DeepSAM: Deep Learning for Search And Matching Models

Jonathan Payne   Yucheng Yang

Princeton   University of Zurich

DSE 2023

August 25, 2023

With outstanding research assistance from Adam Rebei

Introduction

- Deep learning can solve macro models with rich heterogeneity & aggregate shocks.
- Prominent examples: incomplete market heterogeneous agent models (**HAM**) (e.g. Krusell-Smith '98, Kaplan-Moll-Violante '18.)
- We study another model class with heterogeneity: search & matching (**SAM**) models.

	Distribution	How distribution affects agent's decision	Low-dim representation
HAM	Asset wealth and income	via aggregate prices	Typically yes
SAM	Type (productivity) of agents in two sides of matching	Matching probability with all types	No

Heterogeneity in Search and Matching (SAM) Models

- Previous literature: make assumptions (e.g. block recursivity) to get rid of distribution from state space.
- Deep learning handles high dimensional state: suitable for original SAM problem.
- We formulate SAM models with aggregate shocks as high dimensional PDEs, and develop deep learning method, DeepSAM, to solve them.

Table of Contents

Shimer-Smith Model with Non-Transferable Utility (NTU): Setting

- Shimer and Smith (2000) with NTU and aggregate productivity shocks.
- Continuous time, infinite horizon problem.
- Agents indexed by productivity $x \in [0,1]$ with exogenous density $l(x)$. Utility:

$$\mathbb{E}\left[\int_0^\infty re^{-rt}c_t\right]$$

- Agents are matched or unmatched. $u(x)$: unmatched mass function. Unmatched agents randomly meet a new partner in type set Y at rate $\rho \int_Y u(y)dy$.
- Symmetry between two sides of the market: both from the same population. (Applications: money-search, spatial geography.)

Shimer-Smith Model with NTU: Match and surplus division

- When two unmatched agents meet, they observe each other's type. They form a match if both accept.
- Unmatched agents get b goods; in a match (x, y), x gets $zf(x, y)$ and y gets $zf(y, x)$.
- z: two-state Poisson aggregate shock (can be generalized).
- Strategies: define matching function α s.t. $\alpha(x, y) = 1$ iff type x accept type y and vise versa.
- Matches dissolve exogenously at rate δ.

Recursive representation of equilibrium

- State variable: x, y, z, u. Note: $u(x)$ is mass of unemployed agents of type x.
- Master equation for unmatched agent's value $V(x, z, u)$:

$$\begin{aligned} rV(x,z,u) = rb &+ \rho \int_{\mathcal{Y}} \alpha(x,y,z,u)(W(x,y,z,u) - V(x,z,u))u(y)dy \\ &+ \sum_{\tilde{z} \neq z} \lambda_{z\tilde{z}}(V(x,\tilde{z},u) - V(x,z,u)) + D_u V(x,z,u) \cdot \mu_u(x,z,u) \end{aligned}$$

- Master equation of matched agent's value $W(x, y, z, u)$:

$$\begin{aligned} rW(x,y,z,u) = rzf(x,y) &- \delta(W(x,y,z,u) - V(x,z,u)) \\ &+ \sum_{\tilde{z} \neq z} \lambda_{z\tilde{z}}(W(x,y,\tilde{z},u) - W(x,y,z,u)) + D_u W(x,y,z,u) \cdot \mu_u(x,z,u) \end{aligned}$$

- $D_u V(x, z, u), D_u W(x, y, z, u)$: Frechet derivative of V, W w.r.t density u.
- KFE: $\frac{du_t(x)}{dt} := \mu_u(x, z, u) = \delta(l(x) - u(x)) - \rho u(x) \int_{\mathcal{Y}} \alpha(x, y, z, u_t)u(y)dy$.
- High-dimensional PDEs with $D_u V, D_u W$: hard to solve with conventional methods.

Discrete Choice for Matching in DeepSAM

- In the original model,

$$\alpha_t(x,y) = 1, \text{ iff } W_t(x,y) \geq V_t(x) \text{ and } W_t(y,x) \geq V_t(y)$$

- In continuous time, discrete choice of $\alpha \Rightarrow$ jumps of $V(x,u), W(x,y,u)$ at some u.
- To ensure master equation well defined & NN algorithm works, we approximate with

$$\alpha_t(x,y) = \mathbb{P}(x \text{ accepts})\mathbb{P}(y \text{ accepts}) = \left(\frac{1}{1+\left(\frac{V_t(x)}{W_t(x,y)}\right)^{\xi}}\right)\left(\frac{1}{1+\left(\frac{V_t(y)}{W_t(y,x)}\right)^{\xi}}\right)$$

- Interpretation: logit choice model with utility shocks $\sim$ extreme value distribution. $\xi \to \infty \Rightarrow$ discrete values of α.
- After solving V, W, we can compute α with discrete value definition.

Finite type approximation

- Approximate $u(x)$ on finite types: $x \in \mathcal{X} = \{x_1, \dots, x_n\}$. $u_i = u(x_i)$ is mass at x_i.
- Finite state approximation $\Rightarrow$ analytical (approximate) KFE.
- Master equation of unmatched agent:

$$0 = \mathcal{L}^V V = -rV(x,z,u) + rb + \rho \frac{1}{n} \sum_{j=1}^{n} \alpha(x, y_j, z, u)(W_t(x, y_j, z, u) - V(x,z,u))u_t(y_j)$$

$$+ \sum_{\tilde{z} \neq z} \lambda_{z\tilde{z}}(V(x,\tilde{z},u) - V(x,z,u)) + \sum_{i=1}^{n} \partial_{u_i} V(x,z,u)\mu_i^u(x,z,u).$$

- Master equation of matched agent:

$$0 = \mathcal{L}^W W = -rW(x,y,z,u) + rzf(x,y) - \delta(W(x,y,z,u) - V(x,z,u))$$

$$+ \sum_{\tilde{z} \neq z} \lambda_{z\tilde{z}}(W(x,y,\tilde{z},u) - W(x,y,z,u)) + \sum_{i=1}^{n} \partial_{u_i} W(x,y,z,u)\mu_i^u(x,z,u).$$

- KFE: $\mu_i^u(x,z,u) := \delta(l_i - u_i) - \rho u_i \frac{1}{n} \sum_{j=0}^{n} \alpha_t(x_i, y_j, z, u)u_j$.

DeepSAM algorithm

1. Approximate by NN: $V(x, z, u) \approx \widehat{V}(x, z, u; \theta_V)$, $W(x, y, z, u) \approx \widehat{W}(x, y, z, u; \theta_W)$.
2. Start with initial parameter guess $\theta^0 = (\theta_V^0, \theta_W^0)$. At iteration n with θ^n:

   2.1 Generate M sample points, $S^n = \{(x_m, y_m, z_m, u_m)\}_{m \leq M}$ to evaluate loss function.

   2.2 Calculate the weighted average mean squared error of master equations:

$$L(\theta^n, S^n) = \kappa^V L^V(\theta^n, S^n) + \kappa^W L^W(\theta^n, S^n)$$

$$L^V(\theta^n, S^n) := \frac{1}{M} \sum_{m \leq M} \left| \mathcal{L}^V V(x_m, z_m, u_m) \right|^2, L^W(\theta^n, S^n) := \frac{1}{M} \sum_{m \leq M} \left| \mathcal{L}^W W(x_m, y_m, z_m, u_m) \right|^2.$$

   2.3 Update NN parameters with stochastic gradient descent method:

$$\theta^{n+1} = \theta^n - \alpha_n D_\theta L(\theta^n, S^n)$$

   2.4 Repeat until $L(\theta^n, S^n) \leq \epsilon$ with precision threshold ϵ.

Numerical performance: Accuracy I

Calibration

- Mean squared loss as a function of type in the master equations of V and W.

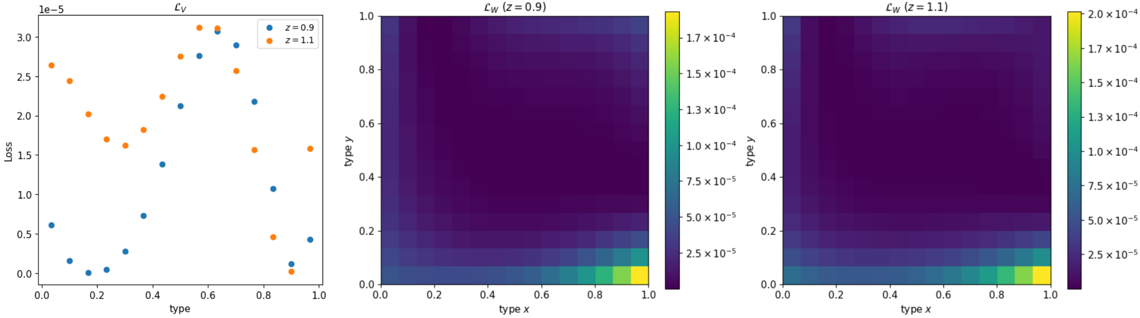


Figure: Mean squared loss as a function of type. Left: loss of master equation of V. Right: W.

Numerical performance: Accuracy II Calibration

- Compare to steady state solution in Smith (2006).

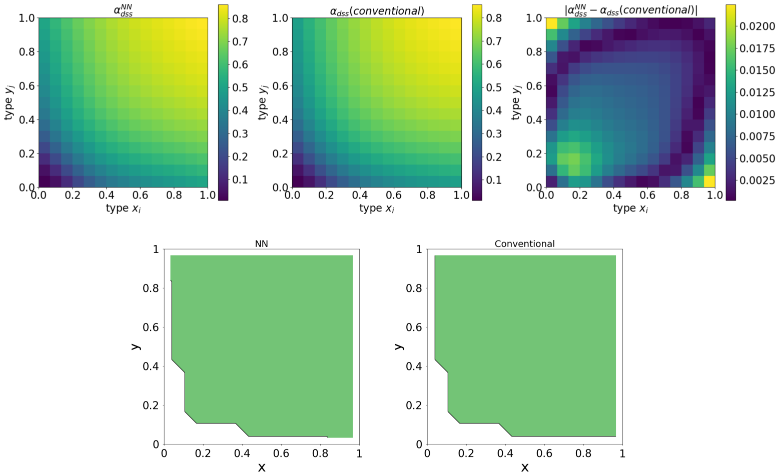
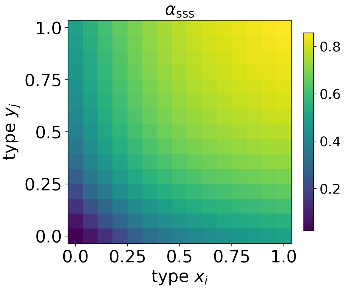


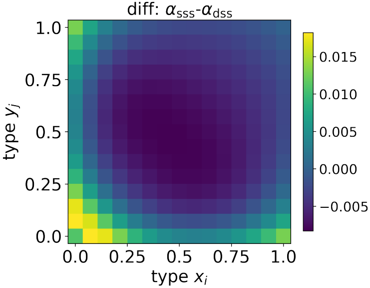
Figure: Comparison with steady-state solution: continuous and discrete α.

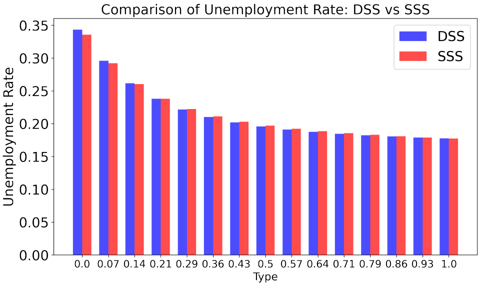
Experiments

1. Compare the stochastic steady state (SSS) to the deterministic steady state (DSS).
   - DSS: calculated from the steady state solution in the model without aggregate shocks.
   - SSS: calculated by simulating a path of aggregate shocks and then computing the long-run empirical distribution.
2. Welfare evaluation outside the steady state.
3. Compare outcomes for different groups following an aggregate productivity shock.

SSS vs DSS: Lower Types More Likely to Match

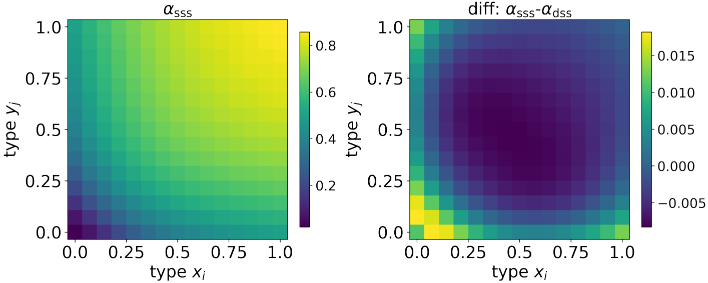


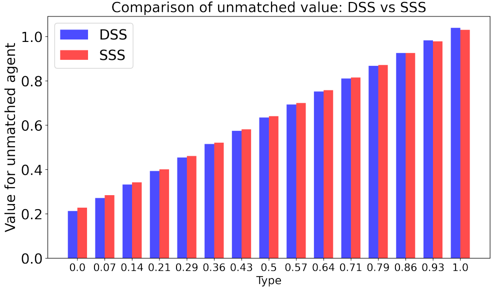




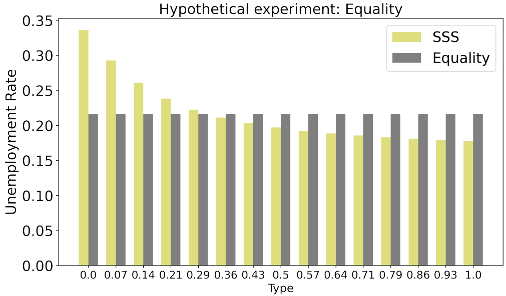
- Introduction of aggregate risk increases the matching rate for low types.
- Why? Joining a match gives agents an option value that they can benefit straight away when z increases.

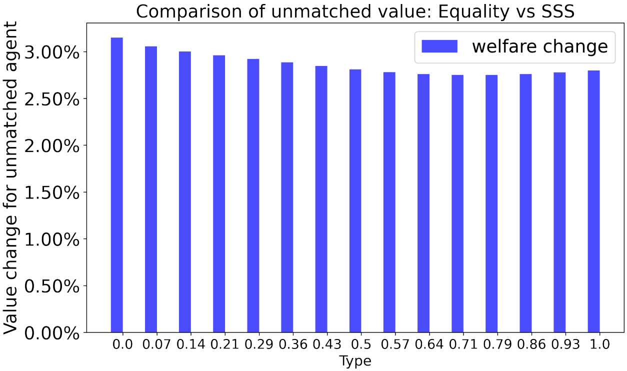
SSS vs DSS: Lower Types Gain From Aggregate Shock





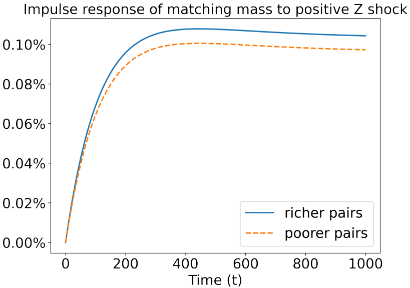
How heterogeneity matters? Welfare under different inequality

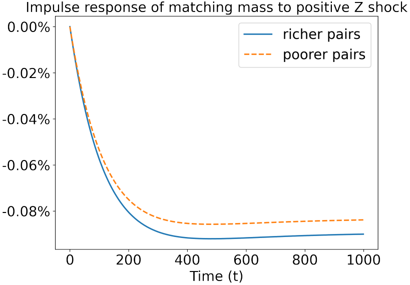




Unemployment rate becomes equal across types $\Rightarrow$ All unmatched agents' welfare increases.

Impulse response of matching to permanent productivity shock





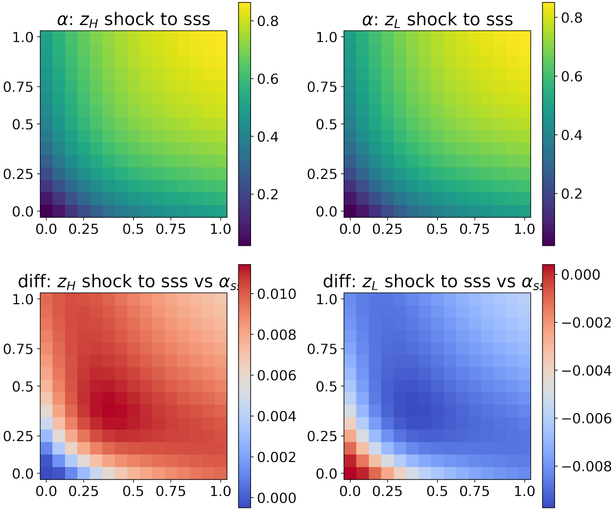


Figure: Long run matching results

Table of Contents

Shimer-Smith Model with Two-sided Heterogeneity

- Same model as before but with three changes (as in Hagedorn et al. (2017)).
- *Change 1:* Asymmetry between two sides of market:
  - Workers $x \in [0,1]$: can be employed (value $V_t^e(x,y)$) or unemployed (value $V_t^u(x)$), and
  - Firms $y \in [0,1]$: can be producing (value $V_t^p(x,y)$) or vacant (value $V_t^v(y)$).
- *Change 2:* Random matching via function $m(U,V)$, where:
  - U = aggregate mass of unemployed workers
  - V = aggregate mass of vacant firms
- *Change 3:* Total production in a match is $zf(x,y)$. Division by Nash bargaining:
  - Surplus from a match $S_t(x,y) := V_t^p(x,y) - V_t^v(y) + V_t^e(x,y) - V_t^u(x)$
  - Workers get fraction β of surplus; firms get $1-\beta$.
  - Acceptance: $\alpha_t(x,y) = 1,$ iff $S_t(x,y) > 0$, approximation $\alpha_t(x,y) = (1 + e^{-\xi S(x,y,z,g^m)})^{-1}$

Recursive equilibrium

- State variable: x, y, z, g^m. Note: $g^m(x, y)$ is "density" of matches (x, y).
- Can characterize equilibrium with the master equation for the surplus:

$$
\begin{aligned}
0 = \mathcal{L}^S S = & -\rho S(x,y,z,g^m) + zf(x,y) - \delta S(x,y,z,g^m) \\
& -(1-\beta)\frac{m(U,V)}{U(z,g^m)V(z,g^m)}\int \alpha(\tilde{x},y,z,g^m)S(\tilde{x},y,z,g^m)g^u(\tilde{x})d\tilde{x} \\
& -b-\beta\frac{m(U,V)}{U(z,g^m)V(z,g^m)}\int \alpha(x,\tilde{y},z,g^m)S(x,\tilde{y},z,g^m)g^v(\tilde{y})d\tilde{y} \\
& +\lambda(z)(S(x,y,\tilde{z},g^m) - S(x,y,z,g^m)) + D_{g^m}S(x,y,z,g^m)\cdot\mu^g(x,y,z,g^m)
\end{aligned}
$$

- KFE:

$$
\mu^g(x,y,z,g^m) = -\delta g^m(x,y) + \frac{m(U(z,g^m),V(z,g^m))}{U(z,g^m)V(z,g^m)}\alpha(x,y,z,g^m)g^v(y)g^u(x)
$$

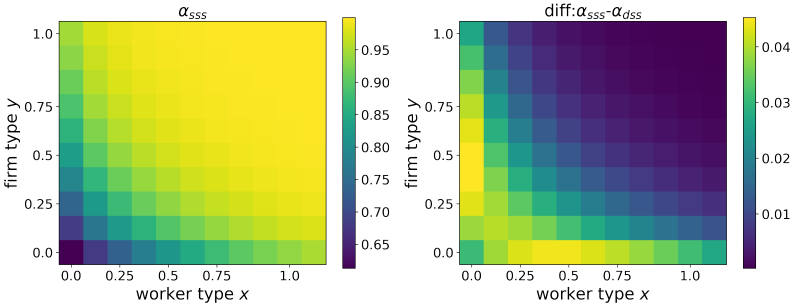
$$
\alpha(x,y,z,g^m) = (1+e^{-\xi S(x,y,z,g^m)})^{-1}
$$

DeepSAM algorithm

- Approximate the surplus by a neural network $S(x, y, z, g) \approx \widehat{S}(x, y, z, g; \theta)$:
  1. Make initial guess for the surplus.
  2. At iteration n with guess θ^n, (a) generate sample points, (b) calculate master equation loss of surplus on the sample, (c) update NN parameters using SGD method.
  3. Repeat until loss is less than ϵ.
- Once S and α have been solved, we can then solve for worker and firm value functions by solving the master equations for them.

SSS vs DSS: Lower Types More Likely to Match

Calibration



- Introduction of aggregate risk increases the matching rate for low types.
- Why? Joining a match gives agents an option value that they can benefit straight away when z increases.

Wage $w(x, y)$ at SSS: redistribution due to aggregate shocks

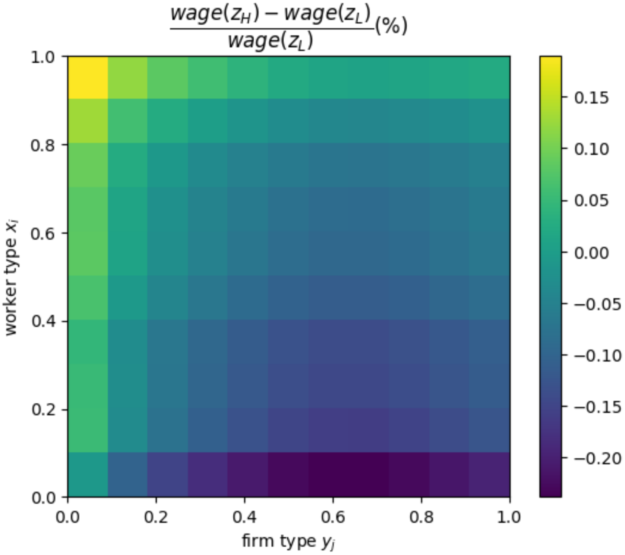


Table of Contents

Conclusion and Future Work

- We develop DeepSAM, a global solution method to SAM models with aggregate shocks.
- Aggregate risk and heterogeneity matter for matching and welfare.
- Next step: quantitative model for assortative matching with business cycle.
- Future work: apply and extend DeepSAM to
  1. Models with large firm size.
  2. Dynamic spatial model.
  3. ...

Thank You!

Calibration for Shimer-Smith model with NTU

Parameter	Interpretation	Value	Target/Source
ρ	Meeting rate	3.0	
r	Discount rate	0.3	
δ	Job destruction rate	0.1	
ξ	Extreme value distribution for α choice	3.0	
$f(x, y)$	Payoff for x in match (x, y)	$xy + x + y$	bilinear
b	Worker unemployment benefit	0.0	
$z, \tilde{z}$	Poisson shocks	0.9, 1.1	
$\lambda_z, \lambda_{\tilde{z}}$	Poisson transition probability	0.4, 0.4	
n	Discretization of types	15	

back

Calibration for Hagedorn et al. model with aggregate risk

Parameter	Interpretation	Value	Target/Source
ρ	Discount rate	0.41	
δ	Job destruction rate	0.01	
ξ	Extreme value distribution for α choice	3.0	
$f(x, y)$	Production function for match (x, y)	$0.6 + 0.4(\sqrt{x} + \sqrt{y})^2$	
β	Surplus division factor	0.5	
$z, \tilde{z}$	Poisson shocks	0.9, 1.1	
$\lambda_z, \lambda_{\tilde{z}}$	Poisson transition probability	0.4, 0.4	
ν	Elasticity parameter for meeting function	0.5	
κ	Scale parameter for meeting function	0.4	
b	Worker unemployment benefit	0.5	
n_x	Discretization of worker types	10	
n_y	Discretization of firm types	11	

back

Recursive equilibrium I

- Master equation for an unemployed worker:

$$0 = -\rho V^u(x,z,g^m) + b + \beta \frac{\mathcal{M}^u(z,g^m)}{V(z,g^m)} \int \alpha(x,\tilde{y},z,g^m) S(x,\tilde{y},z,g^m) g^v(\tilde{y}) d\tilde{y} + D_{g^m} V^u(x,y,z,g^m) \cdot \mu^g$$

- Master equation for an employed worker:

$$0 = -\rho V^e(x,y,z,g^m) + w(x,\tilde{y},z,g^m) - \beta\delta S(x,\tilde{y},z,g^m) + D_{g^m} V^e(x,y,z,g^m) \cdot \mu^g$$

- Master equation for a vacant firm is:

$$0 = -\rho V^v(y,z,g^m) - (1-\beta) \frac{\mathcal{M}^v(z,g^m)}{U(z,g^m)} \int \alpha(\tilde{x},y,z,g^m) S(\tilde{x},y,z,g^m) g^u(\tilde{x}) d\tilde{x} + D_{g^m} V^v(y,z,g^m) \cdot \mu^g$$

- Master equation for a producing firm becomes:

$$0 = -\rho V^p(x,y,z,g^m) + zf(x,y) - w(x,\tilde{y},z,g^m) - \delta(1-\beta) S(x,\tilde{y},z,g^m) + D_{g^m} V^p(x,y,z,g^m) \cdot \mu^g$$

back

Finite type approximation

- Approximate with finite collection of types: $x \in \mathcal{X} = \{x_1, \ldots, x_{n_x}\}$ and $y \in \mathcal{Y} = \{y_1, \ldots, y_{n_y}\}$.

- Master equation for surplus:

$$
\begin{aligned}
0 = & -(\rho+\delta)S(x,y,z,g^m) + zf(x,y) - b \\
& -(1-\beta)\frac{m(U(z,g^m),V(z,g^m))}{U(z,g^m)V(z,g^m)}\frac{1}{n_x}\sum_{i=1}^{n_x}\alpha(x_i,y,z,g^m)S(x_i,y,z,g^m)g^u(x_i) \\
& -\beta\frac{m(U(z,g^m),V(z,g^m))}{U(z,g^m)V(z,g^m)}\frac{1}{n_y}\sum_{j=1}^{n_y}\alpha(x,y_j,z,g^m)S(x,y_j,z,g^m)g^v(y_j) \\
& +\lambda(z)(S(x,y_j,\tilde{z},g^m) - S(x,y_j,z,g^m)) + \sum_{i=1}^{n_x}\sum_{i=1}^{n_y}\partial_{g_{ij}^m}S(x,y,z,g^m)\mu^g(x_i,y_j,z,g^m)
\end{aligned}
$$

back

Approximate Discrete Choice in DeepSAM: Worker and Firm

- To improve the performance of NN algorithm, we approximate

$$\alpha(x, y, z, g^m) := \begin{cases} 1, & \text{if } S(x, y, z, g^m) > 0 \\ 0, & \text{otherwise} \end{cases}$$

  which we approximate by:

$$\alpha(x, y, z, g^m) = \frac{1}{1 + e^{-\xi S(x,y,z,g^m)}}$$

- Interpretation: logit choice model with utility shocks $\sim$ extreme value distribution. $\xi \to 0 \Rightarrow$ discrete values of α.

- After solving V, W, we compute α with discrete value definition.

back