Structural modeling process

Gabriel Petrini

September 1st, 2020

Contents

1 Pre-class reading: Practical Issues in Structural Estimation (Keane YouTube talk, 2015)

1.1 Structural Model Development

Structural estimation has several key stages

1.1.1 Theoretical model development

Assume you want to build a structural model to address an **economic or policy question of interest**. A good starting point is to ask what elements your model must have to credibly address the question:

- The model cannot be so simple (or stylized) that it essentially imposes answers *a priori*
  - You must include all the mechanisms that generates same relevant pattern in order to not rig the results to answer the question of interest
- It must be plausible to assume parameters of the model are **invariant** to policy experiments that you plan to do
  - You should plan what experiments will be employed with this model in the very beginning

```
Example:  "How important is the moral hazard effect of health insurance?"
```

1.1.2 Practical Specification issues

There is an inherent conflict between:

1. Model that is rich enough to credibly address the question of interest
   - As we make model richier, we tend to get more **state variables**
2. A model that is feasible to solve and estimate
   - Too many state variables makes solution infeaseble

The art of Structural Estimation is largely about how to develop "rich" models that are still feasible to solve and estimate. There are strategies to reduce the number of state variables (*e.g.* fertility and labor supply, brand inventories - assuming stationary tastes) so it is not necessarily to keep track of all states altogether.

1.1.3 Solving the model & Understanding how the model works

Let's assume you have settled on a model that you think is fit for purpose. These two steps should be done in tandem. You should be writing programs to at the same time:

- solve the model
- simulate the model
- calculate descriptive statistics

Always begin by programming a simple special case of the model - where theory makes a clear prediction of how it should behave. If your simulations do not lines up with basic theory, you have done something wrong.

- You should add mechanisms of features to the model one at time.
  - Always rig the program so if a parameter (let say, θ) is set to zero, the new mechanism is shut down. Then, make sure you get the **Exact** same simulation results as before if you set this parameter to zero
- Once you have introduced a new mechanism, manipulate the new parameters related to that mechanism to see what they do
  - Make sure the simulation results make intuitive sense
    - If they do not make sense, you probably have a bug
    - Sometimes, basic intuition turns out to be wrong and you have learned some economics
- Continuing introducing new mechanisms step-by-step until you have your full model
- By the end of this process, you will have a good understanding of how each feature of the model (and the parameters) affects behaviour
  - Spending a lot of time learning what each parameter does to behaviour also gives you a good intuitive understanding of how the model is identified and could warn about identification issues

1.1.4 Estimation

Nonlinear estimation is difficult

- You cannot just start the parameter vector at some random place and expect it to coverage
  - **Suggestion:** Finds values that fits the intercepts
- You need to **calibrate** the model to achieve a half way decent fit before iterating
  - This also teaches you a lot of how the model works
  - You should expect that this calibration process will take a long time and lots of patience
- If you find it impossible to get a decent calibration by hand, it means:
  - there is some important mechanism you have omitted from the model
  - you do not really understand the model
  - you are not trying hard enough
- **Strategy:** Presents the simplest model version. Discuss why it does not fit the data and what is needed to achieve it.
- Now that you are ready to estimate you need to write the code for the parameter search algorithm
  - **Recommendation:** BHHH and Simplex (and help the search algorithm by hand)
  - Only build up gradually to the full model as you are sure everything is working properly
- Estimation is not a mechanical process
  - As the parameter iterate, you should look
    - Simulation of key statistics vs. actual data
    - Values of parameters where there are some reasonable prior

  - Two types of things often go wrong
    * Some parameters go to strange values
    * Key moments do not improve or even get worse
    * These are often symptoms of bugs in the code of a flaw in the model (*e.g.* identification problem)
  - Problems in estimation are very frustrating because it can be hard to pin down the source
    * Bug in estimation code
    * Bug in solving the model
    * A flaw in the model
    * OBS: You almost never find the bug by reading the code. Hints:
      · Shut down parts of the model to figure out which part is causing the problem
      · Print out lots of stuff and check if it makes sense

1.1.5 Validation

Let's say the parameters of the model have converged to sensible looking values and the in sample fit looks OK

- A good opportunity for validation is when an experiment has been run, and you can estimate the model on the "control data" and see if you can forecast the "treatment data"
- You should think about whether you can validate the model **before** you estimate it
  - Most structural papers do not do much in the way of model validation. One reason is that data needed to do validation is often not easily available

1.1.6 Policy Experiments

One of the major reasons we do structural estimation is because we can use structural model to do policy experiments

- Or we may want to use the model to optimize the parameters of a policy to maximize some objective
- Experiments only allow us to see effects of policies that have already been implemented
- A common flaw of structural papers is they do lot of work to solve and estimate the model
  - But when that is done, they do not report any interesting experiments
  - You should have some interesting experiments in mind before you even start

1.2 Conclusion: Why do structural estimation

- One of the reasons is because you are interested in a model itself
- Models that we are confident in using for policy evaluation

2 Structural modeling workflow

2.1 An example model

To help fix ideas, let's revisit a commonly used model in introductory econometrics:

$$\log(w_i) = \beta_0 + \beta_1 s_i + \beta_2 x_i + \beta_3 x_i^2 + \varepsilon_i \tag{1}$$

where we have cross-sectional data and where

- i indexes individuals
- w_i is employment income
- s_i is years of schooling
- x_i is years of work experience (or, more commonly, potential work experience)
- ε_i is anything else that determines income

We want to estimate $(\beta_1, \beta_2, \beta_3)$, which are **returns to human capital investment**

2.1.1 Quick review

- It is nearly certain that (1) suffers from omitted variable bias
  - i.e. there are lots of factors in ε_i that are correlated with both s_i and w_i
- Thus, our estimates of $(\beta_1, \beta_2, \beta_3)$ will not be causal
- We could try to get causal estimates using a variety of identification strategies:
  - find a valid instrument for s_i (**angristKrueger1991**; **card1995**)
  - exploit a discontinuity in s_i (**ost_al2018**)
  - randomize s_i (**attanasio_al2011**)

2.1.2 A structural view of Equation (1)

We know that (1) will produced biased estimates, but why? Some possibilities:

- **ability bias:** s_i and w_i are both positively correlated with unobservable cognitive ability
- **comparative advantage:** multidimensional unobservable ability $\implies$ self-selection into schooling
- **credit constraints:** s_i is a costly investment; some people may not be able to borrow enough
- **preference heterogeneity:** (differing tastes for s_i, differing discount rates)

2.2 Theoretical Model Development

- Since schooling has an up-front cost and long-term benefit, need a dynamic model
  - **period 1:** decide how much schooling to get
  - **period 2:** choose whether or not to work; if working, receive income by (1)
  - individuals choose schooling level to maximize lifetime utility
- Preferences (denote utility in period t by u_t, with s, x and w defined previously)

$$u_1(z, c, \eta_1) = f(z, c, \eta_1) \tag{2}$$
$$u_2(w(s, x), k, \eta_2) = g(w(s, x), k, \eta_2) \tag{3}$$

where z is family background, c is schooling costs, k is number of kids in adult household and η_t are unobservable preferences [similar to ε in (1)]

With discount factor $\delta \in [0, 1]$, the discounted lifetime utility function is then

$$V = u_1(z, c, \eta_1) + \delta u_2(w(s, x), k, \eta_2) \tag{4}$$

- Equations (1)–(4) define our model
- A number of important questions arise (But we'll ignore these for today)
  - Where is cognitive ability? What exactly does c represent? Where are loans?
  - Maybe people should care about consumption in period 2, not income
  - Does family background really only enter u_1 and not $\log(w)$?
  - Should x in (1) be a function of s? (Lower $s \implies$ longer working life)
  - What are people's beliefs about future k when deciding s?

2.2.1 Overview of the theoretical model

Exog	**Endog**	**Outcome**	**Unobs**	**Parameters**
family background (z)	schooling (s)	labor income (w)	income (ε)	retn. human capital (β)
schooling costs (c)	period-2 work dec.		preferences (η_t)	discount factor (δ)
children in household (k)				other $f(\cdot)$ and $g(\cdot)$

2.3 Practical Specification Issues

Now that we have a model, we need to figure out how to take it to data

- This is where we apply knowledge about our data and stats/econometrics
- Key data questions:
  - Can we observe the variables of the model in our data set?
  - If so, are they reliably measured?
- Key specification questions:

  - How to model η_t and ε? (Need to make distributional assumptions)
  - Functional forms of $f(\cdot)$ and $g(\cdot)$
  - Should s be continuous (years of schooling) or discrete (college/not)?

What determines the specification is often:

- what is reliably measured in the data
- what is computationally feasible to estimate

Parameters of the model either need to be **estimated** or **calibrated**

- e.g. often we don't have reliable data to allow us to estimate δ; we must calibrate it
- Computational feasibility often governs how we specify the different functions
  - e.g. linear-in-parameters with additively separable unobservables [like (1)]

3 Example real data

Here is some real data from the most recent round of the NLSY97

```
using CSV, DataFrames, Statistics
df = CSV.read("Data/slides3data.csv"; missingstrings=["NA"])
size(df)
# outputs (6009, 12)
describe(df)
# outputs the below:
12×8 DataFrame
 Row  variable        mean      min   median   max      nunique  nmissing
      Symbol          Float64   Real  Float64  Real     Nothing  Union...

 1    id              4534.71   4     4544.0   9022
 2    female          0.52671   0     1.0      1
 3    black           0.269762  0     0.0      1
 4    latin           0.210351  0     0.0      1
 5    white           0.511067  0     1.0      1
 6    employed        0.756532  0     1.0      1
 7    wage            25.5309   8.0   20.0     150.0            933
 8    collgrad        0.350474  0     0.0      1
 9    age             34.967    33    35.0     37
 10   parent_college  0.238975  0     0.0      1
 11   numkids         1.32684   0     1.0      9
 12   efc             4.2243    0.0   0.77763  118.111
```

- We have demographics/background, wages, employment status, education, fertility
- N=6009, age $\in \{33, \ldots, 37\}$, and 35% of respondents graduated college

3.1 Setting up the specification

It looks like we can estimate some form of our model

- We have family background, cost of college (this is the `efc` variable)
- We have employment status, wage and number of children
- It looks like we'll have to have s be binary (`collgrad` variable)
- Also need to assume $x = age - 18$ if non-grad, $x = age - 22$ if grad (**mincer1974**)

Then we just need to add some functional form assumptions, and we'll be ready
$\varepsilon \sim$ Normal, $\eta_t \sim$ Logistic

$$u_{i1} = \alpha_0 + \alpha_1 \text{ parent\_college} + \alpha_2 \text{ efc} + \eta_1$$

$$u_{i2} = \gamma_0 + \gamma_1 \mathbb{E} \log w_i + \gamma_2 \text{ numkids} + \eta_2$$

3.2 Parameters of the empirical model

We can now detail the parameters of the empirical model

- **wage parameters** $(\beta, \sigma_\varepsilon)$
  - The latter is the std. dev. of income shocks
- **schooling parameters** (α)
- **employment parameters** (γ, δ)

Then write down a statistical objective function as a fn. of data and parameters

- e.g. maximize the likelihood, or minimize the sum of the squared residuals

3.3 Solving and Understanding How the Model Works

Solving the model:

- solve the dynamic utility max problem for given parameter values
- (we aren't estimating parameter values yet)

Understanding the model:

- simulate data from the model
- make sure the simulated data is consistent with the model's implications
- look at descriptive statistics from the simulated data

Start with as simple of a model as possible; make sure things are working

- When introducing more complexities, do "numerical comparative statics"
- Make sure the parameters move in the correct directions
  - e.g. $\uparrow \beta_1 \implies \uparrow$ schooling (ceteris paribus)

3.4 Julia Example

```
N = size(df,1)
beta = [1.65,.4,.06,-.0002]
sigma = .4;
df.exper = df.age .- ( 18*(1 .- df.collgrad) .+ 22*df.collgrad )
df.lwsim = beta[1] .+ beta[2]*df.collgrad .+ beta[3]*df.exper .+ beta[4]*df.exper.^2 .+ sigma*ran
df.lw    = log.(df.wage)
```

We can then compare how `df.lwsim` compares with `df.lw` in the data

```
describe(df;cols=[:lw,:lwsim])
# returns
| Row | variable | mean    | min     | median  | max     | nunique | nmissing | eltype
|     | Symbol   | Float64 | Float64 | Float64 | Float64 | Nothing | Union    | Type
-------------------------------------------------------------------------------------------
| 1   | lw       | 3.06219 | 2.07944 | 2.99573 | 5.01064 |         | 933      | Union{Missing, Fl
| 2   | lwsim    | 2.67169 | 1.12192 | 2.67668 | 3.98557 |         |          | Float64
```

3.5 Estimation

Most structural models require **nonlinear estimation**

- In nonlinear optimization, starting values are crucial
  - Initializing at random starting values is likely to give poor results
    - Keane recommends calibrating the model by hand
    - e.g. match the intercept of each equation to the $\overline{Y}$'s in the data
  - I recommend estimating an intercepts-only model (or with very few X's)
    - But this advice is model-specific!

3.5.1 Example using real data

In our simple model, we can get good starting values by estimating OLS and logits

- The wage equation can be estimated by OLS (on the subsample who are employed)

```
using GLM
\hat\beta= lm(@formula(lw ~ collgrad + exper + exper^2), df[df.employed.==1,:])
# returns
Coefficients:

               Estimate  Std. Error    t value  Pr(>|t|)    Lower 95%   Upper 95%

(Intercept)   2.94607    0.323145     9.11688     <1e-18   2.31255     3.57959
collgrad      0.534326   0.0271395   19.6881      <1e-82   0.481119    0.587532
exper        -0.0265561  0.0412115   -0.644386    0.5194  -0.107351    0.0542385
exper ^ 2     0.0014304  0.00132307   1.08112     0.2797  -0.00116346  0.00402426

df.elwage = predict(\hat\beta, df) # generates expected log wage for all observations
r2(\hat\beta)                                  # reports R2
sqrt(deviance(\hat\beta)/dof_residual(\hat\beta))  # reports root mean squared error
```

The u_t equations can be estimated as simple logits (on the full sample)

```
\hat\alpha = glm(@formula(collgrad ~ parent_college + efc), df, Binomial(), LogitLink())
# returns
Coefficients:

                  Estimate  Std. Error   z value  Pr(>|z|)   Lower 95%   Upper 95%

(Intercept)     -1.20091    0.0364888   -32.9118    <1e-99  -1.27243    -1.1294
parent_college   1.47866    0.068433     21.6074    <1e-99   1.34453     1.61278
efc              0.0450253  0.00437704   10.2867    <1e-24   0.0364464   0.0536041

\hat\gamma= glm(@formula(employed ~ elwage + numkids), df, Binomial(), LogitLink())
# returns
Coefficients:

               Estimate  Std. Error   z value  Pr(>|z|)  Lower 95%   Upper 95%

(Intercept)  -4.25036     0.454826   -9.34503    <1e-20  -5.1418    -3.35892
elwage        1.80081     0.149078   12.0796     <1e-32   1.50863    2.093
numkids      -0.0797204   0.0218106  -3.65512    0.0003  -0.122468  -0.0369724
```

3.5.2 Do these results make sense?

- It can be informative to try and interpret even these simple results
- wage equation:
  - insignificant return to experience is surprising; otherwise makes sense
- schooling choice:
  - If `efc` captures college costs, it should have a negative sign
  - This suggests omitted variable bias in this equation
- employment choice:
  - These results check out; may want to introduce nonlinearities in `numkids`

4 Validation

If you have a good model, it should be **valid** (i.e. predict well out of sample)

- Validation is not always possible, but it's good to do if you can
  - e.g. if experimental data, estimate on control group, validate on treatment group
  - e.g. see if model can replicate major policy change in data
- More simply, you could throw out half your data, then try to predict other half
  - This is typically not done if the full sample isn't huge

5 Policy Experiments

- This is the main reason to do structural estimation!
- Structural estimation $\implies$ recovering the DGP of the model
- Once we know the DGP, we can simulate data from it and do policy experiments
  - requires having policy-invariant parameters!
- We can predict the effects of:
  - proposed policies
  - hypothetical policies
- Contrast with RCTs, which only reveal effects of implemented policies

5.1 Example using real data

- We have two policy variables we could play with
  1. `efc` (i.e. how much gov't subsidizes college tuition & fees)
  2. return to schooling (this could change due to e.g. technological change)
- Here's how we would look at a counterfactual with lower cost:

```
df_cfl     = deepcopy(df)
df_cfl.efc = df.efc .- 1         # change value of efc to be $1,000 less
df.basesch = predict(\hat\alpha, df)     # predicted collgrad probabilities under status quo
df.cflsch  = predict(\hat\alpha, df_cfl) # predicted collgrad probabilities under counterfactual
describe(df;cols=[:basesch,:cflsch])
# returns
 Row  variable  mean      min       median    max       nunique  nmissing
      Symbol    Float64   Float64   Float64   Float64   Nothing  Int64

 1    basesch   0.350474  0.231313  0.24387   0.986715           0
 2    cflsch    0.341794  0.223404  0.235663  0.986111           0
```

Average likelihood of `collgrad` declines from 35% to 34.2%

- This doesn't make sense because the `efc` coefficient didn't make sense
- We can't assess the counterfactual of increasing the return to schooling
- Because `elwage` doesn't directly enter the `collgrad` logit model
- This is because we aren't really estimating the dynamic model yet

6 Conclusions

6.1 In summary: Why structural estimation?

- Want to examine effects of policies not yet implemented
- Learn more about economics by looking through the lens of a model
- Assess performance of theoretical models in explaining real-world data
- Can be used to build up long-run "canonical" models of behavior in many areas
- It can be really fun to do more complicated econometrics beyond simple regressions
- Observational data is much cheaper to collect than experimental data

6.2 In summary: Why <u>not</u> structural estimation?

- It's really difficult to write down and estimate a tractable, realistic model!
- It requires additional effort beyond data preparation and running regressions
- Understanding identification of the model takes a lot of effort, too
- It can be really miserable to try and debug the code of a structural estimation
- Many structural models can take weeks to estimate one specification
  - in addition to months spent coding/debugging beforehand
- As you can see, even with a simple model things have already gotten complicated!