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

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 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:

Train (2009) discusses three main approaches:

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

Train (2009) discusses three main approaches:

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

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

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)

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

In SML, we replace exact probabilities with simulated approximations

Log-likelihood function:

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

In SML, we replace exact probabilities with simulated approximations

Log-likelihood function:

$$\ell(\theta) = \sum_i \log \underbrace{\left\{ \int \prod_j \left[P_{ij}(\theta, z) \right]^{d_{ij}} dF(z) \right\}}_{P_i(\theta)}$$

Simulated log-likelihood:

$$\ell(\theta) = \sum_i \log \left\{ \frac{1}{R} \sum_{r=1}^{R} \prod_j \left[P_{ij}(\theta, z^r) \right]^{d_{ij}} \right\}$$

In SML, we replace exact probabilities with simulated approximations

Log-likelihood function:

$$\ell(\theta) = \sum_i \log \underbrace{\left\{ \int \prod_j \left[P_{ij}(\theta, z) \right]^{d_{ij}} dF(z) \right\}}_{P_i(\theta)}$$

Log-likelihood function:

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

In SML, we replace exact probabilities with simulated approximations

Log-likelihood function:

$$\ell(\theta) = \sum_i \log \underbrace{\left\{ \int \prod_j \left[P_{ij}(\theta, z) \right]^{d_{ij}} dF(z) \right\}}_{P_i(\theta)}$$

Log-likelihood function:

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

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

Key issue: bias from the log transformation

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 $\log P_i(\theta)$

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

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 R grows with N:

Three cases based on how R grows with N:

- If R is fixed: SML is inconsistent

Three cases based on how R grows with N:

- If R is fixed: SML is inconsistent

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

Three cases based on how R grows with N:

- 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

Three cases based on how R grows with N:

- 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

Simulated Method of Moments (SMM) avoids the log bias problem

- Match moments from data to moments from simulated model

Simulated Method of Moments (SMM) avoids the log bias problem

- Match moments from data to moments from simulated model

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

Simulated Method of Moments (SMM) avoids the log bias problem

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

Simulated Method of Moments (SMM) avoids the log bias problem

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

SMM framework parallels classical GMM

SMM framework parallels classical GMM

Classical moment condition:

$$g(\theta) = \frac{1}{N} \sum_i [m_i - m(\theta)]$$

SMM framework parallels classical GMM

Simulated moment condition:

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

SMM framework parallels classical GMM

Simulated moment condition:

$$g(\theta) = \frac{1}{N} \sum_i [m_i - \check{m}(\theta)]$$

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

SMM framework parallels classical GMM

Simulated moment condition:

$$g(\theta) = \frac{1}{N}\sum_i [m_i - \check{m}(\theta)]$$

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

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

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:

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

Stack into $NJ \times 1$ vector

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

For SMM, we replace exact probabilities with simulated ones:

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

For SMM, we replace exact probabilities with simulated ones:

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

Key advantage over SML:

For SMM, we replace exact probabilities with simulated ones:

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

Key advantage over SML:

- Probabilities enter linearly (no log transformation)

For SMM, we replace exact probabilities with simulated ones:

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

Key advantage over SML:

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

For SMM, we replace exact probabilities with simulated ones:

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

Key advantage over SML:

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

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

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

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

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

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

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

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

- 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

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

- 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)

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

- 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

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

- 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
- Rarely used due to implementation challenges