

Simulation allows us to compute complex integrals that arise in structural estimation
Why do we have complex integrals in estimation?

Simulation allows us to	compute complex	integrals that a	arise in	structural	estimation

Why do we have complex integrals in estimation?

Integration over distributions of unobservables (e.g. preference heterogeneity)

Simulation allows us to compute complex integrals that arise in structural estimation

Why do we have complex integrals in estimation?

Integration over distributions of unobservables (e.g. preference heterogeneity)

• cf. PS4: simulation can approximate integrals involving mixture distributions

Simulation allows us to compute complex integrals that arise in structural estimation

Why do we have complex integrals in estimation?

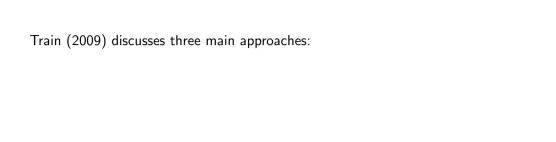
Integration over distributions of unobservables (e.g. preference heterogeneity)

- cf. PS4: simulation can approximate integrals involving mixture distributions
- Alternative approach: quadrature (works only for low-dimensional integrals)

Simulation allows us to compute complex integrals that arise in structural estimation Why do we have complex integrals in estimation?

Integration over distributions of unobservables (e.g. preference heterogeneity)

- cf. PS4: simulation can approximate integrals involving mixture distributions
- Alternative approach: quadrature (works only for low-dimensional integrals)
- For otherwise-intractable models, simulation may be the only option



Train (2009) discusses three main approaches:
1. Simulated Maximum Likelihood (SML, sometimes "MSL" or "SMLE")

Train (2009)	discusses	three	main	approaches:	

2. Simulated Method of Moments (SMM, sometimes "MSM")

1. Simulated Maximum Likelihood (SML, sometimes "MSL" or "SMLE")

Train (2	2009)	discusses	three	main	approaches:
----------	-------	-----------	-------	------	-------------

1. Simulated Maximum Likelihood (SML, sometimes "MSL" or "SMLE")

2. Simulated Method of Moments (SMM, sometimes "MSM")

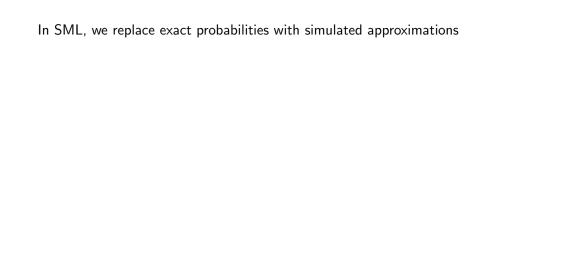
3. Method of Simulated Scores (MSS)

Train (2009) discusses three main approaches:

- 1. Simulated Maximum Likelihood (SML, sometimes "MSL" or "SMLE")
- 2. Simulated Method of Moments (SMM, sometimes "MSM")
  - 3. Method of Simulated Scores (MSS)

PS4 implemented SML for mixed logit estimation

This series: focus primarily on SMM



In SML, we replace exact probabilities with simulated approximations Log-likelihood function:

th: 
$$\ell\left(\theta\right) = \sum_{i} \log \left\{ \int \prod_{j} \left[P_{ij}\left(\theta,z\right)\right]^{d_{ij}} dF\left(z\right) \right\}$$

In SML, we replace exact probabilities with simulated approximations Log-likelihood function:

$$\ell(\theta) = \sum_{i} \log \left\{ \int \prod_{j} \left[ P_{ij} \left( \theta, z \right) \right]^{d_{ij}} dF(z) \right\}$$

 $P_i(\theta)$ 

Simulated log-likelihood:

$$\ell\left( heta
ight) = \sum_{i}\log\left\{rac{1}{R}\sum_{r=1}^{R}\prod_{i}\left[P_{ij}\left( heta,z^{r}
ight)
ight]^{d_{ij}}
ight\}$$

In SML, we replace exact probabilities with simulated approximations

Log-likelihood function:

$$\ell(\theta) = \sum_{i} \log \left\{ \int \prod_{j} \left[ P_{ij} \left( \theta, z \right) \right]^{d_{ij}} dF(z) \right\}$$

 $\check{P}_i(\theta)$ 

Log-likelihood function:

$$\ell\left( heta
ight) = \sum_{i} \log \left\{ \frac{1}{R} \sum_{r=1}^{R} \prod_{j} \left[P_{ij}\left( heta, z^{r}
ight)\right]^{d_{ij}} 
ight\}$$

In SML, we replace exact probabilities with simulated approximations

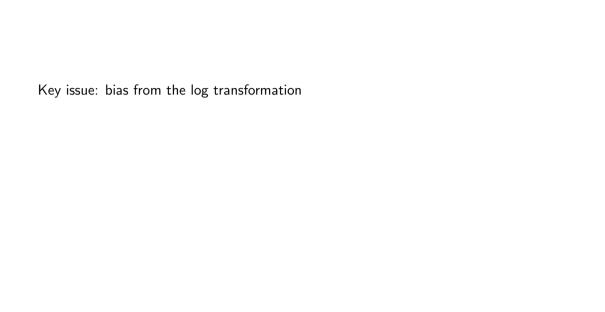
Log-likelihood function:

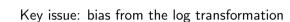
$$\ell(\theta) = \sum_{i} \log \left\{ \int \prod_{j} \left[ P_{ij}(\theta, z) \right]^{d_{ij}} dF(z) \right\}$$

Log-likelihood function:

$$\ell(\theta) = \sum_{i} \log \underbrace{\left\{ \frac{1}{R} \sum_{r=1}^{R} \prod_{j} \left[ P_{ij} \left( \theta, z^{r} \right) \right]^{d_{ij}} \right\}}_{\check{P}_{i}(\theta)}$$

where  $\check{P}_i(\theta)$  is the simulated probability based on R draws  $z^r \sim F(z)$ 





• Even if  $\check{P}_i(\theta)$  is unbiased for  $P_i(\theta)$ ,  $\log \check{P}_i(\theta)$  is biased for  $\log P_i(\theta)$ 

Key issue: bias from the log transformation

• This simulation bias affects the SML estimator

• Even if 
$$\check{P}_i(\theta)$$
 is unbiased for  $P_i(\theta)$ ,  $\log \check{P}_i(\theta)$  is biased for  $\log P_i(\theta)$ 

Key issue: bias from the log transformation

• Even if 
$$\check{P}_i(\theta)$$
 is unbiased for  $P_i(\theta)$ ,  $\log \check{P}_i(\theta)$  is biased for  $\log P_i(\theta)$ 

This simulation bias affects the SML estimator

• Properties depend on relationship between R (no. of draws) and N (sample size)

Three cases based on how	N R grows with N:		

• If R is fixed: SML is inconsistent

• If *R* is fixed: SML is inconsistent

• If R rises with N at any rate: SML is consistent

- If *R* is fixed: SML is inconsistent
- If R rises with N at any rate: SML is consistent
- If R rises faster than  $\sqrt{N}$ : SML is asymptotically efficient

- If R is fixed: SML is inconsistent
- If R rises with N at any rate: SML is consistent
- If R rises faster than  $\sqrt{N}$ : SML is asymptotically efficient

Practical implication: increase R as sample size grows

Simulated Method of Moments (SMM) avoids the log bias problem	
Match moments from data to moments from simulated model	

Match moments from data to moments from simulated model

• General approach: find  $\theta$  such that  $g(\theta) \approx 0$ 

- Match moments from data to moments from simulated model
- General approach: find  $\theta$  such that  $g(\theta) \approx 0$
- ullet g( heta) compares empirical moments to model-implied moments

- Match moments from data to moments from simulated model
- General approach: find  $\theta$  such that  $g(\theta) \approx 0$
- ullet g( heta) compares empirical moments to model-implied moments
- No log transformation needed, so simulation bias is less severe

Classical moment condition:

$$g(\theta) = \frac{1}{N} \sum_{i} \left[ m_i - m(\theta) \right]$$

Simulated moment condition:

$$g(\theta) = \frac{1}{N} \sum_{i} \left[ m_i - \check{m}(\theta) \right]$$

Simulated moment condition:

$$g(\theta) = rac{1}{N} \sum_{i} \left[ m_i - \check{m}(\theta) 
ight]$$

where  $\check{m}(\theta) = \frac{1}{R} \sum_{r=1}^{R} m(\theta, z^r)$  is the simulated moment

·

Simulated moment condition: 
$$a(0) = \frac{1}{2} \sum_{i=1}^{n} a_i a_i$$

 $g( heta) = rac{1}{\mathsf{N}} \sum_i \left[ m_i - reve{m}( heta) 
ight]$ 

where 
$$\check{m}(\theta) = \frac{1}{R} \sum_{r=1}^{R} m(\theta, z^r)$$
 is the simulated moment

where  $m(\sigma) = R \angle r = 1 m(\sigma, 2)$  is the simulated in

Minimize objective:  $J(\theta) = Ng(\theta)'Vg(\theta)$ 

GMM for discrete choice models:

For each observation i and alternative j, define the moment:

 $g_{ij}(\theta) = d_{ij} - P_{ij}(\theta)$ 

GMM for discrete choice models:

For each observation i and alternative j, define the moment:

$$g_{ij}(\theta) = d_{ij} - P_{ij}(\theta)$$

Stack into  $NJ \times 1$  vector

GMM for discrete choice models:

For each observation i and alternative j, define the moment:

$$g_{ii}(\theta) = d_{ii} - P_{ii}(\theta)$$

Stack into  $NJ \times 1$  vector

$$J(\theta) = g(\theta)'Wg(\theta)$$
 becomes NLLS when  $W = \mathbf{I}_{NJ}$ 

$$g(\theta) = \frac{1}{N} \sum_{i} \sum_{j} \left[ d_{ij} - \check{P}_{ij}(\theta) \right]$$

$$g(\theta) = \frac{1}{N} \sum_{i} \sum_{j} \left[ d_{ij} - \check{P}_{ij}(\theta) \right]$$

Key advantage over SML:

$$g(\theta) = \frac{1}{N} \sum_{i} \sum_{i} \left[ d_{ij} - \check{P}_{ij}(\theta) \right]$$

Key advantage over SML:

• Probabilities enter linearly (no log transformation)

$$g( heta) = rac{1}{N} \sum_{i} \sum_{i} \left[ d_{ij} - \check{P}_{ij}( heta) 
ight]$$

Key advantage over SML:

- Probabilities enter linearly (no log transformation)
- ullet No simulation bias if  $\check{P}_{ij}( heta)$  is unbiased

$$g(\theta) = \frac{1}{N} \sum_{i} \sum_{j} \left[ d_{ij} - \check{P}_{ij}(\theta) \right]$$

Key advantage over SML:

- Probabilities enter linearly (no log transformation)
- No simulation bias if  $\check{P}_{ii}(\theta)$  is unbiased
- ullet Consistent even with fixed R (though not as efficient as SML with  $R o \infty$ )

Method of Simulated Scores (MSS) can simulate the score in different ways

• Score function:  $s(\theta) = \frac{\partial \log \ell(\theta)}{\partial \theta}$ 

- Score function:  $s(\theta) = \frac{\partial \log \ell(\theta)}{\partial \theta}$
- Key distinction: can use unbiased score simulators (not just  $\frac{\partial \log \check{P}(\theta)}{\partial \theta}$ )

• Score function: 
$$s(\theta) = \frac{\partial \log \ell(\theta)}{\partial \theta}$$

- Key distinction: can use unbiased score simulators (not just  $\frac{\partial \log \check{P}(\theta)}{\partial \theta}$ )
- With unbiased simulators: consistent even for fixed R

• Score function: 
$$s(\theta) = \frac{\partial \log \ell(\theta)}{\partial \theta}$$

- ullet Key distinction: can use unbiased score simulators (not just  $rac{\partial \log \check{P}( heta)}{\partial heta}$ )
- With unbiased simulators: consistent even for fixed R
- Efficient when R rises at any rate with N (better than MSL)

• Score function: 
$$s(\theta) = \frac{\partial \log \ell(\theta)}{\partial \theta}$$

- Key distinction: can use unbiased score simulators (not just  $\frac{\partial \log \check{P}(\theta)}{\partial \theta}$ )
- With unbiased simulators: consistent even for fixed R
- Efficient when *R* rises at any rate with *N* (better than MSL)
- Practical limitation: constructing unbiased score simulators is difficult

• Score function: 
$$s(\theta) = \frac{\partial \log \ell(\theta)}{\partial \theta}$$

- Key distinction: can use unbiased score simulators (not just  $\frac{\partial \log P(\theta)}{\partial \theta}$ )
- With unbiased simulators: consistent even for fixed R
- Efficient when R rises at any rate with N (better than MSL)
- Practical limitation: constructing unbiased score simulators is difficult
- Rarely used due to implementation challenges