DATA 5610 4/18/2022

- Readings:
  - Elliot & Timmermann Chp. 6
  - James et al (ISLR) Chps. 5 - 6
  - Chernick & LaBude Chp. 5
  - Sheppard notes
  - Hansen JBES, 2005

Bootstrapping Time-Series Data

See: Sheppard notes AFE Week 1 p. 26/42 (13)

- IID bootstrap is only appropriate for IID data
  - NB: may be okay for data not serially correlated
- Two strategies:
  - Parametric & IID bootstrap
    - If you have a model w/ IID residuals
    - Then you could bootstrap the residuals to generate new draws
    - "Historical filter": filter the new draws through the parametrically estimated model

EX.: AR(1)

$$Y_t = \phi Y_{t-1} + \varepsilon_t$$

$$\hat{\varepsilon}_t = Y_t - \hat{\phi} Y_{t-1}$$

obtain $\hat{\phi}$ via OLS/MLE

IID residuals

- Nonparametric block bootstrap:
  - Weak assumptions: basically that blocks can be sampled so that they are approximately IID

- Go to notebook for simulation of AR model

- $Y_t = 0.5\, Y_{t-1} + \varepsilon_t$ s.t. $\varepsilon_t \sim N(0,1)$

  $\phi \overset{a}{\sim} N(0, 1/\sqrt{n})$ via CLT

  - Q: can we recover this value?

Moving Block Bootstrap

- Samples blocks of m consecutive observations
- uses blocks which start at indices $1, \dots, T-m+1$

Algorithm

1. Initialize $i=1$
2. Draw a uniform integer v_i on $1, \dots, T-m+1$
3. Assign $u_{(i-1)+j} = v_i + j - 1$ for $j = 1, \dots, m$
4. Increment i and repeat 2-3 until $i \geq T/m$
5. Trim u so that only the first T remain if T/m is not an integer

* See Sheppard's Matlab code (which should be easily translated to Python)

Circular Bootstrap

- simple extension of MBB which assumes the data live on a circular buffer so that $Y_{T+1} = Y_1$, $Y_{T+2} = Y_2$, etc.

- Has better finite sample properties since all data points get sampled with equal probability

- only step 2 changes in a very small way

Algorithm

1. Initialize $i = 1$
2. Draw ~~[illegible]~~ a uniform integer v_i on $1, \ldots, T$
3. Assign $u_{(i-1)+j} = v_i + j - 1$ for $j = 1, \ldots, m$
4. Increment i and repeat 2-3 until $i \geq T/m$
5. Trim u so that only the first T remain if T/m is not an integer

* NB: See Matlab code

Stationary Bootstrap (Politis & Romano 1992)

- Differs from MBB and CBB in that the block size is no longer fixed
- Chooses an average block size of m rather than an exact block size
- Randomness in block size is worse when m is known, but helps if m may be suboptimal
- $m \sim Exp(\lambda)$

Algorithm

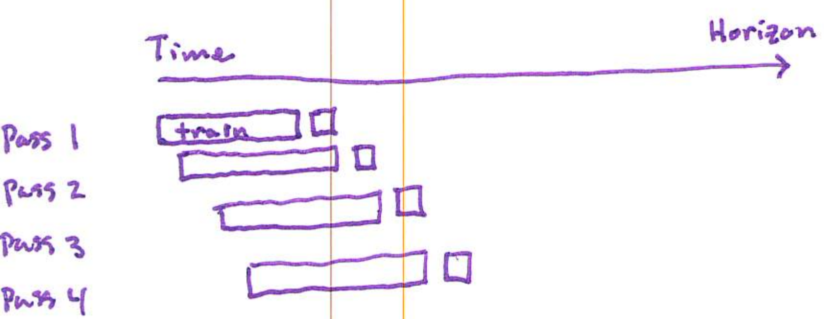
1. Draw uniform u_1 on $1, \dots, T$
2. For $i = 2, \dots, T$
   a. Draw a uniform v on $(0,1)$
   b. if $v \geq 1/m$ then $u_i = u_{i-1} + 1$
      i. if $u_i > T$, then $u_i = u_i - T$
   c. If $v < 1/m$, draw u_i uniform on $1, \dots, T$

NB: See jupyter notebook

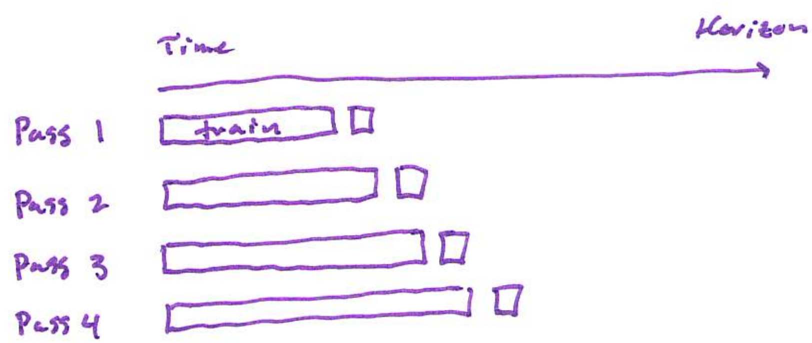
Expanding Window Analysis

- Sliding/Moving Window vs. Expanding Window

Sliding Window



Expanding Window



NB: the window size in sliding window analysis and the training size in both introduce new "tuning parameters"

1-Step Ahead Returns Forecasts

- Recall that we are working w/ the Power Utility function for investors $\gamma = (2, 5, 9)$ where

$$u(c) = \frac{c^{1-\gamma}}{1-\gamma}$$

- The input variable 'c' usually stands for consumption or wealth
- In this application we will use gross returns

$$1 + y_{t+1} S_k(x_t, \beta_k)$$

where

$$x_t = \{X_{t-i}\}_{i=0}^{R}$$

where R is the length of the expanding window period

See STW p. 1651 for hints

$$Y_{t+1} = \frac{P_{t+1} - P_t}{P_t}$$ with P_t = ~~Price~~ level of asset at time t

$S_k(\cdot)$ is a "signal" function that converts a prediction into a market position. It takes two args:

1. Historical training data from $t = 1, \ldots, R$
2. β_k from trained model k

It produces one of the following values:

1 : long position (buy)

0 : hold position (do nothing)

-1 : short position (sell)

- Hansen's SPA test looks for a model with lower loss than a benchmark

- Convert utility to loss as:

$$L_k(y_{t+1}, S_k(x_t, \beta_k)) = -U[1 + y_{t+1} S_k(x_t, \beta_k)]$$

- For the benchmark use a long-only strategy

$$1 + y_{t+1} S_0(x_t, \cdot)$$

where $S_0(x_t) = 1$ always