Kalman filter and Maximum Likelihood

Petr Sedláček

Tools for Macroeconomists:
The essentials

Overview for today

- looking at the big picture
  - what you've done in the past days
  - what we'll do today and Friday
- intro into estimation
  - Kalman filter (based on Hamilton)
  - Maximum Likelihood estimation
- estimating DSGE models
  - DSGE and time-series models
  - Maximum Likelihood in Dynare
- extensions

Looking at the big picture

What tools are needed in modern macro?

(according to Victor Rios-Rull)

1. theoretical tools
    - use models to look at data
2. computational tools
    - characterize model outcomes
3. empirical tools
    - analyze statistical properties of data and model

What you'll learn in the next days

What does it mean to "solve" a model?

- what are we solving for?
- why is this a tough problem?

Which (general) tools will you learn towards this end?

- function approximation
- numerical integration

What ways of constructing model solutions will you cover?

- perturbation
- projection

What will we learn today and Friday?

How to parametrize models

- alternative methods
    - calibration
    - matching moments
    - estimation
        - **Maximum Likelihood**
        - **Bayesian methods**

Main thing to remember

"The goal of computing is insight, not numbers"

Richard Hamming

Mathematician and computer scientist

Alternative parametrization methods

Alternative methods of parametrizing models

- calibration
- matching moments
  - general methods of moments
  - simulated method of moments
  - indirect inference
- estimation
  - Maximum Likelihood
  - Bayesian estimation

Calibration

- wide-spread methodology
  - at least since Kydland and Prescott (1982)
- although calibration is also an empirical exercise
- it lacks the probabilistic interpretation
- constrained by (a priori identified) features in the data

Kydland and Prescott (1996):
It is important to emphasize that the parameter values selected are not the ones that provide the best fit in some statistical sense.

Calibration

Main idea:

- model parameters pinned down by *selected* real-world features
- "evaluate" model based on a *different* set of features

Criticism of calibration

- no formal rules on selecting targets
- no formal rules on selecting dimensions of model fit
- no formal rules on comparing alternatives

We'll go through an example to give you an idea

Matching moments (GMM, SMM, II)

- idea similar to calibration:
  - parametrize by a set of moments (features) of the data
  - judge model performance by a different set of moments
- matching moments adds statistical rigor
  - estimation (limited information)
  - hypothesis testing

Generalized method of moments (Hansen, 1982)

Main idea:

- select a set of moments (orthogonality conditions)

$$\mathbb{E}[f(x_t, \Psi)] = 0$$

  - x_t is a vector of variables, Ψ are model parameters

- choose Ψ s.t. sample analogs $g(X, \Psi) = 1/T \sum_t f(x_t, \Psi)$ hold
  - exactly identified case: # of params. $=$ # of conditions
  - over-identified case: # of params. $<$ # of conditions

Simulated method of moments

- sometimes orthog. conditions cannot be assessed analytically
- moment-matching estimation based on simulations
    - retains asymptotic properties of GMM

Indirect inference

Main idea:

- parameters pinned down by selected reduced-form estimates
  - choose parameters s.t. simulated data from structural model
  - replicates reduced-form results in the data
- judge model performance by a different set of moments

Estimation

Full information method

- need to specify entire distributions (of driving forces)
    - model fit (and parameters) based on implied likelihood function
    - and prior information in case of Bayesian estimation

We'll talk about estimation in detail!

Introduction into Maximum Likelihood estimation

The Kalman filter

Time series model

$$
\begin{aligned}
y_t &= H'\zeta_t + w_t, & \mathbb{E}(w_t, w_t') &= R \quad \forall t \\
\zeta_{t+1} &= F\zeta_t + v_{t+1}, & \mathbb{E}(v_t, v_t') &= Q \quad \forall t
\end{aligned}
$$

- y_t is observed, but ζ_t is not
- Kalman filter enables you to get an estimate of ζ_t

Some preliminaries

- for now assume coefficients are known
  - later on we will show how to estimate them
  - they could even be time-varying
- initial conditions: ζ_1 has mean $\widehat{\zeta}_{1|0}$ and variance $P_{1|0}$
- state and observation disturbances are
  - uncorrelated over time
  - uncorrelated with each other (at all leads and lags)
  - orthogonal to ζ_1

Purpose of the Kalman filter

- calculate the expectation of the unobserved states
- given observations on y

$$\widehat{\zeta}_{t+1} = \widehat{\mathbb{E}}(\zeta_{t+1} | \mathcal{Y}_t),$$

$$\mathcal{Y}_t = (y'_t, y'_{t-1}, ..., y'_1)$$

What is the objective?

How to pick the (right) forecast?

- specify a *loss function* (objective function)
- evaluate the "usefulness" of the forecast according to it
- convenient results obtained with a quadratic loss function

$$\min E(\zeta_{t+1} - \widehat{\zeta}_{t+1|t})^2$$

- assuming a linear functional form
- $\rightarrow \widehat{\zeta}_{t+1|t}$ is a *linear projection* of ζ_{t+1} on its regressors

Linear projection digression

- closely related to OLS
- assume existence of following first and second moments:
  - $\overline{z} = E[z]$, $\overline{x} = E[x]$
  - $\Sigma_{z,z} = E[(z - \overline{z})(z - \overline{z})']$
  - $\Sigma_{x,x} = E[(x - \overline{x})(x - \overline{x})']$
  - $\Sigma_{z,x} = E[(z - \overline{z})(x - \overline{x})']$
- linear projection (of z on x) is a function $\widehat{E}[z|x] = a + bx$

Linear projection digression

linear projection picks a and b to minimize MSE:

$$b = \frac{\Sigma_{z,x}}{\Sigma_{x,x}}$$

$$a = \overline{z} - b\overline{x}$$

$$\widehat{z} = \widehat{E}(z|x) = \overline{z} + \frac{\Sigma_{z,x}}{\Sigma_{x,x}}(x - \overline{x})$$

Linear projection vs. linear regression

- linear regression (OLS) seeks effect of x on z
- keeping all else (including error term) constant
- linear projection is concerned "only" with forecasting
- $\rightarrow$ doesn't matter if $x \rightarrow z$ or $z \rightarrow x$

Back to the Kalman filter

- purpose of Kalman filter is to estimate unobserved states
- will do so using linear projections (given data and structure)
- main trick is to formulate it recursively

Formulating a recursive scheme

Main idea of the Kalman filter recursions:

- given starting values for $\zeta_{1|0}$ and observation of y_1
- $\rightarrow$ forecast $\widehat{\zeta}_{2|1}$
- given $\widehat{\zeta}_{2|1}$ and observation of y_2
- $\rightarrow$ forecast $\widehat{\zeta}_{3|2}$ ...

Formulating a recursive scheme

For convenience, split the above into the following steps:

- given starting values for $\zeta_{1|0}$ and observation of y_1
- $\rightarrow$*update* state $\widehat{\zeta}_{1|1}$
- given $\widehat{\zeta}_{1|1}$ and model structure
- $\rightarrow$ forecast $\widehat{\zeta}_{2|1}$ ...

1. Update step

use linear projection to produce update of ζ_t

- conditional on expectations from $t-1$
- conditional on observation of y_t

$$\widehat{z} = \widehat{E}(z|x) = \overline{z} + \frac{\Sigma_{z,x}}{\Sigma_{x,x}}(x - \overline{x})$$

$$\begin{aligned}\widehat{\zeta}_{t|t} =& \widehat{\mathbb{E}}[\zeta_t|\mathcal{Y}_t] = \widehat{\zeta}_{t|t-1} + \\ & \mathbb{E}\left[(\zeta_t - \widehat{\zeta}_{t|t-1})(y_t - \widehat{y}_{t|t-1})'\right] x \mathbb{E}\left[(y_t - \widehat{y}_{t|t-1})(y_t - \widehat{y}_{t|t-1})'\right]^{-1} x \\ & (y_t - \widehat{y}_{t|t-1})\end{aligned} \tag{1}$$

Using the model structure

covariance term

$$
\begin{aligned}
&\mathbb{E}\left[(\zeta_t - \widehat{\zeta}_{t|t-1})(y_t - \widehat{y}_{t|t-1})'\right] \\
&= \mathbb{E}\left[(\zeta_t - \widehat{\zeta}_{t|t-1})(H'(\zeta_t - \widehat{\zeta}_{t|t-1}) + w_t)'\right] \\
&= \mathbb{E}\left[(\zeta_t - \widehat{\zeta}_{t|t-1})(\zeta_t - \widehat{\zeta}_{t|t-1})'H\right] \\
&= P_{t|t-1}H
\end{aligned}
\tag{2}
$$

$P_{t|t-1}$ is the related MSE

Using the model structure

variance term

$$\begin{aligned} &\mathbb{E}\left[(y_t - \widehat{y}_{t|t-1})(y_t - \widehat{y}_{t|t-1})'\right] \\ &\quad = \mathbb{E}\left[H'(\zeta_t - \widehat{\zeta}_{t|t-1})(\zeta_t - \widehat{\zeta}_{t|t-1})H\right] + \mathbb{E}\left[w_t w_t'\right] \\ &\quad = H' P_{t|t-1} H + R \end{aligned} \tag{3}$$

Using the model structure

error term

$$\widetilde{y}_{t|t-1} = y_t - \widehat{y}_{t|t-1} = y_t - H'\widehat{\zeta}_{t|t-1} \quad (4)$$

Putting it all together

Our update of ζ_t given information up until period t:

$$\widehat{\zeta}_{t|t} = \widehat{\zeta}_{t|t-1} + P_{t|t-1}H(H'P_{t|t-1}H + R)^{-1}(y_t - H'\widehat{\zeta}_{t|t-1})$$

2. Forecast step

use model structure to forecast the state

$$\begin{aligned}\widehat{\zeta}_{t+1|t} =& \widehat{\mathbb{E}}\left[\zeta_{t+1}|\mathcal{Y}_t\right] \\ =& F\widehat{\mathbb{E}}\left[\zeta_t|\mathcal{Y}_t\right] + \widehat{\mathbb{E}}\left[v_{t+1}|\mathcal{Y}_t\right] \\ =& F\widehat{\mathbb{E}}\left[\zeta_t|\mathcal{Y}_t\right]\end{aligned} \tag{5}$$

Collapsing the two steps

Combining (1) and (5) and using (2) to (4) we can write

$$\widehat{\zeta}_{t+1|t} = F\widehat{\zeta}_{t|t-1} + \underbrace{FP_{t|t-1}H(H'P_{t|t-1}H + R)^{-1}}_{\text{Kalman gain}}(y_t - H'\widehat{\zeta}_{t|t-1})$$

We're not done yet

Still need to define recursions for P's:

Update step (use (1) to substitute out $\widehat{\zeta}_{t|t}$)

$$\begin{aligned}
P_{t|t} =& \mathbb{E}\left[(\zeta_t - \widehat{\zeta}_{t|t})(\zeta_t - \widehat{\zeta}_{t|t})'\right] \\
=& \mathbb{E}\left[(\zeta_t - \widehat{\zeta}_{t|t-1})(\zeta_t - \widehat{\zeta}_{t|t-1})'\right] \\
& - \mathbb{E}\left[(\zeta_t - \widehat{\zeta}_{t|t-1})(y_t - \widehat{y}_{t|t-1})'\right] x \\
& \mathbb{E}\left[(y_t - \widehat{y}_{t|t-1})(y_t - \widehat{y}_{t|t-1})'\right]^{-1} x \\
& \mathbb{E}\left[(y_t - \widehat{y}_{t|t-1})(\zeta_t - \widehat{\zeta}_{t|t-1})'\right] \\
=& P_{t|t-1} - P_{t|t-1}H(H'P_{t|t-1}H + R)^{-1}H'P_{t|t-1}
\end{aligned} \tag{6}$$

We're not done yet

Forecast step (use (5) to substitute out $\widehat{\zeta}_{t+1|t}$)

$$\begin{aligned} P_{t+1|t} =& \mathbb{E}\left[(\zeta_{t+1} - \widehat{\zeta}_{t+1|t})(\zeta_{t+1} - \widehat{\zeta}_{t+1|t})'\right] \\ =& \mathbb{E}\left[(F\zeta_t + v_{t+1} - F\widehat{\zeta}_{t|t})(F\zeta_t + v_{t+1} - F\widehat{\zeta}_{t|t})'\right] \\ =& F\mathbb{E}\left[(\zeta_t - \widehat{\zeta}_{t|t})(\zeta_t - \widehat{\zeta}_{t|t})'\right]F' + \mathbb{E}\left[v_{t+1}v_{t+1}'\right] \\ =& FP_{t|t}F' + Q \end{aligned} \tag{7}$$

Collapsing the two steps

Combining (6) and (7) we can write

$$P_{t+1|t} = F\left[P_{t|t-1} - P_{t|t-1}H(H'P_{t|t-1}H + R)^{-1}H'P_{t|t-1}\right]F' + Q$$

Summary of recursive formulation

update:

$$\widehat{\zeta}_{t|t} = \widehat{\zeta}_{t|t-1} + P_{t|t-1} H (H' P_{t|t-1} H + R)^{-1} (y_t - H' \widehat{\zeta}_{t|t-1})$$

$$P_{t|t} = P_{t|t-1} - P_{t|t-1} H (H' P_{t|t-1} H + R)^{-1} H' P_{t|t-1}$$

forecast:

$$\widehat{\zeta}_{t+1|t} = F \widehat{\zeta}_{t|t}$$

$$P_{t+1|t} = F P_{t|t} F' + Q$$

Summary of recursive formulation

combined specification:

$$\widehat{\zeta}_{t+1|t} = F\widehat{\zeta}_{t|t-1} + K_t(y_t - H'\widehat{\zeta}_{t|t-1})$$

$$K_t = FP_{t|t-1}H(H'P_{t|t-1}H + R)^{-1}$$

$$P_{t+1|t} = F\left[P_{t|t-1} - P_{t|t-1}H(H'P_{t|t-1}H + R)^{-1}H'P_{t|t-1}\right]F' + Q$$

Maximum Likelihood

Estimating parameters

- up to now, we assumed that model parameters are known
- we can also estimate them with Maximum Likelihood (ML)
  - i.e. given data on y_t and initial conditions
  - estimate $\Psi = [H, F, Q, R]$
- the Kalman filter is particularly convenient for this task

Preliminaries

- if $\zeta_{1|0}$ is Gaussian and $\{w_t, v_t\}_{t=1}^{T}$ are Gaussian
- $\rightarrow$ distribution of y_t conditional on $\mathcal{Y}_{t-1}$ is also Gaussian

$$\begin{aligned}\widetilde{y}_{t|t-1}|\mathcal{Y}_{t-1} \sim & N(0, H'P_{t|t-1}H + R) \\ y_t|\mathcal{Y}_{t-1} \sim & N(H'\widehat{\zeta}_{t|t-1}, H'P_{t|t-1}H + R)\end{aligned}$$

Preliminaries

- given values of $\Psi \rightarrow$ calculate mean and variance of y
- we know the distribution of y
- $\rightarrow$ calculate the probability (likelihood) of $(y_1, ..., y_T)$

Likelihood function

the likelihood of a given (Gaussian) observation is

$$\frac{1}{\sigma\sqrt{2\pi}}\exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$

in our case this is

$$\begin{aligned} f(y_t|\mathcal{Y}_{t-1};\Psi) =&(2\pi)^{-1/2}(H'P_{t|t-1}H+R)^{-1/2}x \\ &\exp\left\{-1/2(y_t-\widehat{y}_{t|t-1})'(H'P_{t|t-1}H+R)^{-1}(y_t-\widehat{y}_{t|t-1})\right\} \\ &\text{for } t=1,...,T \end{aligned} \tag{8}$$

Likelihood function

we are interested in the likelihood of the entire sample

- because forecast errors are orthogonal to each other
- $\mathcal{L}(\mathcal{Y}_t|\Psi) = f(y_0;\Psi)\Pi_{t=1}^{T} f(y_t|\mathcal{Y}_{t-1};\Psi)$

it is convenient to work with the sample log-likelihood:

$$\log \mathcal{L}(\mathcal{Y}_t|\Psi) = \log f(y_0) + \sum_{t=1}^{T} \log f(y_t|\mathcal{Y}_{t-1};\Psi)$$

Why is the Kalman filter convenient?

- the Kalman filter produces $\widehat{y}_{t|t-1}$ and $P_{t|t-1}$
- the (log)-likelihood is easy to construct with the Kalman filter
- one can then maximize it with respect to the parameters Ψ
- this will be your task in the afternoon session

Back to DSGE models

What is the relation to DSGE models?

What next?

- neoclassical growth model example
  - linearized equations
  - simple example of constructing the likelihood
- casting DSGE models into state-space form
- singularity problem
- Maximum Likelihood in Dynare

Neoclassical growth model

- representative household maximizing expected lifetime utility
- household owns production technology
- capital is the only factor of production
- resources spent on consumption and investment into capital
- each period existing capital depreciates at certain rate
- production subject to exogenous fluctuations in productivity

Production

$$y_t = Z_t k_t^{\alpha}$$

$$Z_t = 1 - \rho + \rho Z_{t-1} + \epsilon_t$$

$$\mathbb{E}\epsilon_t = 0$$

$$\mathbb{E}\epsilon_t^2 = \sigma_z^2$$

Household decision

$$\max_{\{c_t, k_{t+1}\}_{t=0}^{\infty}} \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t \log(c_t)$$

$$\text{s.t.} \quad c_t + k_{t+1} = y_t + (1-\delta)k_t$$

$$k_0 \quad \text{given}$$

$$Z_0 \quad \text{given}$$

Neoclassical growth model

$$c_t^{-1} = \mathbb{E}_t \left[\beta c_{t+1}^{-1} (\alpha z_{t+1} k_t^{\alpha-1} + 1 - \delta) \right]$$

$$c_t + k_t = z_t k_{t-1}^{\alpha} + (1-\delta) k_{t-1}$$

$$z_t = 1 - \rho + \rho z_{t-1} + \epsilon_t$$

$$\epsilon_t \sim N(0, \sigma^2)$$

Solution

What is the solution?

- a sequence $\{c_t, k_{t+1}\}_{t=0}^{\infty}$
- maximizing the expected discounted sum of per-period utilities

Sounds like a tough problem!

- different $k_0 \rightarrow$ optimal sequences different!
- different realizations of $Z_t \rightarrow$ optimal sequences different!

Neoclassical growth model

$$c_t^{-1} = \mathbb{E}_t \left[\beta c_{t+1}^{-1} (\alpha z_{t+1} k_t^{\alpha-1} + 1 - \delta) \right]$$

$$c_t + k_t = z_t k_{t-1}^{\alpha} + (1-\delta) k_{t-1}$$

$$z_t = 1 - \rho + \rho z_{t-1} + \epsilon_t$$

$$\epsilon_t \sim N(0, \sigma^2)$$

Linearized version

Wouter will tell you all about solving models...

$$
\begin{aligned}
k_t =& \overline{k} + a_{kk}(k_{t-1} - \overline{k}) + a_{kz}(z_t - \overline{z}) \\
z_t =& 1 - \rho + \rho z_{t-1} + \epsilon_t \\
& \epsilon_t \sim N(0, \sigma^2) \\
& k_0, z_0 \text{ given}
\end{aligned}
$$

- a_{kk}, a_{kz} and $\overline{k}$ are *known* functions of structural parameters Ψ
- $\Psi = [\alpha, \beta, \delta, \rho, \sigma, z_0]$

Estimating structural parameters

consider estimating the structural parameters using ML

- how to write down the likelihood of the model?
  - for Kalman filter, we still need to figure out H, F
  - can we do something else (simpler) instead?

Simple case of evaluating the likelihood

$$k_t = \overline{k} + a_{kk}(k_{t-1} - \overline{k}) + a_{kz}(z_t - \overline{z})$$
$$z_t = (1 - \rho)\overline{z} + \rho z_{t-1} + \epsilon_t$$

Simple case of evaluating the likelihood

$$
\begin{aligned}
k_t &= \overline{k} + a_{kk}(k_{t-1} - \overline{k}) + a_{kz}(z_t - \overline{z}) \\
z_t &= (1-\rho)\overline{z} + \rho z_{t-1} + \epsilon_t
\end{aligned}
$$

$$
\begin{aligned}
z_t &= \overline{z} + \frac{k_t - \overline{k} - a_{kk}(k_{t-1} - \overline{k})}{a_{kz}} \\
\epsilon_t &= \overline{z}(\rho - 1) + z_t - \rho z_{t-1}
\end{aligned}
$$

- given a guess of Ψ and given k_0, k_1 and z_0
- $\rightarrow$ calculate z_1 $\rightarrow$ calculate ϵ_1 $\rightarrow$ calculate z_2 etc.

Log-likelihood

in this case, the log-likelihood is simply

$$\log\left(\frac{1}{\sigma\sqrt{2\pi}}\right) - \frac{\epsilon_t(\Psi)^2}{2\sigma^2}$$

Putting model into state-space form

- the neoclassical growth model is relatively simple
- for more complex models, policy function inversion is tough
- but we know that
    - the Kalman filter is convenient for likelihood construction
    - because it produces $y_t - \widehat{y}_{t|t-1}$ and $P_{t|t-1}$
- the question is how to cast DSGE model into state-space form

DSGE mode in state-space form

$$\begin{aligned} \zeta_{t+1} &= F\zeta_t + v_{t+1}, & \mathbb{E}(v_t, v_t') &= Q \quad \forall t \\ y_t &= H'\zeta_t + w_t, & \mathbb{E}(w_t, w_t') &= R \quad \forall t \end{aligned}$$

- what is the observable and what are the unobserved states?

DSGE mode in state-space form

$$\begin{bmatrix} k_t - \overline{k} \\ z_{t+1} - \overline{z} \end{bmatrix} = \begin{bmatrix} a_{kk} & a_{kz} \\ 0 & \rho \end{bmatrix} \begin{bmatrix} k_{t-1} - \overline{k} \\ z_t - \overline{z} \end{bmatrix} + \begin{bmatrix} 0 \\ \epsilon_{t+1} \end{bmatrix}$$

$$k_{t-1} - \overline{k} = [1 \quad 0] \begin{bmatrix} k_{t-1} - \overline{k} \\ z_t - \overline{z} \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix}$$

Different observables?

What if we don't have data on capital, but only on output (p_t)?

$$\begin{bmatrix} k_t - \overline{k} \\ z_{t+1} - \overline{z} \\ p_t - \overline{p} \end{bmatrix} = \begin{bmatrix} a_{kk} & a_{kz} & 0 \\ 0 & \rho & 0 \\ \alpha \overline{z}\overline{k}^{\alpha-1} & \overline{k}^{\alpha} & 0 \end{bmatrix} \begin{bmatrix} k_{t-1} - \overline{k} \\ z_t - \overline{z} \\ p_{t-1} - \overline{p} \end{bmatrix} + \begin{bmatrix} 0 \\ \epsilon_{t+1} \\ 0 \end{bmatrix}$$

$$[p_{t-1} - \overline{p}] = [0 \;\; 0 \;\; 1] \begin{bmatrix} k_{t-1} - \overline{k} \\ z_t - \overline{z} \\ p_{t-1} - \overline{p} \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix}$$

Different observables?

An alternative state-space for the same setup would be

$$\begin{bmatrix} k_t - \overline{k} \\ z_{t+1} - \overline{z} \end{bmatrix} = \begin{bmatrix} a_{kk} & a_{kz} \\ 0 & \rho \end{bmatrix} \begin{bmatrix} k_{t-1} - \overline{k} \\ z_t - \overline{z} \end{bmatrix} + \begin{bmatrix} 0 \\ \epsilon_{t+1} \end{bmatrix}$$

$$\left[p_t - \overline{p}\right] = \begin{bmatrix} \alpha \overline{z}\overline{k}^{\alpha-1} & \overline{k}^{\alpha} \end{bmatrix} \begin{bmatrix} k_{t-1} - \overline{k} \\ z_t - \overline{z} \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix}$$

What if we also observe productivity?

What if we observe capital and also productivity (z_t)?

- if our model is the true data-generating process
  - $\rightarrow$ likelihood $=$ 1 for true Ψ and 0 otherwise
- if our model is not the true data-generating process
  - $\rightarrow$ likelihood $=$ 0 for any values of Ψ

What if we also observe productivity?

To understand the above notice that with productivity data

- 4 periods are enough to pin down $\overline{k}$, ρ, a_{kk}, a_{kz}

What about the other periods?

- shocks adjust such that productivity is matched perfectly

$$z_t = 1 - \rho + \rho z_{t-1} + \epsilon_t$$

What if we also observe productivity?

What if we now add data on capital?

- same logic applies:
  - 4 productivity observations enough to pin down $\overline{k}$, ρ, a_{kk}, a_{kz}
  - rest of productivity time-series matched by shocks
- capital data consistent with model only if model is true DGP!

$$k_t = \overline{k} + a_{kk}(k_{t-1} - \overline{k}) + a_{kz}(z_t - \overline{z})$$

Ways out?

can we simply add an error term?

$$k_t = \overline{k} + a_{kk}(k_{t-1} - \overline{k}) + a_{kz}(z_t - \overline{z}) + u_t$$

Ways out?

- if u_t is a structural shock (e.g. preferences)
  - $\rightarrow$ its law of motion influences policy function ($\bar{k}$, a_{kk}, a_{kz})
- if u_t is measurement error
  - OK from an econometric point of view
  - but is it truly measurement error?

Ways out?

what if we also observe consumption (but not productivity)?

$$k_t = \overline{k} + a_{kk}(k_{t-1} - \overline{k}) + a_{kz}(z_t - \overline{z})$$

$$c_t = \overline{c} + a_{ck}(k_{t-1} - \overline{k}) + a_{cz}(z_t - \overline{z})$$

won't work!

- given Ψ, you can back out z_t from both equations
- with actual data this will give inconsistent answers

Singularity problem

- (stochastic) singularity:
  - many endogenous variables ...
  - driven by a smaller number of structural shocks
- $\rightarrow$ some observables are linear combinations of others
- $\rightarrow$ the var-covar matrix of observables is *singular*
- what is the problem mathematically?

General rule

- for every observable, you need at least one unobservable shock
- (letting them be measurement error is hard to defend)

Note that:

- more shocks (measurement errors) than observables is OK
- the choice of observables for estimation is not innocent
- there are ways to choose observables carefully
    - see e.g. Canova, Ferroni, Matthes (2012)

Maximum Likelihood in Dynare

What is Dynare and why use it?

- (free) software for perturbation solutions and more
  - also estimation: ML, Bayesian
  - many options
- you MUST know what it is doing
- once you do, its a very useful tool

Where/how to get Dynare

- download at `www.dynare.org`
- install *and*
- in Matlab set path to `.../Dynare/Matlab`
- read the documentation

What does Dynare do?

- main file type is a $*.mod$ file
- into this file you specify
  - variables of your model
  - parameters and their values
  - model equations (linearized or not)
  - initial values (ideally steady state)
  - solution method (1st or higher order)
    - many other options (IRFs, simulations, moments etc.)
    - you can also estimate models

Notation in Dynare

- variables known at the beginning of period
- are dated as $t-1$!
  - k_t: capital *choice* in period t
  - k_{t-1}: capital stock available in t

Policy rules

- Dynare produces perturbation approximation to policy rules
- for now consider linear approximations
- linear in what?!
  - Dynare doesn't know that "k" means capital
  - k could be
    - level of capital
    - log of capital
- its up to you to decide
- Dynare will produce policy rules for specified variables

Policy rules

- in neoclassical growth model
- Dynare generates following policy rules

$$k_t = \overline{k} + a_{kk}(k_{t-1} - \overline{k}) + a_{kz}(z_{t-1} - \overline{z}) + a_{k\epsilon}\epsilon_t$$

- i.e. it splits structural shocks into
  - past value and
  - innovation
  - i.e. if $z_t = 1 - \rho + \rho z_{t-1} + \epsilon_t$ then $a_{kz} = \rho a_{k\epsilon}$

Dynare blocks

A Dynare file has several blocks:

1. list of variables
2. list of exogenous shocks
3. list of model parameters and their values
4. model block (optimality conditions)
5. shock properties
6. initial values
7. solution (and other) commands

Definitions and parametrization

1. Specify variables
   - specified by typing “var” and then listing variables
2. Specify exogenous shocks
   - specified by typing “varexo” and then listing shocks
3. Specify parameters and their values
   - specified by typing “parameters” and then listing parameters
   - each parameter must then be assigned a value
     - either directly in Dynare file
     - or by loading it from outside Dynare file
     - the latter is more convenient for calibration

Model block

4. Model block contains equilibrium conditions
   - initialize block by typing “model;”
   - end it by typing “end;”
   - in between simply write your model equations

Specifics

- Dynare figures out there are expectations when you write $t+1$
- e.g. the Euler equation:

```
c^(-gamma)=beta*c(+1)^(-gamma)*(alpha*Z(+1)k^(alpha-1)+1-delta)
```

Shock properties

5. Shock properties

- initialize the block by typing “shocks;”
- end it by typing “end;”
- in between specify shock properties
  - e.g. “var e; stderr sigZ;”
  - can specify more, like correlations etc.

Initial values

6. Initial values

- initialize block by typing “initval;”
- end it by typing “end;”
- inbetween list the initial values of all variables
  - ideally give Dynare the steady state
  - often difficult to compute, so supply it yourself

Solution

7. Give Dynare the green light to solve the model

- “`stoch_simul(options)`”
- options include
  - order of perturbation: e.g. “order=1” for linear
  - length of IRFs: e.g. IRF=20
  - many, many more

To actually run Dynare type `dynare filename.mod`

Other useful features

- "resid" command shows equation errors
  - it plugs initial values into model equations
  - they should all be zero in steady state
  - useful for finding out typos

Neoclassical growth model again

$$c_t^{-\nu} = \mathbb{E}_t \left[\beta c_{t+1}^{-v} (\alpha z_{t+1} k_t^{\alpha-1} + 1 - \delta) \right]$$

$$c_t + k_t = z_t k_{t-1}^{\alpha} + (1-\delta) k_{t-1}$$

$$z_t = 1 - \rho + \rho z_{t-1} + \epsilon_t$$

$$\epsilon_t \sim N(0, \sigma^2)$$

ML estimation in Dynare: initialization

- initialize as usual

```
var c, k, z, y;
varexo epsilon;
parameters beta, rho, alpha, nu, delta, sigma;
```

- set values for all parameters (even those that are estimated)

```
alpha = 0.36;
rho = 0.95;
beta = 0.99;
nu = 1;
delta = 0.025;
sigma = 0.01;
```

ML estimation in Dynare: setting it up

- after model part, and specification of steady state
- tell Dynare which parameters he should estimate

```
estimated_params;
stderr epsilon, 0.01, 0, 0.2;
end;
```

- the above tells Dynare to
  - estimate σ, the st. error of the productivity disturbance
  - 0.01 is the initial value (starting point for minimization routine)
  - 0 is the lower and 0.2 is the upper bound (optional)

ML estimation in Dynare: steady state

- steady state calculated for many different values of Ψ!
- solve for the steady state yourself (linearizing makes it easier)
- give the exact steady state to Dynare for the initial values
- option to provide own function that calculates steady state!
  - `modfilename_steadystate.m` or
  - `steady_state_model;` block
- have not specified it in the above, why?

ML estimation in Dynare: estimation command

- then also tell Dynare which are the observable variables

```
varobs y;
estimation(options)[VARIABLE_NAME];
```

- options include
  - specify data file for estimation: `datafile=data`
  - assess convergence to max: `mode_check`
  - optimization options: `optim(options)`
  - many more!

ML estimation in Dynare: decomposition

- decompose endogenous variables into contribution of shocks
- possible also after `stoch_simul`

```
shock_decomposition(options) [VARIABLE_NAME];
```

- options include e.g. `parameter_set`
  - use calibrated values: `=calibration`
  - use prior/posterior mode: `=prior_mode`/`=posterior_mode`
- variables specifies for which variables to run the decomposition

ML estimation of neoclassical growth model

- use neoclassical growth model as data generating process
- 265 observations of output
- use ML to estimate
  - σ
  - σ, ρ, δ, α

ML estimation of neoclassical growth model

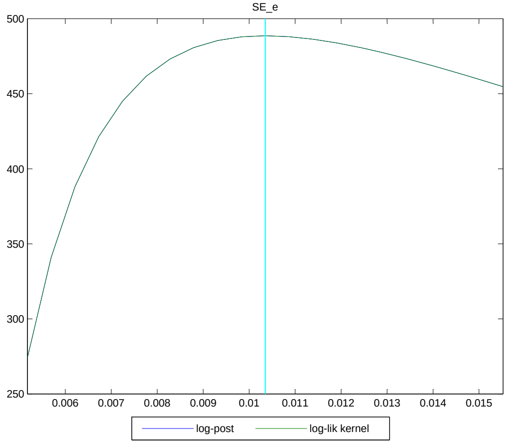
Estimating only σ:

```
estimated_params;
stderr e, 0.01, 0, 0.2;
end;
varobs y; estimation(datafile=y,mode_check) c, k, y;
```

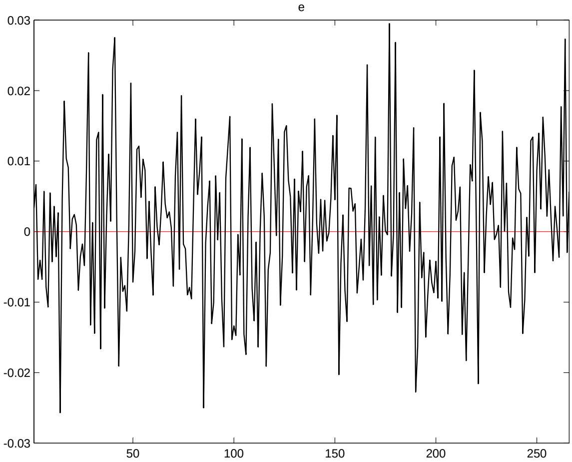
Results:

σ 0.0103 (0.0004)

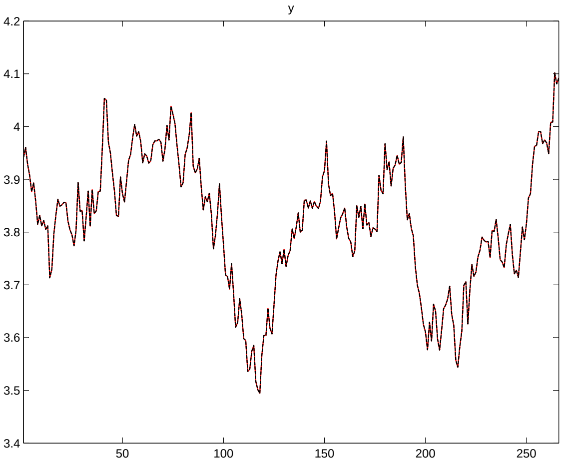
ML estimation of easy case: mode check



ML estimation of easy case: shocks



ML estimation of easy case: fitted values



ML estimation of neoclassical growth model

Estimating only σ:

```
estimated_params;
stderr e, 0.1, 0, 0.2;
end;
varobs y; estimation(datafile=y,mode_check) c, k, y;
```

Results:

σ 0.0103 (0.0004)

ML estimation of neoclassical growth model

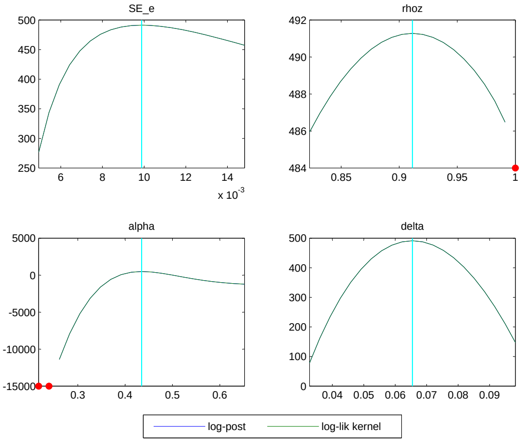
Estimating more than just σ:

```
estimated_params;
stderr e, 0.01, 0, 0.2;
rho, 0.95, 0, 1;
alpha, 0.36, 0, 1;
delta, 0.025, 0, 0.2;
end;
varobs y; estimation(datafile=y,mode_check) c, k, y;
```

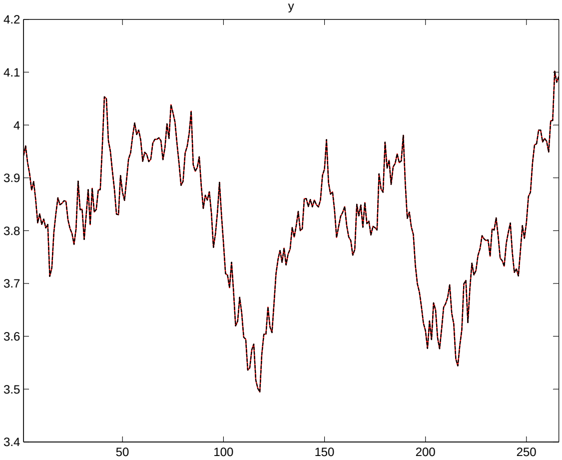
Results:

σ	0.0099 (0.0004)	ρ	0.9114 (0.0676)
α	0.4349 (0.1290)	δ	0.0655 (0.0921)

ML estimation of tough case: mode check



ML estimation of tough case: fitted values



ML estimation of neoclassical growth model

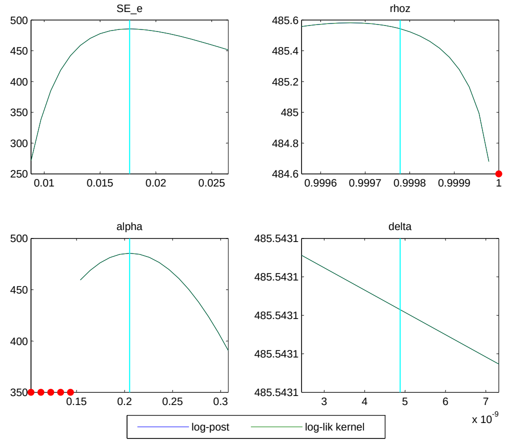
Estimating more than just σ:

```
estimated_params;
stderr e, 0.1, 0, 0.2;
rho, 0.95, 0, 1;
alpha, 0.36, 0, 1;
delta, 0.025, 0, 0.2;
end;
varobs y; estimation(datafile=y,mode_check) c, k, y;
```

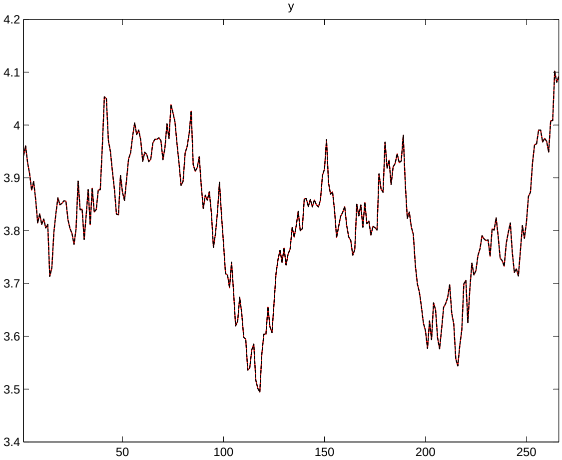
Results:

σ	0.0177 (0.0046)	ρ	0.9998 (0.0005)
α	0.2053 (0.0444)	δ	0.0000 (0.0003)

ML estimation of tough case: mode check



ML estimation of tough case: fitted values



ML estimation of neoclassical growth model

- observed data consistent with many parameter combinations
- the likelihood is typically quite flat
- $\rightarrow$ Maximum Likelihood has trouble converging
- solutions:
  - sometimes alternative optimization algorithms work
  - use of extra (prior) information $\rightarrow$ Bayesian estimation

What we did today

- looking at the big picture
  - what you'll do in the next days
  - what we'll do today and Friday
- intro into estimation
  - Kalman filter (based on Hamilton)
  - Maximum Likelihood estimation
- estimating DSGE models
  - DSGE and time-series models
  - Maximum Likelihood in Dynare

Extensions

Extensions

- Kalman smoother
- non-linear filter
- missing observations
- allowing for exogenous regressors
- allowing for covariance between w_t and v_{t+1}

Kalman smoother

- main idea: one can use more information for forecasting states
- Kalman filter uses information up until the current period
- one can also use information beyond the period of the state
- objective is to calculate $\widehat{\zeta}_{t|T} = \widehat{E}[\zeta_t|\mathcal{Y}^T]$
- where $\mathcal{Y}^T = (y_1, ..., y_{t-1}, y_t, ..., y_T)$
- again uses linear projections

Smoothing recursions

- first run the Kalman filter and obtain $\widehat{\zeta}_{t|t}$, $P_{t|t-1}$ and $P_{t|t}$
- start from the end of the sample
- the smoothing recursions are:

$$\widehat{\zeta}_{t|T} = \widehat{\zeta}_{t|t} + J_t(\widehat{\zeta}_{t+1|T} - \widehat{\zeta}_{t+1|t})$$

$$P_{t|T} = P_{t|t} + J_t(P_{t+1|T} - P_{t+1|t})$$

$$J_t = P_{t|t}F'P_{t+1|t}^{-1}$$

A nonlinear state-space

Up until now, we assumed a linear state-space

$$
\begin{aligned}
y_t =& H'\zeta_t + w_t, & \mathbb{E}(w_t, w_t') = R \quad \forall t \\
\zeta_{t+1} =& F\zeta_t + v_{t+1}, & \mathbb{E}(v_t, v_t') = Q \quad \forall t
\end{aligned}
$$

However, non-linear forms can easily arise:

- higher-order solutions to DSGE models
- also in reduced-form empirical work

$$
\begin{aligned}
y_t =& h(\zeta_t) + w_t, & \mathbb{E}(w_t, w_t') = R \quad \forall t \\
\zeta_{t+1} =& f(\zeta_t) + v_{t+1}, & \mathbb{E}(v_t, v_t') = Q \quad \forall t
\end{aligned}
$$

Extended Kalman filter

- the idea behind the Extended Kalman filter is simple
- use a 1st-order Taylor expansion at each point in time

Extended Kalman filter recursions

update:

$$\begin{aligned}\widehat{\zeta}_{t|t} &= \widehat{\zeta}_{t|t-1} + P_{t|t-1} H_t (H_t' P_{t|t-1} H_t + R)^{-1} (y_t - h(\widehat{\zeta}_{t|t-1})) \\ P_{t|t} &= P_{t|t-1} - P_{t|t-1} H_t (H_t' P_{t|t-1} H_t + R)^{-1} H_t' P_{t|t-1}'\end{aligned}$$

forecast:

$$\begin{aligned}\widehat{\zeta}_{t+1|t} &= f(\widehat{\zeta}_{t|t}) \\ P_{t+1|t} &= F_t P_{t|t} F_t' + Q\end{aligned}$$

where F_t and H_t are Jacobian matrices:

$$\begin{aligned}F_t &= \frac{\partial f}{\partial \widehat{\zeta}}|_{\widehat{\zeta}_{t|t}} \\ H_t &= \frac{\partial h}{\partial \widehat{\zeta}}|_{\widehat{\zeta}_{t|t-1}}\end{aligned}$$

Missing observations

- the Kalman filter also conveniently handles
    - missing observations
    - mixed-frequency data
- the idea is that in periods of no observations
    - the Kalman gain $K_t = 0$
    - the "prediction error" $y_t - \widehat{y}_{t|t-1} = 0$
- careful with mixed-frequency data
    - average?
    - sum?

Allowing for regressors

- up until now we assumed that observations and states depend
    - only on the states themselves
- however, they may depend on other observables

Time-series model with exogenous regressors

$$\begin{aligned} y_t &= H'\zeta_t + Ax_t + w_t, & \mathbb{E}(w_t, w_t') = R \quad \forall t \\ \zeta_{t+1} &= F\zeta_t + Gx_t + v_{t+1}, & \mathbb{E}(v_t, v_t') = Q \quad \forall t \end{aligned}$$

- where x_t are observable (explanatory) variables

Kalman recursions with explanatory variables

The combined Kalman filter recursions become:

$$
\begin{aligned}
\widehat{\zeta}_{t+1|t} =& F\widehat{\zeta}_{t|t-1} + Gx_t + K_t(y_t - Ax_t - H'\widehat{\zeta}_{t|t-1}) \\
K_t =& FP_{t|t-1}H(H'P_{t|t-1}H + R)^{-1} \\
P_{t+1|t} =& FP_{t|t-1}F' + Q \\
& - FP_{t|t-1}H(H'P_{t|t-1}H + R)^{-1}H'P_{t|t-1}F'
\end{aligned}
$$

Covariance between innovations

- up until now we assumed that $cov(w_t, v_{t+1}) = 0$
- i.e. that innovations to the states ...
- are independent of observation equation innovations
- here we allow them to covary: $\mathbb{E}[w_t, v_{t+1}] = C$
- in other words $\mathbb{E}[(w_t, v_{t+1})(w_t, v_{t+1})'] = \begin{pmatrix} R & C \\ C' & Q \end{pmatrix}$

Kalman recursions with $C \neq 0$

The combined Kalman filter recursions become:

$$\widehat{\zeta}_{t+1|t} = F\widehat{\zeta}_{t|t-1} + K_t(y_t - H'\widehat{\zeta}_{t|t-1})$$

$$K_t = (FP_{t|t-1}H + C)(H'P_{t|t-1}H + R)^{-1}$$

$$\begin{aligned} P_{t+1|t} = & FP_{t|t-1}F' + Q \\ & - (FP_{t|t-1}H + C)(H'P_{t|t-1}H + R)^{-1}(FP_{t|t-1}H + C)' \end{aligned}$$