MOVING THE NEXT NOTES <u>10 August 2020</u>

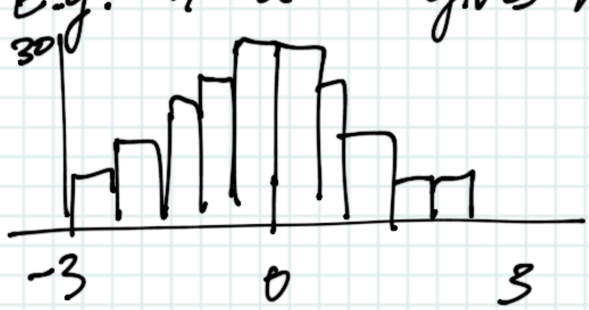
to the iPad.

So: trying to generate truth w/ more action in expectations (R, b).

Obsered:

1) If you increase the fe-support, fe's in the simulation become more tight around zero, and extrema are smaller. (E.g. $fe \in (-2,2)$ vs. $fe \in (-0.5, 0.5)$).
2) If you scale up α's, fe-distribution becomes more spread out: fatter tails

E.g. $4 \cdot \alpha^{true}$ gives the following fe-distrib



instead of sthg like

and $\bar{\bar{a}}$ fluctuates much more.

Estimating 4. α

- loss was initially much smaller 4.95 instead of on the order of 4000.
- estimation gets a lot of "fe was nan" messages and also a new one: "obj function returned NaN, trying a new point". then...
  - ↳ I already got this message 9 times
- It took 585 sec.
- Implied k^{-1} is <0 4 times!
- → maybe that's for negative k?
- Again, $loss(\hat{\alpha}) = 3.2 < loss(\alpha^{true}) = 416$.
- Not possible to evaluate loss function?!
  - ↳ It seems to be nans everywhere?
    Is that why the loss fct returned nan b/c all simulations were nan in the cross-section?
- Look at the loss: it sucks:
  - for α_1, α_5, it's apparently always nan
  - for α_2, α_4 it blows up
  - for α_3, it's oscillatory.

Any $\alpha > 0.5$ gives all nans. Maybe that's b/c since the truth is higher, it more quickly explodes?

- I'm not thrilled that many histories explode.
- Also very many VAR instability problems.

α's do more or less peak at the truth even now.
(well, α_3 doesn't and α_1 & α_5 have 2 minima around the true value, but still)

→ Does this indicate that the data-generation is wrong or the estimation?

- True moments changed very much:
  - moments pertaining to x scaled up, ones pertaining to π or i scaled down / were unchanged
  - own moments or cross-moments w/ π or i have the same shape
  - cross-moments w/ x have changed shape: are now more U- or reverse U-shaped.

↳ I think these make sense b/c since k is low, and $i = 1.5\pi$, x is the one that absorbs movements in expectations, while the pass-thru to π and to i is small.

⇒ The data-generation I think is correct. But this kinda suggests that the mean moments are also (more or loss) computed correctly, of which I'm not convinced.

↳ Let's try an estim w/ $\Psi_x = 0.5$.

- But suppose the data is generated correctly. Why isn't the "x absorbs all" mechanism matchable by the estimation? I mean, moments didn't move far from initial ones.

  ↳ Explosion % reveals that in the estimation, either 0% or 100% of the N simulations explode. No, sometimes it's 14, 79 or 99%. And it seems that the obj fit returns 'nan' when 100% explode. Yes.

  I learn also, by the way, that each iter of the solver evaluates the loss function exactly 6 times.

Ask Ryan

  ↳ Why? It's not changing α, or if it is, only one and only marginally

  <u>I'm also noticing that it's only considering α's really close to α_0 ⇒ multistart!</u>

I'm also noticing that the simulation is not giving the 'fe was nan' error most of the time, even though the solver gets a positive expl-percent. Hm!

When the "fe was nan" error comes, then the explosion-counter does catch it. But apparently there are a bunch of explosions that come not from these.

Ok, it's clear: when fe is nan, it's always caught. The other thing that causes "explosive" simulations is $k^{-1} < 0$. These get caught and register as explosions, but aren't really.

→ My check for "global nonnegativity" one wasn't catching everything!

Iter 40 : that's where the troubles start.

⇒ Initially, fe is nan (maybe the gain is too high) but then the bulk of the supposed explosions come from $k < 0$.

Ok... even when fe is nan it seems it's b/c or at the same time as $k < 0$ ...

→ It's at the same time. The initial iteration involves 5% "fe was nan" (5/100 simulations) but 100% have $k < 0$ at one point!

Let's see for the July 6 dataset ... and yes,
the initial iters involve 1% fe nan
69% k < 0
and later iters have 2% fe nan
86% k < 0

Converges after 23 iter.
Thus no more problems after iter 4.

Why is fe often nan early in the search and not late? Is it b/c of the initial values α_0?
And could a finer, and more broad fe-grid for the global nonnegativity test work?
↳ I introduced the variable broaden = 2,
which extends the fegrid-fine by ± broaden at both ends. It seems like this catches & subsumes all the previous errors!

!

It still finds the same min though :S
(But at least it's 2/3 of the time)

If I make the loss fct = nan when k < 0 globally, I can engineer the "Obj fct returned nan" error message.

Does this influence the estimate? (Previously I had set the objective to 1e+10) No it doesn't.
→ I don't think so but it does affect the plots of the loss I would think!

Let's go back to the scaled-up truth and plot the loss
Should understand why $k < 0$: is it, as I think, that α isn't convex and so a fe outside the grid is extrapolated to a $k < 0$?

The other thing is that this doesn't seem to really affect the behavior of the loss fct. It still converges to the same (wrong) thing. Actually, for the $4 \cdot \alpha$ truth, it converges to sthg different, but most likely that's b/c it took an avg of a smaller set.

The loss: $k < 0$ and $fe = nan$ return. Does that mean that fegrid-fine needs to be even broader?
Not necessarily b/c now it happens that e.g.

$fe = nan$ in 14%

but $k < 0$ in 0%.

In fact, $k < 0$ doesn't happen a lot! fe is the problem here!

Why is it that for the estimation, $k \approx 0$ is the root of all evil, but in plotting the loss, suddenly $fe = nan$ becomes the problem #1? Is it that the estim avoids high α's (or maybe also very low ones) that could cause turmoil?

<u>Materials 40, Section 4: Look into behavior</u> <u>11 August 2020</u>
<u>of simulation</u>. I think I learn a lot from this. Fig 7, which shows means & distribs of k^{-1}, $\bar{\pi}$ and fe are indicative.

1) While k^{-1} spends a lot of time being near 0 (see histogram), mean$(k^{-1}) \approx 0.0185$ (see means)
   ↳ If the moments capture the latter aspect, then the estim should have a hard time pushing α_3 down.
   But the moments should more capture the first. Moreover, what does it matter if mean$(k^{-1}) = 0.019$? It still can enter the lower parts of its state space, and it does! So scrap that.
2) The cross-sectional mean of $\bar{\pi}$ & of fe ↓ as N↑.
   Maybe this is why loss ↑ in N?
   I'm not sure this should happen? ⟶

As $N \to \infty$, $e \to 0$. So $fe \to 0$ and thus $\bar{\pi}$ too.

But that means that mean moments will reflect a sim in which $\bar{\pi}$ isn't moving much, and fe are very small.

Now turn to changing α's: do the following exercises:

- $4 \cdot \alpha$
- More edges: α_1 & α_5
- More middle: α_2 & α_4
- More 0-point: α_3

Obs. 1: If all α's ≤ 0.1, simulation doesn't blow up. At 0.11, they do. (2 occurances of "fe was nan", one of which was also $k_{-t-1} < 0$)

Obs. 2: $\alpha = 0$ doesn't cause explosions.

Obs 3: In the early scenarios, "fe was nan" is the only message. In the late ones, it comes together w/ "$k_{t-1} < 0$". Does this mean that $k < 0$ when α_3 is large?

↳ Making the code output α suggests that:

- "fe was nan" alone occurs when α_1 & $\alpha_5 \geq 0.11$
- "fe was nan" + "$k_{t-1} < 0$" occurs when α_2 & $\alpha_4 \geq 0.11$

- α_3 doesn't shit any water: it can also be 0.5, that doesn't cause explosions. (0-neighborhood indifference problem)
- these seem to occur across shock histories.

Now let's look at the plots of the means:

<u>Scenario 1: Varying α_1 and α_5:</u> 0, 0.05, 0.1

- Shifts k^{-1} up from (0.01, 0.012) to (0.02, 0.03)
- Shifts $\bar{\pi}$ from (−0.01, 0.01) to (−0.02, 0.04)
- Barely shifts fe: it remains in the (−0.2, 0.2) range

<u>Scenario 2: Varying α_2 and α_4</u>

- Shifts k^{-1} up from (0.005, 0.001) to (0.05, 0.06)
- Barely shifts $\bar{\pi}$: it remains in the (−0.01, 0.03) range
- Barely shifts fe: it remains in the (−0.2, 0.2) range

<u>Scenario 3: Varying α_3:</u>

- Shifts k^{-1} up from (0.015, 0.02) to (0.05, 0.06)
- Barely shifts $\bar{\pi}$: it remains in the (−0.01, 0.03) range
- Barely shifts fe: it remains in the (−0.2, 0.2) range

↳ So: why aren't $\alpha_{2,3,4}$ moving $\bar{\pi}$ & fe?
  ↳ is it b/c in that range $k^{-1} \cdot fe$ is a really small number? I.e.: is the 0-neighborhood indifference a problem

here too?

→ I think so! I think what is happening is that the entire $k \in (-1, 1)$-region yields too small $k^{-1} \cdot k$ products such that they do matter a little bit (the loss does take a min there) but not a lot (the loss is very flat)

⇒ To be identified, you need to consider α's associated w/ $k > 1$ (possibly even greater). But: you also need a truth that's not too big, b/c $\alpha > 0.1$ seems to be ill tolerated by the model.

→ TO DO!

For Peter meeting:

① Loss does have a min, almost at right spot. (Fig 1)

a) Need convexity assumption – don't get removal (Fig. 2)

b) Rescaling W affects loss, although no indication of inversion issues. (Fig 3.b)

c) Adding E(·) screws things up in a way I don't get. (Fig 4)

(② Truth w/ more action in E(·)) (Fig 6)

I thought that if E(·) aren't moving, then both rescaling

and informativeness of moments might be screwed up.
Explosions hinted at something else.
$\alpha \overset{!}{\leq} 0.1$ to avoid explosions.

(3) Behavior of simulation

a) Loss ↑ in N ↑ b/c $\bar{\pi}$ & fe ↓ in α? (Fig 7 b&d)

b) Fig 8: Only α_1 & α_5 can affect $\bar{\pi}$, and even that can't affect fe. (Fig 8)

Peter meeting

11 Aug 2020

Fig 1. Loss higher at $\hat{\alpha}$ vs α^{true}

Again: b/c data from model is nonlinear, this may not suggest that α^{true} is a local min.

Thinks that a lot of problems come from that est a nonstat model is tricky.

The next issue: holding α's fixed, but estimating more than one α may not work. This would happen if you est $Y = \beta_1 X_1 + \beta_2 X_2$ and X_1 & X_2 multicollinear.

↳ might be some weird interaction between α's so that the concavity restriction helps provide curvature.

Fig 8 conclusions sound exactly right.

Rescaling : still quite strange

Fix α, check loss w/ and w/o scaling → should recover the scaling factor.

Also explains why imposing convexity helps.

Wanna guard against explaining what is really a coding error. → Should go back to that.

The level of loss fct has no meaning; only the curvature.

Min in Fig 1. don't line up w/ truth: not a problem b/c if you sim using a diff c, you'd get a diff min.

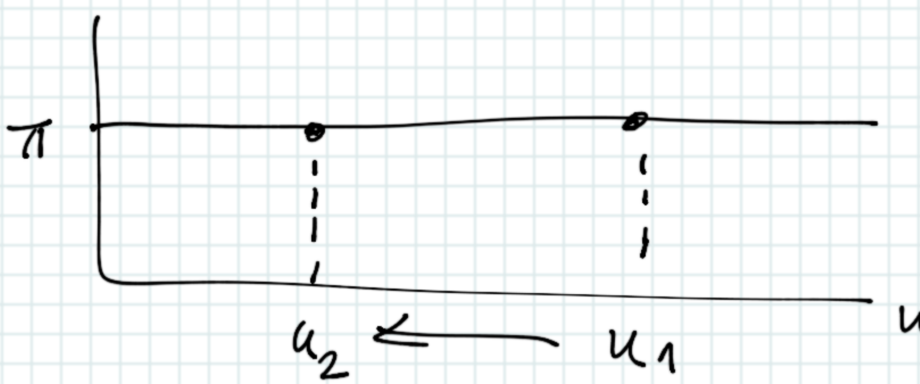
If std. errs large, can be b/c of
- → ID → doesn't go away as $N \to \infty$
- → sampling error
  - ↳ goes away as $N \to \infty$

Recall the logistic functional specification for anchoring pt: somehow more structure has to be imposed on the anchoring function to guide the info in the data

Analogy: Sellena?
Silvana Tenreyro: $\hat{PC}$: slope is ≈ 0 → b/c CB is credible! So est a PC w/ these data doesn't tell you what would happen if CB abandoned target. (b/c that's just not in the data!)

→ b/c obs that have modest fe, we don't learn a lot about how agents learn from forecast errors.

"flat PC": "π isn't moving in tandem w/ u anymore"



↳ he says that the "flat PC" term refers to the idea that π doesn't follow u, not the other way around!

low u doesn't lead to any diff π!

- A fall-back approach: a simpler functional form for anch. fct.
- Or: pick unconditional moments w/o VAR or weighting matrix

He called it an in-between btwn est & calibration.
"Moment-matching": take sample moments & match those.

Successful esti will involve picking restrictions on 1st OR on the anchoring fct to allow to glean info that <u>is</u> in the data.

↳ And I'm not sure why he calls the "moment-matching" by that name: it's not much different than my SMM.