Simulated Method of Moments

Simulated Method of Moments

- SMM is a simulated version of GMM

Simulated Method of Moments

- SMM is a simulated version of GMM
- Uses moments from simulated data instead of analytical moments

Simulated Method of Moments

- SMM is a simulated version of GMM
- Uses moments from simulated data instead of analytical moments
- Objective: make simulated and actual data match

Simulated Method of Moments

- SMM is a simulated version of GMM
- Uses moments from simulated data instead of analytical moments
- Objective: make simulated and actual data match
- See McFadden (1989) and Evans (2018) for details

Pros of SMM

Pros of SMM

- Can estimate models with P's that don't have closed form (e.g. probit)

Pros of SMM

- Can estimate models with P's that don't have closed form (e.g. probit)

- Can handle otherwise intractable models (dynamic models, high-dim integrals)

Pros of SMM

- Can estimate models with P's that don't have closed form (e.g. probit)
- Can handle otherwise intractable models (dynamic models, high-dim integrals)
- Can estimate micro-models using only aggregated data

Pros of SMM

- Can estimate models with P's that don't have closed form (e.g. probit)
- Can handle otherwise intractable models (dynamic models, high-dim integrals)
- Can estimate micro-models using only aggregated data
- Simulation code already done—easy to move to counterfactuals

Pros of SMM

- Can estimate models with P's that don't have closed form (e.g. probit)
- Can handle otherwise intractable models (dynamic models, high-dim integrals)
- Can estimate micro-models using only aggregated data
- Simulation code already done—easy to move to counterfactuals
- Straightforward to interpret which moments the model is fitting

Pros of SMM

- Can estimate models with P's that don't have closed form (e.g. probit)
- Can handle otherwise intractable models (dynamic models, high-dim integrals)
- Can estimate micro-models using only aggregated data
- Simulation code already done—easy to move to counterfactuals
- Straightforward to interpret which moments the model is fitting
- Easier to compare with reduced-form evidence

Cons of SMM

Cons of SMM

- Much more computationally intensive than GMM

Cons of SMM

- Much more computationally intensive than GMM
- Must fully specify model (including error distribution)

Cons of SMM

- Much more computationally intensive than GMM
- Must fully specify model (including error distribution)
- Loss of statistical efficiency relative to MLE (larger standard errors)

Cons of SMM

- Much more computationally intensive than GMM
- Must fully specify model (including error distribution)
- Loss of statistical efficiency relative to MLE (larger standard errors)
- Selection of moments can feel ad hoc

SMM Example: Linear Regression

Consider a simple linear regression model:

$$
\begin{aligned}
y &= X\beta + \varepsilon, \\
\varepsilon &\sim N(0, \sigma^2)
\end{aligned}
$$

SMM Example: Linear Regression

Consider a simple linear regression model:

$$
\begin{aligned}
y &= X\beta + \varepsilon, \\
\varepsilon &\sim N(0, \sigma^2)
\end{aligned}
$$

y and X are data; we want to estimate β and σ

As mentioned earlier, we must make a strong assumption about DGP: $\varepsilon \sim N(0, \sigma^2)$

SMM Estimation Steps

For each guess of $\theta = [\beta', \sigma]'$:

SMM Estimation Steps

For each guess of $\theta = [\beta', \sigma]'$:

- Compute data moments

SMM Estimation Steps

For each guess of $\theta = [\beta', \sigma]'$:

- Compute data moments

- Draw *N* ε's *D* times (typically $D > 1000$)

SMM Estimation Steps

For each guess of $\theta = [\beta', \sigma]'$:

- Compute data moments

- Draw N ε's D times (typically $D > 1000$)

- For each draw, compute $\tilde{y} = X\beta + \varepsilon$

SMM Estimation Steps

For each guess of $\theta = [\beta', \sigma]'$:

- Compute data moments
- Draw N ε's D times (typically $D > 1000$)
- For each draw, compute $\tilde{y} = X\beta + \varepsilon$
- Compute model moments using $\tilde{y}$ (same structure as data moments)

SMM Estimation Steps

For each guess of $\theta = [\beta', \sigma]'$:

- Compute data moments
- Draw N ε's D times (typically $D > 1000$)
- For each draw, compute $\tilde{y} = X\beta + \varepsilon$
- Compute model moments using $\tilde{y}$ (same structure as data moments)
- Average model moments across all D draws

SMM Estimation Steps

For each guess of $\theta = [\beta', \sigma]'$:

- Compute data moments
- Draw N ε's D times (typically $D > 1000$)
- For each draw, compute $\tilde{y} = X\beta + \varepsilon$
- Compute model moments using $\tilde{y}$ (same structure as data moments)
- Average model moments across all D draws
- Update objective function: minimize distance between data and model moments

Moments to Match

Data moments: $\left\{ y_i, i = 1, \ldots, N; \widehat{V}(y) \right\}$

Moments to Match

Data moments: $\left\{y_i, i = 1, \ldots, N; \widehat{V}(y)\right\}$

Model moments: $\left\{\tilde{y}_i, i = 1, \ldots, N; \widehat{V}(\tilde{y})\right\}$

SMM Optimization

SMM Optimization

- Can use any optimizer for the SMM objective function

SMM Optimization

- Can use any optimizer for the SMM objective function
- SMM objective may have local optima (poorly behaved)

SMM Optimization

- Can use any optimizer for the SMM objective function
- SMM objective may have local optima (poorly behaved)
- Tactics for finding global optimum:
  - Use L-BFGS from many starting values
  - Use Simulated Annealing or Particle Swarm

SMM Optimization

- Can use any optimizer for the SMM objective function
- SMM objective may have local optima (poorly behaved)
- Tactics for finding global optimum:
  - Use L-BFGS from many starting values
  - Use Simulated Annealing or Particle Swarm
- Simple problems (like OLS) should be well-behaved

SMM Optimization

- Can use any optimizer for the SMM objective function
- SMM objective may have local optima (poorly behaved)
- Tactics for finding global optimum:
  - Use L-BFGS from many starting values
  - Use Simulated Annealing or Particle Swarm
- Simple problems (like OLS) should be well-behaved
- Be sure to use same draw of ε in every iteration!