, simply clear markets along way!

$$p_t \text{ solves } D_t(p_t, z_t) = \mathbf{b}'(z_t, p_t)^\top \mathbf{g}_t = 0$$



Market clearing is part of environment, not some outer loop!

Using structural knowledge of individual dynamics

Value function for given policy $\pi(s, z, p) = \{c(s, z, p), b'(s, z, p)\}$

$$v_\pi(s, z, p) = \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t u(c(s_{i,t}, z_t, p_t)) \middle| s_{i,0} = s, z_0 = z, p_0 = p\right] \qquad (*)$$

Partition state space (s, z, p) into known dynamics and unknown dynamics

- use transition matrix $\mathbf{A}$ to keep track of all s-transitions
- expectation $\mathbb{E}$ only over trajectories of $\{p_t\}_{t=0}^{\infty}$ and $\{z_t\}_{t=0}^{\infty}$

Write $v_\pi(s, z, p)$ in $(*)$ as vector $\mathbf{v}_\pi(z, p)$:

$$\mathbf{v}_\pi(z, p) = \mathbb{E}\left[\mathbf{u}_0 + \beta \mathbf{A}_0 \mathbf{u}_1 + \beta^2 \mathbf{A}_0 \mathbf{A}_1 \mathbf{u}_2 + \ldots | z, p\right] = \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t \mathbf{A}_{0\to t} \mathbf{u}_t \middle| z, p\right]$$

where $\mathbf{u}_t = u(\boldsymbol{c}(z_t, p_t))$ and $\mathbf{A}_t = \mathbf{A}_{\pi(z_t, p_t)}$ and $\mathbf{A}_{0\to t} = \mathbf{A}_0 \cdots \mathbf{A}_{t-1}$

Summary: problem to be solved (important slide)

Find optimal policy $\pi(s, z, p)$ or $\boldsymbol{\pi}(z, p) = \{\boldsymbol{c}(z, p), \boldsymbol{b}'(z, p)\}$ that maximizes

$$\mathbf{v}_\pi(z, p) = \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t \mathbf{A}_{\pi, 0\to t} u(\boldsymbol{c}(z_t, p_t)) \middle| z_0 = z, p_0 = p\right]$$

taking as given evolution of equilibrium prices p_t (stop-gradient operator)

$$p_t = P^*(\mathbf{g}_t, z_t), \quad \mathbf{g}_{t+1} = \mathbf{A}^\top_{\pi(z_t, p_t)} \mathbf{g}_t, \quad z_{t+1} \sim \mathcal{T}_z(\cdot|z_t), \quad t = 0, 1, \ldots$$

with $(\mathbf{g}_0, z_0)$ given and $\mathbf{A}_{\pi, 0\to t} = \mathbf{A}_{\pi(z_0, p_0)} \cdots \mathbf{A}_{\pi(z_{t-1}, p_{t-1})}$

Key observation:

- State of economy = $(\mathbf{g}, z)$ = very high-dimensional
- But state in value/policy functions = (s, z, p) = very **low-dimensional!**
- No perceived law of motion, inner loop / outer loop (like in Krusell-Smith)
- GE problem only mildly more difficult than PE

Structural policy gradient (SPG) method for maximizing $\mathbf{v}_\pi$

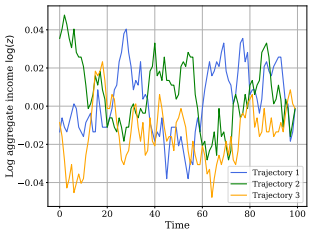
Find optimal policy $\boldsymbol{\pi}(z, p)$ that maximizes estimate of $\mathbb{E}_{z_0 \sim \psi_z, p_0 \sim \psi_p}[\mathbf{v}_{\boldsymbol{\pi}}(z_0, p_0)]$:

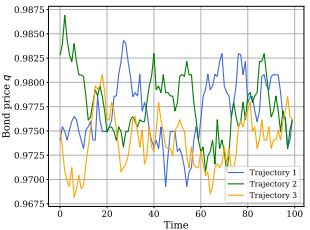
$$\widehat{\mathbf{v}}_{\boldsymbol{\pi}} = \frac{1}{N} \sum_{n=1}^{N} \left[\sum_{t=0}^{T} \beta^t \mathbf{A}^n_{\boldsymbol{\pi}, 0 \to t} u(\boldsymbol{c}(z_t^n, p_t^n)) \right]$$

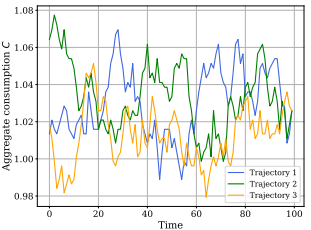
with N price trajectories p_t^n sampled from interacting with environment (rollouts):

$$p_t^n = P^*(\mathbf{g}_t^n, z_t^n), \quad \mathbf{g}_{t+1}^n = \mathbf{A}^{\mathsf{T}}_{\boldsymbol{\pi}(z_t^n, p_t^n)} \mathbf{g}_t^n, \quad z_{t+1}^n \sim \mathcal{T}_z(\cdot | z_t^n), \quad t = 0, 1, \ldots$$

with $\mathbf{g}_0^n \sim \psi_g(\cdot)$, $z_0^n \sim \psi_z(\cdot)$ and $\mathbf{A}^n_{\boldsymbol{\pi}, 0 \to t} = \mathbf{A}_{\boldsymbol{\pi}(z_0^n, p_0^n)} \cdots \mathbf{A}_{\boldsymbol{\pi}(z_{t-1}^n, p_{t-1}^n)}$







Structural policy gradient (SPG) method for maximizing $\mathbf{v}_\pi$

Find optimal policy $\boldsymbol{\pi}(z, p)$ that maximizes estimate of $\mathbb{E}_{z_0\sim\psi_z, p_0\sim\psi_p}[\mathbf{v}_\pi(z_0, p_0)]$:

$$\widehat{\mathbf{v}}_\pi = \frac{1}{N}\sum_{n=1}^{N}\left[\sum_{t=0}^{T}\beta^t \mathbf{A}^n_{\pi,0\to t} u(\boldsymbol{c}(z_t^n, p_t^n))\right]$$

with N price trajectories p_t^n sampled from interacting with environment (rollouts):

$$p_t^n = P^*(\mathbf{g}_t^n, z_t^n), \quad \mathbf{g}_{t+1}^n = \mathbf{A}^\top_{\pi(z_t^n, p_t^n)}\mathbf{g}_t^n, \quad z_{t+1}^n \sim \mathcal{T}_z(\cdot|z_t^n), \quad t = 0, 1, \ldots$$

with $\mathbf{g}_0^n \sim \psi_g(\cdot)$, $z_0^n \sim \psi_z(\cdot)$ and $\mathbf{A}^n_{\pi,0\to t} = \mathbf{A}_{\pi(z_0^n, p_0^n)} \cdots \mathbf{A}_{\pi(z_{t-1}^n, p_{t-1}^n)}$

In practice, maximize scalar objective using gradient ascent:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbf{d}_0^\top \widehat{\mathbf{v}}_\pi = \sum_{j=1}^{J} d_0(s_j)\widehat{v}_\pi(s_j), \quad d_0(s) = \text{uniform dist. over } s$$

Policy either grid based $\boldsymbol{\theta} = [\boldsymbol{\pi}(z_1, p_1), \ldots, \boldsymbol{\pi}(z_K, p_L)]$ or neural net $\pi(s, z, p; \boldsymbol{\theta})$

- low dimensional $\Rightarrow$ grid-based method works well, no need for neural nets!

Algorithm 1: Structural Policy Gradient Method

Input: Initial policy parameters $\boldsymbol{\theta}^0$; step size sequence $\{\eta_k\}$; number of simulated trajectories N; horizon T. **Output:** Optimal policy parameters θ^*.

1. Initialize parameters $\boldsymbol{\theta}^0$.
2. For each iteration $k = 0, 1, 2, \ldots$:
   - 2.1 Simulate N trajectories

$$\{(z_t^n, p_t^n, \mathbf{g}_t^n)\}_{t=0}^T, \qquad n = 1, \ldots, N,$$

   using policy $\boldsymbol{\pi}(\cdot; \boldsymbol{\theta}^k)$ and market clearing conditions.
   - 2.2 Compute the sample objective

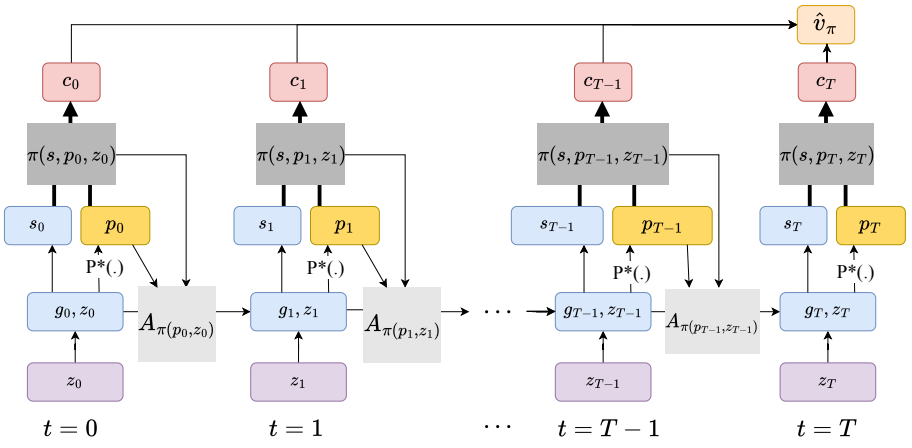
$$\mathcal{L}(\boldsymbol{\theta}^k) = \boldsymbol{d}_0^\top \widehat{\boldsymbol{v}}_\pi \quad \text{where} \quad \widehat{\boldsymbol{v}}_\pi = \frac{1}{N} \sum_{n=1}^N \left[\sum_{t=0}^T \beta^t \, \mathbf{A}_{\pi, 0\to t}^n \, u(\boldsymbol{c}(z_t^n, p_t^n)) \right].$$

   - 2.3 Update parameters by stochastic gradient ascent:

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k + \eta_k \, \nabla_\theta \mathcal{L}(\boldsymbol{\theta}^k).$$

   - 2.4 Stop when convergence criteria are met.

Computational graph for construction of $\widehat{\mathbf{v}}_\pi$



Computational experiments

Runtimes

- Efficient implementation in JAX for GPUs, run on Google Colab
- Algorithm = stochastic (Monte Carlo) $\Rightarrow$ present averages over multiple runs

Model	Average converge epoch	# Runs	Average Runtime (sec)
Krusell-Smith	462.3	10	36.77
Huggett with agg. shocks	573.0	10	45.91
Portfolio choice	637.5	10	84.3
HANK with agg. shocks	707.5	10	246.49
Partial Equilibrium (Huggett)	355.8	10	24.50

Note: all experiments were implemented on the A100 GPU on Google Colab

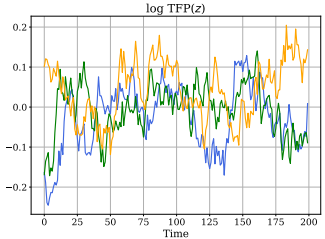
Krusell-Smith model

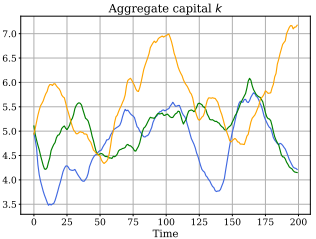
Computational experiments: Krusell-Smith model

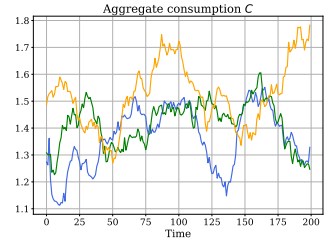
Parameter	Description	Value
α	Capital share	0.36
δ	Capital depreciation rate	0.08
γ	Discount factor	0.95
σ	Coefficient of relative risk aversion	3
ρ_z	Persistence of AR(1) for z_t (log TFP)	0.9
ν_z	Volatility of AR(1) for z_t (log TFP)	0.03

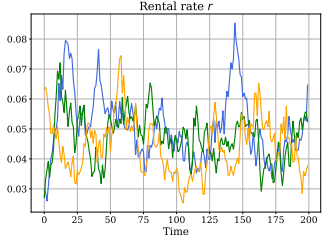
Hyperparameter	Description	Value
J_{s_1}	Number of s_1 (wealth) grid points	200
J_{s_2}	Number of s_2 (income) states	3
J_{p_1}	Number of p_1 (rental rate) grid points	50
J_{p_2}	Number of p_2 (wage) grid points	70
N	Sample size = number of p trajectories	256,512,1024,...
T	Time truncation s.t. $\beta^T < 0.01$	90
ϵ	Convergence criterion on $\widehat{v}_\pi$	0.001
$\eta_{\texttt{ini}}$	Initial learning rate	0.01
$\eta_{\texttt{decay}}$	Learning rate decay (exponential)	0.5

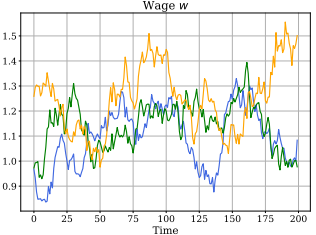
Some simulated trajectories under the optimal policy

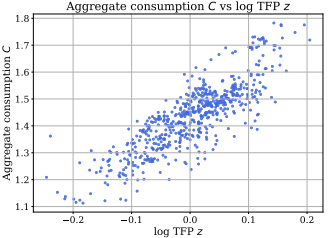




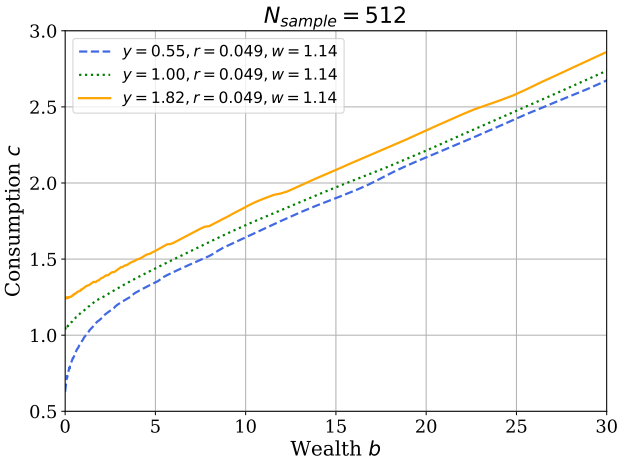




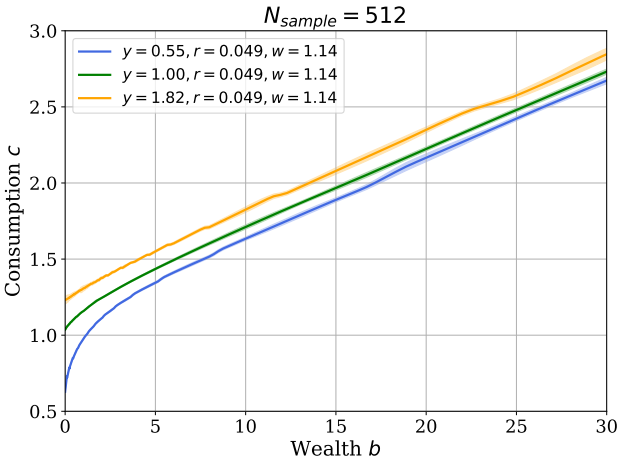




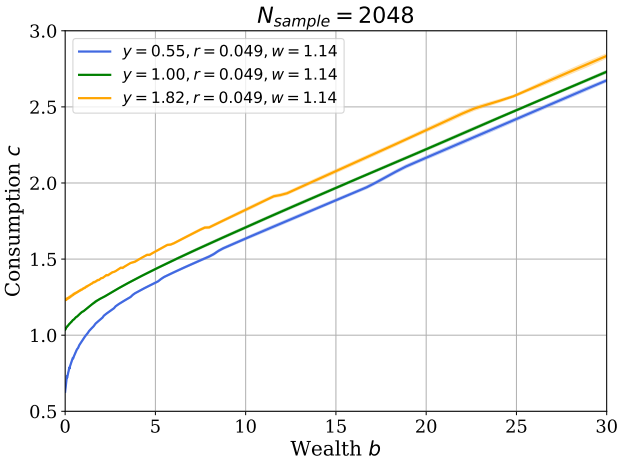
Consumption policy function: single run



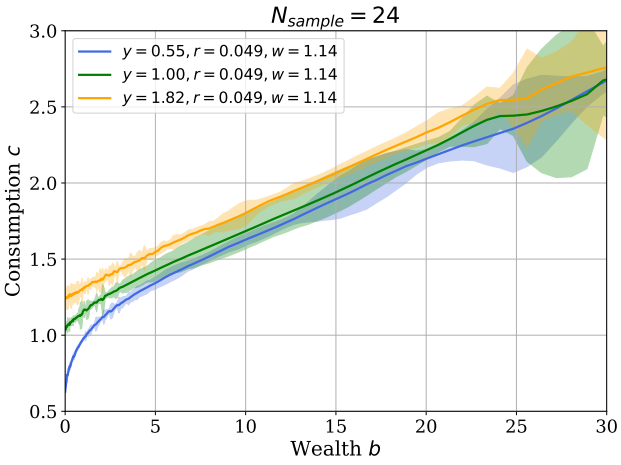
Consumption policy function: multiple runs



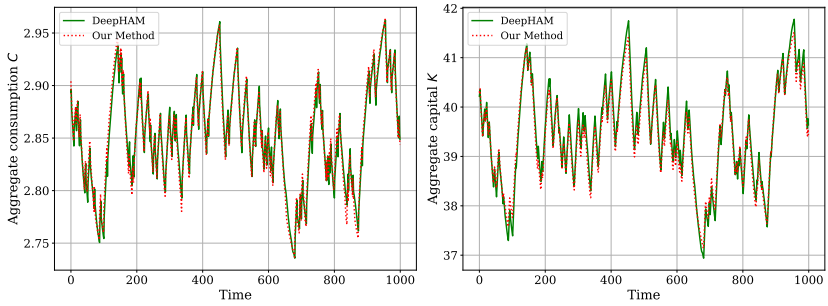
Larger sample size $N \Rightarrow$ more precise estimate



Smaller sample size $N \Rightarrow$ noisier estimate



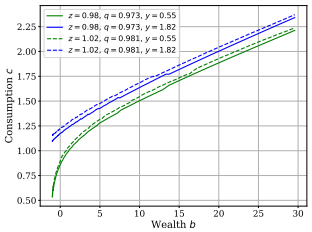
Structural RL method recovers RE solutions

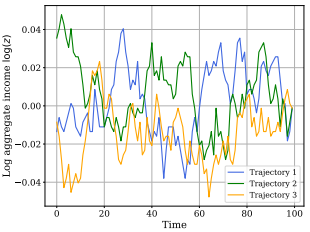


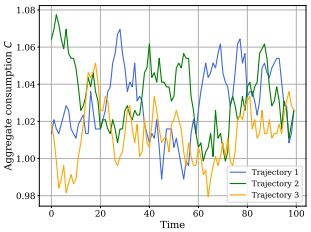
RE solutions are obtained with a deep learning based method (DeepHAM).

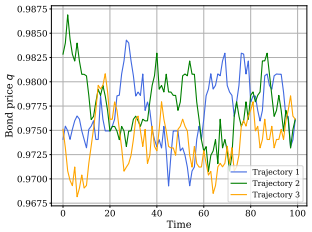
Huggett with aggregate risk

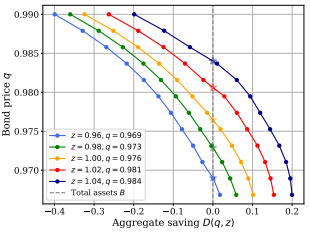
Some simulated trajectories under the optimal policy

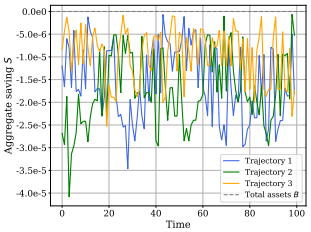




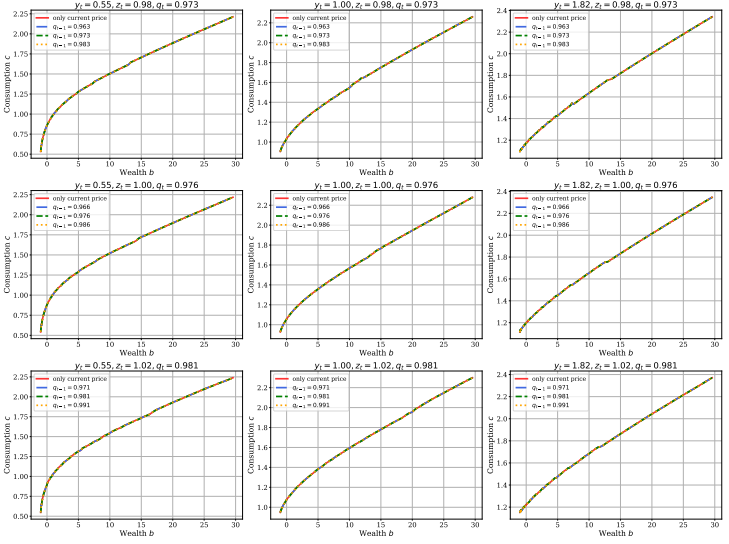








Huggett: adding one lagged price p_{t-1} into state space



Portfolio Choice

Portfolio choice: Krusell-Smith 1997

- RBC model with household portfolio choice
- Three individual states $(b_{i,t}, k_{i,t}, y_{i,t})$, three prices (q_t, R_t, w_t)
- Households choose consumption $c_{i,t}$ to maximize

$$v_{i,0} = \max_{\{c_{i,t}\}} \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t u(c_{i,t}) \quad \text{subject to}$$

$$c_{i,t} + q_t b_{i,t+1} + k_{i,t+1} = b_{i,t} + R_t k_{i,t} + w_t y_{i,t}, \quad y_{i,t+1} \sim \mathcal{T}_y(\cdot|y_{i,t}), \quad b_{i,t+1} \geq \underline{b}$$

Note: capital price $q_t^k = 1$ for usual reasons

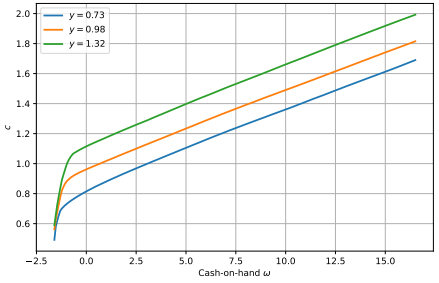
- Rental rate and wage

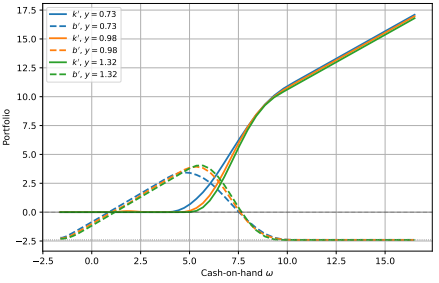
$$R_t = \alpha z_t K_t^{\alpha-1} + 1 - \delta, \quad w_t = (1-\alpha) z_t K_t^{\alpha}, \quad K_t = \int k \; dG_t(b,k,y)$$

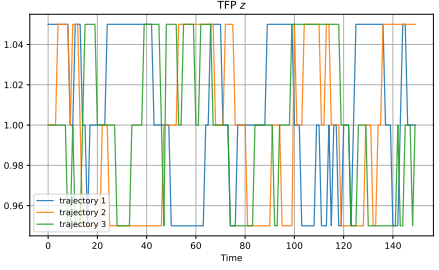
- Bond price q_t such that

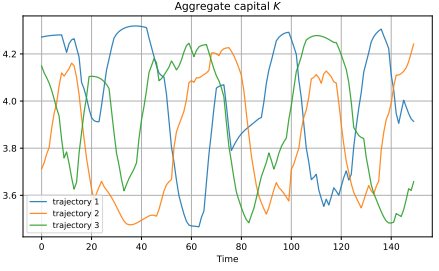
$$\int b_t'(b,k,y) \; dG_t(b,k,y) = 0, \quad \text{all } t$$

Some simulated trajectories under the optimal policy









HANK with forward-looking Phillips curve

HANK with forward-looking Phillips curve

Household block (similar to before): states $s = (b, y)$ and prices $p = (q, w)$

$$\text{policies} = \{c(s,p), n(s,p)\} \quad \text{that maximize PDV of utility}$$

Firm block: price setting $\Rightarrow$ forward-looking Phillips curve = added difficulty

$$\Pi_t = \frac{\varepsilon}{\theta}\left(\frac{w_t}{z_t} - m^*\right) + \mathbb{E}\left[\Lambda_{t\to t+1}\frac{Y_{t+1}}{Y_t}\Pi_{t+1}\middle|\mathcal{I}_t\right], \qquad m^* = \frac{\varepsilon - 1}{\varepsilon}$$

Conventional approach: parameterize $\mathbb{E}[\Pi_{t+1}|\mathcal{I}_t]$ $\Rightarrow$ complicated fixed-point
(e.g. Kase-Melosi-Rottner)

HANK with forward-looking Phillips curve

Household block (similar to before): states $s = (b, y)$ and prices $p = (q, w)$

$$\text{policies} = \{c(s, p), n(s, p)\} \quad \text{that maximize PDV of utility}$$

Firm block: price setting $\Rightarrow$ forward-looking Phillips curve = added difficulty

Our solution: solve firm price-setting problem using SPG method

$$J_{j,0} = \max_{\{P_{j,t}\}} \mathbb{E}_0 \left[\sum_{t=0}^{\infty} \Lambda_{0\to t} \left\{ \text{Profits} \left(\frac{P_{j,t}}{P_t}, \frac{w_t}{z_t}, Y_t \right) - \frac{\theta}{2} \left(\frac{P_{j,t} - P_{j,t-1}}{P_{j,t-1}} \right)^2 \right\} \right]$$

HANK with forward-looking Phillips curve

Household block (similar to before): states $s = (b, y)$ and prices $p = (q, w)$

$$\text{policies} = \{c(s,p), n(s,p)\} \quad \text{that maximize PDV of utility}$$

Firm block: price setting $\Rightarrow$ forward-looking Phillips curve = added difficulty

Or in terms of relative price $\phi_{j,t} = P_{j,t}/P_t$ and inflation $\Pi_{j,t} = (P_{j,t} - P_{j,t-1})/P_{j,t}$

$$J_{j,0} = \max_{\{\Pi_{j,t}\}} \mathbb{E}_0 \left[\sum_{t=0}^{\infty} \Lambda_{0\to t} \left\{ \text{Profits} \left(\phi_{j,t}, \frac{w_t}{z_t}, Y_t \right) - \frac{\theta}{2} \left(\Pi_{j,t} \right)^2 \right\} \right], \quad \phi_{j,t} = \frac{1+\Pi_{j,t}}{1+\Pi_t} \phi_{j,t-1}$$

HANK: policy gradient method for both households & firms

Household block (similar to before): states $s = (b, y)$ and prices $p = (q, w)$

$$\text{policies} = \{c(s, p), n(s, p)\} \quad \text{that maximize PDV of utility}$$

Firm block: states (z, Y) and prices $p = (q, w)$

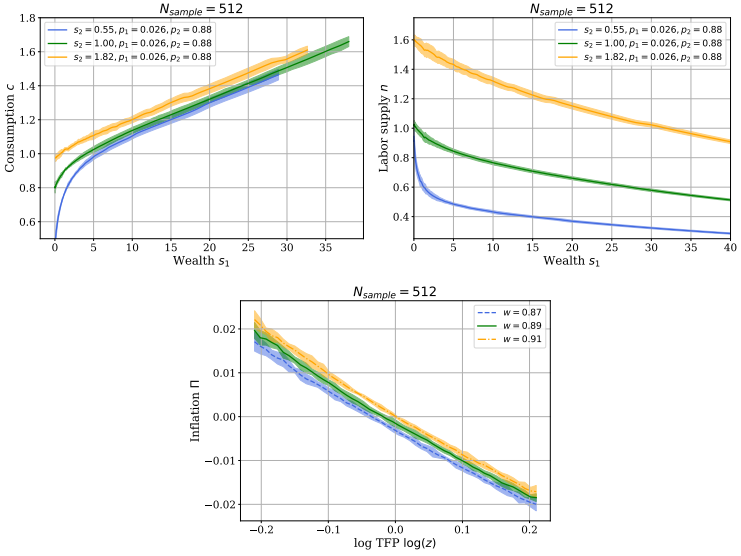
$$\text{policy} = \Pi_j(z, Y, p) \quad \text{that maximizes}$$

$$J_{j,\Pi} = \mathbb{E}_0 \left[\sum_{t=0}^{\infty} \Lambda_{0\to t} \left\{ \text{Profits} \left(\phi_{j,t}, \frac{w_t}{z_t}, Y_t \right) - \frac{\theta}{2} \left(\Pi_j(z_t, Y_t, p_t) \right)^2 \right\} \right], \quad \phi_{j,t} = \frac{1 + \Pi_j(z_t, Y_t, p_t)}{1 + \Pi_t} \phi_{j,t-1}$$

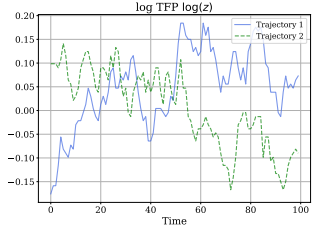
Symmetric treatment of firms and households, update policies simultaneously

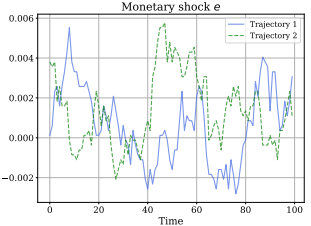
In practice: good convergence properties

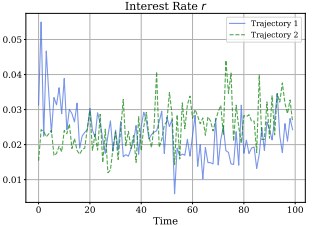
HANK: Household and firm policy functions

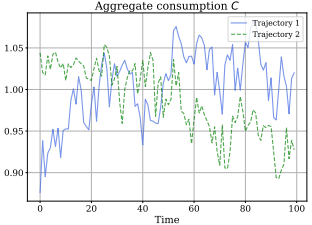


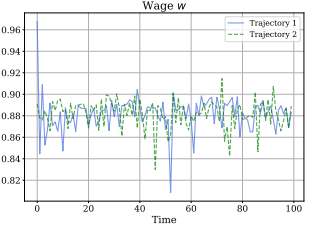
HANK simulations

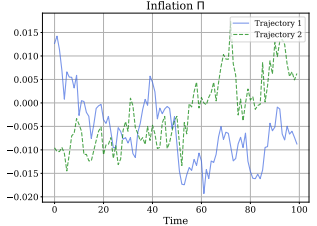












Summary

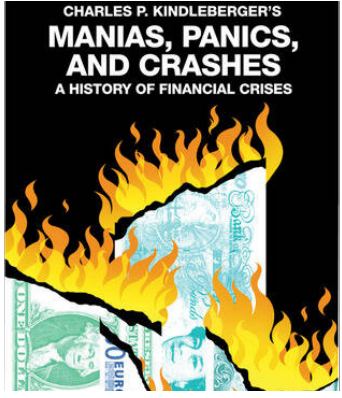
Efficient and flexible global solution method for HA models with aggregate risk

"Structural RL": hybrid RL method that exploits known model structure

- RL about eqm prices but not individual states
- sidestep infinite-dimensional Master equation
- solve much lower-dimensional problem

Solves problems traditional methods struggle with

- non-trivial market clearing conditions
- HANK with forward-looking Phillips curve
- next: models of large crises, booms/busts



Thanks!

In stationary world, lagged prices are enough for RE

back

Recall **Assumption 1:** agents observe prices p_t but not distribution $G_t(s)$

Important: Assumption 1 still consistent with rational expectations

Why? Wold representation theorem!

Step 1 (Wold): if p_t-process is stationary, it has Wold representation = VMA(∞)

$$p_t = \sum_{j=0}^{\infty} c_j \varepsilon_{t-j}, \qquad c_j = \text{some unknown coefficients}$$

Step 2: if VMA(∞) is invertible, it can be expressed as a VAR(∞) and hence

$$p_{t+1} \sim \mathcal{T}_p(\cdot|p_t, p_{t-1}, ...)$$

In practice, include finitely many lags

Assumption 2: extreme case with zero lags $p_{t+1} \sim \mathcal{T}_p(\cdot|p_t)$