Discussion of “Self-Justified Equilibria: Existence and Computation”

Thomas Winberry
Chicago Booth and NBER

Robustness in Economics and Econometrics
April 5th, 2019

Overview

- Heterogeneous agent macro models: distribution $\in$ state variable
    - Project distribution onto low-dimensional subspace

Two main contributions of this paper

1. Computational algorithm: new subspace + projection method
    - Compare to other methods in the literature
2. Economic interpretation of resulting approximation based on bounded rationality
    - How does this help us interpret the approximation?

Benchmark Model: Krusell and Smith (1998)

$$Y_t = e^{Z_t} K_t^{\alpha} \overline{L}^{1-\alpha}, \quad K_t = \int a g_t(a,z) da dz$$

$$w_t = (1-\alpha) e^{Z_t} K_t^{\alpha} \overline{L}^{-\alpha}, \quad r_t = \alpha e^{Z_t} K_t^{\alpha-1} \overline{L}^{1-\alpha} - \delta$$

$$v_t(a,z) = \max_{c,a'} u(c) + \beta \mathbb{E}[v_{t+1}(z',a')|z,Z] \text{ s.t. } c + a' = w_t z + r_t a$$

$$g_{t+1}(a',z') = \text{consistent with } a'(a,z) \text{ and } z'|z$$

Challenge: aggregate state = $(Z_t, g_t(z,a))$

1. Decision rules are functions of g_t
2. Law of motion for g_t

Benchmark Model: Krusell and Smith (1998)

$$Y_t = e^{Z_t} K_t^{\alpha} \overline{L}^{1-\alpha}, \quad K_t = \sum_{i,j} a_i g_t(a_i, z_j)$$

$$w_t = (1-\alpha) e^{Z_t} K_t^{\alpha} \overline{L}^{-\alpha}, \quad r_t = \alpha e^{Z_t} K_t^{\alpha-1} \overline{L}^{1-\alpha} - \delta$$

$$v_t(a_i, z_j) = \max_{c,a'} u(c) + \beta \mathbb{E}[v_{t+1}(z', a') | z_j, Z] \text{ s.t. } c + a' = w_t z_j + r_t a_i$$

$$g_{t+1}(z', a') = \text{consistent with } a'(a, z) \text{ and } z' | z$$

Challenge: aggregate state = $(Z_t, \mathbf{g}_t = \left[g(a_i, z_j)\right]_{i,j})$

1. Decision rules are functions of g_t
2. Law of motion for g_t

Method 1: Krusell and Smith (1998)

1. **Subspace on which distribution $g_t(a, z)$ is projected**

2. **How projection is performed**

Method 1: Krusell and Smith (1998)

1. **Subspace on which distribution** $g_t(a, z)$ **is projected**
    - Approximate distribution with aggregate capital K_t
    - Can view as $K_t = \hat{g}_t = X^{\mathrm{T}} g_t$, where $X = \begin{bmatrix} \mathbf{a} \\ \vdots \\ \mathbf{a} \end{bmatrix}$

2. **How projection is performed**

Method 1: Krusell and Smith (1998)

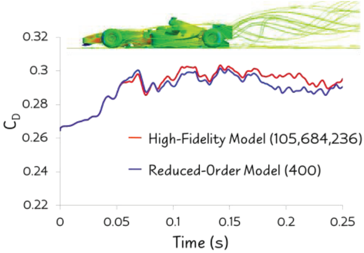
1. **Subspace on which distribution $g_t(a, z)$ is projected**
    - Approximate distribution with aggregate capital K_t
    - Can view as $K_t = \hat{g}_t = X^{\mathrm{T}} g_t$, where $X = \begin{bmatrix} \mathbf{a} \\ \vdots \\ \mathbf{a} \end{bmatrix}$
2. **How projection is performed**
    - Assume parametric form $\log K_{t+1} = \gamma_0 + \gamma_1 Z_t + \gamma_2 \log K_t$
    - Given coefficients $(\gamma_0, \gamma_1, \gamma_2)$:
        - Solve decision rules $v(a, z; Z, K)$
        - Simulate model $\implies \{Z_t, K_t\}_{t=0}^{T}$
        - Update coefficients $(\gamma_0, \gamma_1, \gamma_2)$ using OLS

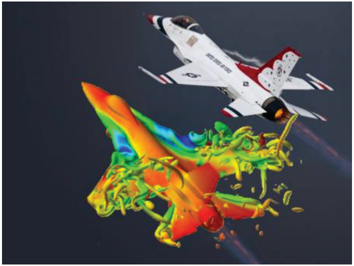
Method 1: Krusell and Smith (1998)

1. **Subspace on which distribution $g_t(a, z)$ is projected**
   - Approximate distribution with aggregate capital K_t
   - Can view as $K_t = \hat{g}_t = X^{\mathrm{T}} g_t$, where $X = \begin{bmatrix} \mathbf{a} \\ \vdots \\ \mathbf{a} \end{bmatrix}$
2. **How projection is performed**
   - Assume parametric form $\log K_{t+1} = \gamma_0 + \gamma_1 Z_t + \gamma_2 \log K_t$
   - Given coefficients $(\gamma_0, \gamma_1, \gamma_2)$:
     - Solve decision rules $v(a, z; Z, K)$
     - Simulate model $\implies \{Z_t, K_t\}_{t=0}^{T}$
     - Update coefficients $(\gamma_0, \gamma_1, \gamma_2)$ using OLS

- **Strengths**: straightforward, global approximation
- **Weaknesses**: slow, small aggregate state

Method 2: Ahn, Kaplan, Moll, Winberry, and Wolf (2017)





Method 2: Ahn, Kaplan, Moll, Winberry, and Wolf (2017)

1. **Subspace on which distribution** $g_t(a, z)$ **is projected**
    - Approximate $\hat{g}_t = X^{\mathrm{T}} g_t$, where $X = [\mathbf{x}_1, ..., \mathbf{x}_k]$
    - Choose X to match first k periods of IRF of K_t to Z_t shock
    - Model reduction literature calls this “moment matching”
      (Alternative: first k principle components of g_t)

Method 2: Ahn, Kaplan, Moll, Winberry, and Wolf (2017)

1. **Subspace on which distribution $g_t(a, z)$ is projected**
   - Approximate $\hat{g}_t = X^{\mathrm{T}} g_t$, where $X = [\mathbf{x}_1, ..., \mathbf{x}_k]$
   - Choose X to match first k periods of IRF of K_t to Z_t shock
   - Model reduction literature calls this "moment matching"
     (Alternative: first k principle components of g_t)

2. **How projection is performed**
   - Linearize dynamics of $g_{t+1} = Ag_t + bZ_t$
     $\implies$ reduced system dynamics $\hat{g}_{t+1} = X^{\mathrm{T}} AX\hat{g}_t + X^{\mathrm{T}} bZ_t$
   - Model reduction literature: how to compute X from A and b

Method 2: Ahn, Kaplan, Moll, Winberry, and Wolf (2017)

1. **Subspace on which distribution $g_t(a, z)$ is projected**
    - Approximate $\hat{g}_t = X^{\mathrm{T}} g_t$, where $X = [\mathbf{x}_1, \ldots, \mathbf{x}_k]$
    - Choose X to match first k periods of IRF of K_t to Z_t shock
    - Model reduction literature calls this "moment matching"

    (Alternative: first k principle components of g_t)

2. **How projection is performed**
    - Linearize dynamics of $g_{t+1} = Ag_t + bZ_t$

      $\Longrightarrow$ reduced system dynamics $\hat{g}_{t+1} = X^{\mathrm{T}} A X \hat{g}_t + X^{\mathrm{T}} b Z_t$
    - Model reduction literature: how to compute X from A and b

- **Strengths**: fast, smart aggregate state
- **Weaknesses**: local linear approximation w.r.t. aggregate state

Method 3: Kubler and Scheidegger (2019)

1. **Subspace on which distribution $g_t(a, z)$ is projected**
   - Approximate $\widehat{f}(g_t) = X^{\mathrm{T}} g_t$, where $X = [\mathbf{x}_1, \ldots, \mathbf{x}_d]$
   - Choose X to minimize (clever measure of) average forecast error of $\hat{f}(g_t)$
   - Choose d based on patterns of eigenvalues

Method 3: Kubler and Scheidegger (2019)

1. **Subspace on which distribution $g_t(a, z)$ is projected**
    - Approximate $\widehat{f}(g_t) = X^{\mathrm{T}} g_t$, where $X = [\mathbf{x}_1, ..., \mathbf{x}_d]$
    - Choose X to minimize (clever measure of) average forecast error of $\hat{f}(g_t)$
    - Choose d based on patterns of eigenvalues
2. **How projection is performed**: *kernel regression* $\widehat{f}_{t+1} = \mathcal{R}^{n,d}(Z_t, \widehat{f}_t)$
    - Given "coefficients" $\mathcal{R}^{n,d}$, solve decision rules $v(z, a; Z, \widehat{f})$
    - Simulate model $\implies \{Z_t, \widehat{f}_t\}$
    - Update coefficients $\mathcal{R}^{n,d}$ using kernel regression

Method 3: Kubler and Scheidegger (2019)

1. **Subspace on which distribution $g_t(a, z)$ is projected**
    - Approximate $\widehat{f}(g_t) = X^{\mathrm{T}} g_t$, where $X = [\mathbf{x}_1, ..., \mathbf{x}_d]$
    - Choose X to minimize (clever measure of) average forecast error of $\hat{f}(g_t)$
    - Choose d based on patterns of eigenvalues
2. **How projection is performed**: *kernel regression* $\widehat{f}_{t+1} = \mathcal{R}^{n,d}(Z_t, \widehat{f}_t)$
    - Given "coefficients" $\mathcal{R}^{n,d}$, solve decision rules $v(z, a; Z, \widehat{f})$
    - Simulate model $\implies \{Z_t, \widehat{f}_t\}$
    - Update coefficients $\mathcal{R}^{n,d}$ using kernel regression

- **Strengths**: smart aggregate state, global approximation
- **Weaknesses**: slow? scaleable?

Method 3: Kubler and Scheidegger (2019)

1. **Subspace on which distribution $g_t(a, z)$ is projected**
   - Approximate $\widehat{f}(g_t) = X^{\mathrm{T}} g_t$, where $X = [\mathbf{x}_1, \ldots, \mathbf{x}_d]$
   - Choose X to minimize (clever measure of) average forecast error of $\hat{f}(g_t)$
   - Choose d based on patterns of eigenvalues
2. **How projection is performed**: *kernel regression* $\widehat{f}_{t+1} = \mathcal{R}^{n,d}(Z_t, \widehat{f}_t)$
   - Given "coefficients" $\mathcal{R}^{n,d}$, solve decision rules $v(z, a; Z, \widehat{f})$
   - Simulate model $\implies \{Z_t, \widehat{f}_t\}$
   - Update coefficients $\mathcal{R}^{n,d}$ using kernel regression

- **Strengths**: smart aggregate state, global approximation
- **Weaknesses**: slow? scaleable?

$\implies$ *for which set of models is this method best choice?*

Interpretation of Numerical Solution

- Agents' computational cost $c(d, n)$ not used in computations
  1. Size of subspace d picked based on eigenvalues
  2. Observations in regression n picked based on accuracy

  $\implies$ typical approach in numerical methods

- $c(d, n)$ is implicitly "reverse engineered" to justify these choices

- **How does self-justified equilibrium help interpret the solution?**
  - Can always view numerical approximations as boundedly rational decision rules (e.g. Judd's Euler equation errors)

Other Food for Thought

1. **Why use complicated kernel regression over OLS?**
    - What is estimated smoothing parameter λ?
    - What happens with $\lambda = 0$?

2. **Issues with measure of forecast accuracy (Den Haan 2010)**
    - Average error could mask large errors in extreme (but interesting) states
    - One step ahead errors could mask errors accumulating over lifecycle

Conclusion: Nice Paper!

1. **What is the method's comparative advantage over existing lit?**
    - For which models should practitioners use this method?

2. **How does bounded rationality help interpretation?**